COS30018 – Intelligent System

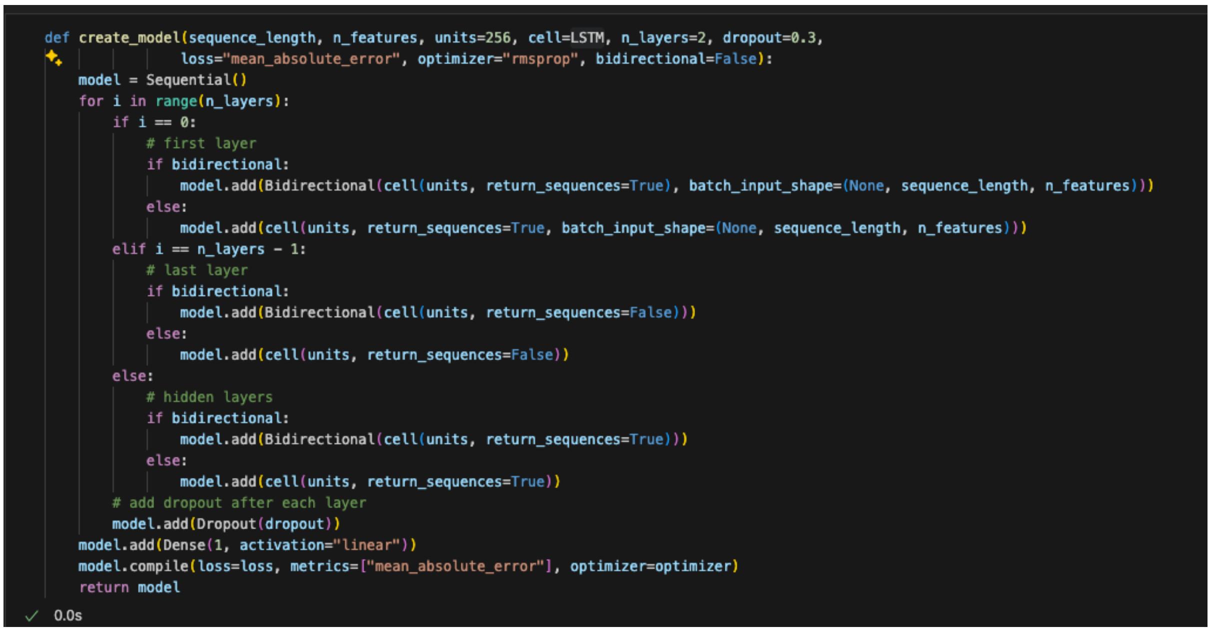
Option B: Stock Price Prediction

Report v0.4

Name Phong Tran
Student Id:104334842

Class: 1-5
Tutor: Tuan Dung Lai

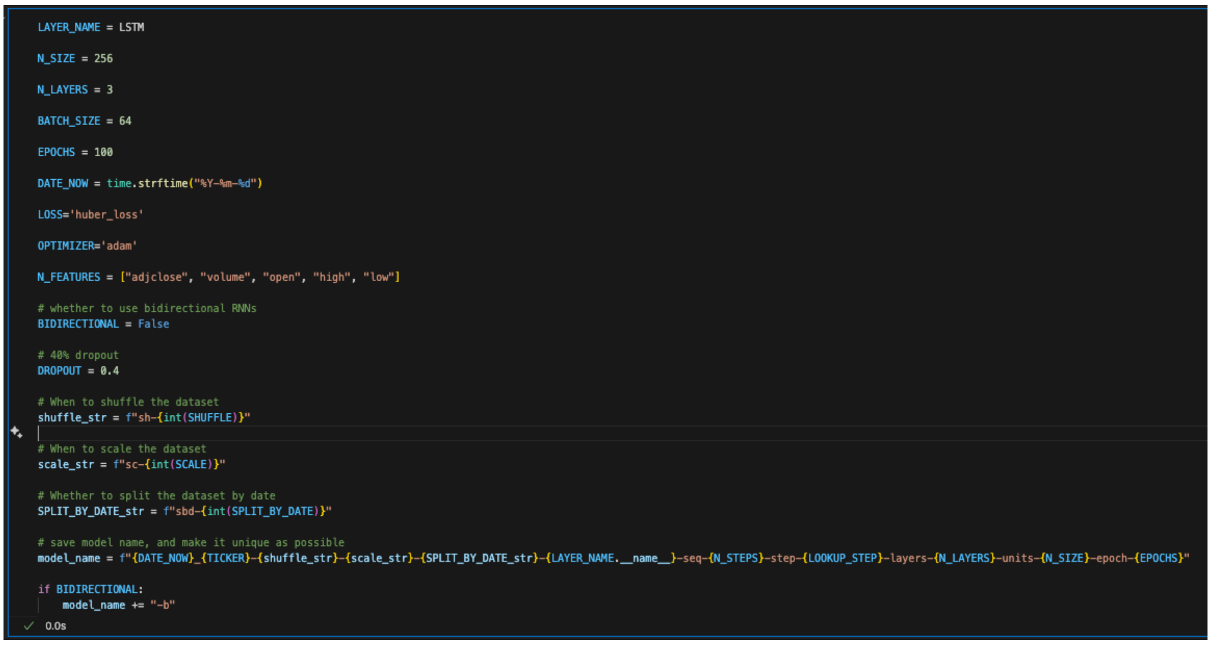
Summary function to create DL models



```python
def create_model(sequence_length, n_features, units=256, cell=LSTM, n_layers=2, dropout=0.3,
                loss="mean_absolute_error", optimizer="rmsprop", bidirectional=False):
    model = Sequential()
    for i in range(n_layers):
        if i == 0:
            # first layer
            if bidirectional:
                model.add(Bidirectional(cell(units, return_sequences=True), batch_input_shape=(None, sequence_length, n_features)))
            else:
                model.add(cell(units, return_sequences=True, batch_input_shape=(None, sequence_length, n_features)))
        elif i == n_layers - 1:
            # last layer
            if bidirectional:
                model.add(Bidirectional(cell(units, return_sequences=False)))
            else:
                model.add(cell(units, return_sequences=False))
        else:
            # hidden layers
            if bidirectional:
                model.add(Bidirectional(cell(units, return_sequences=True)))
            else:
                model.add(cell(units, return_sequences=True))
        # add dropout after each layer
        model.add(Dropout(dropout))
    model.add(Dense(1, activation="linear"))
    model.compile(loss=loss, metrics=["mean_absolute_error"], optimizer=optimizer)
    return model
✓ 0.0s
```

After creating model at putting input and output into sequence data (Sequential) I create for loop depending on the number layers we want to add, inside that there are three layers inside with choice of whether include bidirectional or simple layer. At the end of the loop, add the Dropout layer into sequential model. After the loop, add Dense layer with unit and linear activation, then complie the model with same loss and metrics, and use optimiser that we want (like adam). Finally, return model in overall function.

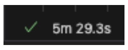
Summary experiments of different configuration of DL models and model training



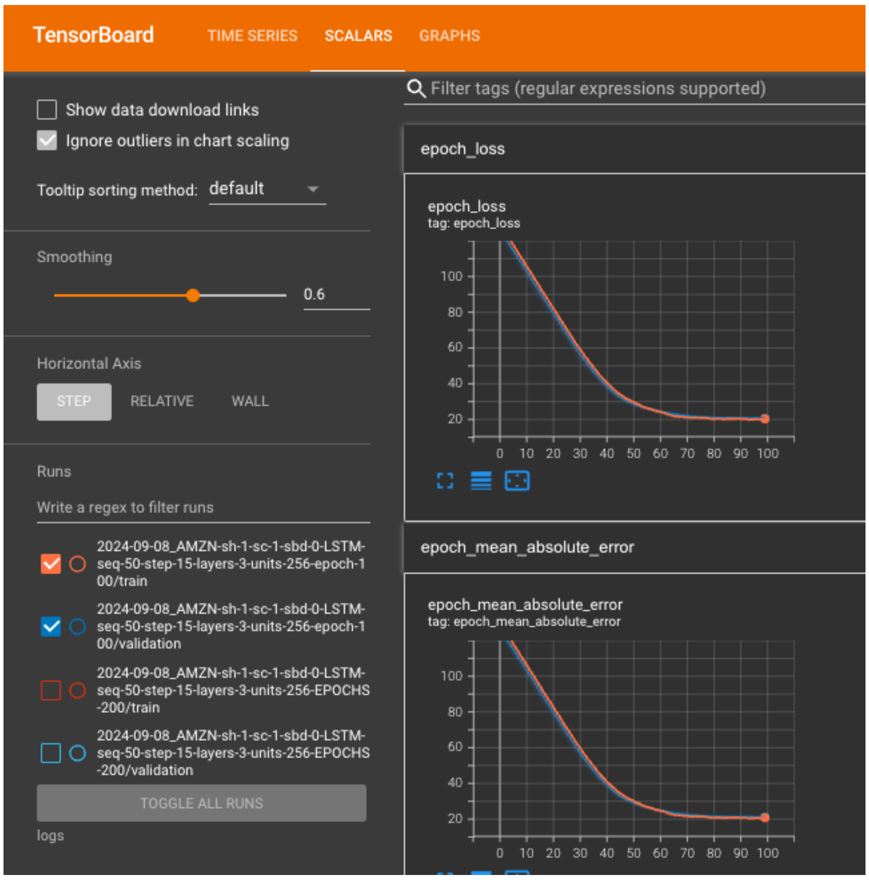
```python
LAYER_NAME = LSTM

N_SIZE = 256

N_LAYERS = 3

BATCH_SIZE = 64

EPOCHS = 100

DATE_NOW = time.strftime("%Y-%m-%d")

LOSS='huber_loss'

OPTIMIZER='adam'

N_FEATURES = ["adjclose", "volume", "open", "high", "low"]

# whether to use bidirectional RNNs
BIDIRECTIONAL = False

# 40% dropout
DROPOUT = 0.4

# When to shuffle the dataset
shuffle_str = f"sh-{int(SHUFFLE)}"

# When to scale the dataset
scale_str = f"sc-{int(SCALE)}"

# Whether to split the dataset by date
SPLIT_BY_DATE_str = f"sbd-{int(SPLIT_BY_DATE)}"

# save model name, and make it unique as possible
model_name = f"{DATE_NOW}_{TICKER}-{shuffle_str}-{scale_str}-{SPLIT_BY_DATE_str}-{LAYER_NAME.__name__}-seq-{N_STEPS}-step-{LOOKUP_STEP}-layers-{N_LAYERS}-units-{N_SIZE}-epoch-{EPOCHS}"

if BIDIRECTIONAL:
    model_name += "-b"
```
✓ 0.0s

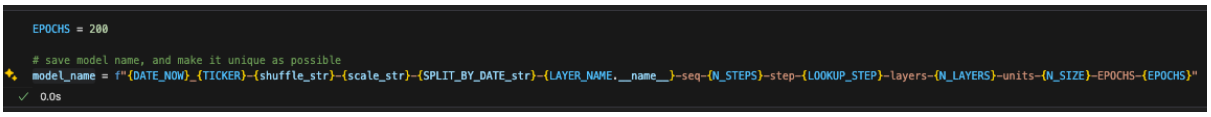
Experiment with the first one in details:

- Layer name: LSTM
- Size number: 256
- Layer number: 3
- Batch size: 64
- Epochs: 100
- LOSS: huber_loss
- Optimizer: adam
- Bidirectional: False
- Dropout: 0.4

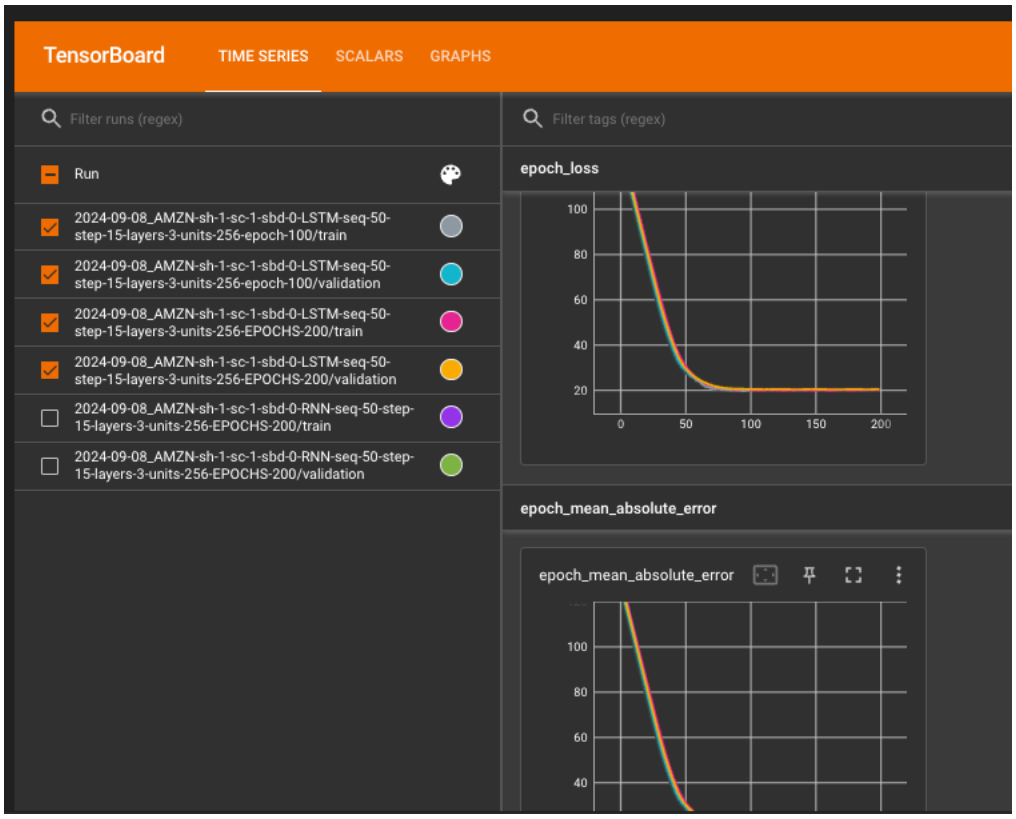

✓ 5m 29.3s

Training takes 5 minutes 29 seconds


TensorBoard
TIME SERIES
SCALARS
GRAPHS
Filter tags (regular expressions supported)
Show data download links
Ignore outliers in chart scaling
Tooltip sorting method: default
Smoothing
0.6
Horizontal Axis
STEP
RELATIVE
WALL
Runs
Write a regex to filter runs
2024-09-08_AMZN-sh-1-sc-1-sbd-0-LSTM-seq-50-step-15-layers-3-units-256-epoch-100/train
2024-09-08_AMZN-sh-1-sc-1-sbd-0-LSTM-seq-50-step-15-layers-3-units-256-epoch-100/validation
2024-09-08_AMZN-sh-1-sc-1-sbd-0-LSTM-seq-50-step-15-layers-3-units-256-EPOCHS-200/train
2024-09-08_AMZN-sh-1-sc-1-sbd-0-LSTM-seq-50-step-15-layers-3-units-256-EPOCHS-200/validation
TOGGLE ALL RUNS
logs
epoch_loss
epoch_loss
tag: epoch_loss
epoch_mean_absolute_error
epoch_mean_absolute_error
tag: epoch_mean_absolute_error
The graph show the curve line from 0 to 80 before decrease slightly till 100

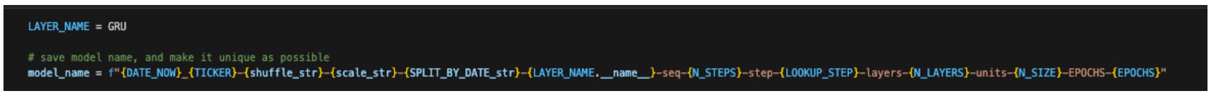

```
EPOCHS = 200

# save model name, and make it unique as possible
model_name = f"{DATE_NOW}_{TICKER}-{shuffle_str}-{scale_str}-{SPLIT_BY_DATE_str}-{LAYER_NAME.__name__}-seq-{N_STEPS}-step-{LOOKUP_STEP}-layers-{N_LAYERS}-units-{N_SIZE}-EPOCHS-{EPOCHS}"
✓ 0.0s
```
Second experiment with different Epochs, others keep the same, reducing to 200 epochs

```
[56]  ✓  11m 3.6s
```
Training takes 11 minutes 3 seconds



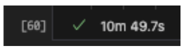
2024-09-08_AMZN-sh-1-sc-1-sbd-0-LSTM-seq-50-step-15-layers-3-units-256-epoch-100/train
2024-09-08_AMZN-sh-1-sc-1-sbd-0-LSTM-seq-50-step-15-layers-3-units-256-epoch-100/validation
2024-09-08_AMZN-sh-1-sc-1-sbd-0-LSTM-seq-50-step-15-layers-3-units-256-EPOCHS-200/train
2024-09-08_AMZN-sh-1-sc-1-sbd-0-LSTM-seq-50-step-15-layers-3-units-256-EPOCHS-200/validation
2024-09-08_AMZN-sh-1-sc-1-sbd-0-RNN-seq-50-step-15-layers-3-units-256-EPOCHS-200/train
2024-09-08_AMZN-sh-1-sc-1-sbd-0-RNN-seq-50-step-15-layers-3-units-256-EPOCHS-200/validation

The graph shows the time series in 200 epoches, with decreasing significantly from 0 to more than 50, before stabilising till the end.



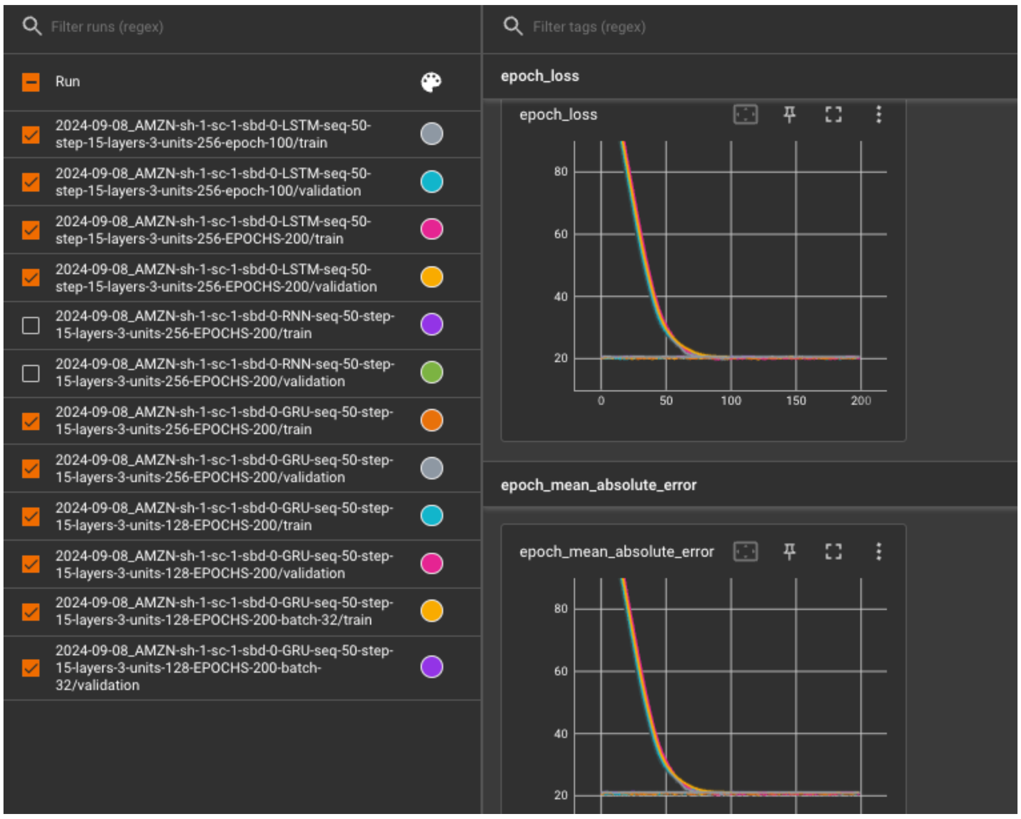
```
LAYER_NAME = GRU

# save model name, and make it unique as possible
model_name = f"{DATE_NOW}_{TICKER}-{shuffle_str}-{scale_str}-{SPLIT_BY_DATE_str}-{LAYER_NAME.__name__}-seq-{N_STEPS}-step-{LOOKUP_STEP}-layers-{N_LAYERS}-units-{N_SIZE}-EPOCHS-{EPOCHS}"
```

Third experiment with different layer name, use GRU layer name



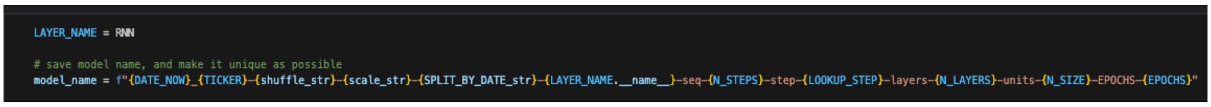
```
[60]  ✓  10m 49.7s
```

Training takes 10 minutes 49 seconds



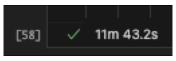
- [x] 2024-09-08_AMZN-sh-1-sc-1-sbd-0-LSTM-seq-50-step-15-layers-3-units-256-epoch-100/train
- [x] 2024-09-08_AMZN-sh-1-sc-1-sbd-0-LSTM-seq-50-step-15-layers-3-units-256-epoch-100/validation
- [x] 2024-09-08_AMZN-sh-1-sc-1-sbd-0-LSTM-seq-50-step-15-layers-3-units-256-EPOCHS-200/train
- [x] 2024-09-08_AMZN-sh-1-sc-1-sbd-0-LSTM-seq-50-step-15-layers-3-units-256-EPOCHS-200/validation
- [ ] 2024-09-08_AMZN-sh-1-sc-1-sbd-0-RNN-seq-50-step-15-layers-3-units-256-EPOCHS-200/train
- [ ] 2024-09-08_AMZN-sh-1-sc-1-sbd-0-RNN-seq-50-step-15-layers-3-units-256-EPOCHS-200/validation
- [x] 2024-09-08_AMZN-sh-1-sc-1-sbd-0-GRU-seq-50-step-15-layers-3-units-256-EPOCHS-200/train
- [x] 2024-09-08_AMZN-sh-1-sc-1-sbd-0-GRU-seq-50-step-15-layers-3-units-256-EPOCHS-200/validation
- [x] 2024-09-08_AMZN-sh-1-sc-1-sbd-0-GRU-seq-50-step-15-layers-3-units-128-EPOCHS-200/train
- [x] 2024-09-08_AMZN-sh-1-sc-1-sbd-0-GRU-seq-50-step-15-layers-3-units-128-EPOCHS-200/validation
- [x] 2024-09-08_AMZN-sh-1-sc-1-sbd-0-GRU-seq-50-step-15-layers-3-units-128-EPOCHS-200-batch-32/train
- [x] 2024-09-08_AMZN-sh-1-sc-1-sbd-0-GRU-seq-50-step-15-layers-3-units-128-EPOCHS-200-batch-32/validation

The graph shows



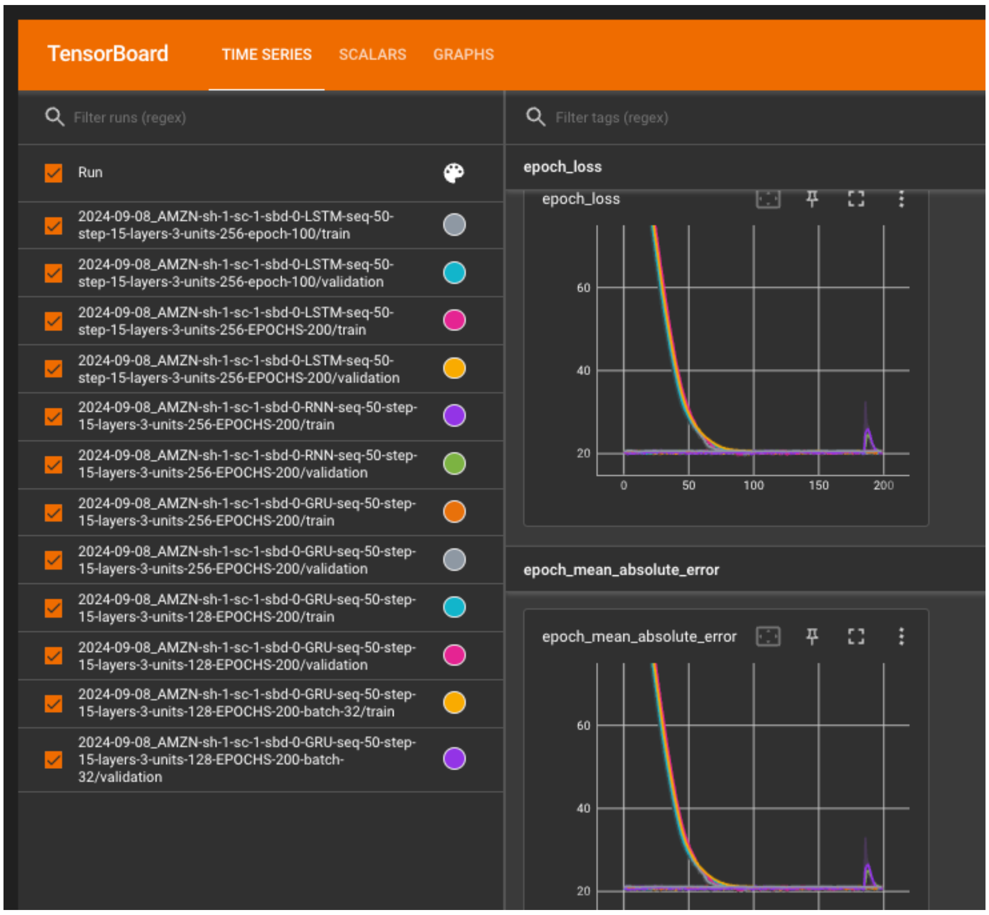
```
LAYER_NAME = RNN

# save model name, and make it unique as possible
model_name = f"{DATE_NOW}_{TICKER}-{shuffle_str}-{scale_str}-{SPLIT_BY_DATE_str}-{LAYER_NAME.__name__}-seq-{N_STEPS}-step-{LOOKUP_STEP}-layers-{N_LAYERS}-units-{N_SIZE}-EPOCHS-{EPOCHS}"
```

Fourth experiment with different layer name, use RNN layer name



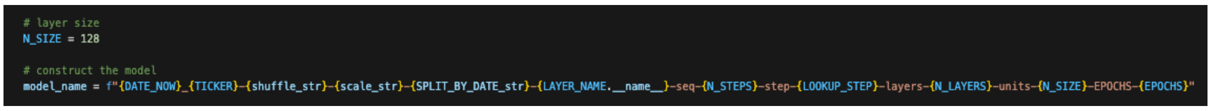
```
[58]  ✓  11m 43.2s
```

Training takes 11 minutes 43 seconds



Run
- 2024-09-08_AMZN-sh-1-sc-1-sbd-0-LSTM-seq-50-step-15-layers-3-units-256-epoch-100/train
- 2024-09-08_AMZN-sh-1-sc-1-sbd-0-LSTM-seq-50-step-15-layers-3-units-256-epoch-100/validation
- 2024-09-08_AMZN-sh-1-sc-1-sbd-0-LSTM-seq-50-step-15-layers-3-units-256-EPOCHS-200/train
- 2024-09-08_AMZN-sh-1-sc-1-sbd-0-LSTM-seq-50-step-15-layers-3-units-256-EPOCHS-200/validation
- 2024-09-08_AMZN-sh-1-sc-1-sbd-0-RNN-seq-50-step-15-layers-3-units-256-EPOCHS-200/train
- 2024-09-08_AMZN-sh-1-sc-1-sbd-0-RNN-seq-50-step-15-layers-3-units-256-EPOCHS-200/validation
- 2024-09-08_AMZN-sh-1-sc-1-sbd-0-GRU-seq-50-step-15-layers-3-units-256-EPOCHS-200/train
- 2024-09-08_AMZN-sh-1-sc-1-sbd-0-GRU-seq-50-step-15-layers-3-units-256-EPOCHS-200/validation
- 2024-09-08_AMZN-sh-1-sc-1-sbd-0-GRU-seq-50-step-15-layers-3-units-128-EPOCHS-200/train
- 2024-09-08_AMZN-sh-1-sc-1-sbd-0-GRU-seq-50-step-15-layers-3-units-128-EPOCHS-200/validation
- 2024-09-08_AMZN-sh-1-sc-1-sbd-0-GRU-seq-50-step-15-layers-3-units-128-EPOCHS-200-batch-32/train
- 2024-09-08_AMZN-sh-1-sc-1-sbd-0-GRU-seq-50-step-15-layers-3-units-128-EPOCHS-200-batch-32/validation

The epoch loss and mean_absolute_error shows decreasing substantially from 0 to more than 50, before stabilizing till the end.

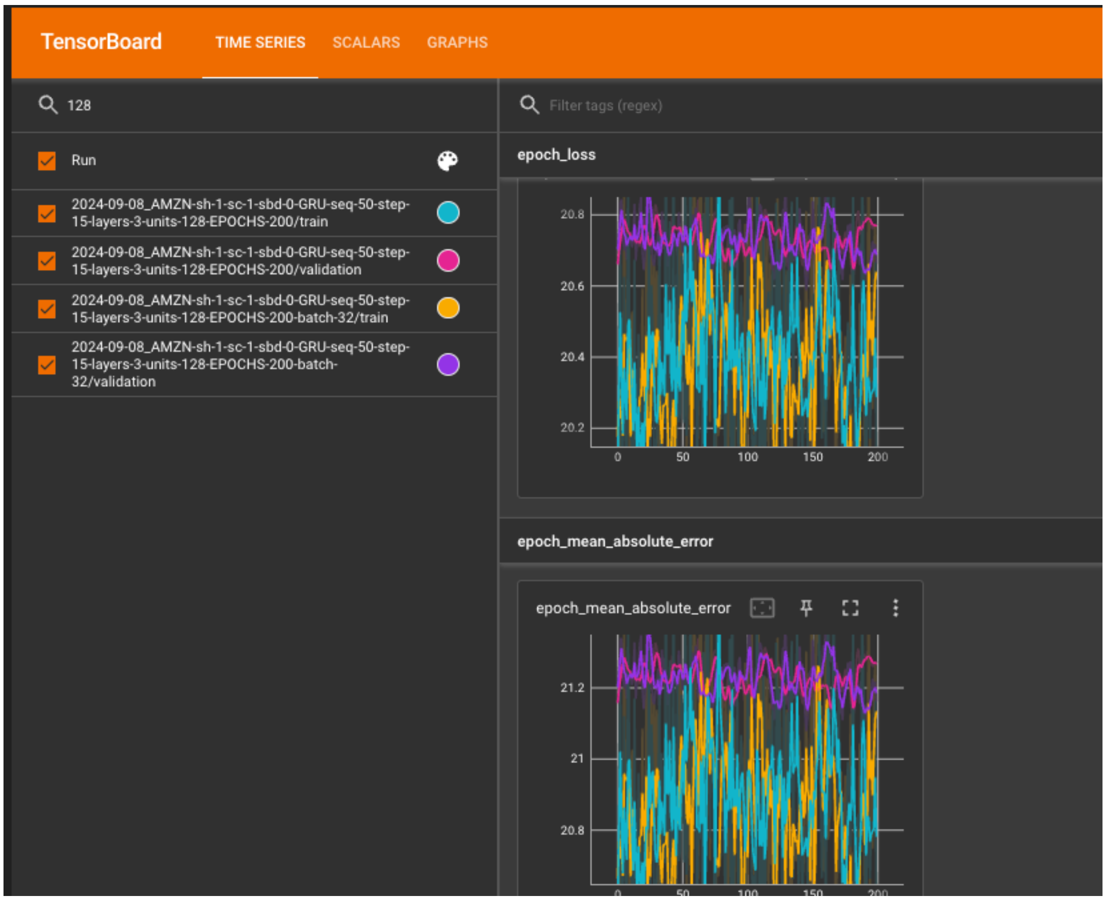


```
# layer size
N_SIZE = 128

# construct the model
model_name = f"{DATE_NOW}_{TICKER}-{shuffle_str}-{scale_str}-{SPLIT_BY_DATE_str}-{LAYER_NAME.__name__}-seq-{N_STEPS}-step-{LOOKUP_STEP}-layers-{N_LAYERS}-units-{N_SIZE}-EPOCHS-{EPOCHS}"
```

Fifth experiment with different layer size, with 128 layer size

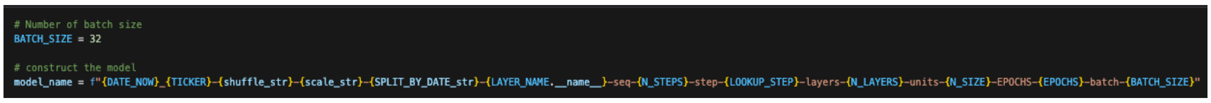
```
[62]  ✓  10m 44.0s
```

Training takes 10 minutes 44 seconds



2024-09-08_AMZN-sh-1-sc-1-sbd-0-GRU-seq-50-step-15-layers-3-units-128-EPOCHS-200/train ●
2024-09-08_AMZN-sh-1-sc-1-sbd-0-GRU-seq-50-step-15-layers-3-units-128-EPOCHS-200/validation ●
2024-09-08_AMZN-sh-1-sc-1-sbd-0-GRU-seq-50-step-15-layers-3-units-128-EPOCHS-200-batch-32/train ●
2024-09-08_AMZN-sh-1-sc-1-sbd-0-GRU-seq-50-step-15-layers-3-units-128-EPOCHS-200-batch-32/validation ●

Timeseries at 128 size number, with fluctuation as it seen, however the validation seems always in the higher range than the others.

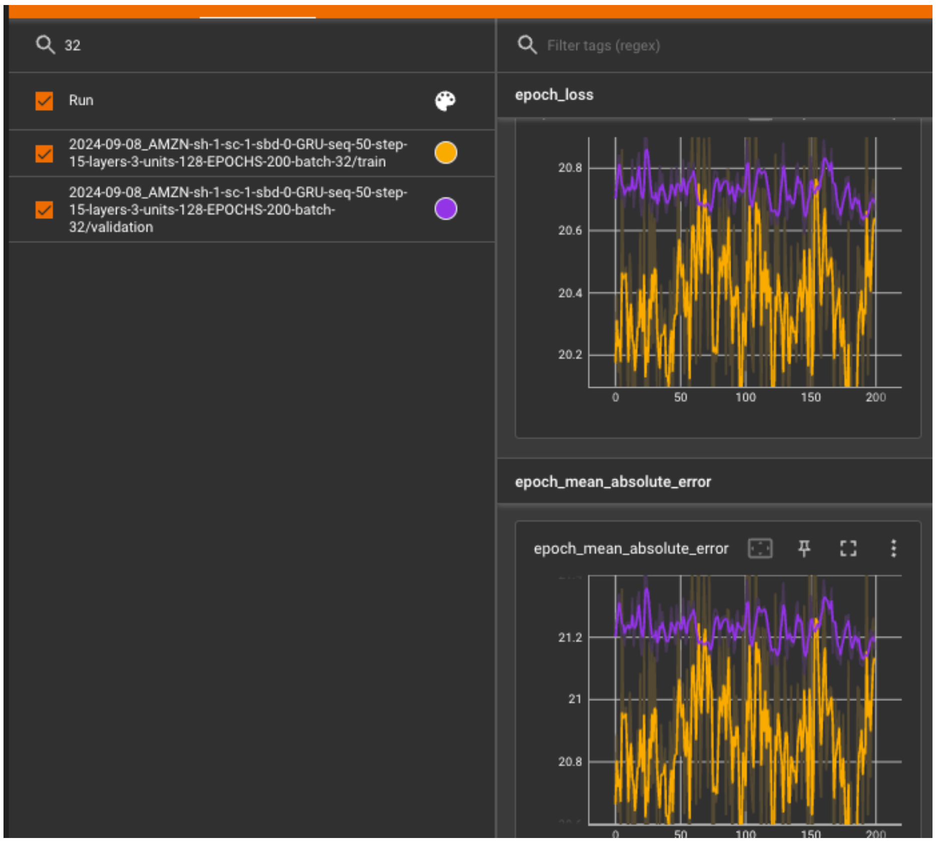


```
# Number of batch size
BATCH_SIZE = 32

# construct the model
model_name = f"{DATE_NOW}_{TICKER}-{shuffle_str}-{scale_str}-{SPLIT_BY_DATE_str}-{LAYER_NAME.__name__}-seq-{N_STEPS}-step-{LOOKUP_STEP}-layers-{N_LAYERS}-units-{N_SIZE}-EPOCHS-{EPOCHS}-batch-{BATCH_SIZE}"
```

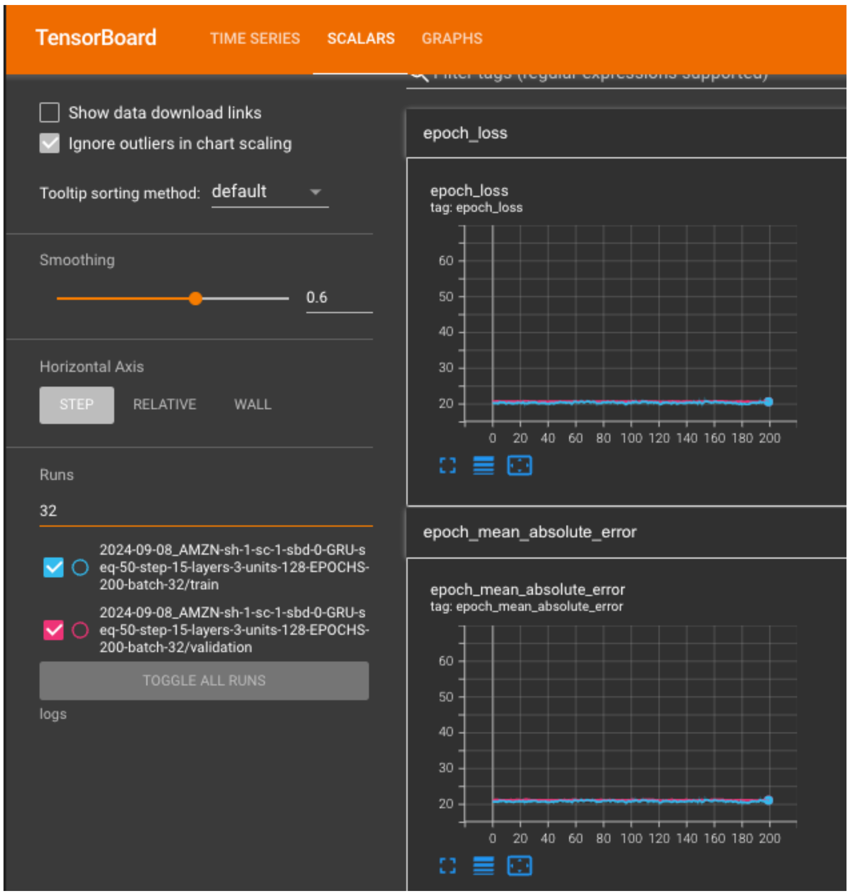
Sixth experiment with batch size, use 32 batch sizes

```
[64]  ✓  13m 34.7s
```

Training takes 13 minutes 34 seconds


32
Run
2024-09-08_AMZN-sh-1-sc-1-sbd-0-GRU-seq-50-step-15-layers-3-units-128-EPOCHS-200-batch-32/train
2024-09-08_AMZN-sh-1-sc-1-sbd-0-GRU-seq-50-step-15-layers-3-units-128-EPOCHS-200-batch-32/validation
Filter tags (regex)
epoch_loss
20.8
20.6
20.4
20.2
0
50
100
150
200
epoch_mean_absolute_error
epoch_mean_absolute_error
21.2
21
20.8
0
50
100
150
200

Time series at training and validation at batch size 32, with fluctuation at loss and mean absolute error.


TensorBoard
TIME SERIES
SCALARS
GRAPHS
Filter tags (regular expressions supported)
TOGGLE ALL RUNS
logs
epoch_loss
tag: epoch_loss
epoch_mean_absolute_error
tag: epoch_mean_absolute_error

☐ Show data download links
☑ Ignore outliers in chart scaling

Tooltip sorting method: default

Smoothing 0.6

Horizontal Axis: [✓ STEP] RELATIVE WALL

Runs: 32

☑ 2024-09-08_AMZN-sh-1-sc-1-sbd-0-GRU-seq-50-step-15-layers-3-units-128-EPOCHS-200-batch-32/train

☑ 2024-09-08_AMZN-sh-1-sc-1-sbd-0-GRU-seq-50-step-15-layers-3-units-128-EPOCHS-200-batch-32/validation

Scalars at training and validation at batch size 32, with stabilised data from 0 till 200.