Computer Assignment 3: Comparing SVM, BPNN and CNN performance for Pattern Recognition on MNIST Dataset

Christopher Robbiano — Due: April 19, 2018 — ECE656

1 Introduction

The goal of this computer assignment is to design a Support Vector Machine (SVM) classifier and compare its performance for classifying MNIST digit database samples against the performance of two previously trained neural networks, one using a back propagation neural network (BPNN) and one using convolutional neural network (CNN). The MNIST database of handwritten digits, containing 60,000 training and 10,000 testing samples, is used for comparing the performance of all classifiers. Python and OpenCV were used as the programming language. The OpenCV package contains a very complete implementation of SVMs, allowing for different kernels and learning parameters among other things.

2 Theory

The theory of Back Propagation Neural Nets as well as Convolutional Neural Nets is omitted since it was discussed in the previous report.

2.1 Support Vector Machines

Support Vector Machines are learning machines that use the structure of the data to form decision boundaries instead of an empirical risk minimization technique.

Given a training data set $\{\mathbf{x}_p, d_p\}_{p=1}^{P}$ where $\mathbf{x}_p \in \mathcal{R}^N$ is the pth input pattern, $d_p = \pm 1$is the desired label for two-class problems. If classes are linearly separable,then $N-D$ hyperplanes (DF) $g(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + b$ exist such that

$$\mathbf{x}_p \in C_1 \Longrightarrow g(\mathbf{x}_p) \geq 0 \text{ then } d_p = +1 \tag{1}$$

$$\mathbf{x}_p \in C_2 \Longrightarrow g(\mathbf{x}_p) \geq 0 \text{ then } d_p = -1 \tag{2}$$

The goal is to find $\mathbf{w}_{opt}$ and b_{opt} such that if $d_p = 1$ then $\mathbf{w}_{opt}^T\mathbf{x}_p + b_{opt} \geq 1$ while if $d_p = -1$ then $\mathbf{w}_{opt}^T\mathbf{x}_p + b_{opt} \leq -1$. Samples that meet the equality of previous formulas are called *support vectors*. SVMs aim to maximize the distance between each *canonical hyperplane*, the hyperplanes that satisfies the equality conditions.

The cost function used to find the weight vectors is $J(\mathbf{w}, b, \mathbf{x}) = \frac{1}{2}\mathbf{w}^T\mathbf{w}\sum_{p=1}^{P}\alpha_p[(\mathbf{w}^T\mathbf{x}_p + b)d_p - 1]$, which has $\mathbf{w}_{opt} = \sum_{p=1}^{P}\alpha_p d_p \mathbf{x}_p$.

The SVM often introduces the *kernel trick* to implicitly map the data to a higher dimensional space. The idea is that if the data is not linearly separable in the original dimensional space, than it might be linearly separable in a higher dimensional space. Several different kernel mappings are available, including Gaussian and polynomial as examined in this assignment.

Modifications to SVMs have been discovered such that the decision boundaries determined by an SVM can be extended beyond the 2 class case. The OpenCV implementation used in this assignment allows for any number of classes, and hence a 10 class SVM was used.

3 Network Configuration

Several network configurations were tested throughout the training phase for the SVM classifier. In particular, two different kernel types, polynomial and radial basis function (RBF), were used and are ultimately compared. The network configurations described below are representative of the classifiers that provided the best overall prediction accuracy on the validation set. No discussion of other kernel parameter values will be discussed as they provided worse classification accuracy and are therefore not worth mentioning.

The first kernel type used is a Gaussian kernel, which falls into the RBF family. The OpenCV implementation allows for multi-class SVMs with relatively little effort. A series of RBF kernel SVMs were trained and cross validation was used to identify the optimal values for C and γ which for this problem are 2 and 0.0014, respectively.

The second kernel type used is a polynomial kernel. Again, cross validation was used to identify a polynomial degree of 3 that provided the best classification accuracy.

4 Data Processing and Training

The data training set from MNIST contains 28×28 pixel images covering digits $0 - 9$ with a class representation label corresponding to the digit, i.e. a 4 has a class label of '4'. Some random examples of the MNIST data set can be seen in Figure 1. The 60,000 training images and 10,000 testing images were combined into a total set of 70,000 unique samples. From the combined data set a training set, testing set, and validation set were formed randomly with 15%, 42.5% and 42.5% of the total number of samples, respectively. Each set of images contains an equal amount of samples from each class. Each training image was vectorized into a 784×1 vector for input into the SVM.

The dual optimization problem was solved to find the optimal αs and thus the corresponding weights. In both the polynomial and RBF kernel cases, the training data set was able to get classified with 100% accuracy. The selected parameters for each kernel that were the parameters that provided the highest classification accuracy.

5 Results and Discussion

The SVM was able to essentially match the performance of the best CNN while using the polynomial kernel of degree 3. This can be seen in the ROC curve and confusion matrix in Figure 2a and 2b. The classification accuracy of the SVM with the polynomial kernel is 99.63%. The accuracy of the SVM with the RBF kernel was essentially identical to the accuracy with the polynomial kernel. The best classification accuracy using the RBF kernel was only 99.41% using $C = 2$ and $\gamma = 0.0014$. The ROC curve and confusion matrix for the RBF kernel can be seen in Figure 2c and 2d.

The confusion matrix for the BPNN can be seen in Figure 3b, with the overall accuracy of the classifier being 96.3%.

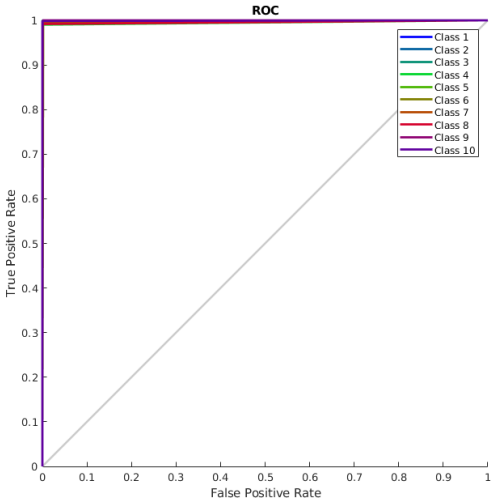
Figure 1: Samples from the MNIST data set

It should be noted that due to limitations in scripts used for generating the confusion matrix that all of the class labels have been incremented by 1, i.e. class 0 is represented by 1 in the figure. During the training phase there were times when the accuracy was fairly poor, around 60%, due to the uniform random sampling of the entire dataset not being truly random resulting in the training set missing representatives from one or more classes. This was readily apparent when looking at the ROC curves for the outputs and seeing one or more classes that had a concave curve. The ROC curves for the best BPNN can be seen in Figure 3a.

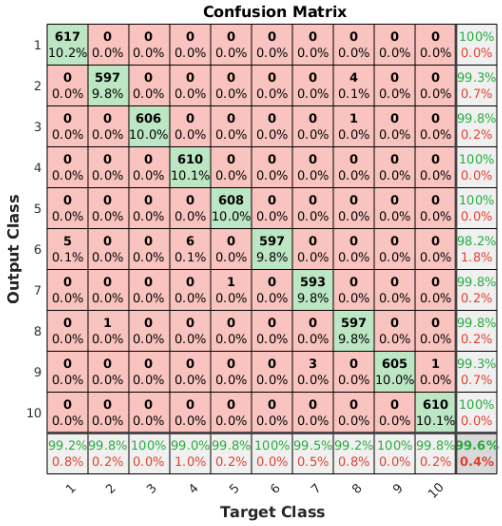
The confusion matrix for the CNNs can be seen in Figure 4b and 4d for the two and three layer networks. The two layer was able to classify correctly 98.6% of the time while the three layer was able to get another 1% performance boost on top of that. This is likely due to an additional amount of location invariance from the third pair of convolutional and pooling layers that extracted lower level features. The ROC curves for the best CNNs can be seen in Figure 4b and 4d.

It is clear that the CNNs are able to out perform the BPNN when trained and validated on the same data, while both kernel type SVMs can match the performance of the CNNs. One major advantage of the SVM classifiers over both the BPNN and CNN is that it is able to find the optimal decision boundary *much* faster during the training phase. This is due, in part, to a complexity that is much smaller than that of the CNN and BPNN networks. Only P weights need to be stored, where P is the number of support vectors in the data set compared to the 1189 weights of the smaller CNN, with $P << 1189$ in all cases.

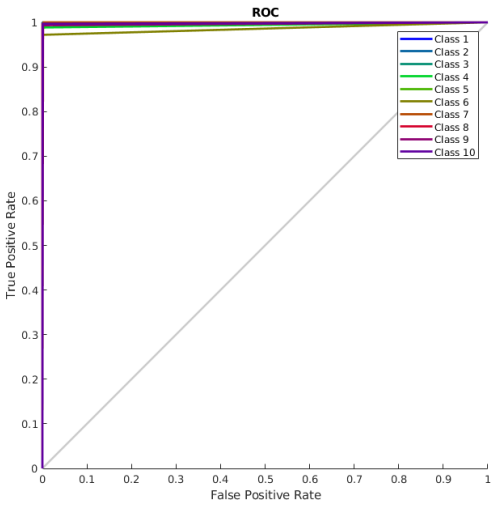
Some interesting things to note for the RBF kernel SVM is that the value of γ is incredibly small. I was originally using a much larger (> 1) value of γ which was causing the network



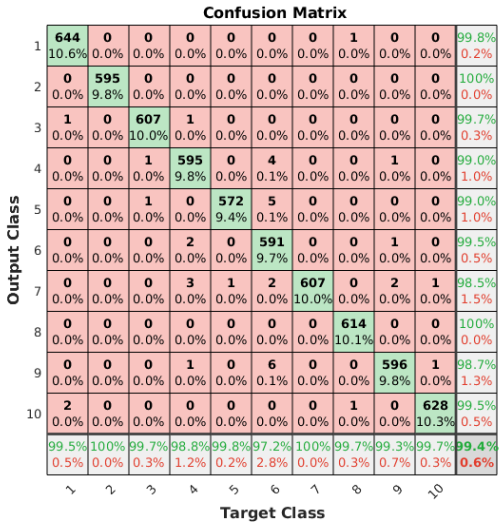
(a) ROC curve for polynomial kernel



(b) Confusion matrix for polynomial kernel



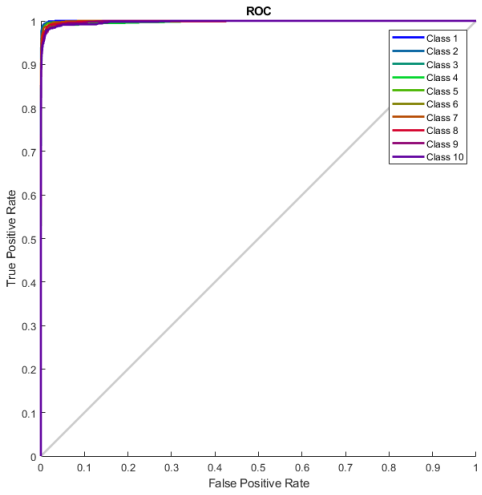
(c) ROC curve for RBF kernel



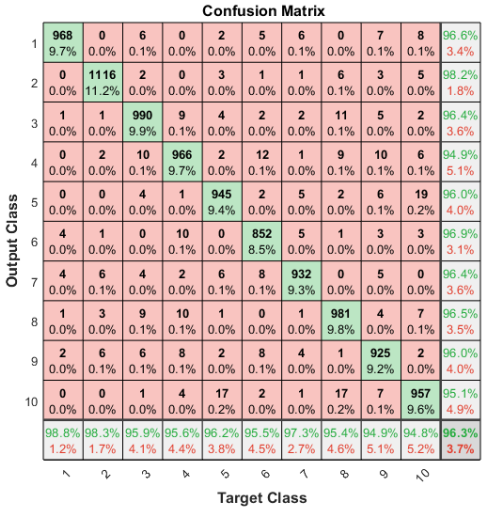
(d) Confusion matrix for RBF kernel

Figure 2: ROC curve and Confusion matrix for SVMs

to classify all samples in the test and validation sets as a single class while being able to correctly classify all samples in the test set correctly. Because of this poor performance, I decided to perform HOG feature extraction and train on those features instead of the raw image. Doing so provided a 100% accuracy on the test, validation and training sets. The results of the classification on the HOG features are not presented here though, as after inquiring with Professor Azimi it was determined that only networks operating on raw images should be evaluated.



(a) ROC curve



(b) Confusion matrix

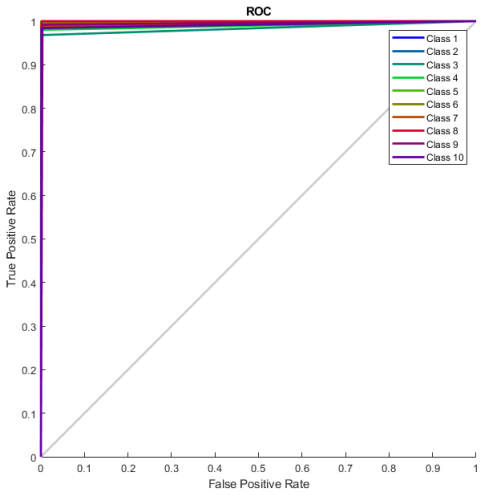
Figure 3: ROC curve and Confusion matrix for BPNN

6 Conclusion

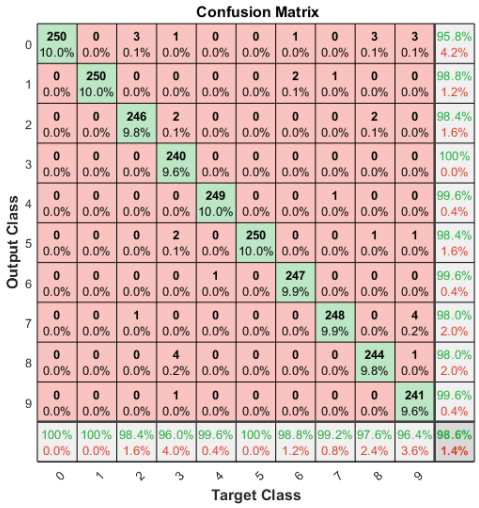
In this assignment we looked at the performance of SVM networks for use in pattern recognition with the MNIST digit database and compared the previously attained results from BPNN and CNN networks. The two different kernel type SVMs used in this assignment were able to provide comparable performance to the slightly better CNN while training in a much faster time and with much less model complexity.

7 References

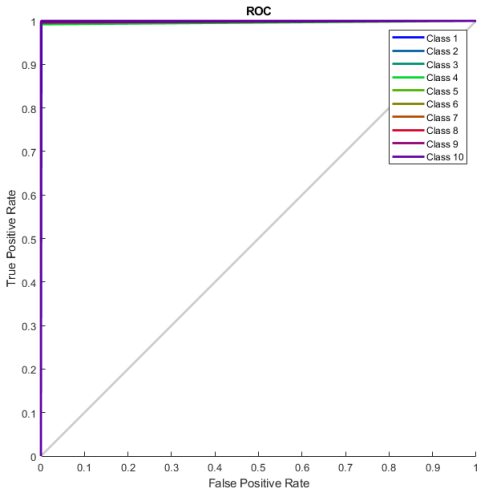
Neural Networks and Learning Machines
ECE656 Class Notes



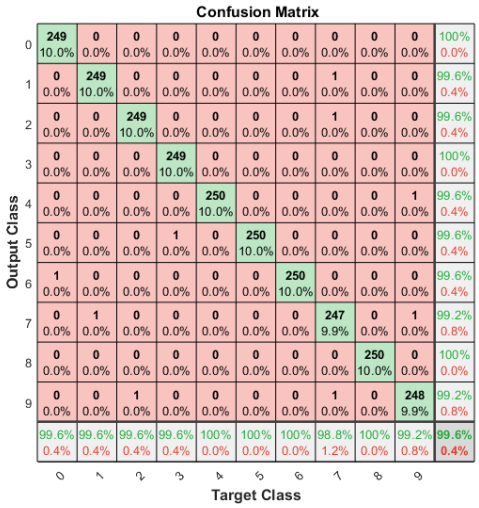
(a) ROC curve for 2 layer



(b) Confusion matrix for 2 layer



(c) ROC curve for 3 layer



(d) Confusion matrix for 3 layer

Figure 4: ROC curve and Confusion matrix for CNNs