INTEGRATION OF ORTHOGONAL FEATURE DETECTORS IN PARAMETER LEARNING OF ARTIFICIAL NEURAL NETWORKS TO IMPROVE ROBUSTNESS AND THE EVALUATION ON HAND-WRITTEN DIGIT RECOGNITION TASKS

Chia-Ping Chen, Po-Yuan Shih

Department of Computer Science and Engineering
National Sun Yat-sen University
Kaohsiung, Taiwan
{cpchen@cse,m033040066@student}.nsysu.edu.tw

Wei-Bin Liang

Hon Hai Technology Group
38, Hsin-Kuang Road
Kaohsiung, Taiwan
liangnet@gmail.com

ABSTRACT

We propose to use orthogonal feature detectors in artificial neural networks for the robustness of performance under noisy conditions. The motivation is grounded on the principle that orthogonal decomposition is the most efficient among all representation of a signal. In this paper, we incorporate orthogonalization in the process of learning the network weights. In our implementation, the constraint of orthogonality is enforced by applying Gram-Schmidt processes to the feature detectors during network training. The proposed method is evaluated on MNIST database for hand-written digit recognition. The images in the training set are not corrupted, while the images in the test set are artificially corrupted with white noises. Experimental results show that the proposed orthogonalization method achieves 56.4% relative improvement in recognition error rate over a conventional learning method without orthogonalization. Given that the clean training data and the noisy test data are clearly mismatched, such an improvement with artificial neural networks is indeed very remarkable. For engineering insight, we devise a visualization tool which illuminates interesting features of the neurons learned by the proposed method.

Index Terms— orthogonality, noise-robustness, feature detector, neural network

1. INTRODUCTION

Artificial neural networks (ANNs) are machine-learning models based on the implications of biological neural systems [1]. In ANN, neurons are interconnected and they exchange messages via weighted links. The learning ability of ANN stems from the fact that the weights in the connective neurons can be modified to maximize the likelihood or minimize the error of a training data set. With the ability of automatically adapt to real data, neural networks have been used to solve a wide variety of tasks which are hard to solve using traditional methods [2].

A deep neural network (DNN) is a feed-forward neural network which has more than one hidden layers between its input and output layers [3]. With sufficient training data and appropriate training strategies [4, 5], DNN performs very well in certain difficult machine-learning tasks [6]. In a wide range of research domains, e.g. speech recognition, visual object recognition and text processing, the state-of-the-art performance has been achieved or even beaten by DNN [7, 8, 9]. Given the huge success of deep learning in so many applications, one critical issue remains to be addressed, i.e., *data mismatch*. It is well-known that data mismatch often leads to severe degradation for a classification/recognition system based on data-driven learning paradigm.

The issue of noise-robustness within the framework of deep learning has been addressed in the application of automatic speech recognition. Audio and visual features have been combined in deep learning [10]. Two masking functions have been estimated to separate speech from noises for improving DNN acoustic models [11]. A DNN-based system has been shown to reduce word error rate (WER) by up to one third over a discriminatively trained Gaussian mixture model-based (GMM-based) system on a challenging conversational speech transcription task [12].

To look into the issue of noise-robustness, we propose to apply orthogonality constraints to the feature detectors in feed-forward neural networks, and evaluate the proposed method with hand-written digit recognition in this work. Feed-forward neural networks have been applied to hand-written digit recognition [13]. With clean data, the performance is generally good [14]. However, when the data is corrupted, e.g. by stains or spots, the recognition accuracy degrades severely. In our survey, it has been reported that deep learners benefit more from out-of-distribution examples than a corresponding shallow learner, at least in the area of hand-written character recognition [15]. Furthermore, a supervised deep learning approach has been proposed to remove structural noise in the images of hand-written digits [16].

This remainder of this paper is organized as follows. The baseline neural network system for hand-written digit recognition is outlined, which is followed by the proposed orthogonality method for parameter estimation in Section 2. Evaluation schemes and results are presented and followed by comments on the results in Section 3. Finally, concluding remarks are given in Section 4, and we hint a few tentative directions for our future research works.

2. SYSTEM AND PROPOSED METHOD

2.1. Basic System

We begin with an neural-network system for hand-written digit recognition [17]. The overall system architecture is depicted in Figure 1. This simple system provides a very decent baseline with a low recognition error rate when the images of hand-written digits are not corrupted.

The open data set of MNIST [18] is used in the experiments. MNIST consists of images of scanned hand-written digits. Each image is represented by 28×28 pixels. Thus, there are $I = 784$ neurons in the input layer, each corresponding to a pixel. There are

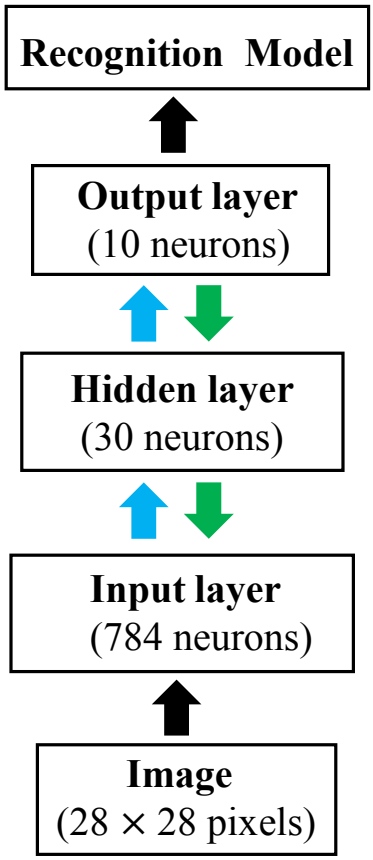


Fig. 1. Composition of a 3-layer neural network of hand-written digit recognition implemented for this work. The architecture is kept simple to illuminate the main point of using orthogonalization.

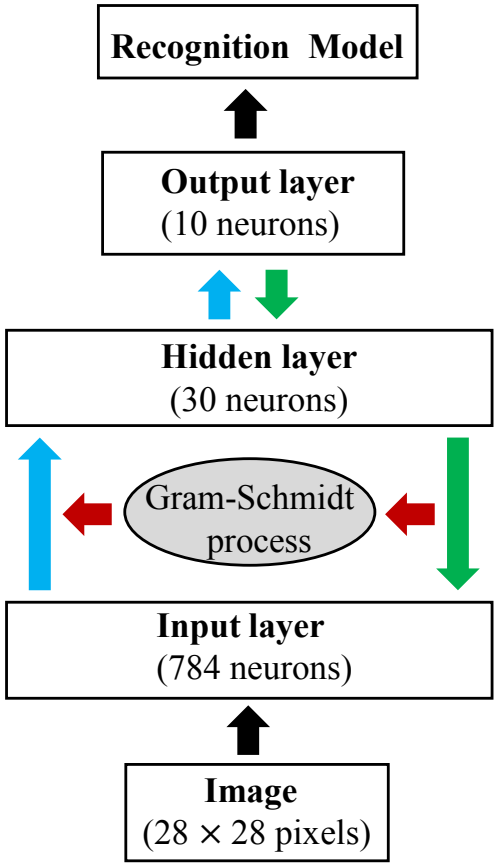


Fig. 2. Parameter learning incorporating the proposed orthogonalization method for noise-robustness. On top of the conventional parameter-learning algorithm of error back-propagation, we introduce an orthogonalization module based on the Gram-Schmidt process.

$K = 10$ sigmoid neurons in the output layer, each corresponding to a digit class. The hidden layer contains J neurons (we experiment with the cases of $J = 10, 15, 30$). Note that we also call the hidden-layer neurons the feature detectors, which are automatically learned from data via stochastic gradient descent with error back propagation [19].

2.2. Noise-robustness through Orthogonalization

Our goal is to achieve noise-robustness when the training data and test data are mismatched with simple and effective methods. In this work, we propose to use orthogonal feature vectors. Specifically, the weight vectors of the hidden-layer feature detectors are orthogonalized. Before orthogonalization, the set of weight vectors is a basis of a subspace in $\mathbb{R}^{28\times 28}$. Given these basis vectors, we apply the Gram-Schmidt process. After orthogonalization, the set of weight vectors is orthonormal.

In our implementation, the Gram-Schmidt process is applied within each epoch when the weights have just been updated (through error back propagation). The alternation of weight update and orthogonalization is repeated for a predetermined number of epochs. The block diagram of the proposed method is depicted in Figure 2.

For precise illustration, consider a neural network with $J = 30$ hidden-layer neurons. Let the weights of the links from the input-layer neurons to the hidden-layer neurons be denoted by a matrix

$$\mathbf{W}_{J\times I} = \{w_{ji}\}, \quad \text{where } j = 1, \ldots, J \text{ and } i = 1, \ldots, I.$$

The weight vectors are the vectors

$$\mathbf{w}_j = [w_{j1}, \ldots, w_{jI}]^T, \quad j = 1, \ldots, J.$$

At the end of stochastic gradient descent of epoch e, let the just-updated weight vectors be denoted by

$$\mathbf{w}_j^{(e)}, \quad j = 1, \ldots, J.$$

The Gram-Schmidt process is applied to $\{\mathbf{w}_1^{(e)}, \ldots, \mathbf{w}_J^{(e)}\}$, with

$$\mathbf{w}'^{(e)}_j = \mathcal{N}\left(\mathbf{w}_j^{(e)} - \sum_{i=1}^{j-1} \left\langle \mathbf{w}_j^{(e)}, \mathbf{w}'^{(e)}_i \right\rangle \mathbf{w}'^{(e)}_i\right), \quad j = 1, \ldots, J,$$

where $< \mathbf{u}, \mathbf{w} >$ is the inner-product of vectors $\mathbf{u}$ and $\mathbf{w}$ and $\mathcal{N}(\mathbf{v})$ normalizes a vector $\mathbf{v}$ to be of unit length, to obtain an orthonormal basis

$$\mathbf{w}'^{(e)}_j, \quad j = 1, \ldots, J.$$

This completes an epoch. The updated weights $\mathbf{w}'^{(e)}_1, \ldots, \mathbf{w}'^{(e)}_J$ are used to run the stochastic gradient descent of the next epoch, i.e., epoch $e + 1$.

3. EXPERIMENTS

In hand-written digit recognition, there may be corruptions such as pen stains or dirty spots to the data. We simulate data corruption by artificially adding noises to the images of the test data set. In other words, we create a situation of data mismatch, which makes the recognition tasks more challenging but also more interesting.

3.1. Data

The MNIST hand-written digit database [18] has been used throughout this study. The database has a training set of 60,000 examples, and a test set of 10,000 examples. The examples are scanned images of hand-written digits from 250 people, half of whom are US Census Bureau employees, and half of whom are high school students. All the digits have been size-normalized and centered in a fixed-size image with 28×28 pixels. The value at each pixel is the greyscale, normalized to the range from 0.0 (white) to 1.0 (black). A total number of 784 pixel values are fed to the neural network input-layer neurons.

White noises are added only to the test images, while the training data set remains clean. Specifically, we add white noises to each image in the test data set with different signal-to-noise ratio (SNR) levels of 0, 5, 10, 15, and 20 dBs. For example, Figure 3 shows the images of a digit corrupted by different levels of noise.

3.2. Stochastic Gradient Descent Parameters

The implementation is based on Numpy [20], a Python library, for doing fast linear algebra. The biases and weights in the network are all initialized randomly, and stored as lists of Numpy matrices. The number of epochs is set to 30 and the learning rate η in

$$\mathbf{w}^{(\text{new})} \leftarrow \mathbf{w}^{(\text{old})} - \eta \, \nabla_{\mathbf{w}} E(\mathbf{w}) \Big|_{\mathbf{w}^{(\text{old})}}$$

is set to 3.0.[1] In each epoch, the training data are shuffled and partitioned into mini-batches of 10 samples each. With each mini-batch, the learning process invokes the back propagation algorithm to update the parameters.

3.3. Evaluation Results

We report the results of three configurations of experiments. The first configuration, denoted by **C**, is the case where clean test images are used. The second configuration, denoted by **N**, is the case where noise-corrupted test images are used. In both **C** and **N**, the proposed orthogonalization is not applied. The third configuration, denoted by **O**, is the case where noise-corrupted test images are used, and the proposed orthogonalization method is applied in the process of learning network parameters.

The results of configuration **C** is shown in Table 1. Here the number of neurons in the hidden layer is varied. From this table, we can see that increasing the number of neurons in the hidden layer improves the performance, upto 30 neurons. In addition, we can see that the trained neural network achieves 95.1% accuracy when the test images are not corrupted by noises.

The results of configuration **N** is shown in Table 2. In this case, the test images are corrupted by noises with various levels of SNR. The hidden layer contains 30 neurons. From this table, we can see that when the test images are corrupted by noises, the performance of the neural network significantly degrades. In the case of 0 dB, the error rates increase by more than 3 times. Very clearly, the neural network trained by the baseline method is not robust to noise.

[1] Here $E(\mathbf{w})$ is an error function, and the sum of squared error is used in this work.

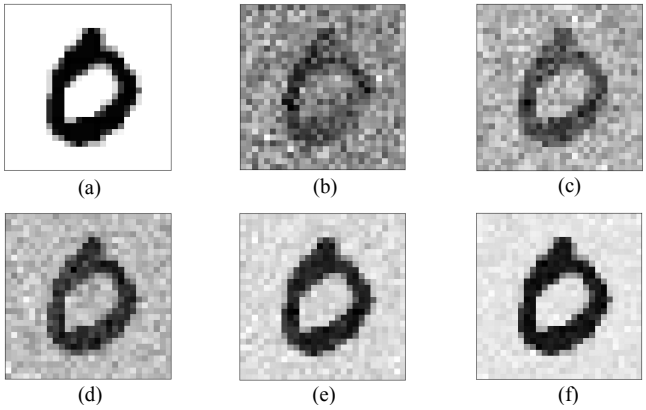


Fig. 3. The images of a digit corrupted by white noises. (a) the original image, (b) – (f) images of the same digit corrupted by noises with 0, 5, 10, 15, 20 dB SNRs.

Table 1. Results of a neural-network system trained with a conventional method. Both the training data and the test data are clean. This is also referred to as configuration **C**. Based on these results, subsequent experiments (with noisy test images) use 30 hidden-layer neurons.

# neurons in the hidden layer	recognition accuracy
10	91.1
15	93.3
30	95.1

The results of configuration **O** is shown in Table 3. In this case, the neural network is trained by the proposed orthogonalization method. Again, the test images are corrupted by noises with various levels of SNR, and the hidden layer contains 30 neurons. From this table, we can see that a neural-network system trained with the proposed orthogonalization method is very robust to noise. The degradation of performance with noise is very graceful. Compared to configuration **N**, one can see that the noisier the test images, the more significant the relative improvement. Take the 0 dB case for example, the relative improvement over a system trained without the proposed orthogonalization method is 56.4%. These results are truly remarkable.

3.4. Visualization of Learning

A neuron in the hidden layer, say neuron j, can be visualized by an image which is as large as the input image. In such an image, the grey-scale value at pixel i corresponds to the weights w_{ij} of the link from the input neuron i to the hidden-layer neuron j. We call such an image a *weight image*.

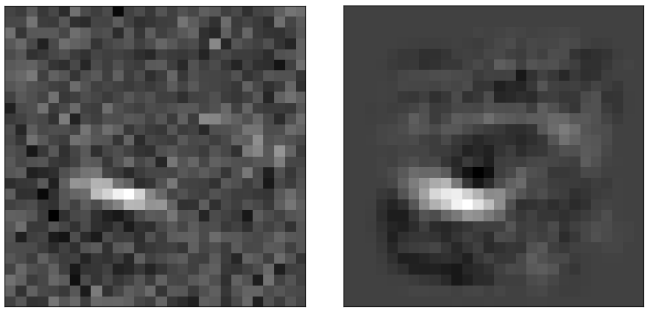
In Figure 4, we show the weight images of one neuron with and without incorporating the proposed orthogonalization process during parameter learning. The proposed orthogonalization method brings a significant difference to the learned feature detectors (neurons). Without orthogonalization, the learned weights are distributed over

Table 2. Results of a neural-network system trained with a conventional method (without orthogonalization), and tested with corrupted images, i.e., configuration **N**. The rightmost column summarizes the performance of the system using noisy test images relative to using clean test images, showing very severe degradation as the noise level increases.

SNR in dB	recognition accuracy	rel. imp. over ∞
∞ (clean)	95.1	=
20	94.9	-4.1
15	94.5	-12.2
10	93.6	-30.1
5	91.4	-75.5
0	81.9	-269.4

Table 3. Results of a neural-network system trained with the proposed orthogonalization method incorporated, and tested with corrupted images, i.e., configuration **O**. The rightmost column summarizes the relative improvement in recognition error rates over the baseline without orthogonalization, i.e., configuration **N**. One can see very significant improvements. Note the degradation of **O** relative to **C** is much more graceful than **N**.

SNR in dB	recognition accuracy	rel. imp. over **N**
20	94.3	-12.2
15	95.0	9.1
10	95.3	26.6
5	94.6	38.1
0	92.1	56.4



without orthogonalization with orthogonalization

Fig. 4. Visualization of feature detectors. Here we show the **weight images** of one neuron in the hidden layer. Left: the weight image of a neuron without incorporating orthogonalization (baseline). Right: the weight image of the same neuron with orthogonalization incorporated (proposed).

the entire image, leading to a mosaic texture. With the proposed orthogonalization, the learned weights are more focused in the central area of the image, where the critical information for digit recognition is located. As a result, the features are more salient when they are required to be orthogonalized in each epoch. Put another way, the weights in the peripheral area are subdued, and thus a neuron is less likely to be activated by spurious correlation when the input is noisy.

4. CONCLUSION

In this paper, we propose a neural-network learning method which incorporates orthogonalization to achieve noise-robustness. Evaluated on the MNIST database for hand-written digit recognition, the proposed method achieves 56.4% relative improvement over a baseline neural-network learning method without orthogonalization. In addition, we have devised a visualization method which provides insights into the nature of the learning process.

In the future, we will combine the basic idea of orthogonalization with other noise-robustness techniques to see if incremental improvements can be achieved. Furthermore, we will apply the proposed method to spoken-language-processing applications, such as speech emotion recognition and automatic speech recognition under noisy conditions.

5. ACKNOWLEDGEMENTS

We would like to thank those pioneers who are generous enough to share their ideas and tools, with special thanks to **Dr. Michael Nielsen** and to the **University of Montreal**. We would also like to thank **Dr. Nelson Morgan** of International Computer Science Institute and **Dr. Dan Ellis** of Columbia University for inspirational discussions on neural networks and deep learning during a visit to ICSI at Berkeley. Finally, we would like to thank the **Ministry of Science and Technology** of Taiwan for funding this research.

6. REFERENCES

[1] Geoffrey G Towell and Jude W Shavlik, "Knowledge-based artificial neural networks," *Artificial intelligence*, vol. 70, no. 1, pp. 119–165, 1994.

[2] Donald Michie, David J Spiegelhalter, and Charles C Taylor, "Machine learning, neural and statistical classification," *Ellis Horwood*, 1994.

[3] Li Deng and Dong Yu, "Deep learning: methods and applications," *Foundations and Trends in Signal Processing*, vol. 7, no. 3–4, pp. 197–387, 2014.

[4] Geoffrey E Hinton, Simon Osindero, and Yee-Whye Teh, "A fast learning algorithm for deep belief nets," *Neural computation*, vol. 18, no. 7, pp. 1527–1554, 2006.

[5] Yoshua Bengio, Aaron Courville, and Pierre Vincent, "Representation learning: A review and new perspectives," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 35, no. 8, pp. 1798–1828, 2013.

[6] Bo Li and Khe Chai Sim, "A spectral masking approach to noise-robust speech recognition using deep neural networks," *Audio, Speech, and Language Processing, IEEE/ACM Transactions on*, vol. 22, no. 8, pp. 1296–1305, 2014.

[7] Li Deng, Geoffrey Hinton, and Brian Kingsbury, "New types of deep neural network learning for speech recognition and related applications: An overview," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 8599–8603.

[8] Ronan Collobert and Jason Weston, "A unified architecture for natural language processing: Deep neural networks with multitask learning," in *Proceedings of the 25th international conference on Machine learning*. ACM, 2008, pp. 160–167.

[9] Dan Ciresan, Ueli Meier, and Jürgen Schmidhuber, "Multi-column deep neural networks for image classification," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012, pp. 3642–3649.

[10] Jing Huang and Brian Kingsbury, "Audio-visual deep learning for noise robust speech recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 7596–7599.

[11] Bo Li and Khe Chai Sim, "Improving robustness of deep neural networks via spectral masking for automatic speech recognition," in *Automatic Speech Recognition and Understanding (ASRU), 2013 IEEE Workshop on*. IEEE, 2013, pp. 279–284.

[12] Frank Seide, Gang Li, and Dong Yu, "Conversational speech transcription using context-dependent deep neural networks.," in *Interspeech*, 2011, pp. 437–440.

[13] Janusz Starzyk, Nasser Ansari, et al., "Feedforward neural network for handwritten character recognition," in *Circuits and Systems, 1992. ISCAS'92. Proceedings., 1992 IEEE International Symposium on*. IEEE, 1992, vol. 6, pp. 2884–2887.

[14] Li Wan, Matthew Zeiler, Sixin Zhang, Yann L Cun, and Rob Fergus, "Regularization of neural networks using dropconnect," in *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, 2013, pp. 1058–1066.

[15] Frédéric Bastien, Yoshua Bengio, Arnaud Bergeron, Nicolas Boulanger-Lewandowski, Thomas Breuel, Youssouf Chherawala, Moustapha Cisse, Myriam Côté, Dumitru Erhan, Jeremy Eustache, et al., "Deep self-taught learning for handwritten character recognition," *arXiv preprint arXiv:1009.3589*, 2010.

[16] Gang Chen and Sargur N Srihari, "Removing structural noise in handwriting images using deep learning," in *Proceedings of the 2014 Indian Conference on Computer Vision Graphics and Image Processing*. ACM, 2014, p. 28.

[17] Michael A. Nielsen, "Neural networks and deep learning," 2015, Available at: `http://neuralnetworksanddeeplearning.com/`.

[18] Yann LeCun, Corinna Cortes, and Christopher JC Burges, "The mnist database of handwritten digits," 1998.

[19] B Boser Le Cun, John S Denker, D Henderson, Richard E Howard, W Hubbard, and Lawrence D Jackel, "Handwritten digit recognition with a back-propagation network," in *Advances in neural information processing systems*. Citeseer, 1990.

[20] NumPy Developers, "Numpy," *NumPy Numpy. Scipy Developers*, 2013.