Assignment 3: Tensorflow

Kazi Injamamul Haque - 204873

January 2018

1 Introduction

The objective of this assignment is to get familiarize with how deep networks work in machine learning and to get hands on experience on one of the most used deep learning library, Tensorflow. It is very flexible and has the ability to handle a wide range of neural architects with addition of being cluster-friendly. For this assignment, we are using Tensorflow over OCR dataset in order to train a deep network model for classifying alphabets and predicting correct labels for the test dataset. Firstly, we used shallow softmax regression to build the model, secondly, we cross validated the softmax regression to increase the accuracy of our model by selecting the best hyperparameter. Finally, we applied deep architecture to increase the model accuracy even more. The baseline accuracy for the given OCR dataset is 0.75 and our objective is to avail a higher accuracy than the baseline accuracy using deep architecture. In the next section we discuss about the methodology, in section 3, we briefly discuss the deep architecture that we applied to increase the model accuracy. We present the results of our techniques in section 4. Finally we discuss the final outcome in the concluding section.

2 Methodology

The given OCR dataset was already divided into train and test sets with separate csv files for features and labels. The train set consists of 41,721 examples and labels whereas the test set has 10,431 examples and labels. The examples has 128 columns for 128 features because the OCR images are [16x8] bitmap images flattened into 128 features. On the other hand, the labels are just categorical labels (alphabets) ranging from "a" to "z", that is 26 classes. For training the model, we used softmax regression and then applied deep networks in order to acquire increased model accuracy. Although, the cross validation of shallow softmax regression gave us more than the baseline accuracy which is 0.77, we exploited deep architecture to increase the model accuracy even more and finally we achieved model accuracy of 0.86.

Firstly, we had to manipulate the given data in order make it more amenable to our model learning. In order to do that, the labels were encoded into a

"one-hot" representation. Each label for one given example was encoded into a vector of 26 (0 or 1) element, in which only the position associated with the true alphabet was set to 1 and all the other columns were set to 0. Therefore the label sets were tensors of shape (numberOfExamples, 26) whereas the feature sets were shape of (numberOfExamples, 128). Figure 1 shows one of the examples, alphabet - i, reshaped to [16x8] bitmap image. And the corresponding one-hot encoded label is (0,0,0,0,0,0,0,0,1,0.....,0,0).

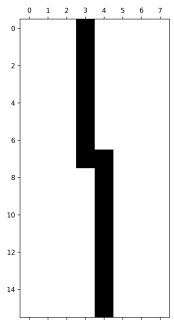

Figure 1: Visualization of one example (i) in the dataset

We used a shallow softmax regression to learn our model using tensorflow. It is a generalization of logistic regression to multi-class classification. In order to do that, we first declare the placeholders for our variables in the declarative step. Following the softmax regression, we declare variable x as dimension of [None, 128] where None will be replaced by the number of training examples and 128 depicts the number of features. W is declared with the dimension of [128, 26] where 128 is the number of features and we have 26 different kind of labels. Finally, the bias variable b is initialized with a vector of 26 elements. Furthermore, we have y_hat that will hold the predictions of our model. The vectorized form of the model looks like, y = softmax(W . x + b). Furthermore, in order to train our model, we used cross-entrpy loss and in the process of training, we selected gradient descent algorithm with a step-size of 0.5 and we tried to minimize the cross-entropy loss to get the training model. We acquired a model accuracy of 0.745 without any cross validation of the hyperparameter, step-size.

In the next step we used KFold cross-validation on the values of the hyperparameter, step-size of the gradient descent algorithm and we found out that the accuracy increases with the increasing value of the step-size until it converges after a certain value. In this case for step-size value 2.5, we obtained the best accuracy of 0.77, but the accuracy is somewhat similar to that of step-size value 1.5. Therefore, it converges after step-size value 1.5. Figure 2 shows the visualization of what our softmax regression model learned from the data and we can easily see that most of the alphabets are learned properly from the visualization.

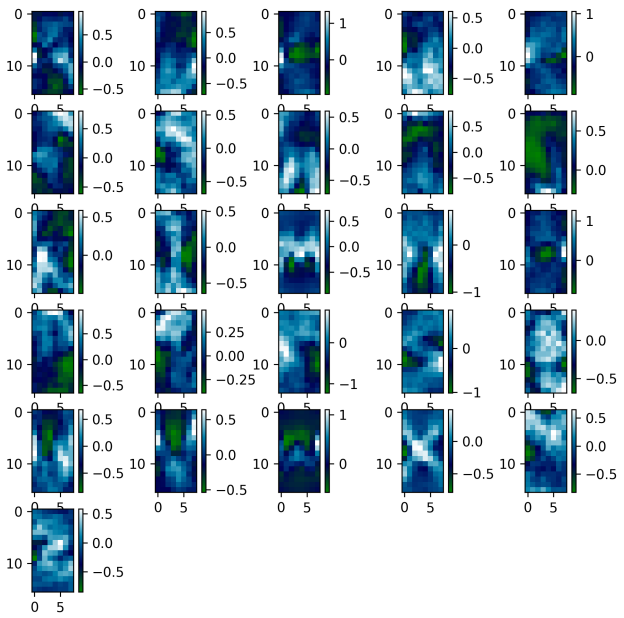

Figure 2: Visualization of learning by softmax regression

In order to increase the model accuracy, we exploited a deep architecture. The structure of the consists of convolutional layers, max-pool layers and a regularization layer and finally we obtainted a significant increase in the model accuracy and the final average model accuracy was 0.86. The structure of the deep architecture we used to obtain the desired result is briefly explained in the next section.

3 Deep Architecture

By defining a deep neural network we can significantly increase the performace of classification compared to the shallow softmax regression on image data. Figure 3 shows the schematic visualization of the neural network that we used for this assignment. The neural network consists of two convolutional layers and two max pool layers after each convolutional layer. Then a ReLU (Rectified Linear Unit) regularized with dropout. In the final stage, we again use a softmax layer in order to obtain the predictions. Dropout acts a regularization technique for the neural network by setting up a probability for dropping random neurons

from the network allowing us to create a more robust classifier where the model does not rely too much on a single neuron in the prediction stage. By using a dropout regularization, we decrease the chance of overfitting the model. Furthermore, the dropout probability is set to 1 in the prediction stage as we want to use the whole network for prediction. For training this deep neural network, we used ADAM SGD algorithm with a learing rate of 0.0001.

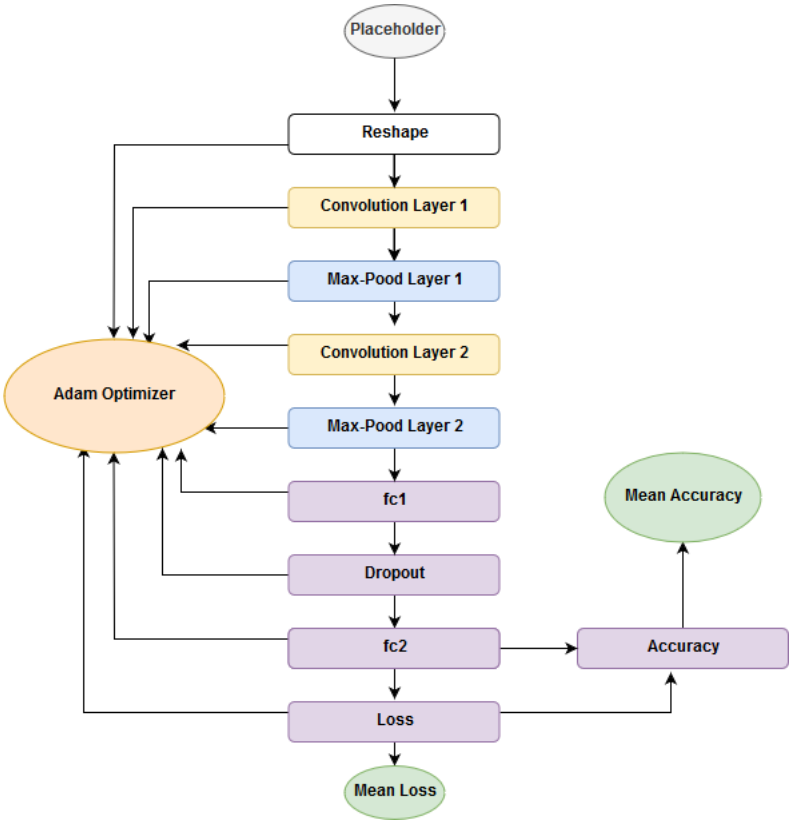

Figure 3: Deep architecture that was used

4 Results And Discussion

In this section we discuss about the obtained results with both cross-validated softmax regression and the result obtained from using the deep network.

4.1 With Cross Validated Softmax Regression

With the softmax regression classifier, we obtained a model accuracy of 0.74 with gradient descent algorithm with step-size 0.5. The test predictions are saved in

"test-pred1.txt" file. Furthermore, we used Kfold cross validation in order to find the best value for the step-size needed in gradient descent algorithm. We found that with increasing value for step size, the model accuracy increases until it converges after a certain value. In our case the accuracy converges after step-size of value 1.5. Table 1 shows the obtained result of cross validation on softmax regression. The final test predictions after using the best step-size value are saved in "test-pred2.txt" file.

Step-size	Accuracy
0.01	0.4493768
0.2	0.71294343
0.5	0.74362415
0.8	0.7543624
1.0	0.7586769
1.5	0.7645254
2.5	0.767977

Table 1: Cross-validation result for different step-size value for gradient descent

4.2 With Deep Architecture

Table 2 depicts the obtained result from using deep architecture with running 1000 epochs using 1000 examples mini-batch. We can see that by the end of the 1000 steps, the accuracy we obtained is 0.88 and the average accuracy obtained is 0.86, which is a significant increase in model accuracy. The test predictions for this process is saved in "predicted-targets.txt" file.

Step number	Accuracy
0	0.032999999821186066
100	0.5360000133514404
200	0.6840000152587891
300	0.7590000033378601
400	0.7879999876022339
500	0.8349999785423279
600	0.8190000057220459
700	0.8429999947547913
800	0.8569999933242798
900	0.8880000114440918
Avg. Test Accuracy	0.8612000346183777

Table 2: Accuracy result for deep architecture

5 Conclusion

Table 3 compares the results of the two classifiers, that are, shallow softmax regression and using the deep architecture mention in section 3. Although the softmax regression classifier gives us a better accuracy than the baseline accuracy, using a deep network significantly increases the model accuracy. This accuracy can be increased even more by selecting a different and more robust deep network and by tweaking the adam-optimizer value. In conclusion, we can say that using a deep architecture can significantly improve the classification performance when the classification is related to predict labels based on image data.

Classifier	Accuracy
Softmax Regression	0.767977
Using Deep Architecture	0.861200

Table 3: Comparison of the Classifiers