


In assignment 2, there is also a task about LSTM forecasting.

I suggest you,

1. Go to TensorFlow's official website.
https://www.tensorflow.org/tutorials/structured_data/time_series
2. You can either download the code or view the code in Google Colab

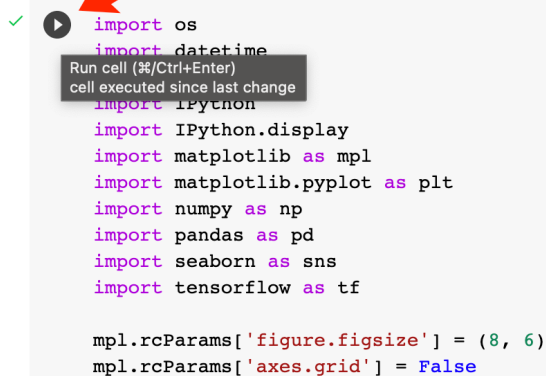
 Run in Google Colab

 View source on GitHub

 Download notebook

3. For example, if I view the code in Google Colab,
I will be able to click the 'run' icon to run the code block.

▼ Setup



```
import os
import datetime
import IPython
import IPython.display
import matplotlib as mpl
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import seaborn as sns
import tensorflow as tf

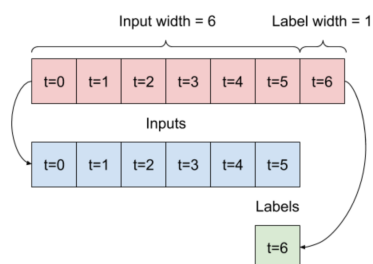
mpl.rcParams['figure.figsize'] = (8, 6)
mpl.rcParams['axes.grid'] = False
```

4. In this example, the dataset is the weather dataset, which includes very easy-to-understand features.
However, if your assignment data includes many terms like PM_{10} , please tell the reader what these are.

5. Read the 'data windowing' section carefully.

Given a list of consecutive inputs, the `split_window` method will convert them to a window of inputs and a window of labels.

The example `w2` you define earlier will be split like this:



6. In the section of defining a reusable [compile and fit](#) function.

This tutorial trains many models, so package the training procedure into a function:

```
[ ] MAX_EPOCHS = 20

def compile_and_fit(model, window, patience=2):
    early_stopping = tf.keras.callbacks.EarlyStopping(monitor='val_loss',
                                                        patience=patience,
                                                        mode='min')

    model.compile(loss=tf.losses.MeanSquaredError(),
                  optimizer=tf.optimizers.Adam(),
                  metrics=[tf.metrics.MeanAbsoluteError()])

    history = model.fit(window.train, epochs=MAX_EPOCHS,
                        validation_data=window.val,
                        callbacks=[early_stopping])

    return history
```

Please try to tune the hyperparameters.

Such as the **max_epochs**, **batch_size**, and **learning_rate**. (You should also know what the Adam optimizer is).

```
model.compile(loss='mean_squared_error',
              optimizer=Adam(learning_rate=0.01)
              )
model.fit(x_train,
          y_train,
          epochs=100,
          batch_size=4
          )
```

7. Please read the tutorial carefully, and learn the code about generating charts to improve the quality of your report.

The plot in your report should be nice and clear, such as Fig. 1 compares the label and your predicted value.

```
wide_window.plot(lstm_model)
```

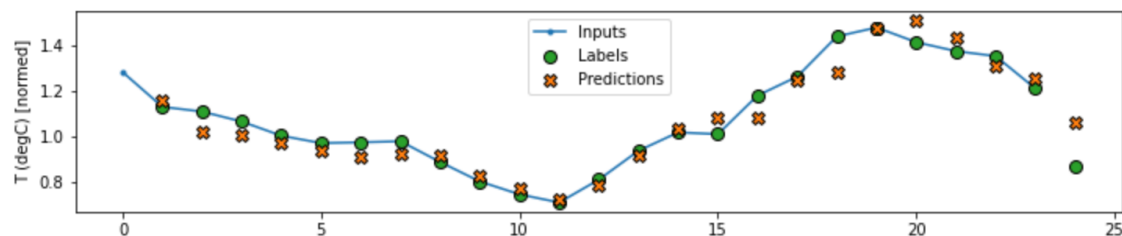


Figure 1. result.

You should also interpret it.

...

Such as Fig. 2 shows the result comparison between LSTM and other algorithms.

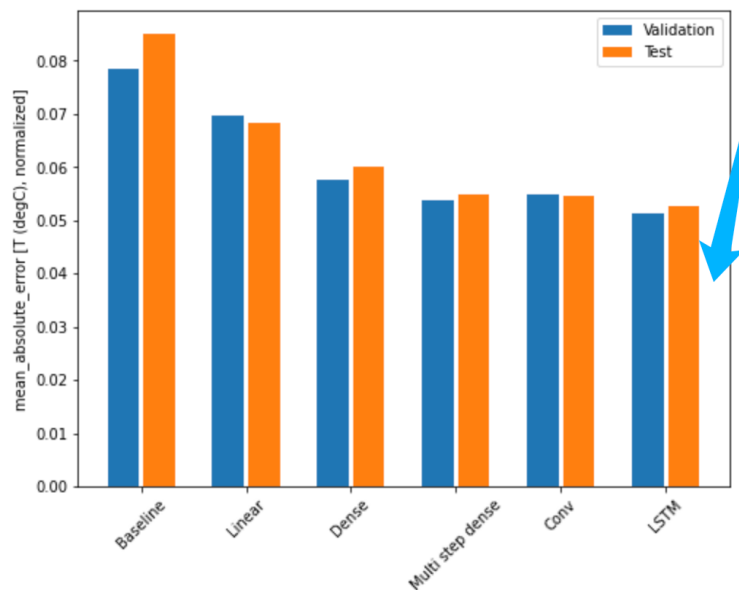


Figure 2. Mean-absolute-Error comparison.

You should also interpret it.

etc.

8. This TensorFlow official tutorial compares different kinds of algorithms. **You should understand the [LSTM](#) part completely; otherwise, you may not be able to complete your assignment 2.**