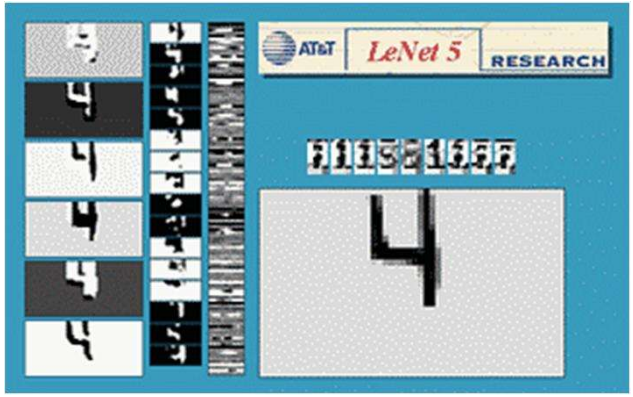
22.3 Convolutional neural networks

- improvement over the multilayer networks
  - performance, accuracy, and some degree of invariance to distortions in the input images.

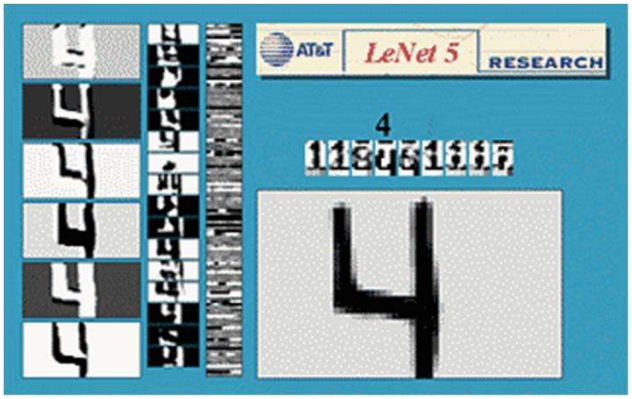
LeNet5 http://yann.lecun.com/exdb/lenet/index.html

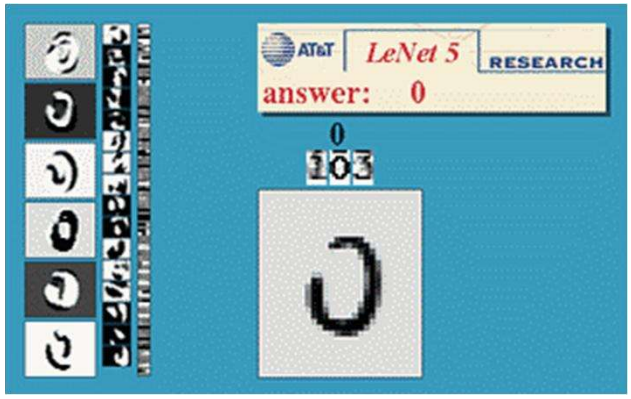
- 1998年Yann LeCun提出。

Invariance: shift

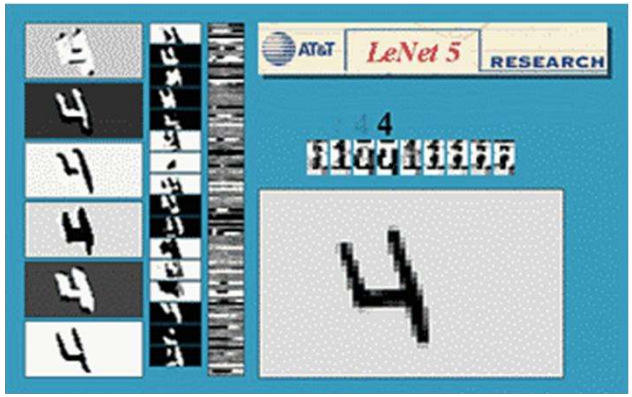


Invariance: scale

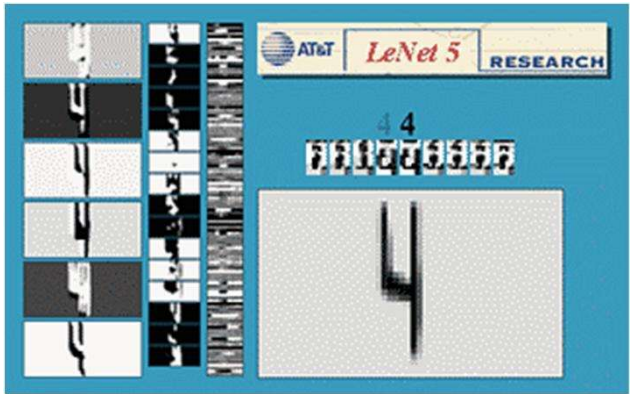




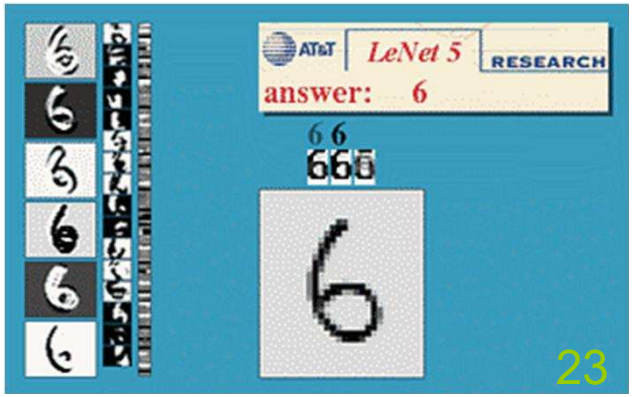
Invariance: rotation



Invariance: squeezing

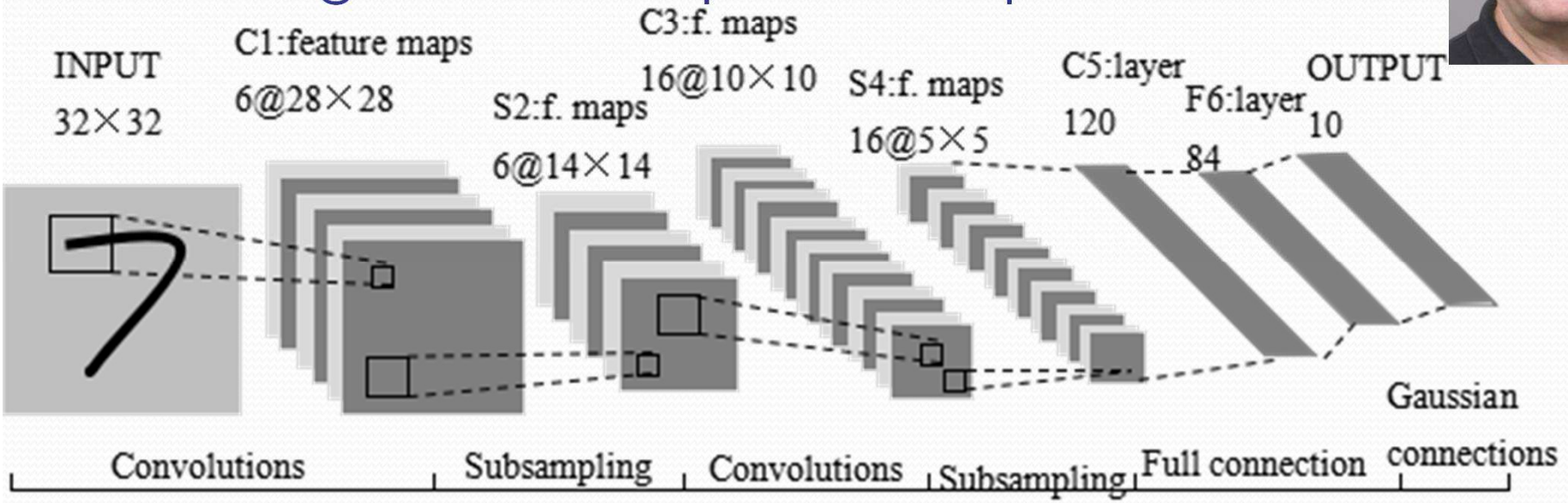


Invariance: stroke width



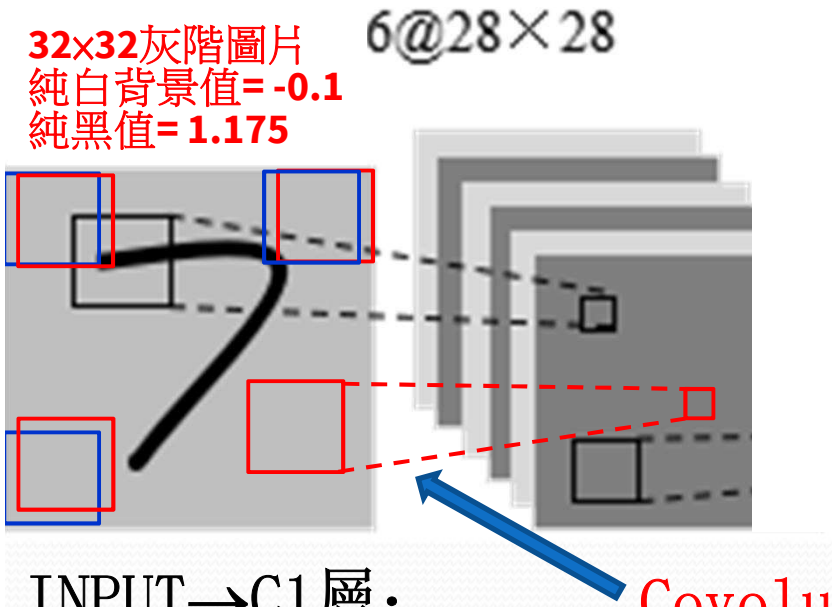
典型的Convolutional neural networks

- **LeNet-5手寫數字識別，**1998年Yann LeCun提出。正確率高達99.2%。
- raw image of 32 x 32 pixels as input.

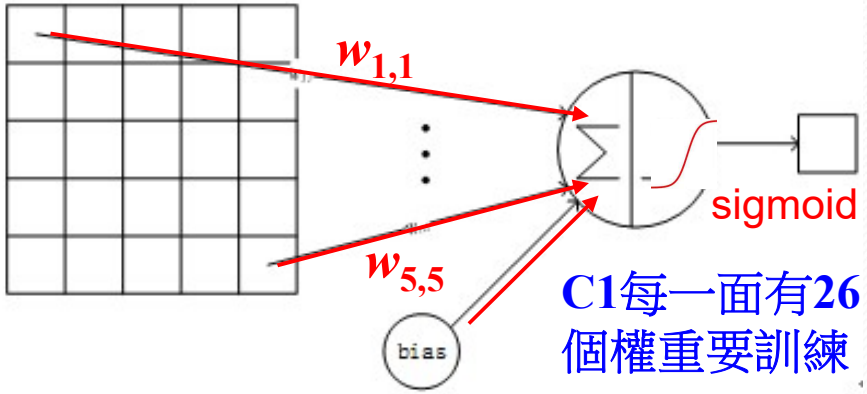


- 共有7層。
- C1,C3,C5 : Convolutional layer. 5×5 Convolution window/kernel/matrix.
- S2 , S4 : Subsampling layer. Subsampling factor=2.
- F6 : Fully connected layer.

32x32灰階圖片
純白背景值= -0.1
純黑值= 1.175



卷積窗大小：5×5



C1每一面有26個權重要訓練

INPUT→C1層：

輸入圖片大小：1024=(32×32)

卷積窗大小：5×5

卷積窗種類：6(因c1有6個平面)

輸出特徵圖數量：6

輸出特徵圖大小：28×28

神經元數量：4704=(28×28)×6

連接線數量：122304=(28×28)×6×(5×5+1)

可訓練參數：156 = 6面×(5×5+1)

Covolutions例：6x6黑白相片

1	0	0	0	0	1
0	1	0	0	1	0
0	0	1	1	0	0
1	0	0	0	1	0
0	1	0	0	1	0
0	0	1	0	1	0

卷積窗#1
(另有一個bias的參數，假設是0)

1	-1	-1
-1	1	-1
-1	-1	1

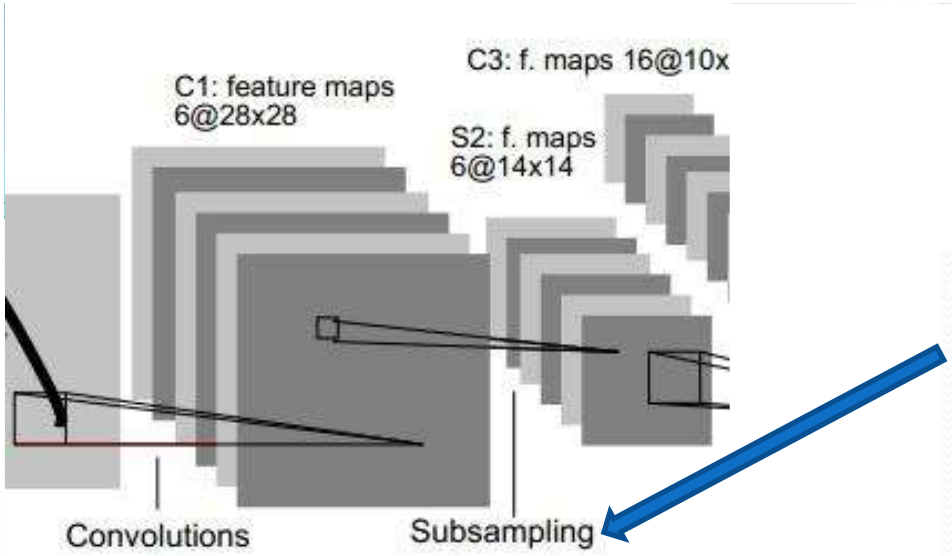
3	-1	-3	-1	C1層1st面
-3	1	0	-3	
-3	-3	0	1	
3	-2	-2	-1	

卷積窗#2

-1	1	-1
-1	1	-1
-1	1	-1

-1	-1	-1	-1	C1層2nd面
-1	-1	-2	1	
-1	-1	-2	1	
-1	0	-4	3	

優點：局部連接、參數共用！



C1→S2層：4格變1格

輸入圖片大小：(28×28)×6

子採樣窗大小：2×2

子採樣窗種類：6

輸出子採樣圖數量：6

輸出子採樣圖大小：(14×14)×6

神經元數量：1176=(14×14)×6

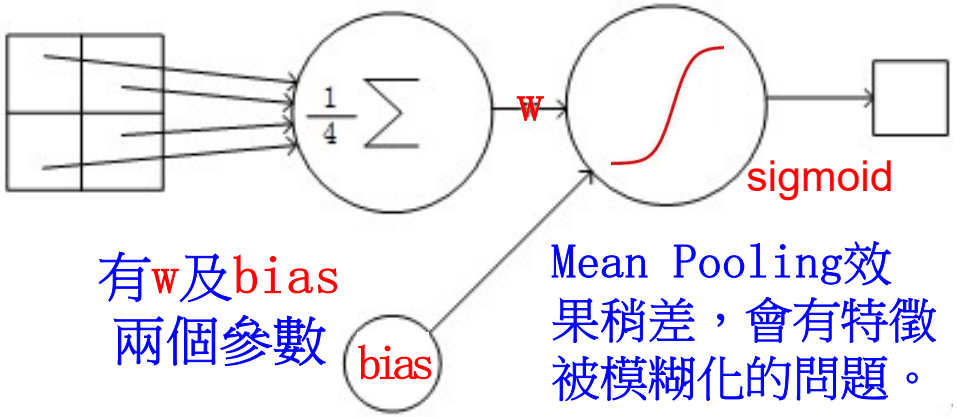
連接線數量：5880=(4+1)×(14×14)×6

可訓練參數：12=6×2　　//w及bias

C3→S4層也是一樣的設計。

Subsampling有兩種作法：

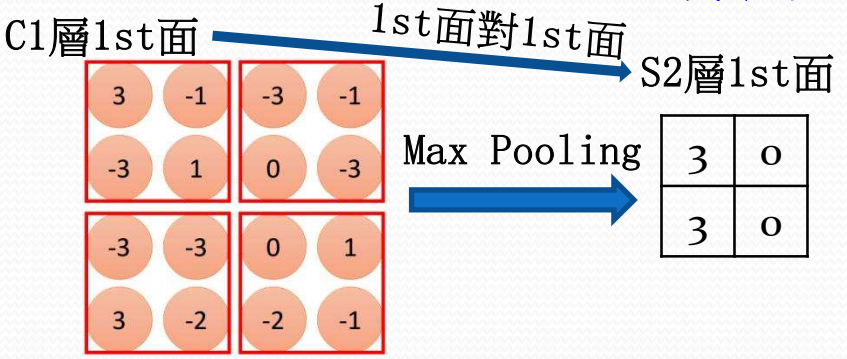
甲、Mean Pooling(LeNet-5採用)

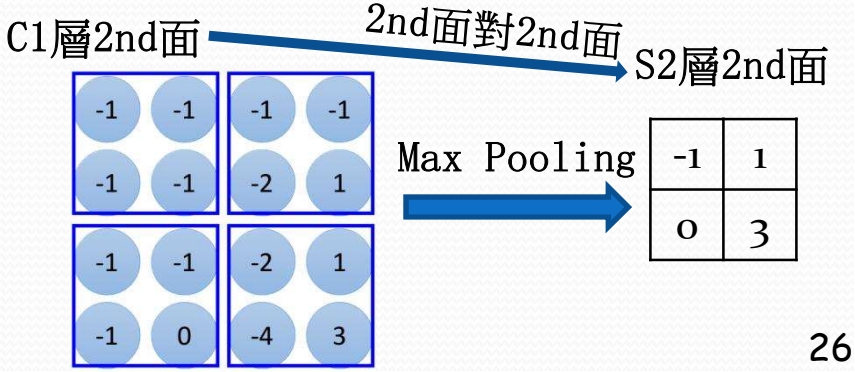


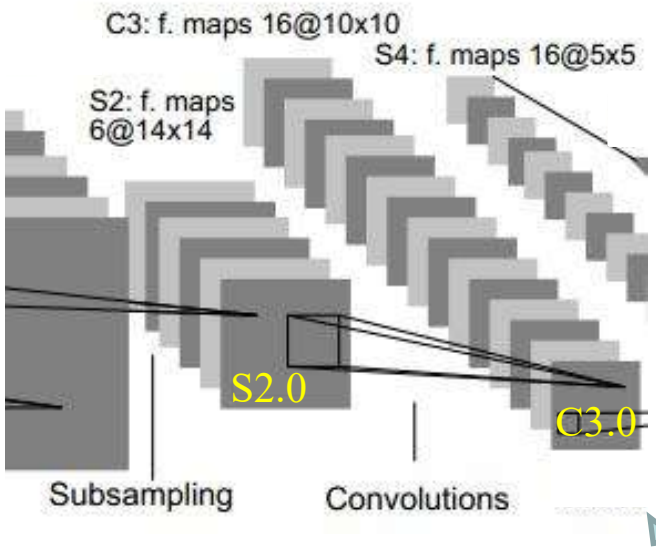
有w及bias
兩個參數

Mean Pooling效果稍差，會有特徵被模糊化的問題。

乙、Max Pooling(AlexNet採用)







C3 / S2	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
S2.0	X				X	X	X			X	X	X	X		X	X
S2.1	X	X				X	X	X			X	X	X	X		X
S2.2	X	X	X				X	X	X			X		X	X	X
S2.3		X	X	X			X	X	X	X			X		X	X
S2.4			X	X	X			X	X	X	X		X	X		X
S2.5				X	X	X			X	X	X	X		X	X	X

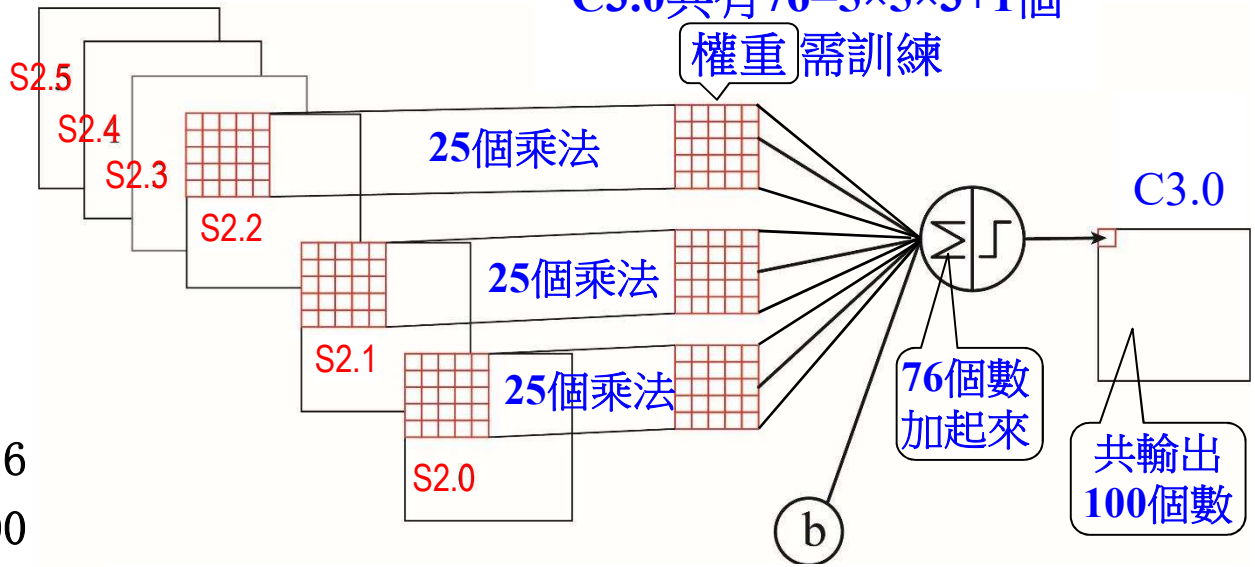
相鄰3個　相鄰4個　相鄰2個 1個

=3+4+2+1

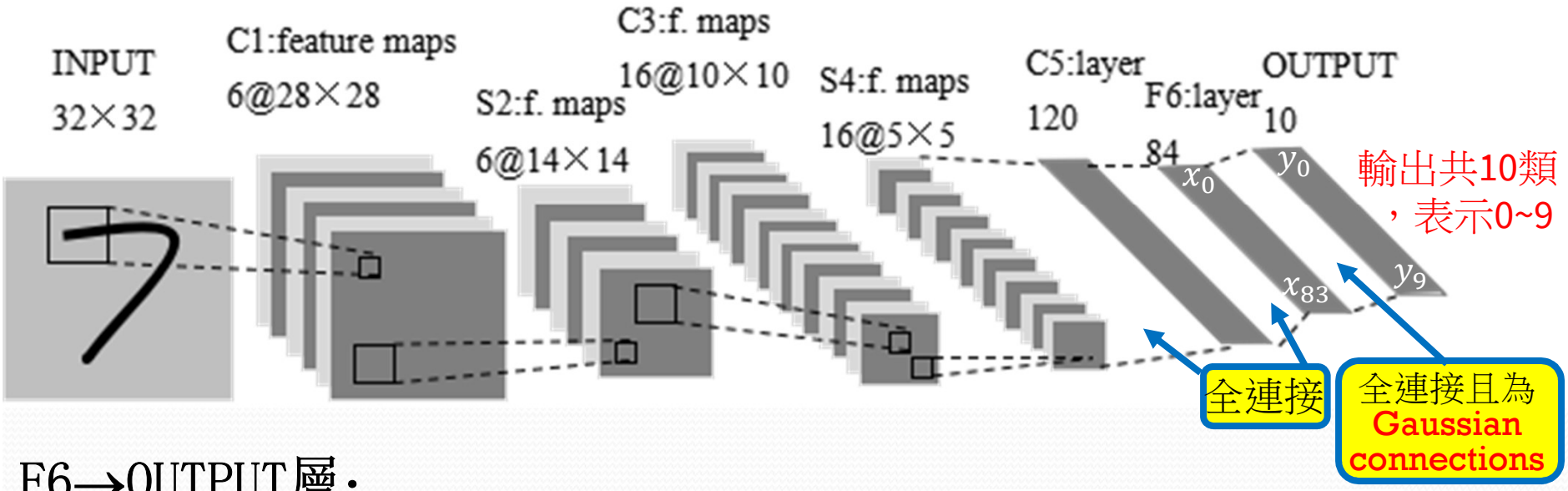
卷積窗種類=3×6+4×9+6=60，
也就是X的個數

S2→C3層：選擇性1個對10個的連接

輸入圖片：(14×14)×6
卷積窗大小：5×5
卷積窗種類：60
輸出特徵圖數量：16
輸出特徵圖大小：10×10
神經元數量：160=(10×10)×16
連接線數量：151600=1516×100
可訓練參數：1516



C3.0～C3.5均各自有76個權重需訓練
C3.6～C3.14均各自有101=5×5×4+1個權重需訓練
C3.16有151=5×5×6+1個權重需訓練
∴ C3總共有1516=76×6+101×9+151個權重需訓練



F6→OUTPUT層：

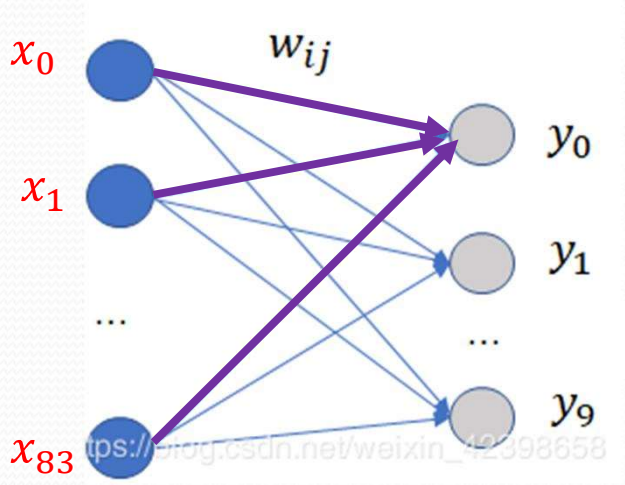
輸入圖片大小：1×84

輸出特徵圖數量：1×10

神經元數量：10

連接數：840=84×10

可訓練參數：850=(84+1)×10



Gaussian connections：計算較特別

$$y_i = \sum_{j=0}^{83} (x_j - w_{ij})^2$$

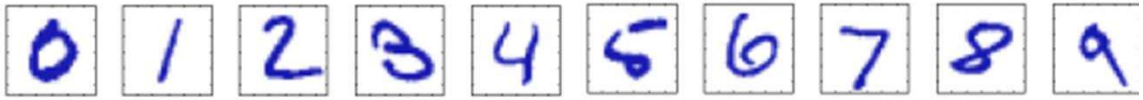
- 輸出 y_i=所有輸入減去權值的距離平方和。
- 輸入 x_j 離參數 w_{ij} 向量越遠，y_i 值越大，代表匹配數字 i 的程度越低。
- 反之，y_i 值越接近於 0，則表示識別結果越接近數字 i。

LeNet-5共約有14,000個訓練參數，187,000個連接線。

Case study: Handwritten Digits Recognition

實戰演練： **Everything is here.**

Mike O'Neill



http://www.codeproject.com/Articles/16650/Neural-Network-for-Recognition-of-Handwritten-Digi

- A Convolutional Neural Network achieves 99.26% accuracy on a Modified National Institute of Standards and Technology (MNIST) database of hand-written digits. 此CNN總共有5 layers, 3215 neurons, 134066 weights, 184974 connections.
- MNIST database : Consist of 60,000 hand written digits uniformly distributed over 0-9.
  - 電腦中需先灌**Microsoft Visual Studio**。
  - 打開**MNist.dsw (in \Demo-MNist)**, 選擇要轉換並開啟這個專案。

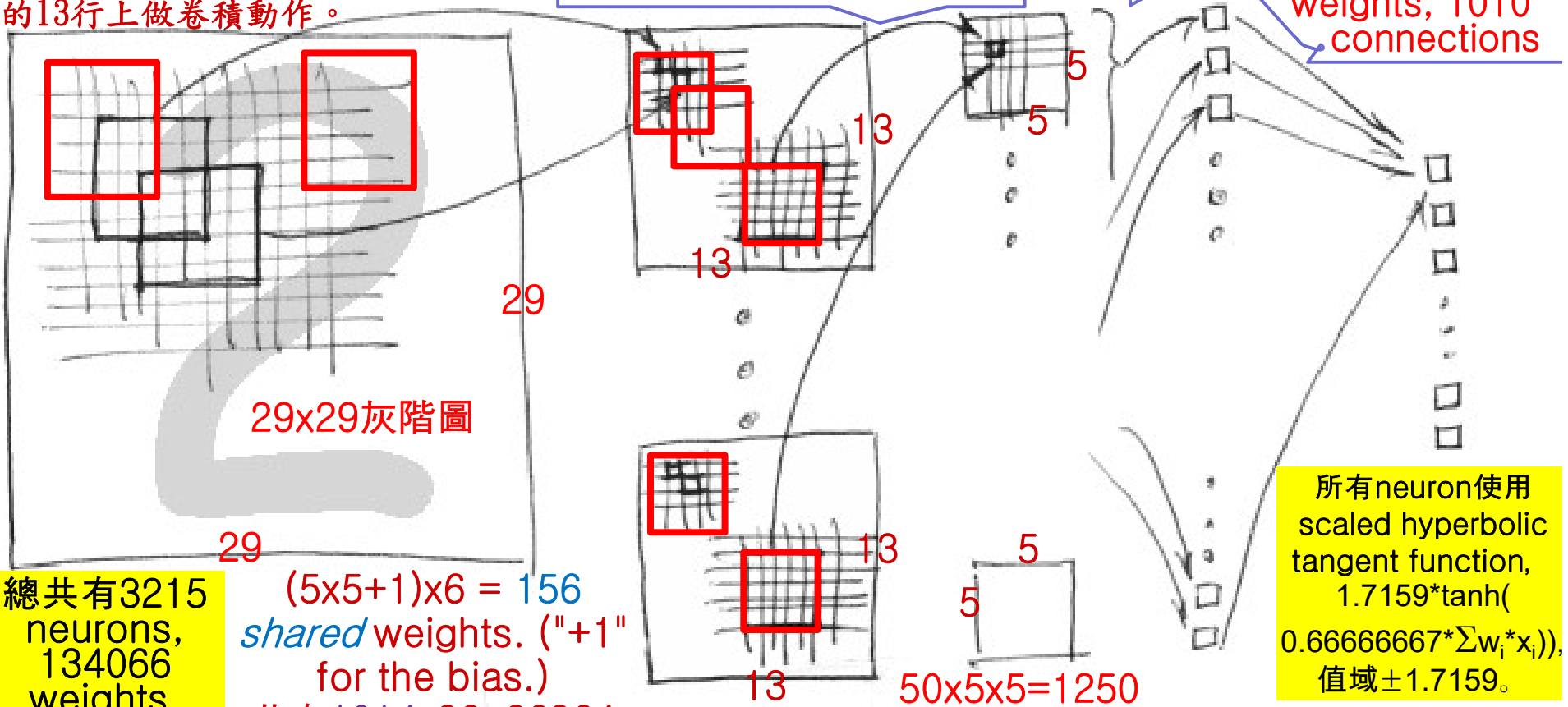
神經網路架構

5x5 convolutional kernel，只在行、列編號爲0、2、4、…、24的13行上做卷積動作。

只在行、列編號爲0、2、4、6、8的5行上做卷積動作。
(5x5+1)x6x50=7800 weights,
1250x26=32500 connections

100×(1250+1)=125100
weights, 125100 connections

10x(100+1)=1010
weights, 1010
connections



所有neuron使用
scaled hyperbolic
tangent function,
1.7159*tanh(
0.66666667*Σw_i*x_i)),
值域±1.7159。

總共有3215
neurons,
134066
weights,
184974
connections.

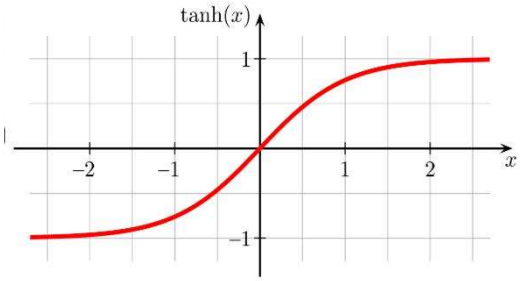
(5x5+1)x6 = 156
shared weights. ("+1"
for the bias.)
共有1014x26=26364
connections

50x5x5=1250
neurons

Input Layer
29x29 =
841 neurons

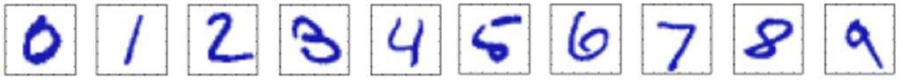
Layer #1
6 Feature Maps
Each 13x13
6x13x13 =1014 neurons,
每個都有(5x5+1)=26 connections

Layer #2
50 Feature
Maps
Each 5x5



操練步驟

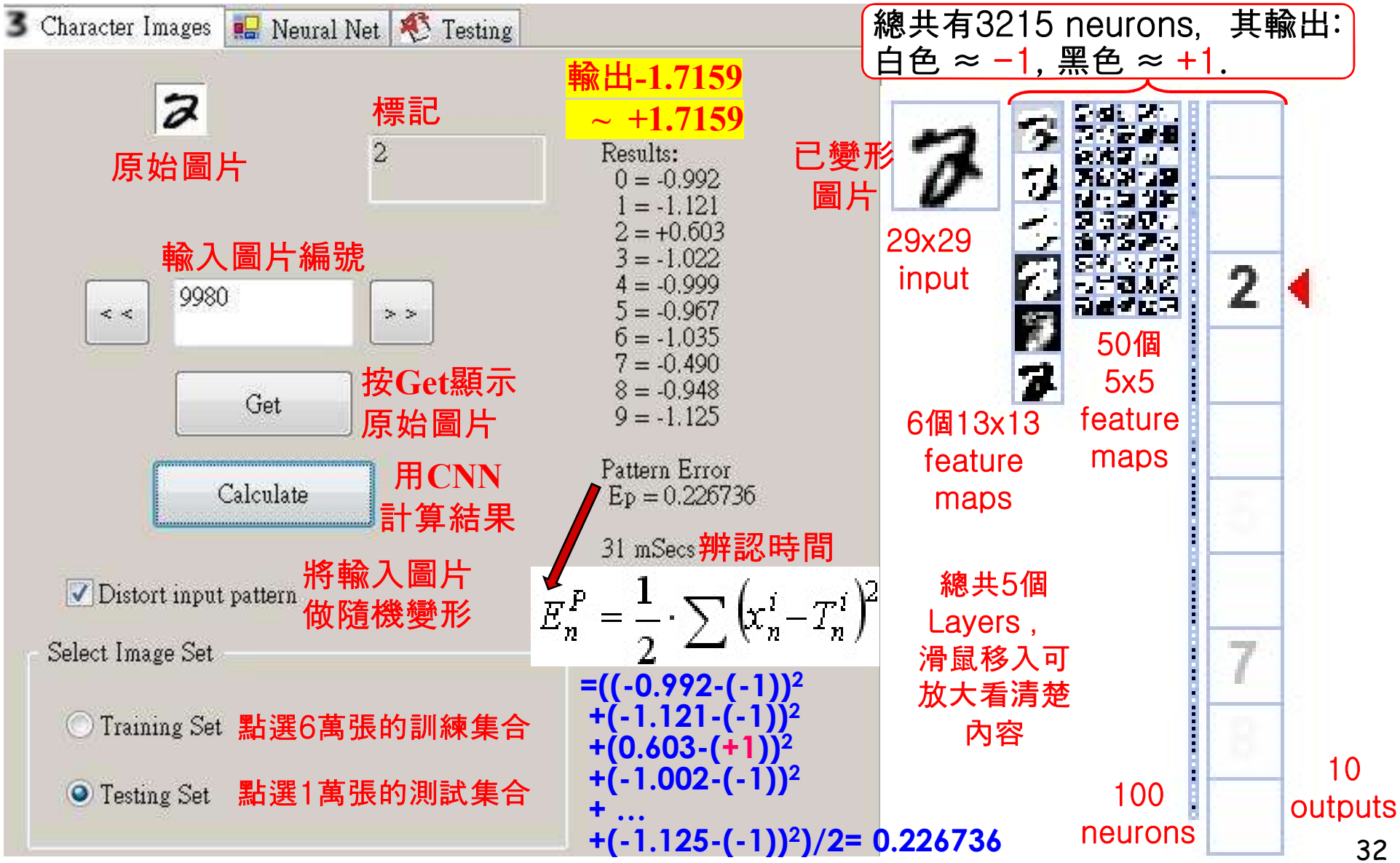
http://www.codeproject.com/Articles/16650/Neural-Network-for-Recognition-of-Handwritten-Digi

- Download the Neural Network demo project - 203 Kb, 取得Demo-Mnist整個開源碼及執行檔專案。203KB
- Download a sample neuron weight file - 2,785 Kb (achieves the 99.26% accuracy mentioned above)simpleneutronweightfile.zip解壓縮取得10September-PolishedWithUndistorted-7epochs-dot026MSE-74Errors.nnt。這是已訓練好的神經網路權重參數檔案。2.7MB
- Download the MNIST database - 11,594 Kb total for all four files，解壓縮取得 train-images.idx3-ubyte訓練用的6萬張手寫數字檔案、46MB
  train-labels.idx1-ubyte訓練用的6萬張手寫數字的標記檔案、59KB
  t10k-images.idx3-ubyte測試準確率用的1萬張手寫數字檔案、7.6MB
  t10k-labels.idx1-ubyte測試準確率用的1萬張手寫數字的標記檔案。10KB
- Run the Demo Program MNist.exe(in \Demo-MNist\Unicode Release), click the "Open MNist" button, and sequentially select the four MNIST database files: train-images.idx3-ubyte, train-labels.idx1-ubyte , t10k-images.idx3-ubyte , t10k-labels.idx1-ubyte. Then choose File->Open to open the weights file10September-PolishedWithUndistorted-7epochs-dot026MSE-74Errors.nnt.
- 再點選Image Set, 就可開始辨認數字了。 

Character Images：立刻逐張辨識，厲害吧。

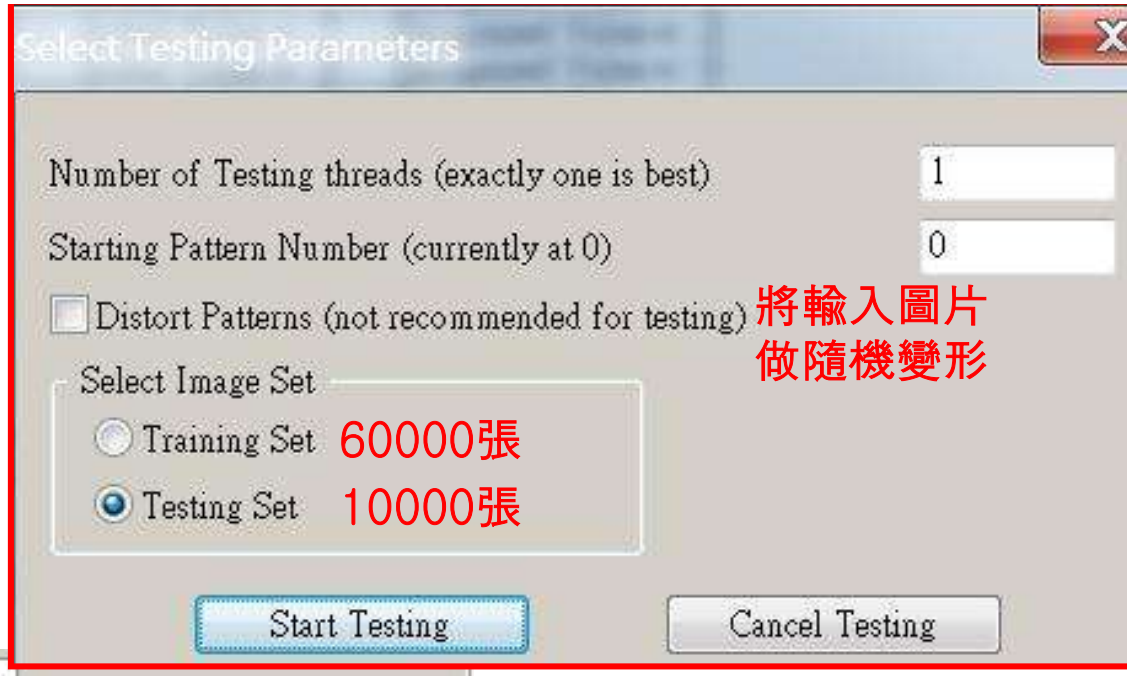
Run the Demo Program MNist.exe(in \Demo-MNist\Unicode Release)

建議開始時先不要將輸入圖片做隨機變形，直接看辨識結果。



Testing View

點選**Testing**
再按**Start Testing**
做大量測試。



←**10000張共有74張認錯字。**

認錯字，**4認成9**。

Accuracy

$$= \frac{10000 - 74}{10000}$$

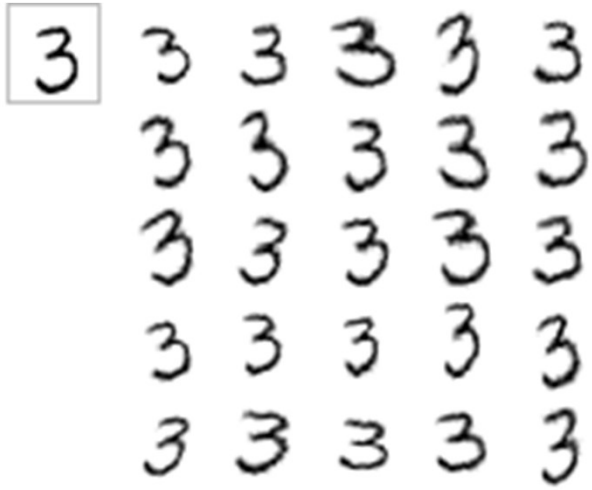
$$= 99.26\%$$

MSE(mean squared error)=

10000張 $E_n^P = \frac{1}{2} \cdot \sum \left(x_n^i - T_n^i\right)^2$ 的平均

Distortion techniques for improving generalization

在訓練時將輸入圖片做隨機變形。每次看到都不一樣，等同於增加訓練資料集合的數量。

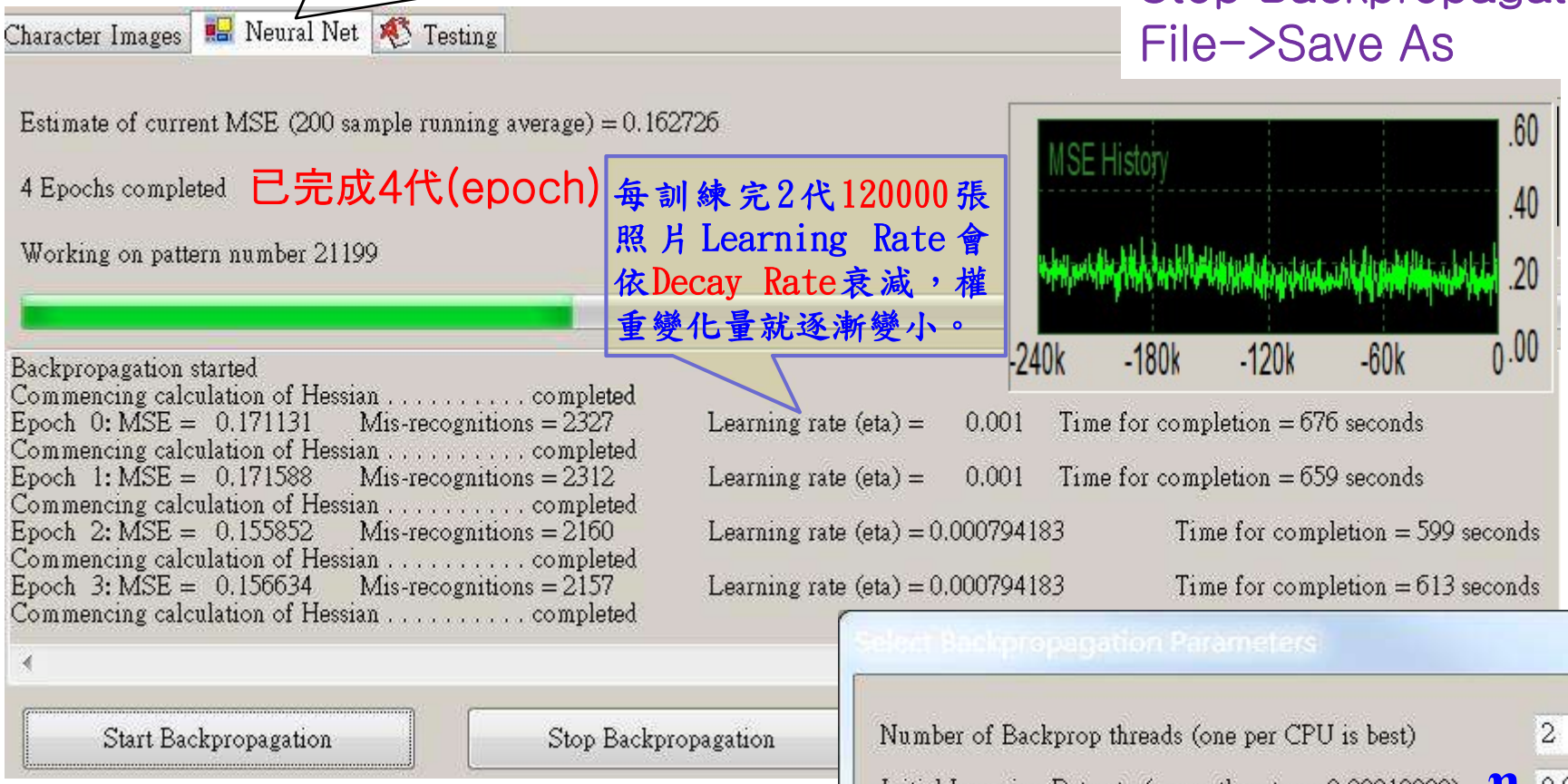


Three types of distortions are applied:

1. **Scale Factor:** 水平及垂直方向各自獨立地拉長或縮短。
2. **Rotation:** 順時針或反時針旋轉整個圖。
3. **Elastic:** 類似水波紋將部份**pixels**隨機彈性推擠。

Training View你也來訓練它吧。

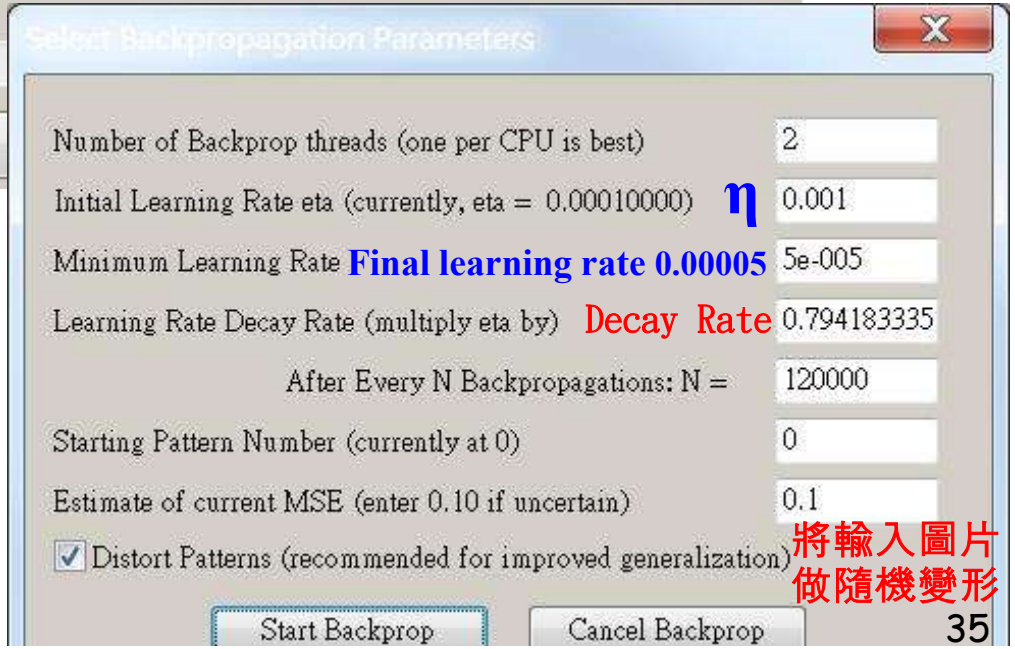
Start Backpropagation ->
Stop Backpropagation ->
File->Save As



每代用backpropagation algorithm (BP)以隨機次序處理完60000張訓練照片。

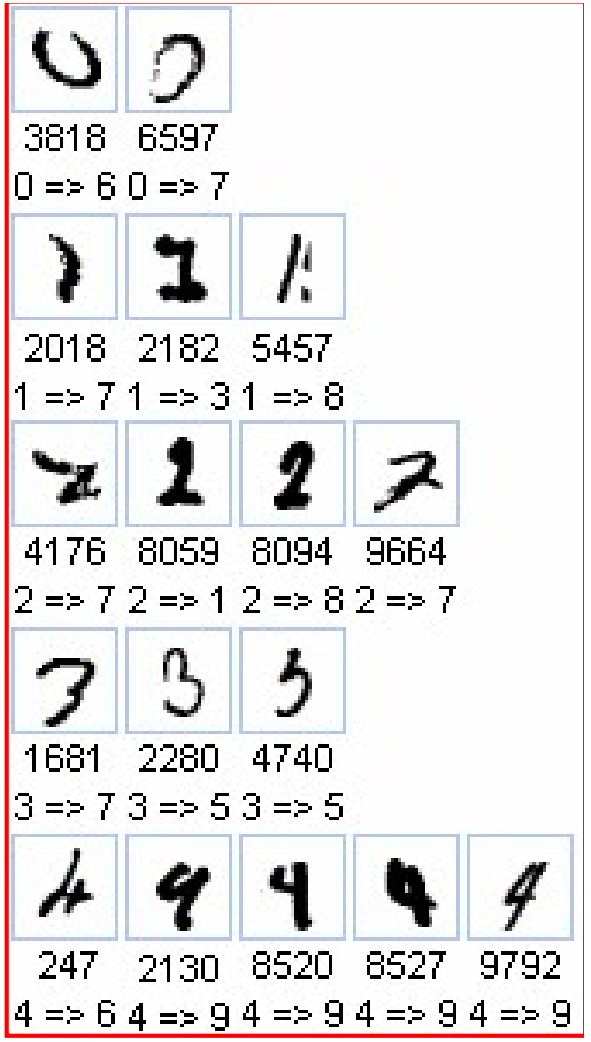
MSE(mean squared error)=

60000張 $E_n^P = \frac{1}{2} \cdot \sum \left(x_n^i - T_n^i\right)^2$ 的平均



Misrecognized results in Testing

10000張測試圖片共有
2+3+4+3+5+
12+12+7+13+13=74張認錯字。



Accuracy

$$= \frac{10000 - 74}{10000}$$

=99.26%