GRADIENT BASED ADAPTIVE REGULARIZATION

Robert Eigenmann and Josef A. Nossek
Institute for Network Theory and Circuit Design
Munich University of Technology, D-80333 Munich, Germany
Phone: ++49 89 289 28510, Fax: ++49 89 289 28504
email: eigenmann@ei.tum.de, nossek@ei.tum.de

Abstract. A technique to optimize regularization parameters for a given supervised training problem is presented. A training database is applied to minimize a regularized cost function, and a validation database is used to estimate and optimize generalization properties by means of a modification of regularization. The performance is validated for a vowel classification task and compared to other approaches.

INTRODUCTION

In supervised training, finding the optimal complexity of a model is of major concern in the model design. Given only a very few number of training examples, an overly complex model tends to memorize the training set rather then representing the systematic structure of the data. A less complex model can represent the data only in a coarse approximation. In both cases, the model performs poorly on unknown data, expressed by a high generalization error [4].

Optimizing the model complexity therefore requires an estimation of generalization ability, achieved by a partitioning of the given data into a training set and a validation set. The model parameters are optimized in order to fit the training data, whereas the model complexity is optimized in order to minimize the validation error.

Besides growing [3, 2] and pruning techniques [6, 9, 10] which modify the number of model parameters, regularization affects the effective model complexity, encouraging smooth mappings by adding a penalty term to the error function [1, 5]. The necessary amount of regularization depends on the data structure and therefore has to be adapted during the training process. In [7] an adaptive regularization method was presented, based on the computation of the inverse Hessian of the loss function. In this work we propose an alternative, where the gradient information is sufficient to adapt the proportion of regularization.

0-7803-5673-X/99/$10.00 © 1999 IEEE

TRAINING AND ADAPTIVE REGULARIZATION

Suppose a given nonlinear and differentiable model with adaptive parameters $\boldsymbol{w}$ and an error function $E(\boldsymbol{w}, \mathcal{T})$ to be minimized in a supervised training process for a given database $\mathcal{T}$. Preventing the model from overfitting the given data, a model based regularizer term $R(\boldsymbol{w}, \boldsymbol{r})$ is added to penalize immoderate curvature mappings. The amount of regularization is parameterized with vector $\boldsymbol{r}$ and limits the effective model complexity.

Usually, the regularizer is composed of a weighted sum of penalty terms

$$R(\boldsymbol{w}, \boldsymbol{r}) = \sum_i r_i \, p(\boldsymbol{w}_i),$$

each regarding a subset of adaptive parameters $\boldsymbol{w}_i \subset \boldsymbol{w}$ and contributing with a positive weight $r_i \geq 0$ to the augmented cost function

$$C(\boldsymbol{w}, \boldsymbol{r}, \mathcal{T}) = E(\boldsymbol{w}, \mathcal{T}) + R(\boldsymbol{w}, \boldsymbol{r}). \tag{1}$$

Let $\hat{\boldsymbol{w}}(\boldsymbol{r})$ be the minimum of the cost function (1) for a given regularization weight $\boldsymbol{r}$:

$$\hat{\boldsymbol{w}}(\boldsymbol{r}) = \arg\min_{\boldsymbol{w}} C(\boldsymbol{w}, \boldsymbol{r}, \mathcal{T}). \tag{2}$$

Further let $E(\hat{\boldsymbol{w}}(\boldsymbol{r}), \mathcal{V})$ be the estimated generalization error for this solution, based on a validation dataset $\mathcal{V}$. The subject of adaptive regularization is to find optimal regularization parameters $\boldsymbol{r}^*$, minimizing the validation error in the manifold $\hat{\boldsymbol{w}}(\boldsymbol{r})$:

$$\boldsymbol{r}^* = \arg\min_{\boldsymbol{r}} E(\hat{\boldsymbol{w}}(\boldsymbol{r}), \mathcal{V}) \quad ST \; \boldsymbol{r} \geq \boldsymbol{0}. \tag{3}$$

With this, the optimal model parameters are given by

$$\boldsymbol{w}^* = \arg\min_{\boldsymbol{w}} C(\boldsymbol{w}, \boldsymbol{r}^*, \mathcal{T}). \tag{4}$$

We suppose an iterative update of parameters $\boldsymbol{w}$ and $\boldsymbol{r}$ for the combined optimization problem (2) and (3) with the following heuristic: Let (2) be optimized using an update rule

$$\boldsymbol{w}^{new} = \boldsymbol{w}^{old} - \mu \; \boldsymbol{s}(\boldsymbol{r}), \tag{5}$$

with an appropriate step size μ and a search direction

$$\boldsymbol{s}(\boldsymbol{r}) = \frac{\partial}{\partial \boldsymbol{w}} C(\boldsymbol{w}, \boldsymbol{r}, \mathcal{T}). \tag{6}$$

To avoid the effect of overfitting, this update (6) must also decrease the validation error. A necessary condition to avoid overfitting is that the search direction $\boldsymbol{s}(\boldsymbol{r})$ points in the direction of the validation error gradient. Therefore, we suggest minimizing the divergence of the gradient vectors form loss

function (1) and validation error $E(\boldsymbol{w}, \mathcal{V})$ in a least squares sense w.r.t. the regularization parameters $\boldsymbol{r}$

$$\min_{\boldsymbol{r}} \left\| \frac{\partial}{\partial \boldsymbol{w}} C(\boldsymbol{w}, \boldsymbol{r}, \mathcal{T}) - \frac{\partial}{\partial \boldsymbol{w}} E(\boldsymbol{w}, \mathcal{V}) \right\|_2^2. \tag{7}$$

Optimal regularization parameters are determined for each update (5), constituting the program for adaptive regularization

$$\boldsymbol{r}^{new} = \arg\min_{\boldsymbol{r}} \|\boldsymbol{g}_T - \boldsymbol{g}_V + \boldsymbol{g}_R(\boldsymbol{r})\|_2^2, \tag{8}$$

$$\boldsymbol{w}^{new} = \boldsymbol{w}^{old} - \mu\, \boldsymbol{s}(\boldsymbol{r}^{new}), \tag{9}$$

with the substitution

$$\boldsymbol{g}_T := \frac{\partial}{\partial \boldsymbol{w}} E(\boldsymbol{w}, \mathcal{T}), \quad \boldsymbol{g}_V := \frac{\partial}{\partial \boldsymbol{w}} E(\boldsymbol{w}, \mathcal{V}), \quad \boldsymbol{g}_R(\boldsymbol{r}) := \frac{\partial}{\partial \boldsymbol{w}} R(\boldsymbol{w}, \boldsymbol{r}).$$

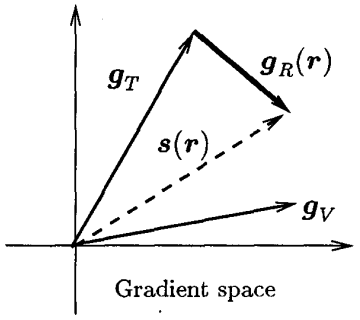

Figure 1: Key motivation of the gradient based adaptive regularization method.

Update (8), (9) is repeated, until $\boldsymbol{r}$ converges to an equilibrium $\hat{\boldsymbol{r}}$, i.e.

$$\|\boldsymbol{r}^{new} - \boldsymbol{r}^{old}\| \leq \delta \quad \Longrightarrow \quad \text{Stop adaptive regularization, } \hat{\boldsymbol{r}} = \boldsymbol{r}^{new}.$$

for a small value δ. At this point, the cost function (1) is finally optimized with constant regularization parameters $\hat{\boldsymbol{r}}$ and the complete available training data

$$\mathcal{T}^{new} = \mathcal{T} \cup \mathcal{V}.$$

THE CHOICE OF VALIDATION DATA

One critical aspect of validation is to appropriately choose the training and validation set from an available training database $\mathcal{X}$. A common technique (see e.g [7]) is to split $\mathcal{X}$ in two disjoint sets $\mathcal{T}$ and $\mathcal{V}$, whereas the choice of the split ratio is a nontrivial task.

$$\mathcal{T} \subset \mathcal{X}, \quad \mathcal{V} \subset \mathcal{X}, \quad \mathcal{T} \cup \mathcal{V} = \mathcal{X}, \quad \mathcal{T} \cap \mathcal{V} = \emptyset \tag{10}$$

According to this technique, we constitute the training set to be a subset $\mathcal{T} \subset \mathcal{X}$, e.g. half of the available training data. Optimization of the cost function with this limited representation of data will favor an overfitting of the training set $\mathcal{T}$. This overfitting effect is compensated by the regularizer, based on the estimation of generalization. The validation database $\mathcal{V}$ should represent the underlying data structure as well as possible, to give a reliable estimation of the generalization ability of the model. Therefore we propose to use all available training data for validation (including $\mathcal{T}$).

$$\mathcal{T} \subset \mathcal{X}, \quad \mathcal{V} = \mathcal{X}. \tag{11}$$

Moreover, the regularizer is prevented from immoderately favoring an arbitrary selection of validation data, provided by a selection of $\mathcal{V}$ according to (10).

The standard benchmark databases usually are provided with a disjoint test set $\mathcal{G}$ in addition to the training data $\mathcal{X}$. This dataset represents *unknown data* of the application and is applied to compare the performance of different approaches. This data is never applied to optimize any of the model weights or regularization parameters.

$$\mathcal{G} \cup \mathcal{T} = \emptyset, \quad \mathcal{G} \cup \mathcal{V} = \emptyset.$$

$\mathcal{X}$ available training data	$\mathcal{T} \subseteq \mathcal{X}$	optimization of weights
	$\mathcal{V} \subseteq \mathcal{X}$	estimation of generalization
$\mathcal{G}$ test database to benchmark the performance		

Table 1: Databases of a benchmark problem. $\mathcal{X}$ is available for optimization and $\mathcal{G}$ is used to test the performance.

REGULARIZING NEURAL CLASSIFIERS

Consider a K-class classification problem with features $\boldsymbol{x} \in \mathbb{R}^D$ to be mapped on a corresponding binary target vector $\boldsymbol{t} \in \{0;1\}^K$, indicating feature $\boldsymbol{x}$ to be of class ω_i, if $t_i = 1$ and $t_j = 0 \; \forall j \neq i$. The database $\mathcal{X} = \{\boldsymbol{x}; \boldsymbol{t}\}^P$ consists of P training tuples and is taken for validation $\mathcal{V} = \mathcal{X}$. The training set $\mathcal{T} \subset \mathcal{X}$ is composed of $P/2$ patterns from $\mathcal{X}$ with prior probabilities $\mathrm{p}(\omega_i|\mathcal{T}) = \mathrm{p}(\omega_i|\mathcal{X})$.

Let a given classifier model perform a differentiable decision function $\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{w}) : \mathbb{R}^D \mapsto (0,1)^K$, parameterized by weight vector $\boldsymbol{w}$. Denoting f_i^j as the i-th output of the classifier for an input pattern $\boldsymbol{x}^j$, the classifier is trained in order to provide an estimation of the posterior probabilities $f_i^j \stackrel{!}{=} \mathrm{p}(\omega_i|\boldsymbol{x}^j)$ by minimizing the normalized negative log-likelihood cost function [1]

$$E(\boldsymbol{w}, \mathcal{T}) = -\frac{1}{P} \sum_{j=1}^{P} \sum_{i=1}^{K} \left(t_i^j \ln(f_i^j) + (1 - t_i^j) \ln(1 - f_i^j) \right). \tag{12}$$

Let $\boldsymbol{f}(\boldsymbol{w}, \boldsymbol{x})$ be a L-layer feedforward neural network with layers $i = 1 \ldots L$ performing the nonlinear mapping

$$\boldsymbol{y}_i(\boldsymbol{x}) = \tanh(\boldsymbol{W}_i\, \boldsymbol{y}_{i-1}(\boldsymbol{x}) - \boldsymbol{s}_i), \quad \boldsymbol{y}_0(\boldsymbol{x}) := \boldsymbol{x}$$

The j-th row of weight matrix $\boldsymbol{W}_i$ contains the weights to neuron j of the i-th layer and the corresponding bias is the j-th component of vector $\boldsymbol{s}_i$. With the affine transformation $f_j(\boldsymbol{x}) = 0.5(1 + y_{L_j}(\boldsymbol{x}))$, the output values are normalized to the interval $(0, 1)$.

An appropriate regularizer for neural networks is given with the sum-of-squares penalty term (weight decay [5]) $p(\boldsymbol{w}) = \sum_i w_i^2/2$. Following [1], to ensure consistency of the weight decay with certain scaling properties of network mapping, we apply one penalty term for each layer and exclude the bias values $\boldsymbol{s}$ from regularization. Thus, the regularizer is

$$R(\boldsymbol{w}, \boldsymbol{r}) = \sum_{i=1}^{L} r_i \frac{\boldsymbol{w}_i^{\mathsf{T}} \boldsymbol{w}_i}{2} \tag{13}$$

with vector $\boldsymbol{w}_i$ containing all components of weight matrix $\boldsymbol{W}_i$. Let

$$\boldsymbol{g}_{V,i} := \frac{\partial}{\partial \boldsymbol{w}_i} E(\boldsymbol{w}, \mathcal{V}), \quad \boldsymbol{g}_{T,i} := \frac{\partial}{\partial \boldsymbol{w}_i} E(\boldsymbol{w}, \mathcal{T})$$

be the gradient w.r.t. weights $\boldsymbol{w}_i$, the least squares problem (7) with regularizer (13) has the solution

$$r_i = \frac{\boldsymbol{w}_i^{\mathsf{T}}(\boldsymbol{g}_{V,i} - \boldsymbol{g}_{T,i})}{\boldsymbol{w}_i^{\mathsf{T}} \boldsymbol{w}_i}, \; i = 1 \ldots L. \tag{14}$$

In the first few iterations of the adaptive regularization training, (14) may provide invalid negative solutions for r_i. Then, instead of solving the quadratic programming problem (7) subject to inequality constraints $r_i \geq 0$, we suggest using the absolute value of the solution (14) for the regularization parameters.

SIMULATIONS

We used the Peterson Barney vowel benchmark database [8, 11] for verification of the presented adaptive regularization method. The database consists of $K = 10$ different vowels pronounced twice by 76 speakers (33 male, 28 female and 15 children), represented by the $D = 4$ first formant frequencies. The total number of 1520 input-output pairs were split into two disjoint data sets, i.e. a data set $\mathcal{X}$ with 760 patterns, and a test set $\mathcal{G}$ with 760 patterns to report the performance of the proposed adaptive regularization algorithm. Set $\mathcal{X}$ contains data from 16 male, 14 female and 8 child speakers, set $\mathcal{G}$ contains the remaining 17 male, 14 female and 7 child speakers. According

to section 2, a subset $\mathcal{T} \subset \mathcal{X}$ of 8 male, 7 female and 4 child speakers was applied for training and the set $\mathcal{V} = \mathcal{X}$ was used for validation.

For the first simulation we applied a 2-layer neural network with 4 inputs, 10 hidden neurons and 10 output neurons, thus having a total number of 160 weights (including bias) and 2 regularization parameters. The reported results are based on 40 training runs and show the mean error with its standard deviation on the test set $\mathcal{G}$.

In the second simulation we used 100 hidden neurons (i.e. 1510 weights) to report the effect of adaptive regularization in a highly oversized network.

The results of the presented adaptive regularization (ADREG) method are compared with the early-stopping (ES) technique [1], a common applied heuristic to prevent the network from overtraining. Here, no regularization term limits the complexity. Instead, training is stopped at a minimum of the validation error. For the ES method, the same sets of training patterns $\mathcal{T}$, validation patterns $\mathcal{V}$ and test patterns $\mathcal{G}$ have been applied.

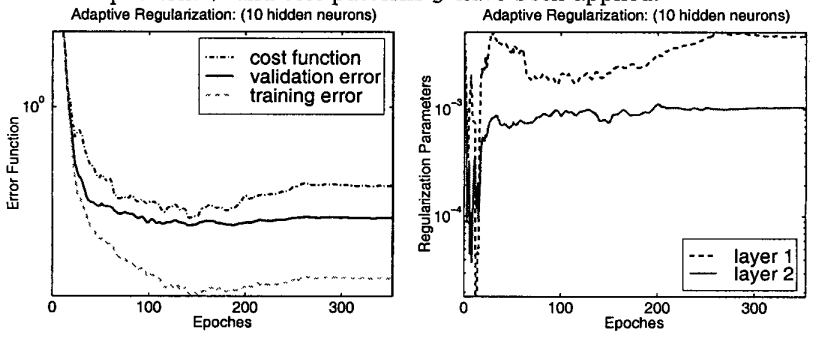

Figure 2: Typical run of the adaptive regularization

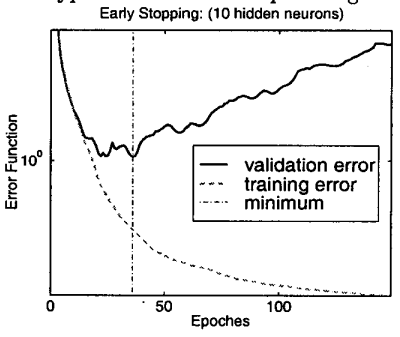

Figure 3: Typical run of the early stopping training

NN	ADREG	ES
4-10-10	12.3 (±0.74)	14.1 (±1.17)
4-100-10	11.7 (±0.52)	13.8 (±0.80)

Table 2: Misclassifications for the Peterson Barney test set $\mathcal{G}$ (in %) based on 40 runs

Lots of results for the applied database have been published recently, based on different classification strategies. Referring to [7], a different adaptive regularization method for neural classifiers combined with a pruning strategy was reported, misclassifying 12.2% of the test data without pruning and 11.9% after pruning. This method requires the computation of the inverse Hessian matrix to adapt the regularization parameters but has about the same performance as our conceptually simple gradient based update method. As reported in [7], the k-nearest neighbor (KNN) classification rule performs a higher error of 15.3% on this benchmark problem.

CONCLUSIONS

This paper presented a method to adapt regularization parameters for a given supervised training problem. The benefit of this approach to other concepts is based on the conceptually simple update rule for the regularization parameters, where only the gradient of the error function is required. Compared to other techniques working with a validation database (like early stopping), the presented method allows the use of all available data for training, after optimal regularization parameters are determined. Therefore, the model approximates the underlying data structure more exactly, resulting in an improvement of generalization. Is was shown, that with this method, even highly oversized models do not tend to overfit the given training data.

REFERENCES

[1] C. Bishop, **Neural Networks for Pattern Recognition**, Clarendon Press, 1995.

[2] R. Eigenmann and J. Nossek, "Constructive and Robust Combination of Perceptrons," in **Proc. of the International Conference on Pattern Recognition**, 1996, vol. IV, pp. 195–198.

[3] S. Fahlman, "The cascade-correlation learning architecture," in D. Touretzky, ed., **Advances in Neural Information Processing Systems 2**, San Mateo, CA: Morgan Kaufmann Publishers, 1990, pp. 524–532.

[4] S. Geman, E. Bienenstock and R. Dourstat, "Neural networks and the bias/variance dilemma," **Neural Computation**, vol. 4, no. 1, pp. 1–58, 1992.

[5] A. Krogh and J. Hertz, "A simple weight decay can improve generalization," in J. Moody, S. Hanson and R. Lippmann, eds., **Advances in Neural Information Processing Systems 4**, San Mateo, CA: Morgan Kaufmann Publishers, 1992, pp. 450–957.

[6] Y. Le Cun, J. Denker and S. Solla, "Optimal brain damage," in D. Touretzky, ed., **Advances in Neural Information Processing Systems 2**, San Mateo, CA: Morgan Kaufmann Publishers, 1990, pp. 598–603.

[7] L. Nonboe Andersen, J. Larsen, L. Hansen and M. Hintz-Madsen, "Adaptive Regularization of Neural Classifiers," in **Proc. of the IEEE Workshop on Neural Networks for Signal Processing**, 1997, pp. 24–33.

[8] G. Peterson and H. Barney, "Control Methods Used in a Study of the Vowels," **JASA**, vol. 24, pp. 175–184, 1952.

[9] R. Reed, "Pruning algorithms – a survey," **IEEE Trans. on Neural Networks**, vol. 4, no. 5, pp. 740–747, 1993.

[10] J. Sietsma and R. Dow, "Creating Artificial Neural Networks that Generalize," **Neural Networks**, vol. 4, pp. 67–79, 1991.

[11] R. Watrous, "Current Status of PetersonBarney Vowel Formant Data," **JASA**, vol. 89, pp. 2459–2460, 1991.