Neural Net Vector Quantizers for discrete HMM-Based On-Line Handwritten Whiteboard-Note Recognition

Joachim Schenk and Gerhard Rigoll
Institute for Human-Machine Communication
Technische Universität München
Theresienstraße 90, 80333 München
{schenk, rigoll}@mmk.ei.tum.de

Abstract

In this work we evaluate a recently published vector quantization scheme, which has been developed to handle binary features like the pressure feature occurring in on-line handwriting recognition using discrete Hidden-Markov-Models (HMMs) with two neural net based vector quantizers (VQs). One of these uses a "Winner-Take-All" (WTA) update rule and the other implements the "Neural Gas" (NG) approach. Both approaches are believed to be more efficient VQs than the standard k-means VQ used in our earlier publication. In an experimental section we prove that both the WTA and NG neural net VQ significantly (significance is measured by the one-sided t-test) outperform our previously used k-means VQ by $r_W = 0.9\,\%$ and $r_N = 0.8\,\%$, respectively, referring to word-level accuracy. In addition, no significant difference in recognition accuracy between the WTA-VQ and the NG-VQ could be observed.

1. Introduction

Adopted from automatic speech recognition (ASR) Hidden-Markov-Models (HMMs, [1]) are becoming quite popular for on-line handwriting recognition, [2]. More recently, HMMs have also been introduced for on-line handwritten whiteboard note recognition, [3]. One distinguishes between continuous and discrete HMMs. In case of continuous HMMs, the observation probability is modeled by mixtures of Gaussians [1], whereas in the discrete case the probability computation is a simple table look-up. A vector quantizer (VQ) is used to map the continuous data to discrete symbols. While in ASR continuous HMMs are widely accepted, it remains unclear whether discrete or continuous HMMs should be used in on-line handwritingand whiteboard note recognition.

In our previous work (see [4]), the use of discrete HMMs in on-line handwriting recognition is further investigated using a standard k-means algorithm (see [5]) as VQ. Thereby it has been observed that the pressure information of the trajectory gets lost due to quantization error. To overcome this effect, in [4] we present a novel VQ scheme using switching codebooks, wherein the pressure information is kept without any loss and the statistical dependencies between the pressure and the remaining features are modeled. However one might argue that there are more sophisticated VQs than the k-means based. In this paper we therefore evaluate and confirm the findings of [4] by using two neural net based VQs, implementing the "Winner-Take-All" (WTA) update rule and the "Neural Gas" (NG) approach, respectively. One major outcome is that while both neural net VQs perform better than the k-means VQ, their performance can still be improved by the codebook switching approach.

The next section gives a short overview on the recognition system used. Section 3 summarizes vector quantization in general and the WTA and NG approaches in particular. Then, codebook switching as presented in [4] is reviewed. In an experimental section, both neural net VQs compete against each other and against a standard k-means VQ. Finally we give a conclusion and discussion of future work.

2. System Overview

This section briefly summarizes the recognition system used for the final experiments. The handwritten whiteboard data is recorded using the EBEAM-System as described in [3] and resampled space-equidistantly. Then, a histogram based skew- and slant-correction is performed according to [6] and the script is size-normalized. After preprocessing, 24 features are extracted from the recorded data and form the feature vector $\mathbf{f}(t) = (f_1(t), \ldots, f_{24}(t))$, [3; 7]. The extracted on-line features are: the binary "pen-pressure" f_1, $f_1 = 1$ if the pen's tip is on the whiteboard and $f_1 = -1$ otherwise,

978-1-4244-2175-6/08/$25.00 ©2008 IEEE

an interpolated velocity equivalent (f_2) computed *before* resampling, the high pass filtered x- and y-coordinate ($f_{3,4}$), the "writing direction" ($f_{5,6}$), and the "curvature" ($f_{7,8}$). In addition, on-line features describing the relation between the sample point $\mathbf{s}(t)$ to its neighbors are used: a logarithmic transformation of the "vicinity aspect" (f_9). The "vicinity slope" ($f_{10,11}$), the "vicinity curliness" (f_{12}), and the average square distance to each point of the trajectory and the line $[\mathbf{s}_{t-\tau}, \mathbf{s}_t]$ (f_{13}). As off-line features we extracted a 3×3 "context map" (f_{14-22}) and the "ascenders" and "descenders" ($f_{23,24}$), i. e. the number of pixels above and beneath the current sample point. All features are mean and variance normalized.

Finally, the handwritten data is recognized by a *discrete* Hidden Markov Model based classifier. In order to map the continuous feature vectors to a discrete observation sequence, vector quantization is performed which is described in the next section.

3. Vector Quantization

In this section we briefly explain vector quantizers (VQs), review the codebook switching as presented in [4], and describe the notations.

Vector quantization describes the joint mapping of a sequence of N-dimensional, continuous features $\mathbf{O} = (\mathbf{f}(1), \dots, \mathbf{f}(T))$, $\mathbf{f}(t) \in \mathbb{R}^N$ to a discrete, one dimensional sequence of codebook indices $\hat{\mathbf{o}} = (\hat{f}(1), \dots, \hat{f}(T))$, $\hat{f}(t) \in \mathbb{N}$ provided by a codebook $\mathbf{C} = [\mathbf{c}_1, \dots, \mathbf{c}_{N_{\text{cdb}}}]$, $\mathbf{c}_k \in \mathbb{R}^N$ containing $|\mathbf{C}| = N_{\text{cdb}}$ centroids $\mathbf{c}_i$, [5]. Once a codebook $\mathbf{C}$ is generated, the assignment of the continuous sequence to the codebook entries is a minimum distance search

$$\hat{f}(t) = \underset{1 \le k \le N_{\text{Cdb}}}{\operatorname{argmin}} \; d(\mathbf{f}(t), \mathbf{c}_k), \tag{1}$$

where $d(\mathbf{f}(t), \mathbf{c}_k)$ is e. g. the Euclidean distance. The quality of the VQ quantizing the continuous sequence $\mathbf{O}$ is measured by the signal to quantization noise ratio (SNR, see [5])

$$\text{SNR} = 10 \cdot \log_{10} \frac{{}^1\!/_T \sum_{t=1}^{T} ||\mathbf{f}(t)||_2^2}{{}^1\!/_T \sum_{t=1}^{T} ||\mathbf{f}_t - \mathbf{c}_{\hat{f}_t}||_2^2}. \tag{2}$$

In this paper we investigate two approaches for vector quantization based on neural nets (see e. g. [8]), realizing "competitive learning." From a neural net's point of view each prototype of a codebook $\mathbf{c}_i \in \mathbb{R}^N$ is represented by a neuron i which is associated with a weight vector $\mathbf{w}_i \in \mathbb{R}^N$. All weight vectors $\mathbf{w}_i$ are summarized in $\mathbf{W} = [\mathbf{w}_1, \dots, \mathbf{w}_{N_{\text{Cdb}}}]$. This is illustrated in Fig. 1. The mapping between the continuous input vectors and the corresponding codebook indices (neurons) is performed according to Eq. 1, replacing $\mathbf{c}_k$ by $\mathbf{w}_i$, [9]. Different codebooks are generated by various training approaches of the underlying neural net in a "competitive learning" manner. All competitive learning approaches have the

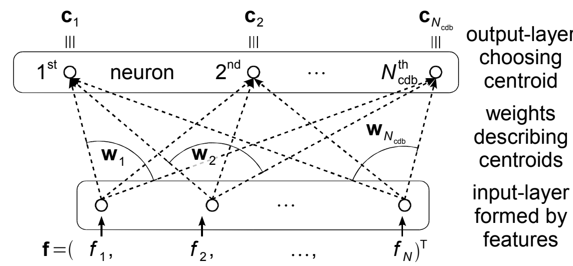

Figure 1: Neural net interpretation of vector quantization.

first stage in common, in which each neuron's extent of weight-adaptation to a new stimulus $\mathbf{f}(t)$ is determined, [10]. The approaches differ in the learning stage. Two common approaches, both realizing the above mentioned "competitive learning," are the "Winner-Take-All" (see [11]) and the "neural gas" (see [9]) strategies. They are described in the next sections.

3.1. Winner-Take-All (WTA)

The "Winner-Take-All" strategy [11] updates the neuron $\hat{i}$ which best fits the input stimulus $\mathbf{f}(t)$, i. e the prototype $\mathbf{c}_{\hat{i}}(t-1)$ (the weight vector $\vec{w}_{\hat{i}}(t-1)$) which lies next to the continuous vector $\mathbf{f}(t)$:

$$\begin{aligned} \mathbf{w}_{\hat{i}}(t) &= \mathbf{w}_{\hat{i}}(t-1) + \epsilon(t) \cdot [\mathbf{f}(t-1) - \mathbf{w}_{\hat{i}}(t-1)] \\ \mathbf{w}_i(t) &= \mathbf{w}_i(t-1), 1 \le i \le N_{\text{Cdb}}, i \ne \hat{i}, \end{aligned} \tag{3}$$

with t, the time variable, indicating the time dependency of the weights during training and $\epsilon(t)$ denoting a time-dependent step size, [10; 12]. The WTA-update rule as described in Eq. 3 with well-chosen step size $\epsilon(t)$ comprises the on-line version of the k-means algorithm, [5; 9].

3.2. Neural Gas (NG)

In contrast to the WTA, in the Neural Gas VQ the weights of the neurons are updated according to their "proximity" to the input vector $\mathbf{f}(t)$ [9]:

$$\begin{aligned} \mathbf{w}_i(t) = \mathbf{w}_i(t-1) + \\ + \epsilon(t) \cdot h_\lambda(\, k_i(\mathbf{W}, \mathbf{f}(t))\,) \cdot [\mathbf{f}(t-1) - \mathbf{w}_i(t-1)], \end{aligned} \tag{4}$$

whereby the proximity is expressed by

$$h_\lambda(\, k_i(\mathbf{W}, \mathbf{f}(t))\,) = \exp\left(-\frac{k_i(\mathbf{W}, \mathbf{f}(t))}{\lambda}\right), \tag{5}$$

with $k_i(\mathbf{W}_t, \mathbf{f}(t))$ the number of neurons j, with $||\mathbf{w}_j(t) - \mathbf{f}(t)||_2 \le ||\mathbf{w}_i - \mathbf{f}(t)||_2$. For $\lambda \to 0$ the update rule as presented in Eq. 4 becomes the WTA, and for $\lambda \ne 0$ not only the "winner" neuron's weights $\mathbf{w}_{\hat{i}}$ but also those of its neighbors are updated.

3.3. Codebook Switching

Standard k-means VQ cannot adequately model the pen-pressure information (see [4]), while it is shown in [13] that the pressure information is a vital feature in on-line whiteboard note recognition. Our results indicate (see Sec. 4) that the pressure information also gets lost when using neural net based VQs (which are known to reduce quantization noise, [9]). By switching between two neural nets during training and quantization depending on the pen's pressure, the statistical dependencies between the pressure and the remaining features are modeled while keeping the exact pressure information, [4]. The number of output neurons of the first neural net N_{s} (for feature vectors with $f_1 < 0$) and N_{g}, the number of neurons in the output layer of the second neural net (for feature vectors with $f_1 > 1$) can be chosen arbitrarily and form the ratio $R = \frac{N_{\text{g}}}{N_{\text{s}}}$.

4. Experiments and Results

We conduct experiments on the IAM-onDB-t1 benchmark of the IAM-OnDB, a database containing handwritten whiteboard notes, [14]. Comparability of the results is provided by using the same settings as in [4]. The training set of the IAM-onDB-t1 is used to train both the parameters of the discrete HMMs and the weights of the neural nets. The optimal number of output neurons (N_{cdb}), the step-size $\epsilon(t)$ of Eqs. 3 and 4, and the neighbor factor λ are chosen by evaluating the character-level accuracy on both IAM-onDB-t1's validation sets. With these parameters the test set is recognized on the word-level. Significance of the results is proven by the one-sided t-test, giving the probability p_N of rejecting the hypothesis "both approaches perform equally."

In the first experiment (*Exp. 1*) we use the WTA-VQ and the NG-VQ as described in Sec. 3.1 and Sec. 3.2 to quantize the whole feature vector $\mathbf{f}$. The results are shown in Fig. 2: the highest character accuracies are $a_{\text{b,W}} = 63.3\,\%$ for the WTA-VQ ($N_{\text{ceb}} = 5000$) and $a_{\text{b,N}} = 63.1\,\%$ for the NG-VQ ($N_{\text{cdb}} = 7500$). Compared to the results presented in [4] and also listed in Tab. 1. This is a relative improvement of $r_{\text{b,W}} = 1.1\,\%$ and $r_{\text{b,N}} = 0.8\,\%$.

In the second experiment (*Exp. 2*) the pressure information f_1 is left out for both the WTA-VQ and the NG-VQ. As can be seen from Fig. 2, performance only drops slightly, although the pressure information is considered a crucial feature in [13]: for $N_{\text{cdb}} = 7500$ the WTA-VQ achieves a character accuracy of $a_{\text{r,W}} = 63.0\,\%$ (a relative change of $r_{\text{r,W}} = -0.5\,\%$ compared to the baseline drawn in *Exp. 1*), and $a_{\text{r,N}} = 63.0\,\%$ for the NG-VQ approach (a relative drop of $r_{\text{r,N}} = -0.2\,\%$).

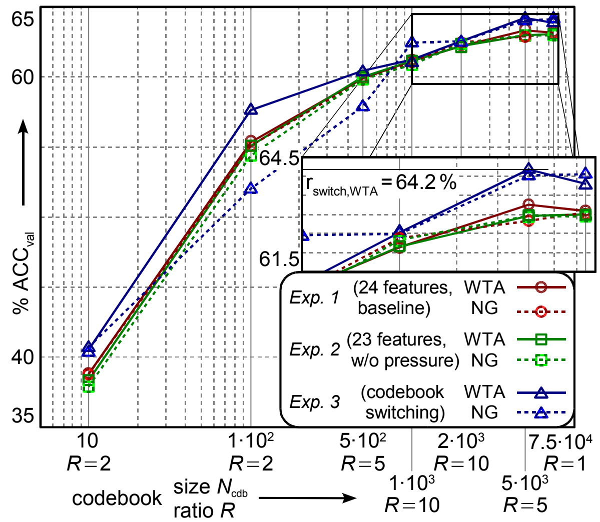

Figure 2: Evaluation of different systems' character accuracies with respect to the codebook size N_{cdb} and, in case of codebook-switching, the used ratio $R = \frac{N_{\text{g}}}{N_{\text{s}}}$.

	k-means [4]		WTA		NG	
	char.	word	char.	word	char	word
Exp. 1	62.6 %	63.5 %	63.3 %	65.1 %	63.1 %	64.9 %
Exp. 2	62.5 %	63.3 %	63.0 %	64.8 %	63.0 %	64.7 %
Exp. 3	63.7 %	65.5 %	64.2 %	66.2 %	64.1 %	66.1 %

Table 1: Results of the experiments.

In the last experiment (*Exp. 3*), the pressure information is modeled separately (though statistically dependently on the remaining features) by a switching codebook (see Sec. 3.3). Thereby, the ratio $R = \frac{N_{\text{g}}}{N_{\text{s}}}$ is chosen according to the best performing values found in [4] and is also given in Fig. 2, as well as the character accuracies: for the WTA-VQ we achieve $a_{\text{sw,W}} = 64.2\,\%$ ($r_{\text{sw,W}} = 1.4\,\%$ relative improvement, $N_{\text{cdb}} = 5000$), and $a_{\text{sw,N}} = 64.1\,\%$ ($r_{\text{sw,N}} = 1.6\,\%$ relative improvement, $N_{\text{cdb}} = 7500$) for the NG-VQ.

For the final task, we use the best-performing configurations from the above experiments and conduct a word-level evaluation on the IAM-onDB-t1's test set. The results of our experiments are shown in Tab. 1. For the WTA-VQ a word-level accuracy of $A_{\text{b,W}} = 65.1\,\%$, and $A_{\text{b,N}} = 64.9\,\%$ for the NG-VQ is achieved. The slight drop in character accuracy when leaving out the pressure information also decreases the word-level accuracy (however, for the NG-VQ the relative drop of $r_{\text{r,N}} = -0.3\,\%$, is *not* significant as $p_N = 93.8\,\%$). Finally, a significant improvement for both the WTA-VQ and the NG-VQ can be reported when the switching codebook approach is used: $A_{\text{sw,W}} = 66.2\,\%$ (a relative improvement of $r_{\text{sw,W}} = 1.7\,\%$, $p_N \geq 99.9\,\%$) and $A_{\text{sw,N}} = 66.1\,\%$ (a relative improvement of $r_{\text{sw,N}} = 1.8\,\%$, $p_N \geq 99.9\,\%$).

Compared to our result for the k-means VQ (see Tab. 1), a significant relative improvement of $r_{\mathrm{W}} = 0.9\,\%$ and $r_{\mathrm{N}} = 0.8\,\%$ for both neural net VQs can be observed ($p_N \geq 99.9\,\%$). When comparing the best results of the WTA-VQ and the NG-VQ the WTA-VQ performs slightly better by $r = 0.2\,\%$ relatively, which is *not* significant ($p_N = 71.6\,\%$).

5. Conclusion

In this paper we, investigated two neural net-based vector quantizers (VQs), namely the "Winner-Take-All" (WTA)-VQ and the "Neural Gas" (NG)-VQ for quantizing the features used in on-line handwriting recognition with discrete HMMs. We successfully applied the novel VQ design presented in [4] to preserve the pen's pressure information that would otherwise be lost due to quantization error. Significance of the result has been proven by the one-sided t-test.

Our experiments delivered three major results: First, by using the codebook switching approach a significant relative improvement of $r_{\mathrm{sw,W}} = 1.7\,\%$ (WTA-VQ) and $r_{\mathrm{sw,N}} = 1.8\,\%$ (NG-VQ), both measured in word-level accuracy, compared to a baseline system where all features are quantized jointly, can be observed. The second observation is that, compared to standard k-means quantization, both neural VQs perform better: a peak word accuracy of $A_{\mathrm{sw,W}} = 66.2\,\%$ for the WTA-VQ and $A_{\mathrm{sw,N}} = 66.1\,\%$ for the NG-VQ is reached — a significant relative improvement of $r_{\mathrm{W}} = 0.9\,\%$ and $r_{\mathrm{N}} = 0.8\,\%$, respectively. Finally, the relative change of $r = 0.2\,\%$ between the WTA-VQ and NG-VQ was proven to be *not* significant. However the NG-VQ uses more output neurons compared to the WTA-VQ in order to achieve competitive results.

As shown in Fig. 2, no improvement can be observed if raising the number of output neurons from $N_{\mathrm{cdb}} = 2000$ to $N_{\mathrm{cdb}} = 5000$ in case of NG-VQ and codebook switching. This indicates that the ratio R has not been chosen properly. The computationally expensive optimization of the ratio R will be done in future work. The WTA-neural net VQ delivers a higher SNR (as defined in Eq. 2) than the k-means VQ for $N_{\mathrm{cdb}} = 10$ and $N_{\mathrm{cdb}} = 1000$. This is illustrated in Fig. 3 (left). However, the higher SNR does not translate to a higher recognition accuracy in all cases: while a relative improvement of $r = 0.5\,\%$ is achieved for $N_{\mathrm{cdb}} = 1000$ centroids, a relative drop of $r = -5.9\,\%$ can be observed for a codebook size of $N_{\mathrm{cdb}} = 10$. One explanation might be the change in the per feature SNR distribution illustrated in Fig. 3 (right) for selected features. In future work we will investigate the influence of the per feature SNR distribution on the recognition accuracy.

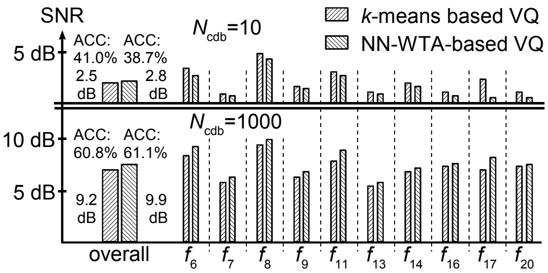

Figure 3: Overall (left) and per feature SNR (right).

Acknowledgments

The authors thank M. Liwicki for sharing the final benchmark's lattice and G. Weinberg for comments.

References

[1] L.R. Rabiner, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition," *Proc. of the IEEE*, vol. 77, no. 2, pp. 257–285, February 1989.

[2] R. Plamondon and S.N. Srihari, "On-Line and Off-Line Handwriting Recognition: A Comprehensive Survey," *IEEE Trans. on PAMI*, vol. 22, no. 1, pp. 63–84, 2000.

[3] M. Liwicki and H. Bunke, "HMM-Based On-Line Recognition of Handwritten Whiteboard Notes," *Proc. of the IWFHR*, pp. 595–599, 2006.

[4] J. Schenk, S. Schwärzler, G. Ruske, and G. Rigoll, "Novel VQ Designs for Discrete HMM On-Line Handwritten Whiteboard Note Recognition," *Proc. of the DAGM*, pp. 234–243, 2008.

[5] R.M. Gray, "Vector Quantization," *IEEE ASSP Magazine*, pp. 4–29, April 1984.

[6] E. Kavallieratou, N. Fakotakis, and G. Kokkinakis, "New Algorithms for Skewing Correction and Slant Removal on Word-Level," *Proc. of the ICECS*, vol. 2, pp. 1159–1162, 1999.

[7] S. Jaeger, S. Manke, J. Reichert, and A. Waibel, "The NPen++ Recognizer," *Int. J. on Document Analysis and Recognition*, vol. 3, pp. 169–180, 2001.

[8] S. Gossberg, "Neural-Gas Network for Vector Quantization and its Application to Time-Series Prediction," *IEEE Trans. on NN*, vol. 4, no. 4, pp. 558–569, 1993.

[9] T.M. Martinetz, S.G. Berkovich, and K.J Schulten, "Neural-Gas Network for Vector Quantization and its Aplication to time-Series Prediction," *IEEE Trans. on NN*, vol. 4, no. 4, pp. 558–569, 1993.

[10] T. Hoffmann and J.M.Buhmann, "Competitive Learning Algorithms for Robust Vector Quantization," *IEEE Trans. on SP*, vol. 46, no. 6, pp. 1665–1675, 1998.

[11] J. Hertz, A. Krogh, and R.G. Palmer, *Introduction to the Theory of Neural Computation*, Addison Wesley, 1991.

[12] T. Kohonen, "The Self Organizing Map," *Proc. of the IEEE*, vol. 78, no. 9, pp. 1464–1480, 1990.

[13] M. Liwicki and H. Bunke, "Feature Selection for On-Line Handwriting Recognition of Whiteboard Notes," *Proc. of the Conf. of the IGS*, pp. 101–105, 2007.

[14] M. Liwicki and H. Bunke, "IAM-OnDB - an On-Line English Sentence Database Acquired from Handwritten Text on a Whiteboard," *Proc. of the ICDAR*, vol. 2, pp. 1159–1162, 2005.