TUM School of Engineering and Design

Data-Driven Feature Learning with Discriminative Models for Satellite Time Series

Marc Rußwurm

Vollständiger Abdruck der von der TUM School of Engineering and Design
der Technischen Universität München zur Erlangung des akademischen Grades
eines Doktors der Ingenieurwissenschaften
genehmigten Dissertation.

Vorsitz: Prof. Dr.-Ing. habil. Xiaoxiang Zhu

Prüfende/-r der Dissertation:

1. Priv.- Doz. Dr. rer. nat. habil. Marco Körner
2. Prof. Dr.-Ing. habil. Richard Bamler
3. Prof. Devis Tuia, Ph.D.

Die Dissertation wurde am 30.08.2021 bei der Technischen Universität München eingereicht und durch die TUM School of Engineering and Design am 31.01.2022 angenommen.

Contents

Abstract

Neural networks are flexible functions that can, by approximation theorem, learn any function over a local training dataset. This includes learning preprocessing and feature extraction functions alongside finding a final decision boundary in a feature space. In practice, however, no universal learner exists and estimation and optimization errors increase when we use neural networks to approximate increasingly complex relationships. Hence, model designers use their prior knowledge about the particular task and the distribution of the data to help their model obtain good solutions on a targeted family of problems.

The family of problems that are targeted in this dissertation is the classification of the crop types on agricultural field parcels in Europe from sequentially acquired optical satellite imagery. The central questions investigated in this work are:

1. how can we induce our prior knowledge into deep model architectures for satellite time series classification and crop type mapping?
2. how can we augment existing deep learning architectures to estimate model and data uncertainties for satellite time series forecasting?
3. how can we address domain shift in data-distributions induced by temporal and regional variability of representations on the Earth's surface?

The central contributions of this work towards these questions are:

1. a comparison of model- and data-driven methods on raw and pre-processed datasets and the analysis of self-attention mechanisms on raw satellite time series data (Rußwurm and Körner, 2020),
2. a centralized crop type mapping benchmark dataset for the comparison of state-of-the-art convolution, recurrent, and attention networks for crop type mapping (Rußwurm et al., 2020),
3. an evaluation of two methods to estimate model and data uncertainty for satellite time series forecasting (Rußwurm et al., 2020a).
4. two contributions towards learning models on datasets-of-tasks with an algorithm from few-shot meta-learning. This addresses the shift in data distribution on globally distributed remote sensing data for satellite image land cover classification and segmentation (Rußwurm et al., 2020b), as well as time series classification (Wang et al., 2020).

1. Neural Networks and Data-driven Feature Learning

Artificial neural networks are loosely inspired by biological neurons and have been used in machine learning with varying popularity over decades. Individual layers $f_{\mathbf{W}}(\mathbf{x}) : \mathbb{R}^N \mapsto \mathbb{R}^M$ transform an input vector $\mathbf{x}^T = (x_0, x_1, \dots, x_N)$ to an output $\mathbf{y}^T = (y_0, y_1, \dots, y_M)$ with a linear projection $\mathbf{W} \in \mathbb{R}^{(N+1)\times M}$ followed by an elementwise applied non-linear activation function $\sigma : \mathbb{R} \mapsto \mathbb{R}$. A dense neural network layer is formalized as

$$y_i = f_{\mathbf{W}}(\mathbf{x}) = \sigma(\mathbf{w}_i\mathbf{x}) = \sigma\left(\sum_{j=0}^{N} w_{ji}x_j + b_i\right) \tag{1.1}$$

where a translational bias term $b_i = w_{(N+1),i}$ can be also included in the last weight row $N+1$ on a 1-concatenated input vector $(\mathbf{x}^T, 1)$.

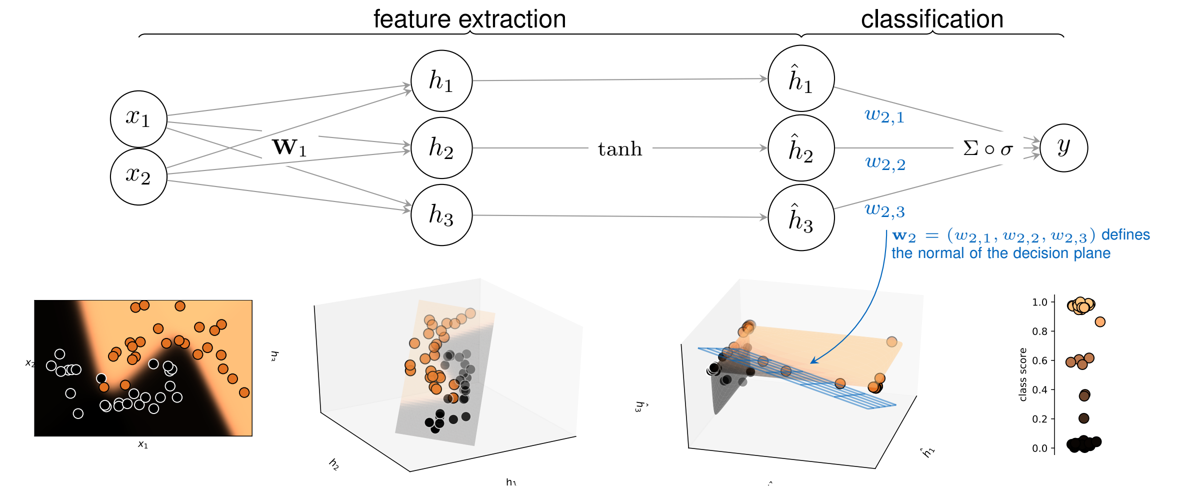
Let's consider two interpretations of Eq. (1.1). From the *biological* view-point, a neuron receives signals $\mathbf{x}$ (over time) and stores the $\mathbf{w}_i$-weighted sum internally. The neuron itself transmits a signal if this sum exceeds a threshold defined by the activation function σ. This is reflected in early activation functions, such as the Heaviside step function $1_{x>0}(x)$ and the logistic function $\sigma(x) = (1 + e^{-x})^{-1}$ as its smooth approximation. From the *linear algebraic* perspective, a neural network layer is composed of the two functions $f_{\mathbf{W}} = \mathbf{W} \circ \sigma$: a linear projection $\mathbf{W}$ and a non-linear transformation $\sigma(\cdot)$ where the $\circ$-operator applies functions sequentially. An input vector $\mathbf{x}$ is projected from $\mathbb{R}^N$ into a (often higher-)dimensional space $\mathbb{R}^H$ and non-linearly transformed by the activation function σ. In a neural network,

$$f_{\mathrm{W}} = f_{\{\mathbf{W}_1, \mathbf{W}_2, \dots, \mathbf{W}_L\}} = f_{\mathbf{W}_1} \circ f_{\mathbf{W}_2} \circ \dots \circ f_{\mathbf{W}_L} \tag{1.2}$$

L cascaded layers transform the input into increasingly higher-level representations. Each layer projects the input space into a $H^{(l)}$-dimensional *feature* space where $H^{(l)}$ can vary for each respective layer l. The number of layers L and the dimensionality of the feature space are typically termed the *depth* and *width* of the network as they linearly and quadratically deterimine the number of parameters. Figure 1 shows an example of linear transformations in a higher-dimensional feature space on a classification task.

1.1. Universal Function Approximaton

A neural network of arbitrary width (Cybenko, 1989; Hornik et al., 1989; Haykin, 2007; Hassoun et al., 1995) or depth (Lu et al., 2017; Kidger and Lyons, 2020) can approximate any target function $f(\mathbf{x})$ of inputs $\mathbf{x} \in \mathcal{X}$. Formally, if the space of neural network weights $\mathcal{A}$ is



(a) Input space $\mathbf{x} = (x_1, x_2)$

(b) linear projection $\mathbf{w}_1^T \mathbf{x}$ into $\mathbb{R}^3$

(c) non-linear distortion with $\tanh(\mathbf{w}_1^T \mathbf{x})$

(d) Output class probabilities $y \in [0, 1]$

Figure 1: This figure visualizes the hidden projections of the 2-layer neural network $y = \sigma(\mathbf{w}_2^T \tanh(\mathbf{W}_1^T \mathbf{x}))$ on a classification toy example. In (a), the input point coordinates $\mathbf{x} = (x_1, x_2) \in \mathbb{R}^2$ are drawn. The associated labels are indicated by color. In (b), this $\mathbb{R}^2$ input space is then linearly projected into $\mathbb{R}^3$ by the projection matrix $\mathbf{W}_1 \in \mathbb{R}^{2\times 3} : \mathbb{R}^2 \mapsto \mathbb{R}^3$. The translation bias terms are omitted for clarity. Figure (c) shows this feature space after it is non-linearly distorted by the $\tanh$ activation function. This projection into $\mathbb{R}^3$ and the activation function distorted the feature space such that a linear decision plane can be defined by its normal vector $\mathbf{w}_2$ as drawn in blue. The distance of individual data-points to this decision plane produces probabilities after being squeezed into the $[0, 1]$ interval by the sigmoid function σ, as shown in (d).

unlimited in width or depth, there exists a neural network f_{w} with $\mathrm{w} \in \mathcal{A}$ that satisfies

$$\int_{\mathcal{X}} |f(\mathbf{x}) - f_{\mathcal{A}}(\mathbf{x})| \, \mu(\mathrm{d}\mathbf{x}) < \varepsilon_{\text{app}} \tag{1.3}$$

with an infinitesimal small error ε_{app}. Here, we measure the approximation error over a probability distribution $\mu(\mathcal{X})$ on the input space $\mathcal{X}$ using the L1-norm or absolute error.

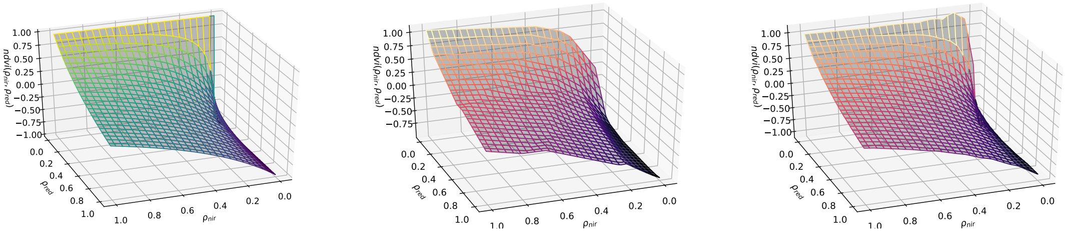
In practice, the space of neural network weights $\mathcal{W} \subset \mathcal{A}$ is restricted by the number of layers or dimensions and the approximation error

$$\varepsilon(\mathrm{w}) = \int_{\mathcal{X}} |f(\mathbf{x}) - f_{\mathrm{w}}(\mathbf{x})| \, \mu(\mathrm{d}\mathbf{x}) \tag{1.4}$$

is a function of weights $\mathrm{w} \in \mathcal{W}$. The best approximation

$$\varepsilon_{\text{app}} = \inf_{\mathrm{w} \in \mathcal{W}} \varepsilon(\mathrm{w}) \tag{1.5}$$

is obtained with the set of weights $\mathrm{w} \in \mathcal{W}$ that minimizes ε. In a general sense (Shalev-



(a) NDVI target function (b) approximation $h = 16$ (c) approximation $h = 64$

Figure 2: The target NDVI function and two neural network approximations of different number of neurons and layers. The approximation error decreases with lhe larger the weight space $\mathcal{W}$.

Shwartz and Ben-David, 2014), w represents a hypothesis in a (restricted) hypothesis space $\mathcal{W}$. We choose the hypothesis that minimizes the *risk* $\varepsilon(\mathrm{w})$ on the *objective function* ε.

Example: Normalized Difference Vegetation Index

Let's illustrate this approximation on a concrete function that is commonly used in remote sensing: the Normalized Difference Vegetation Index (NDVI) (Tucker, 1979)

$$f(\mathbf{x}) = \text{NDVI}(\rho_{\text{red}}, \rho_{\text{nir}}) = \frac{\rho_{\text{nir}} - \rho_{\text{red}}}{\rho_{\text{nir}} + \rho_{\text{red}}}. \tag{1.6}$$

Here, the input vector $\mathbf{x}^T = (\rho_{\text{nir}}, \rho_{\text{red}})$ contains the reflectances ρ in the red (625 – 700 nm) and near infrared spectrum (750 nm – 900 nm). Reflectances range from no reflectance $\rho = 0$ to complete reflectance $\rho = 1$ which defines the data space $\mathcal{X} = [0, 1]^2$. The NDVI has been designed as a feature for vegetation analysis, as it measures the slope of the absorption in the red spectrum contrasted by high reflectance in near-infrared of photosynthetically active vegetation, as detailed later in Section 2.2.

Let's modify the 2-layer neural network of Fig. 1 to approximate the NDVI function. As regression problem, we can remove the outer sigmoid function. We also replace the $\tanh$ activations with ReLU$(x) = \max(x, 0)$ function for illustration purposes. Now, the neural network can be written in closed form as

$$\hat{y}_{\text{NDVI}} = \mathbf{w}_2^T \max(\mathbf{W}_1^T \mathbf{x}, 0), \tag{1.7}$$

where $\mathbf{x}^T = (\rho_{\text{nir}}, \rho_{\text{red}}, 1)$, $\mathbf{W}_1 \in \mathbb{R}^{(2+1)\times h}$, and $\mathbf{w}_2 \in \mathbb{R}^{(h+1)}$. We can increase the width of the network by increasing the number of hidden dimensions h. This increases the number of intermediate feature dimensions. In Fig. 2, we show the target NDVI function along with two neural network approximations using with $h = 16$ and $h = 64$ different hidden dimensions. Increasing the hidden dimensions leads better approximations. On the $h = 16$ approximation, one can see that the ReLU activation function allows the neural network to approximate the non-linear NDVI function by piecewise linear planes.

1.2. Approximation, Estimation, and Optimization Errors

Let's continue with the theory on neural network approximation and take an additional step towards concrete application by modifying Eq. (1.4) to the discrete case

$$\varepsilon(\mathrm{w};\mathcal{D}) = \frac{1}{|\mathcal{D}|}\sum_{\mathbf{x}_i\in\mathcal{D}} |f(\mathbf{x}_i) - f_{\mathrm{w}}(\mathbf{x}_i)| \tag{1.8}$$

where we evaluate the mean absolute error between target function f and neural network f_{w} on a dataset $\mathcal{D} \sim \mu(\mathcal{X})$ and choose the weights

$$\mathrm{w}^\star = \arg\min_{\mathrm{w}\in\mathcal{W}} \varepsilon(\mathrm{w};\mathcal{D}) \tag{1.9}$$

that minimize the objective function $\varepsilon(\mathrm{w};\mathcal{D})$ given a training dataset $\mathcal{D}$ sampled from a distribution μ over the input space $\mathcal{X}$.

estimation error. The dataset $\mathcal{D}$ should optimally represent the entire distribution $\mu(\mathcal{X})$. So, naturally, we would like to sample a large dataset $\mathcal{D}$ to find model weights that result in a low estimation error

$$\varepsilon_{\text{est}} = \varepsilon(\mathrm{w};\mathcal{D}) - \varepsilon_{\text{app}} \tag{1.10}$$

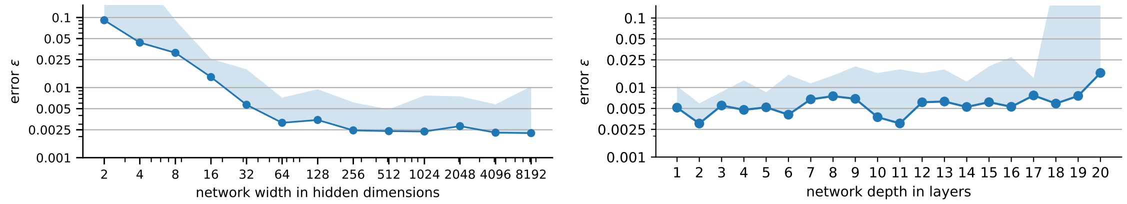
between the evaluated error $\varepsilon(\mathrm{w};\mathcal{D})$ given a sampled dataset $\mathcal{D}$ and the theoretical approximation error ε_{app} from the data distribution $\mu(\mathcal{X})$ defined in Eq. (1.5).

This means that, given a dataset $\mathcal{D} \sim \mu(\mathcal{X})$, we can only measure the joint error

$$\varepsilon(\mathrm{w};\mathcal{D}) = \varepsilon_{\text{est}} + \varepsilon_{\text{app}} \tag{1.11}$$

which is also known as *empiric risk*, *bias*, or *training loss* of a model on the dataset $\mathcal{D}$.

bias-variance tradeoff. In practice, however, we are restricted in the number of samples we can draw. Let's sample a training dataset $\mathcal{D}_{\text{train}} \sim \mu(\mathcal{X})$ and determine our model weights w based on the *empiric risk* $\varepsilon(\mathrm{w};\mathcal{D}_{\text{train}})$ using Eq. (1.9). If we now obtain a new dataset $\mathcal{D}_{\text{test}} \sim \mu(\mathcal{X})$, we can use our model f_{w} to measure the *expected risk* $\varepsilon(\mathrm{w};\mathcal{D}_{\text{test}})$. A model with large weight space (e.g., large in depth and/or width) trained on a small dataset can estimate a low empiric risk by fitting the training data well. The same model, however, may have a high expected risk on the unseen test data. In other words, we *overfit* to the training data and fail to *generalize* on the unseen test data even though both training and test data were drawn from the same distribution. Let's say we randomly sample multiple datasets and train/test on different partitions. We would like our *expected risks* to be similar to each other and have a low *variance*. The balance between empiric and expected risk is known in the literature as *bias-variance* or *bias-complexity* tradeoff (Shalev-Shwartz and Ben-David, 2014).



(a) varying the neural network width with a single layer neural network as in Eq. (1.7)

(b) varying the neural network depth with fixed width $h = 64$

Figure 3: This figure shows the evaluated error when approximating the NDVI target function with neural networks of varying depth (a) and width (b). We trained the neural network with 5 different weight initializations and plot the best approximation as blue line. The shaded area reflects the range between best and worst approximations.

optimization error. Additionally, finding a global optimum with an analytic solution of Eq. (1.5) is not tractable due to the non-convex nature of the optimization problem caused by the non-linearity of the activation function. Hence, a gradient descent (GD) algorithm

$$\mathrm{w}^* = \mathsf{GD}(\varepsilon(\mathrm{w};\mathcal{D}), \mathrm{w}_{\text{init}}) \approx \arg\min_{\mathrm{w}\in\mathcal{W}} \varepsilon(\mathrm{w};\mathcal{D}) \tag{1.12}$$

is used that iteratively updates a weight initialization $\mathrm{w}_{\text{init}} \sim p(\mathcal{W})$ sampled from a distribution p over the weight space $\mathcal{W}$. This gradient descent algorithm depends on several a-priori chosen parameters, such as the choice of the algorithm itself, learning rate, or weight initialization and is an approximation of the true risk optimum of Eq. (1.9). This introduces an additional *optimization error* ε_{opt} which means accordingly that, in practice, only the combined error

$$\varepsilon(\mathrm{w}^*;\mathcal{D}) = \varepsilon_{\text{app}} + \varepsilon_{\text{est}} + \varepsilon_{\text{opt}} \tag{1.13}$$

on a concrete dataset $\mathcal{D}$ is accessible to us.

The approximation error will decrease with larger neural networks which is consistent with the universal approximation theorem on Section 1.1. The estimation error depends on the size of the dataset and decreases on larger datasets that represent the data distribution well. The optimization error depends simultaneously on the choice of gradient descent algorithm and the difficulty of the optimization problem posed by the neural network model.

Continued Example: Normalized Difference Vegetation Index

Let's continue the previous example of NDVI approximation and systematically increase the neural network in depth (number of layers) and width (number of neurons in each layer) and observe how closely we can approximate the target NDVI function.

In this toy example, we have access to the target function and can uniformly sample form the data distribution $\mathcal{D} \sim \mathcal{U}(\mathcal{X})$ repeatedly without restrictions. Hence, we can assume that our estimation error $\varepsilon_{\text{est}} \approx 0$ is negligible and the observed error $\varepsilon = \varepsilon_{\text{app}} + \varepsilon_{\text{opt}}$ contains approximation and optimization components only. Following the approximation theorem, we expect that the approximation error decreases when adding layers l (increasing depth) or hidden dimensions h (increasing width).

In Fig. 3a, we can observe the overall error decreases when we increase the width until $h = 64$ (257 weights in total). However, when increasing the width further until $h = 8192$ (32689 weights), we are only able to improve upon the performance marginally.

When we increase the depth with a fixed width of $h = 64$ in Fig. 3b, we observe that the error actually increases beyond 2 layers. At first glance, this seems to contradict the universal approximation theorem of neural networks since, in this example, deeper models lead to poorer approximations. However, deeper neural networks also lead to increasingly complex intermediate features. This poses a highly non-convex optimization problem where the gradient descent algorithm converges on a sub-optimal local minimum instead of the desired global one. This increases the optimization error ε_{opt} with network depth and in-effect leads to poorer approximations. In the figure, we see the variance in optimization error in the shaded area which indicates the best and worst approximation on different weight initialization w_{init}.

In this example, we saw that an increasingly complex neural network can approximate a target function initially well. However, when adding more neural network layers, we trade a lower approximation error with increasing optimization errors caused by the non-convexity of the optimization problem.

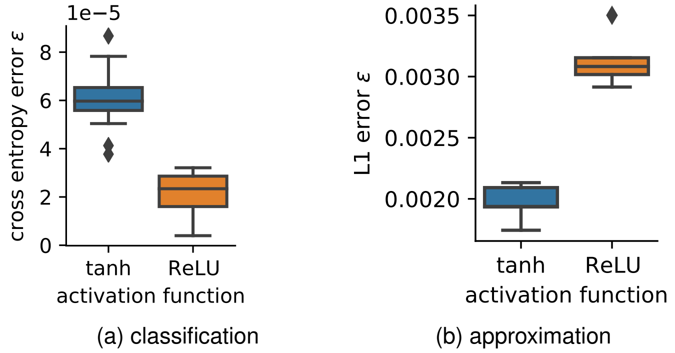
Inductive biases on previous toy examples

In general, for an arbitrary optimization problem, such as the previous NDVI approximation or the toy classification, we are not guaranteed to achieve a good solution by only increasing the space of hypotheses, i.e., weight-space $\mathcal{W}$, from which we choose our model w^* using a learning algorithm, i.e., gradient descent.

This is the notion of the **No Free Lunch Theorem(s)** (Wolpert and Macready, 1997), that briefly state that: "Averaged over all optimization problems, without re-sampling all optimization algorithms perform equally well" (Adam et al., 2019).

Since no universal learner for all problems exists, we must use *prior knowledge* about the task to succeed (Shalev-Shwartz and Ben-David, 2014). In other words, we restrict the space hypotheses $\mathcal{W}$ based on our prior knowledge on the problem before the learner sees the training data. This *induces* a *bias* to our learning problem which is called *inductive bias*.

So, how can we include prior knowledge when approximating target functions? From the previous chapter and Eq. (1.13), we have seen that we can decrease the overall error by reducing approximation, estimation, or optimization errors. However, many design choices affect multiple error components. For instance, when reducing the approximation error by choosing an increasingly complex model, we may increase our estimation error which may



(a) classification (b) approximation

Figure 4: Effect of the activation function on the classification and approximation toy example tasks of this chapter

perform poorly on an unseen dataset (high variance).

Let's loosely gather a non-exhaustive list of inductive biases: Primarily, we can choose an objective function that best reflects the problem at hand. Also, we can restrict the hypothesis space by choosing an appropriate model architecture. For instance, we have control over the width and depth of a neural network, the activation function, the type of connections, such as convolutions, recurrence, attention or dense layers. Additionally, we can add inter-layer normalization, such as BatchNorm or LayerNorm. To reduce the estimation error and optimization errors specifically, we can add regularization to the objective function, such as weight-decay as L2-regularization. Also, we can add dropout layers to the neural network architecture or (artificially) increase the data diversity through data augmentation. Also, we can tune the optimization algorithm by choosing the complexity of the gradient descent algorithm, for instance, by including momentum terms (Kingma and Ba, 2015). Finally, we can choose the model initialization by changing the distribution of how we sample our initial weights from the weight space as a prior over the hypothesis space.

We employed a neural network to two seemingly unrelated tasks: In Fig. 1, we performed classification on a toy dataset. In the previous two chapters, we approximated the NDVI target function. Let's investigate the effect of the activation function on both of these tasks. We used a 2 layer neural network with 16 hidden dimensions for both tasks and only vary the activation function. We train 20 different weight initializations which result in the distributions shown in Fig. 4. One can see that the ReLU function was a better choice for the classification task, while the neural network with hyperbolic tangens function could approximate the normalized difference vegetation index better. With our prior knowledge about the nature of these functions, we could argue that the smooth hyperbolic tangens function may be better suited for approximating the continuous NDVI function. In contrast, the ReLU function $\max(x, 0)$ ultimately approximates functions by piecewise linear segments which was well-suited for a regression problem in this example. Conversely, for the classification tasks, piecewise linear decision boundaries, as in the ReLU, may be a good choice when separating data into categories.

1.3. Inductive Biases in Deep Model Architectures

In the previous sections, we saw how neural networks can be used to approximate any target functions up to a certain error composed of approximation, estimation, and optimization components. Still, no universal learner exists and prior knowledge on a target learning task is necessary to achieve good results.

Identifying inductive biases for particular distributions of tasks is a central research objective in many machine learning fields. For instance, ResNets (He et al., 2016) are ubiquitous in computer vision since they excel at extracting spatial features from images. Self-attention modules in transformer networks (Vaswani et al., 2017) are the state-of-the-art in natural language processing today.

The primary task explored in this dissertation is finding the mapping function from a temporal sequence of satellite images to categories of agricultural crop type classes. In contrast to prior literature which focused on finding a functional relationship in a model-driven way through heavy data preprocessing and using hand-crafted features like the NDVI, we explore in this work how neural networks can approximate this mapping solely in a data-driven way.

Let's use the remainder of this section to outline the differences of satellite time-series data to common tasks in machine learning and in particularly the fields of computer vision and natural language processing. The central difference between algorithms used in these fields is that they are designed for a certain *expected structure in the data*. We can categorize this in *spatial*, *spectral*, and *temporal* structure. Natural images in **computer vision** show a strong spatial autocorrelation. Neighboring pixels depend on each other and relevant (spatial) features range over multiple pixels or even the entire image. The spectral information contains usually only three bands and provides some hints about the particular object, but is less relevant compared to the spatial component. The temporal dimension becomes relevant for video analysis where individual frames are usually highly correlated. It is often sufficient to enforce a certain consistency between neighboring video frames. Convolutional neural networks are particularly useful for these types of tasks since they exploit a structural consistency between neighboring data points by fixed-size convolutional kernels. In **natural language processing**, the sequential characteristics are highly relevant. In contrast to pixels in an image, language does not follow a strict local sequential order. Relevant words for the meaning of a sentence may be located in various parts of the sequence with words of less significance placed in between. In this field, first recurrence-based and later attention-based layers are widely used.

Similarly, we can exploit the characteristics of optical **satellite time series** data. Multi-spectral satellite images with medium resolution of up to 10m ground sampling distance are spatially correlated. However, the spatial correlation is weaker compared to natural images. Semantically connected objects on the Earth are usually no further away than several hundred meters and often range only a few pixels. Similarly, while the sequence of observations is very relevant for vegetation-related downstream tasks, the acquisition frequency of days or weeks decorrelates the sequences significantly and high-frequency processes, such as cloud cover, can appear in the time series without connection to previous or later observations. Satellite sensors also capture a rich spectral signature. Hence, a single pixel in satellite imagery is semantically more informative than a pixel in natural RGB images. This is reflected in the success of pure pixel-wise classifiers in remote sensing, such as a random forest, that do not consider any spatial information.

The following sections of this chapter will focus on the three primary neural network modules to extract temporal information from sequential data.

1.3.1. Convolutional Neural Networks

The examples introduced so far considered data of no particular order or structure. In the NDVI approximation problem, we could have introduced first red and then near-infrared reflectances in the data vector to the neural network. Likewise, we could have chosen the reversed order. The order of dimensions in the input space carried no particular meaning. Similarly for the classification example, the neural network would have achieved identical accuracy if we had used $\mathbf{x} = (x_1, x_2)$ or $\hat{\mathbf{x}} = (x_2, x_1)$ coordinates consistently.

We can formally see this invariance to the order by analyzing the

$$\text{linear projection} \qquad y = \sum_{i=1}^{N} w_i x_i \tag{1.14}$$

of Eq. (1.1). Each data-dimension x_i is scaled by a dedicated weight w_i. The commutative summation ensures that the sequential order of observations is irrelevant. Many applications, however, measure data with a distinct sequential structure. For instance, our perception of gray-scale images depends solely on the neighborhood relations of their pixels. Temporal smoothness is an important assumption in any temporal process. Often, observations are taken in close succession and are more similar than if they were taken after longer periods. For illustration purposes, let's consider a sequence $\mathbf{x} = (x_0, \dots, x_t, \dots, x_T)$ of T 1-dimensional observations, such as daily temperature readings. These sequences could be of infinite length if the data is captured continuously. Associating each temporal observation with a dedicated weight, as in the NDVI approximation example, would require inefficiently large models. For temperature (or gray-scale images), it would be more intuitive to rather consider the local structure to identify distinct patterns in the change of temperature. To integrate this local structure, we can restrict the summation bounds of the linear projection to consider only a small perceptive field of K elements. This leads us to the

$$\text{discrete convolution} \qquad y_t = \sum_{k=-\lfloor \frac{K}{2} \rfloor}^{\lfloor \frac{K}{2} \rfloor} w_k x_{t+k} \tag{1.15}$$

over a K-sized convolutional kernel $\mathbf{w} \in \mathbb{R}^K$. Note how this subtle modification changes the role of single weight parameter w_k: instead of scaling a dedicated dimension, it now weighs the influence of previous $x_{t-\lfloor\frac{K}{2}\rfloor}$ and next $x_{t+\lfloor\frac{K}{2}\rfloor}$ data on the current output y_t. For D-dimensional data $\mathbf{x}_t \in \mathbb{R}^D$ we use the inner product $\mathbf{w}_k^\intercal \mathbf{x}_{t-k}$ instead of the scalar multiplication $w_k x_{t-k}$.

Analogous to the linear layer Eq. (1.1), we can write a convolutional layer of a neural network

$$\mathbf{y}_i = \sigma(\mathbf{w}_i * \mathbf{x}) \qquad \mathbf{y}_i \in \mathbb{R}^{T-K}, \mathbf{x} \in \mathbb{R}^{T\times D}, \mathbf{w}_i \in \mathbb{R}^{K\times D} \tag{1.16}$$

for feature i conveniently using the convolutional star notation $*$.

1.3.2. Recurrent Neural Networks

Convolutional neural networks assume a fixed local structure in the data by weighting the direct neighborhood using a sliding fixed-size kernel. Often, however, data is not strictly locally ordered. In language, the semantic meaning of sentences depends on few keywords that are usually not located nearby. Similarly, causal relationships may be delayed. An engine failure may lead to a car crash after several seconds or a plane crash after minutes. A year of low precipitation can affect crop yield in the following year. For capturing these relationships, a more state-based modeling approach may be beneficial where a memory state provides context to new data observations to make a decision. This state-based perspective is the motivation behind recurrent neural networks.

On a **general level**, a recurrent layer encodes this time series into a representation

$$\mathrm{h}_T = f_\mathrm{W}(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T) = f_\mathrm{W}(\mathbf{x}_T, \mathrm{h}_{T-1}) \tag{1.17}$$

using the same weights W for each time instance. This is realized by iteratively applying the non-linear transformation f_W on a new observation $\mathbf{x}_T$ given the fixed-length temporal context h_{T-1} from previous observations. In turn, h_{T-1} is the fixed size representation of the $\mathbf{X}_{\rightarrow T-1}$ time series up to $T-1$. At $t=1$, h_1 is initialized with zeros.

First **recurrent neural networks** (Rumelhart et al., 1986) realized this by two linear transformations

$$\mathbf{h}_T = \sigma\left(\mathbf{W}_\mathsf{x}\mathbf{x}_T + \mathbf{W}_\mathsf{h}\mathbf{h}_{T-1}\right) \tag{1.18}$$

that combine current input $\mathbf{x}_T$ and temporal context $\mathbf{h}_{T-1} \in \mathbb{R}^H$ using the weights $\mathbf{W}_\mathsf{x} \in \mathbb{R}^{D\times H}$ and $\mathbf{W}_\mathsf{h} \in \mathbb{R}^{H\times H}$.

This formulation, however, applies the same weight matrices T times which leads to *vanishing* or *expoding* gradients (Bengio et al., 1994) and inhibits learning long-term relationships. While exploding gradients can be avoided by gradient clipping, vanishing gradients have been addressed by adding multiple *gates* (Hochreiter and Schmidhuber, 1997) that provide more control over which features are propagated through time.

The **Long Short-Term Memory** (LSTM) recurrent neural network adopts Eq. (1.18) in four distinct recurrent layers, termed gates, where the gate-activations at time T were defined as

$$\text{the forget gate} \qquad \mathbf{f}_T = \sigma\left(\mathbf{W}_{\mathsf{fx}}\mathbf{x}_T + \mathbf{W}_{\mathsf{fh}}\mathbf{h}_{T-1}\right) \tag{1.19}$$

$$\text{the input gate} \qquad \mathbf{i}_T = \sigma\left(\mathbf{W}_{\mathsf{ix}}\mathbf{x}_T + \mathbf{W}_{\mathsf{ih}}\mathbf{h}_{T-1}\right) \tag{1.20}$$

$$\text{the modulation gate} \qquad \mathbf{g}_T = \tanh\left(\mathbf{W}_{\mathsf{gx}}\mathbf{x}_T + \mathbf{W}_{\mathsf{gh}}\mathbf{h}_{T-1}\right) \tag{1.21}$$

$$\text{the output gate} \qquad \mathbf{o}_T = \sigma\left(\mathbf{W}_{\mathsf{ox}}\mathbf{x}_T + \mathbf{W}_{\mathsf{oh}}\mathbf{h}_{T-1}\right). \tag{1.22}$$

These gate-activations are used to update

$$\text{the cell state} \qquad \mathbf{c}_T = \mathbf{f}_T \circ \mathbf{c}_{T-1} + \mathbf{i}_T \circ \mathbf{g}_T, \text{ and} \tag{1.23}$$

$$\text{the output} \qquad \mathbf{h}_T = \mathbf{o}_T \circ \mathbf{c}_T \tag{1.24}$$

from $T-1$ to T.

Let's compare the LSTM with the initial recurrent neural network on the general of Eq. (1.17). The initial formulation modeled the temporal context as single vector $\mathrm{h}_T^{\mathsf{RNN}} = \{\mathbf{h}_T\}$ while the LSTM maintains two state vectors $\mathrm{h}_T^{\mathsf{LSTM}} = \{\mathbf{h}_T, \mathbf{c}_T\}$. Each of the four gates serves as separate recurrent connection which leads to $\{\mathsf{i},\mathsf{f},\mathsf{g},\mathsf{o}\} \times \{\mathsf{x},\mathsf{h}\}$ weight matrices for the LSTM model.

In later decades, several gated recurrent neural network variants have been proposed (Gers et al., 2002). Most notably the more memory-efficient Gated Recurrent Unit (GRU) (Chung et al., 2014) that simplifies the four LSTM gates into two GRU gates (update and reset) with similar accuracy on a variety of tasks or the recently proposed StarRNN (Turkoglu et al., 2021b). However, no connection scheme has been found that was superior on a broader family of tasks (Jozefowicz et al., 2015).

Some data is both locally structured in some dimensions but require encoding larger sequential contexts in other dimensions. One example of this data may be videos where the local pixel structure individual frames should be represented by convolutions but actions in the temporal dimension require a state-based encoded via recurrent neural networks. This requires combining convolutions with recurrent state-based memory units in **convolutional recurrent neural networks**.

These convolutional recurrent networks use convolutions instead of matrix multiplications in Eqs. (1.18) to (1.21) (Shi et al., 2015). This allows transforming spatio-temporal data $\mathbf{X} \in \mathbb{R}^{T\times H\times W\times D}$ of T D-dimensional images of a certain height H and width W. In effect, every recurrent neural network can be modified as a convolutional recurrent network with LSTM (Shi et al., 2015) and GRU (Siam et al., 2017) variants proposed in the literature.

1.3.3. Attention Neural Networks

In contrast to convolutions and recurrence that exploit a pre-defined local structure in the data by a fixed-size weight kernel or by iterative sequential integration of new observations, attention dynamically extract features from the local structure in a time series.

This is done with attention scores $\boldsymbol{\alpha} \in [0,1]^T$ that dynamically adapt to the data structure by

$$\text{attention-weighted average} \qquad y = \sum_{t=1}^{T} \alpha_t x_t, \qquad \text{where} \sum_{t=1}^{T} \alpha_t = 1, \tag{1.25}$$

over a data vector $\mathbf{x} \in \mathbb{R}^T$ of T elements. In concept, this is analogous to the linear projection of the dense neural network of Eq. (1.14). The crucial difference, however, is that the attention scores α_t are evaluated dynamically for each data point while weights w_i are learned once using gradient descent on a training dataset. Overall, these attention scores allow the model to dynamically focus to the relevant sub-structure in the data.

An attention mechanism

$$\alpha(\mathbf{q}, \mathbf{K})_t = \frac{\exp\big(k(\mathbf{q}, \mathbf{k}_t)\big)}{\sum_{\tau=0}^{T} \exp\big(k(\mathbf{q}, \mathbf{k}_\tau)\big)} \tag{1.26}$$

calculates these scores from one **query** $\mathbf{q} \in \mathbb{R}^H$ and T **key** $\mathbf{K} = (\mathbf{k}_t)_{t \in \llbracket 0,T \rrbracket} \in \mathbb{R}^{T \times H}$ vectors. The query provides a semantic context that is compared to a key $\mathbf{k}_t$ for each sequence element t by an *alignment function* $k(\mathbf{q}, \mathbf{k}_t)$. The softmax normalization $\frac{\exp(\cdot)}{\sum \exp(\cdot)}$ ensures that $\sum_{t=1}^{T} \alpha_t = 1$. The terminology of query and key vectors is motivated by the analogy to a relational look-up operation where a query vector $\mathbf{q}$ is compared to a database of keys $\mathbf{K}$.

A variety of alignment kernels that utilize the

cosine distance	(Graves et al., 2014)	$k(\mathbf{q}, \mathbf{k}) = \frac{\mathbf{q}^\intercal \mathbf{k}}{\Vert\mathbf{q}\Vert_2 \Vert\mathbf{k}\Vert_2}$,
dot-product	(Luong et al., 2015)	$k(\mathbf{q}, \mathbf{k}) = \mathbf{q}^\intercal \mathbf{k}$,
D-scaled dot-product	(Vaswani et al., 2017)	$k(\mathbf{q}, \mathbf{k}) = \frac{\mathbf{q}^\intercal \mathbf{k}}{\sqrt{D}}$,
radial basis function	(Tsai et al., 2019)	$k(\mathbf{q}, \mathbf{k}) = \exp(-\gamma \Vert\mathbf{q} - \mathbf{k}\Vert_2)$

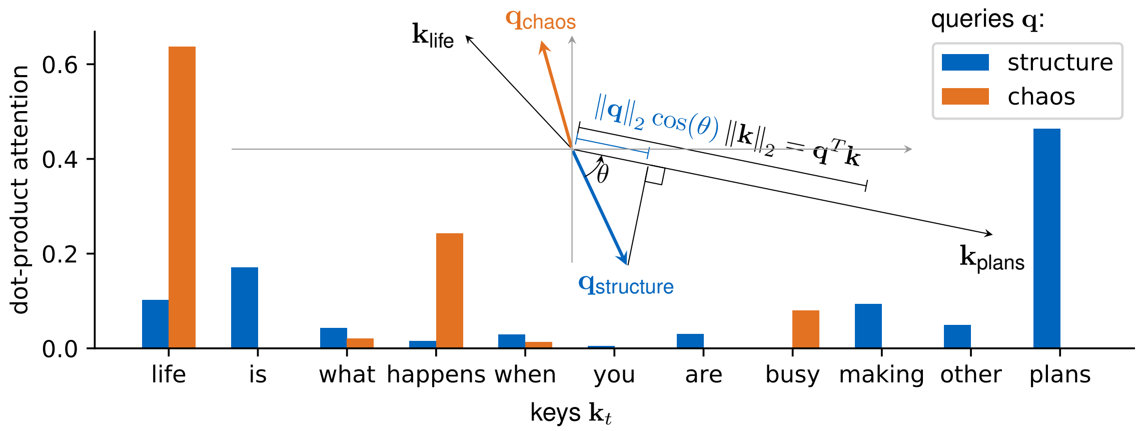


Figure 5: This figure shows the effect of dot-product attention on a language task. The GloVe (Pennington et al., 2014) word embeddings are used to project a word token into a 300-dimensional feature space that describes the meaning of the respective word. Two queries $\mathbf{q}_{\text{structure}}, \mathbf{q}_{\text{chaos}} \in \mathbb{R}^{300}$ are compared with the words: "life is what happens when you are busy making other plans". Each word in this sentence is embedded as key $\mathbf{k}_t \in \mathbb{R}^{300}$. Querying the sentence for "structure" leads to high attention on "plans" and "is" while a query on "chaos" attends mostly to words like "life", "happens", "busy". A geometric interpretation of the underlying dot-product $\mathbf{q}^T\mathbf{k}$ is shown in the center in 2-PCA space. Large attention scores are produced by a large dot-product $\mathbf{q}^T\mathbf{k}$. This dot-product can be interpreted as the length of the $\|\mathbf{k}\|_2$-scaled projection $\|\mathbf{q}\|_2 \cos(\theta)$ of $\mathbf{q}$ on $\mathbf{k}$. The dot-product is largest at $\theta = 0$ when both embedding vectors point towards the same direction which implies a similar semantic meaning.

has been proposed alongside learning an

embedding $\mathbf{w}_\text{q}$	(Luong et al., 2015)	$k(\mathbf{k}) = \mathbf{w}_\text{q}^\mathsf{T}\mathbf{k}$,
a linear projection $\mathbf{W}$	(Luong et al., 2015)	$k(\mathbf{q},\mathbf{k}) = \mathbf{q}^\mathsf{T}\mathbf{W}\mathbf{k}$,
a feed forward network	(Bahdanau et al., 2015)	$k(\mathbf{q},\mathbf{k}) = \mathbf{w}_\text{a}^\mathsf{T} \tanh\left(\mathbf{W}_\text{q}\mathbf{q} + \mathbf{W}_\text{k}\mathbf{k}\right)$, and
a feed forward network	(Velickovic et al., 2018)	$k(\mathbf{q},\mathbf{k}) = \text{LReLU}\left(\mathbf{w}^\mathsf{T}\mathbf{W}\mathbf{k} + \mathbf{w}^\mathsf{T}\mathbf{W}\mathbf{q}\right)$

where $\mathbf{w}$ and $\mathbf{W}$ are weight parameters and LReLU represents a Leaky Rectified Linear Unit activation function.

An example of keys and queries is shown in Fig. 5 along with a geometric interpretation of the dot-product in word-embedding space.

Extending Eq. (1.25), we can use N H-dimensional queries $\mathbf{Q} = (\mathbf{q}_i)_{i=[\![1,N]\!]}$. Each query $\mathbf{q}_i \in \mathbb{R}^H$ on keys $\mathbf{K} = (\mathbf{k}_{k=[\![1,T]\!]}), \mathbf{k}_t \in \mathbb{R}^H$ produce attention scores to aggregate T **values**

$\mathbf{V} = (\mathbf{v}_t)_{t=[\![1,T]\!]}, \mathbf{v}_t \in \mathbb{R}^D$ to an output $\mathbf{y} \in \mathbb{R}^D$ using

$$\text{attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) \qquad \mathbf{y}_i = \sum_{t=1}^{T} \alpha(\mathbf{q}_i, \mathbf{K})_{i,t} \mathbf{v}_t. \tag{1.27}$$

with $\mathbf{Q} \in \mathbb{R}^{N \times H}, \mathbf{K} \in \mathbb{R}^{T \times H}, \mathbf{V} \in \mathbb{R}^{N \times D}$.

Recurrent neural networks, such as $\{\mathbf{h}_t, \mathbf{c}_t\} = \text{LSTM}(\mathbf{x}_t, \mathbf{h}_{t-1})$ with $(\mathbf{x}_t)_{t=[\![1,T]\!]}$ Eq. (1.17) and $\{\mathbf{h}_0, \mathbf{c}_0\} =: \{\mathbf{0}, \mathbf{0}\}$, provide a natural context for attention mechanisms (Bahdanau et al., 2015). They encode a sequence $\mathbf{X} = (\mathbf{x}_t)_{t=[\![1,T]\!]}$ into a representation $\mathbf{H} = (\mathbf{h}_t)_{t=[\![1,T]\!]}$ that contain both 1) individual representations $\mathbf{h}_t$ and 2) global sequence context $\mathbf{c}_T$ that can be used as query $\mathbf{q} =: \mathbf{c}_T$ on keys $\mathbf{k}_t =: \mathbf{h}_t$ to calculate attention scores $\alpha_t = \alpha(\mathbf{c}_T, \mathbf{H})_t$.

Transformer networks (Vaswani et al., 2017) popularized **self-attention** where each input $\mathbf{x}_t \in \mathbb{R}^{D_\text{in}}$ of the input sequence $\mathbf{X} = (\mathbf{x}_t)_{t=[\![1,T]\!]}$ is linearly transformed into

$$\text{queries}(\mathbf{x}_t) \qquad \mathbf{q}_t = \mathbf{x}_t^\intercal \mathbf{W}_\text{q} = \left(\sum_{d=1}^{D_\text{in}} x_{t,d} w_{h,d}^{(q)} \right)_{h \in [\![1,H]\!]} \tag{1.28}$$

$$\text{keys}(\mathbf{x}_t) \qquad \mathbf{k}_t = \mathbf{x}_t^\intercal \mathbf{W}_\text{k} = \left(\sum_{d=1}^{D_\text{in}} x_{t,d} w_{d,h}^{(k)} \right)_{h \in [\![1,H]\!]} \tag{1.29}$$

$$\text{values}(\mathbf{x}_t) \qquad \mathbf{v}_t = \mathbf{x}_t^\intercal \mathbf{W}_\text{v} = \left(\sum_{d=1}^{D_\text{in}} x_{t,d} w_{d,v}^{(v)} \right)_{v \in [\![1,D_\text{out}]\!]} \tag{1.30}$$

by projection weights $\mathbf{W}_\text{q}, \mathbf{W}_\text{k} \in \mathbb{R}^{D_\text{in} \times H}, \mathbf{W}_\text{v} \in \mathbb{R}^{D_\text{in} \times D_\text{out}}$. Here, T H-dimensional queries $\mathbf{Q} \in \mathbb{R}^{T \times H}$, are aligned to T H-dimensional keys $\mathbf{K} \in \mathbb{R}^{T \times H}$ to produce a self-attention matrix $\mathbf{A} \in [0,1]^{\{T \times T\}}$ where each projected sequence token serves as query on the other tokens.

This configuration allows self-attention $\mathbb{R}^{T \times D_\text{in}} \mapsto \mathbb{R}^{T \times D_\text{out}}$ to be stacked in multiple layers. In practice, self-attention is applied in parallel in multi-headed self-attention layers alternated with time-wise applied feed-forward networks and skip connections. These layers form the core of Transformer architectures (Vaswani et al., 2017; Devlin et al., 2019; Lan et al., 2020) that is widely used in natural language processing.

1.4. Uncertainty Estimation with Deep Neural Networks

So far, we implicitly assumed that data and models are error-free and deterministic. However, we gather data in a physical world where any data has been measured by sensors and detectors, real-world models are rarely deterministic by nature or due to unobserved variables, and our models are only approximations of the theoretical optimum. Gawlikowski et al. (2021) break the steps from raw information to model predictions into the four distinct steps of data acquisition, (deep) model building, application of the model for inference, and estimating the prediction's uncertainty. Each of these steps is influenced by factors that introduce uncertainty. For instance, the real-world situation may vary. This can be caused by a non-deterministic nature of a problem or by unobserved variables not available to us. In a remotely sensed vegetation use-case, we can hardly capture all factors that influence the development of vegetation with space-borne sensors. Similarly, any measurement system introduces observation noise. The heat of the Sun's nuclear fusion creates heat that causes electrons to change energy states which emit photons. These photons are absorbed in some wavelengths or reflected at others by objects on Earth which are then converted by detectors into analog voltages and eventually digital numbers. If we measured the Earth's surface at the same place and time twice, we would not obtain identical measurements. To break it down to the NDVI example of Fig. 2, we can not expect to obtain the same input reflectances in an identical acquisition scenario. Additional two factors of uncertainty are introduced in the design decisions of the model architecture and and training procedure. A final factor involves errors introduced by a change in the particular task, for instance when a model trained on one task in one label space is asked to predict classes it has not seen before. This factor is addressed in a multi-task framework the next section.

We can conceptually aggregate these physical processes into a non-linear transformation $g_z(\cdot)$ of unknown parameters z that transforms theoretical error-free input data $\mathbf{x}_0$ with additive noise e. This acquisition model (Wang et al., 2019a)

$$\mathbf{x}_i = g_z(\mathbf{x}_0) + e \tag{1.31}$$

can be seen as the decomposition of a sample $\mathbf{x}_i$ drawn from a data distribution $\mathbf{x}_i \sim \mu(\mathcal{X})$ from Section 1.2 into deterministic $\mathbf{x}_0$ and random components $e, z \sim p(e, z)$. If we determine the model parameters w using Eq. (1.9) by minimizing an objective function on a sampled dataset $\mathcal{D} = \{\mathbf{x}_i\}_{i=1}^N$, we obtain parameters w that are affected by variance in the training data as well as the choice of neural network architecture. Any model prediction

$$\mathbf{y}_i = f_{\mathrm{w}^*}(\mathbf{x}_i) \tag{1.32}$$

will, thus, be affected by uncertainty in the model induced by w and in the data, since $\mathbf{x}_i$ is one of many possible realizations of $\mathbf{x}_0$.

Malinin and Gales (2018) summarized this relationship between model and data uncertainty in a Bayesian framework

$$P(y = c|\mathbf{x}^*, \mathcal{D}) = \int \underbrace{P(y = c|\mathbf{x}^*, \mathrm{w})}_{\text{data}} \underbrace{p(\mathrm{w}|\mathcal{D})}_{\text{model}} \, d\mathrm{w} \tag{1.33}$$

where a combination of data and model uncertainty influences the categorical probability $P(y = c|\mathbf{x}^*, \mathcal{D})$ of class c given the test sample $\mathbf{x}^*$ and training dataset $\mathcal{D} = \{\mathbf{x}_i, y_i\}_{i=1}^N \sim$

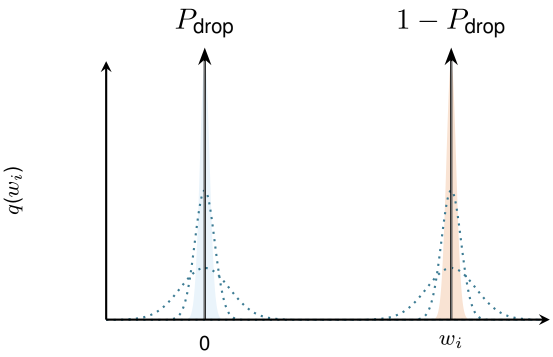


Figure 6: Dropout on one weight $w_i \in \mathrm{w}$ can be seen as sampling from a mixture of Gaussians with infinitesimally small variance where one is centered on 0 (weight set to zero) and the other on a pre-determined weight w_i. This interpretation as a distribution over weights $q(w_i)$ embeds Monte-Carlo Dropout in the approximate variational inference framework. The predictive uncertainty can be estimated from multiple dropout realizations at test time. The dotted lines indicate Gaussian mixture models with increasingly smaller variances that approach Dirac functions (arrows) at 0 and w_i. Sampling from a mixture of these two Diracs can be interpreted as setting the weight from w_i to zero at a p_{drop} probability which corresponds to dropout layers at neural networks.

$\mu(\mathcal{X} \times \mathcal{Y})$. The uncertainty in the model parameters from the training dataset $p(\mathrm{w}|\mathcal{D})$ is considered in the prediction $P(y=c|\mathbf{x}^*, \mathrm{w})$ as a distribution over distributions.

This formulation, however, is too computationally expensive for neural networks, as estimating the posterior over weights $p(\mathrm{w}|\mathcal{D})$ and the integral Eq. (1.33) requires marginalization over the entire weight space, i.e., evaluating every possible weight. In practice, several approximations are made. In variational inference, the intractable posterior $p(\mathrm{w}|\mathcal{D}) \approx q_\phi(\mathrm{w})$ is approximated in a tractable family of distributions q_ϕ. The distribution parameter ϕ are found by minimizing the Kullback-Leibler (KL) divergence to the true model posterior (Kendall and Gal, 2017). A conceptually simpler but computationally more expensive strategy is building an explicit ensemble (Lakshminarayanan et al., 2017). Here, multiple point estimates $\mathrm{w}^{(i)} \sim p(\mathrm{w}|\mathcal{D}^{(i)}; \mathrm{w}^{(i)}_{\text{init}})$ can be found by gradient descent using Eq. (1.12) from different weight initialization $\mathrm{w}^{(i)}_{\text{init}}$ and dataset partitions $\mathcal{D}^{(i)}$.

Similarly, the integral of Eq. (1.33) is not tractable and can be approximated via sampling

$$P(y=c|\mathbf{x}^*, \mathcal{D}) \approx \frac{1}{M} \sum_{i=1}^{M} P(y=c|\mathbf{x}^*, \mathrm{w}^{(i)}) \tag{1.34}$$

of M model parameterizations $\mathrm{w}^{(i)}$ that can be obtained via Monte-Carlo Dropout $\mathrm{w}^{(i)} \sim q_\phi(\mathrm{w})$ (Gal and Ghahramani, 2016), or explicit ensembling (Lakshminarayanan et al., 2017) $\mathrm{w}^{(i)} \sim p(\mathrm{w}|\mathcal{D}^{(i)}; \mathrm{w}^{(i)}_{\text{init}})$.

1.4.1. Monte-Carlo Dropout

One broadly used approach is to model $q(\mathrm{w})$ as the mixture of two Gaussians with small variance where one Gaussian is centered on zero and the other Gaussian at w_i, as shown in Fig. 6. This weight $w_i \in \mathrm{w}$ can be determined via gradient descent. If we choose the variance of the Gaussians infinitesimally small, we can see these Gaussian distributions as Dirac delta functions. When we then sample from these two Diracs located at zero and w_i, we effectively set the weight w_i to zero at a certain probability P_{drop}. This dropout operation (Srivastava et al., 2014) is broadly used in neural networks as regularization when training. It can be performed efficiently and at a large scale. Gal and Ghahramani (2016) recognized this interpretation of dropout as approximate variational inference where different weight realizations $\mathrm{w}^{(i)} \sim q(\mathrm{w})$ can be sampled from w by setting some w_i to zero. At test time, the variance of M realizations $\{P(y = c|\mathbf{x}^*, \mathrm{w}^{(i)})\}_{i=1}^{M}$ can then model the uncertainty in the model weights by effectively measuring the divergence of the in individual realizations.

1.4.2. Aleatoric Uncertainty

While weight-samples drawn with Monte-Carlo Dropout can be used to model variance in the model weights, we can not infer uncertainty about the data since we assume that the test sample $\mathbf{x}^*$ is given in Eq. (1.33) while, in practice, it is drawn from an unknown test data distribution $\mathbf{x}^* \sim p(\mathbf{x}|y = c)$ conditioned on class c.

To estimate the variance over this data distribution, we can use an methodology that was initially proposed to balance the influence of losses from auxiliary objectives (Kendall et al., 2018). We can design the output of the neural network such that it predicts both $\{\hat{\mathbf{y}}, \hat{\boldsymbol{\sigma}}\} = \{\hat{y}_i, \hat{\sigma}_i\}_{i=1}^{D} = f_{\mathrm{w}}(\mathbf{x})$ a prediction $\hat{\mathbf{y}}$ and a Gaussian variance $\hat{\boldsymbol{\sigma}}$. In an image segmentation case, we can see i as a single pixel in an image of D pixels. Alternatively, for time series, we can see i as temporal index of a sequence of D observations.

Since we only have samples $\boldsymbol{y} = \{y_i\}_{i=1}^{D}$ from the desired target distribution and don't know the variance, we need to cast the loss function (Kendall and Gal, 2017)

$$\mathcal{L}(\underbrace{\hat{\boldsymbol{y}}, \hat{\boldsymbol{\sigma}}}_{f_{\mathrm{w}}(\mathbf{x})}, \mathbf{y}) = \frac{1}{D} \sum_{i=1}^{D} \frac{\|y_i - \hat{y}_i\|^2}{2\hat{\sigma}_i^2} + \frac{1}{2} \log \hat{\sigma}_i^2 \tag{1.35}$$

into a regularization framework where the model minimizes the objective $\mathcal{L}$ by either 1) predicting more accurately which minimizes the L2 Norm $\|y_i - \hat{y}_i\|^2$ or 2) reducing the penalty of a wrong prediction by increasing the variance $\hat{\sigma}_i^2$. The second additive term $\frac{1}{2} \log \hat{\sigma}_i^2$ prevents the trivial solution of $\mathcal{L} \to 0$ by predicting $\sigma_i \to \infty$.

A neural network optimized with this loss function estimates two values: A prediction $\hat{y}_i$ close to the ground truth y_i and a parameter $\hat{\sigma}_i^2$ for each pixel/observation i of one sample. If the model predicts a large $\hat{\sigma}_i^2$, the loss-penalty of an potential error in the prediction $\|y_i - \hat{y}_i\|^2$

is reduced. We can interpret this parameter as variance in a squared Mahalanobis distance[1] between a sample from the target data distribution y_i and an estimated Gaussian distribution parameterized by $\hat{y_i}, \hat{\sigma_i}$.

1.5. Distribution Shift and Transfer Learning

The central assumption in machine learning is that the samples stored in training and testing datasets are drawn *independently* from an *identical* (labeled) data distribution (Shalev-Shwartz and Ben-David, 2014). The independence assumptions is necessary to factorize the likelihoods of independent data samples from dataset when finding model parameters. The assumption on identical (labeled) data distributions ensures that we measure the generalization of a model on the identical task and domain. It is this assumption on identical data distributions that is violated to varying degrees in real-world applications. With recent progress on current machine learning, researches, such as Bengio et al. (2020), state that "it is not enough to obtain good generalization on a test set sampled from the same distribution as the training data, we would also like what has been learned in one setting to generalize well in other related distributions". Similarly, Marcus (2020) associates the lack of intelligence in modern machine learning in his proposal of robust artificial intelligence for the next decade.

In this context, we can introduce the definitions of **domain and task** according to Pan and Yang (2009) to systematically structure how this assumption can be relaxed. The set of feature space $\mathcal{X}$ and probability distribution $\mu(\mathcal{X})$ form a domain $D = \{\mathcal{X}, \mu(\mathcal{X})\}$. In computer vision, the input space $\mathcal{X}$ may be all possible 8-bit RGB pixel values that an image can take. In natural language, the input space may be all permutations over a vocabulary of words. It is the specific data distribution $\mu(\mathcal{X})$ that makes certain sequences of words more likely when reading a sentence with a certain semantic meaning to us. The set of label space $\mathcal{Y}$ and predictive function $f : \mathcal{X} \mapsto \mathcal{Y}$ form a task $\mathcal{T} = \{\mathcal{Y}, f(\cdot)\}$ on this domain. A predictive function associates a data sample with a discrete label or continuous target variable y within a label space $\mathcal{Y}$. For instance, the NDVI function that we approximated in previous examples is a predictive function. The predictive function is usually not directly accessible to us, but we can use its predictions $y_i = f(\mathbf{x}_i)$ that we store in a labeled dataset $\mathcal{D} = \{\mathbf{x}_i, y_i\}_{i=1}^N \sim p(\mathcal{X} \times \mathcal{Y})$. From a probabilistic viewpoint, we can also see the predictive function as probability $P(y|\mathbf{x})$ over $y \in \mathcal{Y}$ given some data $\mathbf{x} \in \mathcal{X}$.

The remaining sections of this chapter outline the limitations of enforcing the equal distribution assumption by common random sampling of a diverse dataset in Section 1.5.1. Section 1.5.2 introduces current working examples of designed transfer learning and multi-task learning while Section 1.5.2 introduces meta-learning as the data-driven automation of transfer learning with datasets of datasets. The content of this chapter is also summarized in Fig. 7 for a concise picture of the key components of each section.

[1] Mahalanobis Distance $d(x, \{\mu, \sigma\}) = \sqrt{\frac{(x-\mu)^2}{\sigma^2}}$ between a sample x and a distribution μ, σ of mean μ and

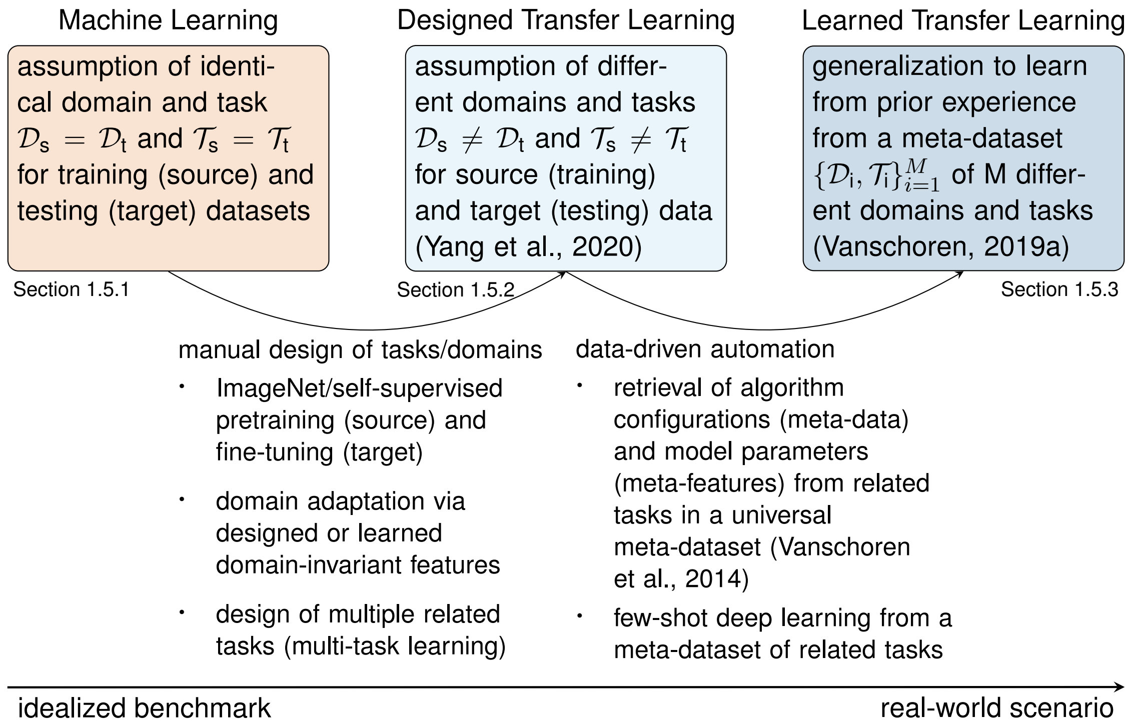

Machine Learning
assumption of identical domain and task $\mathcal{D}_\mathrm{s} = \mathcal{D}_\mathrm{t}$ and $\mathcal{T}_\mathrm{s} = \mathcal{T}_\mathrm{t}$ for training (source) and testing (target) datasets
Section 1.5.1
Designed Transfer Learning
assumption of different domains and tasks $\mathcal{D}_\mathrm{s} \neq \mathcal{D}_\mathrm{t}$ and $\mathcal{T}_\mathrm{s} \neq \mathcal{T}_\mathrm{t}$ for source (training) and target (testing) data (Yang et al., 2020)
Section 1.5.2
Learned Transfer Learning
generalization to learn from prior experience from a meta-dataset $\{\mathcal{D}_\mathrm{i}, \mathcal{T}_\mathrm{i}\}_{i=1}^{M}$ of M different domains and tasks (Vanschoren, 2019a)
Section 1.5.3
manual design of tasks/domains
• ImageNet/self-supervised pretraining (source) and fine-tuning (target)
• domain adaptation via designed or learned domain-invariant features
• design of multiple related tasks (multi-task learning)
data-driven automation
• retrieval of algorithm configurations (meta-data) and model parameters (meta-features) from related tasks in a universal meta-dataset (Vanschoren et al., 2014)
• few-shot deep learning from a meta-dataset of related tasks
idealized benchmark
real-world scenario

Figure 7: Schematic map of concepts introduced in Section 1.5. The assumption of identical tasks and domains in machine learning (Section 1.5.1) does not necessarily hold true in real-world applications. Transfer learning and multi-task learning approaches relax this assumption by manually defining a transfer between source and target tasks and domains (Section 1.5.2). Meta-learning in (Section 1.5.3) generalizes the domain transfer to learn from prior experience in a data-driven way.

1.5.1. Single Domain and Task Assumption in Machine Learning

In machine learning, we approximate a predictive function from the observed data. Model-driven approaches break it into sub-functions of data preprocessing, feature extraction, and classification while data-driven approaches employ flexible non-linear functions, such as neural networks, end-to-end.

The central assumption we make when approximating the predictive function is that source domain D_s and task $\mathcal{T}_\mathrm{s}$ from training dataset are identical to the target domain and task $D_\mathrm{s} = D_\mathrm{t}, \mathcal{T}_\mathrm{s} = \mathcal{T}_\mathrm{t}$ of our test dataset. Since data and label space are defined by us, this often boils down to having the identical data distribution $\mu_\mathrm{s}(\mathcal{X}) = \mu_\mathrm{t}(\mathcal{X})$ and predictive function $f_\mathrm{s}(\cdot) = f_\mathrm{t}(\cdot)$. In an image classification use-case, data distributions may vary if one domain contains images taken at night, while the other contains daytime images. Predictive functions can vary if two human labelers have different opinions on what qualifies as some label and their labels are dis-proportionally split between training and testing data.

The trivial way to mitigate this disparity between training and test datasets is to split one combined dataset into training and test partitions randomly. This ensures that training and testing datasets are from the same data distribution and that the predictions of two label-

standard deviation σ

ers are equally distributed. While this may be sufficient to evaluate model performances on benchmark datasets, it is not always desired in real-world scenarios. In practice, acquiring data, i.e., sampling from an underlying data distribution, is more practicable in some circumstances than others. For instance, obtaining large bodies of text to train language models is easier from internet sources than from day-to-day conversations (Bordia and Bowman, 2019). This selection bias leads to poorer accuracies for groups underrepresented in the dataset (Buolamwini and Gebru, 2018). Similarly, not everyone contributes to content on the internet. This can overrepresent hegemonic viewpoints and encode biases potentially damaging to marginalized populations (Bender et al., 2021). In remote sensing, labeled data is abundant in administratively developed regions, such as Europe or North America that collect fine-grained geographical data periodically at a large scale. Conversely, learning patterns and predicting these statistics has little value in Europe where they are available compared to regions that don't gather geographical statistics at a large scale. It is still possible to train and test accurate models that generalize to unseen regions with one global dataset if we choose a task with a small-enough label space or if we define and select features that are more robust to shift in representations. For instance, global land cover classification with 14 classes (Hansen et al., 2000) or binary settlement detection (Esch et al., 2017) achieve accurate predictions at a global scale on tasks with few broadly defined labels. For tasks with fine-grained labels and regionally variable representations, as in vegetation-related applications, the assumption of identical data distributions only holds when data is sampled from one region only. A difficult task with fine-grained labels may require large labeled datasets that are available only in some regions but not in others. Since data distributions can vary between regions, training a predictive function on a data-abundant region and testing it on another violates the assumption of identically distributed data between training and testing datasets.

1.5.2. Designed Domain and Task Transfer

We can relax that assumption of identical data distributions with **transfer learning**. Following the definitions of Pan and Yang (2009); Yang et al. (2020), we employ transfer learning whenever we use knowledge from a source domain and task $D_\text{s}, \mathcal{T}_\text{s}$ to improve the learning of a target predictive function $f_\text{t}(\cdot)$ on data from a target domain D_t. Domain adaptation is one common *feature-based transfer learning* strategy that assumes that source and target tasks are identical $\mathcal{T}_\text{s} = \mathcal{T}_\text{t}$ but the domains differ $D_\text{s} \neq D_\text{t}$. Here, we transform (or adapt) data from the source domain such that it is similar enough to our target domain that we can train a single predictive function on (labeled) data of both domains. *Instance-based transfer learning* follows a similar idea and assumes that source and target domains have sufficient overlap such that we can identify and select some samples from the source domain dataset to improve learning on the target task without transforming them. Kernel Mean Matching (Gretton et al., 2009) is one common algorithm in this category that calculates similarity coefficients of samples from a source domain dataset to a target domain dataset. These coefficients can then be used to pick some samples more frequently for training or to weigh the objective function such that data from both domains can be used to train a model. *Model-based transfer learning* does not require tasks nor domains to be identical. It is sufficient that the structure in the source and target domains are similar and can be learned by a (deep) model. This learned structure can then be transferred from the source domain as model initialization and fine-tuned to data and labels from the target domain and task. Each layer in a neural network encodes more abstract higher-level features until the last layer defines a decision boundary for a fixed set of labels, as shown earlier in the toy example of Fig. 1. In deep neural networks, the weights to extract higher-level representations of the data structure can remain relatively unchanged while intermediate and decision layers have to be re-learned by fine-tuning on the target dataset. The effectiveness of pretraining deep neural networks on ImageNet (Krizhevsky et al., 2012) and fine-tuning on particular target tasks on different domains are broadly known and fall in this category. The field of self-supervised learning relaxes the requirement for labeled data in the source domain by defining specific tasks targeted to learn the data structure directly. These can be, for instance, filling blanked words of sentences in natural language (Devlin et al., 2019), or solving jigsaw puzzles in computer vision (Noroozi and Favaro, 2016).

The field of **multi-task learning** is similar to transfer learning as it also aims to generalize knowledge between different tasks. However, in contrast to transfer learning, there are no source domains but multiple target domains where each has insufficient labeled data to train a classifier independently. The goal of multi-task learning is to jointly learn the target tasks by exploiting the common structure between tasks. While transfer learning aims to improve particular target task(s) from knowledge of the source domains and tasks, multi-task learning aims to improve multiple target tasks simultaneously (Yang et al., 2020). As this definition is based on the motivation of the problem and what we define as source and target tasks, the distinction to transfer learning is not always clear and components of multi-task learning and transfer learning can be combined.

1.5.3. Learned Domain and Task Transfer

The examples of transfer learning and multi-task learning above required a manual selection of tasks, domains and learning algorithm by a model designer that has to ask himself the research questions of "when to transfer?" and "what to transfer?" before deciding on "how to transfer?" and choosing a specific transfer learning or multi-task learning algorithm (Yang et al., 2020). The field of **meta-learning** generalizes this process to learn from prior experience in a systematic and data-driven way (Vanschoren, 2019b,a). Meta-learning conceptually extends transfer learning where an individual task and domain are seen as a sample from a distribution over tasks and domains. This is analogous to a labeled dataset being sampled from the data distribution of a single domain and labeled by the task's predictive function. In practice, this yields a *dataset over datasets* since each task and domain are represented by a labeled dataset. In a meta-dataset each domain and task can be described by meta-data, such as algorithm configurations, and meta-features, such as evaluation results and learned model weights. A new unseen task in a related domain can then be addressed by successful algorithms of similar tasks in the meta-dataset. For example, the OpenML framework (Van-

```
Algorithm 1: Model-Agnostic Meta-Learning
Data: $p(\mathcal{T})$: distribution over tasks, $\beta$: outer step size hyperparameter ;
$\varepsilon(\mathrm{w};\mathcal{D})$ loss function of weights $\mathrm{w}$ on data $\mathcal{D}$ ;
GD gradient descend algorithm Eq. (1.12)
Result: Find meta-parameters $\mathrm{w}$.
1  randomly initialize $\mathrm{w}$;
2  repeat
3      sample batch of tasks $\tau \sim p(\mathcal{T})$;
4      foreach $\tau_i \in \tau$ do
5          sample data $\{\mathcal{D}_{\text{support}}, \mathcal{D}_{\text{query}}\} \sim p(\tau_i)$;
6          adapt parameters $\mathrm{w}^*_{\tau_i} = \text{GD}(\varepsilon(\mathrm{w};\mathcal{D}_{\text{support}}), \mathrm{w}_{\text{init}} = \mathrm{w})$ ;
7          evaluate query loss $\mathcal{L}_{\tau_i} = \varepsilon(\mathrm{w}^*_{\tau_i}, D_{\text{query}})$ ;
8      end
9      update $\mathrm{w} \leftarrow \mathrm{w} - \beta \sum_{\tau_i \sim p(\tau)} \nabla_{\mathrm{w}} \mathcal{L}_{\tau_i}$;
10 until convergence;
```

schoren et al., 2014) provides a universal meta-dataset[2] of a wide variety of tasks on different domains. It can be used to automate the search of a suitable machine learning algorithm in popular programming frameworks (Feurer et al., 2019).

The sub-field of **few-shot meta-learning** research (Finn et al., 2017; Nichol et al., 2018; Rajeswaran et al., 2019; Triantafillou et al., 2020) considers meta-datasets of related tasks. Each task contains only a few (labeled) samples of one domain that are split independently and identically distributed into $\mathcal{D}_{\text{support}}$ and $\mathcal{D}_{\text{query}}$ partitions to train and test each task, respectively. This yields a meta-dataset of task-datasets that is organized into meta-train tasks to find model parameters and meta-test tasks to evaluate the performance on unseen problems. Concretely, Finn et al. (2017) proposed the *model-agnostic meta-learning (MAML)* algorithm alg. 1 that learns a deep neural network initialization w for a family of related tasks that can be adapted to each unseen task within few gradient steps. For each task and domain τ_i in a batch of tasks, data $\mathcal{D}_{\text{support}}, \mathcal{D}_{\text{query}}$ is queried (line 5). Good model parameters w^*_i for this task are found via gradient descent Eq. (1.12) on the support set $\mathcal{D}_{\text{support}}$ from a weight initialization w (line 6). The test performance $\mathcal{L}_{\tau_i}$ on this task is determined on the independent query set $\mathcal{D}_{\text{query}}$ and stored. This inner loop (lines 4-8) is repeated for a batch of tasks and the outer weight parameters w are updated (line 9) via gradient descent on the loss $\mathcal{L}_{\tau_i}$ from the individual query datasets. This algorithm elegantly extends the training of regular neural networks (inner loop) by an outer loop to find a model initialization that encodes knowledge from different-but-related problems. Updating the outer weights (line 9), however, requires second-order gradients that pose computational challenges. First-order approximations are computationally more efficient and can be accurate if tasks are sufficiently related, as shown analytically by Nichol et al. (2018).

[2] In OpenML, tasks and domains can be completely unrelated and the aim of this meta-dataset is to cover a universal range of machine learning problems.

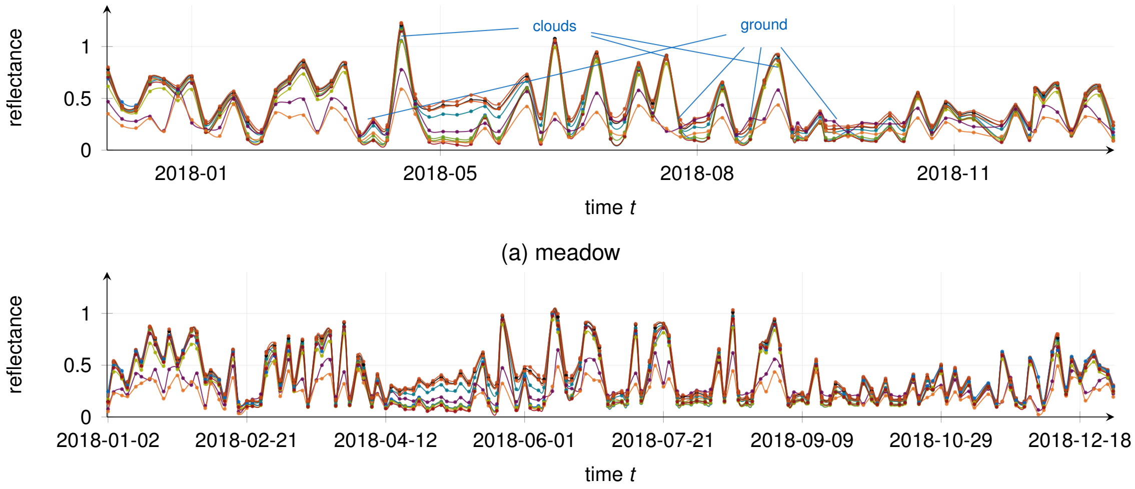
2. Applications and Data

The previous chapter introduced neural networks as flexible functions that can approximate a wide range of target predictive functions. Still, in practice, including prior knowledge is necessary to minimize estimation and optimization errors and to achieve good results in real-world applications. This chapter outlines prior knowledge and inductive bias in optical remote sensing satellite images for vegetation modeling that we explicitly or implicitly utilize in our models. The next section focuses on design choices in satellite orbits that determine the temporal acquisition frequency and design choices in multi-spectral sensors on optical remote sensing satellites. Section 2.2 then provides an overview of discriminative characteristics in the biology of vegetation that can be used to categorize plant types by spectral and temporal characteristics.

2.1. Optical Satellite Time Series

Satellites orbit the Earth at repeating intervals in orbital planes that have been designed with certain properties and applications in mind. For instance, optical remote sensing satellites exploit the oblate ellipsoidal shape of the Earth that causes orbital planes to precess around the Earth's rotation axis. By choosing specific orbit parameters, this precession rate can be tuned to $\frac{360^\circ}{1\text{ year}}$ so that the satellite orbit maintains an identical angle to the sun. In this sun-synchronous orbit, the local time of image acquisition remains identical throughout the year which is a useful property when acquiring images under similar illumination conditions (Luo et al., 2017). Similarly, the number of satellite revolutions per day Q is a function of satellite altitude[1] and determines the spacing of two successive ground tracks at the equator $\Delta\lambda = \frac{360^\circ}{Q}$. While Q can be a full integer, it is often defined as $Q = I + \frac{K}{D}$ where I represents the number of revolutions with a fraction $\frac{K}{D}$. In this fraction, D defines the period in days until $ID + K$ individual ground tracks repeat (Luo et al., 2017). For instance, the MODIS satellite has been designed for a daily global coverage and orbits the Earth in $Q_{\text{modis}} = 14 + \frac{9}{16} \approx 14.5$ revolutions per day. This leads to two successive ground tracks being $\Delta\lambda \approx 25^\circ$ or 2800 km apart at the equator which roughly corresponds to the swath of the data acquisition in 2330 km stripes. While this configuration enables daily acquisitions, the wide swatch leads to single pixels of one acquisition covering roughly 1km ground sampling distance. Since the tracks overlap in a 16-day repeat cycle, the resolution can be improved later to 250m in a 16-day composite image. Conversely, the Sentinel 2 satellites were designed for global coverage at up to 10m ground sampling distance. To achieve global coverage in a $D = 10$ day interval, $Q_{\text{S2}} = I + \frac{K}{D} = 14 + \frac{3}{10}$ revolutions per day have been chosen. The individual track separation $\frac{360^\circ}{ID+K} = 2.5^\circ$ corresponds to 280 km at the equator and matches the instrument swath with 290km. To improve the acquisition period to 5 days, Sentinel 2 operates as a

[1] Keppler's Third law $Q(a) = \frac{2\pi}{24h}\sqrt{\frac{(R_E+a)^3}{GM}}$ with Earth radius R_E, satellite altitude a, and GM Earth constants



(a) meadow

(b) corn

Figure 8: Sentinel 2 reflectance measurements of crop field parcels form (Rußwurm and Körner, 2020)

two-satellite constellation. At higher latitudes, the track distance decreases[2] which enables 2-day acquisition frequency in some locations. These considerations that trade off spatial resolution by a narrow swath with a temporal resolution with frequent revisits are central design questions for any space-borne sensor.

A third consideration for optical satellite sensors is the spectral resolution. The Sentinel 2 satellite constellation carries the Multispectral Instrument (MSI) sensor that gathers incoming light with three mirrors onto a dichroic beam splitter that separates visible and near-infrared light from shot-wave infrared. The photons are then captured by two staggered detector arrays: one for each beam. Further spectral separation is achieved by stripe filters mounted on top of the detectors that pass certain wavelengths. The choice and number of spectral bands is an important design decision. Wider bands pass more photons to the detectors which increase the signal relative to sensor noise. This allows for more pixels on the detector and increases spatial resolution. Positions of spectral bands are determined with certain applications in mind. For instance, Landsat satellites capture thermal infrared which enables temperature measurements while Sentinel 2 has several near-infrared bands to distinguish vegetation. Also, bands are generally placed in atmospheric windows where light can pass through the Earth's atmosphere although individual bands can be strategically placed outside these windows to detect features in the atmosphere[3].

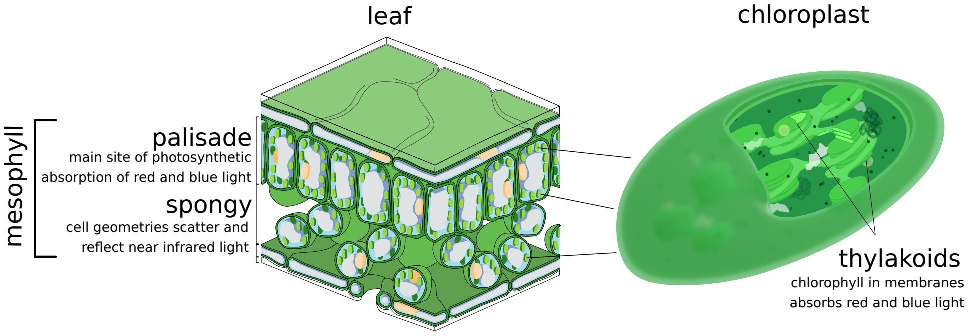

leaf
chloroplast
mesophyll
palisade
main site of photosynthetic absorption of red and blue light
spongy
cell geometries scatter and reflect near infrared light
thylakoids
chlorophyll in membranes absorbs red and blue light

Figure 9: Diagram of the internal structure of a leaf and chloroplast modified from (Zephyris, 2001; Kelvin, 2003). Photosynthesis is driven by the absorption of red and blue light in the thylakoid membranes of chloroplasts. Chloroplasts are primarily located in the vertical palisade mesophyll. Additionally, the geometry of cells in the spongy mesophyll scatters near-infrared light which causes a high near-infrared reflectance in healthy leaves. The combination of red absorption and near-infrared reflectance is a good indicator for vegetation health (Jordan, 1969).

2.2. Classification of Vegetation

The recurring application in this work is the classification of crop types on agricultural field parcels in Europe. While satellite data is abundant, labeled ground truth data is scarce in many Earth observation applications. Fortunately, crop type labels are collected within Europe's Common Agricultural Policy which provides subsidies to farmers based on the land use of their parcels. This creates an incentive to collect crop-type data alongside field geometries on a European scale which can be used for large labeled datasets. Working with large-scale datasets reduces the estimation error (see Eq. (1.10)) and lets us evaluate the architectural choices in models primarily on the approximation and optimization errors.

Earth observation satellites, such as the Sentinel 2 constellation provide surface reflectance measurements at up to ten-meter ground sampling distance every few days. This enables constant monitoring of the crop types throughout the entire vegetative period. Figure 8 shows examples of Sentinel 2 reflectance measurements of a corn and a meadow parcel from Bavaria used in Rußwurm and Körner (2020). The high reflectance of clouds overlays the informative ground signal which makes finding visually salient discriminative features between corn and meadow difficult on the raw satellite time series.

Spectral Characteristics. It is known since decades in remote sensing and plant physiology that the ratio between red and near-infrared reflectances are indicative for vegetation analysis (Jordan, 1969). Chlorophylls, the pigments used to capture photons for photosynthesis, absorb light in the red and blue spectrum. These chlorophyll pigments are embedded within light-harvesting complexes in the thylakoid membrane within the chloroplast organelles, as

[2] Individual track distance at 45° is $200\text{km} \approx \frac{360°}{ID+K}\cos(45°)R_E$ which leads to ≈ 15% overlap between tracks

[3] Sentinel 2 band 1 (442,7nm) was designed for aerosol detection. Only high-altitude cirrus clouds reflect photons at band 10 (1373,5nm) as the lower atmosphere absorbs this wavelength

shown in Fig. 9. These chloroplasts are primarily located in the vertically oriented palisade mesophyll leaf cells (Lambers et al., 2008). Additionally, the geometry of cells in the spongy mesophyll scatters light in the near-infrared spectrum which causes a high reflectance in near-infrared (Slaton et al., 2001).

This spongy mesophyll makes the near-infrared reflectance profile important to discriminate crop types since it is determined by the geometry of the leaf cells which vary between plant types. With this in mind, the Sentinel 2 sensor has been designed with multiple near-infrared bands (B5, B6, B7, B8, B8A) to distinguish vegetation.

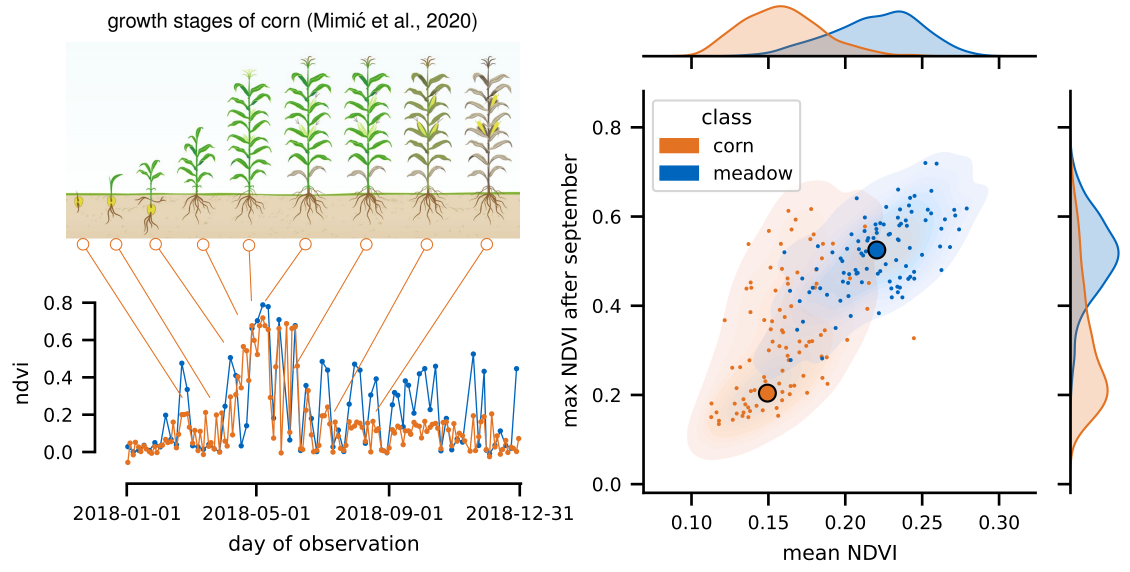
A critical biophysical variable for vegetation is the Fraction of absorbed Photosynthetic Active Radiation (FaPAR) which is the ratio between incoming and absorbed radiation in the 400nm to 700nm spectrum where the absorbed ratiation is the sum of incoming radiation without reflected and transmitted components of plants and soil (Gitelson et al., 2006; Goward and Huemmrich, 1992; Viña and Gitelson, 2005)[4]. While these components can be measured with spectrometers on the ground, remote sensing-based approaches often have to define FaPAR as a task-calibrated function of vegetation indices, such as NDVI in Los et al. (2000), which is not accurate for all vegetation (Gower et al., 1999).

Temporal Characteristics. Another way to discriminate vegetation is to monitor its life cycle events, i.e., phenology, and its reaction to environmental changes and stresses. Plants have evolved to survive in different environments with varying sun exposure, or varying availability of water and soil nutrients. Hence, different crop types experience different growth patterns under identical environmental conditions (Justice et al., 1985). We can observe different vegetation life cycle events, for instance, by monitoring the normalized difference vegetation index (NDVI) (Tucker, 1979) of Eq. (1.6) throughout the entire growth season in Fig. 10a. Here, the NDVI index reveals the distinct growth patterns of the corn and meadow parcels from Fig. 8. While corn is a heavily cultivated crop with a distinct vegetation phase, meadow is left to grow and is periodically cut. This is visible Fig. 10a in the high NDVI values throughout the entire year.

Example: Corn-Meadow Classifier

Our prior knowledge of corn and meadow characteristics can be exploited to build a corn-meadow classifier. Meadow is photosynthetically active throughout the entire year which is contrasted by corn that grows rapidly on agriculturally machined fields and is harvested in September leaving bare soil. Let's design the features "mean NDVI" and "max NDVI after September" (after corn harvest) to discriminate corn and meadow field parcels Fig. 10b. Here, we see the data distributions of corn and meadow fields in this two-dimensional feature space are well-separable. Now, we can use the dense neural network of Fig. 1 to find a decision boundary between the corn and meadow data distributions.

[4] $FaPAR = \frac{APAR}{PAR_{\text{inc}}}$ where $APAR = PAR_{\text{inc}} - PAR_{\text{out}} - PAR_{\text{trans}} + PAR_{\text{soil}}$



(a) NDVI representation of meadow and corn time series from Figs. 8a and 8b. The growth stages of corn (Mimić et al., 2020) are clearly visible in the NDVI features. The distinct phenological profile of corn is contrasted by meadow fields that stay photosynthetically active (high NDVI) throughout the entire year.

(b) With knowledge about corn and meadow, we can design the features "mean NDVI" and "max NDVI after september" (corn harvest) that make corn and meadow parcels (almost) linearly separable. A nonlinear decision boundary can be estimated by, for instance, the neural network employed in Fig. 1.

Figure 10: An illustration and comparison of a raw and a preprocessed Sentinel 2 time series of the same meadow field parcel. Preprocessing allows for a visual interpretation. The onset of growth after time step $t = 5$ is clearly visible. Also, several cutting events can be observed over the vegetation period. The preprocessed time series, however, contains repeated values due to temporal interpolation and cloud removal. In the raw time series, most information from the measured signal is retained. Noise caused by *e.g.*, atmospheric effects and clouds obscures the phenological events.

From an inductive bias perspective (Section 1.3), we used our specific domain knowledge on the task of corn and meadow classification to restrict the hypothesis space. We removed all hypotheses on the cluttered raw data of Fig. 8 where we could hardly distinguish any differences of the corn and meadow signals visually and consider only hypothesis in this task-specific feature space where the corn and meadow classes are well-separated. In this feature space, also a simple linear classifier performs reasonably well, as we could also draw a linear decision boundary between the data distributions with little loss of accuracy.

While this hands-on approach of radically restricting the hypothesis space yields explainable and generally accurate results for this task of corn-meadow classification, the specificity of the prior knowledge makes it hard to apply for other problems. First, we may not have sufficient prior knowledge of other crop types. The distinction of crop types may be less clear. For instance, the distinction between wheat and barley is less obvious. Also, our prior knowledge only accounts for one specific geographic region. Finally, in the process of calculating NDVI and defining temporal NDVI-features, we discarded valuable spectral information in the other (near-infrared) bands.

In this section, we explored the task-specific domain knowledge that allows remote sensing experts to design precise and efficient algorithms for crop type classification. From a learning perspective, we heavily restricted the hypothesis space by manual feature design and transformed the data from the raw high-dimensional satellite-data space into a low-dimensional feature space that separated the corn and meadow classes almost linearly. This feature design, however, restricted our task to one particular binary classification problem that applies to two crop types in a particular geographic region.

This is a trade-off analogous to the bias-variance problem of Section 1.2. Now, however, we are interested in the distribution of tasks, e.g., all time series classification problems or all crop type mapping problems. We can inject prior knowledge in model and feature design which helps our model significantly in the restricted space of tasks we design. However, we need a measured approach to introducing our prior knowledge to be both accurate on one specific task and can also generalize to different-but-related problems.

3. Methods for Satellite Time Series Classification

Let's focus on approaches for time series classification and particular the application of crop type mapping that has been developed in recent decades before outlining the main contributions of this work in the next chapter.

Explicitly modeling the behavior of vegetation in a generative way has a long history. While crop simulation models (Weir et al., 1984) functionally model carbon uptake within leaf cells for decades, various approximations on input variables and model structures have to be made. It is still an active research topic where recent methods fuse detailed phenotypical and genotypical features with weather information (De Los Campos et al., 2020). Finding generative models for vegetation is demanding and typically requires high-quality data and measurements at ground level throughout the year. Generative modeling of vegetation is beyond the scope of this work. Instead, we focus on finding discriminative decision boundaries between vegetation types with features that can be observed by space-borne sensors.

There are two complementary perspectives for this problem that differ primarily in the amount of prior expert knowledge that model designers use for their approaches. Model-driven strategies encode domain knowledge in designed preprocessing and feature extraction pipelines for a specific application. A problem-agnostic classifier can then be used to separate classes in a hand-designed feature space. These approaches require few data samples and explicitly reduce the dimensionality of features to obtain visually interpretable and often physically meaningful representations. In contrast, data-driven approaches approximate these preprocessing and feature extraction pipelines with flexible functions, such as neural networks, that are composed functionally identical transformations. Parameters for these transformations are found by minimizing an error from large annotated datasets. Following the no-free-lunch theorem, prior knowledge is still necessary but used at a more general level on the expected structure of data. Using data-driven approaches can be advantageous if the underlying processes cant be modeled easily without using approximations and simplifications of the problems and sufficient data is available to estimate a large number of parameters.

3.1. Model-driven Methods

Finding good discriminative features for vegetation has a long history in remote sensing. Various vegetation indices, such as the Normalized Difference Vegetation Index (NDVI) (Tucker, 1979) or the Enhanced Vegetation Index (EVI) (Huete et al., 2002), have been developed and are well-understood. These functions are based on physical understanding of processes around photosynthesis and leaf structure, as described in Section 2.2. Still, all vegetation relies on photosynthesis and produces similar responses in vegetation indices. When observing only single dates, the differences between types of vegetation are subtle. Hence, temporal characteristics on the dynamic change of vegetation with changing seasonal environments have been used for vegetation-related applications for decades (Odenweller and Johnson, 1984; Reed et al., 1994). This change in vegetation can be modeled by explicit functions. For instance, the TimeSat software (Jönsson and Eklundh, 2004; Eklundh and Jönsson, 2016) fits piece-wise defined Gaussian curves to temporal NDVI profiles of satellite time series. The parameters of these curves, *i.e.*, the steepest ascent and descent, and their dates indicate key phenological characteristics, such as the onset of greenness or the date of senescence. This allows for detailed phenological analyses (White et al., 2009; Olsson et al., 2005) and can be used as a distinctive feature for further classification (Jia et al., 2014; Singha et al., 2016). These functions are designed with idealized vegetation profiles in mind that are monitored continuously. Hence, perturbations in the data, as induced by clouds, have to be identified and removed. This can be done via explicit cloud classifiers, such as FMask (Zhu and Woodcock, 2012; Zhu et al., 2015) or MAJA (Hagolle et al., 2010), in data preprocessing pipelines. Additionally, atmospheric correction algorithms (Matthew et al., 2000; Richter, 1996; Louis et al., 2016) further harmonize the reflectance measurements between different dates, as in Foerster et al. (2012); Conrad et al. (2010, 2014); Peña-Barragán et al. (2011). Another approach is the Continuous Change Detection and Classification (CCDC) (Zhu and Woodcock, 2014) algorithm that models inter- and intra-annual seasonality by a sum of periodic functions for each pixel fitted with Robust Iteratively Reweighted Least Squares (RIRLS) (Street et al., 1988; Dumouchel et al., 1989) to training data. The parameters of the periodic functions can be used as features for subsequent classification. Problem-agnostic classifiers, such as Random Forests (Azzari and Lobell, 2017; Gislason et al., 2006) or Support Vector Machines (Ok et al., 2012; Inglada et al., 2015; Gómez et al., 2016; Ghazaryan et al., 2018) are broadly used. Wang et al. (2019b) transformed NDVI time series into a frequency feature space through Fast Fourier Transform (FFT) and could distinguish different crop types without labels by employing unsupervised clustering. The Breaks for Additive Seasonal and Trend (BFAST) algorithm (Verbesselt et al., 2010) similarly decomposes a satellite time series into piecewise seasonal and linear components where the remainder can be used to detect anomalies and the periodicity can be used as a feature for classification. LandTrendr (Kennedy et al., 2010) removes high-frequency components from a Landsat time series by finding a combination of successively simpler linear models. A variety of parameters can be adjusted to tune the model to specific regions and data through the number of allowed spikes (robustness to clouds) or the number of segments (model complexity). These methods have been designed to require data preprocessing, *i.e.*, atmospheric correction and cloud filtering, and some degree of direct supervision, *i.e.*, through the choice of parameters with simultaneous visual evaluations of the results. Others are tailored towards specific types of data, *e.g.*, Landsat for CCDC or MODIS for BFAST. These approaches typically focus on finding good features that can be also used for generative tasks. For instance, BFAST or LandTrendr are primarily designed to model the periodicity of seasonal phenology as a proxy to a generative vegetation model. The discrimination between different vegetation models can be then achieved with an off-the-shelf classifier using the estimated model parameters of different vegetations. For instance, Foerster et al. (2012) found phenological features that identified the individual growth stage of vegetation and combined with agrometeorological data. In this feature space, a comparatively simple parallelepiped classification rule was sufficient for accurate crop type classification.

3.2. Data-driven Methods

All model-driven approaches require thought and significant design effort to find an explicit functional or procedural formulation that represents a narrow group of desired processes, such as vegetation analysis with satellite imagery. These functions can then be tuned and evaluated using observed measurements and data. Data-driven approaches, instead, use networks of functionally identical building blocks with different parameters to model a broader family of processes. Biological neurons in our nervous system are an example that this network-based and connectionist computation principle can create complex behaviors and inspired the McCulloch and Pitts (1943) to formulate mathematical approximations of the processes in biological neurons. Later decades identified error back-propagation (Rumelhart et al., 1986) as an effective algorithm to find parameters for these basic functional building blocks. These considerations remained theoretical for many years where defining, computing, and using hand-designed features achieved better results and required less computational effort on limited hardware. A notable exception was hand-written digit classification by LeCun et al. (1998) who showed that artificial neural networks could learn to classify digits from annotated 28px by 28px images at a competitive accuracy to approaches using digit-specific features. It required further computational advances to parallelize linear algebra and matrix multiplications for Krizhevsky et al. (2012) to show that 2D convolutional neural networks can outperform model-driven approaches to classify natural images on large datasets, such as ImageNet (Deng et al., 2009). Similarly, recurrent neural networks started to compete with features calculated on a bag of words or n-gram models in natural language processing and sequence labeling (Graves, 2012) until attention-based transformer networks (Vaswani et al., 2017) started to learn complex language models from text examples gathered from the internet at large scale.

The effectiveness of 2D convolutional neural networks to extract the spatial structure in very high-resolution satellite imagery has been identified early (Marmanis et al., 2015; Volpi and Tuia, 2016; Sherrah, 2016; Audebert et al., 2016). These approaches benefited from the similarity of very high-resolution satellite imagery to natural images used in computer vision where most semantic information is encoded in the pixel neighborhood rather than the spectral dimension. Also, neural networks trained on natural images, for instance on the ImageNet dataset (Krizhevsky et al., 2012), serve as effective weight initializations to classify satellite imagery (Marmanis et al., 2015). This transfer from tasks on natural images to satellite data is less clear for multi/hyper-spectral or multi-temporal imagery. For hyperspectral imagery 3D spatio-spectral convolutions (Chen et al., 2016; Yang et al., 2018) were used early, as summarized by Li et al. (2019). To capture spectral structure, in particular, recurrence (Wu and Prasad, 2017; Mou et al., 2017) or 1D spectral convolutions (Hu et al., 2015) have also been employed and combined with 2D spatial convolutions to extract features from the pixel neighborhoods. Extracting temporal features from sequences of images requires similar considerations. For problem-agnostic time series classification on diverse time series classification benchmarks (Dau et al., 2019), common architectures of convolutional neural networks, such as the AlexNet (Krizhevsky et al., 2012) or InceptionNet (Szegedy et al., 2015) have been modified to time series by replacing 2D- with 1D convolutions (Fawaz et al., 2020; Cui et al., 2016; Wang et al., 2017). For satellite time series applications, 1D convolutions have been similarly employed more recently (Pelletier et al., 2019) while earlier works focused on recurrent neural networks (Rußwurm and Körner, 2017; Jia et al., 2017; Lyu et al., 2016; Sharma et al., 2018; Garnot et al., 2019; Turkoglu et al., 2021a). A series of publications focused on fusion networks to explicitly integrate spatial, temporal, and spectral features with dedicated convolutional and recurrent network modules (Benedetti et al., 2018; Interdonato et al., 2019; Teimouri et al., 2019; Mou et al., 2018). Combinations of attention mechanisms with recurrent layers were tested as well (Interdonato et al., 2019) following advances in natural language processing (Bahdanau et al., 2015). Rußwurm and Körner (2020) compared the mechanisms of convolution recurrence and self-attention for crop type mapping for pure time series problems while Garnot et al. (2020) demonstrated the effectiveness of self-attention models on image time series with focus on model efficiency (Garnot and Landrieu, 2020).

Data-driven methods focus, by design, on the broader distribution of tasks and take methodological inspirations from research fields that are faced with similar structures in data rather than similar applications. A method that can efficiently extract temporal features from satellite time series is applicable for a variety of downstream tasks, such as crop type mapping and land cover classification. The central requirement for data-driven methods is the availability of large-scale annotated datasets, as, for instance, the crop type labels provided via Europe's Common Agricultural Policy. To address this, new directions in data-driven research shift the focus from finding model topologies that can extract specific features on large annotated datasets towards weakly or self-supervised learning techniques (Asano et al., 2020; Caron et al., 2020) that require fewer annotations or optimization techniques (Finn et al., 2017) that can utilize annotated datasets from different-but-related tasks.

4. Contributions

So far, Chapter 1 outlined the theoretical foundation of the contributions described in this chapter. Section 1.1 introduced neural networks as universal function approximators while Section 1.2 outlined errors of approximation, estimation, and optimization that need to be balanced when designing a successful deep learning model. This design process induces prior knowledge on a particular task, as described in Section 1.3 by choosing, for instance, particular deep learning model architectures that exploit different (expected) structures in data. Further, Section 1.4 introduced two methods to integrate an approximation of uncertainty into deep learning models while Section 1.5 embedded meta-learning as a data-driven extension of transfer learning to relax the machine learning assumption of identical data distributions that is often violated in real-world applications where we can't obtain labelled datasets on the entire globe evenly. Chapter 2 introduced the particular problem space of vegetation classification with space-borne optical satellite data where Section 2.1 focused on design decisions that determine the spatial, temporal and spectral resolution of satellite data. Section 2.2 focused on particular spectral and phenological (temporal) characteristics vegetation that can be used for a classification. Finally, Chapter 3 described different model-driven and data-driven approaches for these applications in the literature.

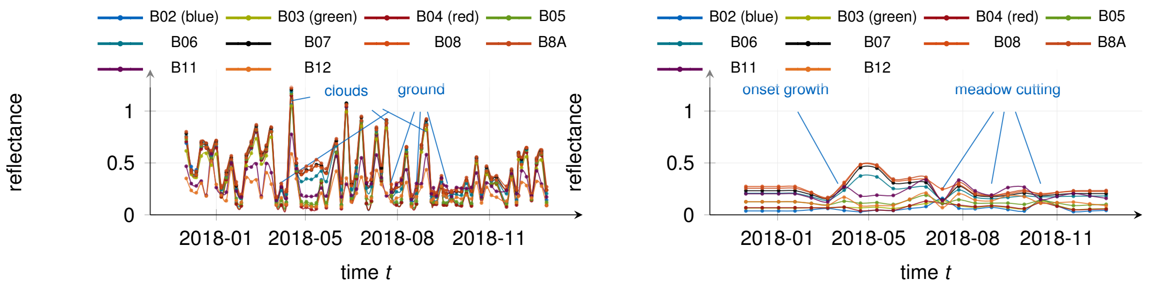
In this chapter, the main contributions towards the central research questions of this dissertation are stated. These are:

Q.1 how can we induce our prior knowledge into deep model architectures for satellite time series classification and crop type mapping?

Q.2 how can we augment existing deep learning architectures to estimate model and data uncertainties for satellite time series forecasting?

Q.3 how can we address domain shift in data-distributions, as that are induced by temporal and regional variability of representations on the Earth's surface?

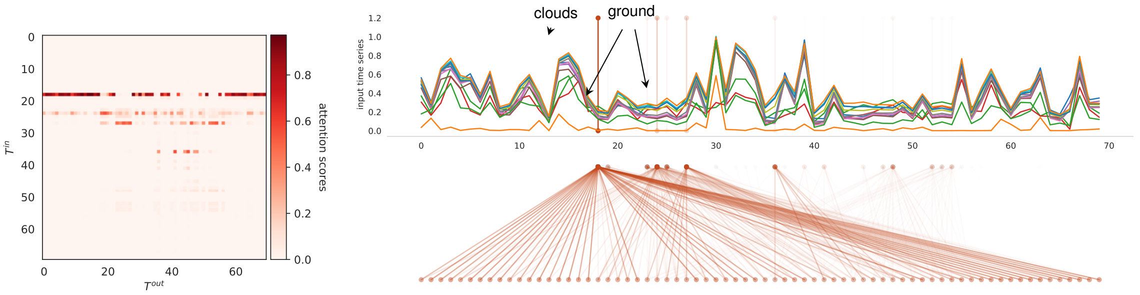
The following four sections introduce contributions to these questions. Towards **Q.1**, we compared the mechanisms of convolution, recurrence, and self-attention on raw and preprocessed satellite time series data in Rußwurm and Körner (2020), as described in Section 4.1. In Section 4.2, we (re)-implemented and tested common time series classification architectures and evaluate them on a public large-scale benchmark dataset Rußwurm et al. (2020) for crop type mapping. In Section 1.4, we investigate two mechanisms towards **Q.2** which estimate model and data uncertainty on the problem of satellite time series forecasting (Rußwurm et al., 2020a) with recurrent neural networks on MODIS vegetation data. The final contributions towards **Q.3** investigate the model-agnostic meta-learning for satellite time series classification (Wang et al., 2020) and land cover classification (Rußwurm et al., 2020b).



(a) raw time series of a meadow parcel (b) preprocessed time series of a meadow parcel

accuracy ↑	RF	LSTM-RNN	Transformer	DuPLO	MS-ResNet	TempCNN
preprocessed	0.83	$0.85^{\pm 0.01}$	$0.85^{\pm 0.02}$	**0.86**	$0.83^{\pm 0.02}$	**0.86**$^{\pm 0.00}$
raw data	0.71	**0.81**$^{\pm 0.01}$	$0.80^{\pm 0.02}$	0.79	$0.79^{\pm 0.03}$	$0.79^{\pm 0.00}$

(c) Comparison of models on preprocessed (pre) and raw datasets on the 23-class land use land cover categories. The values reported are the mean and standard deviation of three models with the best, second-best, and third-best hyperparameter sets trained on the training and validation partitions and tested on the evaluation partition.



(d) Self- attention matrix (e) The self-attention scores a pretrained Transformer self-attention model

Figure 11: Summary of main results of Rußwurm and Körner (2020) that compared various time series classification models on raw (a) and preprocessed (b) satellite time series (c). An additional qualitative assessment of attention shows how attention scores of self-attention models can extract features solely from cloud-free observations.

4.1. Satellite Time Series Classification with Deep Neural Networks

The first contribution towards **Q.1** investigated which neural network architectures are effective for satellite time series classification under different data preprocessing schemes.

Models. Following the inductive biases in neural network architectures described in Section 1.3, we chose classification methods based around recurrence, convolution, and self-attention mechanisms and proposed in the literature. These are a Long Short-Term Memory recurrent neural network (LSTM-RNN) (Rußwurm and Körner, 2017) for recurrence, the 1D-convolutional neural networks TempCNN (Pelletier et al., 2019) and a Multi-Scale 1D ResNet (MSResNet) implementation from Wang et al. (2017). Additionally, we implemented an, at the time novel and untested, self-attention Transformer (Vaswani et al., 2017) model and compared it to the DUal view Point deep Learning architecture (DuPLO) (Interdonato et al., 2019) that combines all three modules. A shallow Random Forest classifier served as a baseline to compare the deep learning models to an off-the-shelf classifier, as commonly used in remote sensing (Azzari and Lobell, 2017; Gislason et al., 2006).

Data. A central motive of this work is the evaluation of these models simultaneously on raw satellite time series data and on pre-processed satellite time-series signals provided by the GAF AG[1] to industry standards. Figures 11a and 11b show the difference of the preprocessing algorithm on a meadow time series example where individual meadow-cutting events are visible in the preprocessed data which are obscured in the raw data by atmospheric noise and clouds in the raw satellite time series. While preprocessed satellite time series are visually interpretable, some information may be lost during this computations step. For instance, the regular sampling of the preprocessing algorithm may discard some high-frequency temporal information or processing artifacts may be introduced. An example of processing artifacts can be seen in the repeated reflectance values from January to March 2018 which are likely caused by the nearest temporal interpolation of cloudy observations.

Results. In Fig. 11c, the central quantitative results of the work are summarized. The primary findings were:

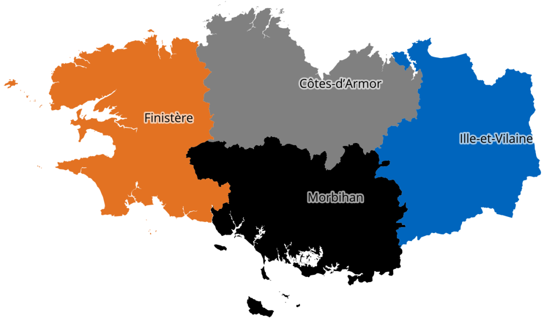
F.1 the data preprocessing was beneficial for the classification performance the accuracy,

F.2 the choice of deep learning model architecture is less important for preprocessed data, as all models achieved similar accuracies, and

F.3 recurrence and self-attention mechanisms outperformed convolutional models on unprocessed data while being equal or inferior on preprocessed time series.

Finding F.1 is consistent with the no-free-lunch theorem (Section 1.3) which suggests that including prior knowledge into one particular family of problems should lead to better results. Designers of model-driven approaches encode their knowledge into the feature extraction, i.e., preprocessing, pipelines which disentangle the feature space, as shown in the corn-meadow classifier example of Section 2.2, so that the final classification can be done accurately with a variety of suitable models, as summarized in F.2. This is exemplified prominently by the good performance of the comparatively shallow random forest classifier compared to the more complex deep learning architectures in Fig. 11c. This is consistent with the success of model-driven remote sensing techniques of the last decades that achieved good accuracies when combining off-the-shelf classifiers with sophisticated data preprocessing pipelines. The better performance of recurrence and self-attention summarized in F.3 can be explained by the prior knowledge on the structure of data that we induce into the model architectures. As outlined in Section 1.3.1, convolutional neural networks exploit patterns in a fixed local neighborhood while recurrence and self-attention can dynamically aggregate features without

[1] Gesellschaft für Angewandte Fernerkundung AG



Departments	NUTS-3	Parcels	# L1C	# L2A
Côtes-d'Armor	FRH01	221,095	178,613	178,632
Finistère	FRH02	180,565	140,645	140,782
Ille-et-Vilaine	FRH03	207,993	166,391	166,367
Morbihan	FRH04	158,522	122,614	122,708
Total		768,175	608,263	608,489

(a) The NUTS-3 departments of Brittany used for data partitioning.

(b) NUTS-3 departments of Brittany with number of field parcels and time series for each processing level.

Figure 12: Overview of the dataset regions within Brittany, France, in the Breizhcrops dataset.

any prior on the local structure, as detailed in Sections 1.3.2 and 1.3.3. The cloud identification and temporal interpolation of clouds in the preprocessed satellite time series ensure that the local temporal neighborhood of the time series remains informative. In contrast, raw satellite time series is randomly interrupted by clouds which requires feature extraction at specific cloud-free dates. This hypothesis is investigated further in Rußwurm and Körner (2020) and supporting evidence is found by a feature importance analysis where recurrence and self-attention models made predictions solely on cloud-free observations in contrast to the convolutional neural architectures. A qualitative analysis of the attention mechanism is shown in Fig. 5 where one head in the first layer of the Transformer model attends to individual cloud-free observations.

4.2. BreizhCrops Benchmark Dataset

A lack of comparable benchmark datasets in this field incentivized us to build a public and easy-to-access dataset to compare satellite time series classification models on an equal footing. The French open-data policy[2] releases crop type labels at national scale which provided the basis for the BreizhCrops (Rußwurm et al., 2020) dataset. It incorporates all Sentinel 2 satellite time series of 2017 and 2018 on a parcel-level of every agriculturally cultivated field plot in Brittany, France, at top-of-atmosphere (L1C) and bottom-of-atmosphere (L2A) processing level. Alongside the dataset, model (re-)implementations and pretrained weights are available[3] alongside a dedicated website[4] to encourage model contributions and to continually reflect the state-of-the-art in satellite time series classification on the application of crop type mapping. In addition to the LSTM-RNN, TempCNN, MS-ResNet, and Transformer evaluated in Rußwurm and Körner (2020), we additionally implement and test the convolutional Inception Time (Fawaz et al., 2020), and Omniscale-CNN (Tang et al., 2020) models, and

[2] https://www.data.gouv.fr/fr/datasets/registre-parcellaire-graphique-rpg-contours-des-parcelles-et-ilots-culturaux-et-leur-groupe-de-cultures-majoritaire/

[3] https://github.com/dl4sits/BreizhCrops

[4] https://breizhcrops.org/

	shallow	convolution				recurrence		attention
FRH04	RF	TempCNN	MS-ResNet	InceptionTime	OmniscCNN	LSTM	StarRNN	Transformer
overall accuracy	0.77	0.79	0.78	0.79	0.77	**0.80**	0.79	**0.80**
average accuracy	0.53	0.55	0.51	0.56	0.53	0.57	0.58	**0.59**
weighted f-score	0.75	0.78	0.76	0.78	0.74	**0.79**	0.77	**0.79**
kappa-metric	0.69	0.73	0.70	0.73	0.70	**0.74**	0.72	**0.74**
FRH01, 02, 04								
overall accuracy	$0.76^{\pm 0.02}$	$0.79^{\pm 0.02}$	$0.72^{\pm 0.06}$	$0.71^{\pm 0.07}$	$0.79^{\pm 0.01}$	$0.79^{\pm 0.04}$	**0.80**$^{\pm 0.02}$	**0.80**$^{\pm 0.01}$
average accuracy	$0.52^{\pm 0.01}$	$0.55^{\pm 0.01}$	$0.56^{\pm 0.05}$	$0.52^{\pm 0.04}$	$0.55^{\pm 0.02}$	$0.56^{\pm 0.03}$	$0.57^{\pm 0.01}$	**0.58**$^{\pm 0.01}$
weighted f-score	$0.75^{\pm 0.03}$	$0.79^{\pm 0.01}$	$0.71^{\pm 0.05}$	$0.70^{\pm 0.08}$	$0.77^{\pm 0.03}$	$0.78^{\pm 0.05}$	$0.78^{\pm 0.02}$	**0.80**$^{\pm 0.01}$
kappa-metric	$0.69^{\pm 0.03}$	$0.73^{\pm 0.02}$	$0.66^{\pm 0.05}$	$0.63^{\pm 0.09}$	$0.72^{\pm 0.02}$	$0.73^{\pm 0.06}$	$0.74^{\pm 0.03}$	**0.75**$^{\pm 0.02}$

(a) Model performances on top-of-atmosphere (L1C) data

	shallow	convolution				recurrence		attention
FRH04	RF	TempCNN	MS-ResNet	InceptionTime	OmniscCNN	LSTM	StarRNN	Transformer
overall accuracy	0.78	0.79	0.77	0.77	0.73	**0.80**	0.79	**0.80**
average accuracy	0.54	0.55	0.54	0.53	0.52	0.57	0.56	**0.58**
weighted f-score	0.77	0.79	0.77	0.77	0.72	**0.80**	0.79	**0.80**
kappa-metric	0.71	0.73	0.70	0.70	0.65	0.74	0.73	**0.75**
FRH01, 02, 04								
overall accuracy	$0.78^{\pm 0.02}$	$0.80^{\pm 0.01}$	$0.77^{\pm 0.02}$	$0.73^{\pm 0.04}$	$0.77^{\pm 0.05}$	$0.80^{\pm 0.02}$	$0.80^{\pm 0.01}$	**0.81**$^{\pm 0.01}$
average accuracy	$0.54^{\pm 0.01}$	$0.57^{\pm 0.01}$	$0.57^{\pm 0.03}$	$0.52^{\pm 0.01}$	$0.55^{\pm 0.03}$	$0.57^{\pm 0.01}$	$0.56^{\pm 0.00}$	**0.59**$^{\pm 0.01}$
weighted f-score	$0.77^{\pm 0.02}$	$0.80^{\pm 0.01}$	$0.76^{\pm 0.01}$	$0.69^{\pm 0.08}$	$0.75^{\pm 0.06}$	$0.80^{\pm 0.03}$	$0.80^{\pm 0.01}$	**0.81**$^{\pm 0.01}$
kappa-metric	$0.71^{\pm 0.03}$	$0.74^{\pm 0.01}$	$0.71^{\pm 0.01}$	$0.66^{\pm 0.05}$	$0.70^{\pm 0.07}$	$0.75^{\pm 0.03}$	$0.74^{\pm 0.02}$	**0.76**$^{\pm 0.02}$

(b) Model performances on bottom-of-atmosphere (L2A) data

Figure 13: Accuracy metrics of all models benchmarked on the Breizhcrops dataset, considering L1C (a) and L2A (b) data. For each table, the top part displays the performance obtained when testing on the FRH04 region while training on the three remaining areas, whereas the bottom part displays average performance (plus one standard deviation) when models were tested on FRH01, FRH02 and FRH04 regions. Bold values show the highest performance.

indclude the recently proposed StarRNN (Turkoglu et al., 2021b) recurrent neural network.

The model hyperparameters are found by training on the two distinct NUTS-3[5] regions Côtes-d'Armor and Finistère and observing the performance on the Ille-et-Vilaine region. A map of the administrative regions with several parcels is displayed in Figs. 12a and 12b alongside the frequency of field parcels. We perform the evaluation separately on top-of-atmosphere and bottom-of-atmosphere data in Figs. 13a and 13b, respectively. Each of these evaluation runs is employed with two strategies. In the top rows, we trained the models on the three regions Côtes-d'Armor (FRH01), Finistère (FRH02), and Ille-et-Vilaine (FRH03) and report accuracies on the completely unseen Morbihan (FRH04) region. This ensures complete separation of model parameters and hyperparameters from the spatially distinct test set. However, it could be that shifts in data and label distribution introduce a selection bias into the performance of the Morbihan region. To accommodate this, we also train on a leave-one-out scheme of three regions while testing on the remaining one, as shown in the bottom rows where the 1σ standard deviation between performances from these three regions is shown as well.

[5] Nomenclature des unités territoriales statistiques: administrative regions within Europe

The main contribution of this work is the provision of data and pretrained models with an open invitation to contribute to the code-base. From the large-scale model evaluations, we can still conclude that

F.1 atmospheric correction, as a common preprocessing method, did not change the classification performances of the models significantly, and

F.2 main results on satellite time-series data from Rußwurm and Körner (2020) that self-attention and recurrence could classify raw, un-preprocessed satellite time series more accurately was validated on a wider variety of models.

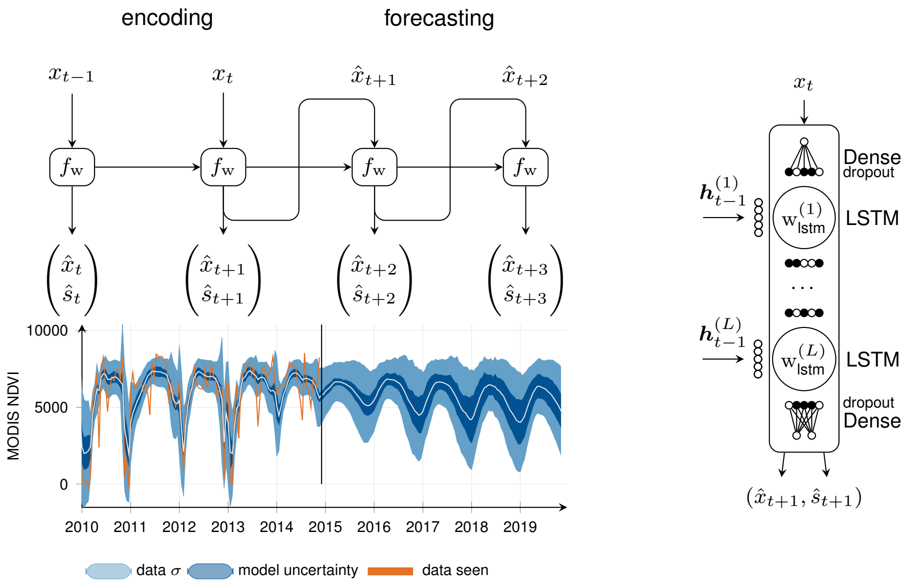
In particular, F.1 is a relevant finding as an atmospheric correction was as a single preprocessing step within the GAF preprocessing pipeline in Rußwurm and Körner (2020) that also included cloud filtering and temporal resampling. While the GAF-preprocessing raised model accuracies for all models on a similar level in Rußwurm and Körner (2020), only atmospheric correction, as in this evaluation, was not enough to improve the performances significantly.

4.3. Uncertainty Estimation for Satellite Time Series

Let's shift the focus from the objective of finding optimal model architectures for a specific family of problems, as stated in **Q.1**, towards assessing the reliability of predictions in deep learning models in general on remote sensing use-cases, as formulated in **Q.2**.

Making reliable predictions involves estimating the uncertainty of the prediction by modeling the predictive distribution based on data samples drawn from a data and weights drawn from a model-weight distribution. Several approximations can be made to estimate data and model uncertainty on deep neural networks with many parameters, as outlined in Section 1.4. Section 1.4.1 described Monte Carlo Dropout (Gal and Ghahramani, 2016) where the distribution over model weights $p(\mathrm{w}|\mathcal{D})$ given some training dataset $\mathcal{D}$, is approximated by setting weights randomly to zero which can be implemented via dropout layers at test time. While Monte Carlo Dropout reveals uncertainty in the model, it can't explain variances in the data. For this, Kendall et al. (2018) proposed a loss function to learn both prediction and its data variance, as described in Section 1.4.2. Subsequently, Kendall and Gal (2017) proposed combining these two methods for computer vision applications, since both approaches cover separate types of uncertainty, can augment existing deep learning models, and can be implemented without major computational burdens. Even though these methods have been tested in other fields, such as medical image analysis (Litjens et al., 2017), they are comparatively unknown in remote sensing. In this contribution, we implemented and tested this combination of model and data uncertainty on satellite time series forecasting in remote sensing (Rußwurm et al., 2020a).

Data. As the first step in that direction, we chose to forecast vegetation-related satellite


encoding
forecasting
x_{t-1}
x_t
$\hat{x}_{t+1}$
$\hat{x}_{t+2}$
f_w
f_w
f_w
f_w
$\begin{pmatrix}\hat{x}_t\\ \hat{s}_t\end{pmatrix}$
$\begin{pmatrix}\hat{x}_{t+1}\\ \hat{s}_{t+1}\end{pmatrix}$
$\begin{pmatrix}\hat{x}_{t+2}\\ \hat{s}_{t+2}\end{pmatrix}$
$\begin{pmatrix}\hat{x}_{t+3}\\ \hat{s}_{t+3}\end{pmatrix}$
MODIS NDVI
10000
5000
0
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
data σ
model uncertainty
data seen
x_t
$\boldsymbol{h}_{t-1}^{(1)}$
Dense
dropout
$\mathrm{w}_\mathrm{lstm}^{(1)}$
LSTM
...
$\boldsymbol{h}_{t-1}^{(L)}$
$\mathrm{w}_\mathrm{lstm}^{(L)}$
LSTM
dropout
Dense
$(\hat{x}_{t+1}, \hat{s}_{t+1})$

(a) Time series forecasting under data and model uncertainty with an RNN model and MC-Dropout on data in central Europe.

(b) RNN-LSTM forecasting model f_w used in (Rußwurm et al., 2020a).

Figure 14: Schematic illustration of the autoregressive time series forecasting process and its model f_w (Rußwurm et al., 2020a).

time-series. Assessing the quality of a predicted uncertainty is often difficult, as we lack quantitative ground truth. For a forecasting problem, on the other hand, we can evaluate the predicted future time series with its uncertainty bounds qualitatively. Looking for satellite time series data with distinct patterns, we chose a MODIS Enhanced Vegetation Index (EVI) and Normalized Difference Vegetation Index (NDVI) 16-day composite and gathered observations over the last 20 years from 2000 to 2020. For experiments on the uncertainty estimation, we chose the EVI time series in central Europe which showed regular seasonal patterns. A dataset of time series that showed a decrease in vegetation activity was chosen in Canada and an example of the effects from a volcano eruption in Peru served as a potential application for anomaly detection.

Model. Recurrent Neural Networks are a natural choice for forecasting problems, as they make a prediction $\hat{x}_{t+1}$ from current input x_t and update context vectors $\mathrm{h}_t = \{\mathbf{h}_t, \mathbf{c}_t\}$ that encode high-level representation of previous states iteratively with new observations. The RNN model can be trained to predict the single next time step by formulating a loss that minimizes the mean squared error $\frac{1}{T}\sum_{t=1}^{T}(x_t - \hat{x}_t)^2$ between the prediction $\hat{x}_t = f_\mathrm{w}(x_{t-1}, \dots, x_1, x_0)$ and next observation x_t in a sequence of T observations. This one-step-ahead prediction can then be extended indefinitely by using the prediction $\hat{x}_t$ as new input to the model to estimate $\hat{x}_{t+1}$. Figure 14 shows this prediction where the model f_w, on the right, predicts until $\hat{x}_{t+1}$ with observational data available $(x_0, x_1, \dots, x_t)$ and uses predictions $(\hat{x}_{t+1}, \hat{x}_{t+2}, \dots)$ for further multi-step forecasting.

To account for **data uncertainty**, we extended the output layer of the model to predict $\hat{s}_t =$

$\log(\sigma_t^2)$ at each prediction instance and optimize with the loss

$$\mathcal{L}(\hat{\boldsymbol{x}}, \hat{\boldsymbol{s}}, \boldsymbol{x}) = \frac{1}{T}\sum_{t=1}^{T}\frac{1}{2}\exp(-\hat{s}_t)(x_t - \hat{x}_t)^2 + \frac{1}{2}\hat{s}_t \tag{4.1}$$

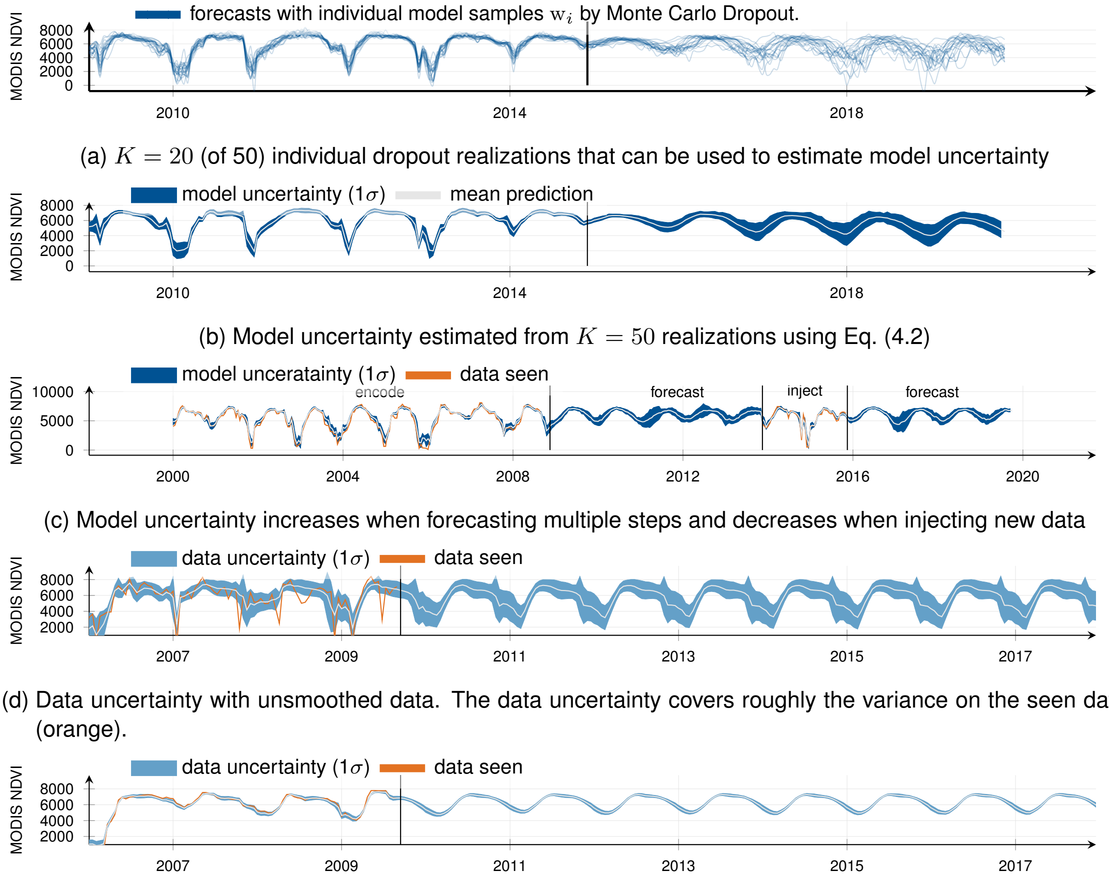
adapted from Eq. (1.35) that balances prediction accuracy $(x_t - \hat{x}_t)^2$ with a log-variance parameter $\hat{s}_t = \log(\sigma_t^2)$, as discussed in Section 1.4.2.

We approximated the **model uncertainty** with Monte-Carlo Dropout (Gal and Ghahramani, 2016) where we trained the model without dropout while setting random weights to zero by dropout at test time. This effectively approximates Eq. (1.33) by sampling, as in Eq. (1.34). Each sampled set of weights w_i can be seen as one of K realization $\{\mathrm{w}_i\}_{i=1}^{K} \sim q(\mathrm{w})$ from a family of models $q(\mathrm{w})$. From these samples, we can estimate the uncertainty through the formula of variance

$$\mathbb{V}[\hat{x}_t] \approx \underbrace{\frac{1}{K}\sum_{i=1}^{K}\hat{x}_{i,t}^2 - \left(\frac{1}{K}\sum_{i=1}^{K}\hat{x}_{i,t}\right)^2}_{\text{model}} + \underbrace{\frac{1}{K}\sum_{i=1}^{K}\hat{\sigma}_{i,t}^2}_{\text{data}} \tag{4.2}$$

of combined model and data uncertainty at each predicted observation $\hat{x}_t$. We show an example of single-step encoding and multi-step forecasting on one test time series in central Europe with regular vegetation cycles at Fig. 14a. In Figs. 15a and 15b, we drew the predictions of $K = 20$ (of 50) individual dropout realizations.

Experiments. In Rußwurm et al. (2020a), we evaluated if the model and data uncertainties estimates reflect our assumptions on data and model uncertainty. Naturally, we would expect that **model uncertainty** increases in the absence of hard observational data when we forecast multiple steps in the future since predicted values are based increasingly on previous predictions rather than data. In Fig. 15b, we can observe this where the model uncertainty remained comparatively small while predicting only the next observation in the encoding (left of the vertical line) and increased when doing multi-step forecasting. This behavior was apparent when we look at the individual realizations drawn in Fig. 15a. All dropout realizations $\{\mathrm{w}_i\}_{i=1}^{K}$ observed the identical observed data in the single-step forecast which lead to similar predictions and, conversely, a lower calculated model uncertainty. When forecasting multiple future steps, each realization w_i estimates the next value from its previous prediction. This lead to diverging predictions from the individual realizations and, following Eq. (4.2), a higher model uncertainty. In Fig. 15c, we tested this behavior further by first forecasting some steps. As before, the model realizations diverged which lead to higher model uncertainty. When we simulated new measured observations by injecting data to all model realizations, the model uncertainty decreased accordingly. This behavior makes sense from a perspective of an ensemble of model realizations since all realizations observed identical measured data instead of their forecasts which leads to convergent predictions and lower model uncertainty. In contrast to model uncertainty, we would not expect **data uncertainty** to increase when



(a) $K = 20$ (of 50) individual dropout realizations that can be used to estimate model uncertainty

(b) Model uncertainty estimated from $K = 50$ realizations using Eq. (4.2)

(c) Model uncertainty increases when forecasting multiple steps and decreases when injecting new data

(d) Data uncertainty with unsmoothed data. The data uncertainty covers roughly the variance on the seen data (orange).

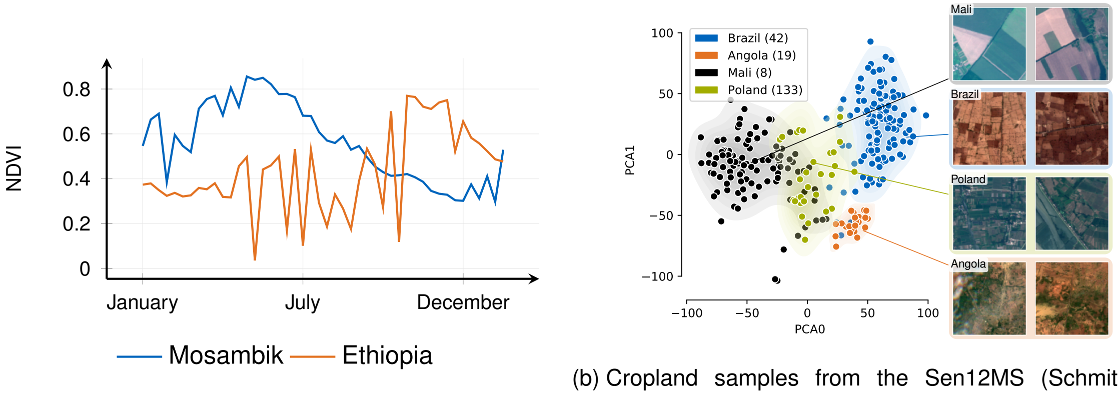
(e) Data uncertainty with data smoothed with median-3 filter. Data uncertainty is noticeably lower on data with lower variability where outliers have been artificially removed by a median filter.

Figure 15: Experiments on model and data uncertainty with satellite time series. Figure (a) shows individual dropout realizations that are used to create the uncertainty bounds of (b). In (c), we test the behavior of model uncertainty when making single-step and multi-step forecasting and observe that the model uncertainty increases in the absence of observed data. In (d) and (e), we test the characteristic of data uncertainty by reducing variance in the data by median filtering (e) which leads to a lower estimated model uncertainty.

forecasting multiple future time steps, since the data variance, for instance, induced by cloud coverage of one year is not different in one year than others. To test the data uncertainty, we artificially removed outliers in the time series data by median-3 filtering the data. This lead to a smoother signal without random outliers in the signal. The estimated data uncertainty reflected this in Figs. 15d and 15e where the data uncertainty was noticeably larger on the original time series compared to the smoothed data.

Conclusions, Limitations, and Takeaways

In summary, it is encouraging to see that uncertainty calculation methods from the computer vision literature can work out-of-the-box for remote sensing time series data. Monte Carlo Dropout (Gal and Ghahramani, 2016) provides a qualitatively reasonable estimate of model uncertainty which is easy-to-implement on existing deep learning models. Sampling multiple



(a) Cropland NDVI time series from two countries in Africa with different vegetative periods. Data from Wang et al. (2020).

(b) Cropland samples from the Sen12MS (Schmitt et al., 2019) dataset used in (Rußwurm et al., 2020b) to show the representation shift of one class on different geographical regions.

Figure 16: Distribution shift in remote sensing data. Land covers, such as cropland, vary systematically between regions. This distribution shift violates the assumption of identical data distribution if we train in one region and test in another, as we do regularly in real-world applications. Meta-learning relaxes this assumption and allows representing remote sensing data as a dataset-of-datasets.

realizations at test time can be done efficiently and does not require training multiple models from scratch, as in explicit ensembling (Lakshminarayanan et al., 2017). These experimental results also confirmed that the combination of model and data uncertainty, as proposed by (Kendall and Gal, 2017) is meaningful as two distinctly different sources of variance are captured. The data-driven recurrent neural networks provided a testbed to evaluate model and data uncertainties. However, for the application of learning vegetation time-series dynamics and to forecast future vegetation states, model-driven approaches may be better suited. A comparatively simple harmonic regression of sine curves with trend was a competitive baseline to the LSTM model that had to learn these dynamics from scratch without any prior knowledge on the regular seasonal dynamics of vegetation that know quite accurately with a frequency of one year. This is echoed by the success of more complex model-based approaches that fit multiple harmonics to vegetation time series, such as BFAST (Verbesselt et al., 2010), or LandTrendr (Kennedy et al., 2010). Nonetheless, our results in Rußwurm et al. (2020a) demonstrated that deep learning models can be augmented with uncertainty bounds relatively easily while they are rarely employed in remote sensing compared to and computer vision contexts. This contribution has been nominated for the Student Best Paper Award at IGARSS 2020 as one of ten finalists from 250 submissions which reflects interest and value in transferring methods from one domain of application, such as computer vision, to another.

4.4. Few Shot Meta Learning for Global Remote Sensing

We investigate the research question **Q.3** "how can we address domain shift in data-distributions, as that is induced by temporal and regional variability of representations on the Earth's surface?" in two publications (Rußwurm et al., 2020b; Wang et al., 2020).

The central contribution of these works is the proposal to see globally distributed remote sensing data as a *dataset-of-datasets* where each dataset is identically and independently distributed rather than a single diverse dataset.

This perspective to learn from a meta dataset-of-datasets to adapt to a new unseen individual task is a familiar framework in the field of few-shot meta-learning which is a sub-field of meta-learning (Vanschoren, 2019a). In Section 1.5.3 the relationship of machine learning, transfer learning, and few-shot meta-learning within meta-learning (Yang et al., 2020) was outlined.

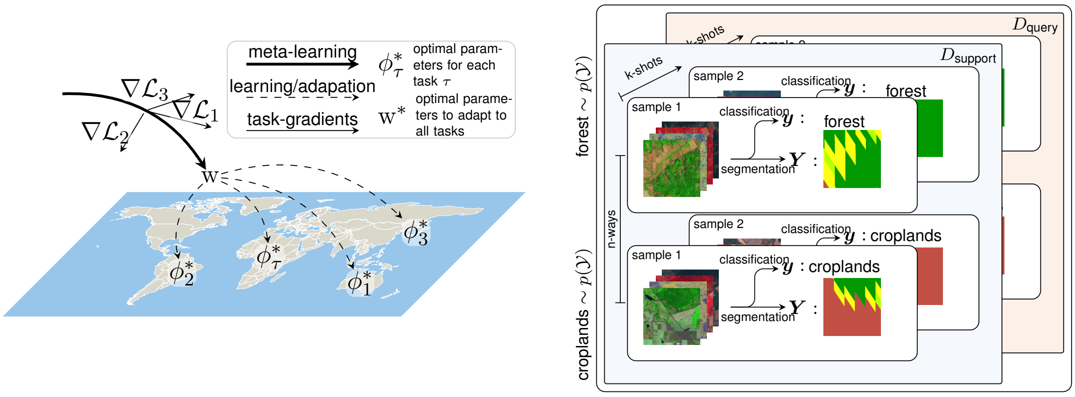
In these contributions, we investigated the applicability of few-shot meta-learning for global remote sensing applications in two publications:

1. In Rußwurm et al. (2020b) we evaluated the model-agnostic meta-learning algorithm (Finn et al., 2017) on the application of land cover classification with image data on the very high resolution DeepGlobe (Demir et al., 2018) dataset and the medium-resolution, globally distributed Sen12MS (Schmitt et al., 2019) dataset. This publication received the shared best-paper award at the CVPR EarthVision 2020 workshop.
2. In Wang et al. (2020), we focused on multi-temporal land cover classification from satellite time series with the intention to train on some continents and test on others. This publication was nominated as one of ten finalists for the Best Student Paper Award at the IGARSS 2020 conference.

First, let's qualitatively evaluate the **distribution shift** between regions to motivate the problem. In Fig. 16b, we show Sentinel 2 RGB images of cropland from different regions as ImageNet-pretrained VGG-16 features projected in a two-dimensional PCA space. While these images are all of class "cropland", we see that the representation varies significantly between regions. Here, we can empirically observe that cropland samples from Brazil are not from the same data distribution as Mali. Hence, training a machine learning model on one region and testing on another violates the assumption of identical data distribution and leads to inferior performance. If we look at NDVI satellite time series of cropland from Mosambik and Ethiopia in Fig. 16a from Wang et al. (2020), we see similarly that the representations of cropland vary within countries in Africa. While this cropland example in Mosambik has a vegetative period in June, the sample from Ethiopia has a high NDVI index in November. These examples do not originate from an identical data distribution since regional variability between regions causes a shift in representations, i.e., data distribution.

The **few-shot meta-learning framework** relaxes the assumption of identical data distributions by using a meta dataset-of-datasets or, in other terminology, a dataset-of-tasks[6]. Sam-

[6] The terminology of a task in few-shot meta-learning also includes the domain from which the input data is sampled. Following the definitions of Yang et al. (2020) strictly, we would have to speak of a "dataset of domains-and-tasks". However, in practice, we work on labeled datasets that contain both input data from some domain labeled by the task's predictive function. Hence, the distinction of domain and task has no practical consequences in the task design in few-shot meta-learning and it is assumed that the domain is implicitly part of the



(a) A schematic of the Model Agnostic Meta Learning (MAML) (Finn et al., 2017) algorithm in alg. 1 on geographic data from Rußwurm et al. (2020b); Wang et al. (2020). MAML finds a neural network initialization w that can be used to adapt do individual regions with few labelled samples.

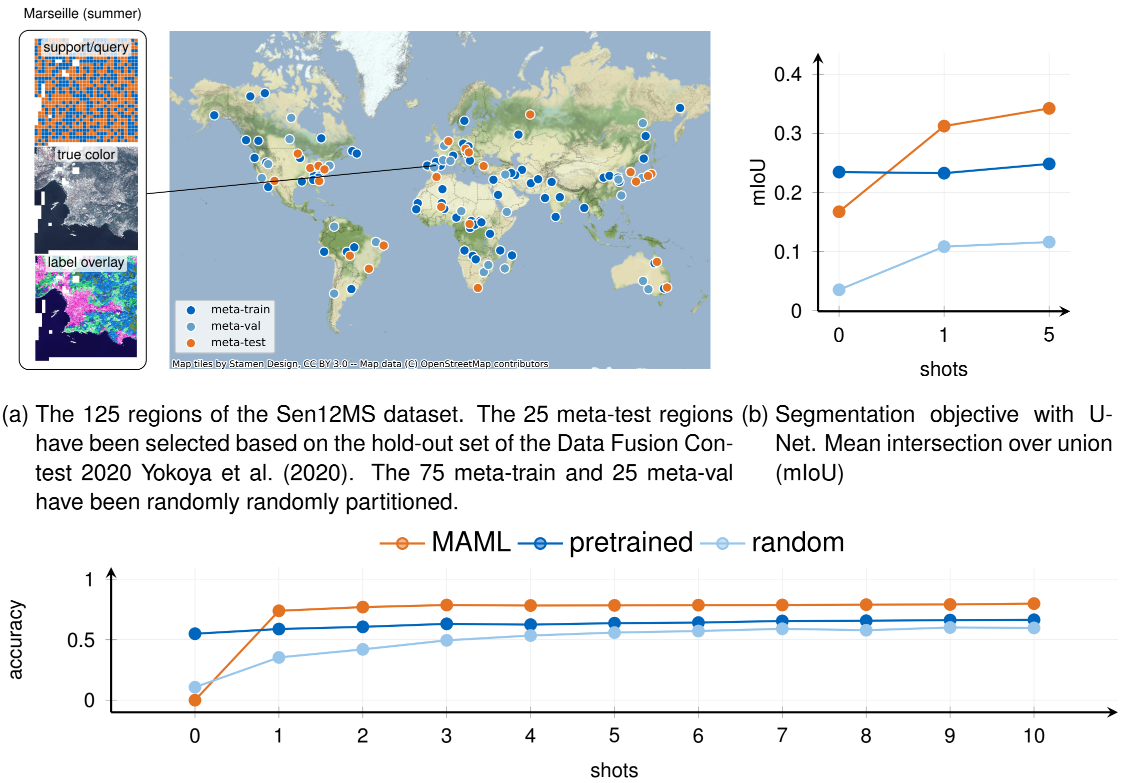
(b) Example of a Sen12MS 2-way-2-shot task from region 87 and in the summer season. Ways determines the number of classes per task while shot the number of samples per class.

Figure 17: Principle of the model-agnostic meta learning algorithm and example of a 2-way-2-shot task.

ples from each dataset must originate from one data distribution and the meta-dataset should be divided independently and identically distributed into meta-training, meta-validation, and meta-test tasks. In Fig. 17b, we show one individual task example from the Sen12MS data (Schmitt et al., 2019) to introduce the task-design in few-shot learning. Data from one task $D_{\text{support}}, D_{\text{query}} \sim p(\tau_i)$ is composed of two partitions. The support set is used to find the optimal model parameters for this task, while the separate query set is necessary to evaluate the performance on this task on unseen and independent data. In few-shot learning, we specify the task in k-shots and n-ways to highlight that it is composed of k samples of n classes in each partition. This means that in the 2-way 2-shot example of Fig. 17b four examples of the support set are used to adapt the model to the particular 2-class classification task while the performance is evaluated on the other four examples.

The **model-agnotic meta-learning** (MAML) algorithm, detailed in alg. 1 of Section 1.5, finds a neural network weight initialization w that can be used to fine-tune on each task individually. This initialization is explicitly optimized to adapt to each different-but-related domain and task within few gradient steps. Figure 17a, shows this principle on geographically distributed tasks. While machine learning methods pretrained on one global dataset would find one set of model weights that minimizes a loss for all geographical regions simultaneously, few-shot meta learning allows each geographical region to be represented as its optimal set parameters ϕ^*_τ. In model-agnostic meta-learning, network weights ϕ_τ close to the optimum ϕ^*_τ are obtained by gradient descent on the support set from the meta-learned initialization w. The meta-learned initialization w itself is obtained by optimizing a loss function (line 9 in alg. 1 of Section 1.5.3) that incorporates the adaptation to a batch of tasks (lines 5-7 in alg. 1) on a set of meta-train tasks.

individual task.



(a) The 125 regions of the Sen12MS dataset. The 25 meta-test regions have been selected based on the hold-out set of the Data Fusion Contest 2020 Yokoya et al. (2020). The 75 meta-train and 25 meta-val have been randomly randomly partitioned.

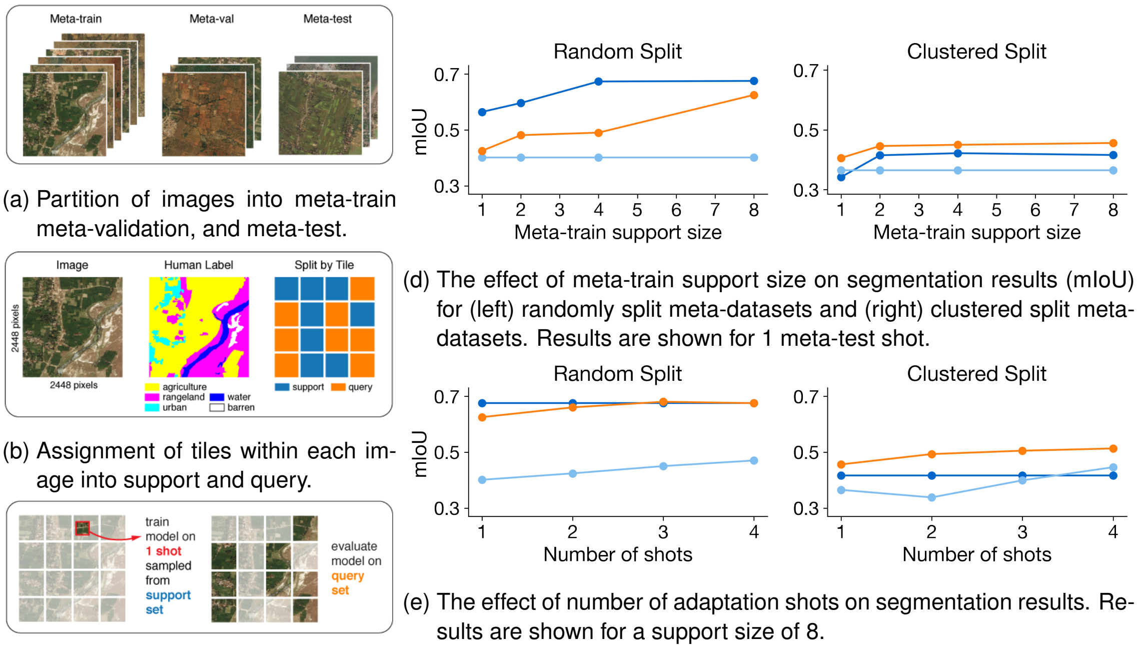
(b) Segmentation objective with U-Net. Mean intersection over union (mIoU)

(c) Evaluation on the classification objective using no data example from meta-test regions (zero-shot) up to ten examples per class.

Figure 18: Overview and resuls from the multi-spectral Sen12MS (Schmitt et al., 2019) experiments of Rußwurm et al. (2020b).

The success of few-shot meta-learning depends on the degree of distribution shift in the particular application. In short, whenever we can assume that individual model weights for each domain may explain the task better than one set of pretrained weights, few-shot meta-learning may be beneficial. We evaluated this question for three typical remote sensing applications in two publications Wang et al. (2020); Rußwurm et al. (2020b).

In Rußwurm et al. (2020b), we focused on mono-temporal satellite image data in two sets of applications. Figure 18 summarizes data and results from **land cover classification and segmentation on the Sen12MS dataset** (Schmitt et al., 2019). This dataset is composed of 125 regions that were split into 25 meta-test regions according to the Data Fusion Contest 2020 (Yokoya et al., 2020), as shown in Fig. 18a. The 75 meta-train and 25 meta-validation have been partitioned randomly. We defined 2-shot 4-way tasks where we combined two examples of four random classes of one region. We then evaluate average classification accuracy and segmentation mean intersection over union (mIoU) on all meta-test regions with an increasing number of examples, i.e., shots, per class in Figs. 18b and 18c. We used the identical neural network architectures and compared MAML-based training with regular pretraining on batches containing all images of the meta-test region and a baseline of random model initialization. In Fig. 18c, we can see that only the regularly pretrained neural network achieved good accuracy at zero-shot learning without any data from the particular target



(a) Partition of images into meta-train meta-validation, and meta-test.

(b) Assignment of tiles within each image into support and query.

(c) Example of a one-shot task.

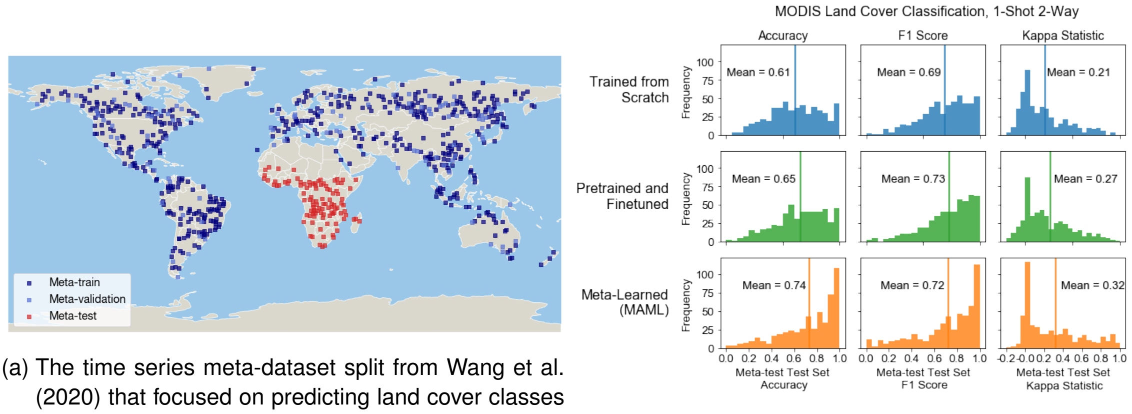
(d) The effect of meta-train support size on segmentation results (mIoU) for (left) randomly split meta-datasets and (right) clustered split meta-datasets. Results are shown for 1 meta-test shot.

(e) The effect of number of adaptation shots on segmentation results. Results are shown for a support size of 8.

Figure 19: The DeepGlobe dataset contains high resolution RGB satellite imagery with land cover labels segmented by humans. To repurpose DeepGlobe for meta-learning, we (a) split the images into meta-train, meta-validation, and meta-test sets. Then (b) each image was split into 16 sub-images, 8 of which were placed in the support set and 8 in the query set. Under such a setup, (c) we trained models on the meta-train set to segment the queries after seeing k shots from the support.

task. This is expected since the pretrained initialization optimizes for a global optimum for all tasks/regions. The meta-learned initialization w did not lead to good accuracies without adapting to individual tasks in a zero-shot manner, since the parameters from MAML-training are specifically optimized for fine-tuning on tasks. With more available data per region, as indicated by shots, we see that the random initialization improved but does not reach the pretrained model. Models adapted from the MAML-trained weights improved quickly with only one example per class (1-shot) and, in these experiments, superseded the accuracy of the pretrained initialization. These results were consistent for the classification Fig. 18c and segmentation experiments Fig. 18b.

Very-high-resolution image segmentation. In the second set of experiments, we compared few-shot meta-learning to regular pretraining on very high-resolution RGB satellite imagery of the DeepGlobe challenge (Demir et al., 2018). Since the geographic location of these images is unknown, we assumed that data from each scene originates from a different distribution. We split the scenes into meta-training, meta-validation, and meta-test partitions and divided each scene into support and query tiles, as shown in Figs. 19a and 19b. We then fine-tuned a regularly pretrained model and a MAML-trained model on a varying number of tiles from the support set of the unseen meta-test scenes and observed the mean intersection over union on the query tiles in the left columns of Figs. 19d and 19e. In contrast to the Sen12MS experiments, the regularly pretrained model outperformed the meta-learned mod-



(a) The time series meta-dataset split from Wang et al. (2020) that focused on predicting land cover classes from MODIS time series on meta-test regions in Africa.

(b) Histogram of accuracy metrics of meta-test regions in Africa after being fine-tuned on the support set.

Figure 20: A summary of the MODIS land cover time series meta-dataset (a) and main results (b) from Wang et al. (2020). The classification task is difficult, as shown in the slightly better accuracy of pretrained and meta-learned models compared to training on each region from scratch in. Still, the meta-learned model did classify more regions at high accuracy compared to pretraining which lead to better accuracy and kappa scores.

els by a large margin if the models were shown only one tile from the query scenes Fig. 19d. If the models were shown half, i.e., all 8 tiles, of the meta-test scenes, the performance of the meta-learned model was at most equal with the pretrained model, as shown in Fig. 19e. This indicates that the representation shift between scenes in the DeepGlobe dataset was not as significant as in the globally distributed Sen12MS images. Given that DeepGlobe has been proposed as a benchmark for computer vision models, it is reasonable to assume that the scenes have been selected such that they do not violate the identical distribution assumption severely when training on one scene and testing on another. We investigated this hypothesis further by artificially splitting the tiles by clustering on PCA feature to synthetically create a representation shift between scenes. This setting artificially violates the identical distribution assumption on which the pretrained model relies. The results are shown in the right column of Figs. 19d and 19e and show that now the meta-learned model outperformed the pretrained model. This experiment demonstrated that few-shot meta-learning can perform worse than pretraining in applications without a significant shift in representation between tasks.

Time series land cover classification. In the third experiment published in Wang et al. (2020), we returned to the initial motivation of training a model on samples of a data-rich region and testing on a data-sparse area where only a few data samples per class would be available. We focused on satellite time series data and chose a land cover classification label space, as it was easier to obtain than vegetation related or crop-type dataset. Land cover data were aggregated in globally distributed regions, as shown in Fig. 20a, with associated MODIS satellite time series of surface reflectances and vegetation indices. In contrast to the previous experiments, we violated the assumption of equal distribution of training and testing tasks in the meta-dataset by selecting exclusively regions in Africa as meta-test regions, as it is a realistic scenario to transfer knowledge of learned patterns from data-rich regions to others.

From the results in Fig. 20b, we can see that this only partly succeeded. The performance on target tasks in Africa varies significantly for the meta-learned time series classification model, as shown in the histograms. The task to utilize knowledge of time series classification on globally distributed data is difficult, as exemplified in the qualitative samples of cropland in Mozambique and Ethiopia in Fig. 16a. The representations of one class can vary significantly such that also the meta-learned and pretrained models outperformed the random initialization only by a small margin. Also, the time series classifier failed in many meta-test tasks where the classification performance remained random, as indicated by a kappa statistic of zero. Still, the meta-trained model did classify more tasks at high accuracy compared to the pretrained model (higher frequency at accuracies between 0.9 and 1) which lead to a better average accuracy and kappa score. Overall, in this problem setup, few-shot meta-learning did not significantly outperform regular pretraining and finetuning which highlights the potential for future research towards a more targeted knowledge transfer, e.g. through descriptive meta-data of the tasks, such as climate, elevation, coordinates, or geographical features.

In **summary**, we tested the principle of few-shot meta-learning on a wide variety of remote sensing applications. Experiments on image classification and segmentation on multi-spectral data and segmentation on very-high-resolution data were conducted in Rußwurm et al. (2020b). In Wang et al. (2020), we returned to time series classification on global land cover classification with a focus on the specific knowledge transfer from data-rich regions to data-sparse regions. As a first test of the family of few-shot meta-learning methods, we focused on a label space of land cover classification tasks. However, the methodology of few-shot meta-learning can be directly employed for vegetation-related applications and crop type mapping. During these contributions, we identified cases in which the MAML-pretraining had positive (Sen12MS) and negative (DeepGlobe, unclustered) effects on the downstream task classification performance. We empirically found that this is related to the degree of domain shift and, ultimately, the degree of violation of the identical-distribution assumption in machine learning, as we artificially increased the domain shift in the DeepGlobe experiment by creating synthetic tasks. In Wang et al. (2020), we took one additional step towards our initial motivation of training on data-rich regions and fine-tuning on others with mixed results. This leaves a potential for future research on how we can utilize the meta-data structure of geographical data (coordinates, climate, topography) for a more targeted domain transfer.

5. Discussion

A good methodology should create meaningful applications while meaningful applications inform and motivate innovations in methodology. This cycle is crucial for technological and methodological progress in real-world applications that have an impact on people's lives at a large scale[1]. Remote sensing and Earth observation create the environment for applications at a global scale that provide information to help to understand the most pressing questions on the planet. Some of these are climate change, environmental pollution, economical disparity, or food security. To make a meaningful methodological contribution in any of these applications, one must specialize in one specific field. The specific field of this dissertation was vegetation mapping with satellite time series.

It is a field that has seen rapid developments in recent years that were tightly bound to advances in data-driven feature learning in machine learning. For instance, it is known that different leaf geometry has an effect on the near-infrared reflectances of vegetation (Section 2.2) and that spectral bands in satellite sensors, such as B5, B6, B7, B8, B8A in Sentinel 2, were designed particularly for vegetation analysis (Section 2.1). Still, this potential remained unused in model-driven approaches (Section 3.1) that focused on temporal profiles of vegetation indices, such as NDVI or EVI, that only utilize one near-infrared band. While designing a feature, i.e., vegetation index, specifically for two distinct vegetation types that utilizes all near-infrared bands Sentinel 2 is certainly possible, it is too specific to a single sensor and vegetation type to justify the effort of a detailed domain-specific investigation. Meanwhile, when we learn discriminative patterns for vegetation classification from the raw data with data-driven methods (Section 3.2) and run a feature importance analysis, we see that all near-infrared bands of Sentinel 2 are distinctly utilized. This highlights the role of data-driven learning to learn relevant features from the (labelled) dataset in an automated way. In particular, the question of how we should design data-driven methods learning vegetation-related patterns with satellite time series was explored in Section 1.3.3 and placed into the context of other models with the BreizhCrops benchmark dataset Section 4.2. Here, new advances in data-driven learning helped to find domain and application-specific features.

Conversely, the limitations of today's data-driven methods become clear when viewed in the scope of some domain-specific problems. For instance, measurements or predictions without confidence (or uncertainty) are of little value in some applications. Data-driven methods, such as recurrent neural networks, can easily learn complex dynamics from data to forecast weather data, sea currents, or vegetation health (Section 4.3). However, every measured data sample and, in extension, model weight is associated with some uncertainty that accumulates and propagates when predicting categories or forecasting future values. When expressed in a Bayesian framework (Section 1.4, Eq. (1.33)) we see that marginalizing over

[1] even though the exact distinction of what is the application and what is methodology is more relative than absolute. The border ultimately depends on the perspective of the practitioner and is arbitrarily drawn.

many weights, as necessary for training and inference, is intractable and requires approximations to varying degrees. However, associating confidences with predictions is a challenging problem. When studying human cognition, which is arguably the most advanced learning system available to us, we can observe that quantifying uncertainty is a challenging problem. According to the Dual Process Theory (Evans, 1984; Kahneman, 2003), we use (at least) two distinct mechanisms: the implicit or automatic System 1 is responsible for making fast and high confident predictions, and the explicit or controlled System 2 build and refines hypotheses iteratively from perceptual data or information retrieved from memory. There are some parallels between our System 1 and the current state of deep learning, as suggested by Marcus (2020). Deep learning is similarly capable to make fast, highly confident decisions from a large pool of memorized training examples (Nakkiran et al., 2021)[2]. Object recognition in computer vision and text generation from natural language is arguably as advanced as humans using their intuition (System 1). However, both System 1 and deep learning have difficulties knowing when they are wrong. In practice, approximations and heuristics, as discussed in Sections 1.4.1 and 1.4.2 and tested in Section 4.3, may be sufficient to deactivate the intuitive System 1 and activate a slower and more expensive evaluation process (System 2) that iteratively improves the uncertainty bounds on hypotheses more similar to the posterior update from data in a Bayesian framework.

A further limitation of current deep learning methods is that they can hardly generalize beyond the training distribution in open-domain real-world datasets (Bengio et al., 2020). This observation would remain undetected if we simply tested our models on closed-domain (and randomly split) ever-increasing datasets, as almost universally practiced in today's methodological research. A larger text corpus helps language models (Brown et al., 2020) to generate reasonable and grammatically correct sentences or a generative convolutional network to produce convincingly looking faces (Karras et al., 2019). Still, deep learning models can not capture any abstract concepts familiar to us. The generated sentences lack any inherent meaning and convolutional classifiers can be fooled by simple adversarial examples (Li and Li, 2017). The example that deep learning models struggle to learn a simple identity relationship in autoencoders, as prominently displayed in Marcus (2020), stands testament that deep learning is only a single step towards increasingly intelligent systems that help us sort and evaluate the large-scale data that our sensors capture on a day-to-day basis. Moving from datasets-of-samples towards datasets-of-tasks to measure generalization on new unseen problems, as done in meta-learning (Vanschoren, 2019a) and discussed in Section 1.5.3, is a logical continuation to benchmark new models on increasingly realistic situations. In the real world, we never see the same object twice in an identical configuration, due to varying lighting, object and view rotation, deformation. Generalization beyond a closed-domain training dataset is crucial. This issue becomes apparent in applications interacting in open-domain situations, such as visual robotics (Levine et al., 2016). This need motivated the model-agnostic meta-learning algorithm (Finn et al., 2017) and spawned a new methodological field

[2] In fact, the double descent phenomenon (Nakkiran et al., 2021) suggests deep learning models smoothly interpolate between training samples in which, to some degree, contradicts the idea of bias-variance tradeoff (Section 1.2) from established machine learning.

of few-shot meta-learning (Hospedales et al., 2021). Remote sensing applications similarly interact with open-domain real-world data. We can query multi-modal satellite data of various resolutions at any place and time without major effort. Similarly, no two places on the globe are identical and, with limited label data, we need to utilize methods that can transfer their knowledge from tasks in one region to another, as explored in Section 4.4.

6. Conclusion

This dissertation sets out to improve the classification of vegetation from satellite time series by evaluating central questions related to the broader research field of learning from data. The question **Q.1** of which architectures are most suitable for satellite time series classification was addressed in the two contributions outlined in Sections 4.1 and 4.2. We showed in Rußwurm and Körner (2020) that self-attention can learn features similar to data-preprocessing in Section 4.1 while domain-specific prior knowledge in preprocessing is still beneficial for accuracy. Different data-driven models were compared in the BreizhCrops benchmark (Rußwurm et al., 2020) outlined in Section 4.2. The search for suitable model architectures has driven the recent years of machine learning research. Today, with established benchmarks to test model configurations, the search for inductive biases in the network architectures has been largely concluded and only marginal improvements are made. This is analogous to computer vision and natural language where good network configurations have been found, i.e., ResNets (He et al., 2016) and Transformer (Vaswani et al., 2017) networks. Analogous to the contributions of this dissertation, the focus has shifted from the individual model architectures towards the learning process itself.

One still unsolved research direction is the augmentation of existing deep learning models with a notion of uncertainty, as formulated in **Q.2**. Uncertainty is central to obtain data-driven results that can be trusted. Finding a distribution over model weights directly, however, is intractable due to the large weight space, as outlined in Section 1.4. Variational Bayes methods provide an approximation. Monte Carlo Dropout, as summarized in Section 1.4.1, serves as one comparatively simple approximation for model uncertainty and was explored for satellite time series forecasting in Section 4.3 alongside a heuristic for data uncertainty (Section 1.4.2). While these methods are computationally efficient, fast to implement and capture two distinct types of uncertainty they still inherit some limitations of deep learning. For instance, all realizations of Monte Carlo dropout, originate from the same source model. Any bias or inability in the model to capture some features or anomalies will affect all realizations equally and lead to an underestimated model uncertainty. If we trained multiple models independently in an explicit ensembling framework, this uncertainty would have been captured. Evaluating predictive uncertainty remains a challenging problem that may not be solved with a single concise mechanism. Following the Dual Process Theory (Evans, 1984; Kahneman, 2003), System 1 in our cognition struggles similarly to make well-calibrated predictions and we rely on a different System 2 to slowly refine hypotheses and evaluate increasingly accurate confidence bounds.

The third research question **Q.3** asked how we can address domain shift in data distribution, as induced by different environmental conditions, on the globe. This question generalizes to how we learn from multiple different-but-related datasets, as a (labelled) dataset is sampled from a task on one domain (Pan and Yang, 2009) (Section 1.5). Any intelligent system needs to be able to adapt to new situations that have not been seen before during training. Moving from benchmark datasets-of-examples towards datasets-of-datasets to measure a model's out-of-domain generalization is a logical continuation of this idea that we proposed for remote sensing applications in Rußwurm and Körner (2020); Wang et al. (2020) by adopting and testing the few-shot meta-learning framework (Finn et al., 2017) on globally distributed remote sensing data (Section 4.4).

While, certainly, improvements towards **Q.1** can be made by further refining architectures on larger labeled datasets, we may have found suitable architectures that utilize certain structures in the data. Today's deep learning methods trained on huge datasets achieve human accuracy of object recognition and improvements in text generation (Brown et al., 2020) are based on larger training datasets with the same core mechanism of self-attention (Vaswani et al., 2017). Allowing deep models to evaluate their prediction by a measure of confidence in **Q.2** remains intractable without (variational) approximations. Finding the right approximations and heuristics for specific fields of applications, as, for instance, Monte Carlo dropout, is in the scope of active research. Further refining uncertainty approximations in deep models have certainly an impact when combining symbolic models that can use known physical relationships with connectionist deep learning when we need to approximate these relationships from data. Regarding **Q.3**, we certainly need to measure out-of-distribution generalization by datasets-of-datasets, rather than relying on test-datasets from in-domain distributions. In contrast of estimating the uncertainty, our cognitive System 1 performs this out-of-distribution generalization remarkably well. This indicates that some further improvements can be made with the tools available to us. Still, the similarity of tasks within meta-datasets in few-shot meta-learning is central to the performance, as shown in the successful application of model-agnostic meta-learning on Sen12MS. The limitations shown on the (unclustered) DeepGlobe in Section 4.4 demonstrated that further research towards **Q.3** is essential to improve the effectiveness of deep learning models at a large-scale on globally distributed datasets of datasets.

Bibliography

Adam, S. P., Alexandropoulos, S.-A. N., Pardalos, P. M., and Vrahatis, M. N. (2019). No free lunch theorem: a review. In *Approximation and Optimization*, pages 57–82. Springer.

Asano, Y. M., Patrick, M., Rupprecht, C., and Vedaldi, A. (2020). Labelling unlabelled videos from scratch with multi-modal self-supervision. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H., editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Audebert, N., Le Saux, B., and Lefèvre, S. (2016). Semantic segmentation of earth observation data using multimodal and multi-scale deep networks. In *Asian conference on computer vision*, pages 180–196. Springer. v1.

Azzari, G. and Lobell, D. (2017). Landsat-based classification in the cloud: An opportunity for a paradigm shift in land cover monitoring. *Remote Sensing of Environment*, 202:64–74.

Bahdanau, D., Cho, K., and Bengio, Y. (2015). Neural machine translation by jointly learning to align and translate. In Bengio, Y. and LeCun, Y., editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Bender, E. M., Gebru, T., McMillan-Major, A., and Shmitchell, S. (2021). On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, pages 610–623.

Benedetti, P., Ienco, D., Gaetano, R., Ose, K., Pensa, R. G., and Dupuy, S. (2018). M3fusion: A deep learning architecture for multiscale multimodal multitemporal satellite data fusion. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 11(12):4939–4949.

Bengio, Y., Deleu, T., Rahaman, N., Ke, R., Lachapelle, S., Bilaniuk, O., Goyal, A., and Pal, C. (2020). A meta-transfer objective for learning to disentangle causal mechanisms. In *8th International Conference on Learning Representations (ICLR)*.

Bengio, Y., Simard, P., and Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5(2):157–166.

Bordia, S. and Bowman, S. (2019). Identifying and reducing gender bias in word-level language models. In *NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics*, NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Student Research Workshop, pages 7–15. Association for Computational Linguistics (ACL). Funding Information: We are grateful to Yu Wang and Jason Cramer for helping to initiate this project, to Nishant Subra-mani for helpful discussion, and to our reviewers for their thoughtful feedback. Bowman acknowledges support from Samsung Research. Publisher Copyright: © 2017 Association for Computational Linguistics. Copyright: Copyright 2020 Elsevier B.V., All rights reserved.; 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL HLT 2019 - Student Research Workshop, SRW 2019 ; Conference date: 03-06-2019 Through 05-06-2019.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. (2020). Language models are few-shot learners. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M. F., and Lin, H., editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Buolamwini, J. and Gebru, T. (2018). Gender shades: Intersectional accuracy disparities in commercial gender classification. In *Conference on fairness, accountability and transparency*, pages 77–91. PMLR.

Caron, M., Misra, I., Mairal, J., Goyal, P., Bojanowski, P., and Joulin, A. (2020). Unsupervised learning of visual features by contrasting cluster assignments. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H., editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Chen, Y., Jiang, H., Li, C., Jia, X., and Ghamisi, P. (2016). Deep feature extraction and classification of hyperspectral images based on convolutional neural networks. *IEEE Transactions on Geoscience and Remote Sensing*, 54(10):6232–6251.

Chung, J., Gülçehre, Ç., Cho, K., and Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. *CoRR*, abs/1412.3555.

Conrad, C., Dech, S., Dubovyk, O., Fritsch, S., Klein, D., Löw, F., Schorcht, G., and Zeidler, J. (2014). Derivation of temporal windows for accurate crop discrimination in heterogeneous croplands of Uzbekistan using multitemporal RapidEye images. *Computers and Electronics in Agriculture*, 103:63–74.

Conrad, C., Fritsch, S., Zeidler, J., Rücker, G., and Dech, S. (2010). Per-Field Irrigated Crop Classification in Arid Central Asia Using SPOT and ASTER Data. *Remote Sensing*, 2(4):1035–1056.

Cui, Z., Chen, W., and Chen, Y. (2016). Multi-scale convolutional neural networks for time series classification. *CoRR*, abs/1603.06995.

Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, 2(4):303–314.

Dau, H. A., Bagnall, A., Kamgar, K., Yeh, C.-C. M., Zhu, Y., Gharghabi, S., Ratanamahatana, C. A., and Keogh, E. (2019). The ucr time series archive. *IEEE/CAA Journal of Automatica Sinica*, 6(6):1293–1305.

De Los Campos, G., Pérez-Rodríguez, P., Bogard, M., Gouache, D., and Crossa, J. (2020). A data-driven simulation platform to predict cultivars' performances under uncertain weather conditions. *Nature communications*, 11(1):1–10.

Demir, I., Koperski, K., Lindenbaum, D., Pang, G., Huang, J., Basu, S., Hughes, F., Tuia, D., and Raskar, R. (2018). Deepglobe 2018: A challenge to parse the earth through satellite images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 172–181.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. IEEE.

Devlin, J., Chang, M., Lee, K., and Toutanova, K. (2019). BERT: pre-training of deep bidirectional transformers for language understanding. In Burstein, J., Doran, C., and Solorio, T., editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Dumouchel, W., O'Brien, F., et al. (1989). Integrating a robust option into a multiple regression computing environment. In *Computer science and statistics: Proceedings of the 21st symposium on the interface*, pages 297–302. American Statistical Association Alexandria, VA.

Eklundh, L. and Jönsson, P. (2016). Timesat for processing time-series data from satellite sensors for land surface monitoring. In *Multitemporal Remote Sensing*, pages 177–194. Springer.

Esch, T., Heldens, W., Hirner, A., Keil, M., Marconcini, M., Roth, A., Zeidler, J., Dech, S., and Strano, E. (2017). Breaking new ground in mapping human settlements from space–the global urban footprint. *ISPRS Journal of Photogrammetry and Remote Sensing*, 134:30–42.

Evans, J. S. B. (1984). Heuristic and analytic processes in reasoning. *British Journal of Psychology*, 75(4):451–468.

Fawaz, H. I., Lucas, B., Forestier, G., Pelletier, C., Schmidt, D. F., Weber, J., Webb, G. I., Idoumghar, L., Muller, P.-A., and Petitjean, F. (2020). Inceptiontime: Finding alexnet for time series classification. *Data Mining and Knowledge Discovery*, 34(6):1936–1962.

Feurer, M., Klein, A., Eggensperger, K., Springenberg, J. T., Blum, M., and Hutter, F. (2019). Auto-sklearn: efficient and robust automated machine learning. In *Automated Machine Learning*, pages 113–134. Springer, Cham.

Finn, C., Abbeel, P., and Levine, S. (2017). Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*, pages 1126–1135. PMLR.

Foerster, S., Kaden, K., Foerster, M., and Itzerott, S. (2012). Crop type mapping using spectral-temporal profiles and phenological information. *Computers and Electronics in Agriculture*, 89:30–40.

Gal, Y. and Ghahramani, Z. (2016). Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059.

Garnot, V. S. F. and Landrieu, L. (2020). Lightweight temporal self-attention for classifying satellite images time series. In Lemaire, V., Malinowski, S., Bagnall, A. J., Guyet, T., Tavenard, R., and Ifrim, G., editors, *Advanced Analytics and Learning on Temporal Data - 5th ECML PKDD Workshop, AALTD 2020, Ghent, Belgium, September 18, 2020, Revised Selected Papers*, volume 12588 of *Lecture Notes in Computer Science*, pages 171–181. Springer.

Garnot, V. S. F., Landrieu, L., Giordano, S., and Chehata, N. (2019). Time-Space Tradeoff in Deep Learning Models for Crop Classification on Satellite Multi-Spectral Image Time Series. In *IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, pages 6247–6250.

Garnot, V. S. F., Landrieu, L., Giordano, S., and Chehata, N. (2020). Satellite image time series classification with pixel-set encoders and temporal self-attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12325–12334.

Gawlikowski, J., Tassi, C. R. N., Ali, M., Lee, J., Humt, M., Feng, J., Kruspe, A. M., Triebel, R., Jung, P., Roscher, R., Shahzad, M., Yang, W., Bamler, R., and Zhu, X. X. (2021). A survey of uncertainty in deep neural networks. *CoRR*, abs/2107.03342.

Gers, F. A., Schraudolph, N. N., and Schmidhuber, J. (2002). Learning precise timing with lstm recurrent networks. *Journal of machine learning research*, 3(Aug):115–143.

Ghazaryan, G., Dubovyk, O., Löw, F., Lavreniuk, M., Kolotii, A., Schellberg, J., and Kussul, N. (2018). A rule-based approach for crop identification using multi-temporal and multi-sensor phenological metrics. *European Journal of Remote Sensing*, 51(1):511–524.

Gislason, P. O., Benediktsson, J. A., and Sveinsson, J. R. (2006). Random forests for land cover classification. *Pattern recognition letters*, 27(4):294–300.

Gitelson, A. A., Viña, A., Verma, S. B., Rundquist, D. C., Arkebauer, T. J., Keydan, G., Leavitt, B., Ciganda, V., Burba, G. G., and Suyker, A. E. (2006). Relationship between gross primary production and chlorophyll content in crops: Implications for the synoptic monitoring of vegetation productivity. *Journal of Geophysical Research: Atmospheres*, 111(D8).

Gómez, C., White, J. C., and Wulder, M. A. (2016). Optical remotely sensed time series data for land cover classification: A review. *ISPRS Journal of Photogrammetry and Remote Sensing*, 116:55–72.

Goward, S. N. and Huemmrich, K. F. (1992). Vegetation canopy par absorptance and the normalized difference vegetation index: an assessment using the sail model. *Remote sensing of environment*, 39(2):119–140.

Gower, S. T., Kucharik, C. J., and Norman, J. M. (1999). Direct and indirect estimation of leaf area index, fapar, and net primary production of terrestrial ecosystems. *Remote sensing of environment*, 70(1):29–51.

Graves, A. (2012). *Supervised Sequence Labelling with Recurrent Neural Networks*. Studies in computational intelligence. Springer, Berlin.

Graves, A., Wayne, G., and Danihelka, I. (2014). Neural turing machines. *CoRR*, abs/1410.5401.

Gretton, A., Smola, A., Huang, J., Schmittfull, M., Borgwardt, K., and Schölkopf, B. (2009). Covariate shift by kernel mean matching. *Dataset shift in machine learning*, 3(4):5.

Hagolle, O., Huc, M., Pascual, D. V., and Dedieu, G. (2010). A multi-temporal method for cloud detection, applied to formosat-2, venμs, landsat and sentinel-2 images. *Remote Sensing of Environment*, 114(8):1747–1755.

Hansen, M. C., DeFries, R. S., Townshend, J. R., and Sohlberg, R. (2000). Global land cover classification at 1 km spatial resolution using a classification tree approach. *International journal of remote sensing*, 21(6-7):1331–1364.

Hassoun, M. H. et al. (1995). *Fundamentals of artificial neural networks*. MIT press.

Haykin, S. (2007). *Neural networks: a comprehensive foundation*. Prentice-Hall, Inc.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.

Hornik, K., Stinchcombe, M., White, H., et al. (1989). Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366.

Hospedales, T. M., Antoniou, A., Micaelli, P., and Storkey, A. J. (2021). Meta-learning in neural networks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1.

Hu, W., Huang, Y., Wei, L., Zhang, F., and Li, H. (2015). Deep convolutional neural networks for hyperspectral image classification. *Journal of Sensors*, 2015.

Huete, A., Didan, K., Miura, T., Rodriguez, E. P., Gao, X., and Ferreira, L. G. (2002). Overview of the radiometric and biophysical performance of the modis vegetation indices. *Remote sensing of environment*, 83(1-2):195–213.

Inglada, J., Arias, M., Tardy, B., Hagolle, O., Valero, S., Morin, D., Dedieu, G., Sepulcre, G., Bontemps, S., Defourny, P., et al. (2015). Assessment of an operational system for crop type map production using high temporal and spatial resolution satellite optical imagery. *Remote Sensing*, 7(9):12356–12379.

Interdonato, R., Ienco, D., Gaetano, R., and Ose, K. (2019). DuPLO: A DUal view Point deep Learning architecture for time series classificatiOn. *ISPRS Journal of Photogrammetry and Remote Sensing*, 149:91–104.

Jia, K., Liang, S., Wei, X., Yao, Y., Su, Y., Jiang, B., and Wang, X. (2014). Land cover classification of landsat data with phenological features extracted from time series modis ndvi data. *Remote sensing*, 6(11):11518–11532.

Jia, X., Khandelwal, A., Nayak, G., Gerber, J., Carlson, K., West, P., and Kumar, V. (2017). Incremental Dual-memory LSTM in Land Cover Prediction. In *23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 867–876.

Jönsson, P. and Eklundh, L. (2004). Timesat—a program for analyzing time-series of satellite sensor data. *Computers & Geosciences*, 30(8):833–845.

Jordan, C. F. (1969). Derivation of leaf-area index from quality of light on the forest floor. *Ecology*, 50(4):663–666.

Jozefowicz, R., Zaremba, W., and Sutskever, I. (2015). An Empirical Exploration of Recurrent Network Architectures. In *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, Proceedings of Machine Learning Research, pages 2342–2350.

Justice, C. O., Townshend, J., Holben, B., and Tucker, e. C. (1985). Analysis of the phenology of global vegetation using meteorological satellite data. *International Journal of Remote Sensing*, 6(8):1271–1318.

Kahneman, D. (2003). Maps of bounded rationality: Psychology for behavioral economics. *American economic review*, 93(5):1449–1475.

Karras, T., Laine, S., and Aila, T. (2019). A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4401–4410.

Kelvin, S. (2003). Chloroplast-cyanobacterium comparison. `https://de.wikipedia.org/wiki/Datei:Chloroplast-cyanobacterium_comparison.svg`. [Online; accessed 12-01-21].

Kendall, A. and Gal, Y. (2017). What uncertainties do we need in Bayesian deep learning for computer vision? In *Advances in neural information processing systems*, pages 5574–5584.

Kendall, A., Gal, Y., and Cipolla, R. (2018). Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7482–7491.

Kennedy, R. E., Yang, Z., and Cohen, W. B. (2010). Detecting trends in forest disturbance and recovery using yearly landsat time series: 1. landtrendr—temporal segmentation algorithms. *Remote Sensing of Environment*, 114(12):2897–2910.

Kidger, P. and Lyons, T. (2020). Universal approximation with deep narrow networks. In *Conference on Learning Theory*, pages 2306–2327.

Kingma, D. P. and Ba, J. (2015). Adam: A method for stochastic optimization. In Bengio, Y. and LeCun, Y., editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105.

Lakshminarayanan, B., Pritzel, A., and Blundell, C. (2017). Simple and scalable predictive uncertainty estimation using deep ensembles. In Guyon, I., von Luxburg, U., Bengio, S., Wallach, H. M., Fergus, R., Vishwanathan, S. V. N., and Garnett, R., editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 6402–6413.

Lambers, H., Chapin, F. S., and Pons, T. L. (2008). Photosynthesis. In *Plant physiological ecology*, pages 11–99. Springer.

Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., and Soricut, R. (2020). ALBERT: A lite BERT for self-supervised learning of language representations. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

LeCun, Y., Bottou, L., Bengio, Y., Haffner, P., et al. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

Levine, S., Finn, C., Darrell, T., and Abbeel, P. (2016). End-to-end training of deep visuomotor policies. *The Journal of Machine Learning Research*, 17(1):1334–1373.

Li, S., Song, W., Fang, L., Chen, Y., Ghamisi, P., and Benediktsson, J. A. (2019). Deep learning for hyperspectral image classification: An overview. *IEEE Transactions on Geoscience and Remote Sensing*, 57(9):6690–6709.

Li, X. and Li, F. (2017). Adversarial examples detection in deep networks with convolutional filter statistics. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5764–5772.

Litjens, G., Kooi, T., Bejnordi, B. E., Setio, A. A. A., Ciompi, F., Ghafoorian, M., Van Der Laak, J. A., Van Ginneken, B., and Sánchez, C. I. (2017). A survey on deep learning in medical image analysis. *Medical image analysis*, 42:60–88.

Los, S., Pollack, N., Parris, M., Collatz, G., Tucker, C., Sellers, P., Malmström, C., DeFries, R., Bounoua, L., and Dazlich, D. (2000). A global 9-yr biophysical land surface dataset from noaa avhrr data. *Journal of hydrometeorology*, 1(2):183–199.

Louis, J., Debaecker, V., Pflug, B., Main-Knorn, M., Bieniarz, J., Mueller-Wilm, U., Cadau, E., and Gascon, F. (2016). Sentinel-2 sen2cor: L2a processor for users. In *Proceedings Living Planet Symposium 2016*, pages 1–8. Spacebooks Online.

Lu, Z., Pu, H., Wang, F., Hu, Z., and Wang, L. (2017). The expressive power of neural networks: A view from the width. In *Advances in neural information processing systems*, pages 6231–6239.

Luo, X., Wang, M., Dai, G., and Chen, X. (2017). A novel technique to compute the revisit time of satellites and its application in remote sensing satellite optimization design. *International Journal of Aerospace Engineering*, 2017.

Luong, T., Pham, H., and Manning, C. D. (2015). Effective approaches to attention-based neural machine translation. In Màrquez, L., Callison-Burch, C., Su, J., Pighin, D., and Marton, Y., editors, *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, pages 1412–1421. The Association for Computational Linguistics.

Lyu, H., Lu, H., and Mou, L. (2016). Learning a Transferable Change Rule from a Recurrent Neural Network for Land Cover Change Detection. *Remote Sensing*, 8(12):506.

Malinin, A. and Gales, M. (2018). Predictive uncertainty estimation via prior networks. *Advances in neural information processing systems*, 31.

Marcus, G. (2020). The next decade in ai: four steps towards robust artificial intelligence. *arXiv preprint arXiv:2002.06177*.

Marmanis, D., Datcu, M., Esch, T., and Stilla, U. (2015). Deep learning earth observation classification using imagenet pretrained networks. *IEEE Geoscience and Remote Sensing Letters*, 13(1):105–109.

Matthew, M. W., Adler-Golden, S. M., Berk, A., Richtsmeier, S. C., Levine, R. Y., Bernstein, L. S., Acharya, P. K., Anderson, G. P., Felde, G. W., Hoke, M. L., et al. (2000). Status of atmospheric correction using a modtran4-based algorithm. In *Algorithms for multispectral, hyperspectral, and ultraspectral imagery VI*, volume 4049, pages 199–207. International Society for Optics and Photonics.

McCulloch, W. S. and Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133.

Mimić, G., Brdar, S., Brkić, M., Panić, M., Marko, O., and Crnojević, V. (2020). engineering meteorological features to select stress tolerant hybrids in maize. *Scientific reports*, 10(1):1–10.

Mou, L., Bruzzone, L., and Zhu, X. X. (2018). Learning spectral-spatial-temporal features via a recurrent convolutional neural network for change detection in multispectral imagery. *IEEE Transactions on Geoscience and Remote Sensing*, 57(2):924–935.

Mou, L., Ghamisi, P., and Zhu, X. X. (2017). Deep recurrent neural networks for hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 55(7):3639–3655.

Nakkiran, P., Kaplun, G., Bansal, Y., Yang, T., Barak, B., and Sutskever, I. (2021). Deep double descent: where bigger models and more data hurt. *Journal of Statistical Mechanics: Theory and Experiment*, 2021(12):124003.

Nichol, A., Achiam, J., and Schulman, J. (2018). On first-order meta-learning algorithms. *arXiv preprint arXiv:1803.02999*.

Noroozi, M. and Favaro, P. (2016). Unsupervised learning of visual representations by solving jigsaw puzzles. In *European conference on computer vision*, pages 69–84. Springer.

Odenweller, J. B. and Johnson, K. I. (1984). Crop identification using Landsat temporal-spectral profiles. *Remote Sensing of Environment*, 14(1-3):39–5.

Ok, A. O., Akar, O., and Gungor, O. (2012). Evaluation of random forest method for agricultural crop classification. *European Journal of Remote Sensing*, 45(1):421–432.

Olsson, L., Eklundh, L., and Ardö, J. (2005). A recent greening of the sahel—trends, patterns and potential causes. *Journal of Arid Environments*, 63(3):556–566.

Pan, S. J. and Yang, Q. (2009). A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359.

Pelletier, C., Webb, G. I., and Petitjean, F. (2019). Temporal convolutional neural network for the classification of satellite image time series. *Remote Sensing*, 11(5):523.

Peña-Barragán, J. M., Ngugi, M. K., Plant, R. E., and Six, J. (2011). Object-based crop identification using multiple vegetation indices, textural features and crop phenology. *Remote Sensing of Environment*, 115(6):1301–1316.

Pennington, J., Socher, R., and Manning, C. D. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Rajeswaran, A., Finn, C., Kakade, S. M., and Levine, S. (2019). Meta-learning with implicit gradients. In Wallach, H. M., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E. B., and Garnett, R., editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 113–124.

Reed, B. C., Brown, J. F., VanderZee, D., Loveland, T. R., Merchant, J. W., and Ohlen, D. O. (1994). Measuring Phenological Variability from Satellite Imagery. *Journal of Vegetation Science*, 5(5):703–714.

Richter, R. (1996). A spatially adaptive fast atmospheric correction algorithm. *International Journal of Remote Sensing*, 17(6):1201–1214.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning representations by back-propagating errors. *nature*, 323(6088):533–536.

Rußwurm, M. and Körner, M. (2017). Temporal vegetation modelling using long short-term memory networks for crop identification from medium-resolution multi-spectral satellite images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 11–19.

Rußwurm, M., Pelletier, C., Zollner, M., Lefèvre, S., and Körner, M. (2020). Breizhcrops: A time series dataset for crop type mapping. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLIII-B2-2020:1545–1551.

Rußwurm, M., Ali, M., Zhu, X. X., Gal, Y., and Körner, M. (2020a). Model and data uncertainty for satellite time series forecasting with deep recurrent models. In *IGARSS 2020 - 2020 IEEE International Geoscience and Remote Sensing Symposium*, pages 7025–7028.

Rußwurm, M. and Körner, M. (2020). Self-attention for raw optical satellite time series classification. *ISPRS Journal of Photogrammetry and Remote Sensing*, 169:421 – 435.

Rußwurm, M., Wang, S., Körner, M., and Lobell, D. (2020b). Meta-learning for few-shot land cover classification. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 788–796.

Schmitt, M., Hughes, L. H., Qiu, C., and Zhu, X. X. (2019). Sen12ms – a curated dataset of georeferenced multi-spectral sentinel-1/2 imagery for deep learning and data fusion. In *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, volume IV-2/W7, pages 153–160.

Shalev-Shwartz, S. and Ben-David, S. (2014). *Understanding machine learning: From theory to algorithms*. Cambridge university press.

Sharma, A., Liu, X., and Yang, X. (2018). Land cover classification from multi-temporal, multi-spectral remotely sensed imagery using patch-based recurrent neural networks. *Neural Networks*, 105:346–355.

Sherrah, J. (2016). Fully convolutional networks for dense semantic labelling of high-resolution aerial imagery. *CoRR*, abs/1606.02585.

Shi, X., Chen, Z., Wang, H., Yeung, D.-Y., Wong, W.-K., and Woo, W.-c. (2015). Convolutional lstm network: A machine learning approach for precipitation nowcasting. *Advances in neural information processing systems*, 28:802–810.

Siam, M., Valipour, S., Jagersand, M., and Ray, N. (2017). Convolutional gated recurrent networks for video segmentation. In *2017 IEEE International Conference on Image Processing (ICIP)*, pages 3090–3094. IEEE.

Singha, M., Wu, B., and Zhang, M. (2016). An object-based paddy rice classification using multi-spectral data and crop phenology in assam, northeast india. *Remote Sensing*, 8(6):479.

Slaton, M. R., Raymond Hunt Jr, E., and Smith, W. K. (2001). Estimating near-infrared leaf reflectance from leaf structural characteristics. *American Journal of Botany*, 88(2):278–284.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958.

Street, J. O., Carroll, R. J., and Ruppert, D. (1988). A note on computing robust regression estimates via iteratively reweighted least squares. *The American Statistician*, 42(2):152–154.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. (2015). Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9.

Tang, W., Long, G., Liu, L., Zhou, T., Jiang, J., and Blumenstein, M. (2020). Rethinking 1d-cnn for time series classification: A stronger baseline. *CoRR*, abs/2002.10061.

Teimouri, N., Dyrmann, M., and Jørgensen, R. N. (2019). A novel spatio-temporal fcn-lstm network for recognizing various crop types using multi-temporal radar images. *Remote Sensing*, 11(8):990.

Triantafillou, E., Zhu, T., Dumoulin, V., Lamblin, P., Evci, U., Xu, K., Goroshin, R., Gelada, C., Swersky, K., Manzagol, P., and Larochelle, H. (2020). Meta-dataset: A dataset of datasets for learning to learn from few examples. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Tsai, Y.-H. H., Bai, S., Yamada, M., Morency, L.-P., and Salakhutdinov, R. (2019). Transformer dissection: An unified understanding for transformer's attention via the lens of kernel. In *EMNLP*.

Tucker, C. J. (1979). Red and photographic infrared linear combinations for monitoring vegetation. *Remote sensing of Environment*, 8(2):127–150.

Turkoglu, M. O., D'Aronco, S., Perich, G., Liebisch, F., Streit, C., Schindler, K., and Wegner,

J. D. (2021a). Crop mapping from image time series: deep learning with multi-scale label hierarchies. *CoRR*, abs/2102.08820.

Turkoglu, M. O., D'Aronco, S., Wegner, J., and Schindler, K. (2021b). Gating revisited: Deep multi-layer rnns that can be trained. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1.

Vanschoren, J. (2019a). Meta-learning. In *Automated Machine Learning*, pages 35–61. Springer, Cham.

Vanschoren, J. (2019b). Meta-learning. In *Automated Machine Learning*, pages 35–61. Springer, Cham.

Vanschoren, J., Van Rijn, J. N., Bischl, B., and Torgo, L. (2014). Openml: networked science in machine learning. *ACM SIGKDD Explorations Newsletter*, 15(2):49–60.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Velickovic, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., and Bengio, Y. (2018). Graph attention networks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Verbesselt, J., Hyndman, R., Newnham, G., and Culvenor, D. (2010). Detecting trend and seasonal changes in satellite image time series. *Remote sensing of Environment*, 114(1):106–115.

Viña, A. and Gitelson, A. A. (2005). New developments in the remote estimation of the fraction of absorbed photosynthetically active radiation in crops. *Geophysical Research Letters*, 32(17).

Volpi, M. and Tuia, D. (2016). Dense semantic labeling of subdecimeter resolution images with convolutional neural networks. *IEEE Transactions on Geoscience and Remote Sensing*, 55(2):881–893.

Wang, G., Li, W., Aertsen, M., Deprest, J., Ourselin, S., and Vercauteren, T. (2019a). Aleatoric uncertainty estimation with test-time augmentation for medical image segmentation with convolutional neural networks. *Neurocomputing*, 338:34–45.

Wang, S., Azzari, G., and Lobell, D. B. (2019b). Crop type mapping without field-level labels: Random forest transfer and unsupervised clustering techniques. *Remote sensing of environment*, 222:303–317.

Wang, S., Rußwurm, M., Körner, M., and Lobell, D. (2020). meta-learning for few-shot time series classification. In *2020 IEEE International Geoscience and Remote Sensing Symposium, IGARSS 2020*. IEEE.

Wang, Z., Yan, W., and Oates, T. (2017). Time series classification from scratch with deep neural networks: A strong baseline. In *2017 international joint conference on neural networks (IJCNN)*, pages 1578–1585. IEEE.

Weir, A. H., Bragg, P., Porter, J., and Rayner, J. (1984). A winter-wheat crop simulation-model without water or nutrient limitations. *The Journal of Agricultural Science*, 102(2):371–382.

White, M. A., de Beurs, K. M., Didan, K., Inouye, D. W., Richardson, A. D., Jensen, O. P., O'KEEFE, J., Zhang, G., Nemani, R. R., van Leeuwen, W. J., et al. (2009). Intercomparison, interpretation, and assessment of spring phenology in north america estimated from remote sensing for 1982–2006. *Global Change Biology*, 15(10):2335–2359.

Wolpert, D. H. and Macready, W. G. (1997). No free lunch theorems for optimization. *IEEE transactions on evolutionary computation*, 1(1):67–82.

Wu, H. and Prasad, S. (2017). Semi-supervised deep learning using pseudo labels for hyperspectral image classification. *IEEE Transactions on Image Processing*, 27(3):1259–1270.

Yang, Q., Zhang, Y., Dai, W., and Pan, S. J. (2020). *Transfer learning*. Cambridge University Press.

Yang, X., Ye, Y., Li, X., Lau, R. Y., Zhang, X., and Huang, X. (2018). Hyperspectral image classification with deep learning models. *IEEE Transactions on Geoscience and Remote Sensing*, 56(9):5408–5423.

Yokoya, N., Ghamisi, P., Hänsch, R., and Schmitt, M. (2020). 2020 ieee grss data fusion contest: Global land cover mapping with weak supervision [technical committees]. *IEEE Geoscience and Remote Sensing Magazine (GRSM)*, 8(1):154–157.

Zephyris (2001). Diagram of the internal structure of a leaf. `https://en.wikipedia.org/wiki/Palisade_cell#/media/File:Leaf_Tissue_Structure.svg`. [Online; accessed 12-01-21].

Zhu, Z., Wang, S., and Woodcock, C. E. (2015). Improvement and expansion of the fmask algorithm: Cloud, cloud shadow, and snow detection for landsats 4–7, 8, and sentinel 2 images. *Remote Sensing of Environment*, 159:269–277.

Zhu, Z. and Woodcock, C. E. (2012). Object-based cloud and cloud shadow detection in landsat imagery. *Remote sensing of environment*, 118:83–94.

Zhu, Z. and Woodcock, C. E. (2014). Continuous change detection and classification of land cover using all available landsat data. *Remote sensing of Environment*, 144:152–171.