

✓ Mounting Google Drive in your VM

```
from google.colab import drive
drive.mount('/content/drive')
```

Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).

```
with open('/content/drive/MyDrive/Colab Notebooks/foo.txt', 'w') as f:
    f.write('Hello Google Drive!')
!cat '/content/drive/MyDrive/Colab Notebooks/foo.txt'
```

Hello Google Drive!

✓ Create the simplest neural network

HW1:請使用別種的optimizer和Loss function, 並更改輸入和輸出的值, 並將Cell輸出畫面截圖上傳

```
import tensorflow as tf
import numpy as np
from tensorflow import keras

model = tf.keras.Sequential([keras.layers.Dense(units=1, input_shape=[1])])
model.compile(optimizer='adamax', loss='mean_absolute_error')# adms Lion...
xs = np.array([-1.0, 0.0, 1.0, 2.0, 3.0, 4.0], dtype=float)
ys = np.array([-2.0, 1.0, 4.0, 7.0, 10.0, 20.0], dtype=float)
# 以上code定義此neural network

model.fit(xs, ys, epochs=1000)
```

```
1/1 ----- 0s 58ms/step - loss: 1.7907
Epoch 998/1000
1/1 ----- 0s 45ms/step - loss: 1.7882
Epoch 999/1000
1/1 ----- 0s 67ms/step - loss: 1.7857
Epoch 1000/1000
1/1 ----- 0s 135ms/step - loss: 1.7832
<keras.callbacks.history.History at 0x79e94daa9500>
```

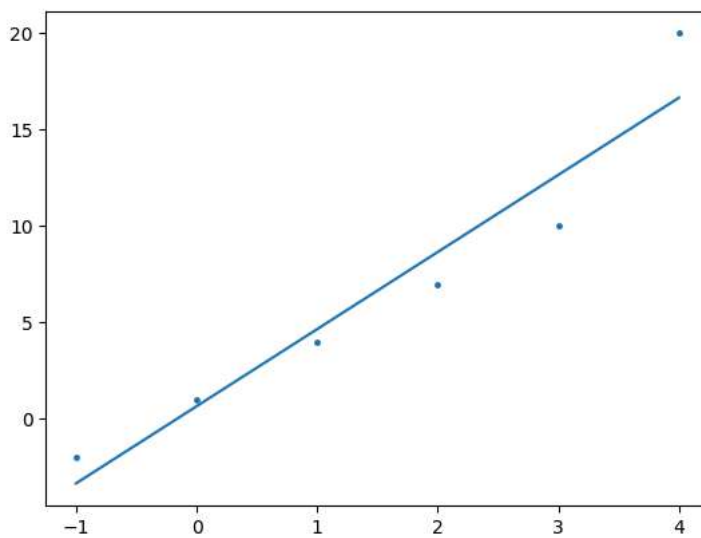
```
model.layers[0].get_weights()
```

```
[array([[2.665071]], dtype=float32), array([0.9999907], dtype=float32)]
```

```
import matplotlib.pyplot as plt
from scipy import stats
slope, intercept, r, p, std_err = stats.linregress(xs, ys)

def myfunc(v):
    return slope * v + intercept

mymodel = list(map(myfunc, xs))
plt.scatter(xs, ys, 6)
plt.plot(xs, mymodel)
plt.show()
```



```
print(model.predict(x=np.array([10.0])))
```

```
WARNING:tensorflow:5 out of the last 5 calls to <function TensorFlowTrainer.make_predict_function.<locals>.one_step_on_data_distributed at 1/1 ----- 0s 137ms/step
[[27.650702]]
```

✓ Charting in Colaboratory

A common use for notebooks is data visualization using charts. Colaboratory makes this easy with several charting tools available as Python imports.

✓ Matplotlib

[Matplotlib](#) is the most common charting package, see its [documentation](#) for details, and its [examples](#) for inspiration.

✓ Line Plots

```
import matplotlib.pyplot as plt

x = [1, 2, 3, 4, 5, 6, 7, 8, 9]
y1 = [1, 3, 5, 3, 1, 3, 5, 3, 1]
y2 = [2, 4, 6, 4, 2, 4, 6, 4, 2]
plt.plot(x, y1, label="line L")
plt.plot(x, y2, label="line H")
plt.plot()

plt.xlabel("x axis")
```