Alexander Rusnak - SCIPER #309939

EE-559 - Deep Learning

Professor Francois Fleuret

May 22, 2020

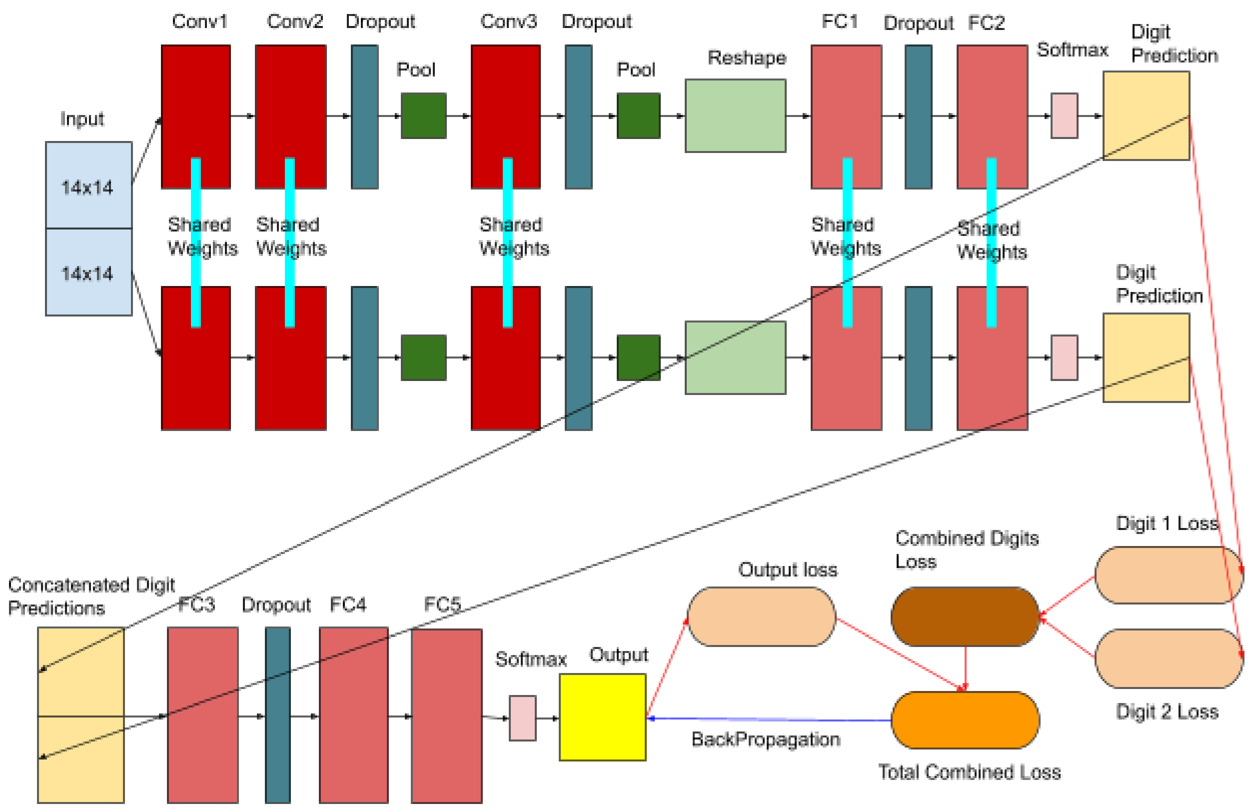
Mini-Project Report

Mini Project 1

In this project we were tasked with building a network which would correctly determine, given two 14x14 images of handwritten digits from the MNIST dataset, which of the two had a higher value. Within the context of this goal, we were also encouraged to implement a section with shared weights and an auxiliary loss function. The data we were given for this project was threefold: 1000 matrices, each representing two 14x14 MNIST images side by side, the associated integer values for each MNIST image, and a label indicating which of the two integers had a higher value.

Initially I approached this task by separating it into two subsections in order to better understand an efficient architecture implementation; going from a handwritten digit to the integer values and going from a pair of integer values to predicting which is larger. I picked a convolutional neural network to use for the handwritten digit, and a short series of fully connected layers to use for the pairs of integers. Both utilized relu activations, an adam optimizer, and cross entropy to determine a cost function. After finding an efficient and effective architecture for these subtasks, I combined them to create one larger model to solve the whole problem.

The larger model differed from a straightforward linear combination of the above in two important ways. First off, I separated the MNIST images into two separate images and ran them through the first half of the network separately but concurrently. This allowed the convolutional layer to optimize for learning just one digit and become more accurate, faster. Secondly, I implemented an auxiliary loss at that point by returning the two predicted integer values along with the final output of the model (which of the two was larger). I then combined the loss for each integer prediction with the loss for the final output (at a weighting of 50% final output, and 25% for the loss of each image digit prediction) which had a substantially increased accuracy. Below is a diagram of this architecture, plus a table of the losses of the different models.



Mini Project 2

In this project we were tasked with creating a deep learning framework from scratch, and then implementing a network to classify cartesian plane coordinates as being inside or outside of a predefined circle. More specifically we had to implement these particular modules: a fully connected layer module that could take forward and backward passes and could return its parameters, a relu and tanh activation function module, mean squared error, and a sequential function that built a model from these components. The model should have an input and output layer, as well as three middle layers each with 25 units.

In working on this project, I had the most difficulty implementing backpropagation in an effective manner, as it seemed that the weights would almost always either catch momentum and end up fitting to only one output independent of input or settling at predicting a middling value in between 0 and 1 with an accuracy of ~50%. In order to rectify this, I implemented sigmoid activation for the final layer, a dropout function to increase generalization, and Xavier weight initialization to combat vanishing/ exploding gradients. Adding each of these improved the number of correct predictions. The next component I wanted to implement would have been a regularizer. In the end, I was not able to achieve very high accuracy with my model, which was quite disappointing. The best combination used Tanh activation, Xavier initialization, dropout rate of 0.2, a learning rate of 0.00001, and trained for 40 epochs, which achieved ~70% accuracy.

Overall, I found both of these projects highly rewarding as I sharpened my skills with and knowledge of neural networks. Particularly the first project improved my model composition ability and creativity using different losses. The second project deepened my understanding of the backpropagation and the inner working of weight and bias functions. Though I found it relatively difficult working on my own rather than with a team (my initial teammates dropped the class after COVID-19 moved us online), I think it ultimately made my learning substantially more robust. In summation, I was able to implement both projects successfully and independently.