**CNN Model for Stock Market Prediction**

**Improving the CNN model**

There are certain classic CNN networks that have shown success in the past. Few famous ones are: LeNet-5, AlexNet, VGG-16 etc. With time and with the development of computational power, these architectures have been able to show higher accuracy in case of computer vision applications. In this project, I have explored two CNN architectures: LeNet-5 and AlexNet in order to improve the model:

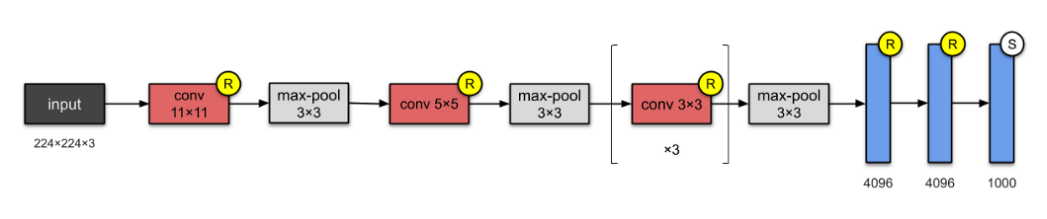
**LeNet-5**

The LeNet-5 has a convolution layer (5x5) followed by average pooling layer, and another convolution layer (5x5) and an average pooling layer. This is followed by 3 fully connected layers with 120, 84 and (number of output) training neurons. Tanh activation functions are used everywhere except for the last layer.

In my implementation of the LeNet-5 for stock market prediction, I first used a convolution layer (1x82) with 8 filters for extracting most important features. This was followed by convolution layer with size (3x1) for extracting durational features. I did not use 5x5 as previous literature uses three consecutive days in case of financial market studies. Instead of average pooling layer I used max pooling. This was followed by another set of convolution and max pooling layers. The droprate was set to 0.1 and the layers flattened to be input into the dense network with 120, 84 and 1 number of neurons. Tanh was used in all the activation layers except for the last where sigmoid was used.

The F1 score was 0.614 which was lower. However, the F1 score on the training set was higher (0.8603 compared to 0.8541). This indicates the model may have overfit the data. Possible remedial measures would be to use L1/L2 regularizers or increase the droprate.

**AlexNet**



The AlexNet architecture has 5 convolutional layers followed by 3 fully connected layers. ReLU activation is used in each of the layers.

For my implementation of the AlexNet, I used two convolutional layers of sizes (1x82) and (3x1) with 8 filters. This was followed by a max pooling layer of size (2x1). Then, I followed it with another set of convolutional and max pooling layer, followed by 3 convolutional layers each of size (3x1) and a max pooling layer of size (2x1). This was followed by 3 dense layers of sizes 4096, 4096 and 1. ReLU activation functions was used in all the layers.

The F1 score for this model vastly improved to 0.699 for the test set.