**CNN PAPER**

The main aim of the paper is to showcase and understand CNN’s utility as an efficient ML algorithm which if put to the correct model and relevant variables can show significant improvements in performances compared to the baseline algorithms.

An input can be considered as a function on which the convolution operation needs to be done using filters, weights, biases which are updated during the process of the training.

Usually after the operation, the output is passed through an activation function before being passed as an input to the next layer.

A pooling layer is responsible for subsampling the data.It is a nice way of handling the overfitting problem so that it can generalise.All the values inside a single pooling window is converted to a single value.

A fully connected layer is responsible for converting extracted features in the previous layers to the final output.

The goal is to predict future prices of a stock mainly using two types of CNNs. 2D and 3D CNNs.

In 2D CNN we shall find a general model that is valid for several markets.However, to extract the desired mapping function, that model needs to be trained by samples from different markets.All this information is aggregated and fed to the 2D model as a two dimensional tensor.

In 3D CNNpred, we assume that different models are needed for making predictions in different markets, but each prediction model can use history of many markets.It extracts features from the historical information of many markets and uses them to train a separate prediction model for each market.It uses a tensor with three dimensions.One can expect the 3D-CNN to be able to combine information from different markets into high level features before making predictions.

2D

The size of the input matrix will be d\*f where d is the number of days and each day is represented by f variables. To extract daily features, f filters are utilised. Each covers all the daily variables and can combine into a single higher level feature.It can also drop some useless variables by setting corresponding weights to zero.What follows next are several convolutional, pooling layers which subsequently go on and keep compressing the input size every time and ends with a fully connected layer giving an output.

3D

The idea in 3D is more or less the same but there is an extra dimension along with d and f called I for different markets.

In both cases our output will be a sigmoid function which can give a number between 0-1 as the probability of an increase in the market for the next day, hence a wise move would be to put more money on such stocks and short sell those which have a probability of going down.