Veena Mounika Ruddarraju (ID: 220269453)

Yarraguntla Sai Swetha (ID:220268894)

CSc 215-01 Artificial Intelligence

Project #2

Mini-Project 2: Time Series Forecasting using NN, LSTM and CNN
Due Date: 10/09/2019

Problem Statement

Time series data is a sequence of data points, measuring the same thing over time, stored in time order. In a time series, time is often the independent variable and the goal is usually to make a forecast for the future. Time series aids us to understand the past behavior and would be helpful for future predictions. In this project we work on the time series data to predict the stock price.

Methodology

Fully connected neural network:

The dataset has 7 columns initially. They are Data, Open, High, Low, Close, Adj_Close and Volume. We started the project with data preprocessing stage. Initially we checked for null values or any special characters and came to know that it is a clean dataset. We then dropped date column and Adj_close column which are not required. Since value to be predicted is 'close' value, we made a copy of 'close' variable to consider it as the output variable. Then we moved on to data transformation phase where we normalized all the input features using z-score. We then converted x and y into a format that should be given as input to neural networks. After that we split the dataset as 70% training data and 30% testing data.

We used a function to_sequences to prepare an input with 7 * 5 = 35 input features to predict the y value. It is then reshaped into (rows,35).

We then trained the models using x_train and y_train data with different activation functions and optimizers. We also used ModelCheckpoint and EarlyStopping while training the model. We used combinations of Relu, Sigmoid, Tanh activation functions with Adam and Sgd optimizers. At this stage while testing the model with Relu activation function and Sgd optimizer, it displayed the loss values as 'nan' and didn't gave RMSE value. Then after reading some articles we came to know that this problem arises while using sgd optimizer when the loss value moves out of bound and it can be resolved by reducing number of layers and number of neurons. Then after trying with less number of layers and neurons, it gave an RMSE value.

LSTM:

Initially, to input data to an LSTM model, we reshaped it into a 3D. Each record can be viewed as a sequence of 7 vectors, each vector with 5 dimensions.

Trained the models using x_train and y_train data with hidden layers and different activation functions. We used combinations of relu, sigmoid and tanh activation functions with Adam and sgd optimizers. We tried different parameters to improve the accuracy and to get the best model. We added some dropout layers to prevent overfitting. We tried using different epochs. The problem which we faced in the neural networks also arise in LSTM. While testing the model with relu activation function with sgd optimizer without dropout layer it gave the RMSE value. We followed the same steps by reducing number of layers and neurons count. But when we tested the model with relu activation function and sgd optimizer with dropout layer, it was giving loss value as nan. We tried different combinations but didn't get through it.

CNN:

In CNN we give x as a 4D vector . Here we tried for both (rows,1,7,5) and (rows,7,1,5) which almost have similar values. The CNN model is built using dropout layer and without dropout layers and different kernel size. We used combinations of relu, sigmoid and tanh activation functions with Adam and sgd optimizers. Initially we started with 10 and 20 epochs but results were not good so increased to 1000 epochs. Out of all the combinations tried we got the best results for the model with relu activation function in both convolution and dense layers and adam optimizer.

Experimental Results and Analysis

Neural Networks:

MSE values

Activation function/optimizer	adam	sgd
Relu	934.75	915.86
Sigmoid	914.99	914.19
Tanh	914.89	917.79

RMSE values:

Activation function/optimizer	adam	sgd
Relu	30.57	30.26
Sigmoid	30.24	30.23
Tanh	30.24	30.29

LSTM

Dropout layer	LSTM neuron	Hidden layer neurons	epochs	# layers	Activation function of LSTM	Activation function on dense	Optimizer	MSE	RMSE
wthout	**64**	**32**	**20**	**2**	**relu**	**relu**	**Adam**	**6.82**	**2.61**
wthout	64	32	10	2	Relu	relu	adam	50.05	7.07
with	64	32	20	4	relu	relu	adam	11.26	3.35
with	64	32	10	4	relu	relu	adam	19.24	4.38
wthout	16	-	20	1	relu	-	sgd	3057415.2	1748.54
with	16	-	20	1	relu	-	sgd	-	-
wthout	64	32	20	2	sigmoid	sigmoid	adam	2330.35	48.27
with	64	32	20	4	sigmoid	sigmoid	adam	2202.45	46.93
wthout	64	32	20	2	sigmoid	sigmoid	sgd	1002.19	31.65
with	64	32	20	4	sigmoid	sigmoid	sgd	1168.14	34.17
wthout	64	32	20	2	tanh	tanh	adam	1621.69	40.27
with	64	32	20	4	tanh	tanh	adam	1572.11	39.64
wthout	64	32	20	2	tanh	tanh	sgd	1195.86	34.58
with	64	32	20	4	tanh	tanh	sgd	1253.8	35.41

CNN

kernel	Conv layer activation function	Dense activation function	optimizer	MSE	RMSE
(3,3)	relu	sigmoid	adam	1203.27	34.68
(4,4)	**relu**	**relu**	**adam**	**4.29**	**2.07**
(4,4)	sigmoid	relu	adam	2075.02	45.5
(5,5)	sigmoid	relu	adam	1275.01	35.70
(5,5)	tanh	tanh	adam	192.20	13.86
(5,5)	tanh	tanh	rmsprop	1409.90	37.54

Task Division and Project Reflection:

Veena Mounika: Neural Networks, LSTM

Swetha: Neural networks, CNN

Challenges:

1.Initially we took lot of time to figure out how to convert or reshape into required format like 2D,3D

2.The model with relu activation function and sgd optimizer displays loss value as nan when we did some research about that particular problem we knew that when loss value exceeds the bound value it displays as nan .To overcome that problem we removed some layers an reduced some neurons. Both in neural networks and LSTM.

3. Initially when we did neural networks taking (Open, Volume, High, Low) as x and Close as y we got RMSE values as 0.2, but later when we took 35 features as input and y as output the RMSE value was around 30.

4.We then figured out it that neural network is not a good model to handle time series data.

What we have learned from this project?

1.We learnt how to handle time-series data and also use it for CNN model

2.learnt how to reshape and give it as input to neural networks, CNN and LSTM.

3. How to improve performance with algorithm tuning.

4. Usage of the models.

5. In CNN out of all the combinations tried we got the best results for the model with relu activation function in both convolution and dense layers and adam optimizer.