Noname manuscript No.
(will be inserted by the editor)

Competitive Cross-Entropy Loss: A Study on Training Single-Layer Neural Networks for Solving Nonlinearly Separable Classification Problems

Kamaledin Ghiasi-Shirazi

This is a post-peer-review, pre-copyedit version of an article published in Neural Processing Letters. The final authenticated version is available online at: `http://dx.doi.org/10.1007/s11063-018-9906-5`.

Abstract After Minsky and Papert (1969) showed the inability of perceptrons in solving nonlinearly separable problems, for several decades people misinterpreted it as an inherent weakness that is common to all single-layer neural networks. The introduction of the backpropagation algorithm reinforced this misinterpretation as its success in solving nonlinearly separable problems passed through the training of multilayer neural networks. Recently, Conaway and Kurtz (2017) proposed a single-layer network in which the number of output units for each class is the same as input units and showed that it could solve some nonlinearly separable problems. They used the MSE (Mean Square Error) between the input units and the output units of the actual class as the objective function for training the network. They showed that their method could solve the XOR and M&S'81 problems, but it could not do any better than random guessing on the 3-bit parity problem. In this paper, we use a soft competitive approach to generalize the CE (Cross-Entropy) loss, which is a widely accepted criterion for multiclass classification, to networks that have several output units for each class, calling the resulting measure the CCE (Competitive Cross-Entropy) loss. In contrast to Conaway and Kurtz (2017), in our method, the number of output units for each class can be chosen arbitrarily. We show that the proposed method can successfully solve the 3-bit parity problem, in addition to the XOR and M&S'81 problems. Furthermore, we perform experiments on several datasets for multiclass classification, comparing a single-layer network trained with the proposed CCE loss against

Kamaledin Ghiasi-Shirazi
Department of Computer Engineering, Ferdowsi University of Mashhad (FUM), Office No.: BC-123, Azadi Sq., Mashhad, Khorasan Razavi, Iran.
Tel.: +98-513-880-5158
Fax: +98-513-880-7181
E-mail: k.ghiasi@um.ac.ir
ORCID: 0000000160431820

LVQ, linear SVM, a single-layer network trained with the CE loss, and the method of Conaway and Kurtz (2017). The results show that the CCE loss performs remarkably better than existing algorithms for training single-layer neural networks.

Keywords Competitive Cross-Entropy · multiclass classification · nonlinearly separable problems · single-layer networks

1 Introduction

The introduction of the perceptron algorithm (Rosenblatt, 1958) for solving classification problems lead to a surge of interest in neural networks. However, after Minsky and Papert (1969) proved that the perceptron could only solve linearly separable problems, this interest faded. Specifically, they demonstrated that the perceptron could not solve even the simple XOR problem. More generally, they proved a 'Group Invariance Theorem' stating that the perceptron was unable to solve classification problems in which input data are invariant to a set of transformations that form a group. The interest in neural networks rose again after Rumelhart et al (1986) introduced the backpropagation algorithm and showed that multilayer networks trained with the backpropagation algorithm could solve nonlinearly separable problems. This historical sequence of events resulted in a myth that single-layer networks are inherently unable to solve nonlinearly separable problems.

Some researchers proposed new models of neurons which could solve nonlinearly separable problems using single-layer networks. Kohonen (1995) introduced learning vector quantization (LVQ) which was a competitive prototype-based algorithm that could solve non-linearly separable problems in a single-layer network based on the Euclidean-distance model of neurons. Urcid et al (2004) introduced the morphological perceptron which was a new computational model based on lattice algebra and showed that their model could solve the N-bit parity problem in a single-layer structure. Siomau (2014) proposed a quantum analog for the perceptron algorithm and showed that it could solve some non-linearly separable problems, including XOR. Truly speaking, the quantum perceptron algorithm is not a quantum implementation of the perceptron algorithm but a completely new learning machine based on linear operator theory (for a similar application of operator theory in clustering see (Bagarello et al, 2017)). Zhu et al (2017) introduced an FPGA-based computational model which could solve the XOR problem in a single-layer network. Recently, Conaway and Kurtz (2017) showed that, in a single-layer network of ordinary additive neurons, if the number of output neurons allotted to each class is equal to the number of input units, then the network can be trained using an autoassociative approach with MSE to solve certain nonlinearly separable problems. The method of Conaway and Kurtz (2017) could solve the XOR and M&S'81 (Medin and Schwanenflugel, 1981) problems, but it was unable to work any better than random guessing on the 3-bit parity problem.

Although Conaway and Kurtz (2017) studied single-layer networks from a pure theoretical perspective, single-layer networks still have applications in the era of big data since there exist fast training algorithms that can operate on massive datasets. For example, linear SVMs, which are essentially single-layer networks, are extensively used for solving large-scale classification problems (Fan et al, 2008; Chan et al, 2015; Girshick et al, 2016). Another example is distance-based classification (Mensink et al, 2013) which is known to be equivalent to training a single-layer dot-product neural network (Martín-del Brío, 1996).

One problem with the method of Conaway and Kurtz (2017) is that it minimizes the MSE loss, which is known to be inappropriate for all classification problems (Bishop, 1995). Two criteria that have been shown to be excellent for classification problems are the hinge loss and CE (Cross-Entropy) loss. The hinge loss is usually used by the kernel methods community while the CE criterion is the accepted choice in the field of neural networks, especially for deep learning. Interestingly, Rosasco et al (2004) have shown that both criteria have very similar and practically indistinguishable properties.

To train neural networks with the CE criterion, the output layer of the network should have the same number of neurons as the number of classes. By feeding the activation values of the output layer to a softmax function, one obtains a probability distribution over the classes. Assuming that the number of classes is c and the vector $z = (z_1, z_2, ..., z_c)$ represents the values of the output units, the estimate for the probability that the input belongs to class i is obtained by the following equation:

$$y_i = \frac{e^{z_i}}{\sum_{j=1}^{c} e^{z_j}}, \qquad for \;\; 1 \leq i \leq c. \tag{1}$$

For each training sample, a target probability distribution is presumed which assigns a value of one to the actual class and a value of zero to all other classes. Assuming that y represents the probability distribution generated by the network and τ represents the target probability distribution, the CE loss is defined as follows:

$$E^{CE} = -\sum_{i=1}^{c} \tau_i log(y_i) \tag{2}$$

In this paper, we propose the CCE (Competitive Cross-Entropy) loss for training networks with several output units for each class. Although the CCE criterion is general and can be utilized for training multilayer neural networks, in this paper we focus on single-layer networks and specifically compare our proposed CCE method with LVQ, linear SVM, a single-layer network trained with the CE loss, and the method of Conaway and Kurtz (2017). In comparison with linear SVMs and single-layer networks trained with the CE loss, a single-layer network trained with the proposed CCE loss has the advantage that it achieves higher recognition accuracies and can solve non-linearly separable problems. LVQ and the proposed CCE loss both allot multiple output neuron

to each class and have a competitive nature. However, in contrast to LVQ, the learning algorithm of the CCE loss has a rigid mathematical-optimization basis. In addition, the CCE loss operates on common neural networks which are based on the additive model of neurons, while LVQ neurons compute the Euclidean distance. Furthermore, while LVQ is inherently a single-layer neural network, the CCE criterion is a general loss function which can be used for training multilayer neural networks as well. Finally, our experiments show that, in practice, the CCE loss works much better than LVQ. Training the network by minimizing the CCE loss has several advantages over the training using an associative approach with MSE (which we call AAMSE for short) that Conaway and Kurtz (2017) proposed. Firstly, while in AAMSE the number of output neurons for each class must be equal to the number of input neurons, CCE allows an arbitrary number of output neurons for each class. Secondly, our experiments show that CCE is more powerful than AAMSE in solving nonlinearly separable problems. For example, while the method of Conaway and Kurtz (2017) was unable to solve the 3-bit parity problem any better than random guessing, CCE could solve it completely on many runs, having an average misclassification error of 2% over 100 runs. Thirdly, the proposed method is highly interpretable with output neurons representing clusters of data within the classes. We illustrate this by visualizing the functionality of each output neuron of a network trained on the MNIST dataset. Fourthly, since CCE is an extension of CE, it is a much more suitable criterion for classification compared with AAMSE, which is based on the MSE criterion. Finally, the CCE is not limited to single-class networks and can be easily applied to multilayer neural networks as well.

2 The Competitive Cross-Entropy Loss

Consider a multiclass classification problem with c classes. For each $k \in \{1, ..., c\}$, let O_k be the set of indices of the output neurons that we have assigned to class k and let $n = \sum_{k=1}^{c} |O_k|$ be the total number of output neurons. We presume that output neurons belonging to a class represent different clusters within that class. Let z be the vector of the values of the neurons in the output layer. By applying a softmax function to z, we obtain a probability distribution y over output neurons. In order to use the cross-entropy measure in this extended setting, we should define a desired target distribution which assigns a value of one to the actual cluster in the correct class, and a value of zero to all other clusters in all the classes. However, the supervised information available in the training data determines only the class of each sample and the exact output neuron to which that sample belongs is unspecified. Therefore, we should use unsupervised learning in accompany with the supervised information to obtain a target distribution. Clearly, the target distribution should assign all of its probability mass to output neurons allotted to that class. We propose that the desired probability distribution for each sample to be determined in a soft competitive way based on the activities

of the output neurons belonging to the class of that sample. Specifically, we obtain the desired probability mass by applying a softmax function only to the output neurons of the target class. We now express the proposed method rigidly in exact mathematical terms. Assuming that z represents the values of the last layer, the probability distribution of the network is obtained by the following equation:

$$y_i = \frac{e^{z_i}}{\sum_{j=1}^{n} e^{z_j}}, \qquad for \;\; 1 \leq i \leq n. \tag{3}$$

Assume that the input sample belongs to a class k. For output neurons i that belong to the true class k, i.e. $i \in O_k$, we set

$$\tau_i = \frac{e^{z_i}}{\sum_{j \in O_k} e^{z_j}} \tag{4}$$

and for those that belong to other classes, we set $\tau_i = 0$. We define the CCE (Competitive Cross-Entropy) loss as follows:

$$E^{CCE} = -\sum_{i=1}^{n} \tau_i log(y_i) \tag{5}$$

The derivative of E^{CCE} with respect to z is

$$\begin{aligned} \frac{\partial E^{CCE}}{\partial z_i} &= \sum_{j=1}^{n} \frac{\partial E^{CCE}}{\partial y_j} \frac{\partial y_j}{\partial z_i} = -\sum_{j=1}^{n} \frac{\tau_j}{y_j} \left(\delta_{ij} y_j - y_i y_j\right) \\ &= -\sum_{j=1}^{n} \tau_j \left(\delta_{ij} - y_i\right) = y_i - \tau_i \end{aligned} \tag{6}$$

where δ_{ij} is the Kronecker delta function which is 1 if i is qual to j and 0 otherwise. In contrast to the CE criterion, E^{CCE} is not convex and so the recognition accuracy of a network trained with the CCE loss depends on the initial values of the weights.

3 Experiments

In this section, we report our experiments on training single-layer networks with CCE. In subsection 3.1 we perform experiments using three nonlinearly separable problems and compare CCE training with AAMSE training. In subsection 3.3 we report our experiments using some standard datasets for multiclass classification.

3.1 Experiments on typical non-linearly separable problems

In this subsection, we perform experiments on the XOR, M&S'81, and 3-bit parity problems which are those that had been studied by Conaway and Kurtz (2017). For each problem, we performed 100 experiments, starting each one from a different random initialization of the weights. In all 300 experiments, we trained the network for 20 epochs. We experimentally found that a learning rate of 10 is suitable for solving these problems. We initialized the weights randomly with a uniform distribution between -0.25 and 0.25. For the XOR, M&S'81, and 3-bit parity problems we chose, respectively, two, three, and four output neurons for each class. Figure 1 shows the learning curves of our proposed method on these datasets averaged over 100 runs. In all of the 100 runs the proposed method could completely solve the XOR and M&S'81 problems. For the 3-bit parity problem, the average misclassification rate after 20 epochs was 2% which is far better than the 50% misclassification rate obtained by Conaway and Kurtz (2017). In fact, on this problem, 93 out of 100 runs ended with an error of 0.0%.

3.2 Experiments on standard datasets for multiclass classification

In this subsection, we perform experiments on Letter, MNIST, Pendigits, Sensorless, USPS, and Vowel datasets which are standard benchmarks for the multiclass classification task[1]. We compare our proposed CCE method with ordinary cross-entropy (CE), the AAMSE method of Conaway and Kurtz (2017), linear SVM, and LVQ. For CE and CCE methods, we chose the initial learning rate using 5-fold cross-validation from the set of values $\{10^{-2}, ..., 10^{2}\}$. During training, we multiplied the learning rate at the end of each epoch by 0.95, training the network for a total of 200 epochs. In addition, we initialized the weights with a uniform distribution between -0.01 and 0.01. For CCE we assigned 6 output neurons to each class[2]. We used the same method of initializing the weights and decreasing the learning rate for the AAMSE method of Conaway and Kurtz (2017) with the difference that, for the AAMSE method, the initial learning rate was chosen using 5-fold cross validation from the set of values $\{10^{-3}, ..., 10^{0}\}$. For linear SVM, we used 5-fold cross-validation for choosing the regularization parameter C from the range $\{2^{-14}, 2^{-13}, ..., 2^{13}, 2^{14}\}$. For LVQ we set the number of iterations and the number of output neurons to the same values as CCE. The other parameters for LVQ were chosen automatically by the heuristics of the LVQ-PAK program package (Kohonen et al, 1996). Table 1 summarizes the results of this experiment. Overall, the CCE method has a clear superiority over the other methods. It must be mentioned

[1] These datasets can be downloaded from `https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets`.

[2] Note that while in this experiment the proposed method has 60 output neurons, the number of output neurons for the method of Conaway and Kurtz (2017) is 7840.

Dataset	LVQ	Linear SVM	Autoassociative MSE	Cross Entropy	Competitive Cross Entropy
Letter	82.66%	63.24%	82.26%	77.54%	**91.40**%
Mnist	93.32%	91.95%	82.12%	92.28%	**96.51**%
Pendigits	96.08%	87.28%	95.20%	92.85%	**97.54**%
Sensorless	41.96%	55.19%	83.52%	89.91%	**95.63**%
Usps	**93.27**%	91.53%	92.18%	91.28%	92.68%
Vowel	44.16%	31.39%	20.13%	47.40%	**50.65**%

Table 1: Accuracy of the LVQ, linear SVM, the AAMSE method of Conaway and Kurtz (2017), CE, and CCE methods on solving different multiclass classification tasks using single-layer networks.

that the results of the CCE method could be improved even further by the weight-decay regularization method.

3.3 Visualizing the network operation on MNIST

In the previous subsection, we performed experiments on several datasets for multiclass classification. In this subsection, we focus on MNIST, which is one of these datasets, and visualize the operation of the CCE method on it. As stated in the previous section, the proposed CCE with six output neurons per class obtained a test accuracy of 96.51% while the best competing method obtained a test accuracy of 93.32%. To visualize the role of each output neuron in the network, we assigned the training samples to the output neurons and drew their average. Specifically, if the training sample i maximally activates the k'th output neuron, then we say that the training sample i belongs to the output neuron k. Figure 2 depicts the average of the training samples that belong to each output neuron. White cells in Figure 2 correspond to output neurons that were not chosen even by a single training sample. As can be seen, output neurons of each class specialize in detecting different styles of writing for that class.

4 Conclusions and future work

In this paper, we introduced the CCE (Competitive Cross-Entropy) loss which is a generalization of the CE (Cross-Entropy) loss to networks having several output neurons for each class. We showed that when applied to single-layer networks, CCE can be used to solve nonlinearly separable problems. Furthermore, by performing experiments on several benchmark datasets for multiclass classification, we demonstrated that using the CCE loss with an appropriate number of output neurons per class would remarkably increase the accuracy of single-layer neural networks. In the future, we plan to add CCE to single-layer toolboxes for large-scale multiclass classification like Online Passive-Aggressive (Crammer et al, 2006) and Liblinear (Fan et al, 2008). We anticipate that

training with the CCE loss would increase the recognition accuracy of these toolboxes. Another line of research is finding the right way for applying clustering algorithms like k-means to initialize the weights of the output neurons. Finally, we plan to assess the CCE loss in the training of deep neural networks.

References

Bagarello F, Cinà M, Gargano F (2017) Projector operators in clustering. Mathematical Methods in the Applied Sciences 40(1):49–59

Bishop CM (1995) Neural networks for pattern recognition. Oxford University Press

Chan TH, Jia K, Gao S, Lu J, Zeng Z, Ma Y (2015) Pcanet: A simple deep learning baseline for image classification? IEEE Transactions on Image Processing 24(12):5017–5032

Conaway N, Kurtz KJ (2017) Solving nonlinearly separable classifications in a single-layer neural network. Neural Computation 29(3):861–866

Crammer K, Dekel O, Keshet J, Shalev-Shwartz S, Singer Y (2006) Online passive-aggressive algorithms. Journal of Machine Learning Research 7(Mar):551–585

Fan RE, Chang KW, Hsieh CJ, Wang XR, Lin CJ (2008) Liblinear: A library for large linear classification. Journal of Machine Learning Research 9(Aug):1871–1874

Girshick R, Donahue J, Darrell T, Malik J (2016) Region-based convolutional networks for accurate object detection and segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence 38(1):142–158

Kohonen T (1995) Learning vector quantization. In: Self-organizing maps, Springer, pp 175–189

Kohonen T, Hynninen J, Kangas J, Laaksonen J, Torkkola K (1996) Lvq pak: The learning vector quantization program package. Tech. rep., Technical report, Laboratory of Computer and Information Science Rakentajanaukio 2 C, 1991-1992

Martín-del Brío B (1996) A dot product neuron for hardware implementation of competitive networks. IEEE Transactions on Neural Networks 7(2):529–532

Medin DL, Schwanenflugel PJ (1981) Linear separability in classification learning. Journal of Experimental Psychology: Human Learning and Memory 7(5):355

Mensink T, Verbeek J, Perronnin F, Csurka G (2013) Distance-based image classification: Generalizing to new classes at near-zero cost. IEEE Transactions on Pattern Analysis and Machine Intelligence 35(11):2624–2637

Minsky M, Papert S (1969) Perceptrons. MIT Press

Rosasco L, De Vito E, Caponnetto A, Piana M, Verri A (2004) Are loss functions all the same? Neural Computation 16(5):1063–1076

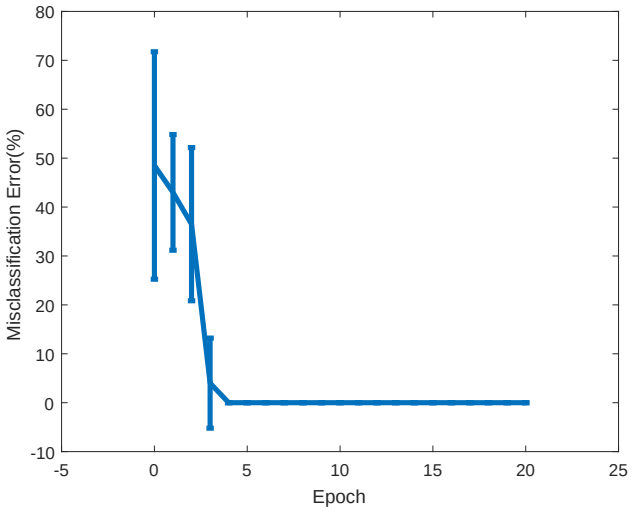
Rosenblatt F (1958) The perceptron: A probabilistic model for information storage and organization in the brain. Psychological Review 65(6):386

Rumelhart DE, Hinton GE, Williams RJ (1986) Learning representations by back-propagating errors. Nature 323(6088):533–538

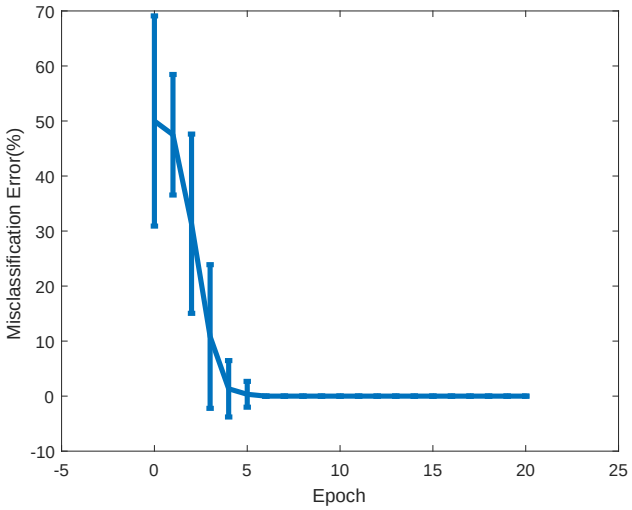
Siomau M (2014) A quantum model for autonomous learning automata. Quantum Information Processing 13(5):1211–1221

Urcid G, Ritter GX, Iancu L (2004) Single layer morphological perceptron solution to the n-bit parity problem. In: Iberoamerican Congress on Pattern Recognition, Springer, pp 171–178

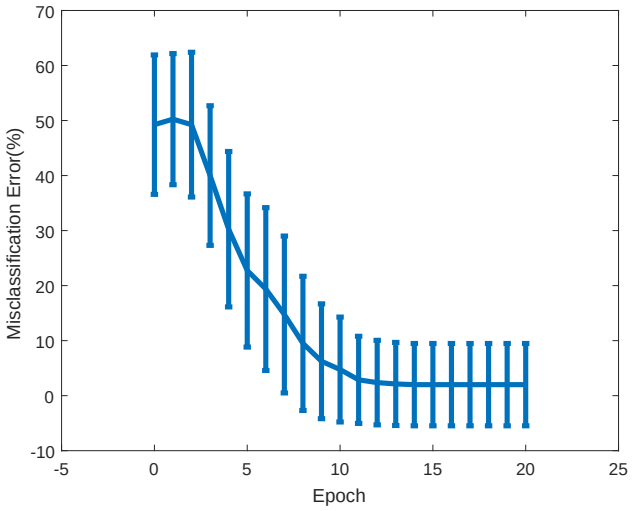
Zhu G, Lin L, Jiang Y (2017) Resolve xor problem in a single layer neural network. In: IWACIII 2017-5th International Workshop on Advanced Computational Intelligence and Intelligent Informatics, Fuji Technology Press Ltd



(a) Learning curve for the XOR problem averaged over 100 runs.



(b) Learning curve for the M&S'81 problem averaged over 100 runs.



(c) Learning curve for the 3bit parity problem averaged over 100 runs.

Fig. 1: Error bar plots for the proposed CCE method of misclassification errors vs epochs for the XOR, M&S'81, and 3-bit parity problems averaged over 100 runs.

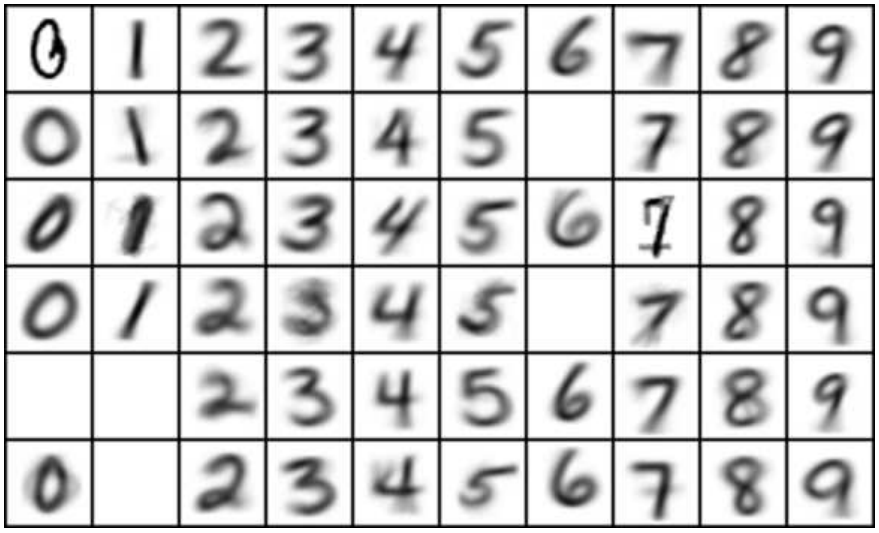


Fig. 2: Illustration of the functionality of each output neuron after training with CCE. Each cell corresponds to an output neuron and shows the average of training images that maximally activate that neuron.