Pairwise Margin Maximization for Deep Neural Networks

Berry Weinstein
School of Computer Science
The Interdisciplinary Center
berry.weinstein@post.idc.ac.il

Shai Fine
Data Science Institute
The Interdisciplinary Center
shai.fine@idc.ac.il

Yacov Hel-Or
School of Computer Science
The Interdisciplinary Center
toky@idc.ac.il

***Abstract*—The weight decay regularization term is widely used during training to constrain expressivity, avoid overfitting, and improve generalization. Historically, this concept was borrowed from the SVM maximum margin principle and extended to multi-class deep networks. Carefully inspecting this principle reveals that it is not optimal for multi-class classification in general, and in particular when using deep neural networks. In this paper, we explain why this commonly used principle is not optimal and propose a new regularization scheme, called *Pairwise Margin Maximization* (PMM), which measures the minimal amount of displacement an instance should take until its predicted classification is switched. In deep neural networks, PMM can be implemented in the vector space before the network's output layer, i.e., in the deep feature space, where we add an additional normalization term to avoid convergence to a trivial solution. We demonstrate empirically a substantial improvement when training a deep neural network with PMM compared to the standard regularization terms.**

I. Introduction

Over the last decade, deep neural networks (DNNs) have become the *machine learning* method of choice in a variety of applications, demonstrating outstanding performance, often close to or even better than human-level performance. Nevertheless, some researchers have shown that DNNs can generalize poorly even with small data transformations [1] as well as overfit arbitrarily corrupted data [2]. Additionally, problems such as adversarial attacks, which cause neural networks to misclassify slightly perturbed input data, can be a source of concern in real-world scenarios (cf. [3], [4]). These challenges have motivated researchers to investigate whether properties that enabled classical machine learning algorithms to overcome the above-mentioned problems can be useful in helping DNNs resolve similar issues. Specifically, for linear classifiers, it has been evident that a classifier with a large margin over different classes in the training data produced better generalization results as well as stronger robustness to input perturbations [5]. The maximal margin principle, i.e., maximizing the smallest distance from the instances to the classification boundary in the feature space, has played an important role in theoretical analysis of generalization, and helped to achieve remarkable practical results [6], as well as robustness to input perturbations [5]. Of particular interest to our study are the extensions to multi-class classification: multi-class perceptron (see Kesler's construction, [7]); multi-class SVM [8]; multi-class margin distribution [2], and the mistake-bound for multi-class linear separability that scales with $(R/\gamma)^2$, where R is the maximal norm of the samples in the feature space, and γ is the margin [9].

In all the above studies, the multi-class extension of the maximal margin principle is realized by the weight decay regularization scheme, which is a natural extension of the regularization term in the binary SVM case. We argue here that the standard weight decay regularization is not optimal in the multi-class case. In particular, we show that the weight decay term aims at maximizing the margins along the *one-vs-all* decision boundaries. According to the way a multi-class classification prediction is implemented, however, these decision boundaries are less of interest and the focus should be on using the *one-vs-one* decision boundaries as the margins. For the one-vs-one decision boundaries, we present a novel regularization term, which we call *Pairwise Margin Maximization* (PMM). The PMM regularization term maximizes the margins around the one-vs-one boundaries and can be added to any loss. We derive the regularization term starting from the first principle in the binary case and generalize it to the multi-class setting.

In systems where multi-class classification uses DNNs, we propose an extension that applies the maximal margin principle in the feature space rather than in the input space. Nevertheless, since the feature space is learnable, the margins can be trivially maximized by scaling the feature space. We address this issue by scaling our formulation by the radius of the feature space, i.e., the maximal norm of the samples in the feature space. This normalization is similar in spirit to the normalization proposed in [9]. We applied PMM on the last layer of various DNNs, in different classification tasks and in various domains. We empirically show that the PMM scheme provides a substantial improvement in accuracy while maximizing the margins between pairwise classes. In particular, we achieve considerable accuracy improvement in various image and text classification tasks, including CIFAR10, CIFAR100, ImageNet, MNLI, and QQP.

To conclude, our contribution is a novel regularization term, specifically designed to enforce the maximization of the margin on the one-vs-one decision boundaries, rather than maximizing the margin on the one-vs-all decision boundaries, which is derived by the weight decay regularization scheme. We show both visually and empirically that this scheme improves the

separation between classes as well as the accuracy over a large set of classification problems.

A. Previous Approaches

The maximal margin principle has proven to be fundamentally important in machine learning as it was shown to correlate with generalization and accuracy [10]. Although most efforts revolved around binary classification, extensions to multi-class classification were also suggested [2], [8], [9], [11]. Soudry et al. [12] proved that cross-entropy loss in linear DNNs, together with *stochastic gradient descent* (SGD) optimization, converges to the maximal margin solution. This proof, however, was not extended to nonlinear DNNs, and indeed, Sun et al. [13] affirmed that cross-entropy alone is not enough to achieve the maximal margin in non-linear DNNs and that an additional regularization term is needed.

Margins in the input space using nonlinear DNNs can be approximated using derivatives assisted by the back-propagation scheme. Elsayed et al. [14] presented a multi-class linear approximation of the margins as an alternative loss function. They applied their margin-based loss at each and every layer of the neural network. Moreover, their method requires a second-order derivative computation due to the presence of first order gradients in the loss function itself. Explicit computation of the second order gradients for each layer of the neural network, however, can be quite expensive, especially when DNNs are getting wider and deeper. To address this limitation, they used a first-order linear approximation to deploy their loss function more effectively. Later, Jiang et al. [15] presented a margin-based measure that strongly correlates with the generalization gap in DNNs. Essentially, they measured the difference between the training and the test performances of a neural network using marginal distribution statistics [16]. Sokolić et al. [17] used the input layer to approximate the margin via the network's Jacobian matrix and showed that maximizing their approximations leads to a better generalization. In contrast, we show that applying our margin-based regularization to the output layer alone achieves substantial improvement.

The rest of the paper is organized as follows. In Section II we explain the need for PMM in deep classification problems with the classical maximal margin principle in binary and multi-class settings. In Section III we derive the PMM regularization scheme for DNNs and show preliminary results on a simple classification task using the CIFAR10 dataset. We expand the experimental results in Section IV by applying PMM to additional vision tasks on various CNN architectures as well as binary *natural language processing* (NLP) classification tasks on the $\text{BERT}_{\text{BASE}}$ model. Lastly, we summarize our contributions in Section V.

II. Margin Analysis for Binary and multi-class Classification

The maximal margin principle is traditionally presented in the context of a shallow linear classifier [8], [11]. It is the core principle behind the *support vector machine* (SVM) classifier. It was shown that maximizing the margin between the data samples and the decision boundary also maximizes the classifier's generalization capabilities [8]. In the original work by Vapnik [6], the maximal margin principle was applied to the data points closest to the boundary (support vectors), while later works [18] extended the maximal margin principle to the mean and variance of the distances.

We start our discussion with the classical derivation of the maximal margin principle and its traditional extension to multi-class case. We then show that this extension is not optimal as it refers to the one-vs-all maximal margin principle. Next, we suggest a new maximal margin principle that we term *Pairwise Margin Maximization* (PMM). In fact, the PMM principle emerged from the maximal margin principle where the one-vs-one classification scheme is carried out. Lastly, we describe the necessary components for adapting PMM to DNNs.

Consider a classification problem with two classes $\mathcal{Y} \in \{+1, -1\}$. We denote by $\mathcal{X} \in \mathcal{R}^d$ the input space. Let $f(\mathbf{w}^T\mathbf{x} + b)$ be a linear classifier, where $\mathbf{x} \in \mathcal{X}$ and

$$f(z) = \begin{cases} +1 & \text{if } \ z \geq 0 \\ -1 & \text{otherwise} \end{cases}$$

The classifier is trained using a set of examples $\{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \cdots, (\mathbf{x}_n, y_n)\} \in (\mathcal{X} \times \mathcal{Y})^n$ where each example is sampled identically and independently from an unknown distribution $\mathcal{D}$ over $\mathcal{X}$. The goal is to classify correctly new samples drawn from $\mathcal{D}$.

Denote by ℓ the (linear) decision boundary defined by the classifier f:

$$\ell = \{\mathbf{x} \mid \mathbf{w}^T\mathbf{x} + b = 0\} \tag{1}$$

The geometric distance of a point $\mathbf{x}$ from ℓ is given by

$$d(\mathbf{x}) = \frac{\mathbf{w}^T\mathbf{x} + b}{\|\mathbf{w}\|} \tag{2}$$

For a linearly separable training set, the maximum margin classifier demonstrates the best generalization capability, which is achieved by selecting the classifier that maximizes the margin $\hat{d}$ [6]:

$$\hat{d} = \arg\max_{\mathbf{w},b} d \quad \text{s.t.} \quad y_i \frac{\mathbf{w}^T\mathbf{x}_i + b}{\|\mathbf{w}\|} \geq d, \qquad \forall i = 1, \cdots, n$$

This optimization is redundant with the length of $\mathbf{w}$ and b. Namely, if $(\mathbf{w}^*, b^*)$ is the optimal solution, then so is $(\alpha\mathbf{w}^*, \alpha b^*)$. Imposing $y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1$ removes this redundancy and results in the following equivalent minimization problem [6]:

$$\min_{\mathbf{w},b} \|\mathbf{w}\|^2 \quad \text{s.t.} \quad y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1, \qquad \forall i = 1, \cdots, n$$

The above optimization forces all samples to be classified correctly (with a margin no less than 1). To handle noisy and linearly inseparable data, the set of linear constraints is relaxed using soft margins by replacing it with the hinge loss,

$$\min_{\mathbf{w},b} \|\mathbf{w}\|^2 + \lambda \sum_i \max(0, 1 - y_i(\mathbf{w}^T\mathbf{x}_i + b)) \tag{3}$$

The two terms in the above minimization problem employ two complementary forces. The left term is the *regularization*

component and it promotes increasing the margin between the data points and the decision boundary, thus improving the generalization capability. The right term of the formula is the *empirical risk* component, promoting correct classification of the training samples.

We now extend the maximal margin principle to the multi-class case. Let us assume that we have a classification problem with k classes, $\mathcal{Y} \in \{1, \cdots, k\}$, and a set of n training samples: $\{(\mathbf{x}_i, y_i)\} \in (\mathcal{X} \times \mathcal{Y})^n$. For a given input $\mathbf{x}$, a trained classifier assigns a set of scores, i.e., a score to each class: $s_j(\mathbf{x}) : \mathcal{X} \to \mathbb{R},\ \forall\ j \in \mathcal{Y}$. For a linear classification, the j^{th} score of instance $\mathbf{x}$ is:

$$s_j(\mathbf{x}) = \mathbf{w}_j^T \mathbf{x} + b_j$$

The predicted class is then chosen by the maximal score attained over all classes,

$$\hat{y} = \arg\max_{j \in \mathcal{Y}} s_j(\mathbf{x})$$

For a training pair $(\mathbf{x}_i, y_i)$, denote by $s_{y_i}(\mathbf{x}_i)$ the score attained for the true class of $\mathbf{x}_i$ and by $s_{m_i}(\mathbf{x}_i)$ the maximal score attained for the non-true classes, i.e., m_i is the most *competitive* class with respect to y_i:

$$m_i = \arg\max_{j \neq y_i} s_j(\mathbf{x}_i)$$

We define:

$$\xi_i = s_{y_i}(\mathbf{x}_i) - s_{m_i}(\mathbf{x}_i)$$

ξ_i is the difference between the true score and the score of the *competitive* class. When ξ_i is positive, the true class attains the best score; otherwise ξ_i is negative. The larger ξ_i is, the larger the margin we have between the true and the competitive scores.

The commonly used multi-class classification scheme is defined as a one-vs-all classification scheme, where the maximal margin principle of Equation 3 is generalized to the multi-class [11]:

$$\min_{W,b} \sum_{j=1}^{k} \|\mathbf{w}_j\|^2 + \lambda \sum_{i=1}^{n} \max(0, 1 - \xi_i) \tag{4}$$

In the above minimization, the optimization is over $W, b \doteq \{(\mathbf{w}_j, b_j)\}_{j=1}^k$. The left term, as in the binary case, refers to the regularization term and emerges from the maximum margin principle. Since it minimizes $\|\mathbf{w}_j\|^2$, $j = 1..k$, it promotes large margins along the k decision boundaries:

$$\ell_j = \{\mathbf{x} \mid \mathbf{w}_j^T \mathbf{x} + b_j = 0\}, \qquad j = 1..k$$

Nevertheless, the desired property is not the way in which the predicted class is evaluated, maximizing the margin along each ℓ_j. Rather, we believe that the values of ξ_i should be maximized, i.e., the difference between the distance $\mathbf{x}_i$ to ℓ_{y_i} and its distance to ℓ_{m_i}. In other words, the regularization term, as defined in Equation 4, does not maximize the correct margins.

To clarify this point, consider the illustrative example in Figure 1. A data point denoted by $\mathbf{x}$ is given as a training input. Assume the true class of this data point is class 1 (the blue class). The most competitive class to class 1 is class 2 (the green class). Consider the margin in the space: $\ell_{1,2} = \{\mathbf{x} \mid s_1(\mathbf{x}) = s_2(\mathbf{x})\}$. The line $\ell_{1,2}$ splits the space into two half spaces where one side includes the data points closer to ℓ_1 and, in the other side, the points are closer to ℓ_2. In Figure 1, the point $\mathbf{x}$ is located on the right side of $\ell_{1,2}$ and will be classified correctly since $s_1(\mathbf{x}) > s_2(\mathbf{x})$. If, however, we maximize the margins around all of ℓ_i, $i = 1..3$ (the one-vs-all boundaries), the point $\mathbf{x}$ is located on the wrong side of ℓ_2 and the parameters $\mathbf{w}_2, b_2$ will be updated to maximize the margin of ℓ_2, although the predicted classification of $\mathbf{x}$ is correct and the margin width is satisfied.

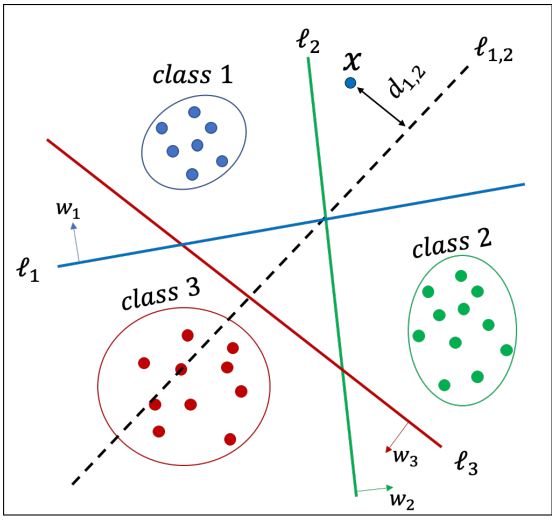

Fig. 1. Illustrative example of pairwise decision boundaries. Three classes along with their one-vs-all decision boundaries are presented. The true class of sample point $\mathbf{x}$ is class 1 (the blue class) while the most competitive class for this point is class 2 (the green class). Thus, the margin in question is in respect to $\ell_{1,2}$ whose distance to $\mathbf{x}$ is $d_{1,2}$.

In the following we suggest an alternative approach that is derived and justified directly from maximization of ξ_i. We start with the observation that although we deal with k decision boundaries, these boundaries can induce one-vs-one boundaries as well. For any two classes, $(p, q) \in \mathcal{Y} \times \mathcal{Y}$, the pairwise decision boundary between p and q is given by (see Figure 1):

$$\ell_{p,q} = \{\mathbf{x} \mid s_p(\mathbf{x}) = s_q(\mathbf{x})\} = \{\mathbf{x} \mid \mathbf{w}_p^T \mathbf{x} + b_p = \mathbf{w}_q^T \mathbf{x} + b_q\}.$$

Denoting $\mathbf{w}_{p,q} = \mathbf{w}_p - \mathbf{w}_q$ and $b_{p,q} = b_p - b_q$, the decision boundary $\ell_{p,q}$ can be rewritten as:

$$\ell_{p,q} = \{\mathbf{x} \mid \mathbf{w}_{p,q}\mathbf{x} + b_{p,q} = 0\}$$

which is similar to the binary case in Equation 1 where $\mathbf{w}_{p,q}$ replaces $\mathbf{w}$ and $b_{p,q}$ replaces b. In fact, although we optimize only for k classification parameters ($(\mathbf{w}_i, b_i)$, $i = 1..k$), the resulting parameters can be interpreted as $k(k-1)/2$ decision boundaries that are associated with all pairwise one-vs-one classifications.

Similarly to Equation 2, the geometric distance of a point $\mathbf{x}$ from $\ell_{p,q}$ is (see Figure 1):

$$d_{p,q}(\mathbf{x}) = \frac{\mathbf{w}_{p,q}^T \mathbf{x} + b_{p,q}}{\|\mathbf{w}_{p,q}\|} \tag{5}$$

For point $\mathbf{x}_i$, the decision boundary between y_i and its most competitive class m_i is ℓ_{y_i,m_i}, whose geometric distance to $\mathbf{x}_i$ is

$$d_{y_i,m_i}(\mathbf{x}_i) = \frac{\mathbf{w}_{y_i,m_i}^T \mathbf{x}_i + b_{y_i,m_i}}{\|\mathbf{w}_{y_i,m_i}\|} \tag{6}$$

Thus, our goal is to maximize the margins around ℓ_{y_i,m_i}, for $i = 1..n$. Note that $d_{y_i,m_i}(\mathbf{x}_i)$ is non-negative if the classification is correct ($s_{y_i}(\mathbf{x}_i) \geq s_{m_i}(\mathbf{x}_i)$) and negative otherwise.

Following the above justification, the same procedure applied to Equation 2 to derive Equation 3 in the binary case can be applied here as well. Thus, Equation 6 yields:

$$\min_{W,b} \sum_i \|\mathbf{w}_{y_i,m_i}\|^2 + \lambda \sum_i \max(0, 1 - \xi_i) \tag{7}$$

where,

$$\xi_i = \mathbf{w}_{y_i,m_i}^T \mathbf{x}_i + b_{y_i,m_i} = s_{y_i}(\mathbf{x}_i) - s_{m_i}(\mathbf{x}_i)$$

as defined above.

The above regularization term aims at increasing $d_{y_i,m_i}(\mathbf{x}_i)$, namely, the margins along ℓ_{y_i,m_i}, as desired. In contrast to Equation 4 and the common L_2 regularization scheme, where the Frobenius norm of W is minimized: i.e., $\sum_{j=1}^k \|\mathbf{w}_j\|^2$, Equation 7 minimizes the pairwise margins $\sum_i \|\mathbf{w}_{y_i,m_i}\|^2$, which is a different regularization objective. For a pair of classes (i, j), the standard L_2 regularization minimizes $\|\mathbf{w}_i\|^2 + \|\mathbf{w}_j\|^2$ while the suggested scheme minimizes $\|\mathbf{w}_i\|^2 + \|\mathbf{w}_j\|^2 - \mathbf{w}_i^T \mathbf{w}_j$. Since the margins are defined over each pairwise boundary, we call this regularization scheme a *pairwise margin maximization* (PMM).

Another point to note here is that the summation in Equation 7 is performed over the instance points (i is the instance index). This means that the larger the existence of the pair (i, j) as a competitive pair, the stronger the applied regularization. This can be interpreted as a minimization over the *margin distribution* rather than the maximal margin per se. If the instances are evenly distributed over the classes, then this is equivalent to summation over the class pairs. Otherwise, this summation compensates for class imbalance in the regularization term. Additionally, the PMM regularization term is applied without any additional computational cost since the pairwise terms are computed per example and the entire $O(k^2)$ pairwise terms are not necessarily computed.

III. Maximal Margin in Deep Networks

Applying PMM directly to DNNs poses several problems. First, these networks employ a nonlinear mapping from the input space into a feature space: $\phi_i = F(\mathbf{x}_i, \theta) : \mathcal{X} \to \Phi$, where θ are the network's parameters. The vector ϕ_i can be interpreted as a feature vector based on which the last layer in the network calculates the scores, for each class, via a fully-connected layer, $s_j(\phi_i) = \mathbf{w}_j^T \phi_i + b_j$. Maximizing the margin in the input space $\mathcal{X}$, as suggested in [17], requires back-propagating derivatives along the network up to the input layer, and calculating distances to the boundary up to the first order approximation. In highly nonlinear mappings, this approximation becomes inaccurate very fast as we move away from the decision boundary.

To overcome this problem, our scheme maximizes the margin only in the last layer, where the distances to the decision boundary are Euclidean in the *feature space* Φ:

$$d_{y_i,m_i}(\phi_i) = \frac{\mathbf{w}_{y_i,m_i}^T \phi_i + b_{y_i,m_i}}{\|\mathbf{w}_{y_i,m_i}\|} \tag{8}$$

Working with the feature space, however, presents a new challenge because the feature space Φ can be modified in the course of training. This is different than maximizing the margins in the input space $\mathcal{X}$ (c.f. Equation 6) or in a kernel-induced feature space (e.g., SVM), since in both cases the space is fixed. If the feature space keeps changing, then maximizing the margins in Equation 8 can be trivially attained by scaling up the feature space Φ.

To avoid the trivial solution, we must normalize the feature space Φ. In our scheme, we divide Equation 8 by $\|\phi_{max}\|$, which is the maximal norm of the samples (in the feature space) in the current batch. This ensures that scaling up the feature space will not increase the distance arbitrarily. Putting all the components of our scheme together, we end up with the following optimization problem:

$$\min_{W,\mathbf{b}} \sum_i \mathcal{R}_i + \lambda \sum_i \mathcal{C}_i \tag{9}$$

where

$$\mathcal{R}_i = \|\mathbf{w}_{y_i,m_i}\|^2 \|\phi_{max}\|^2$$

is the pairwise regularization term, and $\mathcal{C}_i$ is the empirical risk term:

$$\mathcal{C}_i = \max(0, 1 - \xi_i)$$

Finally, for DNNs, better classification results are commonly obtained using cross-entropy rather than hinge loss. Our formulation supports employing cross-entropy as well. The empirical risk term is simply replaced with

$$\mathcal{C}_i = -\log(P_{y_i}) \tag{10}$$

where P_{y_i} is the conditional probability of the true label y_i as obtained from the network after the softmax layer:

$$P_{y_i} = \frac{e^{s_{y_i}(\mathbf{x}_i)}}{\sum_j e^{s_j(\mathbf{x}_i)}}$$

Similarly to the hinge loss formulation, the cross-entropy term will strive for correct classification while the regularization term will maximize the margin.

Note that the regularization term in this scheme is different from the weight decay commonly applied in DNNs. First, the minimization is applied over the differences: $\|\mathbf{w}_{y_i,m_i}\|^2 = \|\mathbf{w}_{y_i} - \mathbf{w}_{m_i}\|^2$. Next, the regularization term is multiplied by

the $\|\phi_{max}\|$. Lastly, the regularization term is implemented only at the last layer.

The effect of the PMM regularization scheme compared to the baseline L_2 regularization is demonstrated in Figure 2. In this plot we consider the feature points in the penultimate layer of the ResNet44 (before the fully connected layer) trained on the CIFAR-10 data set. The features were projected to a lower dimension using t-SNE [19] and visualized in a 2D scatter plot. Each class is indicated by a different color. The left plot presents the feature distribution using the standard L_2 regularization while the right plot shows the distribution using the PMM scheme. It is demonstrated that the PMM plot presents well-separated clusters with large margins between each class while the baseline plot indicates overlapping classes with small margins.

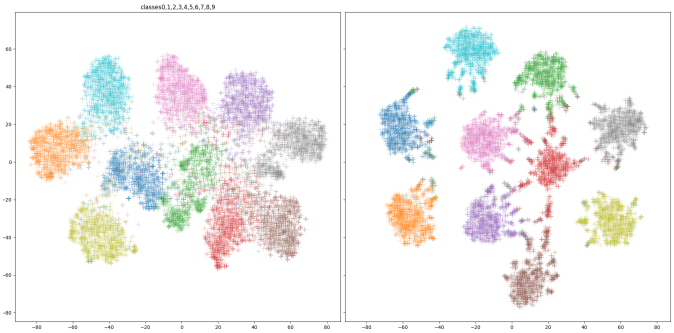

Fig. 2. A scatter plot of the last layer in ResNet44 trained on CIFAR-10 after embedding into 2D space using t-SNE. Left: Baseline using L_2 regularization. Right: PMM using the suggested scheme.

IV. Experiments

In this section[1], we report on a series of experiments designed to evaluate PMM's ability to achieve a higher accuracy score. The experiments were conducted on commonly used datasets and DNN architectures, in *vision* and NLP realms. For image classification, we used CIFAR10, CIFAR100 [20] and ImageNet [21] datasets. For natural language inference, we used Question NLI (QNLI) [22], MultiNLI (MNLI) [23] and Recognizing Textual Entailment (RTE) [24]. Lastly, for text classification and sentence similarity, we used the MSR Paraphrase Corpus (MRPC) [25], Quora Question Pairs (QQP) [26], and the Stanford Sentiment Treebank-2 (SST-2) [27].

A. Image Classification

For small-scale image classification, we used CIFAR10 and CIFAR100 datasets. These datasets comprise 32×32 color images from 10 or 100 classes, consisting of 50k training examples and 10k test examples. The last 5k images of the training set are used as a validation set, as suggested in common practice. For our experiments, we used ResNet-44 [28] and WRN-28-10 [29] architectures. We applied the original hyperparameters and training regime using a batch size of 64. In addition, we used the original augmentation policy as described in [28] for ResNet-44, while adding cutout [30] and auto-augment [31] for WRN-28-10. Optimization was performed for 200 epochs (equivalent to 156k iterations) after which baseline accuracy was obtained with no apparent improvement.

PMM was added to the objective function as an additional regularization term, where α is a trade-off factor between the cross-entropy loss and the regularization [2]:

$$\mathcal{L}(\theta) = \alpha \sum_i \mathcal{R}_i + \sum_i \mathcal{C}_i$$

To find the optimal α, we used a grid search and found that a linear scaling of α in the range of [1e-5..1e-3] works best for CIFAR10/100 and static $\alpha =$ 1e-5 works best for ImageNet.

Figures 2 and 3 show qualitative comparisons between the standard L_2 regularization and the suggested PMM scheme. They both present scatter plots of feature points taken from the penultimate layer of the network (before the classification layer). These feature points were projected into 2D using t-SNE [19]. Figure 3 consists of a set of 10 plots for the standard regularization scheme (upper panel) and for the PMM regularization (lower panel). Each plot presents five randomly selected classes. In both figures (Figs 2 and 3), it is demonstrated that the feature points in the PMM scheme are clustered into distinct classes with large margins, while the baseline scheme presents tight and overlapping clusters with small margins.

With respect to quantitative evaluation, Table I demonstrates our final classification results on CIFAR-10 and CIFAR-100 when implementing PMM on several commonly used architectures. On CIFAR-10, we managed to improve baseline accuracy in ResNet-44 from 93.22% to 93.83% and in VGG, from 93.19% to 93.34%. On CIFAR-100, we show a substantial increase using the WRN-28-20 model, raising its absolute accuracy by more than 1%. In Figure 4 we compare the error rates of the PMM regularization scheme with weight decay and dropout regularization, which are commonly used in DNNs. From Figure 4, it is clear that the other regularization techniques do not match the accuracy gain of the PMM scheme. Additionally, adding weight decay to PMM does not improve the error rate.

For large-scale evaluation, we used the ImageNet dataset [21], containing more than 1.2M images in 1k classes. We used MobileNet [32] architecture and followed the training regime established by [33] (an initial learning rate (LR) of 0.1 is decreased by a factor of 10 in epochs 30, 60, and 80, for a total of 90 epochs). We used a batch size of 256 and L_2 regularization over weights of convolutional layers as well as the standard data augmentation. Comparing the PMM scheme with the baseline scheme shows that accuracy increased from 71.17% to 71.44% (see Table I).

[1]All experiments were conducted using PyTorch; the code will be released on github upon acceptance of the paper.

[2]This formulation is equivalent to Equation 9, where $\alpha = \frac{1}{\lambda}$. It is preferred because it leads to multiplying the regularization term by a small number and keeping the scaling factor of $\mathcal{C}_i$ to be 1, thus avoiding gradient enlargement.

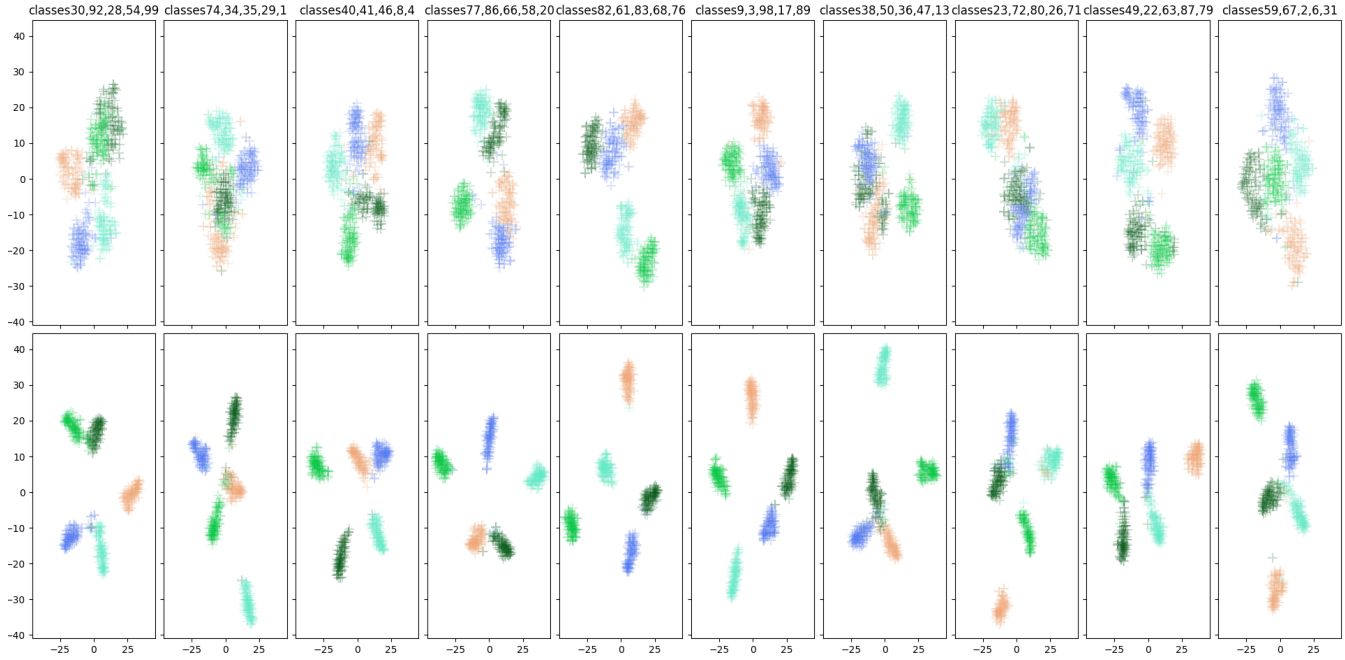

Fig. 3. A scatter plot of the feature points in the last layer in WRN-28 trained on CIFAR-100. Embedding into 2D was performed using t-SNE. Upper panel: Feature points using the baseline regularization scheme. Each plot indicates five classes randomly selected from 100 classes. Lower panel: Feature points of the same five classes using the PMM regularization scheme.

TABLE I
COMPARING ACCURACY RESULTS WITH PMM. THE ACCURACY WAS MEASURED USING THE TOP-1 CRITERION FOR CIFAR10/100 DATASETS.

Model	Dataset	Baseline	PMM
ResNet-44 [28]	CIFAR10	93.22%	**93.83%**
VGG [34]	CIFAR10	93.19%	**93.34%**
WRN-28-10+auto-augment+cutout [29]	CIFAR100	82.51%	**83.52%**
VGG+auto-augment+cutout	CIFAR100	73.93%	**74.19%**
MobileNet [32]	ImageNet	71.17%	**71.44%**

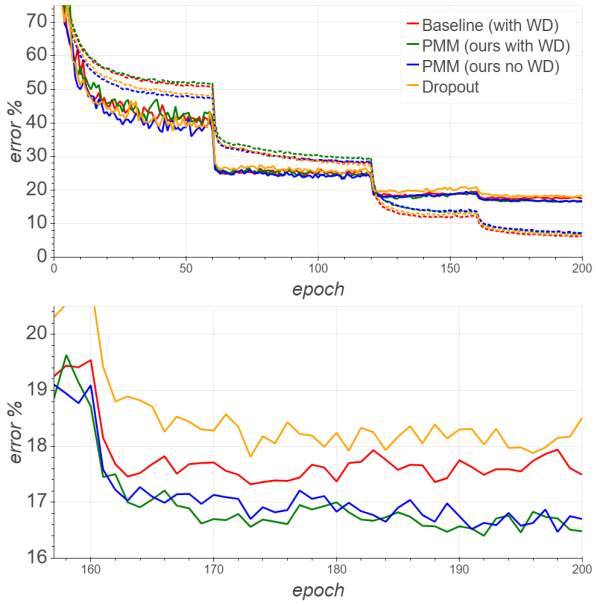

Fig. 4. Training (dashed) and validation errors of CIFAR100 using the WRN28-10 neural network and comparing baseline training and our PMM approach. We use linear scale α, starting with 1e-5 up to 1e-3.

B. Natural Language Classification Tasks

To challenge our premise that we could achieve a higher accuracy score, we tested our PMM on an NLP-related model and datasets. In particular, we used the $\text{BERT}_{\text{BASE}}$ model [35] with 12 transformer layers, a hidden dimensional size of 768 and 12 self-attention heads. Fine-tuning was performed using the Adam optimizer as in the pre-training, with a dropout probability of 0.1 on all layers. Additionally, we used an LR of 2e-5 over three epochs in total for all the tasks. We used the original WordPiece embeddings [36] with a 30k token vocabulary. For our method, similarly to the image classification task, we also used the α factor in the objective function, and found, via a grid search, $\alpha =$ 1e-5 to be the optimal value [3].

We performed experiments on a variety of supervised tasks, specifically by applying a downstream task of fine-tuning natural language inference, semantic similarity, and text classification. All these tasks are available as part of the GLUE multitask benchmark [22].

a) Natural Language Inference: The task of natural language inference (NLI) or recognizing textual entailment means that when a pair of sentences are given, the classifier decides whether or not they contradict each other. Although

[3]We applied $\alpha =$ 1e-6 only to evaluate our method's accuracy with the mismatched MNLI.

there has been a lot of progress, the task remains challenging due to the presence of a wide variety of phenomena such as lexical entailment, coreference, and lexical and syntactic ambiguity. We evaluate our scheme on three NLI datasets taken from different sources, including transcribed speech, popular fiction, and government reports (MNLI), Wikipedia articles (QNLI) and news articles (RTE).

As shown in Table II, our PMM scheme outperformed baseline results on all three tasks. Specifically, on RTE we obtained an absolute improvement of nearly 1.5% (from 68.23% accuracy to 69.67%) with respect to the baseline score.

b) Semantic Similarity: This task involves predicting whether two sentences are semantically equivalent by identifying similar concepts in both sentences. It can be challenging for a language model to recognize syntactic and morphological ambiguity as well as compare the same ideas using different expressions or the other way around. We evaluated our approach on QQP and MRPC downstream tasks, outperforming baseline results as can be seen in Table II. On MRPC in particular, we achieved a 0.75% improvement over the baseline, which is a relative change of more than 8%.

c) Text Classification: Lastly, we evaluated our method on the Stanford Sentiment Treebank (SST-2), which is a binary single-sentence classification task consisting of sentences extracted from movie reviews with human annotations regarding their sentiment. Here too, our approach outperformed the baseline by a small increase in the accuracy.

Overall, applying PMM boosted the accuracy in all the reported tasks, indicating that our approach works well for different tasks from various domains.

TABLE II
COMPARING ACCURACY RESULTS - PMM VS. THE BASELINE. F1 SCORES ARE REPORTED FOR QQP AND MRPC. FOR MNLI, WE REPORT THE AVERAGE OF THE MATCHED (WITH $\alpha =$ 1E-5) AND MISMATCHED SUBSETS (WITH $\alpha =$ 1E-6) FOR BOTH, THE BASELINE AND OUR PMM.

Model	Dataset	Baseline	PMM
$\text{BERT}_{\text{BASE}}$ [35]	MNLI	84.5%	**84.70**%
	QNLI	91.06%	**91.48**%
	RTE	68.23%	**69.67**%
	QQP	87.9%	**88.04**%
	MRPC	90.68%	**91.43**%
	SST-2	92.08%	**92.43**%

V. DISCUSSION

We studied a multi-class margin analysis for DNNs and used it to devise a novel regularization term we call *Pairwise Margin Maximization* (PMM). The PMM term aims at increasing the margin induced by the classifiers, and it is derived directly, for each sample, from the true class and its most competitive class. Note that the standard weight decay or L_2 norm regularization scheme aims at maximizing the margins along the one-vs-all decision boundaries. In contrast, the PMM strive to maximize the margin along the one-vs-one decision boundaries, and we argue that this is the preferred multi-class scheme.

Another difference between PMM and common regularization terms is that PMM is scaled by $\|\phi_{max}\|$, which is the maximal norm of the samples in the feature space. This ensures a meaningful increase in the margin that is not induced by a simple scaling of the feature space. Lastly, since the PMM term is added to each sample, PMM is formulated and performed over the margin distribution to compensate for class imbalance in the regularization term. PMM can be incorporated with any loss, i.e., is not restricted to hinge loss or cross-entropy losses. Using PMM, we were able to demonstrate improved accuracy over a set of experiments in images and text.

Similarly to [15], PMM can be implemented at other layers in the deep architecture. This enables maximal margins that directly impact training at all levels. The additional computation associated with such a framework makes it less appealing from an efficiency perspective, which may be compensated by the gain in accuracy. The design of such additional PMM terms is left for further study.

REFERENCES

[1] A. Azulay and Y. Weiss, "Why do deep convolutional networks generalize so poorly to small image transformations?" *arXiv preprint arXiv:1805.12177*, 2018.

[2] T. Zhang and Z.-H. Zhou, "Multi-class optimal margin distribution machine," in *International Conference on Machine Learning*, 2017, pp. 4063–4071.

[3] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. J. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," *arXiv preprint arXiv:1312.6199*, 2013.

[4] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *arXiv preprint arXiv:1412.6572*, 2014.

[5] O. Bousquet and A. Elisseeff, "Algorithmic stability and generalization performance," in *Advances in Neural Information Processing Systems*, 2001, pp. 196–202.

[6] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

[7] R. O. Duda, P. E. Hart *et al.*, *Pattern classification and scene analysis*. Wiley New York, 1973.

[8] V. Vapnik, *Statistical learning theory*. Wiley New York, 1998.

[9] K. Crammer and Y. Singer, "Ultraconservative online algorithms for multiclass problems," *Journal of Machine Learning Research*, vol. 3, no. Jan, pp. 951–991, 2003.

[10] R. E. Schapire, Y. Freund, P. Bartlett, W. S. Lee *et al.*, "Boosting the margin: A new explanation for the effectiveness of voting methods," *The annals of statistics*, vol. 26, no. 5, pp. 1651–1686, 1998.

[11] K. Crammer and Y. Singer, "On the algorithmic implementation of multiclass kernel-based vector machines," *Journal of machine learning research*, vol. 2, no. Dec, pp. 265–292, 2001.

[12] D. Soudry, E. Hoffer, M. S. Nacson, S. Gunasekar, and N. Srebro, "The implicit bias of gradient descent on separable data," *The Journal of Machine Learning Research*, vol. 19, no. 1, pp. 2822–2878, 2018.

[13] S. Sun, W. Chen, L. Wang, and T. Liu, "Large margin deep neural networks: Theory and algorithms," *ArXiv*, vol. abs/1506.05232, 2015.

[14] G. Elsayed, D. Krishnan, H. Mobahi, K. Regan, and S. Bengio, "Large margin deep networks for classification," in *Advances in neural information processing systems*, 2018, pp. 842–852.

[15] Y. Jiang, D. Krishnan, H. Mobahi, and S. Bengio, "Predicting the generalization gap in deep networks with margin distributions," in *7th International Conference on Learning Representations, ICLR*, 2019. [Online]. Available: https://openreview.net/pdf?id=HJlQfnCqKX

[16] A. Garg, S. Har-Peled, and D. Roth, "On generalization bounds, projection profile, and margin distribution," in *ICML*, 2002, pp. 171–178.

[17] J. Sokolić, R. Giryes, G. Sapiro, and M. R. Rodrigues, "Robust large margin deep neural networks," *IEEE Transactions on Signal Processing*, vol. 65, no. 16, pp. 4265–4280, 2017.

[18] T. Zhang and Z.-H. Zhou, "Optimal margin distribution machine," *IEEE Transactions on Knowledge and Data Engineering*, vol. 32, no. 6, pp. 1143–1156, 2019.

[19] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne." *Journal of machine learning research*, vol. 9, no. 11, 2008.

[20] A. Krizhevsky, G. Hinton *et al.*, "Learning multiple layers of features from tiny images," Citeseer, Tech. Rep., 2009.

[21] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database," in *2009 IEEE conference on computer vision and pattern recognition*, 2009, pp. 248–255.

[22] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. R. Bowman, "Glue: A multi-task benchmark and analysis platform for natural language understanding," *arXiv preprint arXiv:1804.07461*, 2018.

[23] A. Williams, N. Nangia, and S. R. Bowman, "A broad-coverage challenge corpus for sentence understanding through inference," *arXiv preprint arXiv:1704.05426*, 2017.

[24] L. Bentivogli, P. Clark, I. Dagan, and D. Giampiccolo, "The fifth pascal recognizing textual entailment challenge." in *TAC*, 2009.

[25] W. B. Dolan and C. Brockett, "Automatically constructing a corpus of sentential paraphrases," in *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*, 2005.

[26] Z. Chen, H. Zhang, X. Zhang, and L. Zhao, "Quora question pairs," 2018.

[27] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Y. Ng, and C. Potts, "Recursive deep models for semantic compositionality over a sentiment treebank," in *Proceedings of the 2013 conference on empirical methods in natural language processing*, 2013, pp. 1631–1642.

[28] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[29] S. Zagoruyko and N. Komodakis, "Wide residual networks," *arXiv preprint arXiv:1605.07146*, 2016.

[30] T. DeVries and G. W. Taylor, "Improved regularization of convolutional neural networks with cutout," *arXiv preprint arXiv:1708.04552*, 2017.

[31] E. D. Cubuk, B. Zoph, D. Mane, V. Vasudevan, and Q. V. Le, "Autoaugment: Learning augmentation policies from data," *arXiv preprint arXiv:1805.09501*, 2018.

[32] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.

[33] P. Goyal, P. Dollár, R. Girshick, P. Noordhuis, L. Wesolowski, A. Kyrola, A. Tulloch, Y. Jia, and K. He, "Accurate, large minibatch sgd: Training imagenet in 1 hour," *arXiv preprint arXiv:1706.02677*, 2017.

[34] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[35] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[36] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey *et al.*, "Google's neural machine translation system: Bridging the gap between human and machine translation," *arXiv preprint arXiv:1609.08144*, 2016.