



ANSHUL VATS

104491909

COS30018 – INTELLIGENT SYSTEMS

REPORT TASK-B.4

Summary of Implementing the Function to Create Deep Learning Models

Objective is to implement a function that dynamically constructs deep learning models based on user-friendly parameters, and use this function to experiment with different deep learning networks and hyperparameters.

Dynamic Model Creation-

The function `create_model` is designed to create a deep learning model dynamically based on various configuration parameters:

Parameters:

- `sequence_length`: Number of time steps in each sequence.
- `n_features`: Number of features in the input data.
- `units`: Number of units in each LSTM layer.
- `cell`: Type of recurrent cell (LSTM, GRU, etc.).
- `n_layers`: Number of layers in the model.
- `dropout`: Dropout rate for regularization.
- `loss`: Loss function used for training.
- `optimizer`: Optimizer used for training.
- `bidirectional`: Whether to use bidirectional layers.

Explanation-

Explanation:

- **Sequential Model**: Creates a linear stack of layers.
- **Layers**:
 - **Input Layer**: Uses Bidirectional wrapper if `bidirectional=True`.
 - **Hidden Layers**: Added based on `n_layers` parameter. Uses Dropout for regularization.

- **Output Layer:** A Dense layer with a single output unit and linear activation.
- **Compilation:** Configures the model with specified loss function and optimizer.

Training the model

Code part - " history = model.fit(data["X_train"], data["y_train"], epochs=EPOCHS, batch_size=BATCH_SIZE, validation_data=(data["X_test"], data["y_test"]), verbose=1, callbacks=[TrainingPlotCallback()])"

Explanation:

- **Training Data:** X_train and y_train for training; X_test and y_test for validation.
- **Epochs and Batch Size:** Specified by EPOCHS and BATCH_SIZE.
- **Callback:** TrainingPlotCallback is used to plot training and validation loss.

Screenshots of the Outputs

```
632e-04 - val_mean_absolute_error: 0.0182
Epoch 10/500
86/86 [=====] - 21s 241ms/step - loss: 5.4715e-04 - mean_absolute_error: 0.0163 - val_loss: 6.2
067e-04 - val_mean_absolute_error: 0.0171
Epoch 11/500
86/86 [=====] - 21s 248ms/step - loss: 6.4354e-04 - mean_absolute_error: 0.0184 - val_loss: 3.9
589e-04 - val_mean_absolute_error: 0.0156
Epoch 12/500
86/86 [=====] - 21s 247ms/step - loss: 5.3529e-04 - mean_absolute_error: 0.0164 - val_loss: 4.0
008e-04 - val_mean_absolute_error: 0.0134
Epoch 13/500
86/86 [=====] - 21s 247ms/step - loss: 5.5169e-04 - mean_absolute_error: 0.0169 - val_loss: 3.7
962e-04 - val_mean_absolute_error: 0.0123
Epoch 14/500
86/86 [=====] - 21s 247ms/step - loss: 6.0696e-04 - mean_absolute_error: 0.0175 - val_loss: 3.7
737e-04 - val_mean_absolute_error: 0.0132
```

The script will then begin training the models using the stock data. During training, you will see the training loss and validation loss for each epoch.

```

Enter the start date (YYYY-MM-DD): 2020-01-01
Enter the end date (YYYY-MM-DD): 2023-12-31
2024-09-07 11:57:59.204627: I tensorflow/core/platform/cpu_feature_guard.cc:182] This TensorFlow binary
use available CPU instructions in performance-critical operations.
To enable the following instructions: SSE SSE2 SSE3 SSE4.1 SSE4.2 AVX AVX2 FMA, in other c
with the appropriate compiler flags.
Model: "sequential"

```

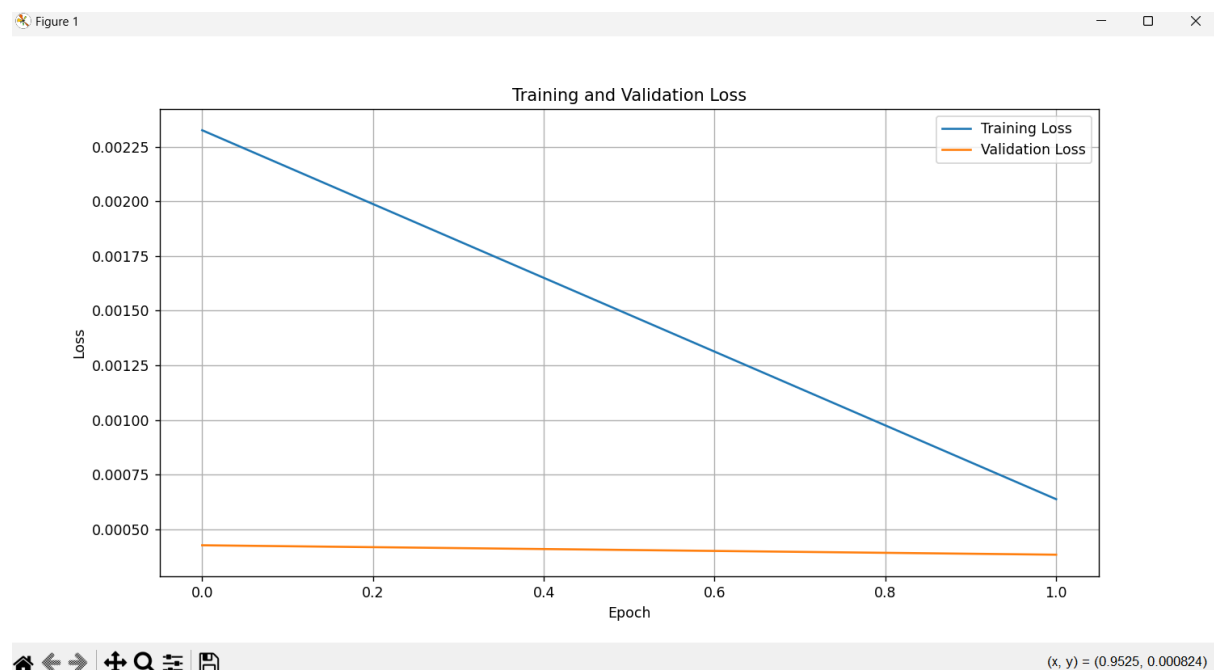
Layer (type)	Output Shape	Param #
lstm (LSTM)	(None, 50, 256)	268288
dropout (Dropout)	(None, 50, 256)	0
lstm_1 (LSTM)	(None, 256)	525312
dropout_1 (Dropout)	(None, 256)	0
dense (Dense)	(None, 1)	257

```

=====
Total params: 793857 (3.03 MB)
Trainable params: 793857 (3.03 MB)
Non-trainable params: 0 (0.00 Byte)
=====
None

```

For each model configuration, it will print a summary of LSTM model being constructed. This will look like as above.



During the training process, after each epoch, a plot will be displayed showing the training and validation loss over time. A new plot will appear after each epoch ends.

Problems –

The major problem I got is switching between the figures, now in this code am getting the expected output but not able to switch between the candlestick chart or the boxplot in the command prompt terminal.

Another thing is, initially I was not getting the plotting of the graph then I make some modifications on the code by taking help from some online sources.

Apart from these things, everything was fine.

Thank You.