Deep Learning for Time Series Analysis

Shuo HAN

Supervised by Prof. James Tin-Yau KWOK

Introduction

Deep learning, which refers to a set of algorithms in machine learning that aim at mimicking the learning mechanism of human brains, has become increasingly popular nowadays. The word 'deep' refers to the relatively large number of learning layers, where concepts are extracted from lower layers and used as input features for upper layers. Similar to the way that human cognition works, the concepts in upper layers are more abstract than those in lower layers. In recent years, deep learning methods have achieved fairly high performance in object recognition, speech recognition, and natural language processing, etc.

However, deep learning has not been widely applied on time series data, especially on financial data such as stock prices. In this thesis, we tried to analyze a set of financial time series data with CNNs as our fundamental modeling method, since we found that such data had an important property called translation invariance, which meant the shape of a financial time series remains unchanged if shifted to another time. We could make good use of CNNs to study this crucial property.

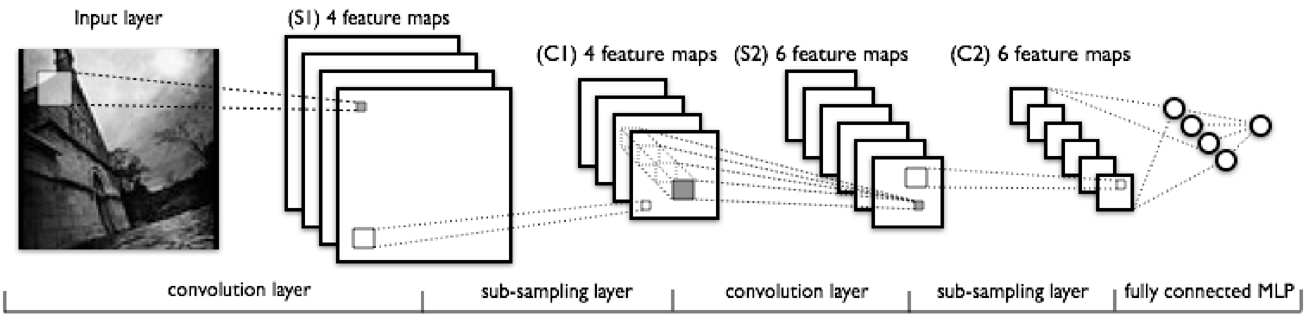

Figure 1: A CNN for image recognition.

We also made use of CNNs to build a convolutional autoencoder, an unsupervised learning model that aims to learn better and more concise representation of the original data, in order to extract transient but significant perturbations from stock price changes.

The Dataset

The dataset we used was a set of financial time series data that contained 24-hour prices of a stock with ticker symbol “CL” in 411 days between September 1, 2012 and September 30, 2014. Prices were extracted from the candlestick chart of this stock, so there were four numbers representing the opening, high, low and closing prices for each particular hour of each day. Furthermore, we labeled each day by the criterion of whether it was a plunge or not.

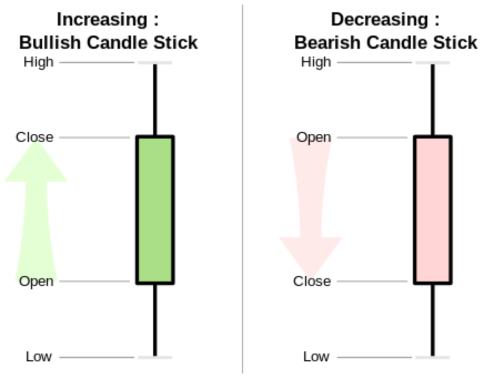

Figure 2: Scheme of two candlestick charts.

The Network Architecture

There are two convolution and two pooling layers alternating as lower layers. A fully connected multilayer perceptron is used as upper layers. The first convolution layer and pooling layer transform the 1×24 input vector into a 1×10 vector. The second convolution layer and pooling layer transform the 1×10 input vector into a 1×3 vector. The fully connected MLPs with 20 neurons in the hidden layer accept the 1×3 vector as the input and output a 1×2 vector indicating the classification result.

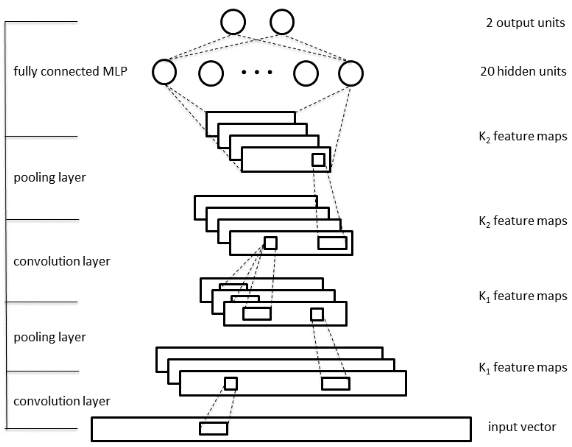

Figure 3: The overall architecture of our CNN model.

Convolotional autoencoder is an expanded concept of the autoencoder, which was first served as an unsupervised learning method that aimed at learning a more concise representation of original data by setting the target values to be equal to the inputs in a neural network. The autoencoder tries to learn an approximation to the identity function $h_{W,b}(x) \approx x$, so that the output is similar to x.

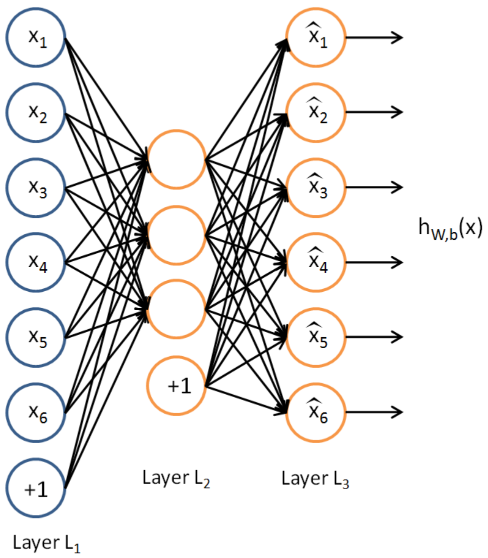

Figure 4: The autoencoder in this figure will learn a simplified representation of the input data after training.

Results

The evaluation results are shown in the table below. Our CNN model outperformed other methods with the given dataset.

Model	Accuracy	*precision*	*recall*	F_1 score
DTW + k-NN[1]	0.932	N/A	N/A	N/A
SVM with the RBF kernel[2]	0.951	N/A	N/A	N/A
CNN without pre-training	**0.954**	**0.683**	**0.776**	**0.722**
CNN using tanh activations (without pre-training)	0.920	0.496	0.629	0.548
CNN with pre-training on the first ConvPool layer	0.942	0.669	0.566	0.591
CNN with pre-training on both ConvPool layers	0.927	0.537	0.530	0.505

[1] k-NN is the abbreviation of "k-nearest neighbors algorithm". Here we used 7-NN.
[2] Support vector machine (SVM) with the radial basis function (RBF) kernel.

Conclusion

In this thesis, we built CNN models to analyze financial time series data and showed that CNNs were superior in this task. However, the convolutional autoencoders we used did not generate useful representations of the original data, and they did not improve the prediction results either. The evaluation result suggests the need of further studies in related topics