Neural-Network 2

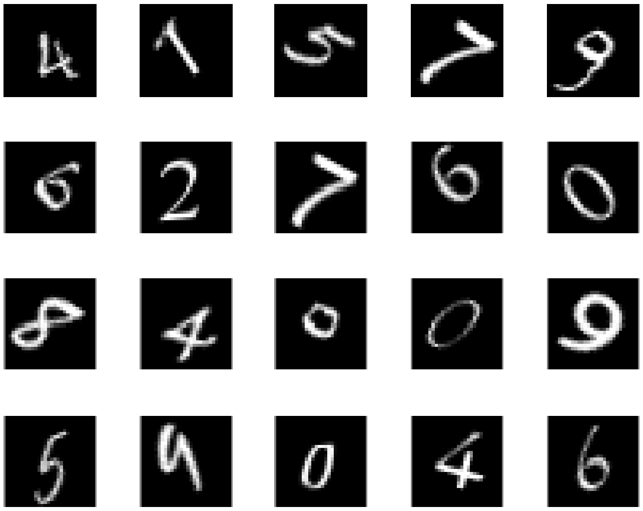
Created time : 2024/5/17 09:20

學號 : 109321019　姓名 : 涂价弘

```
clear
clc
```

步驟 1：加載圖像樣本數據，並顯示其中的部分圖像

```
digitDatasetPath = fullfile(matlabroot,'toolbox','nnet','nndemos', ...
    'nndatasets','DigitDataset');
imds = imageDatastore(digitDatasetPath, ...
    'IncludeSubfolders',true,'LabelSource','foldernames');
figure;
perm = randperm(10000,20);
for i = 1:20
    subplot(4,5,i);
    imshow(imds.Files{perm(i)});
end
```



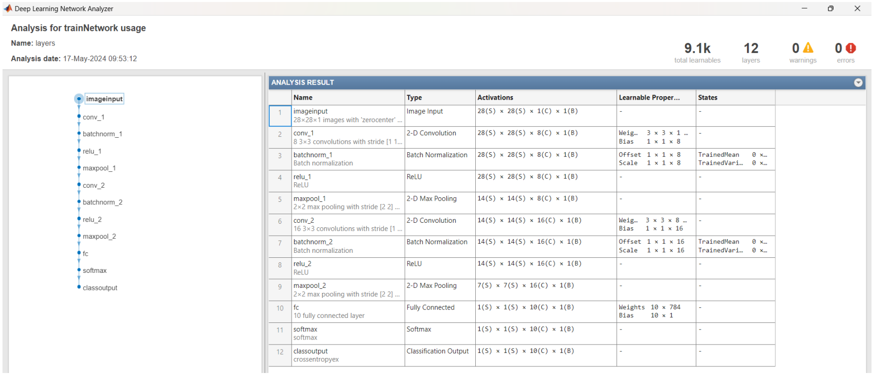
步驟 2：將加載的圖像樣本分為訓練集和測試集（註：在本例中，訓練集的數量為 750 幅，剩餘的為測試集）

```
numTrainFiles = 750;
[imdsTrain,imdsValidation] = splitEachLabel(imds,numTrainFiles,'randomize');
```

步驟 3：構建卷積網絡（註：可以在該部分進行相關參數的設置改進）

```
layers = [
    imageInputLayer([28 28 1])                     % 输入层，1 个通道，像素为 28×28

    convolution2dLayer([3 3],8,'Padding','same')   % 卷积层 1：卷积核大小为 3×3，卷积核
的个数为 8（每个卷积核的通道数与输入图像的通道数相等，本层中每个卷积核 1 个通道）卷积的方式采
用零填充方式（即设定为 same 方式）
    batchNormalizationLayer                        % 批量归一化层 1
    reluLayer                                      % ReLu 非线性激活函数 1
    maxPooling2dLayer(2,'Stride',2)                % 池化层 1：池化方式：最大池化；池化
区域为 2×2，步长为 2

    convolution2dLayer([3 3],16,'Padding','same') % 卷积层 2：卷积核大小为 3×3，卷积核
的个数为 16（每个卷积核的通道数与输入特征图的通道数相等，本层中每个卷积核 8 个通道）卷积的方
式采用零填充方式（即设定为 same 方式）
    batchNormalizationLayer                        % 批量归一化层 2
    reluLayer                                      % ReLu 非线性激活函数 2
    maxPooling2dLayer(2,'Stride',2)                % 池化层 2：池化方式：最大池化；池化
区域为 2×2，步长为 2

    fullyConnectedLayer(10)                        % 全连接层：将全连接层输出的个数设置
为 10 个
    softmaxLayer                                   % softmaxLayer 层：输出每个输出的概
率
    classificationLayer ];                         % 分类层：根据上一层的输入的概率，进
行分类并输出
```

```
analyzeNetwork(layers)
```



Analysis for trainNetwork usage

Name: layers

Analysis date: 17-May-2024 09:53:12

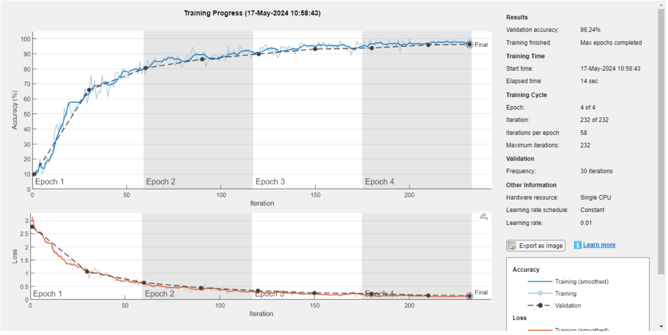
9.1k total learnables | 12 layers | 0 warnings | 0 errors

ANALYSIS RESULT

	Name	Type	Activations	Learnable Proper...	States
1	imageinput 28×28×1 images with 'zerocenter' ...	Image Input	28(S) × 28(S) × 1(C) × 1(B)	-	-
2	conv_1 8 3×3 convolutions with stride [1 1...	2-D Convolution	28(S) × 28(S) × 8(C) × 1(B)	Weig… 3 × 3 × 1 … Bias 1 × 1 × 8	-
3	batchnorm_1 Batch normalization	Batch Normalization	28(S) × 28(S) × 8(C) × 1(B)	Offset 1 × 1 × 8 Scale 1 × 1 × 8	TrainedMean 0 ×… TrainedVari… 0 ×…
4	relu_1 ReLU	ReLU	28(S) × 28(S) × 8(C) × 1(B)	-	-
5	maxpool_1 2×2 max pooling with stride [2 2] ...	2-D Max Pooling	14(S) × 14(S) × 8(C) × 1(B)	-	-
6	conv_2 16 3×3 convolutions with stride [1 ...	2-D Convolution	14(S) × 14(S) × 16(C) × 1(B)	Weig… 3 × 3 × 8 … Bias 1 × 1 × 16	-
7	batchnorm_2 Batch normalization	Batch Normalization	14(S) × 14(S) × 16(C) × 1(B)	Offset 1 × 1 × 16 Scale 1 × 1 × 16	TrainedMean 0 ×… TrainedVari… 0 ×…
8	relu_2 ReLU	ReLU	14(S) × 14(S) × 16(C) × 1(B)	-	-
9	maxpool_2 2×2 max pooling with stride [2 2] ...	2-D Max Pooling	7(S) × 7(S) × 16(C) × 1(B)	-	-
10	fc 10 fully connected layer	Fully Connected	1(S) × 1(S) × 10(C) × 1(B)	Weights 10 × 784 Bias 10 × 1	-
11	softmax softmax	Softmax	1(S) × 1(S) × 10(C) × 1(B)	-	-
12	classoutput crossentropyex	Classification Output	1(S) × 1(S) × 10(C) × 1(B)	-	-

步驟 4：配置訓練選項並開始訓練

```
options = trainingOptions('sgdm', ...
    'InitialLearnRate',0.01, ...
    'MaxEpochs',4, ...
    'Shuffle','every-epoch', ...
    'ValidationData',imdsValidation, ...
    'ValidationFrequency',30, ...
    'Verbose',false, ...
    'Plots','training-progress');

% 配置训练选项
%'sgdm'表示使用具有动量的随机梯度下降方法来训练网络；'InitialLearnRate'设置初始学习率为
0.01；'MaxEpochs'将最大训练轮数设置为 4；'Shuffle'表示打乱数据，'every-epoch'则代表每一轮
训练都打乱一次数据；'ValidationData'用于设置验证数据集，'ValidationFrequency'设置验证频
率为 30；'Verbose'设置为 false 则不显示进度信息；'Plots'打开训练进度图。

net = trainNetwork(imdsTrain,layers,options); %对网络进行训练
```



Results

Validation accuracy: 96.24%

Training finished: Max epochs completed

Training Time

Start time: 17-May-2024 10:58:43

Elapsed time: 14 sec

Training Cycle

Epoch: 4 of 4

Iteration: 232 of 232

Iterations per epoch: 58

Maximum iterations: 232

Validation

Frequency: 30 iterations

Other Information

Hardware resource: Single CPU

Learning rate schedule: Constant

Learning rate: 0.01

步驟 5：將訓練好的網絡用於對新的輸入圖像進行分類，並計算準確率

```
YPred = classify(net,imdsValidation);
YValidation = imdsValidation.Labels;
accuracy = sum(YPred == YValidation)/numel(YValidation)
```

```
accuracy = 0.9624
```

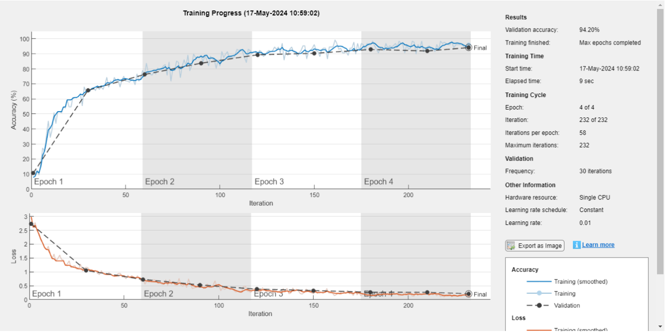
Default layer

```
layers = [
    imageInputLayer([28 28 1])

    convolution2dLayer([3 3],8,'Padding','same')
    batchNormalizationLayer
    reluLayer
    maxPooling2dLayer(2,'Stride',2)

    convolution2dLayer([3 3],16,'Padding','same')
    batchNormalizationLayer
    reluLayer
    maxPooling2dLayer(2,'Stride',2)

    fullyConnectedLayer(10)
    softmaxLayer
    classificationLayer ];
```

Exercise-1-1

去掉第二層卷積層

```
layers = [
    imageInputLayer([28 28 1])

    convolution2dLayer([3 3],8,'Padding','same')
    batchNormalizationLayer
    reluLayer
    maxPooling2dLayer(2,'Stride',2)

    fullyConnectedLayer(10)
    softmaxLayer
    classificationLayer ];

net = trainNetwork(imdsTrain,layers,options);
```



Results
Validation accuracy: 94.20%
Training finished: Max epochs completed
Training Time
Start time: 17-May-2024 10:59:02
Elapsed time: 9 sec
Training Cycle
Epoch: 4 of 4
Iteration: 232 of 232
Iterations per epoch: 58
Maximum iterations: 232
Validation
Frequency: 30 iterations
Other Information
Hardware resource: Single CPU
Learning rate schedule: Constant
Learning rate: 0.01

```
YPred = classify(net,imdsValidation);
YValidation = imdsValidation.Labels;
accuracy = sum(YPred == YValidation)/numel(YValidation)
```

```
accuracy = 0.9420
```

Exercise-1-2

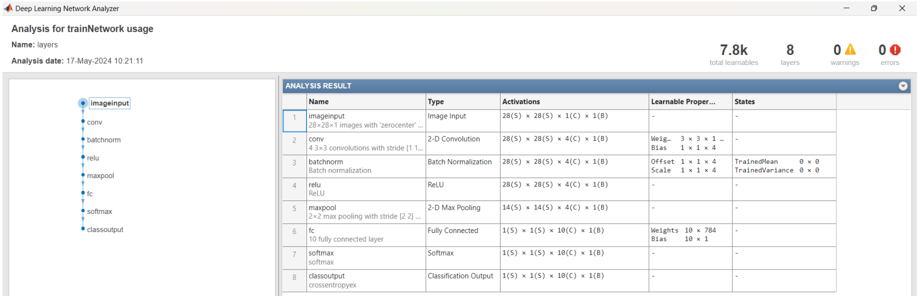
第一層卷積層 filter size = 4

```
layers = [
    imageInputLayer([28 28 1])

    convolution2dLayer([3 3],4,'Padding','same')
    batchNormalizationLayer
    reluLayer
    maxPooling2dLayer(2,'Stride',2)

    fullyConnectedLayer(10)
    softmaxLayer
    classificationLayer ];
```

```
net = trainNetwork(imdsTrain,layers,options);

YPred = classify(net,imdsValidation);
YValidation = imdsValidation.Labels;
accuracy = sum(YPred == YValidation)/numel(YValidation)
```

```
accuracy = 0.8892
```

```
analyzeNetwork(layers)
```



Analysis for trainNetwork usage

Name: layers

Analysis date: 17-May-2024 10:21:11

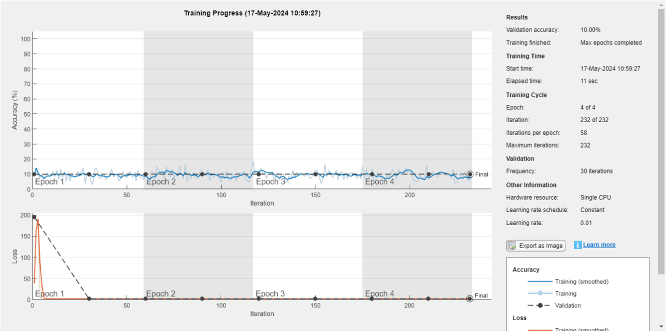
ANALYSIS RESULT

	Name	Type	Activations	Learnable Proper...	States
1	imageinput 28×28×1 images with 'zerocenter' ...	Image Input	28(S) × 28(S) × 1(C) × 1(B)	-	-
2	conv 4 3×3 convolutions with stride [1 1...	2-D Convolution	28(S) × 28(S) × 4(C) × 1(B)	Weig… 3 × 3 × 1 … Bias 1 × 1 × 4	-
3	batchnorm Batch normalization	Batch Normalization	28(S) × 28(S) × 4(C) × 1(B)	Offset 1 × 1 × 4 Scale 1 × 1 × 4	TrainedMean 0 × 0 TrainedVariance 0 × 0
4	relu ReLU	ReLU	28(S) × 28(S) × 4(C) × 1(B)	-	-
5	maxpool 2×2 max pooling with stride [2 2] ...	2-D Max Pooling	14(S) × 14(S) × 4(C) × 1(B)	-	-
6	fc 10 fully connected layer	Fully Connected	1(S) × 1(S) × 10(C) × 1(B)	Weights 10 × 784 Bias 10 × 1	-
7	softmax softmax	Softmax	1(S) × 1(S) × 10(C) × 1(B)	-	-
8	classoutput crossentropyex	Classification Output	1(S) × 1(S) × 10(C) × 1(B)	-	-

Exercise-2-1

去掉所有 batch normalization

```
layers = [
    imageInputLayer([28 28 1])

    convolution2dLayer([3 3],8,'Padding','same')
    reluLayer
    maxPooling2dLayer(2,'Stride',2)

    convolution2dLayer([3 3],16,'Padding','same')
    reluLayer
    maxPooling2dLayer(2,'Stride',2)

    fullyConnectedLayer(10)
    softmaxLayer
    classificationLayer ];

net = trainNetwork(imdsTrain,layers,options);
```



Training Progress (17-May-2024 10:59:27)

Results
Validation accuracy: 10.00%
Training finished: Max epochs completed
Training Time
Start time: 17-May-2024 10:59:27
Elapsed time: 11 sec
Training Cycle
Epoch: 4 of 4
Iteration: 232 of 232
Iterations per epoch: 58
Maximum iterations: 232
Validation
Frequency: 30 iterations
Other Information
Hardware resource: Single CPU
Learning rate schedule: Constant
Learning rate: 0.01

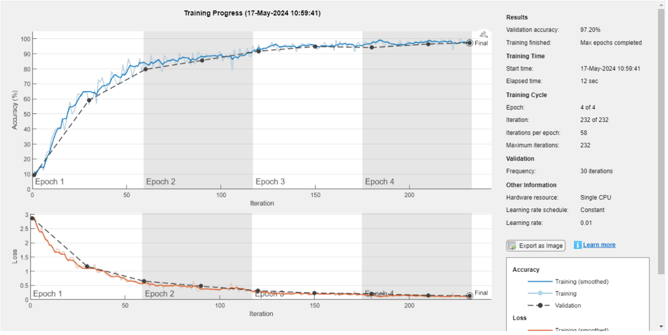
```
YPred = classify(net,imdsValidation);
YValidation = imdsValidation.Labels;
accuracy = sum(YPred == YValidation)/numel(YValidation)
```

```
accuracy = 0.1000
```

Exercise-2-2

僅去掉第一個 batch normalization

```
layers = [
    imageInputLayer([28 28 1])

    convolution2dLayer([3 3],8,'Padding','same')
    reluLayer
    maxPooling2dLayer(2,'Stride',2)

    convolution2dLayer([3 3],16,'Padding','same')
    batchNormalizationLayer
    reluLayer
    maxPooling2dLayer(2,'Stride',2)

    fullyConnectedLayer(10)
    softmaxLayer
    classificationLayer ];

net = trainNetwork(imdsTrain,layers,options);
```



Training Progress (17-May-2024 10:59:41)

Results
Validation accuracy: 97.20%
Training finished: Max epochs completed
Training Time
Start time: 17-May-2024 10:59:41
Elapsed time: 12 sec
Training Cycle
Epoch: 4 of 4
Iteration: 232 of 232
Iterations per epoch: 58
Maximum iterations: 232
Validation
Frequency: 30 iterations
Other Information
Hardware resource: Single CPU
Learning rate schedule: Constant
Learning rate: 0.01

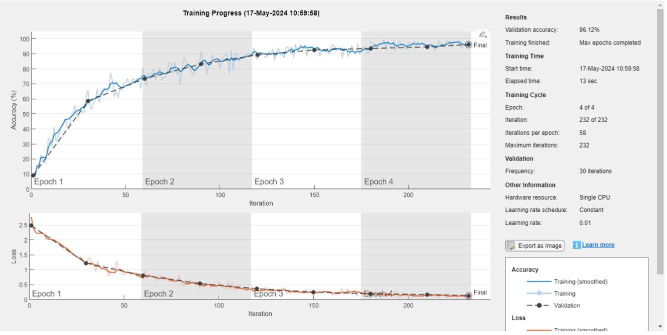
```
YPred = classify(net,imdsValidation);
YValidation = imdsValidation.Labels;
accuracy = sum(YPred == YValidation)/numel(YValidation)
```

```
accuracy = 0.9720
```

Exercise-2-3

僅去掉第二個 batch normalization

```
layers = [
    imageInputLayer([28 28 1])

    convolution2dLayer([3 3],8,'Padding','same')
    batchNormalizationLayer
    reluLayer
    maxPooling2dLayer(2,'Stride',2)

    convolution2dLayer([3 3],16,'Padding','same')
    reluLayer
    maxPooling2dLayer(2,'Stride',2)

    fullyConnectedLayer(10)
    softmaxLayer
    classificationLayer ];

net = trainNetwork(imdsTrain,layers,options);
```



Results
Validation accuracy: 96.12%
Training finished: Max epochs completed
Training Time
Start time: 17-May-2024 10:59:58
Elapsed time: 13 sec
Training Cycle
Epoch: 4 of 4
Iteration: 232 of 232
Iterations per epoch: 58
Maximum iterations: 232
Validation
Frequency: 30 iterations
Other Information
Hardware resource: Single CPU
Learning rate schedule: Constant
Learning rate: 0.01

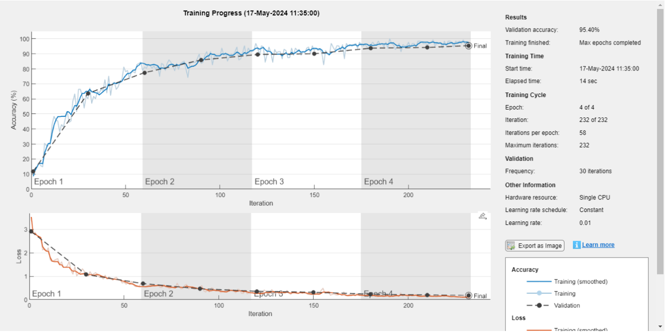
```
YPred = classify(net,imdsValidation);
YValidation = imdsValidation.Labels;
accuracy = sum(YPred == YValidation)/numel(YValidation)
```

```
accuracy = 0.9612
```

Exercise-3-1

去掉所有 relu activation

```
layers = [
    imageInputLayer([28 28 1])
    convolution2dLayer([3 3],8,'Padding','same')
    batchNormalizationLayer
    maxPooling2dLayer(2,'Stride',2)

    convolution2dLayer([3 3],16,'Padding','same')
    batchNormalizationLayer
    maxPooling2dLayer(2,'Stride',2)

    fullyConnectedLayer(10)
    softmaxLayer
    classificationLayer ];

net = trainNetwork(imdsTrain,layers,options);
```



Results
Validation accuracy: 95.40%
Training finished: Max epochs completed
Training Time
Start time: 17-May-2024 11:35:00
Elapsed time: 14 sec
Training Cycle
Epoch: 4 of 4
Iteration: 232 of 232
Iterations per epoch: 58
Maximum iterations: 232
Validation
Frequency: 30 iterations
Other Information
Hardware resource: Single CPU
Learning rate schedule: Constant
Learning rate: 0.01

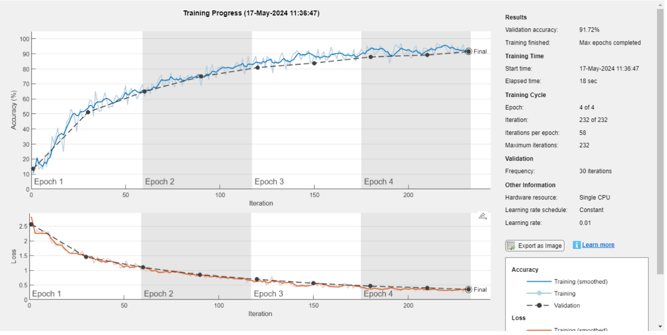
```
YPred = classify(net,imdsValidation);
YValidation = imdsValidation.Labels;
accuracy = sum(YPred == YValidation)/numel(YValidation)
```

```
accuracy = 0.9540
```

Exercise-3-2

activation function 改成 tanh

```
layers = [
    imageInputLayer([28 28 1])

    convolution2dLayer([3 3],8,'Padding','same')
    batchNormalizationLayer
    tanhLayer
    maxPooling2dLayer(2,'Stride',2)

    convolution2dLayer([3 3],16,'Padding','same')
    batchNormalizationLayer
    tanhLayer
    maxPooling2dLayer(2,'Stride',2)

    fullyConnectedLayer(10)
    softmaxLayer
    classificationLayer ];

net = trainNetwork(imdsTrain,layers,options);
```



Results
Validation accuracy: 91.72%
Training finished: Max epochs completed
Training Time
Start time: 17-May-2024 11:36:47
Elapsed time: 18 sec
Training Cycle
Epoch: 4 of 4
Iteration: 232 of 232
Iterations per epoch: 58
Maximum iterations: 232
Validation
Frequency: 30 iterations
Other Information
Hardware resource: Single CPU
Learning rate schedule: Constant
Learning rate: 0.01

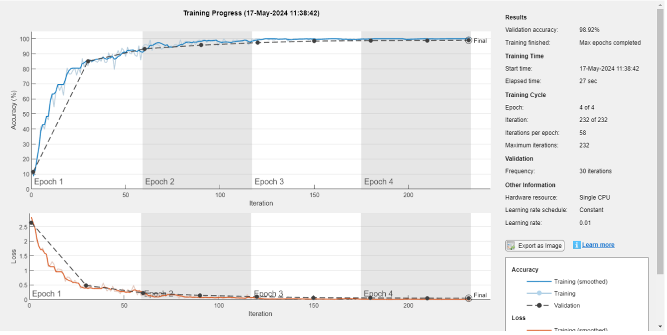
```
YPred = classify(net,imdsValidation);
YValidation = imdsValidation.Labels;
accuracy = sum(YPred == YValidation)/numel(YValidation)
```

```
accuracy = 0.9172
```

Exercise-4-1

去掉所有 pooling

```
```



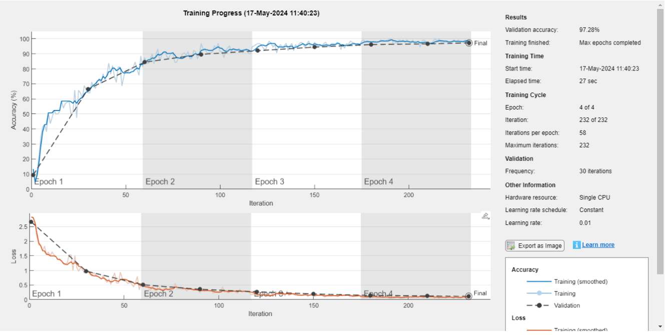
Results
Validation accuracy: 98.92%
Training finished: Max epochs completed
Training Time
Start time: 17-May-2024 11:38:42
Elapsed time: 27 sec
Training Cycle
Epoch: 4 of 4
Iteration: 232 of 232
Iterations per epoch: 58
Maximum iterations: 232
Validation
Frequency: 30 iterations
Other Information
Hardware resource: Single CPU
Learning rate schedule: Constant
Learning rate: 0.01

```
accuracy = 0.9892
```

Exercise-4-2

去掉第一個 pooling

```
layers = [
    imageInputLayer([28 28 1])

    convolution2dLayer([3 3],8,'Padding','same')
    batchNormalizationLayer
    reluLayer
    % maxPooling2dLayer(2,'Stride',2)

    convolution2dLayer([3 3],16,'Padding','same')
    batchNormalizationLayer
    reluLayer
    maxPooling2dLayer(2,'Stride',2)

    fullyConnectedLayer(10)
    softmaxLayer
    classificationLayer ];

net = trainNetwork(imdsTrain,layers,options);
```



Results
Validation accuracy: 97.28%
Training finished: Max epochs completed
Training Time
Start time: 17-May-2024 11:40:23
Elapsed time: 27 sec
Training Cycle
Epoch: 4 of 4
Iteration: 232 of 232
Iterations per epoch: 58
Maximum iterations: 232
Validation
Frequency: 30 iterations
Other Information
Hardware resource: Single CPU
Learning rate schedule: Constant
Learning rate: 0.01

Accuracy
Training (smoothed)
Training
Validation
Loss
Training (smoothed)

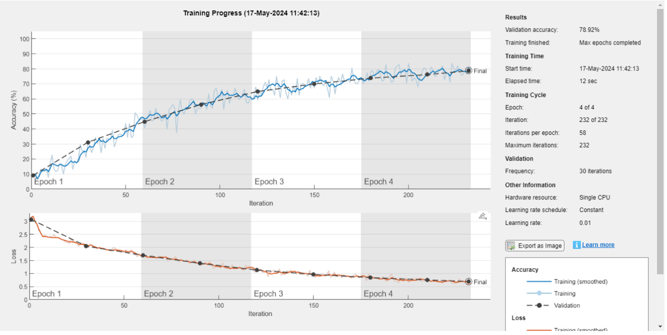
```
YPred = classify(net,imdsValidation);
YValidation = imdsValidation.Labels;
accuracy = sum(YPred == YValidation)/numel(YValidation)
```

```
accuracy = 0.9728
```

Exercise-4-3

將所有 pooling 的 stride 改成 4

```
layers = [
    imageInputLayer([28 28 1])

    convolution2dLayer([3 3],8,'Padding','same')
    batchNormalizationLayer
    reluLayer
    maxPooling2dLayer(2,'Stride',4)

    convolution2dLayer([3 3],16,'Padding','same')
    batchNormalizationLayer
    reluLayer
    maxPooling2dLayer(2,'Stride',4)

    fullyConnectedLayer(10)
    softmaxLayer
    classificationLayer ];

net = trainNetwork(imdsTrain,layers,options);
```


Training Progress (17-May-2024 11:42:13)

```
Results
Validation accuracy:      78.92%
Training finished:        Max epochs completed
Training Time
Start time:               17-May-2024 11:42:13
Elapsed time:             12 sec
Training Cycle
Epoch:                    4 of 4
Iteration:                232 of 232
Iterations per epoch:     58
Maximum iterations:       232
Validation
Frequency:                30 iterations
Other Information
Hardware resource:        Single CPU
Learning rate schedule:   Constant
Learning rate:            0.01
```

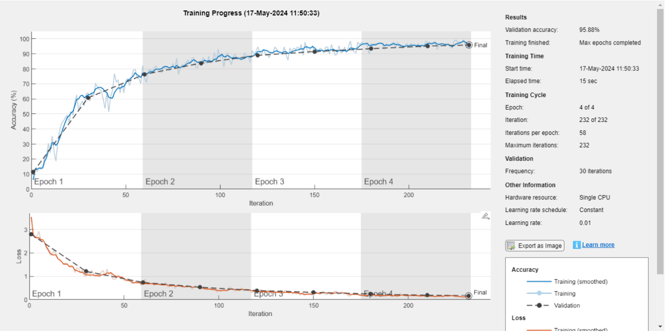
```
YPred = classify(net,imdsValidation);
YValidation = imdsValidation.Labels;
accuracy = sum(YPred == YValidation)/numel(YValidation)
```

```
accuracy = 0.7892
```

Exercise-5

辨識自己的手寫圖

```
layers = [
    imageInputLayer([28 28 1])

    convolution2dLayer([3 3],8,'Padding','same')
    batchNormalizationLayer
    reluLayer
    maxPooling2dLayer(2,'Stride',2)

    convolution2dLayer([3 3],16,'Padding','same')
    batchNormalizationLayer
    reluLayer
    maxPooling2dLayer(2,'Stride',2)

    fullyConnectedLayer(10)
    softmaxLayer
    classificationLayer ];

net = trainNetwork(imdsTrain,layers,options);
```

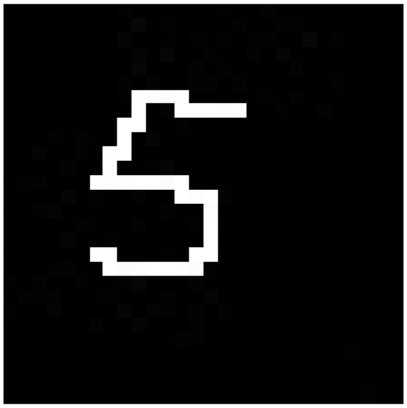

Training Progress (17-May-2024 11:50:33)

```
Results
Validation accuracy:      95.88%
Training finished:        Max epochs completed
Training Time
Start time:               17-May-2024 11:50:33
Elapsed time:             15 sec
Training Cycle
Epoch:                    4 of 4
Iteration:                232 of 232
Iterations per epoch:     58
Maximum iterations:       232
Validation
Frequency:                30 iterations
Other Information
Hardware resource:        Single CPU
Learning rate schedule:   Constant
Learning rate:            0.01
```

```
YPred = classify(net,imdsValidation);
YValidation = imdsValidation.Labels;
accuracy = sum(YPred == YValidation) / numel(YValidation)
```

```
accuracy = 0.9588
```

```
clf('reset')

I = imread('./images/hand_write_5.png');
Ig = reshape(rgb2gray(I), [28, 28, 1]);
imshow(Ig)
truesize([200 200])
```



```
classify(net, Ig)
```

```
ans = categorical
     5
```