Amended Cross-Entropy cost: an approach for encouraging diversity in classification ensemble (Brief Announcement)

Ron Shoham and Haim Permuter

Ben-Gurion University, Beer-Sheva 8410501, Israel
ronshoh@post.bgu.ac.il, haimp@bgu.ac.il

Abstract. In the field of machine learning, the training of an ensemble of models is a very common method for reducing the variance of the prediction, and yields better results. Many researches indicate that diversity between the predictions of the models is important for the ensemble performance. However, for Deep Learning classification tasks there is no explicit way to encourage diversity. Negative Correlation Learning (NCL) is a method for doing so in regression tasks. In this work we develop a novel algorithm inspired by NCL to explicitly encourage diversity in Deep Neural Networks (DNNs) for classification. In the development of the algorithm we first assume that the same training characteristics that hold in NCL must also hold when training an ensemble for classification. We also suggest the Stacked Diversified Mixture of Classifiers (SDMC), which is based on our outcome. SDMC is a layer that aims to replace the final layer of a DNN classifier. It can be easily applied on any model, while the cost in terms of number of parameters and computational power is relatively low.

1 Introduction

Ensemble methods are a simple and efficient way to yield better results by aggregating predictions from multiple models. Many works point out that the key for an ensemble to perform well is to encourage diversity among the models [2, 5, 6, 8, 10]. A well known framework for generating a diversified ensemble for regression tasks uses Negative Correlation Learning (NCL) criteria [1, 6, 8]. In this note we would like to develop a novel analogue framework for the classification problem. For regression, the negative correlation is well motivated from simple decomposition of the error into bias-variance-covariance [1, 6]. However, for classification problems such a framework is less clear. Currently, most of the ensembles in DNNs are obtained by training the same architecture multiple times with different seeds. The randomization achieves some diversity but it is done implicitly, without any clear criteria. We suggest an amended cost function for multiple classifiers which encourages diversity between different model predictions.

In general, the cost function used in machine learning can be motivated by several considerations. For instance, cross-entropy can be motivated by a maximum-likelihood criteria, but also by being a "good match" to sigmoid or softmax nodes for binary or multi-class cases, respectively [7]. Using a "good match" to a sigmoid or softmax node is also what motivates us in developing a cost function for ensemble classification. We show that by adding a penalty that

encourages increasing the cross-entropy between the predictions of the models we get the same learning characteristics as in NCL.

The novelty of our idea lies in our giving an explicit criterion for simultaneously training multiple models for an ensemble, while encouraging diversity explicitly. One of the benefits of this method is that all models are equally strong. We also suggest a variant called Stacked Diversified Mixture of Classifiers (SDMC), which can be applied on any DNN classifier easily, without increasing the number of parameters and computational power significantly. SDMC is a variant for the vanilla final softmax layer used in DNN, based on our outcome in this article.

2 Regression with Negative Correlation Learning

For regression tasks there is a well known technique for encouraging diversity in ensembles called *Negative Correlation Learning* (NCL)[1, 6]. A mathematical analysis shows that reducing the correlation between the regressors in an ensemble might leads to reducing the MSE of the ensemble (bias-variance-covariance decomposition). Its main idea is that by adding a penalty $p_i = (f_i - f_{ens}) \sum_{j \neq i} (f_j - f_{ens})$ for each model cost function, where f_i is the i'th model prediction and $f_{ens} = \frac{1}{M} \sum_{j=1}^{M} f_j$ is the ensemble prediction, we reduce the correlation between the predictors. This yields a new cost function:

$$e_i = \frac{1}{2}(f_i - t)^2 + \gamma p_i, \tag{1}$$

$$= \frac{1}{2}(f_i - t)^2 + \gamma (f_i - f_{ens}) \sum_{j \neq i} (f_j - f_{ens}). \tag{2}$$

When calculating its gradient, and setting $\lambda = 2\gamma(1 - \frac{1}{M})$, we get

$$\frac{\partial e_i}{\partial f_i} = (f_i - t) - \gamma[2(1 - \frac{1}{M})(f_i - f_{ens})] \tag{3}$$

$$= (f_i - t) - \lambda(f_i - f_{ens})$$

$$= (1 - \lambda)(f_i - t) + \lambda(f_{ens} - t). \tag{4}$$

3 Classification

Inspired by the above result, we would like to find a penalty for the classification cost that yields the same characteristics. In order to achieve this, we start from the outcome we got in (4) and integrate it. This procedure is similar to that presented in [7] for finding the cross-entropy as the desired cost function for a sigmoid classifier. The difference between classification and regression is that we use an activation function on the final layer[1]. For binary classification we use the sigmoid function $f_i(z_i) = \frac{1}{1+e^{-z_i}}$, in contrast to regression where $f_i(z_i) = z_i$. Therefore, based on the outcome in (4) we demand

[1] In this Brief Announcement we demonstrate our idea only on a sigmoid (binary classification), but the proof for softmax is similar and is presented in the full version of this paper.

$$\frac{\partial e_i}{\partial z_i} = (1-\lambda)(f_i - y) + \lambda(f_{ens} - y). \tag{5}$$

By applying the chain rule $\frac{\partial e_i}{\partial z_i} = \frac{\partial e_i}{\partial f_i}\frac{\partial f_i}{\partial z_i}$ and the result $\frac{\partial f_i}{\partial z_i} = f_i(1-f_i)$, we get

$$\frac{\partial e_i}{\partial f_i} = \frac{(1-\lambda)(f_i - y) + \lambda(f_{ens} - y)}{f_i(1-f_i)} \tag{6}$$

$$= \frac{f_i - y}{f_i(1-f_i)} - \frac{\lambda}{M}\sum_{j\neq i}\frac{f_i - f_j}{f_i(1-f_i)} \tag{7}$$

$$e_i = \int \frac{\partial e_i}{\partial f_i} df_i \tag{8}$$

$$= -y\log(f_i) - (1-y)\log(1-f_i)$$

$$- \frac{\lambda}{M}\sum_{j\neq i}\{-f_j\log(f_i) - (1-f_j)\log(1-f_i)\} \tag{9}$$

$$= H(y, f_i) - \frac{\lambda}{M}\sum_{j\neq i} H(f_j, f_i). \tag{10}$$

H is the cross-entropy function and $y \in \{0, 1\}$ is the true label. Therefore, by adding a penalty $p_i = -\frac{1}{M}\sum_{j\neq i} H(f_j, f_i)$ and choosing $\lambda \in [0, 1]$ we get a method to encourage diversity in classification ensembles explicitly.

4 Stacked Diversified Mixture of Classifiers

In this section we suggest a new architecture inspired by our above outcome and the D-ConvNet architecture [8]. We train a single DNN to generate features, and on top of the net, instead of using a vanilla softmax layer, we use a Stacked Diversified Mixture of Classifiers (SDMC). A SDMC is structure of multiple softmax layers with multiple amended cost functions for each softmax layer. An illustration of this architecture is shown in Fig. 1. The advantage of using this variant is that we do not need to train multiple networks simultaneously, which might significantly increase the training time and the computational power needed. Instead, we only train a single DNN and stack on top of it multiple classifiers. Each classifier has its own set of weights and is jointly optimized with the other classifiers by an amended cost function that penalizes low cross-entropy with others.

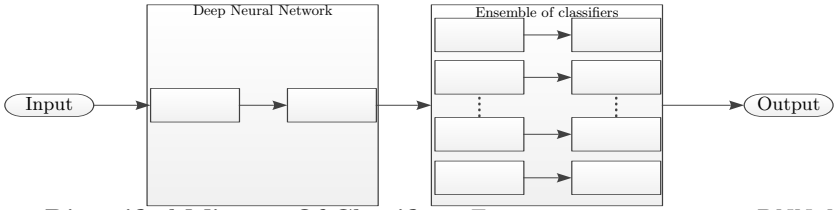

Fig. 1. Diversified Mixture Of Classifiers. First, an input is sent to a DNN. Next, the DNN performs initial processing and feature extraction out of the input. Finally, a pool of classifiers is trained using our suggested cost functions that penalize with respect to the cross-entropy with other classifiers.

5 Results

5.1 MNIST using vanilla diversified classifiers

The MNIST is a standard toy dataset, where the task is to classify the images into 10 digit classes. Our goal here was to get some proof of concept and to observe training behaviour when using our cross-entropy penalty. Here, we used only our vanilla version and did not apply a SDMS variant. Our architecture was of a single hidden layer DNN with ReLU activation. We set the number of models to be $M = 5$ and changed the values of λ. The results are shown in Table 1. Results include both the accuracy and the cross-entropy of the predictions over the test set. We notice from the results that our method reduces the cross-entropy and get higher accuracy for $\lambda > 0$. We observe that even though the performance of a single net deteriorated when increasing λ, the ensemble performs better.

	Ensemble scores		**Single net scores**	
λ	**Accuracy**	**CE**	**Accuracy**	**CE**
0	0.9790	0.0669	0.9767	0.0810
0.05	0.9798	0.0663	**0.9770**	0.0809
0.1	0.9799	0.0664	0.9768	**0.0802**
0.3	0.9797	0.0658	0.9767	0.0806
0.5	**0.9802**	**0.0649**	0.9764	0.0842
0.7	0.9800	0.0659	0.9760	0.0866

Table 1. Results on MNIST using our suggested cost function. Ensemble scores refers to the accuracy and cross-entropy(CE) of the ensemble prediction over the test set. Single net scores refers to the scores of the prediction of a single model in the ensemble. Scores are averaged over 6 experiments with a different seed for each λ.

5.2 CIFAR-10 using SDMC

We conducted studies of the SDMC over the CIFAR-10 dataset [4]. We used the architecture and code of ResNet 110 [3] and stacked on top of it an ensemble of 10 classifiers. This resulted in adding 5850 parameters to a model with an original size of 1731002, i.e. enlarging the model by 0.34%. The results are shown in Table 2. In the results we see that the optimal λ reduces the error by $\sim 7\%$ with almost no cost in the number of parameters and computational power. We also see that the cross-entropy reduces significantly. We notice that the optimal λ is lower than the vanilla usage of our method.

	$M=1$	$M=10$ $\lambda=0$	$M=10$ $\lambda=0.001$	$M=10$ $\lambda=0.01$	$M=10$ $\lambda=0.05$	$M=10$ $\lambda=0.1$	$M=10$ $\lambda=0.3$	$M=10$ $\lambda=0.5$
error(%)	6.43	6.2	6.14	6.12	**5.98**	6.09	6.13	6.31
CE	0.3056	0.3102	0.3041	0.3048	0.2968	**0.2918**	0.3137	0.4957

Table 2. Results on CIFAR-10 test set using SDMC with ResNet 110. M refers to the number of classifiers, and CE stands for cross-entropy. We ran each model 5 times and show "best(mean-std)" as in [3, 9].

6 Conclusion

In this paper we propose a novel Deep Learning Classification Framework for encouraging diversity explicitly, based on cross-entropy penalties. First, we introduced the idea of using an amended cost function for multiple classifiers based on NCL results. Later, we introduce Stacked Diversified Mixture of Classifiers (SDMC) which aims to improve the capabilities of a model without increasing the number of parameters and computational power significantly.

Bibliography

[1] Gavin Brown, Jeremy L Wyatt, and Peter Tiňo. Managing diversity in regression ensembles. *Journal of machine learning research*, 6(Sep):1621–1650, 2005.

[2] Miguel A. Carreira-Perpinan and Ramin Raziperchikolaei. An ensemble diversity approach to supervised binary hashing. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 757–765. Curran Associates, Inc., 2016.

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[4] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.

[5] Stefan Lee, Senthil Purushwalkam Shiva Prakash, Michael Cogswell, Viresh Ranjan, David Crandall, and Dhruv Batra. Stochastic multiple choice learning for training diverse deep ensembles. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 2119–2127. Curran Associates, Inc., 2016.

[6] Yong Liu and Xin Yao. Ensemble learning via negative correlation. *Neural networks*, 12(10):1399–1404, 1999.

[7] Michael A Nielsen. *Neural networks and deep learning*, volume 25.

[8] Zenglin Shi, Le Zhang, Yun Liu, Xiaofeng Cao, Yangdong Ye, Ming-Ming Cheng, and Guoyan Zheng. Crowd counting with deep negative correlation learning.

[9] Rupesh K Srivastava, Klaus Greff, and Jürgen Schmidhuber. Training very deep networks. In *Advances in neural information processing systems*, pages 2377–2385, 2015.

[10] Tianyi Zhou, Shengjie Wang, and Jeff A Bilmes. Diverse ensemble evolution: Curriculum data-model marriage. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 5909–5920. Curran Associates, Inc., 2018.