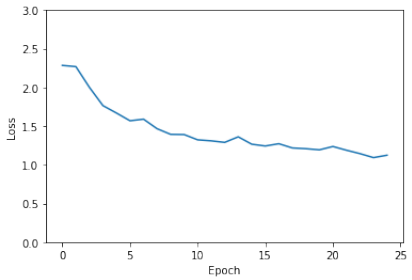
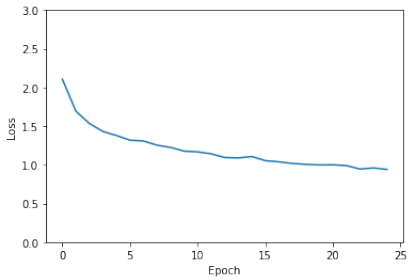
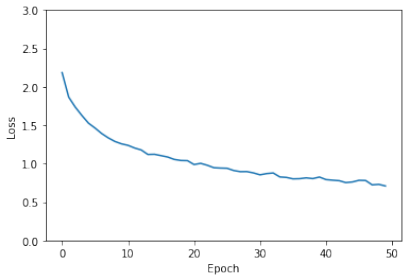
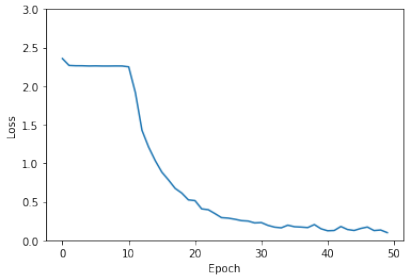
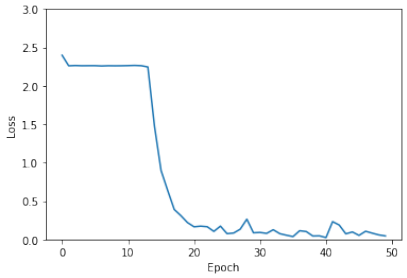
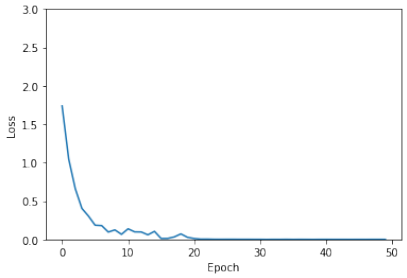
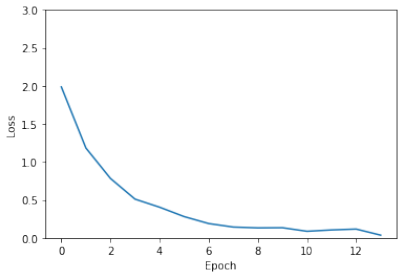
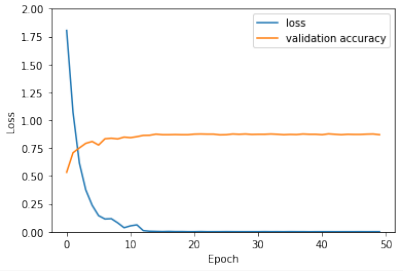
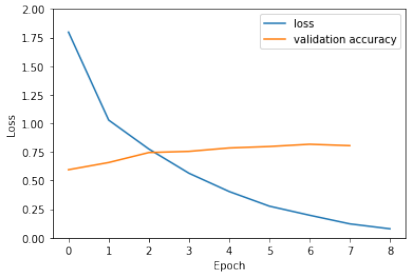
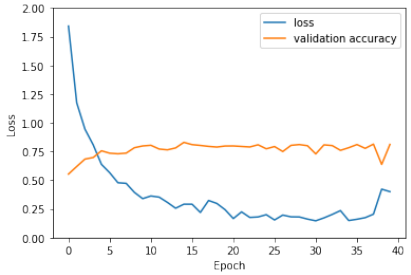
	State_dict	Loss function	Accuracy	Further info
	<bound method Module.state_dict of CNN((conv1): Conv2d(1, 6, kernel_size=(9, 9), stride=(2, 2)) (conv2): Conv2d(6, 18, kernel_size=(7, 7), stride=(2, 2)) (conv2_drop): Dropout2d(p=0.5, inplace=False) (pool): MaxPool2d(kernel_size=2, stride=2, padding=0, 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))>		43%	Criterion = CrossEntropyLoss() Optimizer = optim.SGD(lr=0.001, momentum=0.9)
	<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))>		50%	Criterion = CrossEntropyLoss() Optimizer = optim.SGD(lr=0.001, momentum=0.9)
	<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))>		45%	Criterion = CrossEntropyLoss() Optimizer = optim.SGD(lr=0.001, momentum=0.9)
	<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))>		44%	Criterion = CrossEntropyLoss() Optimizer = optim.SGD(lr=0.001, momentum=0.9)

	<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))>		54%	Criterion = CrossEntropyLoss() Optimizer = optim.SGD(lr=0.001, momentum=0.9)
	<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))>		61%	Criterion = CrossEntropyLoss() Optimizer = optim.SGD(lr=0.001, momentum=0.9)
	<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))>		56%	Criterion = CrossEntropyLoss() Optimizer = optim.SGD(lr=0.001, momentum=0.9)
	<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))>		62%	Criterion = CrossEntropyLoss() Optimizer = optim.SGD(lr=0.001, momentum=0.9)

	<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))>		63%	Criterion = CrossEntropyLoss() Optimizer = optim.Adam(lr=0.001)
	<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))>		53%	Criterion = CrossEntropyLoss() Optimizer = optim.Adam(lr=0.001)