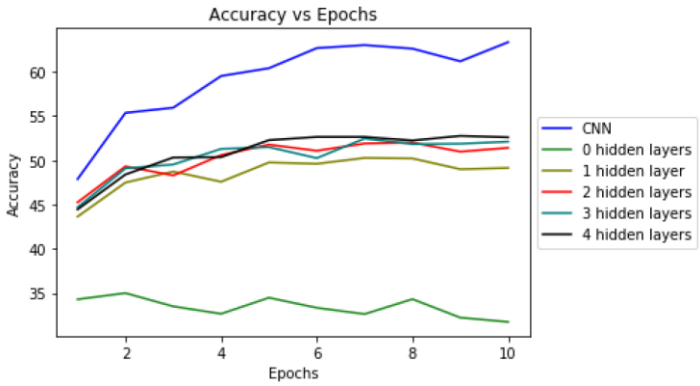
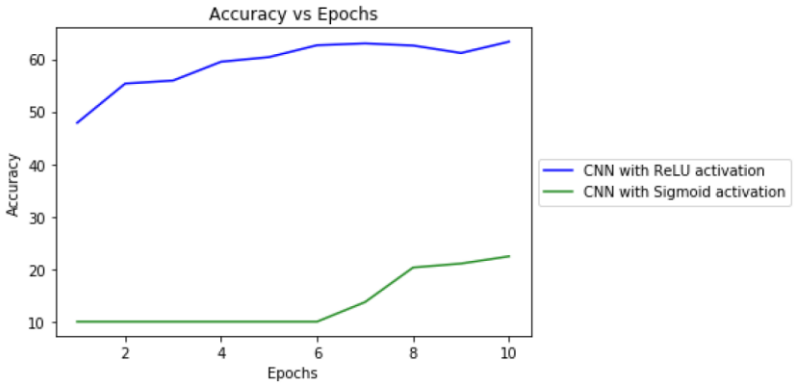
COMP 5900V
Advanced Machine Learning
Winter 2021
Solution to Assignment 1

Q1



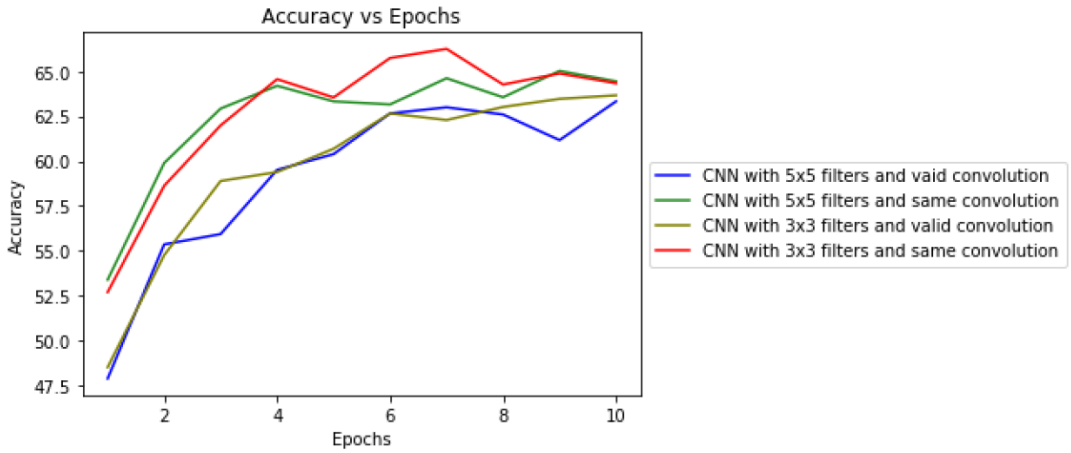
A simple Neural network with 0 hidden layer may not have enough flexibility to learn different classes and therefore performs poorly. Increasing the number of hidden layers improves this situation but on the other hand makes the network prone to overfitting as the number of parameters increases. CNN has the best performance as it has much smaller number of parameters because it shares parameters through convolution.

Q2



The network with Sigmoid activations converges much slower. Relu is defined as h=max(0,a) where a=Wx+b. Relu often helps with avoiding the gradient to vanish. When a>0 the gradient has a constant value. In contrast, the gradient of sigmoids becomes increasingly small as the absolute value of a increases. The constant gradient of ReLUs often results in faster learning.

Q3



Same convolution performs better in this case. Note that using valid convolution, the dimension is decreased more aggressively (compare in_features in fc1 for same versus valid) which may explain the inferior performance. Adding more cnn layers (valid conv) will make this even worst. Regarding the filter size, there is no significant difference between performance with 3x3 or 5x5 filters.

Valid-conv with 5x5 filters

```
Net(
  (conv1): Conv2d(3, 6, kernel_size=(5, 5), stride=(1, 1))
  (pool): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
  (conv2): Conv2d(6, 16, kernel_size=(5, 5), stride=(1, 1))
  (fc1): Linear(in_features=400, out_features=120, bias=True)
  (fc2): Linear(in_features=120, out_features=120, bias=True)
  (fc3): Linear(in_features=120, out_features=10, bias=True)
)
```

Same-conv with 5x5 filters

```
Net(
  (conv1): Conv2d(3, 6, kernel_size=(5, 5), stride=(1, 1), padding=(2, 2))
  (pool): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
  (conv2): Conv2d(6, 16, kernel_size=(5, 5), stride=(1, 1), padding=(2, 2))
  (fc1): Linear(in_features=1024, out_features=120, bias=True)
  (fc2): Linear(in_features=120, out_features=120, bias=True)
  (fc3): Linear(in_features=120, out_features=10, bias=True)
)
```

Valid-conv with 3x3 filters

```
Net(
  (conv1): Conv2d(3, 6, kernel_size=(3, 3), stride=(1, 1))
  (pool): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
  (conv2): Conv2d(6, 16, kernel_size=(3, 3), stride=(1, 1))
  (fc1): Linear(in_features=576, out_features=120, bias=True)
  (fc2): Linear(in_features=120, out_features=120, bias=True)
  (fc3): Linear(in_features=120, out_features=10, bias=True)
)
```

Same-conv with 3x3 filters

```
Net(
  (conv1): Conv2d(3, 6, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
  (pool): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
  (conv2): Conv2d(6, 16, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
  (fc1): Linear(in_features=1024, out_features=120, bias=True)
  (fc2): Linear(in_features=120, out_features=120, bias=True)
  (fc3): Linear(in_features=120, out_features=10, bias=True)
)
```