1. LSTM with Keras

This exercise builds on what was done last week. We used simpleRNNs to forecast Netflix closing price. In this exercise, we use LSTMs and GRUs for forecasting.

a) We want to forecast the future sales using LSTM.
Understand the API documentation at https://keras.io/api/layers/recurrent_layers/lstm/.

b) The LSTM layer has 50 neurons.

c) The LSTM layer is followed by an appropriate Dense layer. The dense layer has activation function relu.

d) For the network you implemented, manually calculate the number of parameters and size of the layers. Does it match the output of model.summary()?

e) Train the model using the training data. Test the model using test data. Try different number of epochs, number of neurons and layers. Can you improve the accuracy of the forecast?

f) Extend your model to include multiple features for forecasting.

2. GRU with Keras

g) We want to forecast the Netflix closing price using GRU.
Understand the API documentation at https://keras.io/api/layers/recurrent_layers/gru/.

h) Change LSTM to GRU layer.

i) Can you explain the change in the number of parameters?

j) Train the model using the training data. Test the model using test data. Try different number of epochs, number of neurons, and layers see if you could forecast sales more accurately.

3. Follow the tutorial at https://www.tensorflow.org/text/tutorials/text_generation for text generation using a dataset of Shakespeare's writing.
Given a sequence of characters from this data ("Shakespear"), train a model to predict the next character in the sequence ("e").
Generative power of RNNs:
Longer sequences of text can be generated by calling the model repeatedly.
Optional: Write your own generative RNN to predict next character prediction for your choice of author.

Field Code Changed