lstm_weather_pred

October 15, 2024

[17]:
```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

[18]:
```python
df = pd.read_csv('./seattle-weather.xls')
```

[19]:
```python
df.head()
```

[19]:
```
         date  precipitation  temp_max  temp_min  wind  weather
0  2012-01-01            0.0      12.8       5.0   4.7  drizzle
1  2012-01-02           10.9      10.6       2.8   4.5     rain
2  2012-01-03            0.8      11.7       7.2   2.3     rain
3  2012-01-04           20.3      12.2       5.6   4.7     rain
4  2012-01-05            1.3       8.9       2.8   6.1     rain
```

[20]:
```python
df.isnull().sum()
```

[20]:
```
date             0
precipitation    0
temp_max         0
temp_min         0
wind             0
weather          0
dtype: int64
```

[21]:
```python
#  Pandas DataFrame
df.duplicated().sum()
```

[21]:
```
0
```

[22]:
```python
#column Open converted into numpy array
training_set = df.iloc[:,2:3].values
training_set
```

[22]:
```
array([[12.8],
       [10.6],
       [11.7],
        …,
        [ 7.2],
        [ 5.6],
        [ 5.6]])
```

[23]:
```
len(training_set)
```

[23]:
```
1461
```

[24]:
```
#
def df_to_XY(df,window_size=10):
 X_train=[]
 y_train=[]

 for i in range(10,len(training_set)):
    X_train.append(training_set[i-10:i,0])
    y_train.append(training_set[i,0])

 X_train, y_train = np.array(X_train), np.array(y_train)
 return X_train, y_train
```

[25]:
```
WINDOW = 10
X,y = df_to_XY(df,WINDOW)
print(len(X),len(y))
X_train = X[:800]
y_train = y[:800]
X_val = X[800:1000]
y_val = y[800:1000]
X_test = X[1000:]
x_test = y[1000:]
```

```
1451 1451
```

[26]:
```
#          RNN/LSTM/GRU
X_train = np.reshape(X_train,(X_train.shape[0],X_train.shape[1],1))
X_val = np.reshape(X_val,(X_val.shape[0],X_val.shape[1],1))
X_test = np.reshape(X_test,(X_test.shape[0],X_test.shape[1],1))
```

[27]:
```
#Building the RNN
from keras.models import Sequential
from keras.layers import Dense, LSTM, Dropout
```

[28]:
```
model = Sequential()
```

[29]:
```
#Addinf the first LSTM layer and some Dropout regularisation
model.add(LSTM(units=50, return_sequences = True, input_shape=(X_train.
 ↪shape[1], 1)))
```

```
model.add(Dropout(0.2))

model.add(LSTM(units=50, return_sequences = True))
model.add(Dropout(0.2))

model.add(LSTM(units=50, return_sequences = True))
model.add(Dropout(0.2))

model.add(LSTM(units=50))
model.add(Dropout(0.2))

#Output layer
model.add(Dense(units=1))
```

[30]:

```
model.compile(optimizer='adam',loss='mean_squared_error')
```

[31]:

```
from tensorflow.keras.callbacks import ModelCheckpoint,EarlyStopping
from tensorflow.keras.losses import MeanSquaredError
from tensorflow.keras.metrics import RootMeanSquaredError
from tensorflow.keras.optimizers import Adam
```

[32]:

```
#fitting the rnn to the training set
history=model.fit(X_train,y_train,validation_data=(X_val,y_val),epochs=100,␣
 ↪batch_size=32)
```

```
Epoch 1/100

2024-10-15 10:11:12.337579: I tensorflow/stream_executor/cuda/cuda_dnn.cc:366]
Loaded cuDNN version 8400
2024-10-15 10:11:12.455153: I tensorflow/stream_executor/cuda/cuda_blas.cc:1774]
TensorFloat-32 will be used for the matrix multiplication. This will only be
logged once.

25/25 [==============================] - 3s 33ms/step - loss: 203.0035 -
val_loss: 264.4172
Epoch 2/100
25/25 [==============================] - 0s 8ms/step - loss: 102.2899 -
val_loss: 180.7950
Epoch 3/100
25/25 [==============================] - 0s 13ms/step - loss: 77.6039 -
val_loss: 146.9615
Epoch 4/100
25/25 [==============================] - 0s 12ms/step - loss: 67.3265 -
val_loss: 125.6385
Epoch 5/100
25/25 [==============================] - 0s 9ms/step - loss: 60.8989 - val_loss:
111.0012
Epoch 6/100
```

```
25/25 [==============================] - 0s 12ms/step - loss: 57.9472 -
val_loss: 99.2511
Epoch 7/100
25/25 [==============================] - 0s 12ms/step - loss: 55.4723 -
val_loss: 91.8026
Epoch 8/100
25/25 [==============================] - 0s 9ms/step - loss: 55.1650 - val_loss:
86.5459
Epoch 9/100
25/25 [==============================] - 0s 10ms/step - loss: 54.1517 -
val_loss: 82.8307
Epoch 10/100
25/25 [==============================] - 0s 9ms/step - loss: 53.8719 - val_loss:
80.5319
Epoch 11/100
25/25 [==============================] - 0s 14ms/step - loss: 54.3165 -
val_loss: 78.2327
Epoch 12/100
25/25 [==============================] - 0s 14ms/step - loss: 53.2201 -
val_loss: 77.8741
Epoch 13/100
25/25 [==============================] - 0s 15ms/step - loss: 53.8704 -
val_loss: 77.0980
Epoch 14/100
25/25 [==============================] - 0s 14ms/step - loss: 54.7782 -
val_loss: 77.0787
Epoch 15/100
25/25 [==============================] - 0s 12ms/step - loss: 54.0774 -
val_loss: 76.4165
Epoch 16/100
25/25 [==============================] - 0s 12ms/step - loss: 54.4011 -
val_loss: 76.3147
Epoch 17/100
25/25 [==============================] - 0s 14ms/step - loss: 53.7023 -
val_loss: 76.1859
Epoch 18/100
25/25 [==============================] - 0s 13ms/step - loss: 54.1639 -
val_loss: 75.8794
Epoch 19/100
25/25 [==============================] - 0s 12ms/step - loss: 54.3188 -
val_loss: 76.9150
Epoch 20/100
25/25 [==============================] - 0s 14ms/step - loss: 52.8877 -
val_loss: 76.3676
Epoch 21/100
25/25 [==============================] - 0s 13ms/step - loss: 53.1914 -
val_loss: 75.9187
Epoch 22/100
```

```
25/25 [==============================] - 0s 15ms/step - loss: 54.3439 -
val_loss: 75.4120
Epoch 23/100
25/25 [==============================] - 0s 14ms/step - loss: 53.6057 -
val_loss: 75.4410
Epoch 24/100
25/25 [==============================] - 0s 14ms/step - loss: 54.3661 -
val_loss: 75.3254
Epoch 25/100
25/25 [==============================] - 0s 15ms/step - loss: 54.5082 -
val_loss: 75.1119
Epoch 26/100
25/25 [==============================] - 0s 14ms/step - loss: 53.6633 -
val_loss: 75.8011
Epoch 27/100
25/25 [==============================] - 0s 14ms/step - loss: 53.9183 -
val_loss: 76.2378
Epoch 28/100
25/25 [==============================] - 0s 12ms/step - loss: 54.2893 -
val_loss: 76.1239
Epoch 29/100
25/25 [==============================] - 0s 14ms/step - loss: 53.7098 -
val_loss: 76.8454
Epoch 30/100
25/25 [==============================] - 0s 13ms/step - loss: 52.3843 -
val_loss: 76.8698
Epoch 31/100
25/25 [==============================] - 0s 13ms/step - loss: 52.7934 -
val_loss: 76.2771
Epoch 32/100
25/25 [==============================] - 0s 14ms/step - loss: 42.3399 -
val_loss: 69.2942
Epoch 33/100
25/25 [==============================] - 0s 13ms/step - loss: 32.3428 -
val_loss: 53.9093
Epoch 34/100
25/25 [==============================] - 0s 14ms/step - loss: 27.8614 -
val_loss: 43.5234
Epoch 35/100
25/25 [==============================] - 0s 15ms/step - loss: 23.2740 -
val_loss: 37.5776
Epoch 36/100
25/25 [==============================] - 0s 14ms/step - loss: 19.7417 -
val_loss: 31.6569
Epoch 37/100
25/25 [==============================] - 0s 12ms/step - loss: 17.5100 -
val_loss: 27.3653
Epoch 38/100
```

```
25/25 [==============================] - 0s 14ms/step - loss: 16.0691 -
val_loss: 24.2274
Epoch 39/100
25/25 [==============================] - 0s 13ms/step - loss: 14.4216 -
val_loss: 22.2853
Epoch 40/100
25/25 [==============================] - 0s 7ms/step - loss: 13.5039 - val_loss:
21.3827
Epoch 41/100
25/25 [==============================] - 0s 16ms/step - loss: 12.3916 -
val_loss: 19.0581
Epoch 42/100
25/25 [==============================] - 0s 15ms/step - loss: 12.0891 -
val_loss: 17.0781
Epoch 43/100
25/25 [==============================] - 0s 16ms/step - loss: 11.8169 -
val_loss: 17.3055
Epoch 44/100
25/25 [==============================] - 0s 12ms/step - loss: 11.0414 -
val_loss: 15.7899
Epoch 45/100
25/25 [==============================] - 0s 15ms/step - loss: 11.6832 -
val_loss: 15.4896
Epoch 46/100
25/25 [==============================] - 0s 14ms/step - loss: 11.6313 -
val_loss: 15.3937
Epoch 47/100
25/25 [==============================] - 0s 13ms/step - loss: 11.0534 -
val_loss: 15.3141
Epoch 48/100
25/25 [==============================] - 0s 14ms/step - loss: 11.0957 -
val_loss: 14.6554
Epoch 49/100
25/25 [==============================] - 0s 12ms/step - loss: 10.0294 -
val_loss: 13.1531
Epoch 50/100
25/25 [==============================] - 0s 10ms/step - loss: 10.8698 -
val_loss: 13.8426
Epoch 51/100
25/25 [==============================] - 0s 11ms/step - loss: 10.4688 -
val_loss: 12.9201
Epoch 52/100
25/25 [==============================] - 0s 13ms/step - loss: 10.3726 -
val_loss: 13.7410
Epoch 53/100
25/25 [==============================] - 0s 9ms/step - loss: 10.7054 - val_loss:
12.9269
Epoch 54/100
```

```
25/25 [==============================] - 0s 15ms/step - loss: 10.4023 -
val_loss: 12.6068
Epoch 55/100
25/25 [==============================] - 0s 13ms/step - loss: 10.2635 -
val_loss: 11.8137
Epoch 56/100
25/25 [==============================] - 0s 14ms/step - loss: 9.8302 - val_loss:
11.6798
Epoch 57/100
25/25 [==============================] - 0s 13ms/step - loss: 9.5413 - val_loss:
11.4466
Epoch 58/100
25/25 [==============================] - 0s 15ms/step - loss: 10.1236 -
val_loss: 11.4588
Epoch 59/100
25/25 [==============================] - 0s 14ms/step - loss: 9.2308 - val_loss:
11.8392
Epoch 60/100
25/25 [==============================] - 0s 14ms/step - loss: 9.5176 - val_loss:
11.5163
Epoch 61/100
25/25 [==============================] - 0s 13ms/step - loss: 9.5010 - val_loss:
11.7398
Epoch 62/100
25/25 [==============================] - 0s 13ms/step - loss: 9.5174 - val_loss:
11.4442
Epoch 63/100
25/25 [==============================] - 0s 14ms/step - loss: 9.3415 - val_loss:
11.1767
Epoch 64/100
25/25 [==============================] - 0s 14ms/step - loss: 9.5053 - val_loss:
11.3466
Epoch 65/100
25/25 [==============================] - 0s 15ms/step - loss: 9.0026 - val_loss:
11.1001
Epoch 66/100
25/25 [==============================] - 0s 14ms/step - loss: 9.1866 - val_loss:
11.3769
Epoch 67/100
25/25 [==============================] - 0s 14ms/step - loss: 9.5347 - val_loss:
11.4859
Epoch 68/100
25/25 [==============================] - 0s 12ms/step - loss: 9.4171 - val_loss:
10.5593
Epoch 69/100
25/25 [==============================] - 0s 15ms/step - loss: 9.2742 - val_loss:
10.5046
Epoch 70/100
```

```
25/25 [==============================] - 0s 15ms/step - loss: 9.0556 - val_loss:
10.7261
Epoch 71/100
25/25 [==============================] - 0s 15ms/step - loss: 9.1739 - val_loss:
10.3059
Epoch 72/100
25/25 [==============================] - 0s 14ms/step - loss: 9.1896 - val_loss:
11.0182
Epoch 73/100
25/25 [==============================] - 0s 15ms/step - loss: 9.2522 - val_loss:
10.5511
Epoch 74/100
25/25 [==============================] - 0s 5ms/step - loss: 9.0566 - val_loss:
10.1163
Epoch 75/100
25/25 [==============================] - 0s 15ms/step - loss: 9.3013 - val_loss:
10.1046
Epoch 76/100
25/25 [==============================] - 0s 12ms/step - loss: 9.3026 - val_loss:
10.5102
Epoch 77/100
25/25 [==============================] - 0s 12ms/step - loss: 9.4632 - val_loss:
10.9640
Epoch 78/100
25/25 [==============================] - 0s 13ms/step - loss: 9.4576 - val_loss:
10.0826
Epoch 79/100
25/25 [==============================] - 0s 14ms/step - loss: 9.1436 - val_loss:
10.3107
Epoch 80/100
25/25 [==============================] - 0s 13ms/step - loss: 9.1432 - val_loss:
10.2158
Epoch 81/100
25/25 [==============================] - 0s 16ms/step - loss: 8.7108 - val_loss:
10.0776
Epoch 82/100
25/25 [==============================] - 0s 15ms/step - loss: 8.9454 - val_loss:
9.9400
Epoch 83/100
25/25 [==============================] - 0s 14ms/step - loss: 8.9352 - val_loss:
10.4775
Epoch 84/100
25/25 [==============================] - 0s 16ms/step - loss: 8.8529 - val_loss:
9.7789
Epoch 85/100
25/25 [==============================] - 0s 15ms/step - loss: 8.2442 - val_loss:
9.9967
Epoch 86/100
```

```
25/25 [==============================] - 0s 14ms/step - loss: 9.1580 - val_loss:
10.5040
Epoch 87/100
25/25 [==============================] - 0s 15ms/step - loss: 8.8401 - val_loss:
10.4785
Epoch 88/100
25/25 [==============================] - 0s 16ms/step - loss: 8.2660 - val_loss:
9.9030
Epoch 89/100
25/25 [==============================] - 0s 13ms/step - loss: 8.7762 - val_loss:
10.0194
Epoch 90/100
25/25 [==============================] - 0s 14ms/step - loss: 8.7576 - val_loss:
9.5539
Epoch 91/100
25/25 [==============================] - 0s 15ms/step - loss: 8.4457 - val_loss:
10.4540
Epoch 92/100
25/25 [==============================] - 0s 14ms/step - loss: 8.5489 - val_loss:
9.6597
Epoch 93/100
25/25 [==============================] - 0s 13ms/step - loss: 8.7434 - val_loss:
9.7322
Epoch 94/100
25/25 [==============================] - 0s 15ms/step - loss: 8.9801 - val_loss:
9.7966
Epoch 95/100
25/25 [==============================] - 0s 16ms/step - loss: 8.8115 - val_loss:
10.0329
Epoch 96/100
25/25 [==============================] - 0s 14ms/step - loss: 8.8199 - val_loss:
9.8995
Epoch 97/100
25/25 [==============================] - 0s 13ms/step - loss: 8.1177 - val_loss:
9.8169
Epoch 98/100
25/25 [==============================] - 0s 13ms/step - loss: 8.8496 - val_loss:
9.5331
Epoch 99/100
25/25 [==============================] - 0s 13ms/step - loss: 8.1043 - val_loss:
9.6355
Epoch 100/100
25/25 [==============================] - 0s 15ms/step - loss: 8.7926 - val_loss:
10.2852
```

[33]:

```
his = pd.DataFrame(history.history)
```

[34]:

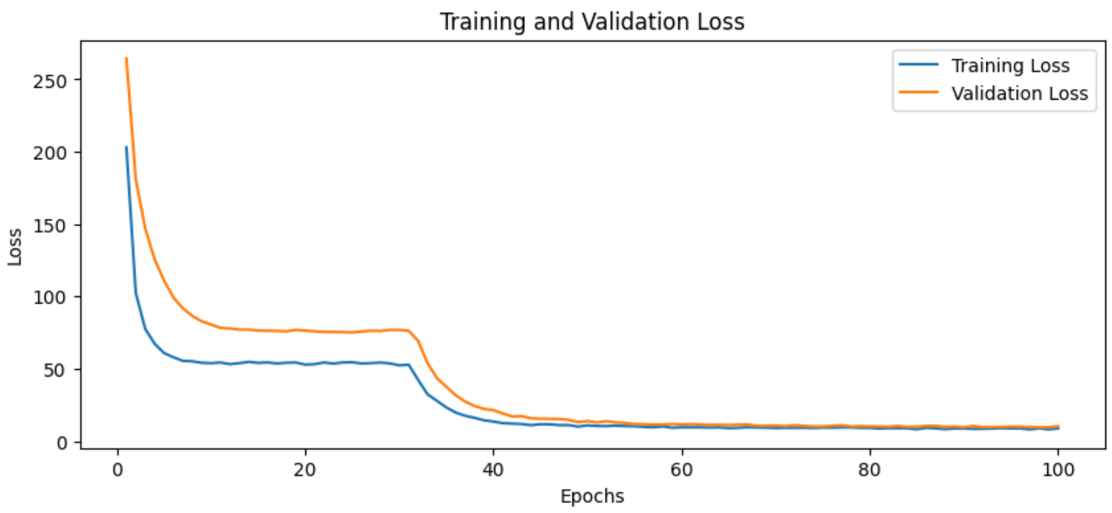
```
his.head()
```

[34]:

```
         loss    val_loss
0  203.003510  264.417236
1  102.289886  180.794983
2   77.603874  146.961472
3   67.326523  125.638519
4   60.898930  111.001152
```

[35]:

```
import seaborn as sns
import matplotlib.pyplot as plt

#
history_loss = history.history['loss']
history_val_loss = history.history['val_loss']

#    x     epoch
epochs = range(1, len(history_loss) + 1)

#
plt.figure(figsize=(10, 4))

#
plt.plot(epochs, history_loss, label='Training Loss')

#
plt.plot(epochs, history_val_loss, label='Validation Loss')

#
plt.title('Training and Validation Loss')
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.legend()

#
plt.show()
```



[36]:

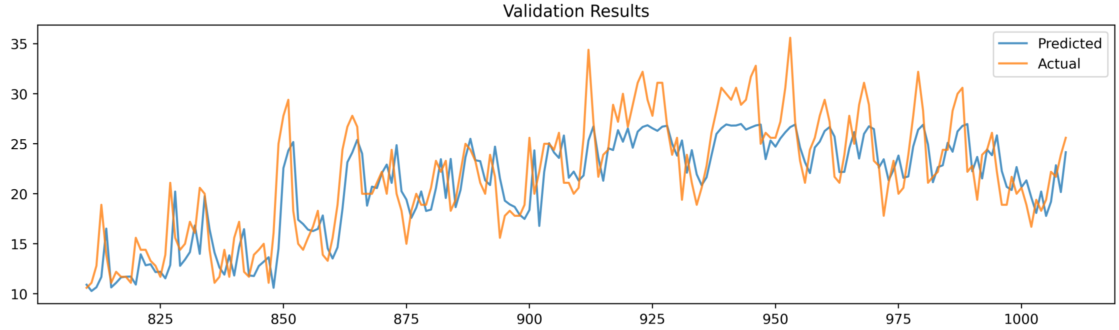
```
train_pred = model.predict(X_train).flatten()
val_pred = model.predict(X_val).flatten()
test_pred = model.predict(X_test).flatten()
```

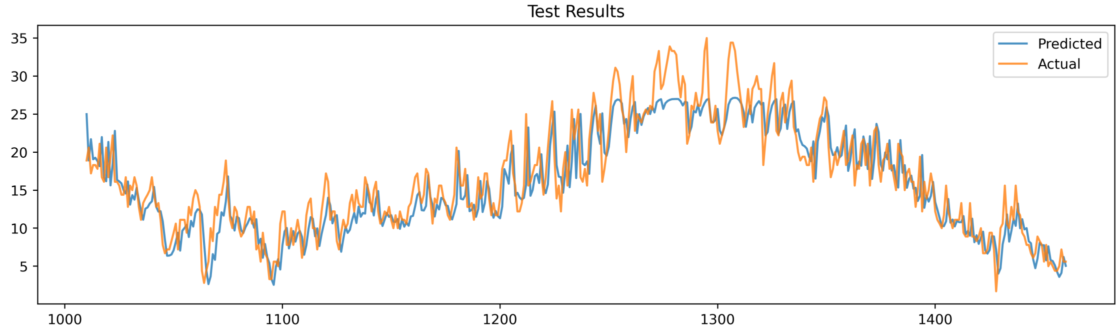
[37]:

```
pred = np.concatenate([train_pred,val_pred,test_pred])
df_pred = pd.DataFrame(df["temp_max"].copy())
df_pred.columns=["actual"]
df_pred = df_pred[WINDOW:]
df_pred["predicted"] = pred

fig,axes = plt.subplots(2,1,figsize=(14,8),dpi=400)

plt.subplot(2,1,1)
plt.title("Validation Results")
plt.plot(df_pred['predicted'][800:1000], label='Predicted',alpha=0.
  ↪8,linestyle=None)
plt.plot(df_pred['actual'][800:1000], label='Actual',alpha=0.8,linestyle=None)
plt.legend()
plt.subplot(2,1,2)
plt.title("Test Results")
plt.plot(df_pred['predicted'][1000:], label='Predicted',alpha=0.
  ↪8,linestyle=None)
plt.plot(df_pred['actual'][1000:], label='Actual',alpha=0.8,linestyle=None)
plt.legend()
```

[37]:

```
<matplotlib.legend.Legend at 0x7f74580d48e0>
```


Validation Results


Test Results