הבינתחומי בית ספר אפי ארזי
הרצליה למדעי המחשב

The Interdiciplinary Center, Herzliya
Efi Arazi School of Computer Science
M.Sc. Program - Research Track

Selective sampling and Margin-Based Regularization in Deep Neural Networks

by
Berry Weinstein

M.Sc. dissertation, submitted in partial fulfillment of the requirements
for the M.Sc. degree, research track, School of Computer Science
The Interdisciplinary Center, Herzliya

October, 2020

This work was carried out under the supervision of Dr. Shai Fine and Prof. Yacov Hel-Or from the Efi Arazi School of Computer Science, The Interdiciplinary Center, Herzliya.

Abstract

In machine learning, selective sampling strategies are inspired by the field of active learning. While the objective of the former is to reduce the number of annotations the latter strives to reduce computation. In this work we elaborate on different active learning approaches and derive one of our main contributions, a selective sampling method designed to accelerate the training of deep neural networks. To this end, we introduce a novel measure, the minimal margin score (MMS), which measures the minimal amount of displacement an input should undergo until its predicted classification is switched. For multi-class linear classification, the MMS measure is a natural generalization of the margin-based selection criterion, which has been thoroughly studied in the binary classification setting. We demonstrate empirically that when training commonly used deep neural network architectures for popular image classification tasks with our method, there is a marked and substantial acceleration in the training process. The efficiency of our method is compared against standard training procedures and against commonly used selective sampling alternative, specifically hard negative mining selection. We demonstrate a substantial speedup via an aggressive learning-rate regime while using the MMS selective sampling method.

Using the same concept, we also derive a new margin-based regularization formulation, termed *multi-margin regularization* (MMR), for *deep neural networks* (DNNs). The MMR is inspired by principles that were applied in margin analysis of shallow linear classifiers, e.g., *support vector machine* (SVM). Unlike SVM, MMR is continuously scaled by the radius of the bounding sphere (i.e., the maximal norm of the feature vector in the data), which is constantly changing during training. We empirically demonstrate that by a simple supplement to the loss function, our method achieves better results on various classification tasks across domains.

Contents

1 Introduction

The general purpose of this study is to address the existing ideas of *active leaning*, and specifically *selective sampling*, and use them to develop methods that will serve to minimize the cost of training deep leaning models. Section 1.1 will provide an overview of the former ideas, section 1.2 will present the deep learning algorithm and the rest of this thesis will focus on a novel approach to address the heavy cost of training these models, the theory behind it and experimental results to support these findings. Specifically, we suggest a novel method to select for the back-propagation pass only those instances that accelerate the training convergence of the deep neural network, thus speeding up the entire training process. Finally, the last section will discuss on the insights gained from this work and the direction for further study.

1.1 Active learning

Active learning is a sub-field of machine learning that studies the idea of a learner selecting actions or making queries that influence what data points are added to its training set. Mainly, the purpose is to use as few annotations as possible in order to perform the same (or better) as if using the entire data set. This is a desirable property for any supervised learning system since label instances is very difficult, time-consuming, or expensive to obtain. One example would be text classification that are taken from knowledge domains such as biology or medicine. The annotation of such documents requires the time of biologists and doctors which can be very expansive. Another example is from the domain of speech recognition; Accurate labeling of speech vocal sounds in the shape of words or phonemes is extremely time consuming and requires trained linguists. It has been shown [1] that annotating words takes ten times longer than the actual audio recording while phonemes can take up to 400 times longer.

Active learning systems main goal is to overcome labeling bottleneck by choosing which instances to label next in order to achieve high accuracy using as few labels as possible, thereby minimizing the cost of the entire training process. There are a few scenarios in which the active learner may be able to ask queries to obtain instance label as well as a many strategies of how to evaluate the information gain of unlabeled instances. Following are the main query settings:

1.1.1 Membership Query Synthesis

In this setting the active learner may request labels for unlabeled samples (make queries) in the input space. Typically the queries are being generated by the learner rather than sampling from some input distribution space [2]. Query synthesis can be achieved by various of techniques, one can think of more sophisticated techniques as Generative Adversarial Networks (GANs) in the vision domain or other computer vision techniques. In the Natural Language Processing (NLP), the generation is usually done by using an auto-regressive decoder that predicts the next word or phoneme given the previous ones. A practical example of query synthesis was employed to train a neural network that classifies handwritten characters [3] when the oracle is a human annotator. They actually encountered the problem of generating characters with no actual meaning for a human oracle. A more innovative example of of the membership query scenario was demonstrated with the employment of the ”robot scientists” [4]. which can execute a series of autonomous biological experiments to discover metabolic pathways in some kind of yeast. In this domain, the labels come from the results of conducted experiments rather than human annotators. This is a much promising direction for the underlying query setting making scientific discoveries.

1.1.2 Selective sampling

There are two approaches for selective sampling schemes, the stream-based [5] and the pool-based [6]. The difference between them is that the former scans through the data sequentially and makes query decisions individually, whereas the latter evaluates and ranks the entire collection before selecting the best samples to query. Moreover, the instances subject to query are being sampled out of an actual distribution and the there is no cost for obtaining an unlabeled sample. The decision whether or not to query an instance is a subject to the information gain measure. One approach is to compute explicitly the instance region of uncertainty [5] and choosing the samples of which the leaner is still uncertain about. One naive way of doing this is to set a threshold on an evaluation score and choose to label instances whose evaluation is under this threshold. A more principled approach is to look at the *version space* [7], i.e., the set of hypotheses consistent with the previous selected training set. The selection in this case rely on instances on which two models of the same model class but with different set of parameters, disagree. Updating the version space after each query is computationally expensive and require the use of approximations [8]. There are many more query strategies to evaluate the information gain of unlabeled samples. Few of them will be discussed in the remainder of this subsection.

1.1.3 Uncertainty sampling

One common strategy to query samples is *uncertainty sampling* [6]. In this scheme, the active learner queries the samples in which it is least certain about its label. In their binary classification model, the general principle is to query labels for samples whose posterior probability is closest to 50% of being right. Alternately, uncertainty sampling uses the *entropy* [9] as a measure of confidence. Entropy is a measure of the expected amount of information to encode samples drawn from some distribution. The entropy is given by:

$$H(x) = -\sum P(y_i|x;\theta)\log P(y_i|x;\theta)$$

Where y_i ranges over the possible labels and θ are the network parameters. High entropy means that the model predicted similar score for all existing labels. As such, the model has a very low confidence of the true label and these are the samples we are interested to query for their label. A common use of this strategy was introduced by applying it to *support vector machine* (SVM) [10] linear model. In this setting the samples to query for a label are those who are the closest to the decision boundary, (the support vectors).

1.1.4 Query by committee

Another selection strategy and probably one of the common algorithms for online selective sampling strategies is the *query by committee* [11] (QBC). The idea is to maintain a set of same class models representing competing hypothesis, called a committee, and that are trained on the same dataset. Each committee member is responsible to label a subject sample, and the samples to be selected and considered as the most informative are the ones the committee most disagree. The main premise of this algorithm is to minimize the version space (as discussed in subsection 1.1.2). As different machine learning algorithms goal is to find the best model within the version space, the QBC constrain the size of this space by querying in controversial regions of the input space, allowing a more precise search with as few labeled samples as possible.

[11] found that the theoretical bound for the number of queries to label that the algorithm will make is $O(log(1/\varepsilon))$ where ε is an upper bound on error. Note that in passive learning the size of the sample needed for learning is $O(1/\varepsilon)$. Hence

in terms of the error ε there is an exponential reduction in the number of labels needed for training. While a naïve implementation of QBC in real world tasks is not feasible due to the complexity of holding a representation of the version space, the kernel QBC (KQBC) [12] was shown to be very useful. This applicable algorithm does not only reduce the complexity of the version space by holding a subset of it in a compact manner, but it also addresses problems that are not essentially linear – using the kernel representation as in the SVM algorithm. An early stage experiments where performed to test this strategy (see Figure 2 and paragraph 3).

1.2 Deep neural networks

Deep neural networks (DNNs) are being widely used for classification tasks. In particular, Convolutional neural networks (CNNs), consist of layers of convolutions and non-linear activations, being the state of the art (SOTA) in image classification [13]. ResNet [14] is one of the popular CNN architectures. This network consists of many stacked convolutional layers followed by batch normalization (BN) [15] and a rectified linear activation units (ReLU) [16], as well-as shortcut connections that perform identity mapping. The main purpose of these feed forward stacked layers is to embed the raw images into a separable feature space while the last part is added on top of these layers to form a linear classifier using a fully connected layer that maps the embedded features into the desired class. It has been shown that residual neural networks achieve SOTA accuracy in typical classification benchmarks like CIFAR10, CIFAR100 [17] and ImageNet [18].

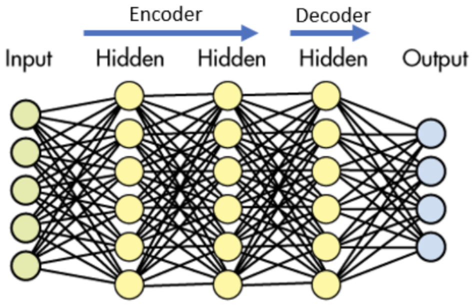


Figure 1: An illustration of a general CNN encoder and decoder.

Training these networks is mainly performed using stochastic gradient descent

(SGD) algorithm [19], which calculates the network classification error on a set of mini-batch examples. The error is calculated with respect to some loss function, while the network parameters are adjusted towards the direction such that all examples will be correctly classified. Each such step is performed via large scale back-propagation [20] while a single pass over the entire dataset is called an epoch.

In contrast to convex and separable problems which can be addressed using the previous query settings, training CNNs is neither convex nor separable during the optimization process. We will examine different selection methods in the convex separable spaces and build heuristics and infrastructure that hold in the non-convex and non-separable spaces of CNNs. For simplicity, we will refer to the first part of stacked layers of the network as the encoder and the last fully connected layer as the decoder (Figure 1). The idea we rely on, is that during the network training, the encoder part compresses the raw images such that they become more and more separable. On the other hand, the decoder is a classifier that resides in a version space of all the linear classifiers, thus we can make linear decision boundary assumptions on its input features.

Over the last decade, deep neural networks have become the machine learning method of choice in a variety of application domains, demonstrating outstanding, often close to human-level, performances in a variety of tasks. Much of this tremendous success should be attributed to the availability of resources; a massive amount of data and compute power, which in turn fueled the impressive and innovative algorithmic and modeling development. However, resources, although available, come with a price. Data in the big data era is available, but reliable labeled data is always a challenge, and so are the ETL (Extract-Transform-Load) processes, data transfer, and storage. With the introduction of GPUs, compute power is readily available, making the training of deep architectures feasible.

1.2.1 Combining active learning and deep learning

In contrast to the previous section approaches presented, that rely on a single sample acquisition, to train deep neural networks we have to select a batch of samples with many images at once. One active learning approach is to maintain a pool of labeled samples and by using one of the various acquisition functions, adding more labeled samples to the training data pool. For image classification tasks, several acquisition functions can be used, such as selecting the samples that maximize the predictive entropy, as discussed in section 1.1.3, and selecting the set of samples that are expected to maximize the information gained about the model parameters, i.e. maximize the mutual information between predictions and model posterior [21]. With this approach, the model will be optimized using least amount of oracle queries for label and achieving and on par accuracy results. Different approach would be to take advantage of the model training properties and to use active learning to acquire the ”best” batch of samples at every given training step. This approach will use the entire dataset but in a more educated form, allowing to achieve the same accuracy but faster.

1.2.2 Selective sampling for accelerating training of deep neural networks

The training phase, which to a large extent, relies on stochastic gradient descent methods, requires a large number of computational resources as well as a substantial amount of time. A closer look at the compute processes highlights the fact that there is a significant difference in the compute effort between the inference (forward pass) and the model update (back-propagation) where the latter being far more demanding. The implication is evidenced by the performance charts that hardware manufactures publish, where performance matrices such as throughput (e.g. image per second) are up to 10x better at inference vs. training for popular deep neural network architectures [22].

The main contribution of this work addresses the computing challenge. Specifically, we suggest a novel method to select for the back-propagation pass only those instances that accelerate the training convergence of the deep neural network, thus speeding up the entire training process. The selection process is continuously performed throughout the training process at each step and in every training epoch. Our selection criterion is based on computations that are calculated anyhow as an integral part of the forward pass, thus taking advantage of the ”cheaper” inference compute.

1.2.3 Margin-based Regularization in Deep Neural Networks

Despite their success, some researchers have shown that neural networks can generalize poorly even with small data transformations [23] as well as overfit to arbitrarily corrupted data [24]. Additionally, problems such as adversarial examples [25, 26], which cause neural networks to misclassify slightly perturbed input data, can be a source of concern in real-world deployment of models. These challenges raise the question as to whether properties that enabled classical machine learning algorithms to overcome these problems can be useful in helping DNNs resolve similar problems. Specifically, [27] introduced margin theory to explain boosting resistance to over-fitting.

Furthermore, the large 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 [10], as well as robustness for input perturbations [28] on unseen data. Can the application of the large margin principle in DNNs lead to similar results?

Although computation of the actual margin in the input space of DNNs is intractable, studies show that the widely used cross-entropy loss function is by itself a proxy for converging to the maximal margin [29]. To date, this was only demonstrated for linear models that, similarly to SVM, have a theoretical guarantee for maximal margin convergence [30]. No such assurance for non-linear DNNs, their being being highly non-convex, has been offered.

Recently, [31] developed a measure for predicting the generalization gap [1] in DNNs that leverages *margin distribution* [2] [32] as a more robust assessment for the margin notion in DNNs. In their work, they also point out that this measure can be used as an auxiliary loss function to achieve better generalization.

In this work, we extend our contribution using the aforementioned ideas and present a novel regularization term, which we denote as *Multi-Margin Regularization* (MMR), that can be added to any existing loss function in DNNs.

2 Previous Works

The prominent work of Gal et al. [33] and Andreas et al. [34] which combined active learning and DNNs in terms of using at least labeling as possible, is an extension of *Bayesian Active Learning by Disagreement* (BALD) [21]. Basically, to acquire a batch of examples to label, they used a tractable approximation to the mutual information between a batch of points and model parameters. Intuitively, they captured how strongly the model prediction for a given data point and the model parameters are coupled, implying that the model has many possible ways to explain the data. To this end, they used *Bayesian Neural Network* (BNN) as their inference model and *Monte Carlo* (MC) dropout [35] as a stochastic regularisation technique to perform the approximate inference in the BNN model.

On the other hand, accelerating the training process is a long-standing challenge that was already addressed by quite a few authors. A common approach is

[1] The difference in accuracy between training and testing performance.

[2] The distribution of distances to the *decision boundaries*.

to increase the batch size, thus mitigating the inherent time load. This approach represents a delicate balance between available compute ingredients (e.g. memory size, bandwidth, and compute elements). Interestingly, increasing the batch size not only impacts the computational burden but may also impact the final accuracy of the model [36, 37, 38]. [39] is an additional well-known work for faster training by finding the optimal hyper parameters at the course of training. It leveraged advanced mechanisms for adjusting the learning rate regime during the training. They used cyclical learning rates (CLR), first suggested by Smith et el. [40]. In this regime, they decide the boundaries of the learning rates at the beginning and the end of the training and the policy of how much to decease the learning rate at each step. They also derive a simplification of the Hessian Free optimization method [41] to compute an estimate of the optimal learning rates.

Sample selection is another approach that has been suggested to accelerate the training. The most notable one is probably the hard negative mining [42] where samples are selected by their loss values. The underlying assumption is that samples with higher losses have a significant impact on the model. Most of the previous work that utilized this approach was mainly aimed at increasing the model accuracy, but the same approach can also be used to accelerate training. Recent works employ selection schemes that examine the importance of the samples [43, 44]. During the training, the samples are selected based on their gradient norm, which in turn leads to a variance reduction in the stochastic gradients. Inspired by the batch size approach, a recent work by Katharopoulos and Fleuret [45] uses selective sampling to choose the training samples that reduce the gradient variance, rather than increasing the size of the batch.

Our work is inspired by the *active learning* paradigm that utilizes selective sampling to choose the most useful examples for training. In active learning, the goal is to reduce the cost of labeling the training data by querying labels for only carefully selected examples. Thus, unlike the common supervised learning setting, where training data is randomly selected, in active learning, the learner is given the power to ask questions, e.g. to select the most valuable examples to query for their labels. This is an advantage in deep neural networks, where the selection is made after the forward pass, allowing to select from a much larger batch of examples than those that are being used for back-propagation. This is due to the large gap in computation between the forward and the backward passes, while the latter takes much more time to perform. Measuring the training value of examples is a subject of intensive research, and quite a few selection criteria have been proposed. The approach most related to our work is the *uncertainty sampling* [6], where samples are selected based on the uncertainty of their predict labels. Two heavily used approaches to measure uncertainty are entropy-based and margin-based [46]. In the entropy-based approach [47], uncertainty is measured by the entropy of the posterior probability distribution of the labels, given the sample. Thus, a higher entropy represents higher uncertainty with respect to the class label. This approach naturally handles both binary and multi-class classification settings, but it relies on an accurate estimate of the (predicted) posterior probabilities. In the margin-based approach [48, 49], uncertainty is measured by the distance of the samples from the decision boundary. For linear classifiers, several works [50, 51] gave theoretical bounds for the exponential improvement in computational complexity by selecting as few labels as possible. The idea is to label samples that reduce the *version space* (a set of classifiers consistent with the samples labeled so far) to the point where it has a diameter at most ε (c.f [52]). This approach was proven to be useful also in non-realizable cases [51], where the learner's hypothesis class is not assumed to contain a target function that perfectly classifies all training and test examples. However, generalizing it to the multi-class setting is less obvious. Another challenge in adapting this approach for deep learning is how to measure the distance to the intractable decision boundary. Ducoffe and Precioso [53] approximate the distance to the decision boundary using the distance to the nearest adversarial examples. The adversarial examples are generated using a Deep-Fool algorithm [54]. The suggested DeepFool Active Learning method (DFAL) labels both, the unlabeled samples and the adversarial counterparts, with the same label.

Our selection method also utilizes uncertainty sampling, where the selection criterion is the closeness to the decision boundary. We do, however, consider the decision boundaries at the (last) fully-connected layer, i.e. a multi-class linear classification setting. To this aim, we introduce the *minimal margin score* (MMS), which measures the distance to the decision boundary of the two most competing predicted labels. This MMS serves us as a measure to score the assigned examples. A similar measure was suggested by Jiang et al. [55] as a loss function and a measure to predict the generalization gap of the network. Jiang et al. used their measure in a supervised learning setting and applied it to all layers. In contrast, we apply this measure only at the last layer, taking advantage of the linearity of the decision boundaries. Moreover, we use it for selective sampling, based solely on the assigned scores, namely without the knowledge of the true labels. The MMS measure can also be viewed as an approximation measure for the amount of perturbation needed to cross the decision boundary. Unlike the DFAL algorithm, we are not generating additional (adversarial) examples to approximate this distance but rather calculate it based on the scores of the last-layer.

Although our selective sampling method is founded by active learning principles, the objective is different. Rather than reducing the cost of labelling, our goal is to accelerate the training. Therefore, we are more aggressive in the selection of the examples to form a batch group at each learning step, at the cost of selecting many examples at the course of training.

The large margin principle has proven to be fundamentally important in the history of machine learning. While most of the efforts revolved around binary classification, extensions to multi-class classification were also suggested, e.g., multi-class perceptron (see Kesler's construction, [56]), multi-class SVM [57] and multi-class margin distribution [24]. Margin analysis have also been shown to correlate with better generalization properties [27]. Of particular interest to our study is 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 [58].

Computing the actual margin in DNNs, though, is intractable. [29] proved that cross-entropy loss in linear DNNs, together with *stochastic gradient descent* (SGD) optimization, converges to a maximal margin solution, but it cannot ensure a maximal margin solution in nonlinear DNNs. [59] affirmed that cross-entropy alone is not enough to achieve the maximal margin in DNNs and that an additional regularization term is needed.

Several works addressed the large margin principle in DNNs. [60] presented a multi-class linear approximation of the margin as an alternative loss function. They applied their margin-based loss at each and every layer of the neural network. Moreover, their method required 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, [31] 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 statistics of the marginal distribution [32]. [61] used the input layer to approximate the margin via the Jacobian matrix of the network 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 improvements.

Furthermore, [62] derive and analyze three variants of margin-based algorithms, each of which address different prediction task. They start their derivation from the *hinge loss* in the binary setting and formulate novel update rules for each of the algorithm variants. Motivated by the work of [63], their new update rule for model weights at each step, formulate the trade-off between the amount of progress made on each step and the amount of information retained from previous step.

The rest of the study is organized as follows. The following section describes a few preliminary explorations that put focus on selective sampling, using the decision boundary query settings discussed in section 1.1. Here we also present the affects of applying these concepts on deep neural networks, laying the groundwork to the main contribution of this paper. In section 4, we present the MMS measure and describe our selective sampling algorithm and discuss its properties. Later, in section 5 we derive a margin-based regularization term that we add to the loss function and show improvement in model accuracy. In Section 6 we present the performances of the MMS algorithm on the common datasets CIFAR10 and CIFAR100 [17] and compare results against the original training algorithm and hard-negative sampling. We demonstrate additional speedup when we adopt a more aggressive learning-drop regime. We present conclusions at Section 7 with a discussion and suggestions for further research. Lastly, we show that applying our novel Multi-Margin Regularization (MMR) on the output layer of various deep neural networks, on different classification task and from different domains, is sufficient to obtain a substantial improvement in accuracy. In particular, we achieve valuable accuracy improvement in image and numerous text classification tasks, including CIFAR10,CIFAR100, ImageNet, MNLI, QQP and more.

3 Preliminary studies

KQBC vs SVM selection. In this preliminary study we recreated the KQBC schemes on a synthetic as well as on the MNIST [64] dataset. This study was conducted to explore the applications of selecting samples which are closer to the margin, later will be used on top of a CNN classifier but instead of KQBC, we will use SVM. Here we consider the KQBC algorithm as presented by [12] in comparison with the vanilla SVM algorithm. First, we used their synthetic dataset, where the examples are normally distributed with $N(\mu = 0, \Sigma = I_d)$ and the target classifier is the vector $\omega* = (1,0,0,\dots,0)$, thus the label of an instance is the sign of its first coordinate. Then we apply the same comparison to the MNIST dataset. The MNIST database of handwritten digits, has a training set of 60,000 examples, and a test set of 10,000 examples. The digits have been size-normalized and centered to a fixed-size image. For this experiment we degenerated the dataset into two classes (1, -1), which are one of the digit vs all the rest. We balance between the two classes by eliminating most of the "rest" class examples. The final data was a sub version of the original one, consisting of 1000 training examples for each class, so 2000 in total, and 200 for each test class in the same manner. The results are presented in Figure 2. In both settings we expect that the KQBC approach will accelerate the training and generalize at least as good as the vanilla SVM. As shown above, this is indeed the case as in the synthetic dataset setting. We can clearly see the exponential boost, while in the MNIST dataset setting we still show a clear boost in fitting the data.

3.1 Selective sampling using SVM on a binary objective CNN

We first lay the background to the main purpose of this work, which is to achieve faster training of CNNs by applying SVM selection methods on the training data and by adjusting the multi class objective of the CNN model into a binary task. Instead of using the vanilla classification layer which projects the embedded features into a sub space of the size of the number of classes, we replaced it by a learanable projections layer of which its input size remains the same but the output replaced by two neurons. Additionally, the *Cross Entropy* (CE) loss function was replaced with *Binary Hinge Loss* or *Binary Squared Hinge Loss*. Then, we applied an SVM classifier to the features that were generated by the encoder along with the target labels, we extracted the support vectors and used the support points to train the model. A reasonable way to describe this model in terms of SVM algorithm is to consider the encoder part of the CNN as a dynamic SVM kernel representation, which transforms the non separable input images into linearly separable features. Although originally the SVM kernel is a static function, in our case it is changing during the optimization process of the CNN.

To evaluate empirically the suggested method we used Cifar10 (Section 6.1) and ResNet-44 [14] architecture and compared the SVM selection scheme against the baseline and hard-negative mining (HNM) as suggested by [65]. We applied the original hyper-parameters and training regime using batch-size of 64. For the SVM selection we started from 10 times larger batch (640 samples) and selected 64 samples from it whose are closest to the boundary. In addition, we used the original augmentation policy as described in [14]. Additionally, we split the dataset into two classes instead of the vanilla ten simply by defining five out of ten classes as one homogeneous class and the rest will be the second homogeneous class. In Figure 3 we demonstrated the accuracy achieved using the binary

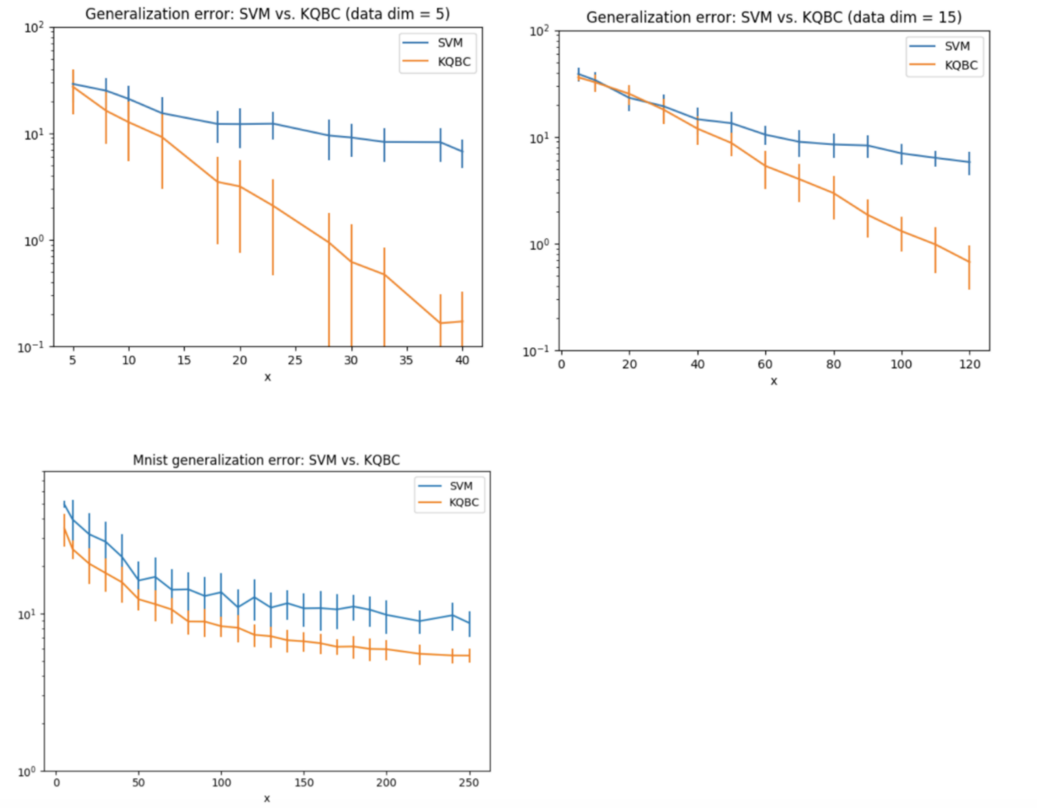


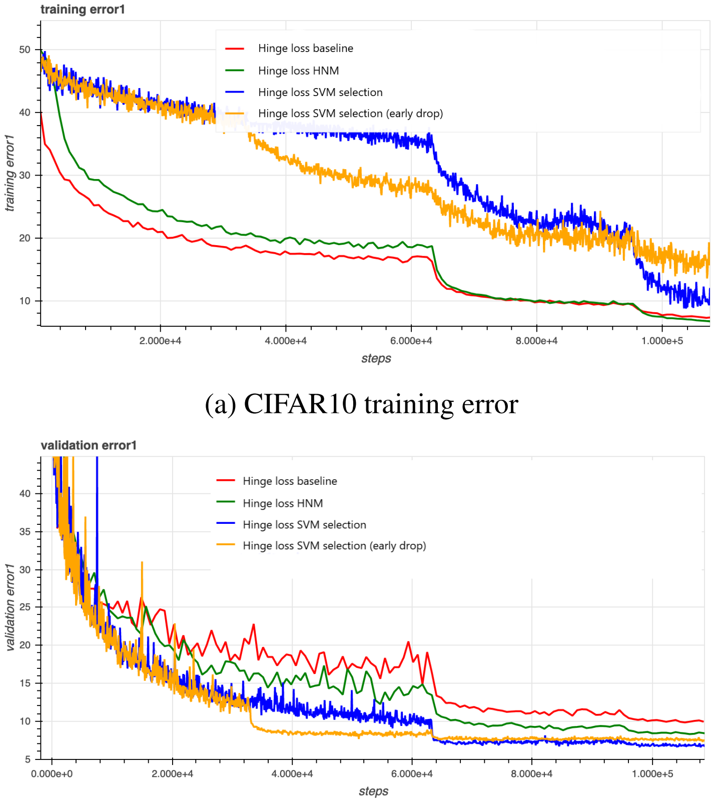
Figure 2: The generalization error vs number of examples on a synthetic data (normal distribution) comparison between SVM classifier and KQBC (two upper plots). The generalization error on MNIST dataset when converting it into one-vs-rest class problem (lower left plot).

setup and the hinge loss. As can be seen, the classification top1 training error decreased in a slower pace when using SVM selection method rather the the baseline (blue line v.s. the red line). This can be explained by the fact that in the selection scheme, we choose examples that are the most uncertain to the model, thus the classification error is bigger. However, we observe that the test error dropped much faster and achieved much lower error all the way to the final training step. Intuitively, the examples that were closest to the boundary (support vectors) forced a bigger margin and a greater class certainty, gives a better classification rate. Although training with hinge loss does not generalize as well as training with BCE loss, the gap between the vanilla setting and the SVM selection setting was significant, what gives a good basis to the multi class setting using the original cross entropy loss. Moreover, we observed that the HNM scheme result in a lower error than the baseline (green v.s. red lines). However, SVM selection achieved better accuracy than HNM. Additionally, we performed an early learning rate drop as discussed in section 6.1.1 (yellow line) and as can be seen, it achieved better accuracy than both the baseline and the HNM, even when training half way (yellow line is lower than the red and the green in half of the way). Finally, We conclude that using SVM for selecting the data points to train on, and by that enlarging the margin between the classes is better than random selection and HNM.

3.2 Fixing an orthogonal classification layer

Following the encoder-decoder scheme of the CNN model, the decoder which represents the classification layer (logit), relies on the embedded features to be linearly separated at some point of the training. Furthermore, the idea of selecting samples that are the closest to the decision boundary of each one of the logit classifiers, induces faster convergence upon the encoder by passing it through the back propagation. Subsequently, it seems that performing the training with a predefined decoder in a manner that it would not be updated during the optimization of the network, will force the encoder to align its parameters to produce the features that would correspond with the ”fixed” decoder. A previous work by [66] shows empirically that the final classification layer can be replaced with a predetermined linear transform with little or no loss of accuracy for most tasks. By decreasing model parameters that usually rely on a very large classification layer, less weights update is needed as well as less memory foot-print and even communication overhead, when training with multiple resources.

This concept can be also beneficial for our selection schemes, as we select samples according to its distance to the classifier. We could fix the classifiers in advance such that their distance from each other is maximal. Performing the selection now would enforce the encoder to update its weights to be optimal with respect to the new fixed classifier. To support this theoretical claim we experimented it using Cifar10 and Cifar100 datasets and Resnet-44 and WRN-28-10 architecture, respectively. Instead of using the vanilla classification layer, we fixed it such that each classifier is orthogonal to rest of the classifiers. We then used the MMS selection scheme as will further discussed in the rest of this work. As can be seen in Figure 4, freezing the last layer to be orthogonal gives slightly better final accuracy for the Cifar10 task and is excessively beneficial to the selection

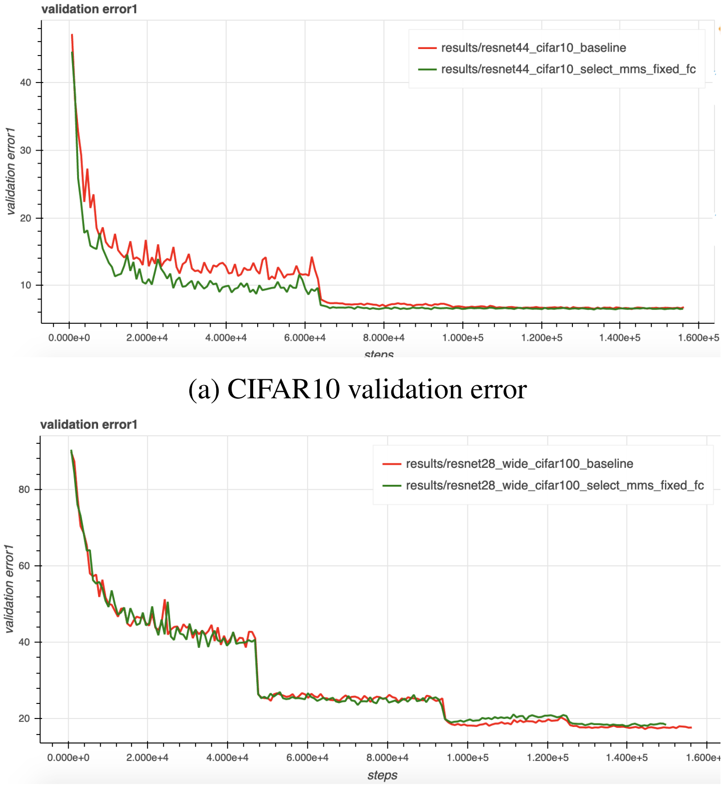


(a) CIFAR10 training error

(b) CIFAR10 validation error

Figure 3: Training and test accuracy (ResNet44, CIFAR10). Comparing vanilla training, HNM-samples selection (hard negative mining), and SVM selection method on the binary setup. **The SVM selection method achieves final test accuracy at a reduces number of training steps for the early drop regime**.

throughout the training (green line). However, for the Cifar100 task we can not see neither improvement in the final accuracy (and it is even slightly worse), nor being beneficial during the course of training. The reason for this behavior might stem from Cifar100 having ten times more classes than Cifar10, allowing a more complex optimum for the classifier than just an orthogonal projection matrix.



(a) CIFAR10 validation error

(b) CIFAR100 validation error

Figure 4: Fixed fully connected compared to vanilla validation accuracy.

4 Accelerating training using Minimal Margin Score selection

The main contribution of this work is based on the evaluation of the minimal amount of displacement a training sample should undergo until its predicted classification is switched. We call this measure *minimal margin score* (MMS). This measure depends on the best and the 2nd best scores achieved for a sample. Our measure was inspired by the margin-based quantity suggested by Jiang et al. [55] for predicting the generalization gap of a given network. However, in our scheme, we apply our measure to the output layer, and we calculate it linearly with respect to the input of the last layer. Additionally, unlike [55], we do not care about the true label, and our measure is calculated based on the best and the 2nd best scores.

An illustrative example, demonstrating the proposed approach, is given in Figure 5. In this example, a multi-class classification problem is composed of three classes: Green, Red, and Blue along with three linear projections: $\mathbf{w}_1, \mathbf{w}_2$, and $\mathbf{w}_3$, respectively. The query point is marked by an empty black circle. The highest scores of the query point are s^1 and s^2 (assuming all biases are 0's), where $s^1 > s^2$ and s^3 is negative (not marked). Since the best two scores are for the Green and Red classes, the distance of the query point to the decision boundary between these two classes is d. The magnitude of d is the MMS of this query point.

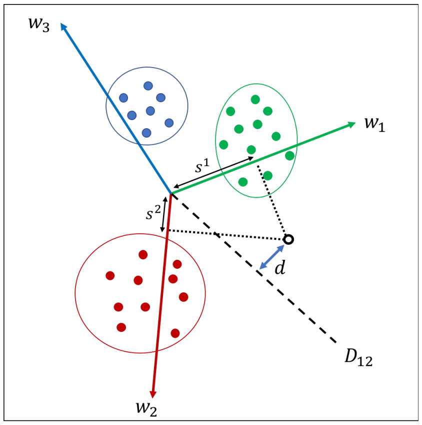


Figure 5: Illustrative example of the MMS measure. For more details see text.

Formally, let $\mathscr{X} = \{\mathbf{x}_1, ..., \mathbf{x}_B\}$ be a large set of samples and $\mathbf{y}_i = F(\mathbf{x}_i; \boldsymbol{\theta}) \in \mathscr{Y}$ be the input to the last layer of the neural network. Assume we have a classification problem with n classes. At the last layer the classifier f consists of n linear functions: $f_i : \mathscr{Y} \rightarrow \mathbb{R}$ for $i = 1 \dots n$ where f_i is a linear mapping $f_i = \mathbf{w}_i^T \mathbf{y} + b_i$. For sample $\mathbf{x}_k \in \mathscr{X}$, the classifier predicts its class label by the maximal score achieved: $\ell_k = \arg\max_i f_i(F(\mathbf{x}_k; \boldsymbol{\theta})) = \arg\max_i f_i(\mathbf{y}_k)$. Denote the sorted scores of $\{f_i(\mathbf{y}_k)\}_{i=1}^n$ by $(s_k^{i_1}, s_k^{i_2}, \cdots, s_k^{i_n})$ where $s_k^{i_j} \geq s_k^{i_{j+1}}$ and $s_k^{i_j} = f_{i_j}(\mathbf{y}_k)$. The classifier $f_{i_1}(\mathbf{y}_k)$ gave the highest score and $f_{i_2}(\mathbf{x}_k)$ gave the second highest score. The *decision boundary* between class i_1 and class i_2 is defined as:

$$D_{12} = \{\mathbf{y} |\; f_{i_1}(\mathbf{y}) = f_{i_2}(\mathbf{y})\}$$

Using this definition, the confidence of the the predicted label i_1 of point $\mathbf{x}_k$ is determined by the distance of $\mathbf{y}_k$ to the decision boundary D_{12}, namely the minimal distance of $\mathbf{y}_k$ to D_{12}:

$$d_k = min_{\delta \mathbf{y}} ||\delta \mathbf{y}|| \quad \text{s.t.} \quad (\mathbf{y}_k + \delta \mathbf{y}) \in D_{12}$$

Let's derive d_k. Assume a given point $\mathbf{x}$ and its DNN latest layer output $\mathbf{y} = F(\mathbf{x},\theta)$. W.l.o.g let's the largest and the second largest scores of the classifier be: $s^1 = \mathbf{w}_1^T\mathbf{y}+b_1$ and $s^2 = \mathbf{w}_2^T\mathbf{y}+b_2$, respectively. We are looking for the smallest $\delta\mathbf{y}$ satisfying:

$$\mathbf{w}_1^T(\mathbf{y}+\delta\mathbf{y})+b_1 = \mathbf{w}_2^T(\mathbf{y}+\delta\mathbf{y})+b_2$$

Re-arranging terms we get:

$$-(\mathbf{w}_1-\mathbf{w}_2)^T\delta\mathbf{y} = (s^1-s^2)$$

The least-norm solution of the above under-determined equation is calculated using the right pseudo-inverse of $(\mathbf{w}_1-\mathbf{w}_2)^T$ which gives:

$$\delta\mathbf{y} = -(s^1-s^2)\frac{\mathbf{w}_1-\mathbf{w}_2}{||\mathbf{w}_1-\mathbf{w}_2||^2}$$

and therefore the MMS of $\mathbf{y}$ is ginven by:

$$d = ||\delta\mathbf{y}|| = \sqrt{\delta\mathbf{y}^T\delta\mathbf{y}} = \frac{s^1-s^2}{||\mathbf{w}_1-\mathbf{w}_2||}$$

therefore,

$$d_k = argmin_k\frac{s_k^{i_1}-s_k^{i_2}}{||\mathbf{w}_{i_1}-\mathbf{w}_{i_2}||}$$

The distance d_k is the *Minimal Margin Score* (MMS) of point $\mathbf{x}_k$. The larger the d_k, the more confident we are about the predicted label. Conversely, the smaller the d_k, the less confident we are about the predicted label i_1. Therefore, d_k can serve as a confidence measure for the predicted labels. Accordingly, the best points to select for the back-propagation step are the points whose MMS are the smallest.

5 Multi-Margin Regularization for Multiclass Classification

We continue to leverage the principle of large margin in neural networks and extend the aforementioned ideas and present a novel regularization term, which we denote as *Multi-Margin Regularization* (MMR), which can be added to any existing loss function in DNNs. We derive the regularization term starting from

Algorithm 1: Selection by Minimal Margin Scores

```
Require: Inputs $\mathscr{X} = \{\mathbf{x}_i\}_{i=1}^{B}$ , $F(\cdot;\theta_0)$ - Training model, b - batch size
  $t \leftarrow 1$
  repeat
    $\mathscr{Y} \leftarrow F(\mathscr{X};\theta_{t-1})$            forward pass on batch of size B
    $MMS \leftarrow d(\mathscr{Y})$                calculates the Minimal Margin Scores of $\mathscr{Y}$
    $S \leftarrow \mathrm{sort\_index}(MMS, b)$         stores the index of the $b$ smallest scores
    $\mathscr{X}_b = \{\mathbf{x}_i |\ i \in S\}$               subset of $\mathscr{X}$ of of size b
    $\theta_t \leftarrow sgd\_step(F(\mathscr{X}_b;\theta_{t-1}))$     back prop. with batch of size b
    $t \leftarrow t+1$
  until reached final model accuracy
```

the binary case of large margin classification and generalize it to the multi-class case. This regularization term aims at increasing the margin induced by classifiers attained from the true class and its most competitive class. By summing over the margin distribution we compensate for class imbalance in the regularization term. Furthermore, due to the dynamic nature of feature space representation when training neural networks, we scale our formulation by the ever-changing maximal norm of the samples in the feature space, $\|\phi_{max}\|$.

We empirically show that applying this regulizer on the output layer of various DNNs, in different classification tasks and from different domains, is sufficient to obtain a substantial improvement in accuracy. In particular, we achieve valuable accuracy improvement in numerous image and text classification tasks, including CIFAR10, CIFAR100, ImageNet, MNLI, QQP and more.

5.1 Margin Analysis for Binary and Multiclass Classification

Consider a classification problem with two classes $\mathscr{Y} \in \{+1,-1\}$. We denote by $\mathscr{X} \in \mathscr{R}^d$ the input space. Let $f(\mathbf{w}^T\mathbf{x}+b)$ be a linear classifier, where $\mathbf{x} \in \mathscr{X}$ 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 $\{(\mathbf{x}_1,y_1),(\mathbf{x}_2,y_2),\cdots,(\mathbf{x}_n,y_n)\} \in (\mathscr{X} \times \mathscr{Y})^m$ where each example is sampled identically and independently from an unknown distribution $\mathscr{D}$ over $\mathscr{X} \times \mathscr{Y}$. The goal is to classify correctly new samples drawn from $\mathscr{D}$.

Denote by ℓ the (linear) decision boundary of the classifier

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

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, there exist numerous consistent classifiers, i.e., classifiers that classify all examples correctly. Better generalization, however, is achieved by selecting the classifier that maximizes the margin $\hat{d}$,

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

This optimization is redundant with the length of $\mathbf{w}$ and b. Imposing $y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1$ removes this redundancy and results in the following equivalent minimization problem [10]:

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

To handle noisy and linearly inseparable data, the set of linear constraints can be relaxed and substituted by 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 left term in Formula 3 is the *regularization* component and it promotes increasing of the margin between the data points and the decision boundary. The right term of the formula is the *empirical risk* component, imposing correct classifications on the training samples. The two terms employ two complementary forces; the former improves the generalization capability while the latter ensures the classification will be carried out correctly.

Next, we extend the large margin principle to the multi-class case. Let us assume we have a classification problem with n classes, $\mathscr{Y} \in \{1, \cdots, n\}$, and a set of m training samples: $\{(\mathbf{x}_i, y_i)\} \in (\mathscr{X} \times \mathscr{Y})^m$. We now assign a score to each class: $s_i : \mathscr{X} \to \mathbb{R}, \ \forall\ i = 1..n$. For a linear classification, the j^{th} score of point i is:

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

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

$$\hat{y}_i = \arg\max_j s_j(\mathbf{x}_i)$$

For any two classes, $(p,q) \in \mathscr{Y} \times \mathscr{Y}$, the decision boundary between these classes is given by (see Figure 5):

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

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. Similarly to Equation 2, the geometric distance of a point $\mathbf{x}$ from $\ell_{p,q}$ is

$$d_{p,q}(\mathbf{x}) = \frac{\mathbf{w}_{p,q}^T\mathbf{x} + b_{p,q}}{\|\mathbf{w}_{p,q}\|} \tag{4}$$

For point $\mathbf{x}_i$, denote by $s_{y_i}(\mathbf{x}_i)$ the score for the true class and by $s_{m_i}(\mathbf{x}_i)$ the maximal score attained for the non-true classes, i.e., $m_i = \arg\max_{j \neq y_i} s_j(\mathbf{x}_i)$. Class m_i is the *competitive* class vis-à-vis y_i. The boundary decision between y_i and its competitive class is ℓ_{y_i,m_i} whose geometric distance to $\mathbf{x}_i$ is

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

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.

For the multi-class case, Equation 3 can be generalized to the following optimization problem,

$$\min_{W,\mathbf{b}} \sum_i \|\mathbf{w}_{y_i,m_i}\|^2 + \lambda \sum_i \max(0, 1 - (\mathbf{w}_{y_i,m_i}^T\mathbf{x}_i + b_{y_i,m_i})) \tag{6}$$

where the optimization is over $W \doteq \{\mathbf{w}_1, \cdots, \mathbf{w}_n\}$, and $\mathbf{b} \doteq \{b_1, \cdots, b_n\}$. Here too, the left-hand term is the regularization penalty while the right-hand term represents the empirical risk with a hinge loss. The regularization term aims to increase the margin between the true class and its competitive class. Note, though, that the summation is over the margin distribution (i is the instance index). If the instances are evenly distributed over the classes, then this is equivalent to summation over the classes. Otherwise, this summation compensates for class imbalance in the regularization term.

5.2 Large Margin in Deep Neural Networks

Applying the above scheme directly to DNNs poses several problems. First, these networks employ a non-linear mapping from the input space into a representation space: $\phi_i = F(\mathbf{x}_i, \theta) : \mathscr{X} \rightarrow \Phi$, where θ are the network parameters. The vector ϕ_i can be interpreted as a feature vector based on which the last layer in a DNN 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 $\mathscr{X}$, as suggested in [61], requires backpropagating derivatives downstream the network up to the input layer, and calculating distances to the boundary up to the first order of approximation. In highly non-linear mappings, this approximation loses accuracy very fast as we move away from the decision boundary. Therefore, we apply the large margin principle in the last layer, where the distances to the decision boundary are Euclidean in the feature space Φ:

$$d_{y_i,m_i}(\phi_i) = \frac{\mathbf{w}_{y_i,m_i}^T \phi_i + b_{y_i,m_i}}{\|\mathbf{w}_{y_i,m_i}\|} \tag{7}$$

The second problem stems from the fact that in Equation 5 the input space $\mathscr{X}$ is fixed along the course of training while the feature space Φ in Equation 7 is constantly changing. Accordingly, maximizing the margins in Equation 7 can be trivially attained by scaling up the space Φ. Therefore, the feature space Φ must be constrained. In our scheme, we divide Equation 7 by $\|\phi_{max}\|$, the maximal norm of the samples in the feature space, of the current batch. This ensures that scaling up the feature space will not increase the distance in a free manner. The proposed formulation is translated, similarly to Equation 6, into the following optimization problem

$$\min_{W,\mathbf{b}} \sum_i \mathscr{R}_i + \lambda \sum_i \mathscr{C}_i \tag{8}$$

where

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

denotes the margin regularization term, and $\mathscr{C}_i$ is the empirical risk term. While for SVM, hinge loss is commonly used, in DNNs the common practice is to use cross-entropy

$$\mathscr{C}_i = -\log(P_{y_i})$$

where P_{y_i} is the probability of the true label y_i 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 hinge loss, cross-entropy will strive for correct classification while the regularization term will maximize the margin. For the rest of this paper we denote $\mathscr{R}_i$ as the *multi-margin regularization* (MMR).

Note that the regularization term in this scheme is different from the weight decay commonly applied in deep networks. First, here, the minimization is applied over the $\mathbf{w}$ differences of: $\|\mathbf{w}_{y_i,m_i}\|^2 = \|\mathbf{w}_{y_i} - \mathbf{w}_{m_i}\|^2$. Additionally, the regularization term is multiplied by the $\|\phi_{max}\|$. Lastly, the regularization term is implemented only at the last layer.

6 Experiments

In this section[3], we report on the series of experiments we designed to evaluate the MMS selection method's ability to achieve a faster convergence than the original training algorithms (the baseline) and data augmentation and the MMR's ability to achieve a higher accuracy score.

The experiments were conducted on commonly used datasets and neural network models, in the vision and *natural language processing* (NLP) realms.

Our experimental workbench is composed of CIFAR10, CIFAR100 [17] and ImageNet [67] for image classification; Question NLI (QNLI) [68], MultiNLI (MNLI) [69] and Recognizing Textual Entailment (RTE) [70] for natural language inference; MSR Paraphrase Corpus (MRPC) [71] and Quora Question Pairs (QQP) [72] for sentence similarity; Stanford Sentiment Treebank-2 (SST-2) [73] for text classification.

6.1 Image Classification

CIFAR10 and CIFAR100. These are image classification datasets that consist of 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 held-out validation set, as suggested in common practice. For our experiments, we used ResNet-44 [14] and WRN-28-10 [74] architectures. We applied the original hyper parameters and training regime using a batch-size of 64. In addition, we used the original augmentation policy as described in [14] for ResNet-44, while adding cutout [75] and auto-augment [76] for WRN-28-10. Optimization was

[3] All experiments were conducted using PyTorch framework, and the code is publicly available at `https://github.com/berryweinst/mms-select`.

performed for 200 epochs (equivalent to $156K$ iterations) after which baseline accuracy was obtained with no apparent improvement.

ImageNet. For large-scale evaluation, we used the ImageNet dataset [67], containing more than 1.2M images in 1k classes. In our experiments, we used MobileNet [77] architecture and followed the training regime established by [36] in which an initial 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 base batch size of 256 over four devices and L_2 regularization over weights of convolutional layers as well as the standard data augmentation.

6.1.1 Convergence speedup via MMS selective sampling

To test our hypothesis that using the MMS selection method we could accelerate training while preserving final model accuracy, we designed a new, more aggressive leaning-rate drop regime than the one used by the authors of the original paper. Figure 6 presents empirical evidence supporting out hypothesis. We compared the results of our MMS method against random selection [4], and against *hard-negative mining* that prefers samples with low prediction scores [78, 42]. For the latter, we used the implementation suggested by [65], termed ”NM-sample”, where the cross-entropy loss is used for the selection. For CIFAR10 and ResNet-44, we used the original LRs $\eta = \{0.1, 0.01, 0.001, 0.0001\}$ while decreasing them at steps $\{24992, 27335, 29678\}$, equivalent to epochs $\{32, 35, 38\}$ with a batch of size 64. As depicted in Figure 6 (left), we can see that our selection method indeed yields validation accuracy extremely close to the one reached by the baseline training scheme, with considerably fewer training steps. Specifically, we reached 93% accuracy after merely $44K$ steps (a minor drop of 0.25% compared to the baseline). We also applied the early drop regime to the baseline configuration as well as to the NM-samples. Both failed to reach the desired model accuracy while suffering from a degradation of 1.57% and 1.22%, respectively.

Similarly, we applied the early LR drop scheme for CIFAR100 and WRN-28-10, using $\eta = \{0.1, 0.02, 0.004, 0.0008\}$ and decreasing steps $\{39050, 41393, 43736\}$ equivalent to epochs $\{50, 53, 56\}$, with batch of size 64. As depicted in Figure 6 (right), MMS accuracy reached 82.2% with a drop of 0.07% compared to the baseline, while almost halving the number of steps ($80K$ vs. $156K$). On the other

[4] Referred to as baseline with and without an early LR drop.

hand, the baseline and the NM-sample schemes failed to reach the desired accuracy after we applied a similar early drop regime. For the NM-sample approach, the degradation was the most significant, with a drop of 2.97% compared to the final model accuracy, while the baseline drop was approximately 1%.

These results are in line with the main theme of selective sampling that strives to focus training on more informative points. Training loss, however, can be a poor proxy for this concept. For example, the NM-sample selection criterion favors high loss scores, which obviously increases the training error, while our MMS approach selects uncertain points, some of which might be correctly classified. Others might be mis-classified by a small margin, but they are all close to the decision boundary, and hence useful for training.

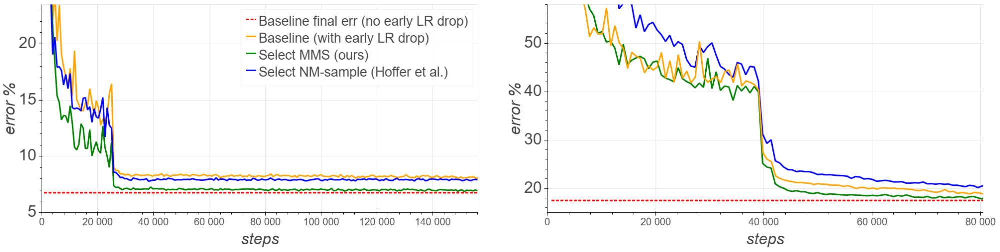


Figure 6: Validation errors of ResNet44, CIFAR10 (left) and WRN-28-10, CIFAR100 (right). We compared the baseline training, NM-sample selection (hard negative mining), and MMS (our) selection method using a faster regime. We ploted the regular regime baseline's final errors as a dotted line for perspective. **The MMS selection method achieves on par final test accuracy using fewer numbers of training steps**.

6.2 Improving accuracy via Multi-Margin Regularization

To examine our MMR scheme as described in details in section 5.2, we added it to the objective function as an additional regularization term. We used a trade-off α factor between the cross-entropy loss and the additional regularization as follows:

$$\mathscr{L}(\theta) = -\sum \log(P_{y_i}) + \alpha \mathscr{R}_i$$

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.

Table 1 demonstrates our final results in increasing the final model accuracy. Specifically, we managed to improve baseline accuracy in ResNet-44 from 93.22% to 93.83% and from 93.19% to 93.34% in VGG. A relative change in error of 9.00% and 2.20%, respectively, on the CIFAR10 dataset. Furthermore, we show a substantial decrease of 5.77% in the error for CIFAR100 using WRN-28-20 model (see Figure 7), raising its absolute accuracy by more than 1%. Altogether, we observed a 2.67% average decrease in error on all datasets.

6.2.1 Natural Language Classification Tasks

To challenge our premise, we chose to further examine our MMR on NLP related model and datasets. In particular, we used $\text{BERT}_{\text{BASE}}$ model [79] with 12 transformer layers, hidden dimensional size of 768 and 12 self-attention heads. Fine-tuning was performed using Adam optmizer as in the pre-training, a dropout probability of 0.1 on all layers. Additionally, we used a learning rate of $2e-5$ over 3 epochs in total for all the tasks. We use the original WordPiece embeddings [80] with a 30k token vocabulary. For our methods, similarly to the image classification task, we also used α factor in the objective function, and found via grid search, $\alpha = 1e-5$ to be the optimal [5].

We performed experiments on a variety of supervised tasks, specifically by applying downstream task fine-tuning on natural language inference, semantic similarity, and text classification. All of these tasks are available as part of the GLUE multi-task benchmark [68].

Natural Language Inference The task of natural language inference (NLI) or recognizing textual entailment, is where a pair of sentences is given and the classifier has to decide whether they contradict one each other or not. Although there has been a lot of progress, the task remains challenging due to the presence of a wide variety of phenomena like 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).

[5] We applied $\alpha = 1e-6$ only to evaluate our method accuracy with miss-matched MNLI

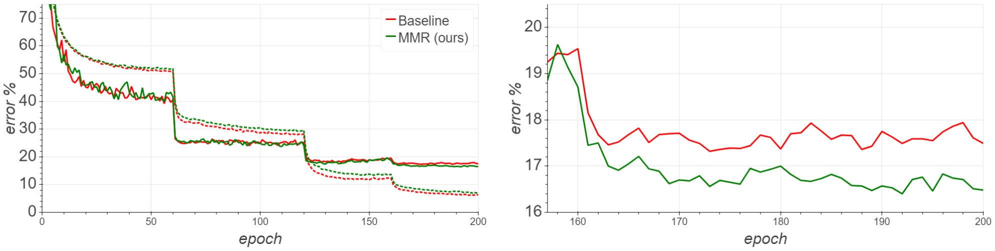
Model	Dataset	Baseline	Our MMR	Change
ResNet-44 [14]	CIFAR10	93.22%	93.83%	**9.00%**
VGG [81]	CIFAR10	93.19%	93.34%	2.20%
WRN-28-10 + autoaugment + cutout [74]	CIFAR100	82.51%	83.52%	**5.77%**
VGG + autoaugment + cutout	CIFAR100	73.93%	74.19%	1.00%
MobileNet [77]	ImageNet	71.17%	71.44%	0.94%
$\text{BERT}_{\text{BASE}}$ [79]	QNLI	91.06%	91.48%	**4.70%**
	SST-2	92.08%	92.43%	**4.42%**
	MRPC	90.68%	91.43%	**8.05%**
	RTE	68.23%	69.67%	**4.53%**
	QQP	87.9%	88.04%	1.16%
	MNLI	84.5%	84.70%	1.29%

Table 1: Test accuracy results. Top1 for CIFAR10/100 datasets. Relative change in error over baseline is listed in percentage, and improvements higher than 4% are marked in bold. F1 scores are reported for QQP and MRPC. For MNLI, we reported the average of the matched (with $\alpha = 1e-5$) and miss-matched (with $\alpha = 1e-6$) for both baseline and our MMR

As shown in table 1, our scheme with using the regularization term outperforms baseline results on all of the three tasks. We achieve absolute improvement of up to 1.44% on RTE and a relative change in error of 4.53%. On QNLI and MNLI we also achieved higher scores of 91.48% (accuracy) and 84.70% (F1), outperforming the baseline results by 0.42% and 0.2%, respectively.

Semantic Similarity This task involves in 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 comparing same ideas using different expressions or the other way around. We evaluate our approach on QQP and MRPC downstream task, outperforming baseline results as can be seen in Table 1. On MRPC in particular, we achieved a relative change of more than 8%.

Text Classification Lastly, we evaluate 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 of their sentiment.



(a) CIFAR100 train error with using MMR (b) CIFAR100 test error with using MMR

Figure 7: Training (dashed) and validation error of CIFAR100 using WRN28-10 neural network. Comparing baseline training and our MMR approach. We use linear scale α, starting with $1e-5$ up to $1e-3$.

Our approach exceeds the baseline by a relative error change of 4.42%.

Overall, applying our MMR boosts the accuracy of all the above tasks. This is also an indication that our approach works well for different tasks from various domains.

7 Discussion

We presented a selective sampling method designed to accelerate the training of deep neural networks. Specifically, we utilized uncertainty sampling, where the criterion for selection is the distance to the decision boundary for the multiclass case. To this end, we introduced a novel measurement, the *minimal margin score* (MMS), which measure the minimal amount of displacement an input should take until its predicted classification is switched. For multiclass linear classification, the MMS measure is a natural generalization of the margin-based selection criterion, which was thoroughly studied in the binary classification setting. We demonstrate a substantial acceleration for training in commonly used DNN architectures and for popular image classification tasks. The efficiency of our method is compared against the standard training procedures, and against Hard negative mining selection. Furthermore, we demonstrate an additional speedup when we adopt a more aggressive learning-drop regime.

Our selection criterion was inspired by the Active Learning methods, but our goal, accelerate training, is different. Active learning mainly concerns about the labelling cost. Hence, it is common to keep on training till (almost) convergence, before turning to select additional examples to label. However, such an approach is less efficient when it comes to acceleration. In such a scenario, we can be more aggressive; since labelling cost is not a concern, we can re-select a new batch of examples in each training step.

An efficient implementation is also crucial for gaining speedup. Our scheme provides many opportunities for further acceleration. For example, fine-tuning the sample size used to select and fill up a new batch. We can balance between the selection effort conducted at the end of the forward pass, and the compute resources and efforts required to conduct the back-propagation pass more efficiently. This also opens an opportunity to design and use a dedicated hardware for the selection. In the past few years, custom ASIC devices that accelerate the inference phase of neural networks were developed [82, 65, 22]. Furthermore, in [83], it was shown that using quantization for low-precision computation induces little or no degradation in final model accuracy. This observation, together with the fast and efficient inference achieved by ASICs, make them appealing to be used as a supplement accelerator in the forward pass of our selection scheme.

The MMS measure doesn't use labels. Thus it can be used to select samples in an active learning setting as well. Moreover, similarly to [55] the MMS measure can be implemented at other layers in the deep architecture. This enables to select examples that directly impact training at all levels. The additional compute associated with such calculating and selecting the right batch content, makes it less appealing for acceleration. However, for active learning, it may introduce an additional gain, since the selection criterion chooses examples which are more informative for various layers. The design of a novel Active Learning method is left for further study.

In addition to the MMS, We studied a multi-class margin analysis for DNNs and use it to devise a novel regularization term, the *multi-margin regularization* (MMR). Similarly to previous formulations, the MMR 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. The main difference between the MMR and common regularization terms is that MMR 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. Additionally, weight differences are minimized rather than the commonly used determinant or other norms of W. Lastly, MMR in formulated and performed over the margin distribution to compensate for class imbalance in the regularization term. The MMR can be incorporated with any empirical risk loss and it is not restrictive to hinge loss or cross-entropy losses. And indeed, using MMR, we demonstrate improved accuracy over a set of experiments in images and text.

References

[1] Xiaojin Jerry Zhu. Semi-supervised learning literature survey. Technical report, University of Wisconsin-Madison Department of Computer Sciences, 2005.

[2] Dana Angluin. Queries and concept learning. *Machine learning*, 2(4):319–342, 1988.

[3] Eric B Baum and Kenneth Lang. Query learning can work poorly when a human oracle is used. In *International joint conference on neural networks*, volume 8, page 8, 1992.

[4] Ross D King, Kenneth E Whelan, Ffion M Jones, Philip GK Reiser, Christopher H Bryant, Stephen H Muggleton, Douglas B Kell, and Stephen G Oliver. Functional genomic hypothesis generation and experimentation by a robot scientist. *Nature*, 427(6971):247–252, 2004.

[5] David Cohn, Les Atlas, and Richard Ladner. Improving generalization with active learning. *Machine learning*, 15(2):201–221, 1994.

[6] David D Lewis and William A Gale. A sequential algorithm for training text classifiers. In *SIGIR'94*, pages 3–12. Springer, 1994.

[7] Tom M Mitchell. Generalization as search. *Artificial intelligence*, 18(2):203–226, 1982.

[8] Sanjoy Dasgupta, Daniel J Hsu, and Claire Monteleoni. A general agnostic active learning algorithm. In *Advances in neural information processing systems*, pages 353–360, 2008.

[9] Claude E Shannon. A mathematical theory of communication. *Bell system technical journal*, 27(3):379–423, 1948.

[10] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[11] H Sebastian Seung, Manfred Opper, and Haim Sompolinsky. Query by committee. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 287–294, 1992.

[12] Ran Gilad-Bachrach, Amir Navot, and Naftali Tishby. Query by committee made real. In *Advances in neural information processing systems*, pages 443–450, 2006.

[13] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[14] 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.

[15] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[16] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 807–814, 2010.

[17] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.

[18] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[19] Sebastian Ruder. An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*, 2016.

[20] Henry J Kelley. Gradient theory of optimal flight paths. *Ars Journal*, 30(10):947–954, 1960.

[21] Neil Houlsby, Ferenc Huszár, Zoubin Ghahramani, and Máté Lengyel. Bayesian active learning for classification and preference learning. *arXiv preprint arXiv:1112.5745*, 2011.

[22] Norman P Jouppi, Cliff Young, Nishant Patil, David Patterson, Gaurav Agrawal, Raminder Bajwa, Sarah Bates, Suresh Bhatia, Nan Boden, Al Borchers, et al. In-datacenter performance analysis of a tensor processing unit. In *2017 ACM/IEEE 44th Annual International Symposium on Computer Architecture (ISCA)*, pages 1–12. IEEE, 2017.

[23] Aharon Azulay and Yair Weiss. Why do deep convolutional networks generalize so poorly to small image transformations? *arXiv preprint arXiv:1805.12177*, 2018.

[24] Teng Zhang and Zhi-Hua Zhou. Multi-class optimal margin distribution machine. In *International Conference on Machine Learning*, pages 4063–4071, 2017.

[25] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

[26] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

[27] Robert E Schapire, Yoav Freund, Peter Bartlett, Wee Sun Lee, et al. Boosting the margin: A new explanation for the effectiveness of voting methods. *The annals of statistics*, 26(5):1651–1686, 1998.

[28] Olivier Bousquet and André Elisseeff. Algorithmic stability and generalization performance. In *Advances in Neural Information Processing Systems*, pages 196–202, 2001.

[29] Daniel Soudry, Elad Hoffer, Mor Shpigel Nacson, Suriya Gunasekar, and Nathan Srebro. The implicit bias of gradient descent on separable data. *The Journal of Machine Learning Research*, 19(1):2822–2878, 2018.

[30] Saharon Rosset, Ji Zhu, and Trevor J Hastie. Margin maximizing loss functions. In *Advances in neural information processing systems*, pages 1237–1244, 2004.

[31] Yiding Jiang, Dilip Krishnan, Hossein Mobahi, and Samy Bengio. Predicting the generalization gap in deep networks with margin distributions. In *7th International Conference on Learning Representations, ICLR*, 2019.

[32] Ashutosh Garg, Sariel Har-Peled, and Dan Roth. On generalization bounds, projection profile, and margin distribution. In *ICML*, pages 171–178, 2002.

[33] Yarin Gal, Riashat Islam, and Zoubin Ghahramani. Deep bayesian active learning with image data. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1183–1192. JMLR. org, 2017.

[34] Andreas Kirsch, Joost van Amersfoort, and Yarin Gal. Batchbald: Efficient and diverse batch acquisition for deep bayesian active learning. In *Advances in Neural Information Processing Systems*, pages 7024–7035, 2019.

[35] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059, 2016.

[36] Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch sgd: Training imagenet in 1 hour. *arXiv preprint arXiv:1706.02677*, 2017.

[37] Xianyan Jia, Shutao Song, Wei He, Yangzihao Wang, Haidong Rong, Feihu Zhou, Liqiang Xie, Zhenyu Guo, Yuanzhou Yang, Liwei Yu, et al. Highly scalable deep learning training system with mixed-precision: Training imagenet in four minutes. *arXiv preprint arXiv:1807.11205*, 2018.

[38] Chris Ying, Sameer Kumar, Dehao Chen, Tao Wang, and Youlong Cheng. Image classification at supercomputer scale. *arXiv preprint arXiv:1811.06992*, 2018.

[39] Leslie N Smith and Nicholay Topin. Super-convergence: Very fast training of neural networks using large learning rates. In *Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications*, volume 11006, page 1100612. International Society for Optics and Photonics, 2019.

[40] Leslie N Smith. No more pesky learning rate guessing games. *CoRR, abs/1506.01186*, 5, 2015.

[41] James Martens. Deep learning via hessian-free optimization. In *ICML*, volume 27, pages 735–742, 2010.

[42] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823, 2015.

[43] Guillaume Alain, Alex Lamb, Chinnadhurai Sankar, Aaron Courville, and Yoshua Bengio. Variance reduction in sgd by distributed importance sampling. *arXiv preprint arXiv:1511.06481*, 2015.

[44] Ilya Loshchilov and Frank Hutter. Online batch selection for faster training of neural networks. *arXiv preprint arXiv:1511.06343*, 2015.

[45] Angelos Katharopoulos and Franccois Fleuret. Not all samples are created equal: Deep learning with importance sampling. *arXiv preprint arXiv:1803.00942*, 2018.

[46] Burr Settles. Active learning literature survey. Technical report, University of Wisconsin-Madison Department of Computer Sciences, 2009.

[47] David D Lewis and Jason Catlett. Heterogeneous uncertainty sampling for supervised learning. In *Machine learning proceedings 1994*, pages 148–156. Elsevier, 1994.

[48] Simon Tong and Daphne Koller. Support vector machine active learning with applications to text classification. *Journal of machine learning research*, 2(Nov):45–66, 2001.

[49] Colin Campbell, Nello Cristianini, and Alex Smola. Query learning with large margin classifiers. In *ICML*, volume 20, page 0, 2000.

[50] Sanjoy Dasgupta. Coarse sample complexity bounds for active learning. In *Advances in neural information processing systems*, pages 235–242, 2006.

[51] Maria-Florina Balcan, Andrei Broder, and Tong Zhang. Margin based active learning. In *International Conference on Computational Learning Theory*, pages 35–50. Springer, 2007.

[52] Sanjoy Dasgupta. Two faces of active learning. *Theoretical computer science*, 412(19):1767–1781, 2011.

[53] Melanie Ducoffe and Frederic Precioso. Adversarial active learning for deep networks: a margin based approach. *arXiv preprint arXiv:1802.09841*, 2018.

[54] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2574–2582, 2016.

[55] Yiding Jiang, Dilip Krishnan, Hossein Mobahi, and Samy Bengio. A margin-based measure of generalization for deep networks. *ICLR 2019*.

[56] Richard O Duda, Peter E Hart, et al. *Pattern classification and scene analysis*. Wiley New York, 1973.

[57] Vladimir Vapnik. *Statistical learning theory*. Wiley New York, 1998.

[58] Koby Crammer and Yoram Singer. Ultraconservative online algorithms for multiclass problems. *Journal of Machine Learning Research*, 3(Jan):951–991, 2003.

[59] Shizhao Sun, Wei Chen, L. Wang, and T. Liu. Large margin deep neural networks: Theory and algorithms. *ArXiv*, abs/1506.05232, 2015.

[60] Gamaleldin Elsayed, Dilip Krishnan, Hossein Mobahi, Kevin Regan, and Samy Bengio. Large margin deep networks for classification. In *Advances in neural information processing systems*, pages 842–852, 2018.

[61] Jure Sokolić, Raja Giryes, Guillermo Sapiro, and Miguel RD Rodrigues. Robust large margin deep neural networks. *IEEE Transactions on Signal Processing*, 65(16):4265–4280, 2017.

[62] Koby Crammer, Ofer Dekel, Joseph Keshet, Shai Shalev-Shwartz, and Yoram Singer. Online passive-aggressive algorithms. *Journal of Machine Learning Research*, 7(Mar):551–585, 2006.

[63] David P Helmbold, Jyrki Kivinen, and Manfred K Warmuth. Relative loss bounds for single neurons. *IEEE Transactions on Neural Networks*, 10(6):1291–1304, 1999.

[64] Yann LeCun and Corinna Cortes. MNIST handwritten digit database. 2010.

[65] Elad Hoffer, Berry Weinstein, Itay Hubara, Sergei Gofman, and Daniel Soudry. Infer2train: leveraging inference for better training of deep networks.

[66] Elad Hoffer, Itay Hubara, and Daniel Soudry. Fix your classifier: the marginal value of training the last weight layer. *arXiv preprint arXiv:1801.04540*, 2018.

[67] 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*, pages 248–255, 2009.

[68] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*, 2018.

[69] Adina Williams, Nikita Nangia, and Samuel R Bowman. A broad-coverage challenge corpus for sentence understanding through inference. *arXiv preprint arXiv:1704.05426*, 2017.

[70] Luisa Bentivogli, Peter Clark, Ido Dagan, and Danilo Giampiccolo. The fifth pascal recognizing textual entailment challenge. In *TAC*, 2009.

[71] William B Dolan and Chris Brockett. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*, 2005.

[72] Zihan Chen, Hongbo Zhang, Xiaoji Zhang, and Leqi Zhao. Quora question pairs, 2018.

[73] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642, 2013.

[74] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.

[75] Terrance DeVries and Graham W Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017.

[76] Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. Autoaugment: Learning augmentation policies from data. *arXiv preprint arXiv:1805.09501*, 2018.

[77] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

[78] Hao Yu, Zhaoning Zhang, Zheng Qin, Hao Wu, Dongsheng Li, Jun Zhao, and Xicheng Lu. Loss rank mining: A general hard example mining method for real-time detectors. In *2018 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2018.

[79] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[80] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*, 2016.

[81] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[82] Goya inference card. `https://habana.ai/wp-content/uploads/pdf/habana_labs_goya_whitepaper.pdf`.

[83] Benoit Jacob, Skirmantas Kligys, Bo Chen, Menglong Zhu, Matthew Tang, Andrew Howard, Hartwig Adam, and Dmitry Kalenichenko. Quantization and training of neural networks for efficient integer-arithmetic-only inference. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2704–2713, 2018.

תקציר

בתחום הלמידה הממוכנת, אסטרטגיות דגימה סלקטיבית של דוגמאות מבוססות על תאוריות שמגיעות מעולם הלמידה האקיבית. בעוד שהמטרה של הראשון היא להמעיט במספר הדוגמאות הנבחרות לצורך תיוג, האחרון נועד על מנת לצמצם את כמות החישוב. בעבודה הזו, אנו מרחיבים את הדיון על שיטות למידה אקיבית שונות וגוזרים את אחת מהתרומות במאמר הזה, שיטת דגימה הנועדה להאיץ את האימון של רשתות נוירונים עמוקות. לצורך כך, אנחנו מציגים מדד חדשני, דירוג שולי המפריד המינימלי (דשמ"מ), אשר מודד את הכמות המינימלית של ההעתקה שיש לכפות על דוגמה מסוימת על מנת שקטגוריית הסיווג שלה תשתנה לאחרת. בסיווג מרובה קטגוריות, מדד הדשמ"מ, היא הכללה טבעית של דגימה סלקטיבית המבוססת על שולי ההפרדה, אשר נחקרה רבות במסגרת הסיווג הבינארי. אנחנו מדגימים אמפירית שכאשר מאמנים ארכיטקטורות של רשתות נוירונים עמוקות פופלריות שנועדו לסווג תמונותבאמצעות השיטה שלנו, יש האצה משתמעותית של תהליך האימון. היעילות של שהיטה שלנו מושווה כנגד שיטת דגימה פופולרית הנקראת "כרייה שלילית קשה". אנחנו מראים האצה משמעותית על ידי שימוש במשטר אימון אגרסיבי במובן של צעד הלמידה, תוך כדי בחירה ע"י שיטת ה דשמ"מ.

תוך שימוש באותו הרעיון, אנו גוזרים ביטוי חדש עבור רגולריזציה כשבבסיסו שולי המפריד, המכונה רגולריזציה מרובת שולי מפרידים (רמש"מ), לרשתות נורירונים עמוקות. הרמש"מ דומה בבסיסו לעקרונות שנוסו על מסווגים לינאריים רדודים, כגון מכונת וקטורים תומכים (מו"ת). שלא כמו מו"ת, רמש"מ ממושקל באופן תמידי על ידי הרדיוס של הספירה החוסמת (כלומר, הנורמה המקסימלית של וקטור הפיצ'רים בדאטה), אשר משתנה לאורך כל האימון. אנחנו מדגימים אמפירית שע"י הוספה מינורית לפונקציית המטרה, השיטה שלנו משיגה תוצאות טובות יותר עבור משימות סיווג שונות בתחומים שונים.

עבודה זו בוצעה בהדרכתם של פרופ' יעקב הל-אור ודר' שי פיין מבי"ס אפי ארזי למדעי המחשב, המרכז הבינתחומי, הרצליה.

הבינתחומי הרצליה

בית ספר אפי ארזי
למדעי המחשב

המרכז הבינתחומי בהרצליה

בית הספר אפי ארזי למדעי המחשב

התכנית לתואר שני (M.Sc.) – מסלול מחקרי

דגימה סלקטיבית ורגולריזציה מבוססי שולי המפריד ברשתות נוירונים עמוקות

מאת

בארי ויינשטיין

עבודת תזה המוגשת כחלק מהדרישות לשם קבלת תואר מוסמך M.Sc. במסלול המחקרי בבית
ספר אפי ארזי למדעי המחשב, המרכז הבינתחומי הרצליה

ספטמבר 2020