神經網路又稱為機器學習

決定目標(自變量,應變量) → 資料清洗 →
模型建立 → 模型診斷

類神經網路優點：無限制, 效率高, 效率高
缺點：黑盒子

## ① 套件匯入

```
import pandas as pd
import numpy as np
from sklearn import datasets, preprocessing
from sklearn.model_selection import train_test_split
from keras.models import Sequential
from keras.layers import Dense,Dropout
from keras.optimizers import SGD, Adam
import matplotlib.pyplot as plt
```

## ② 叫出iris資料

```
iris = datasets.load_iris()
df = pd.DataFrame(iris['data'], columns=iris['feature_names'])
```

## ③ 標準化 → 將單位去除

```
min_max_scaler = preprocessing.MinMaxScaler()
df = pd.DataFrame(min_max_scaler.fit_transform(df),columns=iris['feature_names'])
```

## ④ 挑出應變量和自變量

```
labels = np.array(df['sepal length (cm)']) #應變數
features= df.drop('sepal length (cm)', axis = 1) #自變數
```

## ⑤ 隨機抽樣

```
trainX, testX, trainY, testY = train_test_split(features, labels, test_size = 0.3,
random_state = 42)
trainY = trainY.reshape(-1, 1) #創行
testY = testY.reshape(-1, 1) #創行
```

## ⑥ 建模

過度配適

激活函數

可解決梯度消失的問題

```
model = Sequential()
model.add(Dense(64, input_dim=3, activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(1)) #類別型, activation='softmax'
model.compile(loss='mse', #類別型'categorical_crossentropy'
          optimizer=SGD(lr=0.1),#adam
          metrics=['mse','mape']) #類別型'accuracy'
#%%
```

只保留0.5

隱藏層中神經元的數量

自變量

學習率=0.1

很容易造成局部最小值

## ⑦ 模型訓練

```
dnn = model.fit(trainX, trainY,epochs=20,batch_size=30)
```

訓練自變量

訓練應變量

每次抓30資料出來擬合一次

要做20次 ⇐ 一次

## ⑧預測

```
predictions = model.predict(testX)
predictions = predictions.reshape(-1, 1)
np.mean(np.abs((testY - predictions))) #mae
np.mean(np.abs((testY - predictions) / testY)) * 100 #mape
```

會跟前一個預測結果近似