INVENTORY MANAGEMENT

JEETHAN
Department of computer Science PES University

JOEL
Department of computer Science PES University

GOWTHAM
Department of computer Science PES University

Abstract:
Inventory forecasting is a key component of effective inventory management. In this work, we utilise hybrid deep learning models for inventory forecasting. According to the highly nonlinear and non-stationary characteristics of inventory data, the models employ Long Short-Term Memory (LSTM) to capture long temporal dependencies and Convolutional Neural Network (CNN) to learn the local trend features. However, designing optimal CNN-LSTM network architecture and tuning parameters can be challenging and would require consistent human supervision. To automate optimal architecture searching of CNN-LSTM, we implement three meta-heuristics: a Particle Swarm Optimisation (PSO) and two Differential Evolution (DE) variants. Computational experiments on real-world inventory forecasting problems are conducted to evaluate the performance of the applied meta-heuristics in terms of evolved network architectures for obtaining prediction accuracy. Moreover, the evolved CNN-LSTM models are also compared to Seasonal Auto-regressive Integrated Moving Average (SARIMA) models for inventory forecasting problems. The experimental results indicate that the evolved CNN-LSTM models are capable of dealing with complex nonlinear inventory forecasting problems.

Introduction:

Effective planning is important in production systems that aim at effective management and coordination of related activities and resources for an organization. A common objective in such systems is to achieve optimal production planning and inventory management to meet (often variable) products demand over the planning horizon. This paper tackles the inventory forecasting problem for highly perishable food with a very limited shelf life and with variable customers demand. Much research effort has been invested in developing accurate and robust inventory prediction models. This problem has been treated from various angles, such as time-series, pattern recognition , clustering. Generally, from machine learning point of view, those models can be classified into parametric or non-parametric methods . Parametric methods assume a certain statistical distribution on the data and primarily rely on statistical techniques such as auto-regressive moving average models, linear and nonlinear regressions . These techniques try to detect a function between the past information and the predicted state. Non-parametric approaches do not assume that the data is following any distribution and adopt computational intelligent methods such as fuzzy systems, neural networks, and evolutionary computation.

BACKGROUND:

There are two major approaches for predicting and analyzing the time series data (i)ARIMA ,(ii)LSTM. ARIMA is a form of regression analysis that indicates the strength of a dependent variable relative to other changing variables. The final objective of the model is to predict future time series movement by examining the differences between values in the series instead of through actual values.LSTM assumes that there are input values (time series) which are to be used to predict an output value. Since the time series data only had an input series, the stock price value from time t-1 was used as input for predicting the stock price value from time t as the output.

DATA PREPROCESSING:

We had some extra columns in in superstore.csv(out dataset) which are not of any help much in the process of data analysis of the dataset. And we had renamed the columns of each file as the name of the column contains space, and uppercase letters so we will correct it to make it easy to use.Each algorithm has constraints, which has caused us to sample the dataset.

Operations like Displaying the distribution of dates in the 'Order Date' column and displaying the distribution dates in the 'Ship Data' column.And sorting the data by order date which helps in analyzing the data more efficiently.

Libraries used:numpy ,pandas,os ,math,seaborn,datetime,matplotlib.pyplot.

DATA ANALYTICS:

The primary goal of EDA is to support the analysis of data prior to making any conclusions. It may aid in the detection of apparent errors, as well as a deeper understanding of data patterns, the detection of outliers or anomalous events, and the discovery of interesting relationships between variables.
In the Fig1.1 we plot the sales distribution from the year 2014-2018 to understand the frequency of the item or product sold in the specific year.

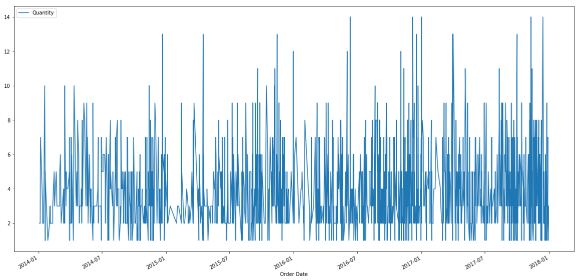


Fig1.1

In the fig 1.2 we check for the factors responsible in overall volume trade and plot the correlation matrix in a heatmap to understand better.

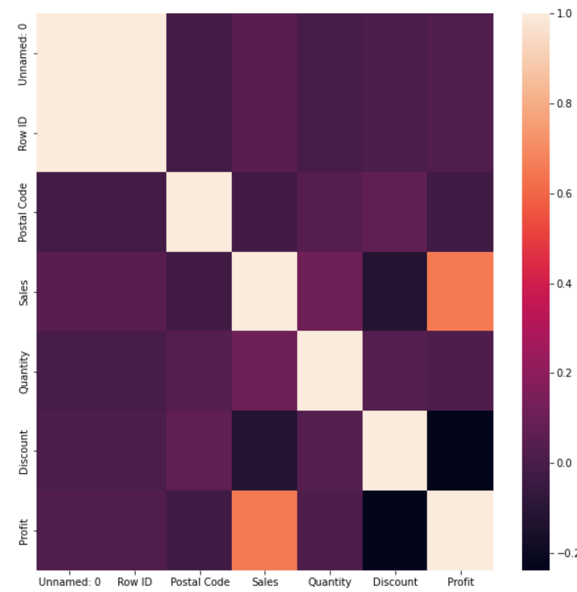
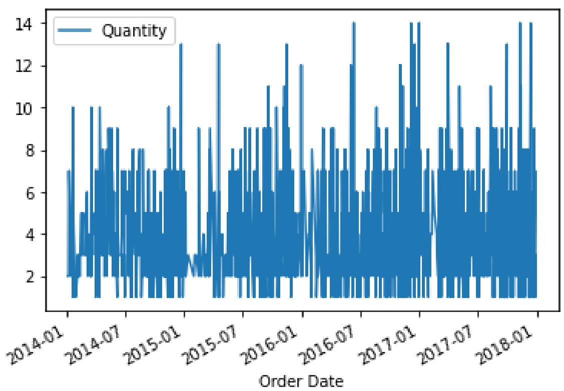


Fig1.2

In the Fig1.3 we plot the sales distribution from the year 2015-2018 to understand the frequency of the item or product sold in the specific year for arima model.



MODEL AND RESULTS

CNN

This layer is the first layer that is used to extract the various features from the input images. In this layer, the mathematical operation of convolution is performed between the input image and a filter of a particular size MxM. By sliding the filter over the input image, the dot product is taken between the filter and the parts of the input image with respect to the size of the filter (MxM).

Computing the ReLU

This function has two major advantages over sigmoidal functions such as sigma(x) or tanh(x).

1. ReLU is very simple to calculate, as it involves only a comparison between its input and the value 0.
2. It also has a derivative of either 0 or 1, depending on whether its input is respectively negative or not.

The latter, in particular, has important implications for backpropagation during training. It means in fact that calculating the gradient of a neuron is computationally inexpensive:

relu’(x)={0,for x<0

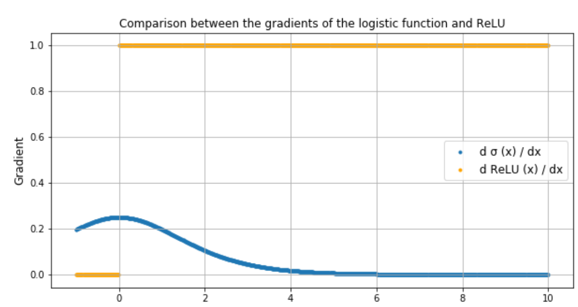
1,for >=0}

Non-linear activation functions such as the sigmoidal functions, on the contrary, don’t generally have this characteristic.

As a consequence, the usage of ReLU helps to prevent the exponential growth in the computation required to operate the neural network. If the CNN scales in size, the computational cost of adding extra ReLUs increases linearly.

ReLUs also prevent the emergence of the so-called “vanishing gradient” problem, which is common when using sigmoidal functions. This problem refers to the tendency for the gradient of a neuron to approach zero for high values of the input.

While sigmoidal functions have derivatives that tend to 0 as they approach positive infinity, ReLU always remains at a constant 1. This allows backpropagation of the error and learning to continue, even for high values of the input to the activation function:



LSTM:

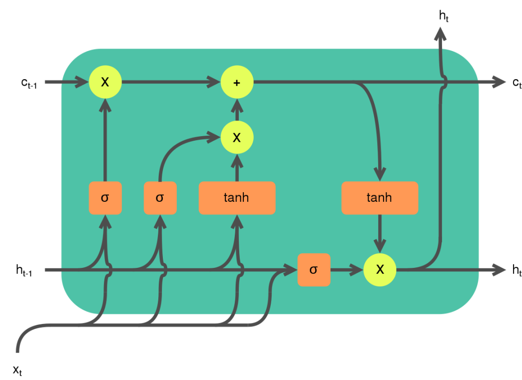
LSTM (Long Short-Term Memory) is a Recurrent

Neural Network (RNN) based architecture that is widely used in natural language processing and time series forecasting.

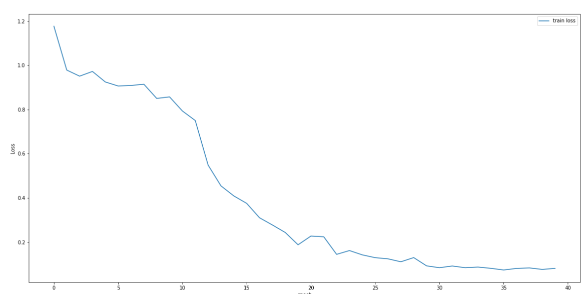
The LSTM rectifies a huge issue that recurrent neural networks suffer from: short-memory. Using a series of 'gates,' each with its own RNN, the LSTM manages to keep, forget or ignore data points based on a probabilistic model.

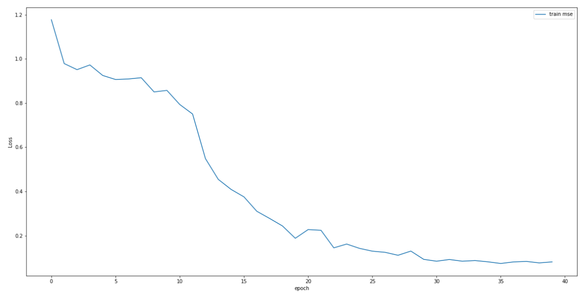
LSTMs also help solve exploding and vanishing gradient problems. In simple terms, these problems are a result of repeated weight adjustments as a neural network trains. With repeated epochs, gradients become larger or smaller, and with each adjustment, it becomes easier for the network's gradients to compound in either direction. This compounding either makes the gradients way too large or way too small. While exploding and vanishing gradients are huge downsides of using traditional RNN's, LSTM architecture severely mitigates these issues.

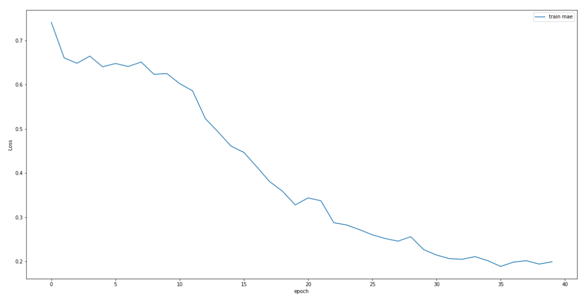
After a prediction is made, it is fed back into the model to predict the next value in the sequence. With each prediction, some error is introduced into the model. To avoid exploding gradients, values are 'squashed' via (typically) sigmoid & tanh activation functions prior to gate entrance & output. Below is a diagram of LSTM architecture:

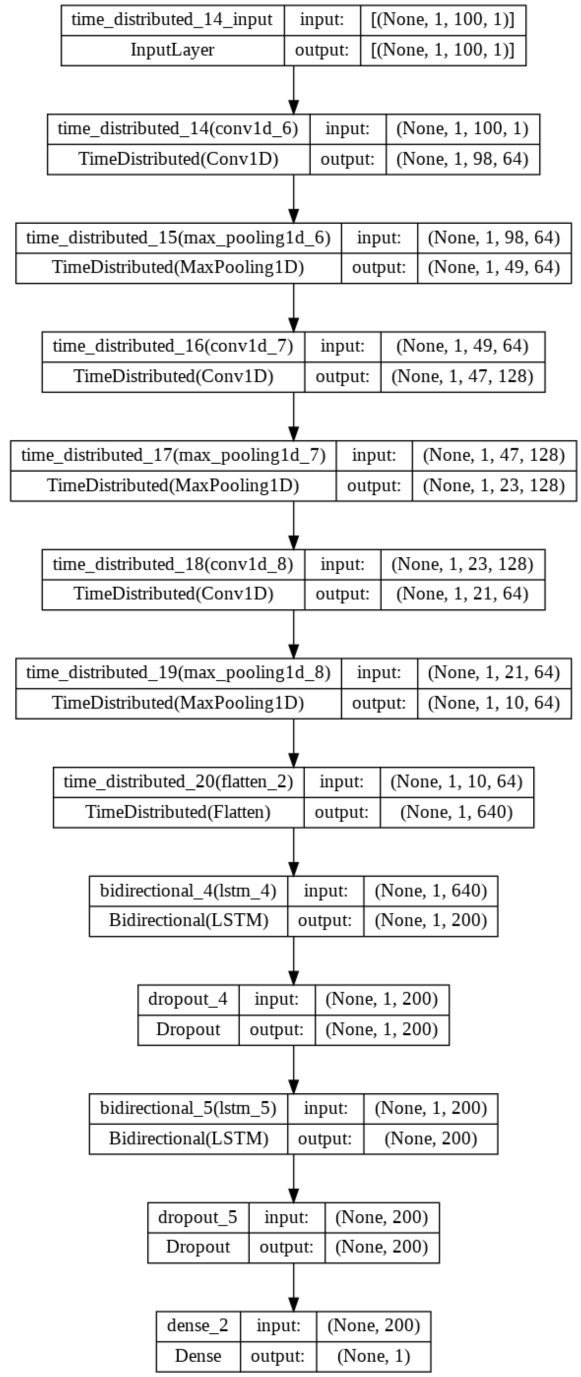


While building the model using LSTM the losses in the error is shown in the below diagrams:









After we built the model using CNN and LSTM summarizing the input and output and layers.

Comparing the real stock price and predicted stock price: