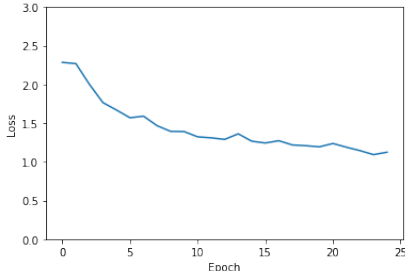
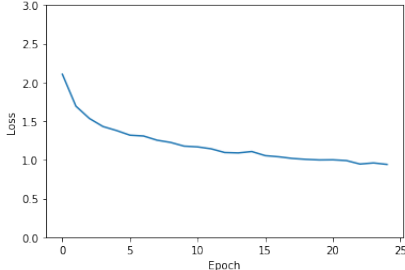
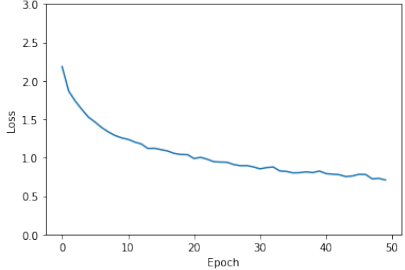
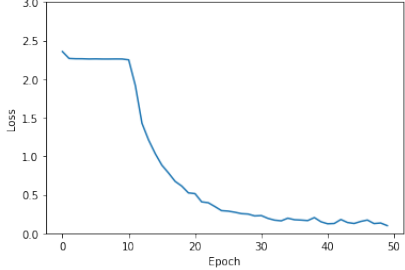
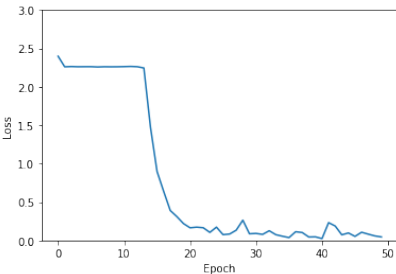
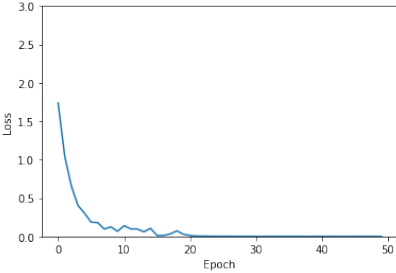
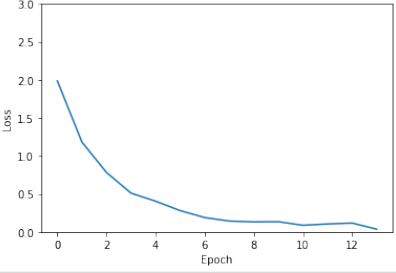
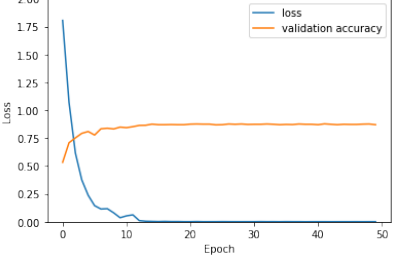
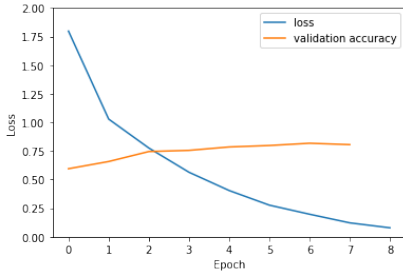
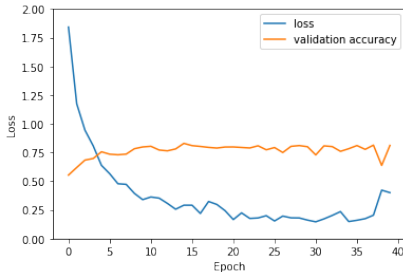


| State_dict | Loss function | Accuracy | Further info |
|---|---|----------|---|
| <pre> <bound method Module.state_dict of CNN((conv1): Conv2d(1, 6, kernel_size=(9, 9), stride=(2, 2)) (conv2): Conv2d(6, 18, kernel_size=(7, 7), stride=(2, 2)) (conv2_drop): Dropout2d(p=0.5, inplace=False) (pool): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False) (fc1): Linear(in_features=648, out_features=100, bias=True) (fc2): Linear(in_features=100, out_features=10, bias=True) (fc3): Linear(in_features=100, out_features=10, bias=True))> </pre> |  | | <p>Criterion = CrossEntropyLoss() Optimizer = optim.SGD(lr=0.001, momentum=0.9)</p> <p>43%</p> |
| <pre> <bound method Module.state_dict of CNN((conv1): Conv2d(1, 6, kernel_size=(7, 7), stride=(2, 2)) (conv2): Conv2d(6, 18, kernel_size=(5, 5), stride=(2, 2)) (conv2_drop): Dropout2d(p=0.5, inplace=False) (pool): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False) (fc1): Linear(in_features=648, out_features=100, bias=True) (fc2): Linear(in_features=100, out_features=10, bias=True) (fc3): Linear(in_features=100, out_features=10, bias=True))> </pre> |  | | <p>Criterion = CrossEntropyLoss() Optimizer = optim.SGD(lr=0.001, momentum=0.9)</p> <p>50%</p> |
| <pre> <bound method Module.state_dict of CNN((conv1): Conv2d(1, 6, kernel_size=(5, 5), stride=(2, 2)) (conv2): Conv2d(6, 18, kernel_size=(3, 3), stride=(2, 2)) (conv2_drop): Dropout2d(p=0.5, inplace=False) (pool): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False) (fc1): Linear(in_features=882, out_features=100, bias=True) (fc2): Linear(in_features=100, out_features=10, bias=True) (fc3): Linear(in_features=100, out_features=10, bias=True))> </pre> |  | | <p>Criterion = CrossEntropyLoss() Optimizer = optim.SGD(lr=0.001, momentum=0.9)</p> <p>45%</p> |
| <pre> <bound method Module.state_dict of CNN((conv1): Conv2d(1, 6, kernel_size=(5, 5), stride=(1, 1)) (conv2): Conv2d(6, 18, kernel_size=(3, 3), stride=(1, 1)) (conv2_drop): Dropout2d(p=0.5, inplace=False) (pool): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False) (fc1): Linear(in_features=15138, out_features=100, bias=True) (fc2): Linear(in_features=100, out_features=10, bias=True) (fc3): Linear(in_features=100, out_features=10, bias=True))> </pre> |  | | <p>Criterion = CrossEntropyLoss() Optimizer = optim.SGD(lr=0.001, momentum=0.9)</p> <p>44%</p> |

| | | | | |
|--|--|---|-----|--|
| | <pre> <bound method Module.state_dict of CNN((conv1): Conv2d(1, 6, kernel_size=(7, 7), stride=(1, 1)) (conv2): Conv2d(6, 18, kernel_size=(5, 5), stride=(1, 1)) (conv3): Conv2d(18, 36, kernel_size=(3, 3), stride=(1, 1)) (conv2_drop): Dropout2d(p=0.5, inplace=False) (pool): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False) (fc1): Linear(in_features=14112, out_features=100, bias=True) (fc2): Linear(in_features=100, out_features=10, bias=True) (fc3): Linear(in_features=100, out_features=10, bias=True))> </pre> |  | 54% | <p>Criterion = CrossEntropyLoss() Optimizer = optim.SGD(lr=0.001, momentum=0.9)</p> |
| | <pre> <bound method Module.state_dict of CNN((conv1): Conv2d(1, 6, kernel_size=(7, 7), stride=(1, 1)) (conv2): Conv2d(6, 18, kernel_size=(5, 5), stride=(1, 1)) (conv3): Conv2d(18, 36, kernel_size=(3, 3), stride=(1, 1)) (conv2_drop): Dropout2d(p=0.5, inplace=False) (pool): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False) (fc1): Linear(in_features=6084, out_features=100, bias=True) (fc2): Linear(in_features=100, out_features=10, bias=True) (fc3): Linear(in_features=100, out_features=10, bias=True))> </pre> |  | 61% | <p>Criterion = CrossEntropyLoss() Optimizer = optim.SGD(lr=0.001, momentum=0.9)</p> |
| | <pre> <bound method Module.state_dict of CNN((conv1): Conv2d(1, 6, kernel_size=(7, 7), stride=(1, 1)) (conv2): Conv2d(6, 18, kernel_size=(5, 5), stride=(1, 1)) (conv3): Conv2d(18, 36, kernel_size=(3, 3), stride=(1, 1)) (conv2_drop): Dropout2d(p=0.5, inplace=False) (pool): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False) (fc1): Linear(in_features=6084, out_features=100, bias=True) (fc2): Linear(in_features=100, out_features=10, bias=True) (fc3): Linear(in_features=100, out_features=10, bias=True))> </pre> |  | 56% | <p>Criterion = CrossEntropyLoss() Optimizer = optim.SGD(lr=0.001, momentum=0.9)</p> |
| | <pre> <bound method Module.state_dict of CNN((conv1): Conv2d(1, 6, kernel_size=(7, 7), stride=(1, 1)) (conv2): Conv2d(6, 18, kernel_size=(5, 5), stride=(1, 1)) (conv2_drop): Dropout2d(p=0.5, inplace=False) (pool): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False) (fc1): Linear(in_features=14112, out_features=1000, bias=True) (fc2): Linear(in_features=1000, out_features=100, bias=True) (fc3): Linear(in_features=100, out_features=10, bias=True))> </pre> |  | 62% | <p>Criterion = CrossEntropyLoss() Optimizer = optim.SGD(lr=0.001, momentum=0.9)</p> |

| <pre><bound method Module.state_dict of CNN((conv1): Conv2d(1, 6, kernel_size=(5, 5), stride=(1, 1)) (conv2): Conv2d(6, 18, kernel_size=(3, 3), stride=(1, 1)) (conv2_drop): Dropout2d(p=0.5, inplace=False) (pool): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False) (fc1): Linear(in_features=15138, out_features=1000, bias=True) (fc2): Linear(in_features=1000, out_features=100, bias=True) (fc3): Linear(in_features=100, out_features=10, bias=True))></pre> |  <table><caption>Approximate data for Epochs 0-8</caption><tr><th>Epoch</th><th>Loss</th><th>Validation Accuracy</th></tr><tr><td>0</td><td>1.80</td><td>0.60</td></tr><tr><td>1</td><td>1.05</td><td>0.65</td></tr><tr><td>2</td><td>0.75</td><td>0.75</td></tr><tr><td>3</td><td>0.55</td><td>0.78</td></tr><tr><td>4</td><td>0.40</td><td>0.80</td></tr><tr><td>5</td><td>0.28</td><td>0.82</td></tr><tr><td>6</td><td>0.20</td><td>0.84</td></tr><tr><td>7</td><td>0.15</td><td>0.85</td></tr><tr><td>8</td><td>0.12</td><td>0.85</td></tr></table> | Epoch | Loss | Validation Accuracy | 0 | 1.80 | 0.60 | 1 | 1.05 | 0.65 | 2 | 0.75 | 0.75 | 3 | 0.55 | 0.78 | 4 | 0.40 | 0.80 | 5 | 0.28 | 0.82 | 6 | 0.20 | 0.84 | 7 | 0.15 | 0.85 | 8 | 0.12 | 0.85 | Criterion = CrossEntropyLoss() 63% Optimizer = optim.Adam(lr=0.001) |
|--|--|---------------------|------|---------------------|---|------|------|---|------|------|----|------|------|----|------|------|----|------|------|----|------|------|----|------|------|----|------|------|----|------|------|--|
| Epoch | Loss | Validation Accuracy | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 1.80 | 0.60 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 1.05 | 0.65 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2 | 0.75 | 0.75 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 3 | 0.55 | 0.78 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 4 | 0.40 | 0.80 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 5 | 0.28 | 0.82 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 6 | 0.20 | 0.84 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 7 | 0.15 | 0.85 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 8 | 0.12 | 0.85 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <pre><bound method Module.state_dict of CNN((conv1): Conv2d(1, 6, kernel_size=(5, 5), stride=(1, 1)) (conv2): Conv2d(6, 12, kernel_size=(3, 3), stride=(2, 2)) (conv2_drop): Dropout2d(p=0.5, inplace=False) (pool): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False) (fc1): Linear(in_features=2700, out_features=1000, bias=True) (fc2): Linear(in_features=1000, out_features=10, bias=True) (fc3): Linear(in_features=500, out_features=10, bias=True))></pre> |  <table><caption>Approximate data for Epochs 0-40</caption><tr><th>Epoch</th><th>Loss</th><th>Validation Accuracy</th></tr><tr><td>0</td><td>1.80</td><td>0.60</td></tr><tr><td>5</td><td>0.70</td><td>0.75</td></tr><tr><td>10</td><td>0.40</td><td>0.80</td></tr><tr><td>15</td><td>0.30</td><td>0.82</td></tr><tr><td>20</td><td>0.25</td><td>0.80</td></tr><tr><td>25</td><td>0.22</td><td>0.82</td></tr><tr><td>30</td><td>0.20</td><td>0.80</td></tr><tr><td>35</td><td>0.22</td><td>0.82</td></tr><tr><td>40</td><td>0.40</td><td>0.80</td></tr></table> | Epoch | Loss | Validation Accuracy | 0 | 1.80 | 0.60 | 5 | 0.70 | 0.75 | 10 | 0.40 | 0.80 | 15 | 0.30 | 0.82 | 20 | 0.25 | 0.80 | 25 | 0.22 | 0.82 | 30 | 0.20 | 0.80 | 35 | 0.22 | 0.82 | 40 | 0.40 | 0.80 | Criterion = CrossEntropyLoss() 53% Optimizer = optim.Adam(lr=0.001) |
| Epoch | Loss | Validation Accuracy | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 1.80 | 0.60 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 5 | 0.70 | 0.75 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 10 | 0.40 | 0.80 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 15 | 0.30 | 0.82 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 20 | 0.25 | 0.80 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 25 | 0.22 | 0.82 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 30 | 0.20 | 0.80 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 35 | 0.22 | 0.82 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 40 | 0.40 | 0.80 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |