# Part A: Deep convolutional neural network

## Introduction

The first part of the assignment serves as an introductory exercise for Convolutional Neural Networks (CNN). In this exercise, a CNN is constructed to recognise MNIST digits. The CNN has the following configuration:

|  |  |  |
| --- | --- | --- |
| **Layer** | **Type** | **Dimensions** |
| 1 | Input Layer | 28 x 28 neurons |
| 2 | Convolutional Layer, C1 | 15 9x9 filters |
| 3 | Max Pooling, S1 | 2x2 window |
| 4 | Convolutional Layer, C2 | 20 5x5 filters |
| 5 | Max Pooling, S2 | 2x2 window |
| 6 | Fully Connected Layer | 100 neurons |
| 7 | Softmax Output Layer | 10 |

## Part 1

The CNN above is trained using the ReLu activation function, with mini-batch size of 128, learning rate of 0.05 and decay parameter of 10-4. Figure 1 shows the training statistics. Subsequently, the feature maps at each pooling and convolutional layers are visualized.

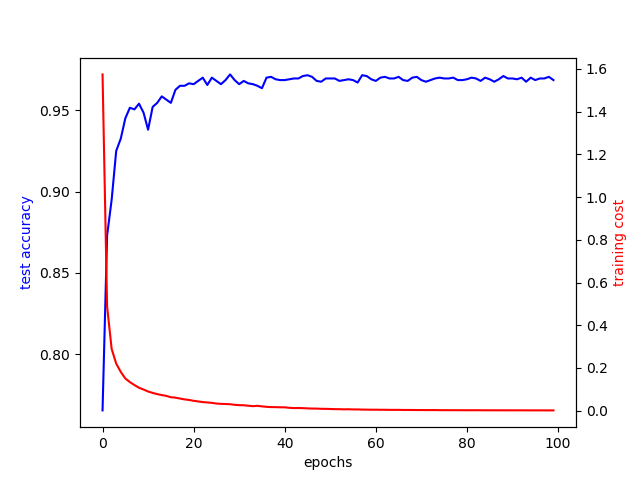


Figure Training cost and test accuracy per epoch

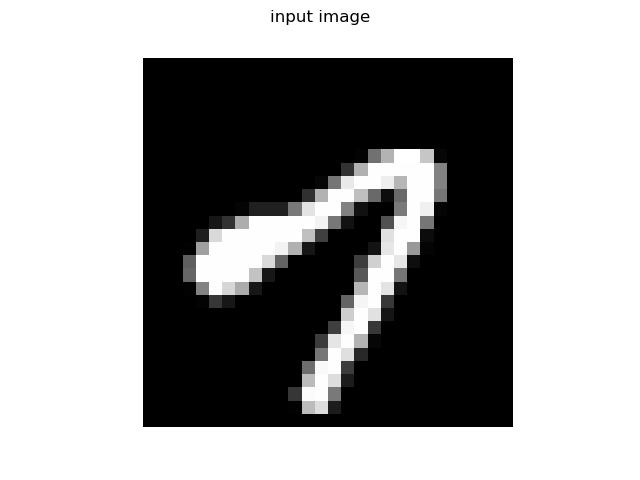


Figure Test input image 1

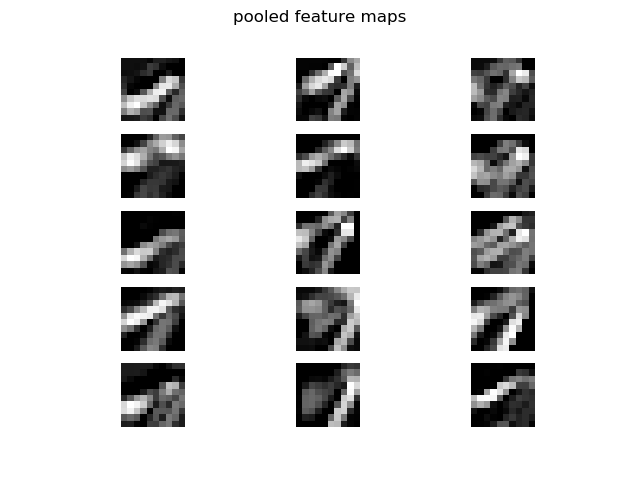
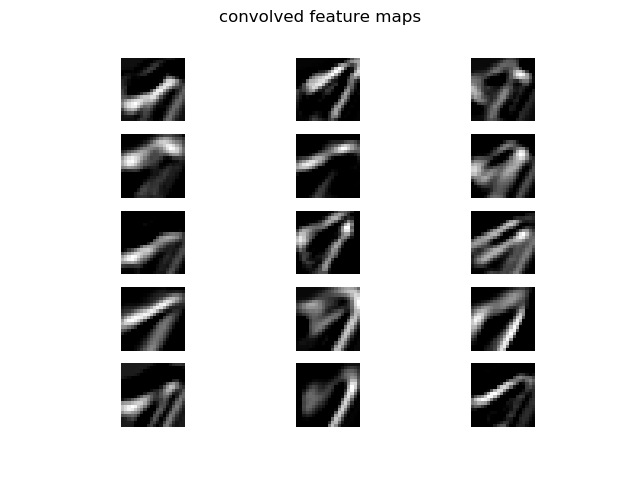


Figure Feature maps from C1 and S1

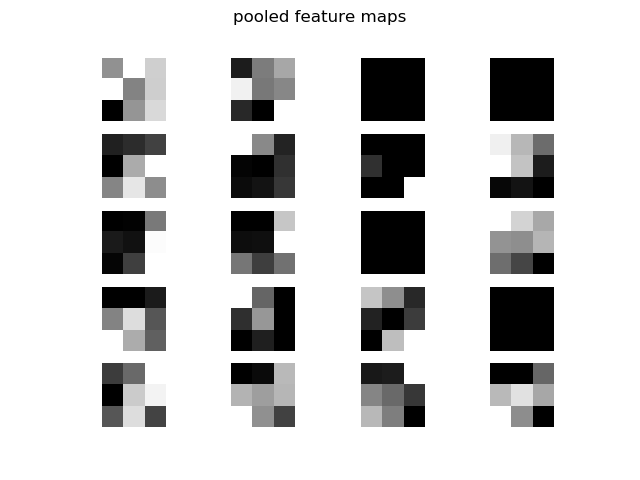
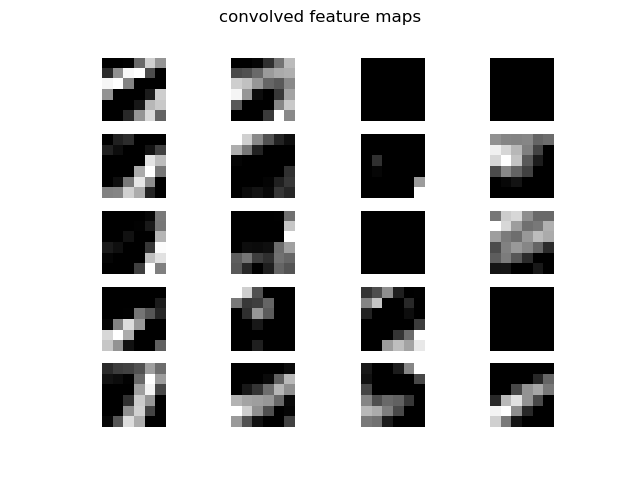


Figure Feature maps from C2 and S2

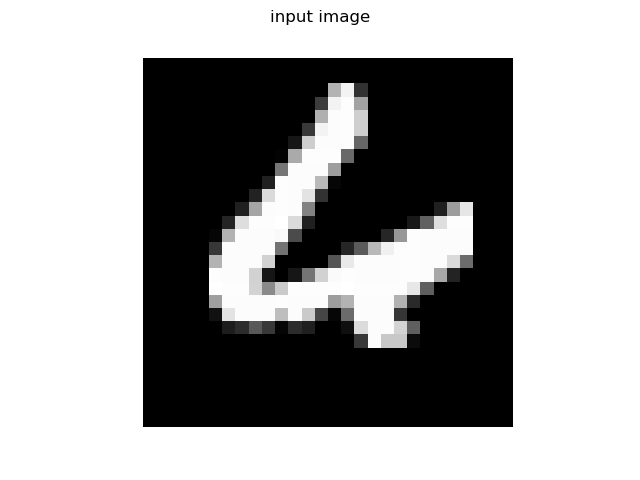


Figure Test input image 2

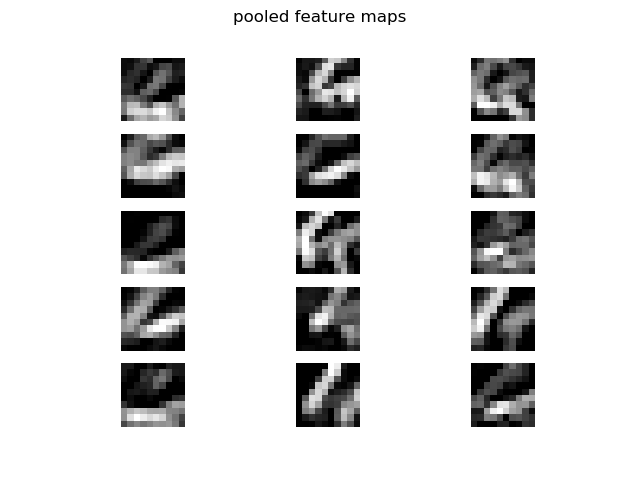
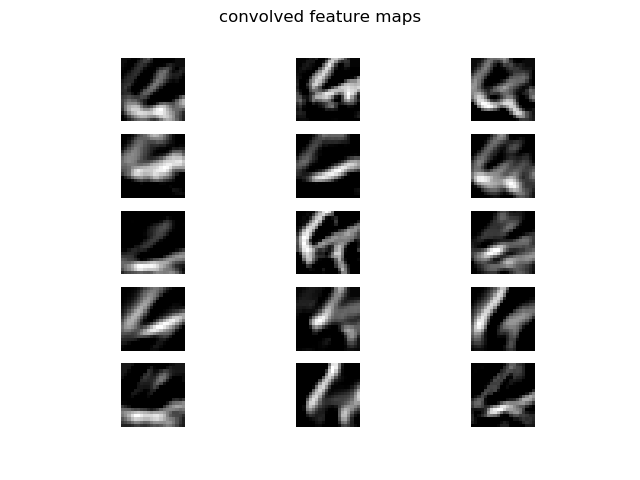


Figure Feature maps from C1 and S1

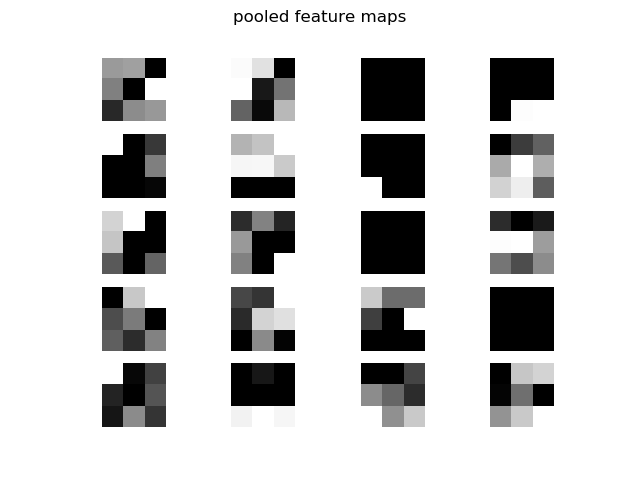
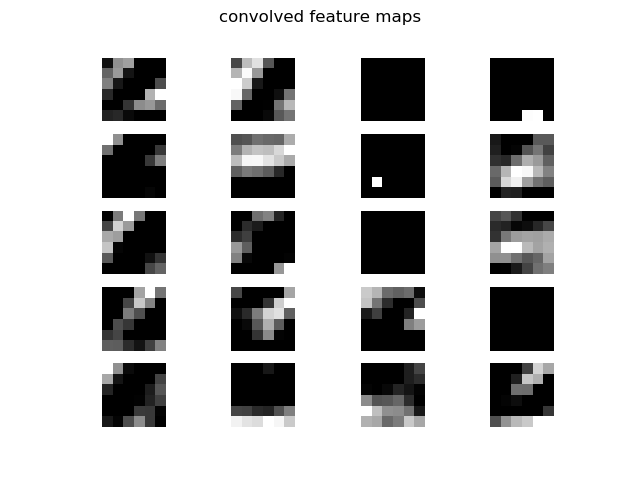


Figure Feature maps from C2 and S2

## Part 2

The exercise in Part 1 is repeated with the addition of a momentum term set to 0.1.

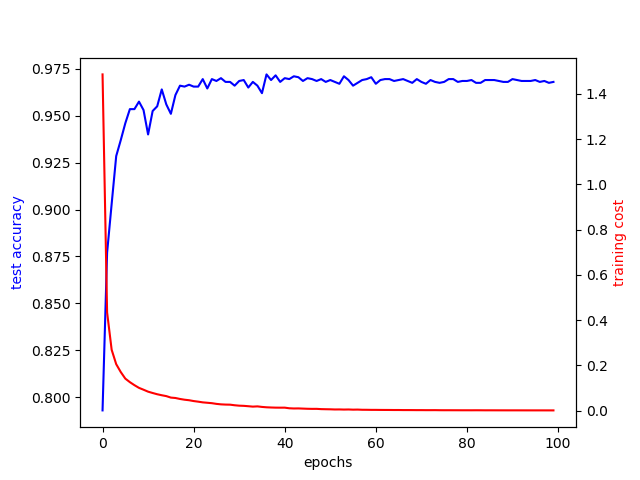


Figure Training cost and test accuracy for Part 2

Compared to the results in Part 1, the test accuracy curve fluctuated to a lesser extent, albeit slightly. This is to be expected as the momentum factor helps the network to quickly search for a local minimum when the cost function reaches a ‘ravine’.

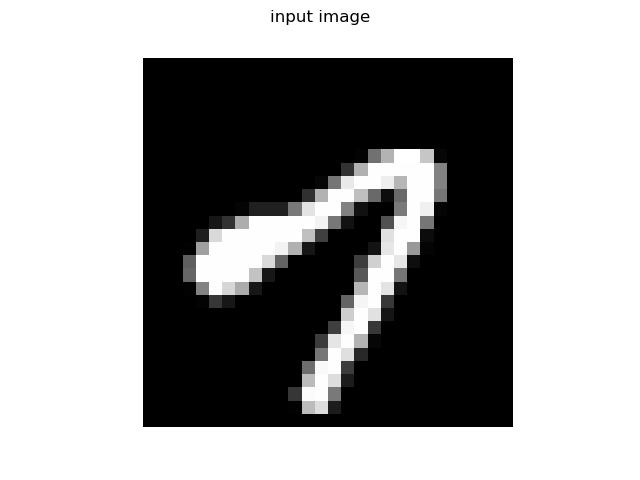


Figure Test input image 1

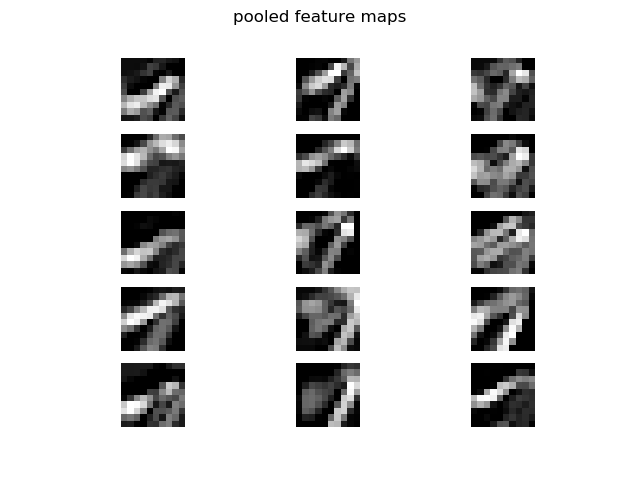
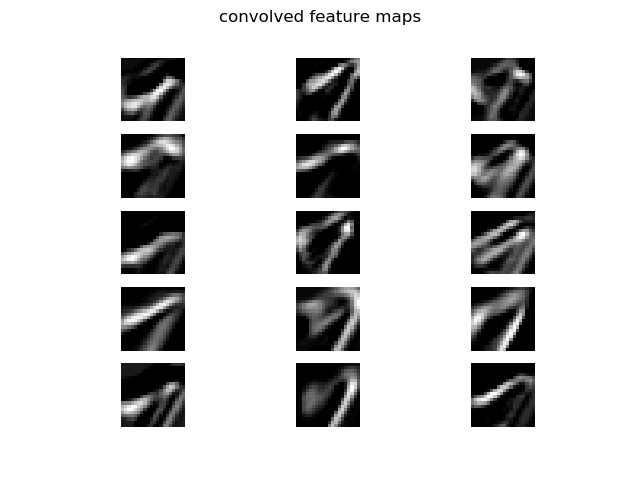


Figure Feature maps from C1 and S1

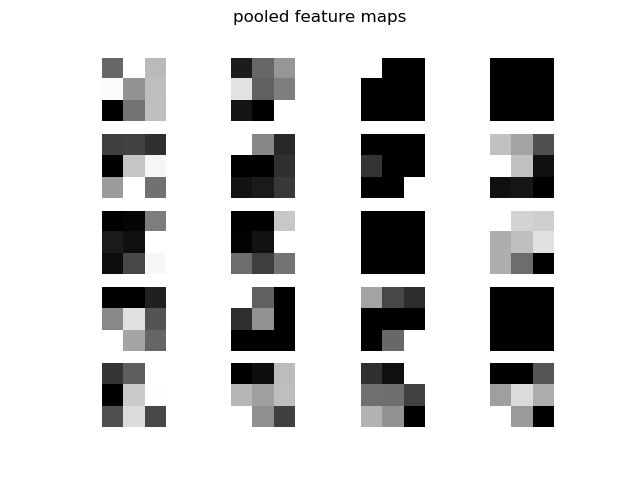
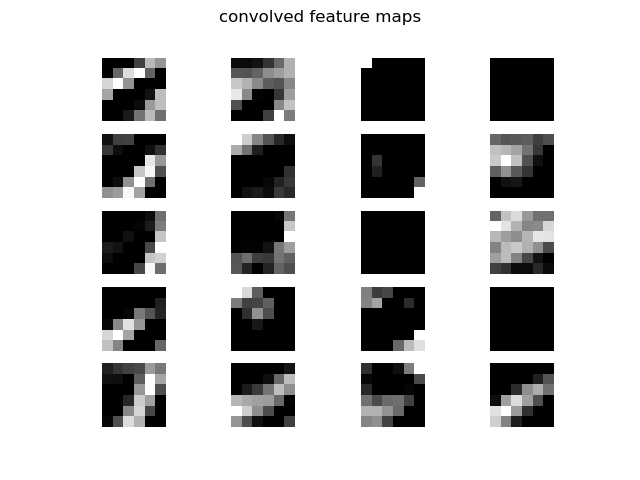


Figure Feature maps from C2 and S2

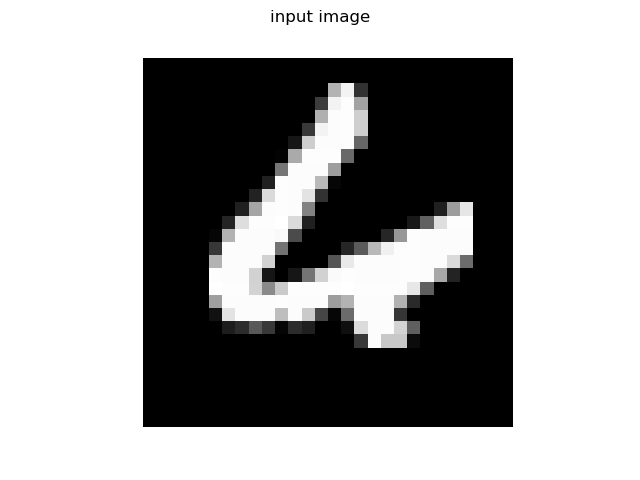


Figure Test input image 2

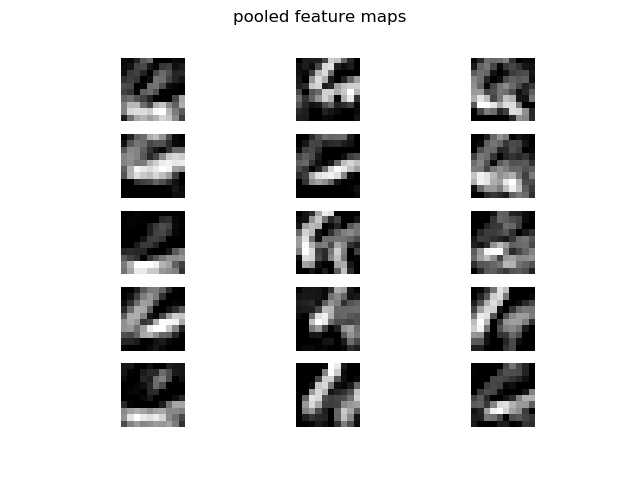
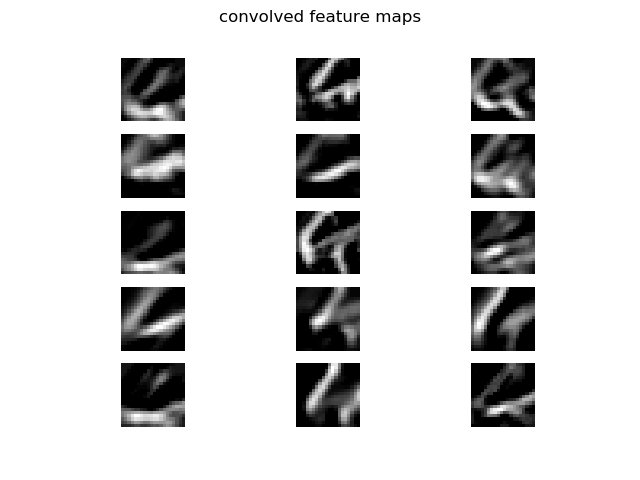


Figure Feature maps from C1 and S1

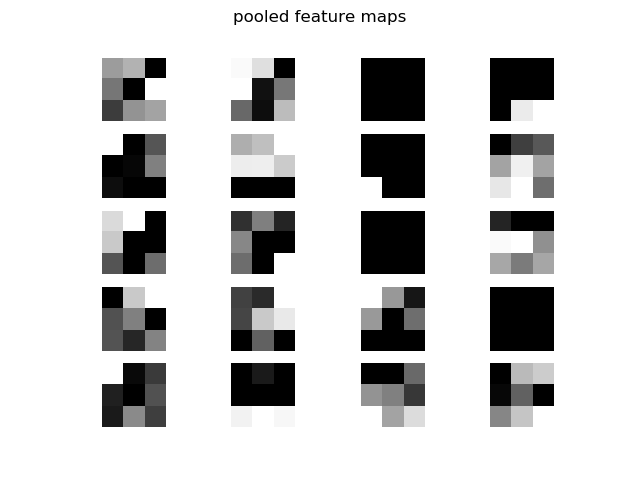
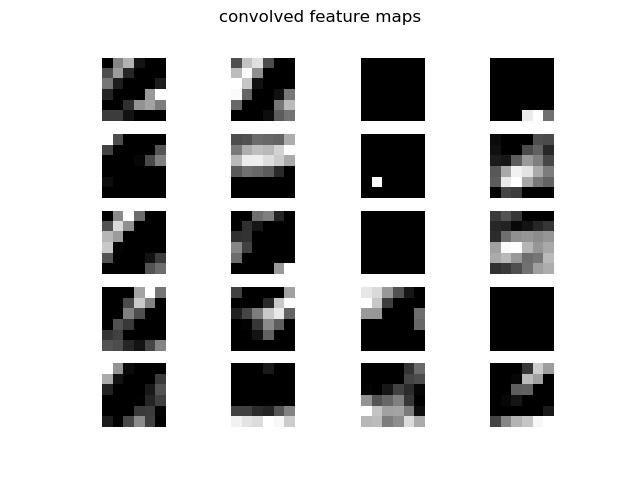


Figure Feature maps from C2 and S2

## Part 3

Similarly, Part 1 is repeated with extra constraints to realise the RMSProp algorithm. The parameters are: 𝛼 = 0.001, 𝛽 = 1𝑒−4, 𝜌 = 0.9, and 𝜖 = 10−6.

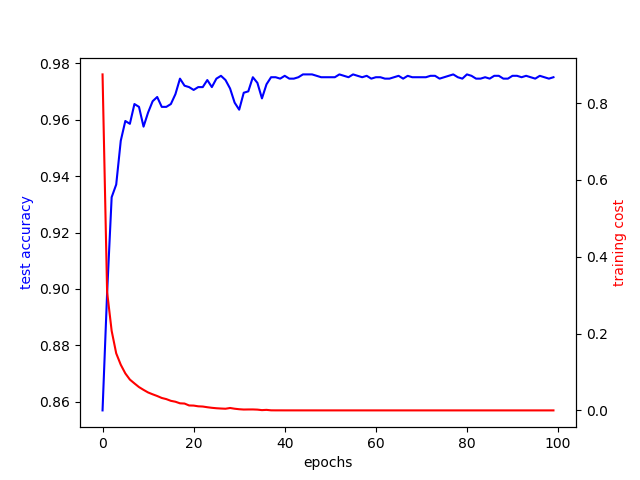


Figure Training statistics for Part 3

The CNN converged slightly faster when using RMSProp and the fluctuations in the test accuracy curve is less pronounced.

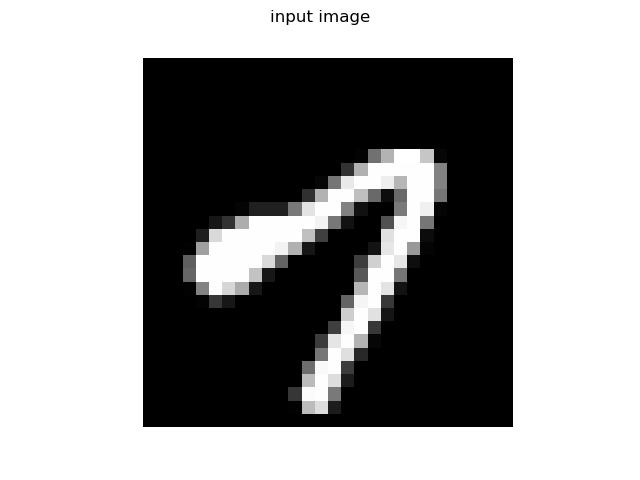


Figure Test image input 1

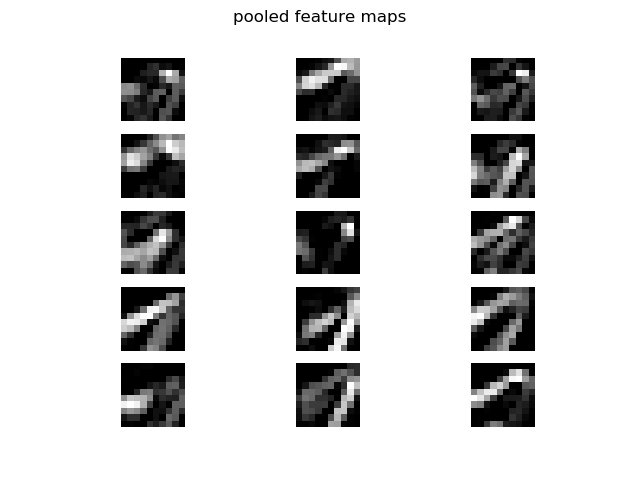
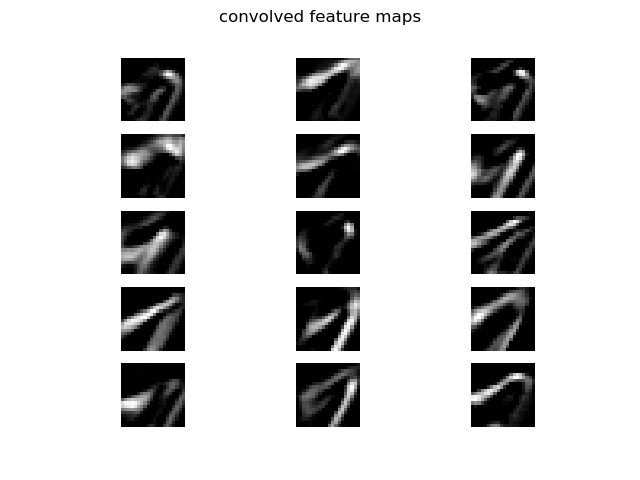


Figure Feature maps from C1 and S1

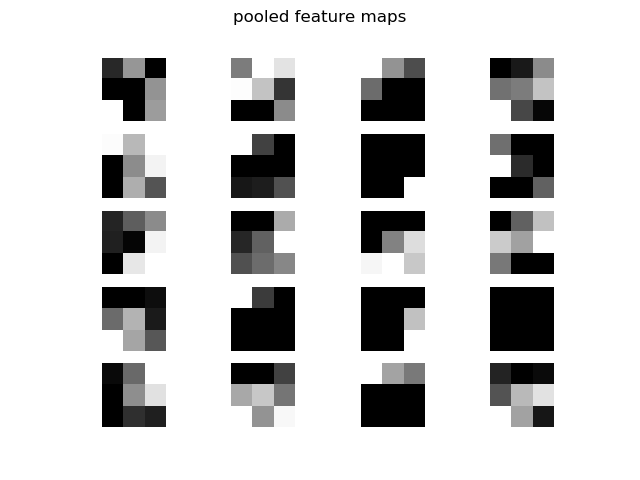
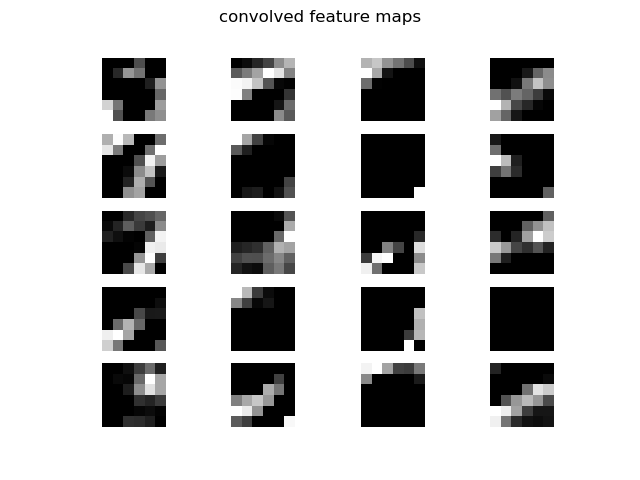


Figure Feature maps from C2 and S2

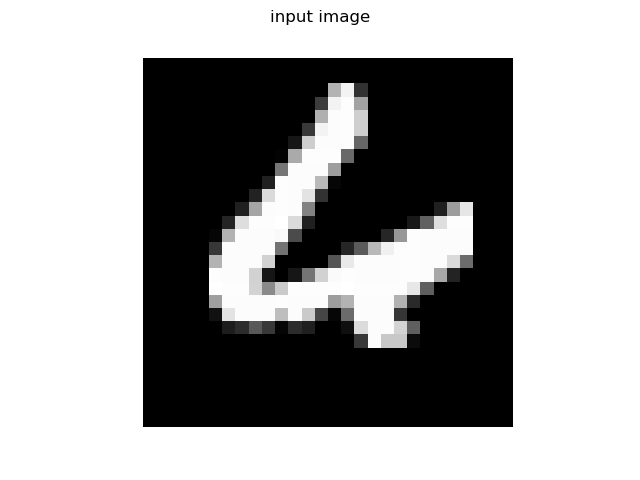


Figure Test image input 2

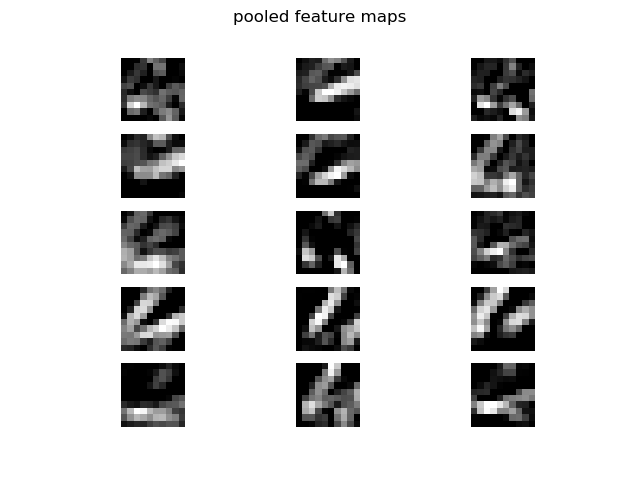
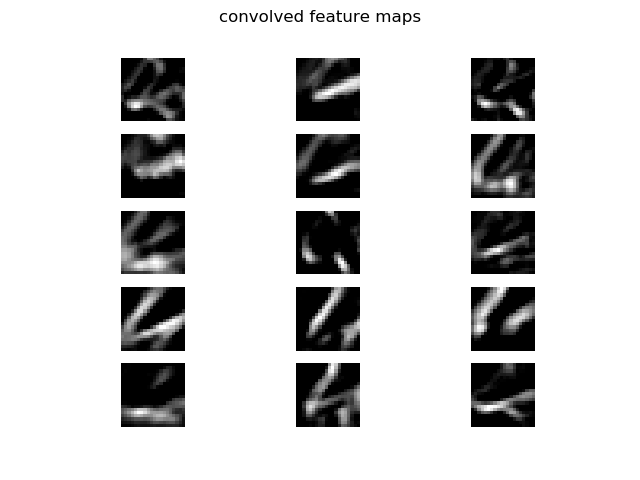


Figure Feature maps from C1 and S1

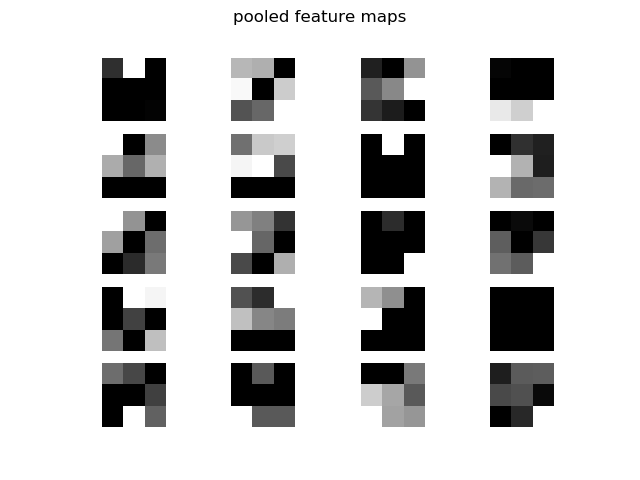
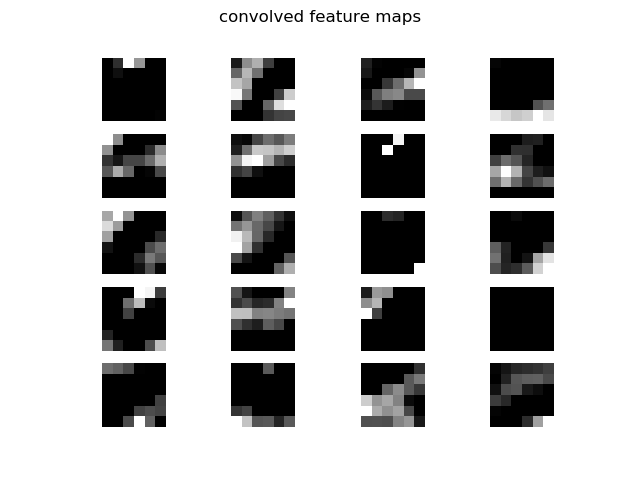


Figure Feature maps from C2 and S2

# Part B: Autoencoders