

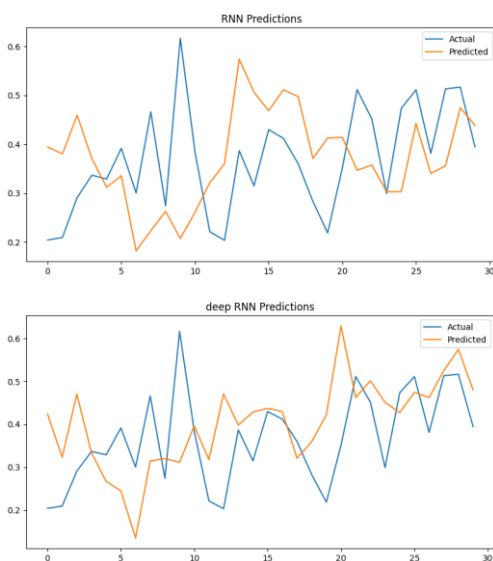
Recurrent Neural Network by JUSTINE Jean using chatGPT on 15/12

I. RNN AND DEEP RNN FOR FORECASTING

For this task, I asked chatgpt to create a time series dataset because it was quicker than downloading one. After that I split the data into validation, test and train set. Then I created two function build RNN, and build deep RNN, to build the two neural networks. I build them and compile then train them with as a compiler: optimizer adam, and loss mse then I trained it with 10 epochs, and 32 batch size.

For the deep rnn I used the same optimizer and loss metric. Compare to the normal RNN, they all have 2 layers of SimpleRNN with activation tanh and a dense of 1 neuron as the output layer.

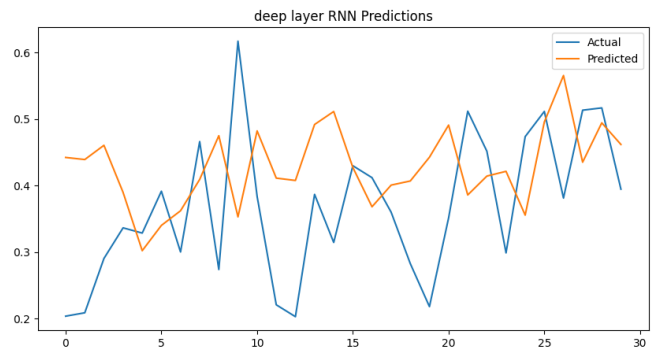
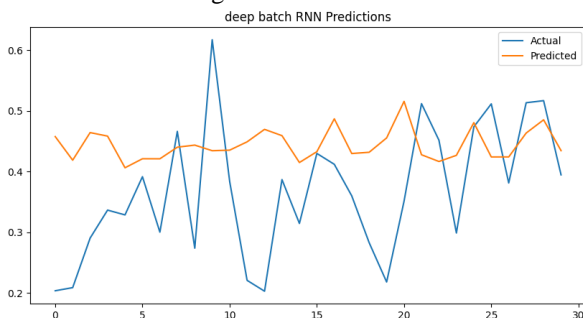
As results I had:



We can witness that the curves are a bit chaotic and the models struggle to capture the overall trend of the data, there is slight improvement in the deep RNN model but it still struggles and is not accurate enough.

II. BATCH AND LAYER FOR DEEP RNN

After applying batch and layer to our previous deep RNN, I got those curve:

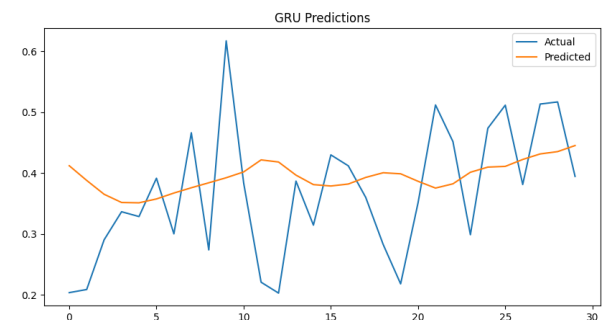
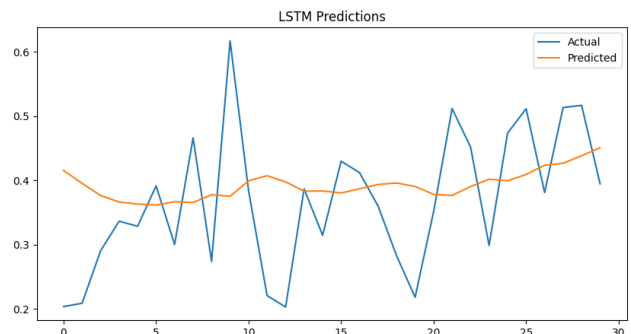


We have slightly better results than with just deep RNN or RNN but that still don't quite follow the line and struggle to understand enough but the layer RNN has a better understanding.

III. LSTM AND GRU

After training those models, we do the exact same with LSTM and GRU model, we create function to create them, we compile then train them over the dataset we created. We use the GRU and LSTM function from keras to create layers in those two models with as the two previous model, adan optimizer, mse as loss, the same input shape, and one dense layer as output.

We had those results :



The results are way better than the other, but it lacks to align with the actual value, as it does not do peaks as the original data, the prediction is too smooth in overall.

IV. RESULTS

We had not good enough results, we can improve them a lot by doing tuning with grid search. This task is already

answered in the first two steps as it was asking to plot the results.

But for the error indicator I had that

- RNN Metrics: {'Train MAE': 0.10993952834402347, 'Train MSE': 0.019594754563765485, 'Test MAE': 0.11954358860758378, 'Test MSE': 0.021371464195411517}
- Deep RNN Metrics: {'Train MAE': 0.09179691981634412, 'Train MSE': 0.012547953329823931, 'Test MAE': 0.10304202046583494, 'Test MSE': 0.01783236256272509}
- Deep RNN batch Metrics: {'Train MAE': 0.14846871679687382, 'Train MSE': 0.03546669902586556, 'Test MAE': 0.10949069769498429, 'Test MSE': 0.017766353529641184}
- Deep RNN layer Metrics: {'Train MAE': 0.10016317921548552, 'Train MSE': 0.016524627763499646, 'Test MAE': 0.11643107332528053, 'Test MSE': 0.01930475259824395}
- LSTM Metrics: {'Train MAE': 0.09869937682365562, 'Train MSE': 0.015952384102262386, 'Test MAE': 0.09059311866838944, 'Test MSE': 0.012177866447143772}
- GRU Metrics: {'Train MAE': 0.10922512301168992, 'Train MSE': 0.01939623697968636, 'Test MAE': 0.09029466396938252, 'Test MSE': 0.012232758244003125}

To conclude we can still witness that LSTM has the lowest error follow by GRU and that those are optimum model for this type of data

V. CHORAL

For this task we were asked to download the dataset and then train a model on it to predict the next notes on a piano, for that I used the code from hands on machine learning chapter 15 that was given to us, using a convolution model.

So firstly I download the dataset and then split it into train valid and test.

Then I took the code of hands on machine learning to create function that can play the different song, to later test our predictions.

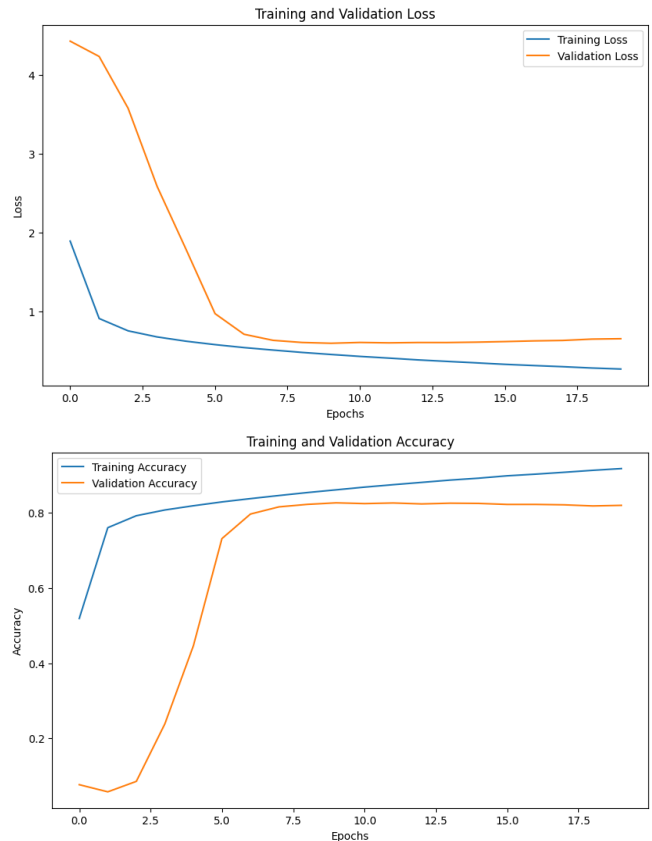
After that I took the same code again, to transform the data into set that the model can learn on.

Finally, I create a model with 4 convolution 2D layers, with a batch normalization between each of them, an LSTM then Dense layer as the two last and Embedding layer as the first.

After I compile it with a loss of `sparse_categorical_crossentropy` and Nadam as optimizer. I

train it with 20 epochs and the validation and train set. I save the model, and then I tried to predict and create our own music using the model. I asked chatgpt to do those function as I didn't have any idea to how to do it.

In the end, our music that we predicted over data that we took in the dataset as a seed to start, the music is arguably correct, it depends on the taste of people.



I plotted the results of the models, and we can see it is learning well but have a slight overfitting after 9 epochs compare to the validation set but the results were good using convolution and LSTM.

VI. CONCLUSION

To conclude, this assignment was really helpful to understand different type of deep RNN, and RNN itself, we learn when to use which type of deep RNN, and on what data it can be used.

Finally, I used chatgpt to create function in the last task to listen and predict music, I also used it to plot some graph, the rest of the 4 task come mainly from hands on machine learning chapter 15 https://github.com/ageron/handson-ml2/blob/master/15_processing_sequences_using_rnns_and_cnns.ipynb. I also used chatgpt in case of error and on which parameters and number of layers was optimum