Assignment 3: Neural Networks

Francesco Penasa

December 21, 2019

1 Introduction

The third assignment of the Machine Learning course consist in familiarize with the deep networks concepts in machine learning. Deep Networks are state of the art techniques widely used for big variety of problems as image recognition and handwritten text recognition. In this assignment we will use one of the most used libraries of Python3 for machine learning and Neural Networks, TensorFlow2. The goal is to use TensorFlow2 to obtain an accuracy of at leat 0.75 for the OCR problem. In the next section we will discuss the code written step by step and we will show an image of the Deep Network created.

2 Methodology

At first we imported the train data and target and the test data and target in the `collab research` editor. We worked on the data splitting the training set in training and validating set (`80% and 20%`) and then modified the data reshaping the matrix and using the one hot encoding representation on the labels.

Secondly, we trained the model using softmax regression to experiment with the tensorflow library and we obtained an accuracy of `59.86%` on the validation set.

Then, in order to increase the model accuracy, we exploited a deep architecture structure. The neural network defined consists of three convolutional layers and three max pool layers. In the piece of code shown below are contained the shape of the input in the layer, the shape of the output and the number of weights.

```
                                          # input shape 8, 16, 1
self.conv1 = MnistConvolutional(1, 16, 5)  # out shape 8, 16, 16
self.pool1 = MaxPool2D([2,2])              # out shape 4, 8, 16
self.conv2 = MnistConvolutional(16, 32, 5) # out shape 4, 8, 32
self.pool2 = MaxPool2D([2,2])              # out shape 2, 4, 32
self.conv3 = MnistConvolutional(32, 48, 5) # out shape 2, 4, 48
self.pool3 = MaxPool2D([2,2])              # out shape 1, 2, 48

self.flatten = Flatten()                   # out shape 1*2*48
self.fc1 = MnistFullyConnected(1*2*48, 1024)# out shape 1024
self.dropout = Dropout(0.5)                # unchanged
self.fc2 = MnistFullyConnected(1024, 26)   # out shape 26
self.softmax = Softmax()                   # unchanged
```

In the image shown below it is displayed a simple diagram of the network architecture.

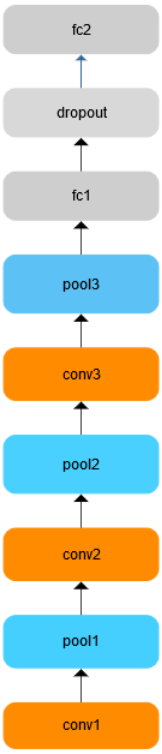

Figure 1: Schematic view of the architecture.

3 Results

In this section we will discuss and display the results obtained.

Using a shallow softmax regression to learn the model with TensorFlow2 we obtained an accuracy of `59.86%`, while defining a model using deep neural network we significantly increased the performance on the classification. To obtain the best results we have observed the accuracy on the validation set with three different learning rates (`1e-2, 1e-3, 1e-4`). Such results highlighted that the best learning rate is `1e-3` and it gives us an accuracy on the validation set of `91.73%` and an accuracy on the test set given the training on the whole set of `90.44%`

4 Conclusion

While the softmax regression classifier gives us an accuracy lower than the thresshold of `0.75%`, using a deep we can observe an accuracy of `90.44%` on the test set given the training on the whole training set.