機器學習於材料資訊的應用
Machine Learning on Material Informatics

陳南佑(NAN-YOW CHEN)

nanyow@narlabs.org.tw

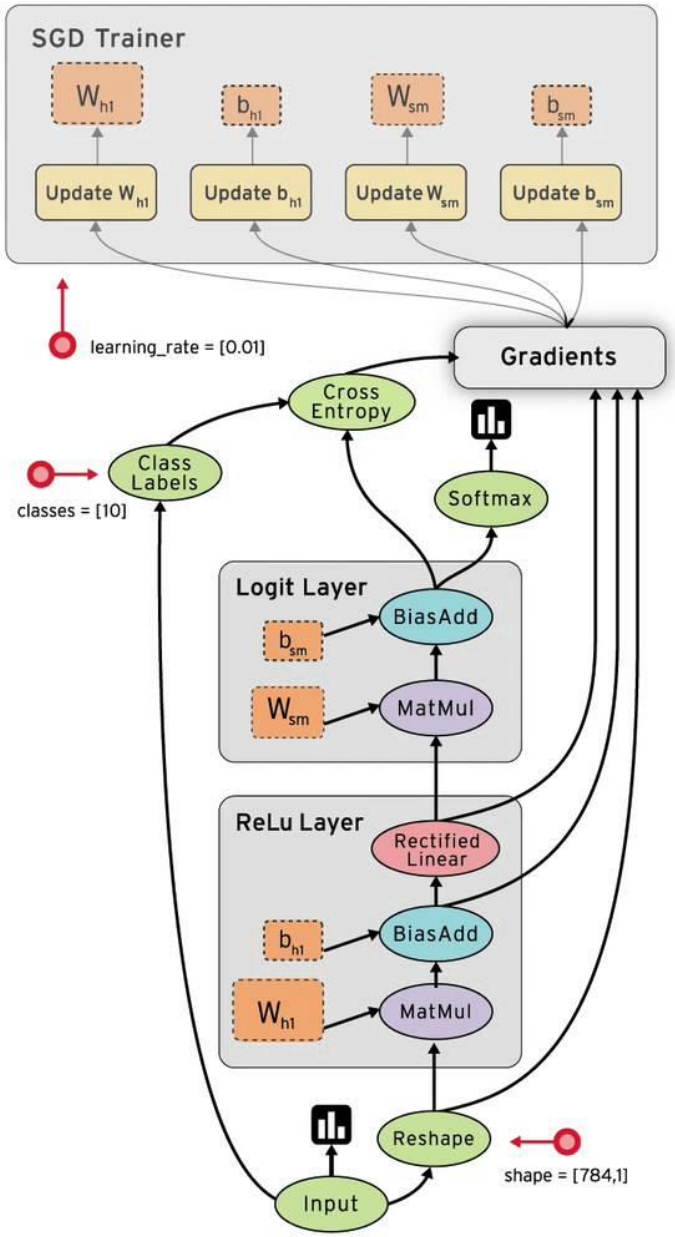
楊安正(AN-CHENG YANG)

acyang@narlabs.org.tw

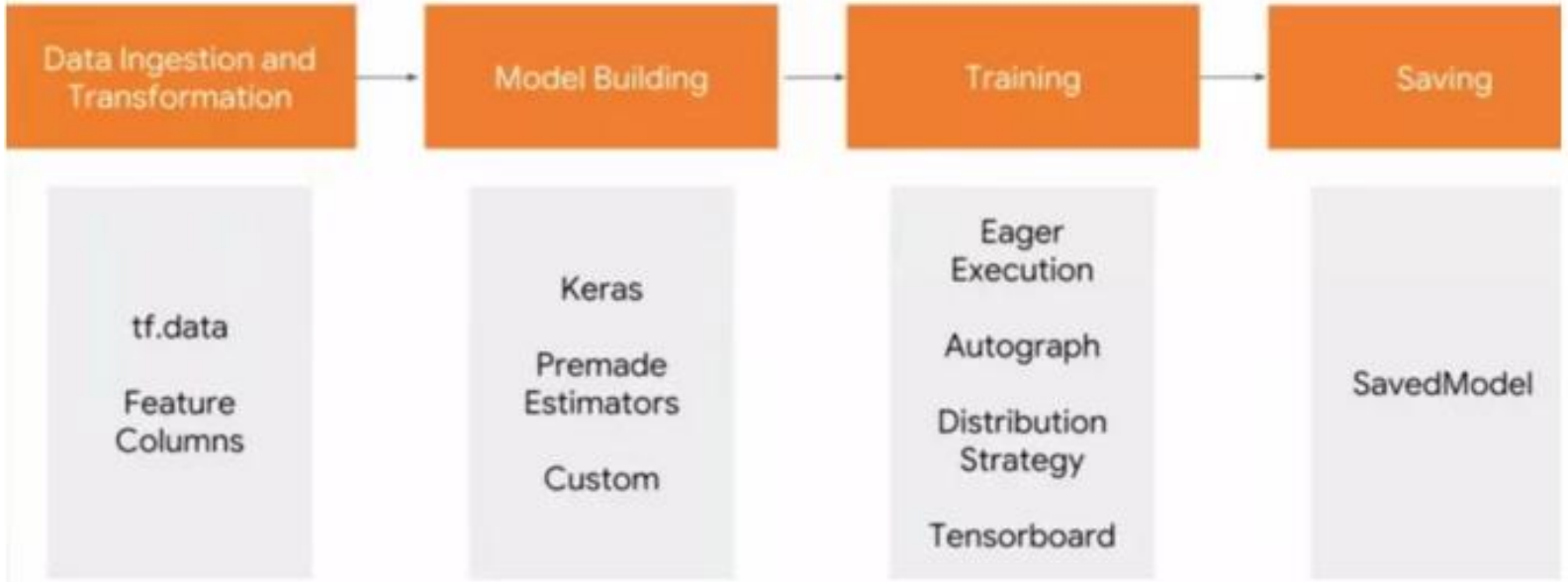
Framework For Machine Learning

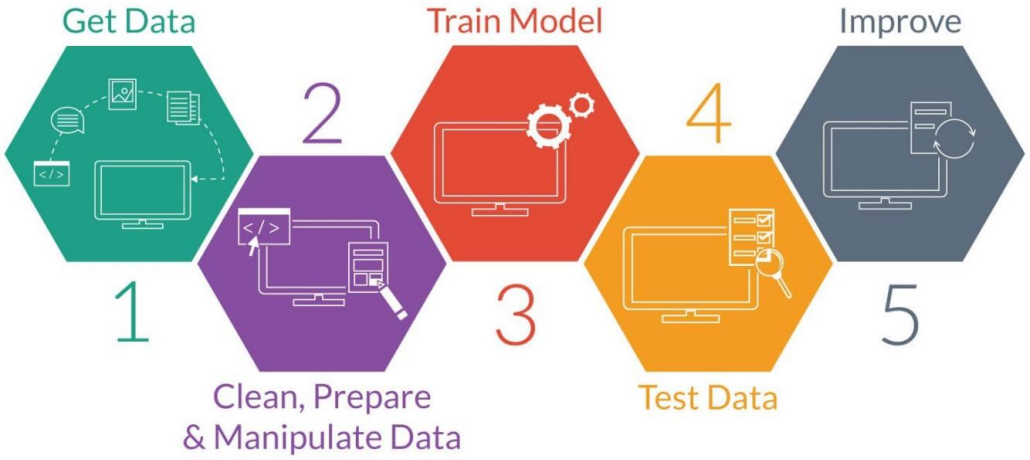
What is Flow?

- Tensorflow主要的運作流程分為以下兩個部分
  - 建立模型(Build Model)
  - 執行運算(Run)
- Tensorflow設計的核心就是Tensor的流動，建立Graph的過程其實只是定義好Tensor如何流動並運算的過程，但真正的資料其實並沒有被運算，真正的計算需要用session來執行。



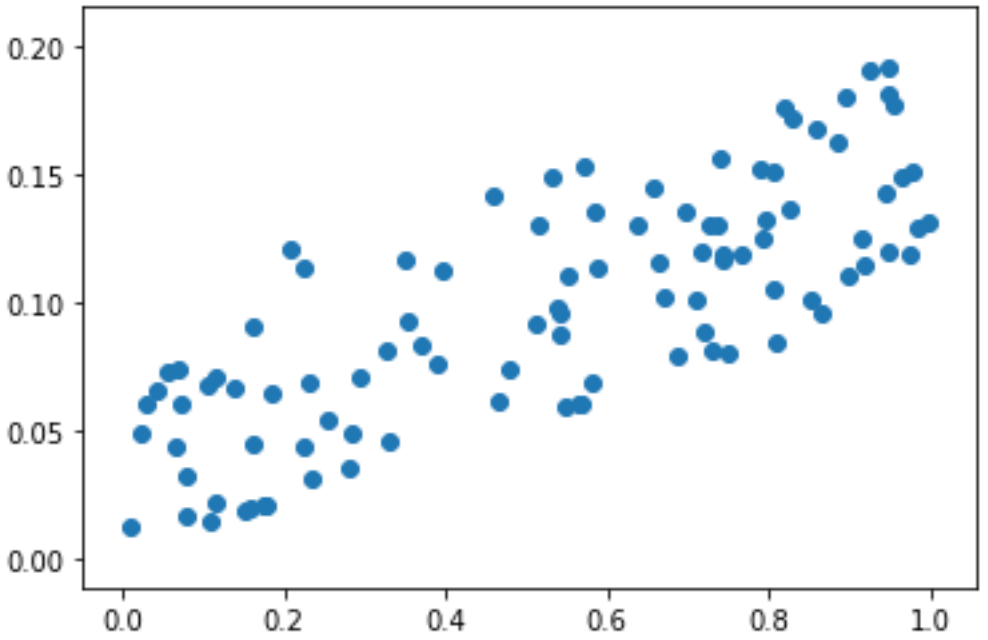
Training Workflow





LinearRegression

- 在這裡鍵入方程式。範例程式：LinearRegression_tf.ipynb
- 目的：使用線性方程式來描述這些資料點，找出線性方程式的兩個參數斜率w_1、截距w_0。
- $y = w_1 x + w_0$



$$\widehat{w_0} = \bar{y} - \widehat{w_1}\bar{x}$$

$$\widehat{w_1} = \frac{\sum_{i=1}^{n}(y_i - \bar{y})(x_i - \bar{x})}{\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

Data

```
# Random 100 points by numpy
x_data = np.random.rand(100).astype(np.float32)
y_data = x_data * 0.1 + 0.1*np.random.rand(100).astype(np.float32)

# plot data
plt.scatter(x_data, y_data)
plt.show()
```

Model building (1)

```
# Try to find values for W and b that compute y_data = W * x_data + b
# (We know that W should be 0.1 and b 0.03, but TensorFlow will
# figure that out for us.)
# Use tensorflow to find weighting of fitting


W = tf.Variable(tf.random_uniform([1], -1.0, 1.0))
b = tf.Variable(tf.zeros([1]))
y = W * x_data + b
```

Model building (2)

```
# Minimize the mean squared errors.
loss = tf.reduce_mean(tf.square(y - y_data))
optimizer = tf.train.GradientDescentOptimizer(0.2)
train = optimizer.minimize(loss)
```

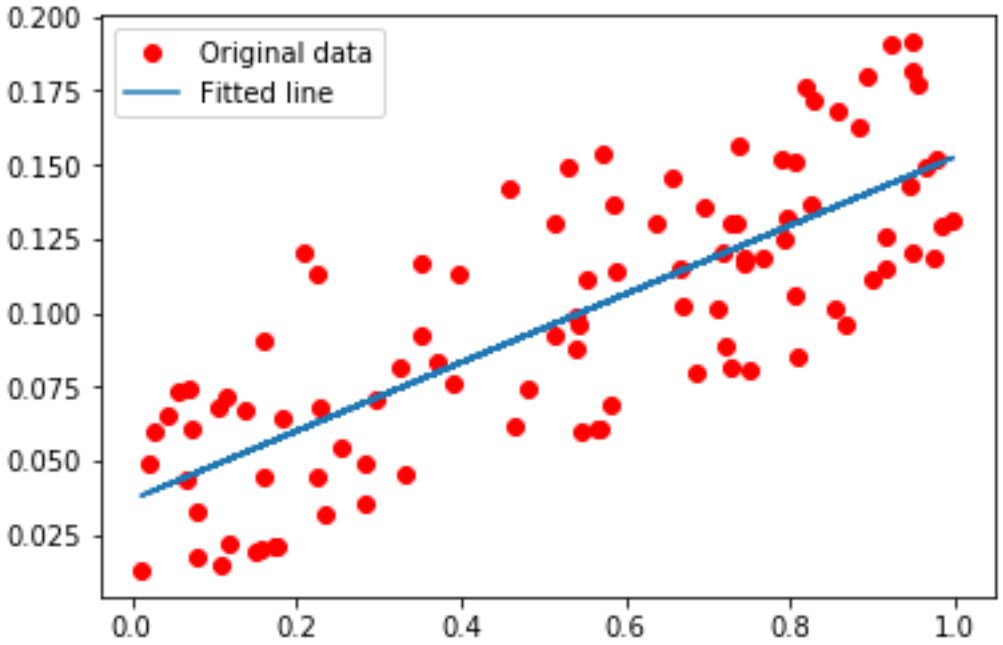
Training(1)

```
# Before starting, initialize the variables.  We will 'run' this first.
init = tf.global_variables_initializer()


# Launch the graph.
sess = tf.Session()
sess.run(init)
```

Training(2)

```
# Fit the line.
for step in range(2001):
    sess.run(train)
    if step % 400 == 0:
        print(step, sess.run(W), sess.run(b))
        plt.plot(x_data, y_data, 'ro', label='Original data')
        plt.plot(x_data, sess.run(W) * x_data + sess.run(b),
label='Fitted line')
        plt.legend()
        plt.show()


# Learns best fit is W: [0.1], b: [0.03]
```

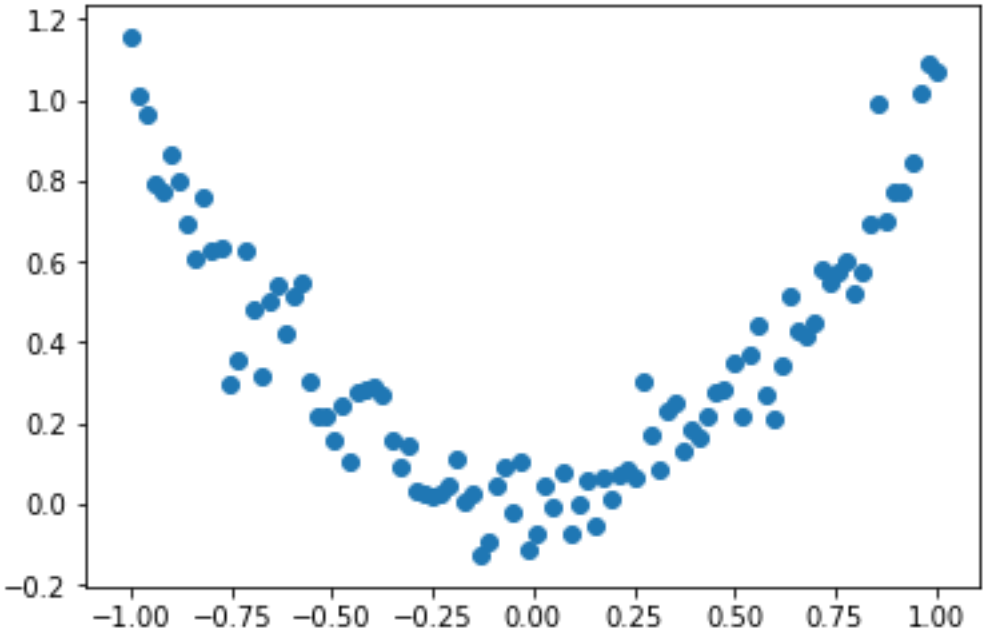
Result

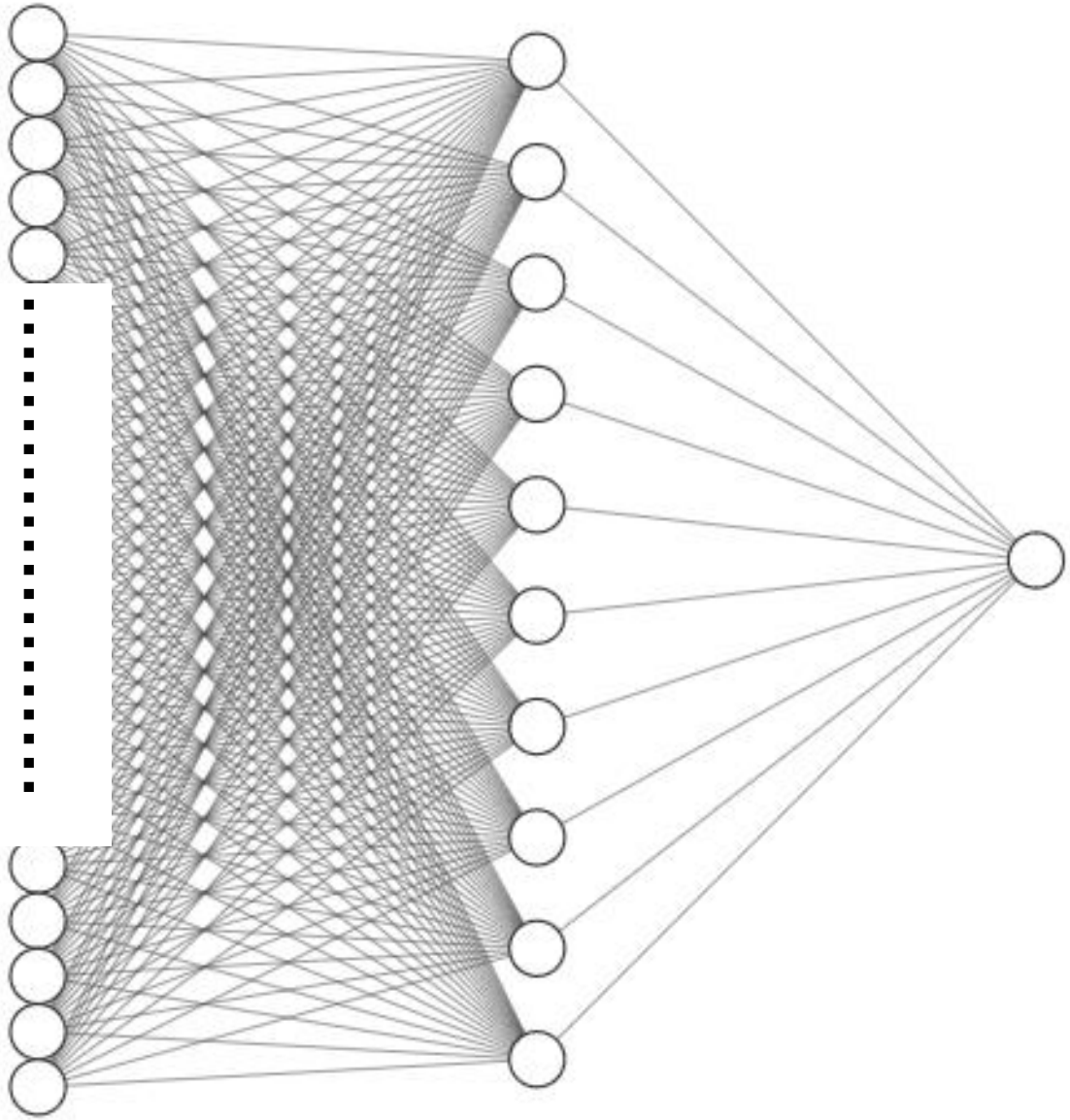


Iter:450 W:[0.11574703] b:[0.03685103]

Regression

- 範例程式： regression_tf_nn.ipynb





Data

```
tf.set_random_seed(342)
np.random.seed(685)


# preparing dataset
x = np.linspace(-1, 1, 100)[:, np.newaxis]          # shape (100, 1)
noise = np.random.normal(0, 0.1, size=x.shape)
# 1st arg: mean of the distribution, 2nd arg: Standard deviation of the
distribution
# y =x^2
y = np.power(x, 2) + noise                          # shape (100, 1) +
white noise


# plot data
plt.scatter(x, y)
plt.show()
```

https://docs.scipy.org/doc/numpy-1.15.0/reference/generated/numpy.random.normal.html

Model building (1)

```
# constructing NN
# https://www.tensorflow.org/api_docs/python/tf/placeholder
tf_x = tf.placeholder(tf.float32, x.shape)     # input x
tf_y = tf.placeholder(tf.float32, y.shape)     # input y
```

Model building (2)

```
# neural network layers
# https://www.tensorflow.org/api_docs/python/tf/layers/dense
l1 = tf.layers.dense(tf_x, 10, tf.nn.relu)          # hidden layer
output = tf.layers.dense(l1, 1)                     # output layer
```

Model building (3)

```
#
https://www.tensorflow.org/api_docs/python/tf/losses/mean_squared_error
loss = tf.losses.mean_squared_error(tf_y, output)   # compute cost


#
https://www.tensorflow.org/api_docs/python/tf/train/GradientDescentOptim
izer
optimizer = tf.train.GradientDescentOptimizer(learning_rate=0.5)
train_op = optimizer.minimize(loss)
```

Training(1)

```
# https://www.tensorflow.org/api_docs/python/tf/Session

sess = tf.Session()                                    # control training
and others

sess.run(tf.global_variables_initializer())            # initialize var in
graph
```

Training(2)

```
for step in range(100):
    # train and net output
    _, l, pred = sess.run([train_op, loss, output], {tf_x: x, tf_y: y})
    if step % 5 == 0:
        # plot and show learning process
        plt.cla()
        plt.scatter(x, y)
        plt.plot(x, pred, 'r-', lw=5)
        plt.text(0.5, 0, 'Loss=%.4f' % l, fontdict={'size': 20, 'color':
'red'})
        plt.pause(0.1)
```

Result

Iter:95