Cross-Entropy vs. Squared Error Training: a Theoretical and Experimental Comparison

Pavel Golik[1], *Patrick Doetsch*[1], *Hermann Ney*[1,2]

[1] Human Language Technology and Pattern Recognition, Computer Science Department, RWTH Aachen University, 52056 Aachen, Germany
[2] Spoken Language Processing Group, LIMSI CNRS, Paris, France
{golik,doetsch,ney}@cs.rwth-aachen.de

Abstract

In this paper we investigate the error criteria that are optimized during the training of artificial neural networks (ANN). We compare the bounds of the squared error (SE) and the cross-entropy (CE) criteria being the most popular choices in state-of-the art implementations. The evaluation is performed on automatic speech recognition (ASR) and handwriting recognition (HWR) tasks using a hybrid HMM-ANN model. We find that with randomly initialized weights, the squared error based ANN does not converge to a good local optimum. However, with a good initialization by pre-training, the word error rate of our best CE trained system could be reduced from 30.9% to 30.5% on the ASR, and from 22.7% to 21.9% on the HWR task by performing a few additional "fine-tuning" iterations with the SE criterion.

Index Terms: hybrid approach, training criterion for ANN training, automatic speech recognition, handwriting recognition

1. Introduction

While the artificial neural networks become more and more substantial parts of state-of-the-art automatic speech and handwriting recognition systems, various questions arise considering the ANN architecture and the fundamentals of training.

It can be shown that the true posterior probability is a global minimum for both the cross-entropy (CE) and squared error (SE) criteria [1, p. 100]. Thus, in theory an ANN can be trained equally well by minimizing either function, as long as it is capable of approximating the true posterior distribution arbitrarily close. When modeling a distribution, SE is bounded and the optimization is therefore more robust to outliers than minimzation of CE. In practice, however, CE mostly leads to faster convergence and better results in terms of classification error rates. Hence, SE became less popular over the last years.

In the literature, the error criterion is often considered only in combination with certain output activation functions, resulting in what is sometimes called "natural pairing", that allows to express the gradient in the last layer as the difference between the actual output and the desired reference output. This choice can also be motivated by means of *canonical link functions*, resulting from the assumption of target variable to have a distribution from the exponential family [2, p. 217], which does not have to hold true in all cases. While this is a convenient choice from the point of view of implementation and training time, in principle, every combination is possible. Thus, for this work, we decided to change only one thing at a time and compare the SE and CE criteria using the same activation function (softmax).

Previous investigations of the error function have usually been evaluated on comparably small tasks. Nowadays, however, with increasing amount of available data and computational power, ANNs with many millions of free parameters can be trained within a few days. The number of classes (i.e. HMM states) in ASR tasks also goes up to several thousands. Previous works mostly investigate the quality of the estimated posterior distributions on synthetic data by measuring deviation from known distrubution [3][4]. This motivates an experimental evaluation on a real-world task with current state-of-the-art systems. The goal of this paper is to analyze the two criteria, both from a theoretical and experimental point of view.

This paper is organized as follows. We analyze both criteria from a theoretical perspective and compare their bounds in Section 2. Then we summarize the training procedure of ANNs and investigate the convergence properties in Section 3. Section 4 provides experimental results of our investigation and the conclusions are drawn in Section 5.

2. Theoretical analysis of training criteria

In this section, we will consider two training criteria derived in [5] and discuss differences in their potential effect on parameter learning.

We assume a normalized "acoustic" model whose output nodes represent the classes, i.e. the labels of the associated HMM states. We will use an ANN with the softmax operation in the output layer. For an observed input vector x, the model computes a score for each class c. These scores can be interpreted as estimates of a class posterior probability $q(c|x)$ and are normalized. In general, the model has a set of free parameters θ: $q(c|x) = q_\theta(c|x)$. To simplify the notation when considering the training criteria, we will drop the parameters θ and simply use $q(c|x)$. We are given annotated training data $\{(x_n, c_n) : n = 1, ..., N\}$, and we consider the following two training criteria:

- cross-entropy (empirical equivocation or logarithm):

$$\hat{q}(c|x) = \underset{\{q(c|x)\}}{\operatorname{argmin}} \Big\{ -\sum_n \log q(c_n|x_n) \Big\} \quad (1)$$

- squared error:

$$\hat{q}(c|x) = \underset{\{q(c|x)\}}{\operatorname{argmin}} \Big\{ \sum_n \sum_c [q(c|x_n) - \delta(c, c_n)]^2 \Big\} \quad (2)$$

Both training criteria [1, p. 100] have the attractive property that, in the case of a model $q(c|x)$ with a sufficient degree of flexibility, the optimum solution $\hat{q}(c|x)$ is the true class poste-

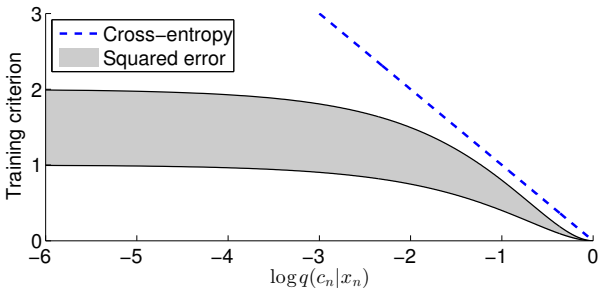

Figure 1: *Training criteria (SE and CE) as a function of* $\log q(c_n|x_n)$*, i.e. the model probability of the correct class.*

rior distribution $pr(c|x)$ that is computed from the distribution of the training data $pr(x,c) = \frac{1}{N}\sum_{n=1}^{N}\delta(x_n,x)\cdot\delta(c_n,c)$ with the Kronecker delta or delta function $\delta(\cdot,\cdot)$ for discrete and continuous-valued arguments, respectively.

We will study the two training criteria in more detail. For this purpose, we will visualize the training criteria. For the CE criterion, this is easy since the criterion depends only on a single value, namely the score of the correct class $q(c_n|x_n)$. For the SE criterion, the visualization is more difficult, since the criterion depends on the set of all scores $\{q(c|x_n)\}$.

Therefore, we will derive lower and upper bounds for the local squared error as a function of the score of the correct class $q(c_n|x_n)$. We re-write the squared error in point x_n:

$$\sum_c [q(c|x_n) - \delta(c,c_n)]^2 = \\ = \quad [1-q(c_n|x_n)]^2 + \sum_{c\neq c_n} q^2(c|x_n) \qquad (3)$$

For a given $q(c_n|x_n)$, we want to bound this error from below and from above. The upper bound is attained if the remaining probability $[1-q(c_n|x_n)]$ is assigned to a single rival class, while the lower bound is attained if it is uniformly distributed over all rival classes. Hence we obtain:

$$[C-1]\cdot\Big[\frac{1-q(c_n|x_n)}{C-1}\Big]^2 \quad \leq \quad \sum_{c\neq c_n} q^2(c|x_n) \qquad (4)$$

$$\leq \quad [1-q(c_n|x_n)]^2 \qquad (5)$$

Thus we obtain the upper bound:

$$\sum_c [q(c|x_n) - \delta(c,c_n)]^2 \leq \\ \leq \quad [1-q(c_n|x_n)]^2 + [1-q(c_n|x_n)]^2 \qquad (6)$$

$$= \quad 2\cdot[1-q(c_n|x_n)]^2 \qquad (7)$$

and the lower bound:

$$\sum_c [q(c|x_n) - \delta(c,c_n)]^2 \geq \\ \geq \quad [1-q(c_n|x_n)]^2 + [C-1]\cdot\Big[\frac{1-q(c_n|x_n)}{C-1}\Big]^2 \qquad (8)$$

$$= \quad \frac{C}{C-1}\cdot[1-q(c_n|x_n)]^2 \;\geq\; [1-q(c_n|x_n)]^2 \qquad (9)$$

The result is plotted in Figure 1: the squared error criterion is bounded from below and from above by quadratic functions in $q(c_n|x_n)$ (grey area); it is limted to values in the intervall $[0,2]$. The cross-entropy criterion is simply the negative logarithm of $q(c_n|x_n)$ and therefore unlimited. Figure 3 shows the distribution of $\log q(c_n|x_n)$ and the corresponding error values on a development set.

The squared error criterion can be interpreted as an error "count" in the training data: the count varies between the lower (no strong rival class) and the upper bound (one single rival class). There is a smooth transition from a totally correct classification and a totally wrong classification. The optimal solution is attained at $\hat{q}(c|x) = pr(c|x)$.

So far, the discussion and comparison of the two training criteria has focussed on the properties of the optimal solutions as such. However, there is another important issue related to any training criterion, namely its associated (gradient) search strategy and its convergence behaviour. This issue will be discussed in the next section.

3. Convergence analysis

This section briefly summarizes the training of an ANN via backpropagation and presents an investigation of how the gradient computation affects the convergence.

An ANN has the model parameters $\theta = \{w_{ij}^l \in \mathbb{R}\}$. The output values of each neuron are calculated by applying nonlinear activation functions to the linear combination of the connected inputs. We will only outline formalisms that are relevant for the further analysis and refer the interested reader e.g. to [2] for details. Omitting the bias terms, the input to the final layer is calculated as $z_c^L = \sum_i w_{ic}^L y_i^{(L-1)}$ and the output y^L results from a transformation with the softmax function: $y_c^L = \sigma(z_c) =: q_\theta(c|x_n)$.

Given a training set of correctly labeled samples $\big\{(x_n,c_n) : 1\leq n\leq N\big\}$, the training consists of minimizing the global error function $E_{global} = \frac{1}{N}\sum_n E(q(\,\cdot\,|x_n),c_n)$. The minimization is usually done by stochastic gradient descent and the weight update is performed in the direction of the negative gradient, which is given by:

$$\frac{\partial E}{\partial w_{ij}^l} = \frac{\partial E}{\partial z_j^l}\cdot\frac{\partial z_j^l}{\partial w_{ij}^l} = \Delta_j^l\cdot y_i^{(l-1)} \qquad (10)$$

The error signal Δ^L in the final layer depends on both the error criterion and the activation function.

As stated in Section 1, some combinations of error criterion and activation function in the last layer, such as {SE, linear} or {CE, softmax}, result in a convenient form of the error Δ_c^L: $y_c - \delta(c_n,c)$. In case of the pair {SE, softmax}, the error becomes $\sum_k (y_k - \delta(k,c_n))\cdot y_k[\delta(c,k) - y_c]$.

How does the error signal (and therefore the gradient in Eq. 10) depend on the actual output of a neuron? Following a similar approach as in [6], we look at the functional dependencies between the two variables. For the CE criterion, the dependency is linear:

$$\Delta_c^{CE}(y_c) = \begin{cases} y_c - 1 & c = c_n \\ y_c & c\neq c_n \end{cases} \qquad (11)$$

while in case of SE, the dependency contains quadratic and cubic terms:

$$\Delta_c^{SE}(y_c) = \begin{cases} 4y_c^2 - 2y_c(1+\sum_k y_k^2) & c = c_n \\ 2y_c^2 - 2y_c(-y_{c_n} + \sum_k y_k^2) & c\neq c_n \end{cases} \qquad (12)$$

In contrast to Δ_c^{CE}, these polynomials can attain small values not only when the output is nearly optimal, but also when it comes close to the opposite value. This can cause the gradient to vanish where the learning should have been continued.

4. Experimental results

In this chapter we describe our evaluation environment and the results obtained on an ASR and an HWR tasks.

There are several methods of integration of neural networks in an ASR system. We will focus on the *hybrid approach* [1] rather than Tandem [7], because it allows to observe the differences between ANNs more directly. The main idea here is to plug in the state posterior estimation of the ANN into the likelihood computation needed in a conventional HMM system: $\tilde{p}(x_n|c) \propto q_\theta(c|x_n)/p(c)^\alpha$. The scaling factor α was empirically tuned on the development data and set to 0.3 for both the ASR and the HWR task.

Note that the recognition result depends not only on the ANN outputs, but also on the other knowledge sources in the overall system, i.e. language model and pronunciation lexicon (ASR) or spelling dictionary (HWR).

4.1. Experimental setup

ASR. The ANN mini-batch training for the ASR task is performed with a multi-layer perceptron (MLP) on 50 hours of speech from the Quaero [8] English database *train11*, which amounts to ca. 16 million input samples. The development and evaluation sets consist of ca. 3.5 hours of speech each, corresponding to about 1.2 million samples. Every input vector is a concatenation of 16-dimensional MFCC vector with its first temporal derivative and the first component of the second derivative (16+16+1=33). A sliding window of 9 frames is applied to build the 297-dimensional vector as input to the ANN. A 4-gram language model (LM) is used during the recognition.

We chose a simple topology of one hidden layer with 2000 nodes and a softmax output layer with 4500 nodes corresponding to the generalized triphones tied by a phonetic classification and regression tree (CART). The number of trainable weights amounts to approx. 600k.

The ASR baseline system is a conventional GMM-HMM based model trained on the same database w.r.t. the maximum likelihood (ML) criterion. We applied linear discriminant analysis (LDA) to 9 consecutive MFCC frames to obtain the final 45-dimensional features. The GMM with a globally pooled diagonal covariance matrix consists of approx. 660k densities, which corresponds to about 30M trainable parameters.

HWR. For the HWR task the IAM database [9] was used, which consists of handwritten English sentences built upon the LOB corpus [10]. The data is provided in three disjoint sets with 747 text lines for training, 116 text lines for development, and 336 text lines for evaluation. A sliding cosine window of 30px width with a shift of 3px is applied to the images producing 4 million frames for the training set, 600k frame for the development set, and 2 million frames for the evaluation set. Each frame is normalized by its 1st and 2nd order moments and reduced by PCA to 20 components. Concatenating the horizontal and vertical moments to the PCA reduced frame results in a 24-dimensional feature vector [11]. We use a 3-gram LM built upon the LOB, Brown, and Wellington corpora [10, 12, 13] with a vocabulary containing the 50k most frequent words.

As baseline system we use a conventional GMM-HMM based model trained on the same data w.r.t. the ML criterion. The GMM consists of approx. 33k densities. A globally pooled diagonal covariance matrix is used leading to about 800k trainable parameters. Experiments were performed with bidirectional Long-Short-Term-Memory recurrent neural networks (LSTM-BRNN) [14]. The LSTM-BRNN is composed of memory cells. Each memory cell contains a central linear node which is controlled by several gating nodes. This structure allows the network to use long-term contextual information selectively such that it is less affected by the vanishing gradient problem compared to tradition recurrent neural network models [15]. The LSTM-BRNN was successfully applied to several sequence learning tasks and evolved to be the topology of choice in HWR.

In our experiments the LSTM-BRNN contained two hidden layers with 100 and 200 nodes respectively. Through the recurrent connections no window on frame level is required and therefore the input layer of the LSTM-RNN consists of one node for each of the 24 dimensions of an input vector. The softmax output layer consists of 79 nodes representing the character labels of the IAM database. In total approx. 800k weights have to be trained.

Table 1: *Effect of random initialization on the convergence. Frame and word error rates in percent.*

Task	System	dev		eval	
		FER	WER	FER	WER
ASR	MLP CE	70.5	24.9	71.6	30.9
	MLP SE	81.1	41.1	82.0	47.4
HWR	BRNN CE	13.2	17.7	15.2	22.7
	BRNN SE	18.3	19.0	19.9	24.4

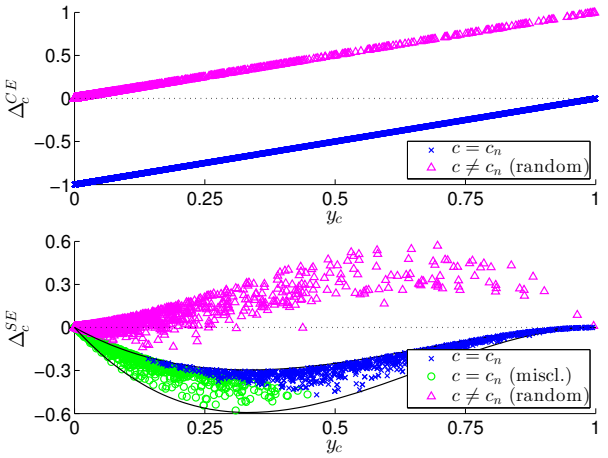

Figure 2: *Error signal Δ_c as a function of network output for CE (top) and SE (bottom). In case of SE, the correct class is plotted in green whenever the frame was misclassified.*

4.2. Results

4.2.1. Convergence issues

Starting with the ASR task we trained two MLPs with randomly initialized weights. Table 1 shows that the SE network has not reached the performance of CE. In fact, one could argue that the SE training requires different parametrization and that the experience of training CE networks does not apply to SE. In order to obtain the best possible SE system, we performed a grid search over various parameters like the learning rate, the weight initialization range and the stopping condition. We found it helpful to increase the learning rate from from 0.008 to 0.01 and to reduce the batch size from 2048 to 256. All other parameters turned out to be not very sensitive to switching from the CE criterion to SE. It took 15 epochs to train both the CE and the SE systems from scratch.

Table 2: *Effect of initialization by CE pre-trained neural network. Frame and word error rates for the baseline GMM systems and hybrid systems.*

Task	System	dev		eval	
		FER	WER	FER	WER
ASR	Baseline GMM	-	24.8	-	31.7
	MLP CE	70.5	24.9	71.6	30.9
	MLP SE	70.3	24.5	71.5	30.5
HWR	Baseline GMM	-	17.4	-	22.3
	BRNN CE	13.2	17.7	15.2	22.7
	BRNN SE	12.2	16.8	13.5	21.9

For the HWR task we trained two LSTM-BRNNs with randomly initialized weights. The results are shown in Table 1. Similar to the ASR task, the CE system outperforms the SE system. In our experiments learning rates of 0.001 for CE and 0.01 for SE gave best results. The CE system reached its minimal cross-entropy score after 5 epochs, while the SE system required 25 epochs to converge.

As stated in Section 3, the inferior performance of the SE training with random initialization can be connected to a gradient vanishing already in the output layer. Figure 2 shows the output error signal Δ_c evaluated on a subset of training data. The plot shows that with SE, in contrast to CE, many misclassified training examples result in a tiny gradient, which can be explained by the existence of the plateau resulting from the bounds derived in Eq. 7 and 9. This representation clearly shows that, other than for CE, the gradient can vanish for the SE criterion with the softmax activation. During the training, this can lead to a slowdown or even full stagnation. This analysis agrees with the observations reported in [6] and [16].

4.2.2. Training with a good initialization

We made sure that the CE ANN fully converged by restarting the training, but the performance did not improve anymore. Finally we used that model to initialize a new SE training. It converged already after 3 more epochs and the recognition results in Table 2 show that this "fine-tuning" step improved all error measures consistently.

For the HWR task we followed the same strategy: we made sure that the CE ANN reached its maximum performance by training an additional epoch. While this epoch decreased the FER from 13.27% to 13.19% it led to an increase in the CE score which in turn decreased the performance of the hybrid HMM system. Finally we trained an SE system initalized with the CE system. As in the ASR task, this "fine-tuning" step improved the overall system and outperformed the GMM baseline.

In order to understand the effect the SE training had on the initalization by CE, we look at the distribution of the network outputs for the correct class as well as the distribution of the SE values within the boundaries derived in Section 2. Figure 3 shows that the SE training slightly shifted the probability mass towards the high values of $q(c_n|x_n)$. Figure 4 shows the histogram of the directions on the error plane in which every vector has been shifted during the SE training. The difference in the SE is inversely proportional to the difference in $\log q(c_n|x_n)$ before and after "fine-tuning" (denoted as $q^{(0)}$ and $q^{(1)}$). While the error criterion increased for 39.2% of the vectors (top-left quadrant), it decreased for the other 47.4% (bottom-right quadrant). The overall squared error was reduced by 1.2%.

5. Conclusions

In this paper we presented an investigation on the properties of the cross-entropy and the squared error criteria for training of ANNs. A theoretical analysis of the error bounds was supported

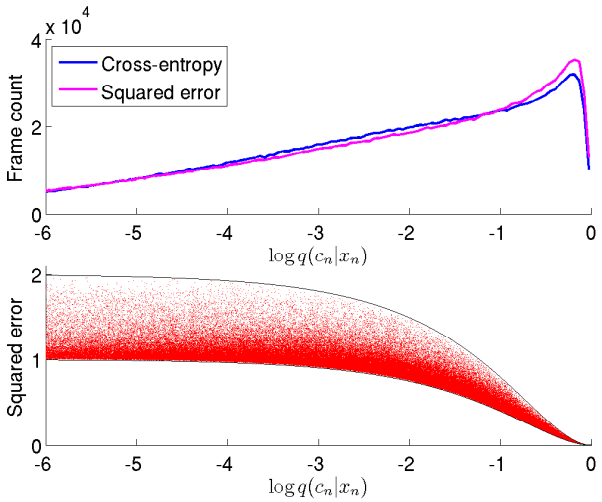

Figure 3: *The upper plot shows the histogram of* $\log q(c_n|x_n)$ *on a development set. The lower plot depicts how the corresponding error values are distributed within the boundaries derived in Eq. 7 and 9. The values are obtained from the best ASR systems (cf. Table 2).*

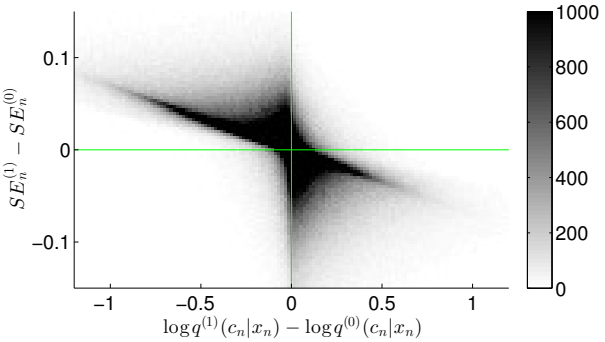

Figure 4: *Histogram of shifts the points in Fig. 3 (bottom) experience after the "fine-tuning" training according to SE.*

by experimental evaluation with well trained ANNs.

The experimental results have shown that, in a comparable environment and with randomly initialized weights, the CE criterion allows to find a better local optimum than the SE criterion. The training of the SE system quickly got stuck in a worse local optimum where the gradient vanished and no further reduction of the classification errors was possible. We presented an analysis of the gradients that explains this convergence issue.

However, starting with a good initialization, the SE criterion could consistently improve the solution found by the CE system from 30.9% to 30.5% (ASR) and from 22.7% to 21.9% (HWR) measured in WER in a HMM-ANN hybrid system.

Acknowledgements

The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 287755 (transLectures). H. Ney was partially supported by a senior chair award from DIGITEO, a French research cluster in Île-de-France. Supported by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Defense U.S. Army Research Laboratory (DoD / ARL) contract no. W911NF-12-C-0012. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DoD/ARL, or the U.S. Government.

6. References

[1] H. A. Bourlard and N. Morgan, *Connectionist speech recognition: a hybrid approach*. Norwell, MA, USA: Kluwer Academic Publishers, 1993.

[2] C. M. Bishop, *Pattern recognition and machine learning*, 1st ed. New York, NY, USA: Springer, Oct. 2006.

[3] R. A. Dunne and N. A. Campbell, "On the pairing of the softmax activation and cross-entropy penalty functions and the derivation of the softmax activation function," in *8th Australian Conference on Neural Networks*, Melbourne, Australia, 1997, pp. 181–185.

[4] D. M. Kline and V. L. Berardi, "Revisiting squared-error and cross-entropy functions for training neural network classifiers," *Neural Computing and Applications*, vol. 14, no. 4, pp. 310–318, Dec. 2005.

[5] H. Ney, "On the relationship between classification error bounds and training criteria in statistical pattern recognition," in *Iberian Conference on Pattern Recognition and Image Analysis*, Puerto de Andratx, Spain, Jun. 2003, pp. 636–645.

[6] P. Zhou and J. Austin, "Learning criteria for training neural network classifiers," *Neural Computing and Applications*, vol. 7, no. 4, pp. 334–342, 1989.

[7] H. Hermansky, D. Ellis, and S. Sharma, "Tandem connectionist feature extraction for conventional HMM systems," in *Proc. of IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, vol. 3, 2000, pp. 1635–1638.

[8] Quaero Programme. http://www.quaero.org.

[9] U.-V. Marti and H. Bunke, "The IAM-database: an English sentence database for offline handwriting recognition," *International Journal on Document Analysis and Recognition*, vol. 5, no. 1, pp. 39–46, Nov. 2002.

[10] S. Johansson, E. Atwell, R. Garside, and G. Leech, *The tagged LOB corpus: user's manual*, Norwegian Computing Centre for the Humanities, 1986.

[11] M. Kozielski, J. Forster, and H. Ney, "Moment-based image normalization for handwritten text recognition," in *Proc. of Int. Conf. on Frontiers in Handwriting Recognition*, Bari, Italy, Sep. 2012, pp. 256–261.

[12] W. Francis and H. Kucera, "Brown corpus manual, manual of information to accompany a standard corpus of present-day edited American English," Dept. of Linguistics, Brown Univ., Tech. Rep., 1979.

[13] L. Bauer, "Manual of information to accompany the Wellington corpus of written New Zealand English," Dept. of Linguistics, Victoria Univ., Tech. Rep., 1993.

[14] M. Liwicki, A. Graves, H. Bunke, and J. Schmidhuber, "A novel approach to on-line handwriting recognition based on bidirectional long short-term memory networks," in *Proc. of the 9th Int. Conf. on Document Analysis and Recognition (ICDAR)*, 2007.

[15] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE Transactions on Neural Networks*, vol. 5, no. 2, pp. 157–166, Mar. 1994.

[16] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proc. of Int. Conf. on Artificial Intelligence and Statistics*, vol. 9, Chia Laguna Resort, Italy, 2010, pp. 249–256.