ENTS669D: Introduction to Machine Learning

Project 02

Pavithra Ezhilarasan

UID: 114842763

Training the MNIST dataset using SVM:

METHOD:

An SVM, RBF kernel based classifier is used to fit the training data. Deciding on the parameters c and gamma is of prime importance. These 2 parameters determine how well the model performs.
C=2.8
Gamma=0.0073
The parameter values are obtained by performing a grid search. It is the search for the best C and gamma values from a range of c and gamma values
On running the script cvm_grid_search.py we get C=2.8 and gamma=0.0073 as the optimum values
Now, we fit the data on the above model.

PRE-PROCESSING:
Normalize the data by dividing the pixel values by 255

To evaluate the model we obtain a confusion matrix and test the accuracy.

The confusion matrix obtained:

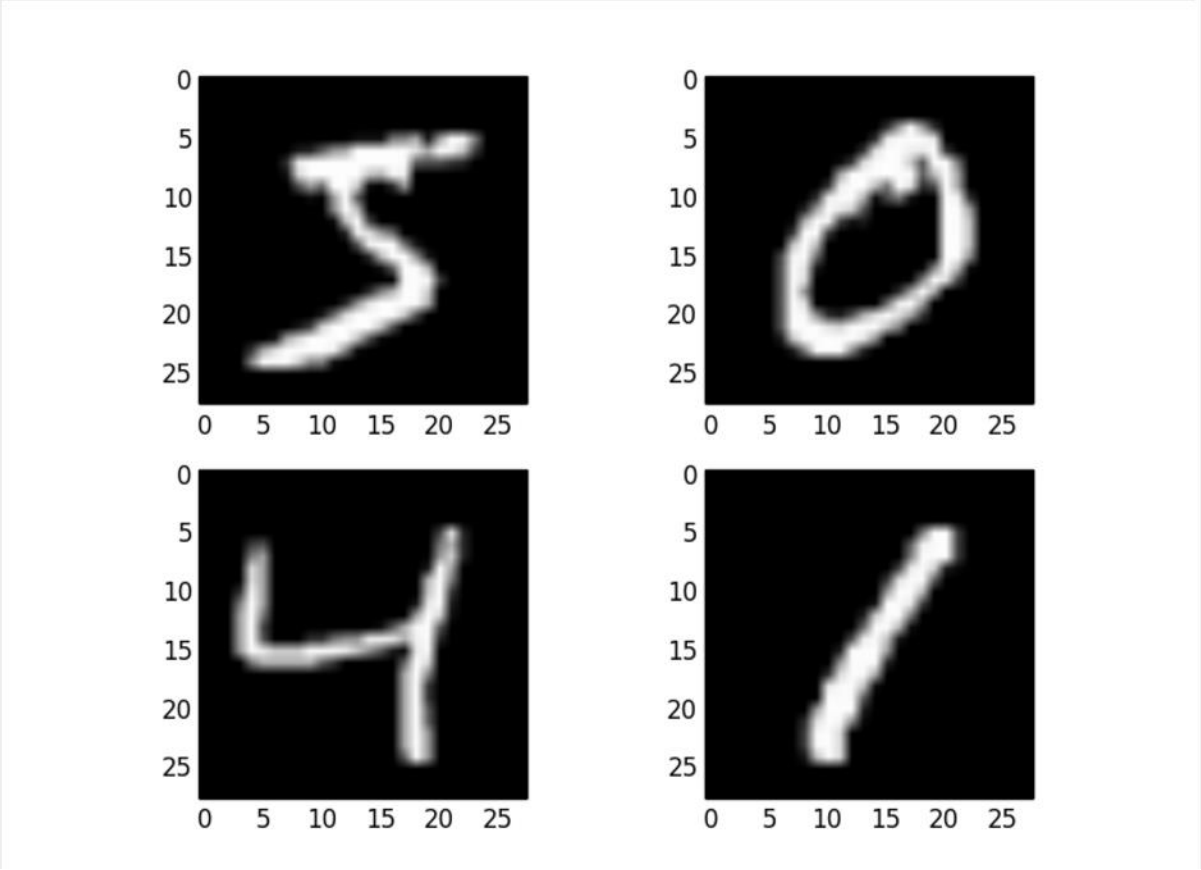
	0	1	2	3	4	5	6	7	8	9
0	2258	1	4	1	2	2	3	1	4	2
1	1	2566	9	1	1	0	0	7	3	0
2	4	1	2280	5	4	0	1	9	8	2
3	0	0	14	2304	1	13	0	6	8	2
4	2	2	2	0	2183	0	7	5	0	10
5	4	0	0	16	3	2026	12	1	4	3
6	7	5	3	0	5	2	2245	0	4	0
7	1	6	11	2	5	1	0	2373	5	13
8	3	9	4	9	4	10	2	3	2166	5
9	3	2	2	6	19	6	0	12	10	2329

RESULT:

The accuracy obtained is 98.4%

Training MNIST on Convolutional Neural Networks

Below is an example of the MNIST data as downloaded from the TensorFlow library.



The idea of a convolution neural network stemmed from the need to make image recognition transition invariant. It is the idea of training the image by diving it into many samples and making the model learn each of them. The images are trained using identical weights and the results are stored. This large array of results are down sampled by the method called max pooling. Now, a prediction can be done by treating the down sampled array as input to the neural network. The more number of convoluted layers, the better the model can recognize complicated features.

In this project, an architecture of 2 pairs of convolution, max-pooling networks, a fully connected network and a drop out network is implemented.

METHOD:

Convolution layer 1:
In this layer, we have one input, weight wc1 and bias bc1 which give an output.

Max pooling layer 1:
In this layer, the output is down sampled. K=2 is chosen. This means out of a square box of 2*2, the output that is maximum and with likely good information is stored away for the next stage computation. Experimentation in choosing k was done but it was found that the best accuracy was for k=2. As a higher k, say k=4 led to loss of valuable information leading to error in accuracy. The output is of size 32

Convolution layer 2:
In this layer, we have 64 inputs, weight wc2 and bias bc2 which give an output.

Max pooling layer 2:
The output from the convolution layer 2 is fed as input to the max pooling layer with k=2. This means out of a square box of 2*2, the output that is maximum and with likely good information is stored away for the next stage computation.

Fully connected neural network:
The input size is 7*7*64 inputs with weights wd1 and bias bd1. The output obtained is of size 1024

Dropout layer: This layer is added to prevent over fitting. This is done so by randomly making a few elements zero. The input to the dropout layer is 1024 and the output is the classification classes (0-9)

RESULTS:

The inputs are trained for 2000000 iterations with batch size of 128 for the 60000images. The training accuracy at the end of the 200000 iterations is at 98.438% whereas the testing accuracy is at 97.65%. The time duration for running all the iteration was about 20 minutes.

A separate result folder called results_svm.doc has the values for all the iterations. We can see the gradual increase in accuracy as the iterations are done.

FUTURE EXPERIMENTATION:

- After the fully connected layer, the output could be fed into an SVM and let it decide the classes. The experimentation could not be done due to lack of time. Grid search for C and gamma alone would take 2 days.
- Increasing the number of convolution layers in the architecture would increase the accuracy. If more convolution layers are added, the accuracy increases to more than 99% (with error % about 0.23%) for 35 convolutional networks as given on the official website. A GPU can handle training the dataset for so many convolution layers.

CONCLUSION:

The SVM and CNN methods were used to successfully classify the digits in the MNIST dataset.

LEARNING POINTERS FROM THE PROJECT:

- It is interesting to note that for a medium sized database (like MNIST), performing a grid search and later running SVM on the obtained parameters gives highly accurate results. The only drawback in this method is that it takes too long to perform the grid search. In that case, a convolution neural network of small number of layers, say 5 layers performs really well.
- Convolution neural network is a highly intuitive method of training the data by learning the features of different parts of the image by sliding a window.

- The max pooling method of down sampling ensures that the features of importance are preserved while decreasing the computational time.
- The major advantage of using a CNN over just an SVM is the computation time. CNN takes approximately 20 min and SVM takes about 2 days to run the grid search.