Dynamic Sample Selection Strategy in Active Learning

Project Summary Report

Submitted by Amit Bracha, Ran Lottem

Contents

[Introduction 2](#_Toc20713775)

[Problem Setting 3](#_Toc20713776)

[Our Method 3](#_Toc20713777)

[Experiment Methodology 4](#_Toc20713778)

[Results 4](#_Toc20713779)

[Discussion 6](#_Toc20713780)

[Experimental Results 6](#_Toc20713781)

[Possible Follow-ups 6](#_Toc20713782)

[Bibliography 7](#_Toc20713783)

[Appendix – Architecture Schematics 8](#_Toc20713784)

[SelectiveNet, used for all experiments except shallow dualnet 8](#_Toc20713785)

[ActiveDualSelective, used for the shallow dualnet experiment 9](#_Toc20713786)

# Introduction

Active learning is a special case of machine learning which gives the learning algorithm some control over the learning process, by letting it actively decide which examples it will learn from. More precisely, given a pool of unlabeled example the algorithm decides which one to label and to feed into the learning process. Active learning is most useful when labeling examples is costly, because then smart selection of which example to label can save resources and effort. There are few algorithms for active learning in neural networks, and most of those algorithms do not integrate the learning procedure of the neural network with the active learning. This fact can cause the active algorithm to be efficient in the beginning of the process but not in later stages or vice versa, depending on the algorithm. For example, we can think of an active learning algorithm for SVM for classification of several classes. In the beginning it might choose examples in places in the space it has no information on, e.g. far from examples with known classes. But as the classification progresses, we might want the algorithm to choose examples with the smallest margin (i.e. closest to the decision boundary) and thus be more accurate. To perform such a task, traditionally we would need to set hyperparameters that will determine the algorithm's choice of the next example to classify – low-information areas vs. more accuracy in known areas. SelectiveNet‎[1] is a neural network architecture used for classifying images or for regression, which has an additional output. That output states the “confidence level” in the classification (or regression) of the example.

In this project we will try to use this mechanism to determine which example to label next, by training the net on labeled examples and using it on examples that are not labeled. Then we will choose the examples (from the unlabeled ones) which have the smallest “confidence level” in their prediction by the current classifier, to be labeled next. By using the mechanism of SelectiveNet, we let the neural network decide where it has the least information to determine a classification. Our hypothesis is that the decision will be adaptive to the stage of the active learning, and will achieve better results through the process, without the need to set any hyperparameters manually.

# Problem Setting

The goal in active learning is to design and analyze learning algorithms that can effectively choose the samples for which they ask the teacher for a label. Utilizing this technique allows reaching high quality results while maintaining low data requirements. This is relevant in cases where data is expensive, scarce, cumbersome (as in hi-res images), and also when data is too plentiful, making training unnecessarily long when the same results could be achieved faster by choosing which datapoints to train on as part of the training process.

The algorithm used for choosing which samples to request the real label for is therefore a major parameter in the performance of the neural network. Additionally, many of the commonly used algorithms have varied performance across different learning problems, such as those with a "XOR"-like structure[2]. Estimating the performance of the selection algorithm or formulating a supervising algorithm to choose between several such algorithms, could be a place where development could lead to great improvements in accuracy and consistency across different learning tasks.

# Our Method

We intend to classify images from the CIFAR10 database and measure our network's accuracy as a function of the number of samples trained on. As a baseline, we will train a NN which selects its next samples randomly, and a NN which selects its samples using the Softmax Response[3] method. Our experiments use SelectiveNet's confidence measure, and additionally we run an experiment combining two confidence measures. SelectiveNet's confidence measure is combined with a confidence derived from a reconstruction loss generated by an autoencoder that's added to SelectiveNet's architecture and is trained together with SelectivNet's other heads. The reconstruction loss from the autoencoder is converted into a confidence measure for sample via the formula:

In the earlier stages of training, more weight is given to the autoencoder's confidence measure than to SelectiveNet's original confidence . The weighted confidence is calculated as follows:

Where represents the shift in importance from reconstruction loss to selective confidence, and changes in constant steps from 0 to 0.6, meaning that the weight distribution changes from 0.8/0.2 to 0.2/0.8. The idea is to use the deep autoencoder for general feature extraction in the early stages of training, when the dataset is small. As training progresses the network relies more on SelectiveNet’s confidence measure.

# Experiment Methodology

We use the CIFAR10 dataset and set aside a test set for the final measurements. In each of the experiments we start by training the network on 10,000 samples, chosen at random. After training, the network predicts classification for the remaining training set, and the samples are sorted by the network’s confidence score. The four confidence schemes are:

* For the ActiveSelectNet experiment, SelectiveNet’s confidence measure is used
* For the SR Baseline experiment, the maximal probability from the softmax layer is used
* For the Random Baseline experiment, no confidence is measured and instead the samples are sorted randomly
* For the shallow dualnet experiment, see the Our Method section

Then, the 5,000 samples with the lowest confidence measure are added to the training set, and a new network is created. We then repeat this process until the training set size is 45,000.

# Results

Figure 1 shows the accuracy of the different experiments over the test set, as the training set increased in size during each experiment. Using SelectiveNet’s confidence measure gave results very similar to the Softmax Response (SR) method. Shallow DualNet, the experiment where an autoencoder’s reconstruction error was considered as an additional confidence measure, gave results no better than selecting the training samples randomly.

Figure 2 shows the accuracy of each experiment over the samples in the training set that were not used for training in each step. From these results it can be seen that the shallow dualnet experiment, while it had lower accuracy, still did not improve its performance from step to step more than the ActiveSelectNet experiment – indicating that its confidence measure was no better than ActiveSelectNet's.

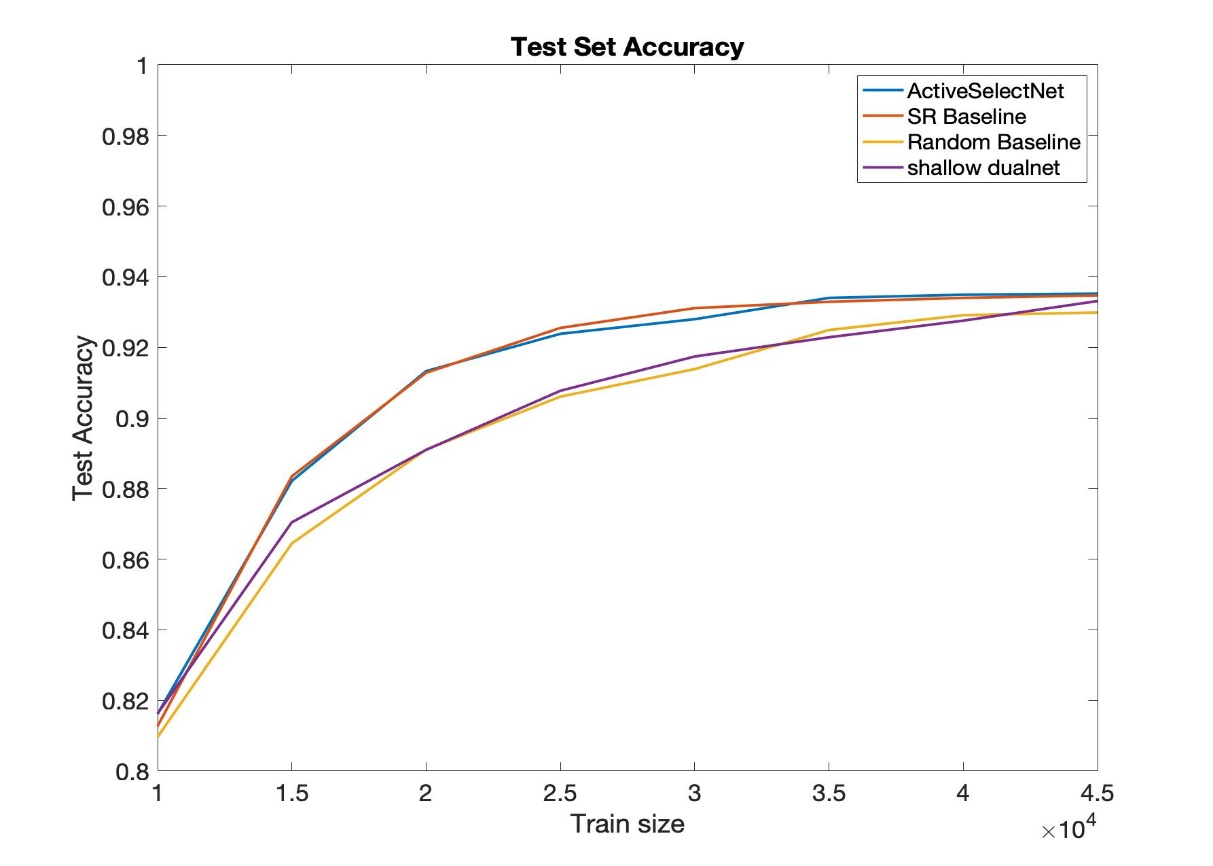


Figure 1: Test set accuracy as a function of seen training set size

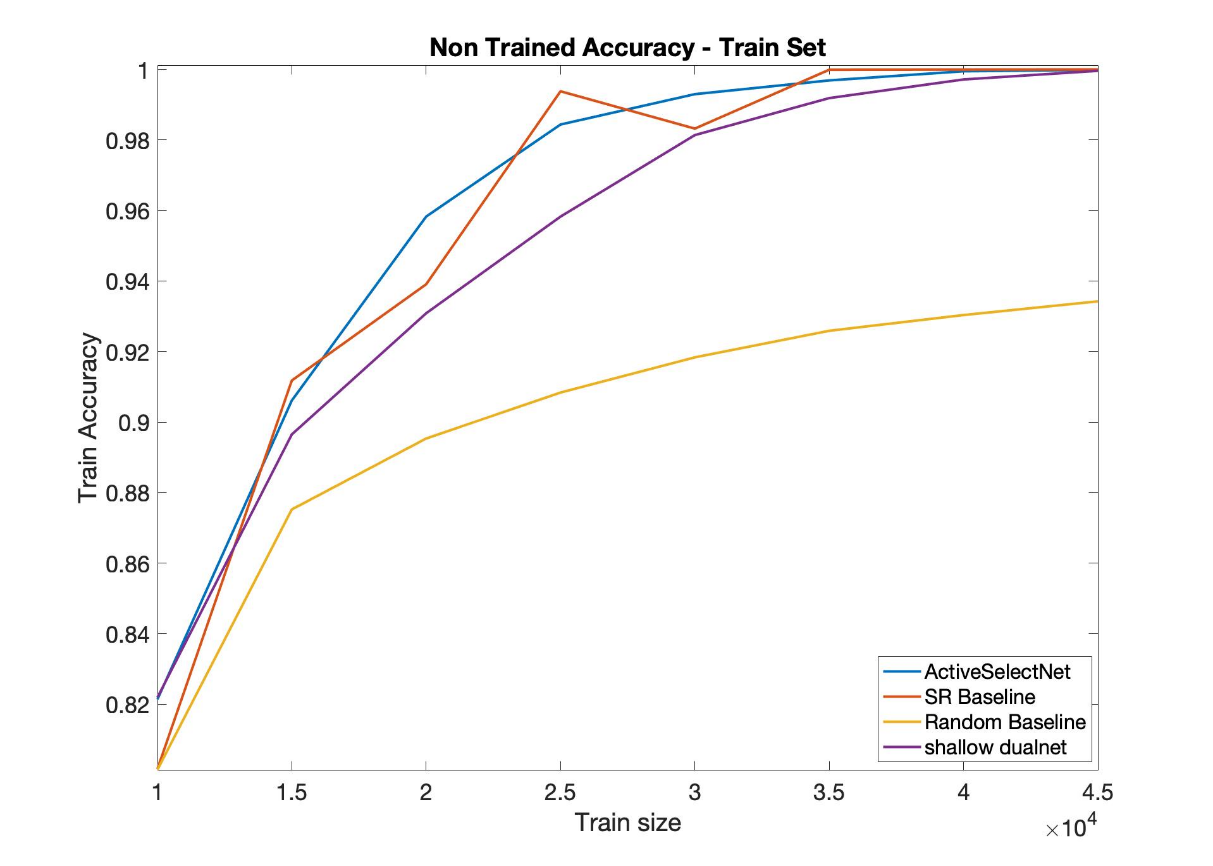


Figure : Accuracy over the unseen training samples as a function of seen training set size

# Discussion

## Experimental Results

We’ve shown SelectiveNet’s use of last-layer embedding to derive a confidence measure can be employed in an active learning setting to give results on par with current state of the art (i.e. Softmax Response).

The autoencoder experiment failed to improve accuracy for a given training set size over a random selection strategy.

## Possible Follow-ups

The following modifications can be tested:

* For the autoencoder experiment:
  + Different architectures of autoencoders can be used, including using the early layers of SelectiveNet’s VGG architecture as the encoder, only adding layers for the decoder as the ‘autoencoder head’.
  + The weights of each confidence measure and the way they change throughout the training process can be modified. The required modifications can be derived by running an experiment with the standard methodology using each of the confidence measures exclusively, and then choosing the accuracy measure which gives the greatest increase in accuracy in the initial stages to be the first measure relied upon, and gradually switching to the second measure as training progresses.
* For the experiment methodology, the number of samples chosen to be added to the training set for the next iteration can by decided each time by setting the confidence cutoff dynamically based on the confidence scores. In the original paper describing Softmax Response[3], Cordelia et al. compared the histograms of the correctly- and incorrectly-predicted samples, to choose a response threshold that would maximize the weight function which computes a weighted sum of right and wrong predictions. For our purposed we could use a unit weight for both right and wrong predictions, and use this same method to set the confidence threshold, this time minimizing the same weight function, so as to choose the most samples where our network was wrong while choosing as few of the samples that the network correctly classified. We would then measure the effect of this modified methodology on all of our experiments, including the SR baseline.

# Bibliography

[1] Y. Geifman and R. El-Yaniv, “SelectiveNet: A Deep Neural Network with an Integrated Reject Option,” 2019.

[2] Y. Baram, R. El-Yaniv, and K. Luz, “Online Choice of Active Learning Algorithms Ran El-Yaniv Kobi Luz,” *J. Mach. Learn. Res.*, vol. 5, pp. 255–291, 2004.

[3] L. Pietro Cordelia, C. De Stefano, F. Tortorella, and M. Vento, “A Method for Improving Classification Reliability of Multilayer Perceptrons,” *IEEE Trans. Neural Networks*, vol. 6, no. 5, pp. 1140–1147, 1995.

# Appendix – Architecture Schematics

## SelectiveNet, used for all experiments except shallow dualnet

|  |  |
| --- | --- |
|  |  |

## ActiveDualSelective, used for the shallow dualnet experiment

