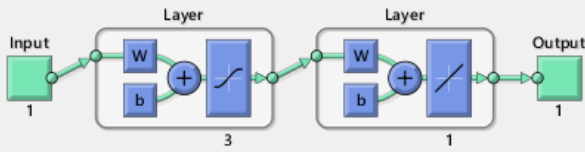


Problem 1 (MATLAB neural network toolbox)

Neural Network Training (nntraintool)

Neural Network



Algorithms

Training: Levenberg-Marquardt (trainlm)
Performance: Mean Squared Error (mse)
Calculations: MEX

Progress

Epoch:	0	58 iterations	2000
Time:	0:00:00		
Performance:	0.466	0.000996	0.00100
Gradient:	0.211	0.00614	1.00e-07
Mu:	0.00100	1.00e-05	1.00e+10
Validation Checks:	0	0	6

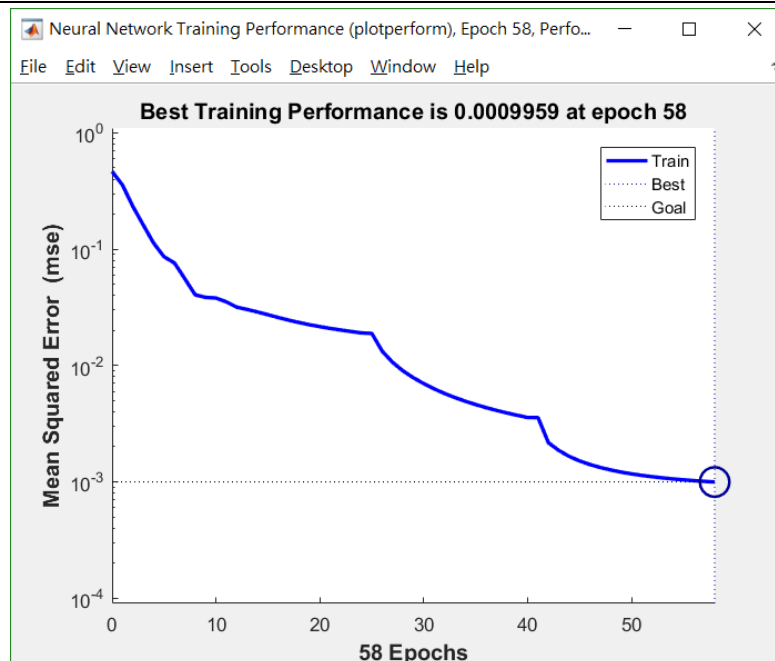
Plots

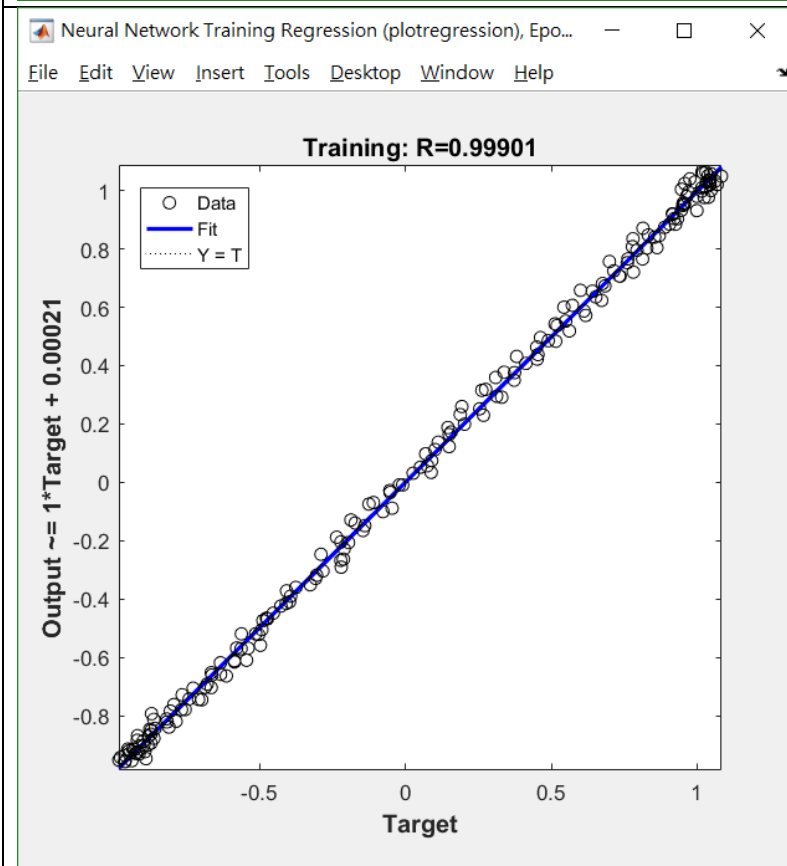
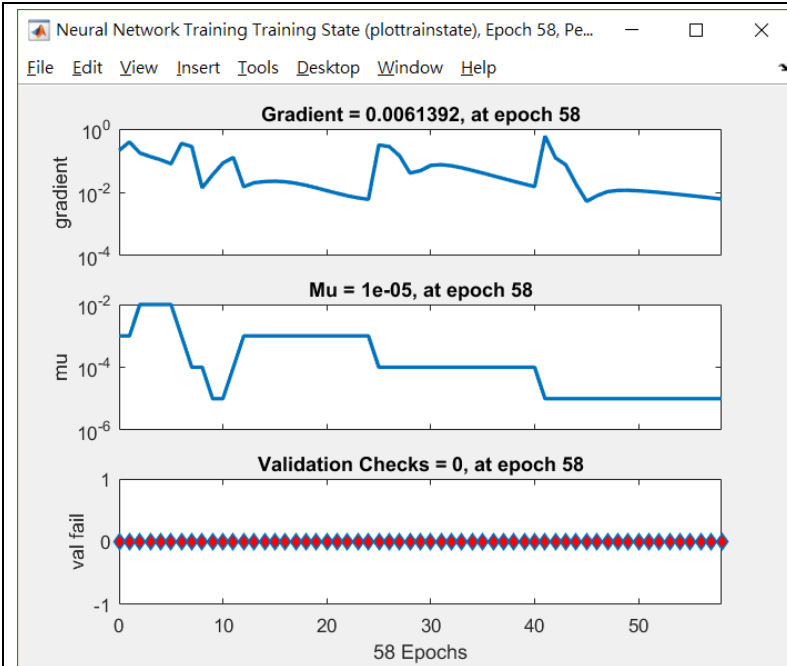
Performance (plotperform)
Training State (plottrainstate)
Regression (plotregression)

Plot Interval: 1 epochs

✓ Performance goal met.

Stop Training Cancel





In how many epochs your NN converges?

因為 Y 每次的雜訊都不同，且 w 開始的數值也不一樣，因此每次達到收斂條件($MSE < 0.001$)所需的 epochs 次數也不同，以上圖這次為例 epochs 次數為 58。

Calculate the root-mean-squared error of simulation result between true output and predicted output.

公式: $\text{Root-mean-squared error} = \sqrt{\text{mean}((Y - \text{simY}).^2)}$

這次的 RMSE 經計算為 0.0316(3.16%)。

Problem 2 (Convolutional neural network)

參數設定:

`rand('state',0)`

```

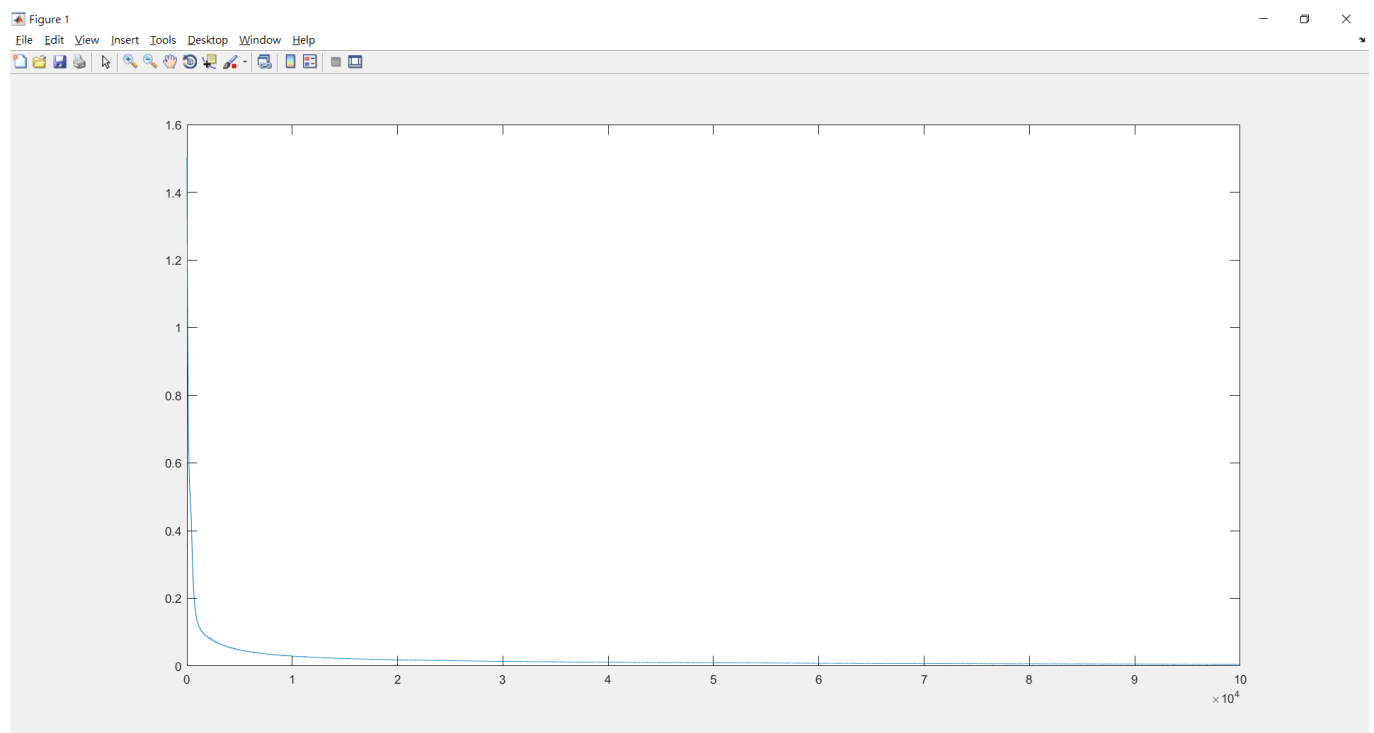
cnn.layers = {
    struct('type', 'i') %input layer
    struct('type', 'c', 'outputmaps', 6, 'kernelsize', 5) %convolution layer
    struct('type', 's', 'scale', 2) %sub sampling layer
    struct('type', 'c', 'outputmaps', 12, 'kernelsize', 5) %convolution layer
    struct('type', 's', 'scale', 2) %subsampling layer
};

```

```

opts.alpha = 1;
opts.batchsize = 50;
opts.numepochs = 1000;

```



Error = 0.032 (3.2%)

上面的測試 **batchsize** 設定為 50，**epoch** 數量設定為 1000，訓練時間約一個半小時，得出來的結果還不錯，準確率大概為 96.8%，但由上圖可以看出其實 **epoch** 數量大約在 500~600 間就收斂到 **error** 很小了。