Machine Learning Report

04

학번: 2017112200

이름: 신현호

1. CNN & MNIST
2. Change the current kernel size of the program to different size. (Change ‘kernel\_size’ parameter of ‘Conv2D’ function.) Repeat this three times and compare the results.

<Program Code>

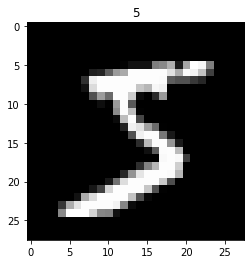
Kernel\_size = 5



<Result>

torch.Size([60000, 28, 28])

torch.Size([60000])



torch.Size([28, 28])

torch.Size([1, 28, 28])

tensor(7)

CNN(

(conv1): Sequential(

(0): Conv2d(1, 16, kernel\_size=(5, 5), stride=(1, 1), padding=(2, 2))

(1): ReLU()

(2): MaxPool2d(kernel\_size=2, stride=2, padding=0, dilation=1, ceil\_mode=False)

)

(conv2): Sequential(

(0): Conv2d(16, 32, kernel\_size=(5, 5), stride=(1, 1), padding=(2, 2))

(1): ReLU()

(2): MaxPool2d(kernel\_size=2, stride=2, padding=0, dilation=1, ceil\_mode=False)

)

(out): Linear(in\_features=1568, out\_features=10, bias=True)

)

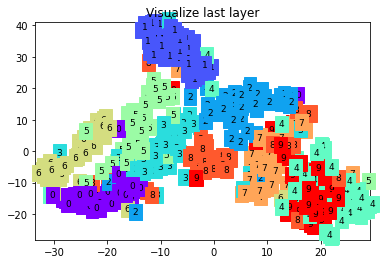
c:\users\hhshi\appdata\local\programs\python\python39\lib\site-packages\torchvision\datasets\mnist.py:69: UserWarning: test\_data has been renamed data

warnings.warn("test\_data has been renamed data")

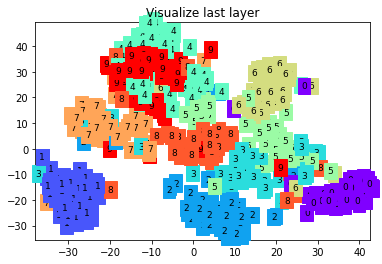
c:\users\hhshi\appdata\local\programs\python\python39\lib\site-packages\torchvision\datasets\mnist.py:59: UserWarning: test\_labels has been renamed targets

warnings.warn("test\_labels has been renamed targets")

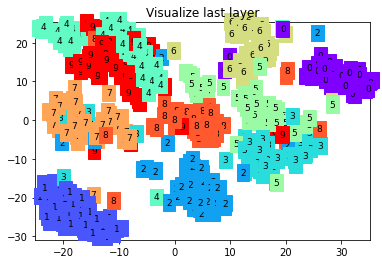
Epoch: 0 | train loss: 2.3125 | test accuracy: 0.18



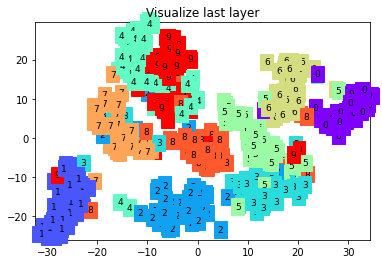
Epoch: 0 | train loss: 0.3737 | test accuracy: 0.90



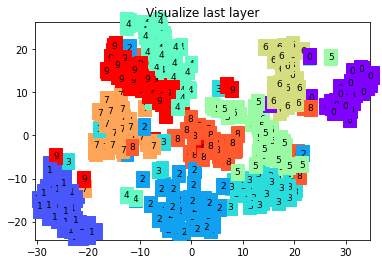
Epoch: 0 | train loss: 0.2347 | test accuracy: 0.93



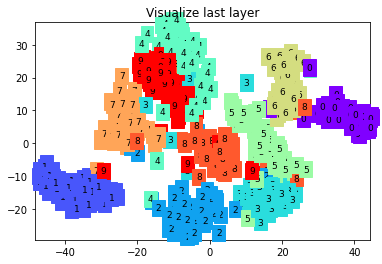
Epoch: 0 | train loss: 0.0560 | test accuracy: 0.95



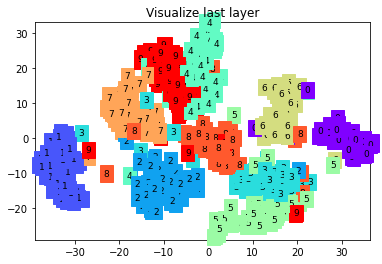
Epoch: 0 | train loss: 0.0692 | test accuracy: 0.96



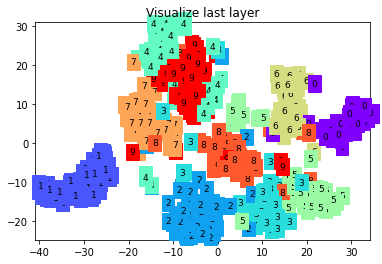
Epoch: 0 | train loss: 0.0858 | test accuracy: 0.97



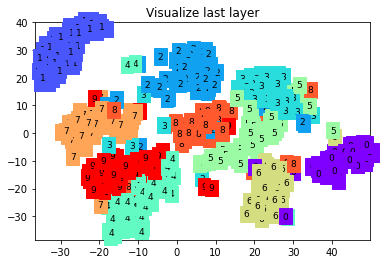
Epoch: 0 | train loss: 0.0406 | test accuracy: 0.96



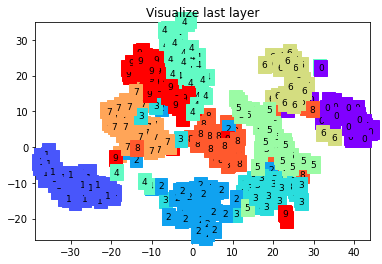
Epoch: 0 | train loss: 0.0192 | test accuracy: 0.96



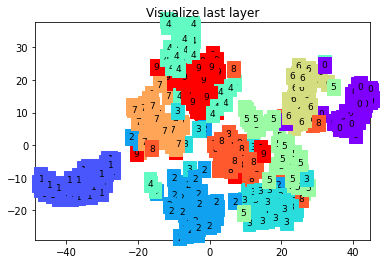
Epoch: 0 | train loss: 0.2302 | test accuracy: 0.96



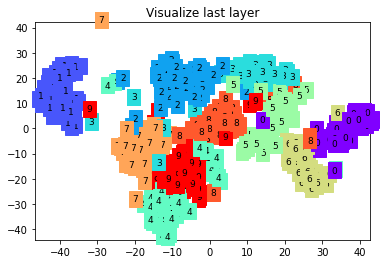
Epoch: 0 | train loss: 0.0435 | test accuracy: 0.97



Epoch: 0 | train loss: 0.0116 | test accuracy: 0.97



Epoch: 0 | train loss: 0.0254 | test accuracy: 0.97



Kernel\_size = 2

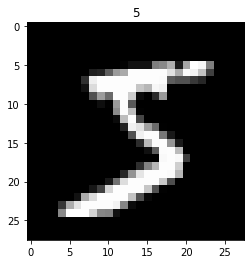
<Program Code>



<Result>

torch.Size([60000, 28, 28])

torch.Size([60000])



torch.Size([28, 28])

torch.Size([1, 28, 28])

tensor(7)

CNN(

(conv1): Sequential(

(0): Conv2d(1, 16, kernel\_size=(2, 2), stride=(1, 1), padding=(2, 2))

(1): ReLU()

(2): MaxPool2d(kernel\_size=2, stride=2, padding=0, dilation=1, ceil\_mode=False)

)

(conv2): Sequential(

(0): Conv2d(16, 32, kernel\_size=(5, 5), stride=(1, 1), padding=(2, 2))

(1): ReLU()

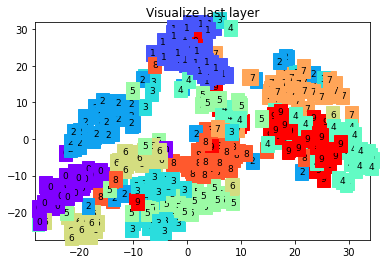
(2): MaxPool2d(kernel\_size=2, stride=2, padding=0, dilation=1, ceil\_mode=False)

)

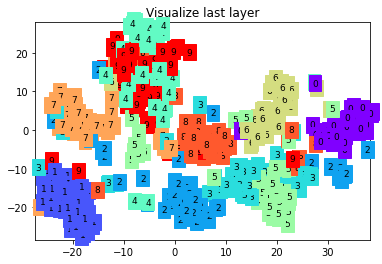
(out): Linear(in\_features=1568, out\_features=10, bias=True)

)

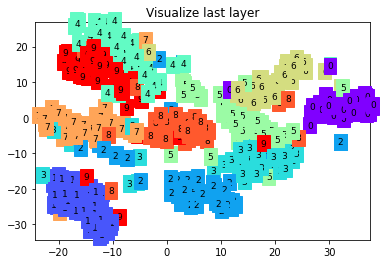
Epoch: 0 | train loss: 2.3008 | test accuracy: 0.18



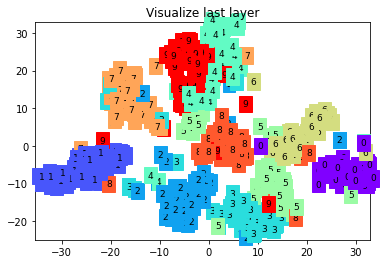
Epoch: 0 | train loss: 0.3984 | test accuracy: 0.87



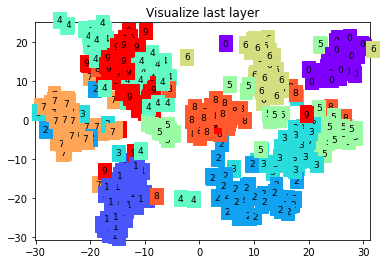
Epoch: 0 | train loss: 0.1475 | test accuracy: 0.90



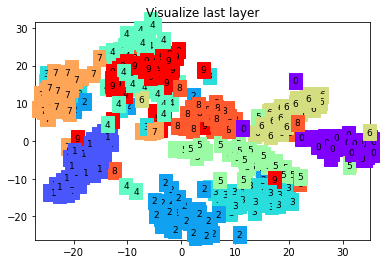
Epoch: 0 | train loss: 0.1792 | test accuracy: 0.94



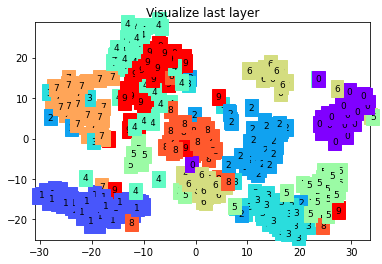
Epoch: 0 | train loss: 0.1819 | test accuracy: 0.94



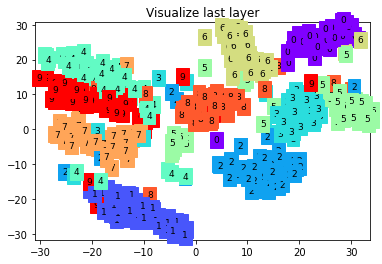
Epoch: 0 | train loss: 0.0734 | test accuracy: 0.95



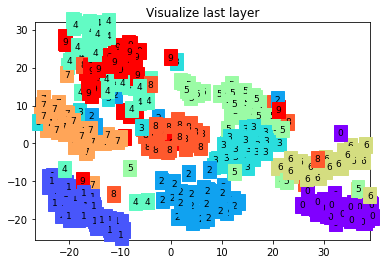
Epoch: 0 | train loss: 0.1659 | test accuracy: 0.95



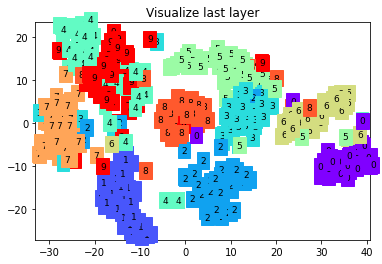
Epoch: 0 | train loss: 0.0518 | test accuracy: 0.97



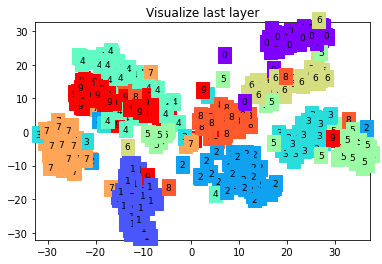
Epoch: 0 | train loss: 0.1663 | test accuracy: 0.97



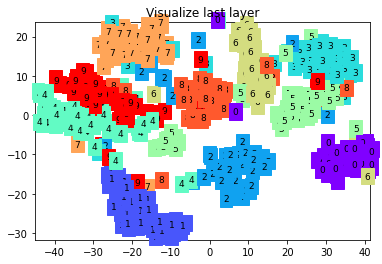
Epoch: 0 | train loss: 0.0569 | test accuracy: 0.97



Epoch: 0 | train loss: 0.0121 | test accuracy: 0.97



Epoch: 0 | train loss: 0.0194 | test accuracy: 0.97



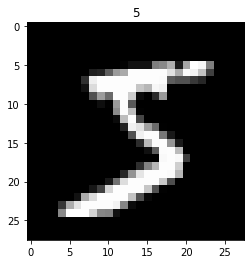
kernel\_size = 3

<Program Code>



torch.Size([60000, 28, 28])

torch.Size([60000])



torch.Size([28, 28])

torch.Size([1, 28, 28])

tensor(7)

CNN(

(conv1): Sequential(

(0): Conv2d(1, 16, kernel\_size=(3, 3), stride=(1, 1), padding=(2, 2))

(1): ReLU()

(2): MaxPool2d(kernel\_size=2, stride=2, padding=0, dilation=1, ceil\_mode=False)

)

(conv2): Sequential(

(0): Conv2d(16, 32, kernel\_size=(5, 5), stride=(1, 1), padding=(2, 2))

(1): ReLU()

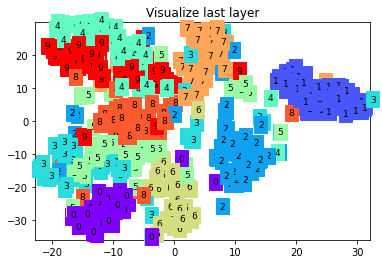
(2): MaxPool2d(kernel\_size=2, stride=2, padding=0, dilation=1, ceil\_mode=False)

)

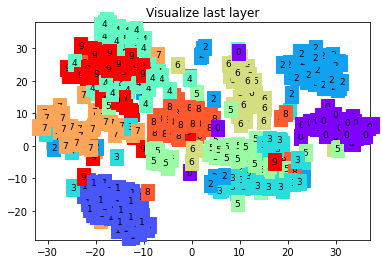
(out): Linear(in\_features=1568, out\_features=10, bias=True)

)

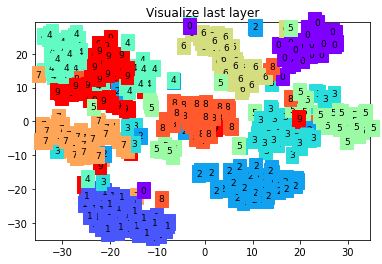
Epoch: 0 | train loss: 2.3128 | test accuracy: 0.10



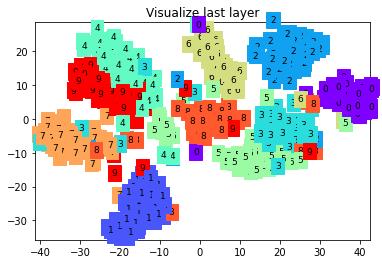
Epoch: 0 | train loss: 0.3744 | test accuracy: 0.88



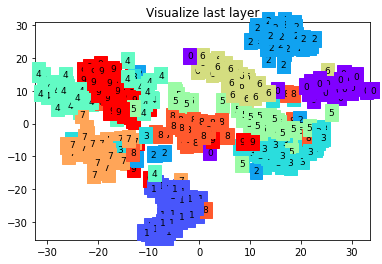
Epoch: 0 | train loss: 0.1303 | test accuracy: 0.93



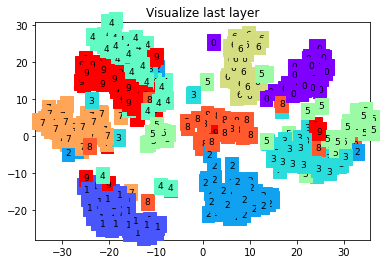
Epoch: 0 | train loss: 0.1236 | test accuracy: 0.94



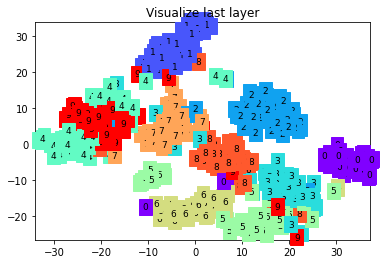
Epoch: 0 | train loss: 0.0587 | test accuracy: 0.95



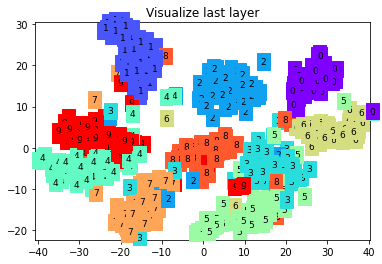
Epoch: 0 | train loss: 0.1617 | test accuracy: 0.96



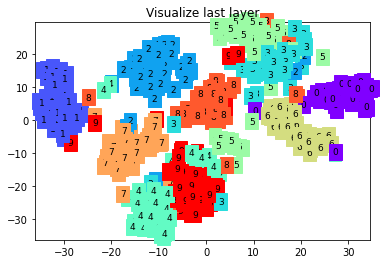
Epoch: 0 | train loss: 0.1740 | test accuracy: 0.96



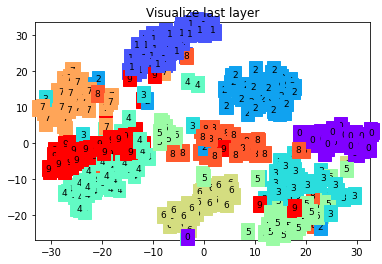
Epoch: 0 | train loss: 0.1260 | test accuracy: 0.96



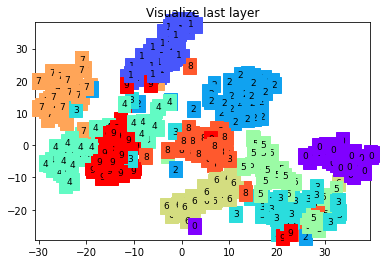
Epoch: 0 | train loss: 0.0348 | test accuracy: 0.97



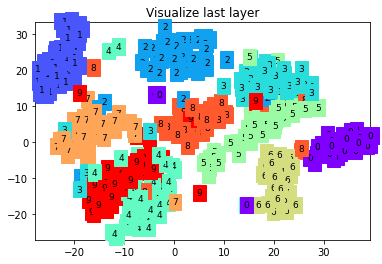
Epoch: 0 | train loss: 0.0802 | test accuracy: 0.97



Epoch: 0 | train loss: 0.0618 | test accuracy: 0.97



Epoch: 0 | train loss: 0.1244 | test accuracy: 0.97



1. Remove pooling layer in the program (you can remove ‘MaxPool2D’ function) and compare the results.

<Program Code>



<Result>

결과값에서 에러가 나서 실행할 수 없었다.

1. Change pooling layer in the program(ex. AvgPool2d, AdoptiveAvgPool2d, etc.) repeat this three times and compare the results.

<AvgPool2d>

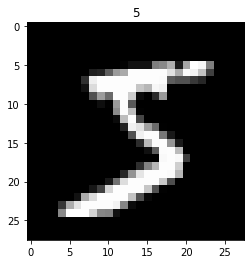
<Program Code>



<Result>

torch.Size([60000, 28, 28])

torch.Size([60000])



torch.Size([28, 28])

torch.Size([1, 28, 28])

tensor(7)

CNN(

(conv1): Sequential(

(0): Conv2d(1, 16, kernel\_size=(5, 5), stride=(1, 1), padding=(2, 2))

(1): ReLU()

(2): AvgPool2d(kernel\_size=2, stride=2, padding=0)

)

(conv2): Sequential(

(0): Conv2d(16, 32, kernel\_size=(5, 5), stride=(1, 1), padding=(2, 2))

(1): ReLU()

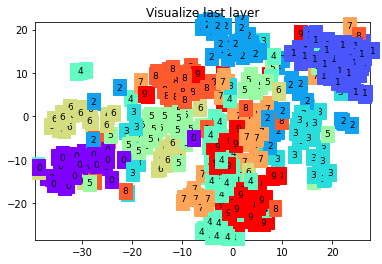
(2): AvgPool2d(kernel\_size=2, stride=2, padding=0)

)

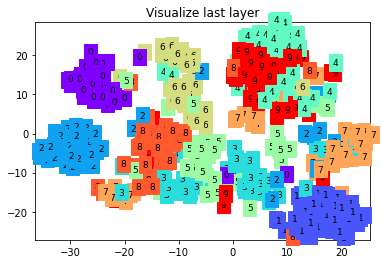
(out): Linear(in\_features=1568, out\_features=10, bias=True)

)

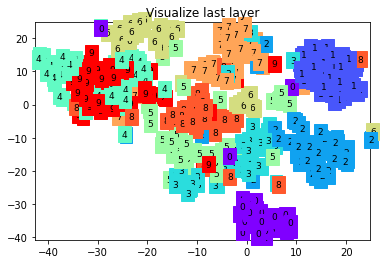
Epoch: 0 | train loss: 2.3088 | test accuracy: 0.18



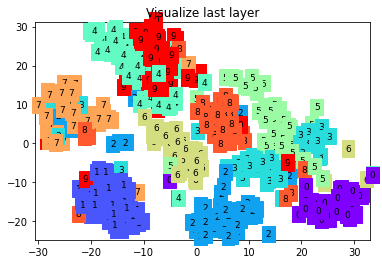
Epoch: 0 | train loss: 0.4829 | test accuracy: 0.87



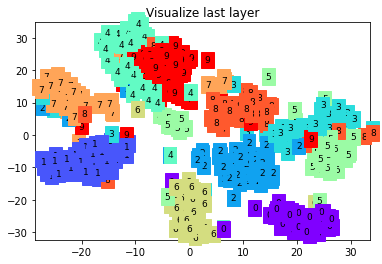
Epoch: 0 | train loss: 0.3913 | test accuracy: 0.90



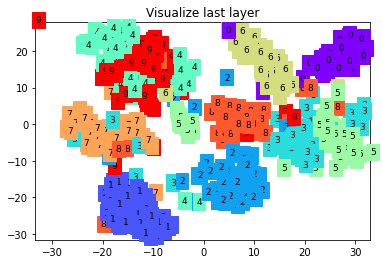
Epoch: 0 | train loss: 0.0858 | test accuracy: 0.92



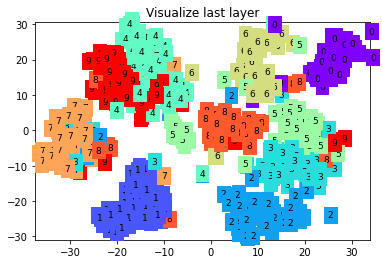
Epoch: 0 | train loss: 0.0980 | test accuracy: 0.94



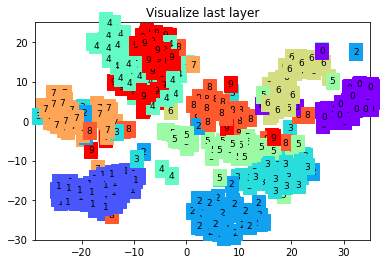
Epoch: 0 | train loss: 0.1788 | test accuracy: 0.94



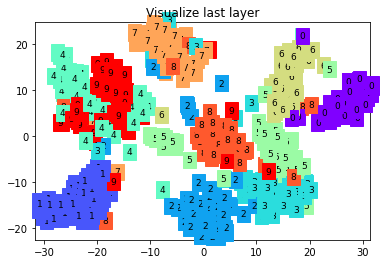
Epoch: 0 | train loss: 0.0716 | test accuracy: 0.94



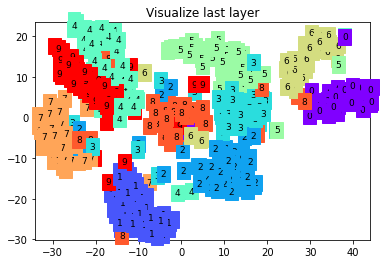
Epoch: 0 | train loss: 0.0258 | test accuracy: 0.95



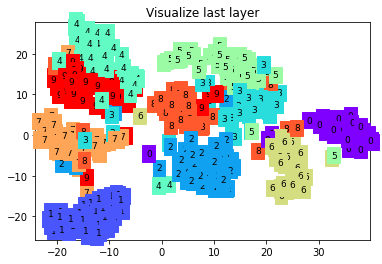
Epoch: 0 | train loss: 0.2510 | test accuracy: 0.96



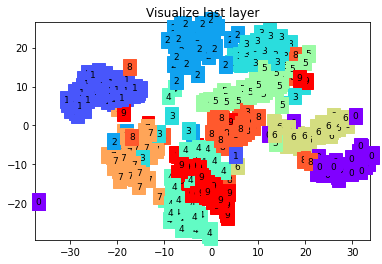
Epoch: 0 | train loss: 0.0938 | test accuracy: 0.96



Epoch: 0 | train loss: 0.0252 | test accuracy: 0.96



Epoch: 0 | train loss: 0.0424 | test accuracy: 0.97



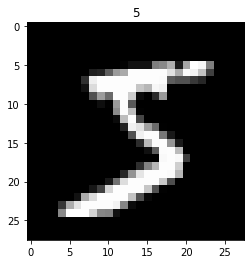
<LPPool2d>



<Result>

torch.Size([60000, 28, 28])

torch.Size([60000])



torch.Size([28, 28])

torch.Size([1, 28, 28])

tensor(7)

CNN(

(conv1): Sequential(

(0): Conv2d(1, 16, kernel\_size=(5, 5), stride=(1, 1), padding=(2, 2))

(1): ReLU()

(2): LPPool2d(norm\_type=2, kernel\_size=2, stride=None, ceil\_mode=False)

)

(conv2): Sequential(

(0): Conv2d(16, 32, kernel\_size=(5, 5), stride=(1, 1), padding=(2, 2))

(1): ReLU()

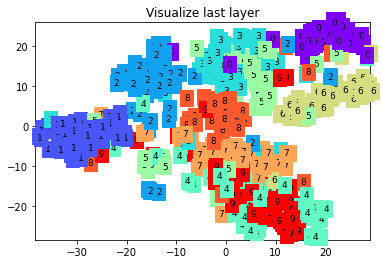
(2): LPPool2d(norm\_type=2, kernel\_size=2, stride=None, ceil\_mode=False)

)

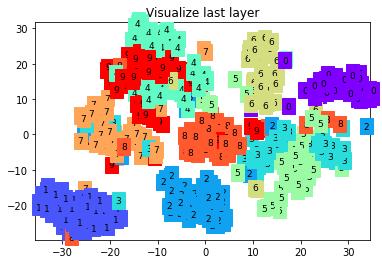
(out): Linear(in\_features=1568, out\_features=10, bias=True)

)

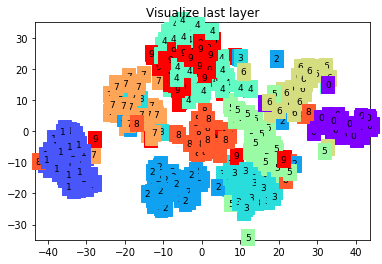
Epoch: 0 | train loss: 2.3422 | test accuracy: 0.18



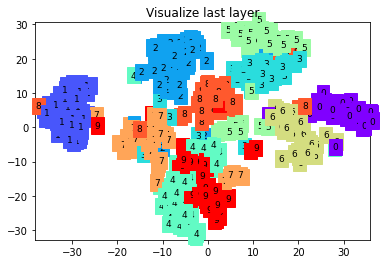
Epoch: 0 | train loss: 0.3763 | test accuracy: 0.91



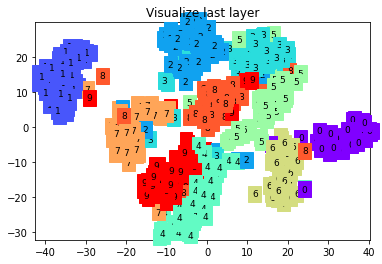
Epoch: 0 | train loss: 0.1736 | test accuracy: 0.94



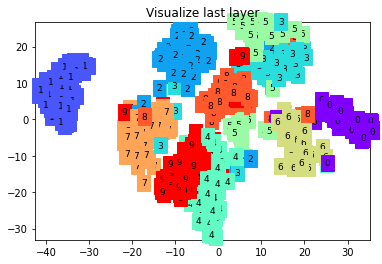
Epoch: 0 | train loss: 0.0445 | test accuracy: 0.95



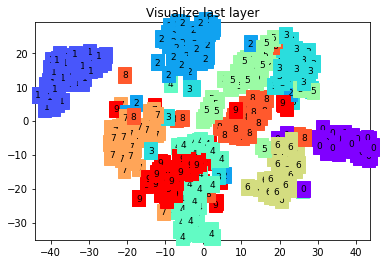
Epoch: 0 | train loss: 0.0609 | test accuracy: 0.97



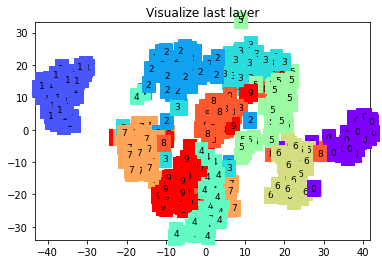
Epoch: 0 | train loss: 0.0768 | test accuracy: 0.97



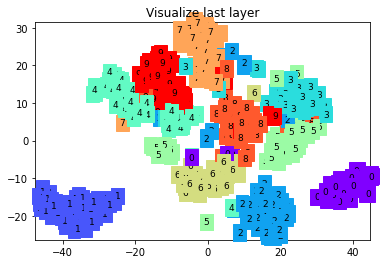
Epoch: 0 | train loss: 0.0427 | test accuracy: 0.96



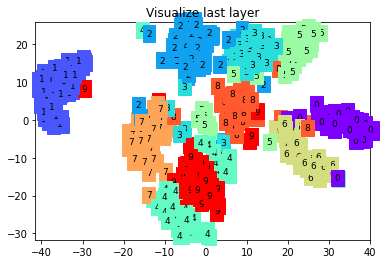
Epoch: 0 | train loss: 0.0191 | test accuracy: 0.97



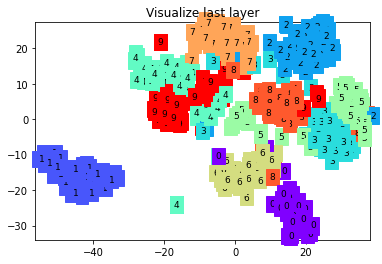
Epoch: 0 | train loss: 0.2095 | test accuracy: 0.96



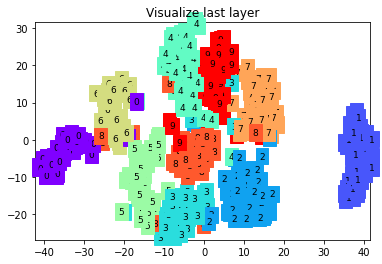
Epoch: 0 | train loss: 0.0448 | test accuracy: 0.97



Epoch: 0 | train loss: 0.0112 | test accuracy: 0.97



Epoch: 0 | train loss: 0.0410 | test accuracy: 0.98



1. Change the current activation function to other non-linear activation function (e.g. sigmoid, tanh, etc). You can do so by nn.Sigmoid() to nn.ReLU(), nn.Tanh(), etc. Repeat this five times and compare the results.

<Program Code>

PRelu



<Result>

torch.Size([60000, 28, 28])

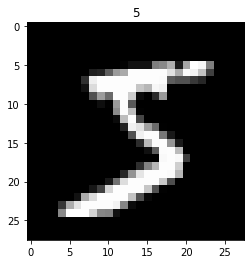
torch.Size([60000])

c:\users\hhshi\appdata\local\programs\python\python39\lib\site-packages\torchvision\datasets\mnist.py:64: UserWarning: train\_data has been renamed data

warnings.warn("train\_data has been renamed data")

c:\users\hhshi\appdata\local\programs\python\python39\lib\site-packages\torchvision\datasets\mnist.py:54: UserWarning: train\_labels has been renamed targets

warnings.warn("train\_labels has been renamed targets")



c:\users\hhshi\appdata\local\programs\python\python39\lib\site-packages\torchvision\datasets\mnist.py:69: UserWarning: test\_data has been renamed data

warnings.warn("test\_data has been renamed data")

c:\users\hhshi\appdata\local\programs\python\python39\lib\site-packages\torchvision\datasets\mnist.py:59: UserWarning: test\_labels has been renamed targets

warnings.warn("test\_labels has been renamed targets")

torch.Size([28, 28])

torch.Size([1, 28, 28])

tensor(7)

CNN(

(conv1): Sequential(

(0): Conv2d(1, 16, kernel\_size=(5, 5), stride=(1, 1), padding=(2, 2))

(1): PReLU(num\_parameters=1)

(2): MaxPool2d(kernel\_size=2, stride=2, padding=0, dilation=1, ceil\_mode=False)

)

(conv2): Sequential(

(0): Conv2d(16, 32, kernel\_size=(5, 5), stride=(1, 1), padding=(2, 2))

(1): PReLU(num\_parameters=1)

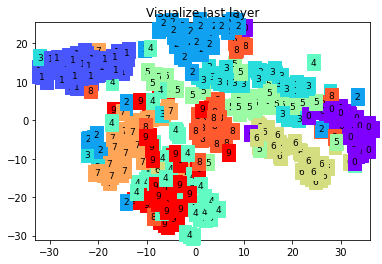
(2): MaxPool2d(kernel\_size=2, stride=2, padding=0, dilation=1, ceil\_mode=False)

)

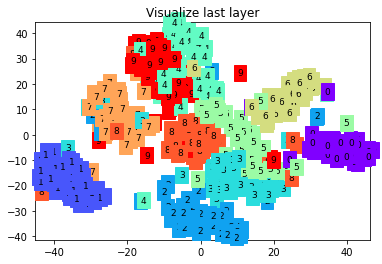
(out): Linear(in\_features=1568, out\_features=10, bias=True)

)

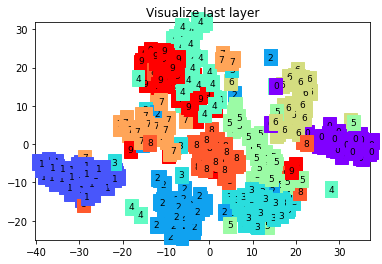
Epoch: 0 | train loss: 2.3136 | test accuracy: 0.19



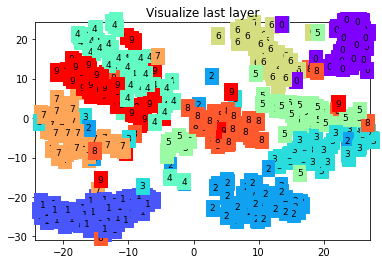
Epoch: 0 | train loss: 0.3940 | test accuracy: 0.90



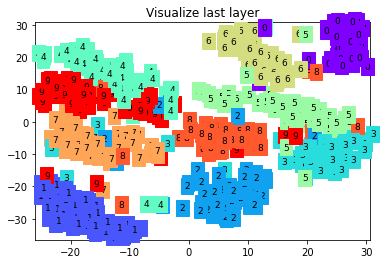
Epoch: 0 | train loss: 0.2604 | test accuracy: 0.93



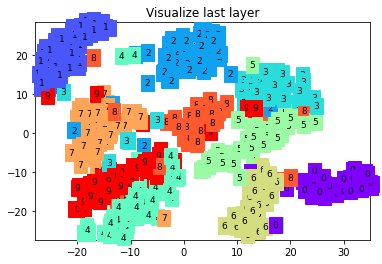
Epoch: 0 | train loss: 0.0616 | test accuracy: 0.94



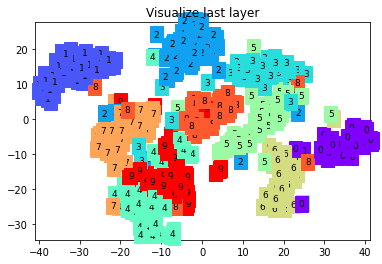
Epoch: 0 | train loss: 0.0430 | test accuracy: 0.96



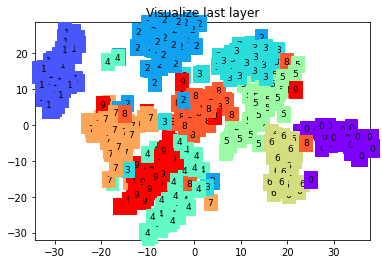
Epoch: 0 | train loss: 0.0718 | test accuracy: 0.97



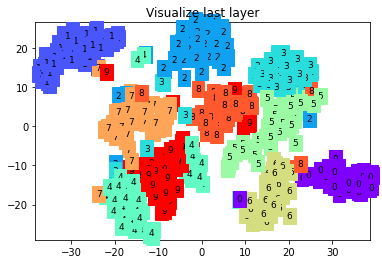
Epoch: 0 | train loss: 0.0556 | test accuracy: 0.96



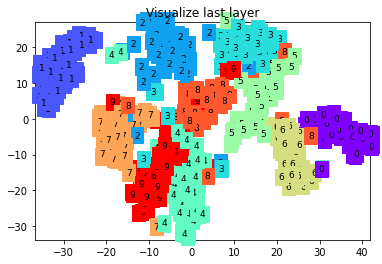
Epoch: 0 | train loss: 0.0208 | test accuracy: 0.96



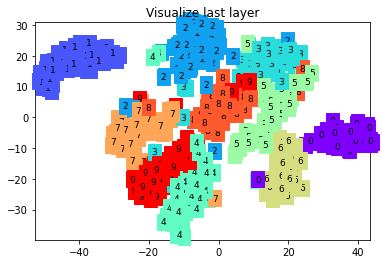
Epoch: 0 | train loss: 0.2310 | test accuracy: 0.96



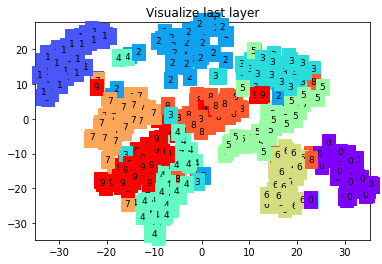
Epoch: 0 | train loss: 0.0580 | test accuracy: 0.97



Epoch: 0 | train loss: 0.0114 | test accuracy: 0.97



Epoch: 0 | train loss: 0.0340 | test accuracy: 0.98



<Program Code>

ReLU6



<Result>

torch.Size([60000, 28, 28])

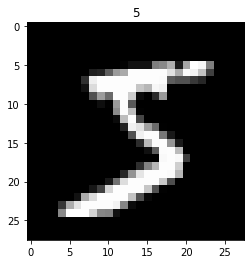
torch.Size([60000])

c:\users\hhshi\appdata\local\programs\python\python39\lib\site-packages\torchvision\datasets\mnist.py:64: UserWarning: train\_data has been renamed data

warnings.warn("train\_data has been renamed data")

c:\users\hhshi\appdata\local\programs\python\python39\lib\site-packages\torchvision\datasets\mnist.py:54: UserWarning: train\_labels has been renamed targets

warnings.warn("train\_labels has been renamed targets")



c:\users\hhshi\appdata\local\programs\python\python39\lib\site-packages\torchvision\datasets\mnist.py:69: UserWarning: test\_data has been renamed data

warnings.warn("test\_data has been renamed data")

c:\users\hhshi\appdata\local\programs\python\python39\lib\site-packages\torchvision\datasets\mnist.py:59: UserWarning: test\_labels has been renamed targets

warnings.warn("test\_labels has been renamed targets")

torch.Size([28, 28])

torch.Size([1, 28, 28])

tensor(7)

CNN(

(conv1): Sequential(

(0): Conv2d(1, 16, kernel\_size=(5, 5), stride=(1, 1), padding=(2, 2))

(1): ReLU6()

(2): MaxPool2d(kernel\_size=2, stride=2, padding=0, dilation=1, ceil\_mode=False)

)

(conv2): Sequential(

(0): Conv2d(16, 32, kernel\_size=(5, 5), stride=(1, 1), padding=(2, 2))

(1): ReLU6()

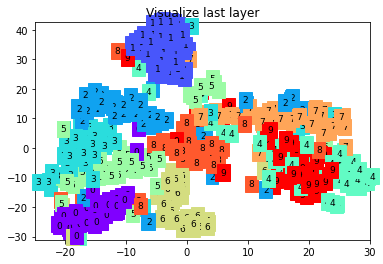
(2): MaxPool2d(kernel\_size=2, stride=2, padding=0, dilation=1, ceil\_mode=False)

)

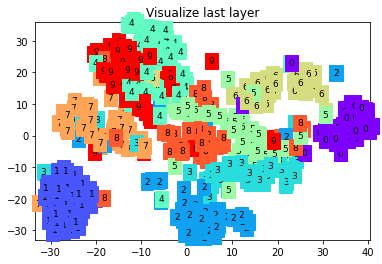
(out): Linear(in\_features=1568, out\_features=10, bias=True)

)

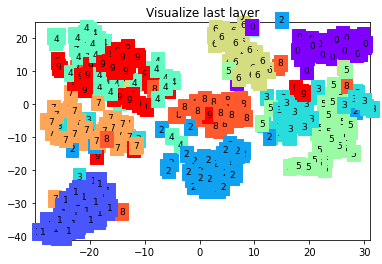
Epoch: 0 | train loss: 2.3125 | test accuracy: 0.18



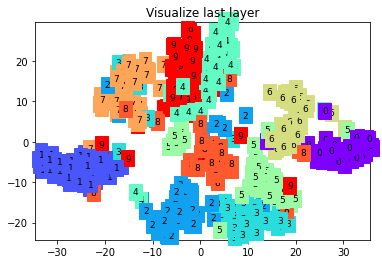
Epoch: 0 | train loss: 0.3739 | test accuracy: 0.90



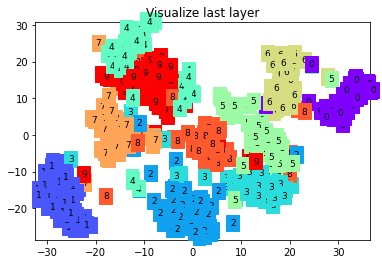
Epoch: 0 | train loss: 0.2341 | test accuracy: 0.93



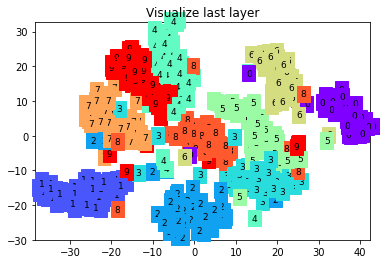
Epoch: 0 | train loss: 0.0551 | test accuracy: 0.95



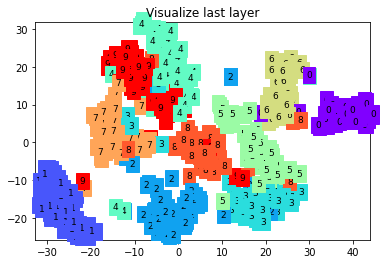
Epoch: 0 | train loss: 0.0681 | test accuracy: 0.96



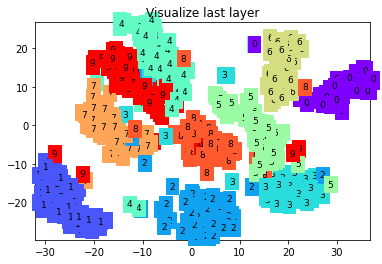
Epoch: 0 | train loss: 0.0854 | test accuracy: 0.97



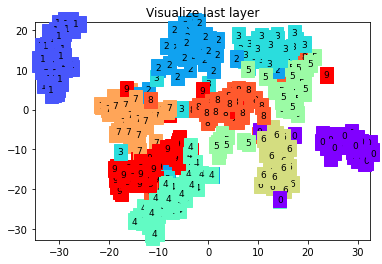
Epoch: 0 | train loss: 0.0408 | test accuracy: 0.96



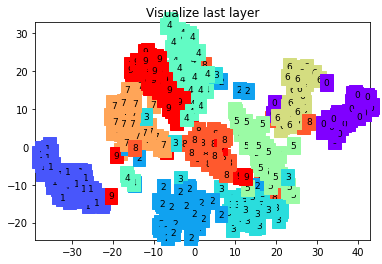
Epoch: 0 | train loss: 0.0199 | test accuracy: 0.96



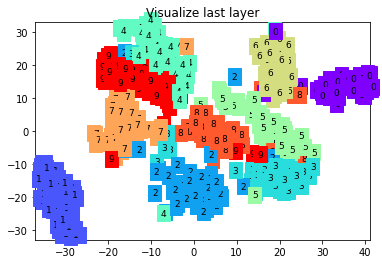
Epoch: 0 | train loss: 0.2294 | test accuracy: 0.96



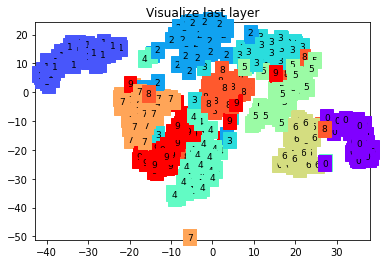
Epoch: 0 | train loss: 0.0443 | test accuracy: 0.97



Epoch: 0 | train loss: 0.0115 | test accuracy: 0.97



Epoch: 0 | train loss: 0.0244 | test accuracy: 0.97



<Program Code>

ELU



<Result>

torch.Size([60000, 28, 28])

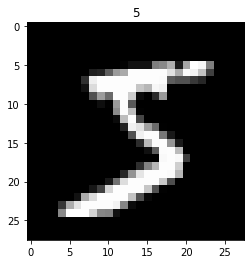
torch.Size([60000])

c:\users\hhshi\appdata\local\programs\python\python39\lib\site-packages\torchvision\datasets\mnist.py:64: UserWarning: train\_data has been renamed data

warnings.warn("train\_data has been renamed data")

c:\users\hhshi\appdata\local\programs\python\python39\lib\site-packages\torchvision\datasets\mnist.py:54: UserWarning: train\_labels has been renamed targets

warnings.warn("train\_labels has been renamed targets")



c:\users\hhshi\appdata\local\programs\python\python39\lib\site-packages\torchvision\datasets\mnist.py:69: UserWarning: test\_data has been renamed data

warnings.warn("test\_data has been renamed data")

c:\users\hhshi\appdata\local\programs\python\python39\lib\site-packages\torchvision\datasets\mnist.py:59: UserWarning: test\_labels has been renamed targets

warnings.warn("test\_labels has been renamed targets")

torch.Size([28, 28])

torch.Size([1, 28, 28])

tensor(7)

CNN(

(conv1): Sequential(

(0): Conv2d(1, 16, kernel\_size=(5, 5), stride=(1, 1), padding=(2, 2))

(1): ELU(alpha=1.0)

(2): MaxPool2d(kernel\_size=2, stride=2, padding=0, dilation=1, ceil\_mode=False)

)

(conv2): Sequential(

(0): Conv2d(16, 32, kernel\_size=(5, 5), stride=(1, 1), padding=(2, 2))

(1): ELU(alpha=1.0)

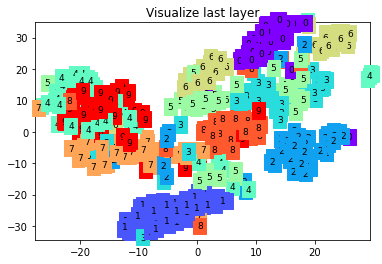
(2): MaxPool2d(kernel\_size=2, stride=2, padding=0, dilation=1, ceil\_mode=False)

)

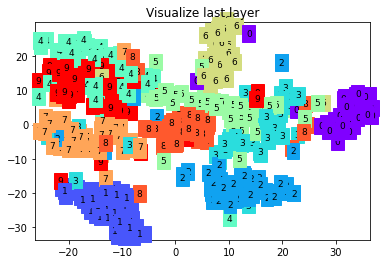
(out): Linear(in\_features=1568, out\_features=10, bias=True)

)

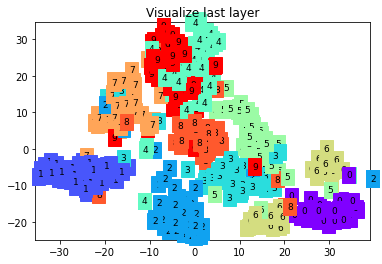
Epoch: 0 | train loss: 2.3177 | test accuracy: 0.22



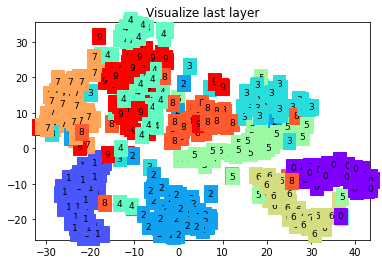
Epoch: 0 | train loss: 0.4017 | test accuracy: 0.89



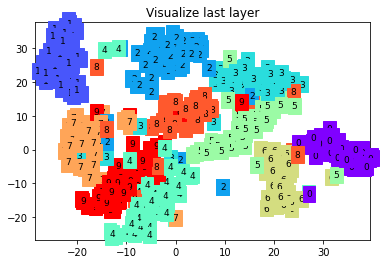
Epoch: 0 | train loss: 0.2759 | test accuracy: 0.93



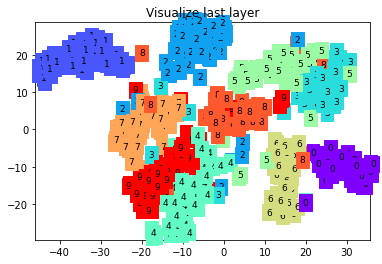
Epoch: 0 | train loss: 0.0678 | test accuracy: 0.94



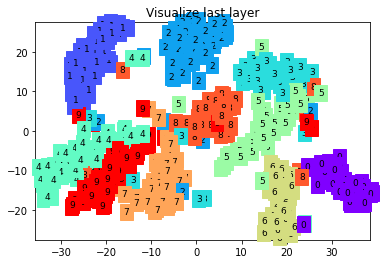
Epoch: 0 | train loss: 0.0249 | test accuracy: 0.96



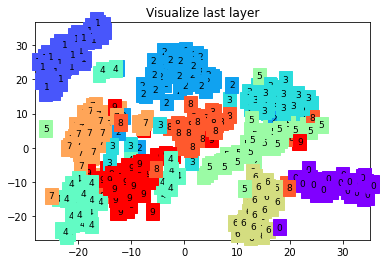
Epoch: 0 | train loss: 0.0606 | test accuracy: 0.97



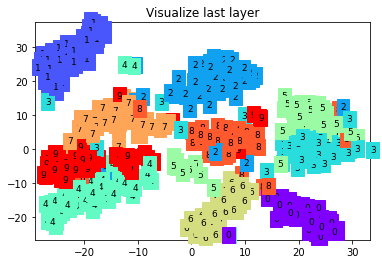
Epoch: 0 | train loss: 0.0475 | test accuracy: 0.96



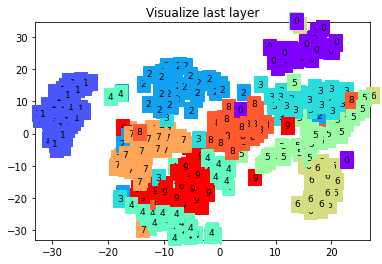
Epoch: 0 | train loss: 0.0297 | test accuracy: 0.97



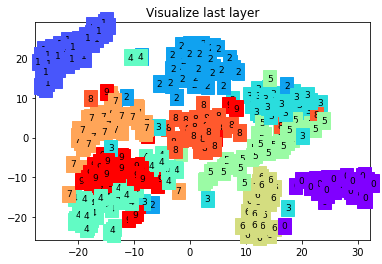
Epoch: 0 | train loss: 0.2154 | test accuracy: 0.96



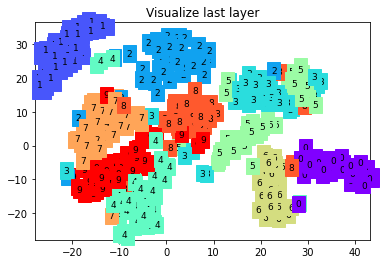
Epoch: 0 | train loss: 0.0696 | test accuracy: 0.97



Epoch: 0 | train loss: 0.0086 | test accuracy: 0.97



Epoch: 0 | train loss: 0.0313 | test accuracy: 0.98



<Program Code>

CELU



<Result>

torch.Size([60000, 28, 28])

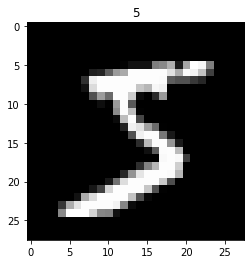
torch.Size([60000])

c:\users\hhshi\appdata\local\programs\python\python39\lib\site-packages\torchvision\datasets\mnist.py:64: UserWarning: train\_data has been renamed data

warnings.warn("train\_data has been renamed data")

c:\users\hhshi\appdata\local\programs\python\python39\lib\site-packages\torchvision\datasets\mnist.py:54: UserWarning: train\_labels has been renamed targets

warnings.warn("train\_labels has been renamed targets")



c:\users\hhshi\appdata\local\programs\python\python39\lib\site-packages\torchvision\datasets\mnist.py:69: UserWarning: test\_data has been renamed data

warnings.warn("test\_data has been renamed data")

c:\users\hhshi\appdata\local\programs\python\python39\lib\site-packages\torchvision\datasets\mnist.py:59: UserWarning: test\_labels has been renamed targets

warnings.warn("test\_labels has been renamed targets")

torch.Size([28, 28])

torch.Size([1, 28, 28])

tensor(7)

CNN(

(conv1): Sequential(

(0): Conv2d(1, 16, kernel\_size=(5, 5), stride=(1, 1), padding=(2, 2))

(1): CELU(alpha=1.0)

(2): MaxPool2d(kernel\_size=2, stride=2, padding=0, dilation=1, ceil\_mode=False)

)

(conv2): Sequential(

(0): Conv2d(16, 32, kernel\_size=(5, 5), stride=(1, 1), padding=(2, 2))

(1): CELU(alpha=1.0)

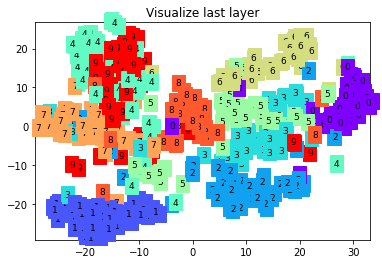
(2): MaxPool2d(kernel\_size=2, stride=2, padding=0, dilation=1, ceil\_mode=False)

)

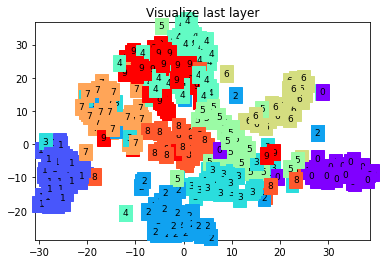
(out): Linear(in\_features=1568, out\_features=10, bias=True)

)

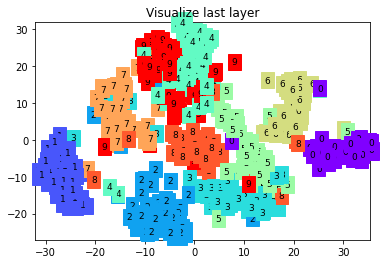
Epoch: 0 | train loss: 2.3177 | test accuracy: 0.22



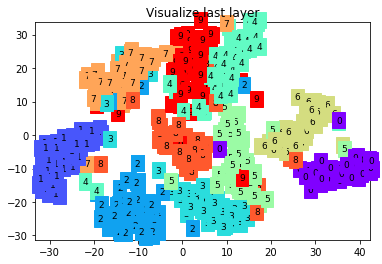
Epoch: 0 | train loss: 0.4017 | test accuracy: 0.89



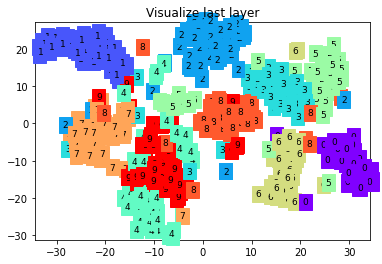
Epoch: 0 | train loss: 0.2759 | test accuracy: 0.93



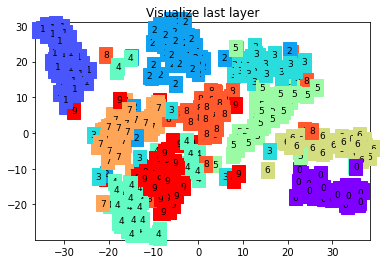
Epoch: 0 | train loss: 0.0678 | test accuracy: 0.94



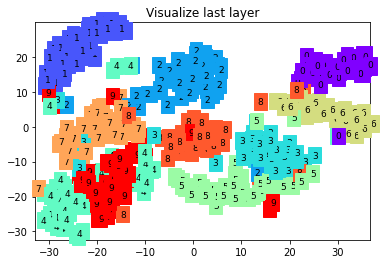
Epoch: 0 | train loss: 0.0249 | test accuracy: 0.96



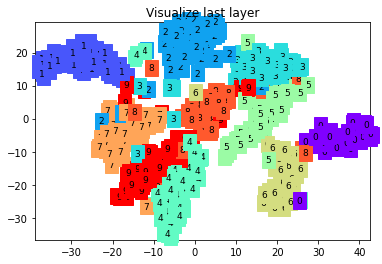
Epoch: 0 | train loss: 0.0606 | test accuracy: 0.97



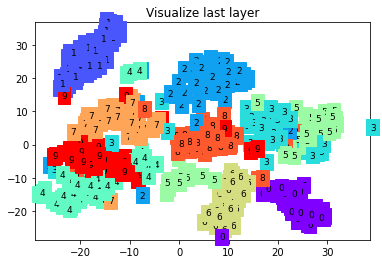
Epoch: 0 | train loss: 0.0475 | test accuracy: 0.96



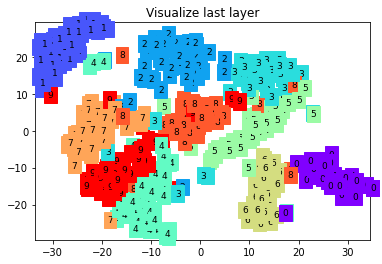
Epoch: 0 | train loss: 0.0297 | test accuracy: 0.97



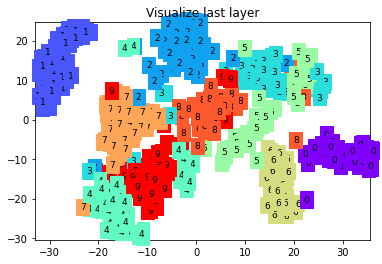
Epoch: 0 | train loss: 0.2154 | test accuracy: 0.96



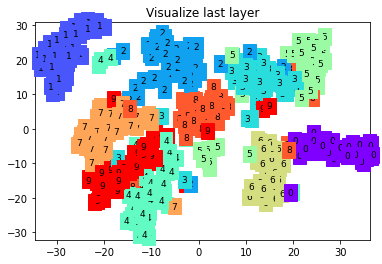
Epoch: 0 | train loss: 0.0696 | test accuracy: 0.97



Epoch: 0 | train loss: 0.0086 | test accuracy: 0.97



Epoch: 0 | train loss: 0.0313 | test accuracy: 0.98



<Program Code>  
SELU



<Result>

torch.Size([60000, 28, 28])

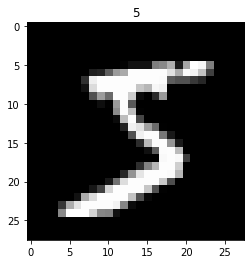
torch.Size([60000])

c:\users\hhshi\appdata\local\programs\python\python39\lib\site-packages\torchvision\datasets\mnist.py:64: UserWarning: train\_data has been renamed data

warnings.warn("train\_data has been renamed data")

c:\users\hhshi\appdata\local\programs\python\python39\lib\site-packages\torchvision\datasets\mnist.py:54: UserWarning: train\_labels has been renamed targets

warnings.warn("train\_labels has been renamed targets")



c:\users\hhshi\appdata\local\programs\python\python39\lib\site-packages\torchvision\datasets\mnist.py:69: UserWarning: test\_data has been renamed data

warnings.warn("test\_data has been renamed data")

c:\users\hhshi\appdata\local\programs\python\python39\lib\site-packages\torchvision\datasets\mnist.py:59: UserWarning: test\_labels has been renamed targets

warnings.warn("test\_labels has been renamed targets")

torch.Size([28, 28])

torch.Size([1, 28, 28])

tensor(7)

CNN(

(conv1): Sequential(

(0): Conv2d(1, 16, kernel\_size=(5, 5), stride=(1, 1), padding=(2, 2))

(1): SELU()

(2): MaxPool2d(kernel\_size=2, stride=2, padding=0, dilation=1, ceil\_mode=False)

)

(conv2): Sequential(

(0): Conv2d(16, 32, kernel\_size=(5, 5), stride=(1, 1), padding=(2, 2))

(1): SELU()

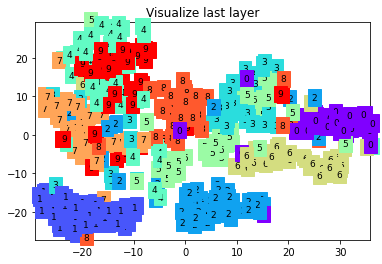
(2): MaxPool2d(kernel\_size=2, stride=2, padding=0, dilation=1, ceil\_mode=False)

)

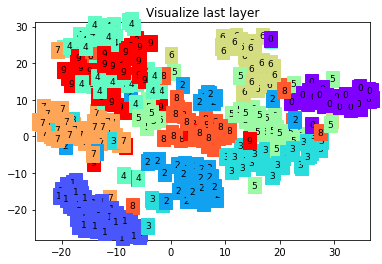
(out): Linear(in\_features=1568, out\_features=10, bias=True)

)

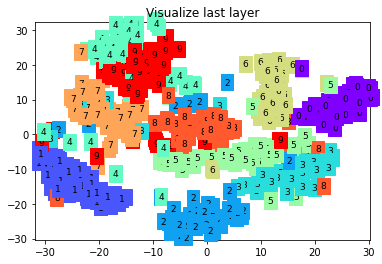
Epoch: 0 | train loss: 2.3252 | test accuracy: 0.21



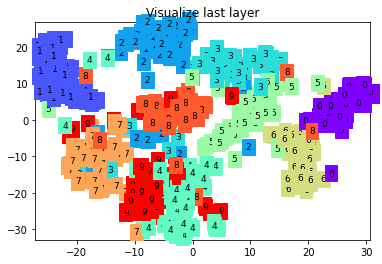
Epoch: 0 | train loss: 0.4005 | test accuracy: 0.90



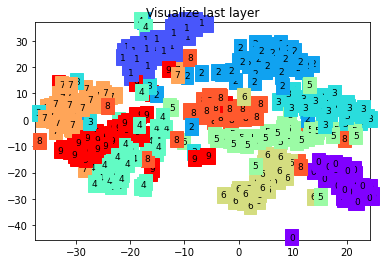
Epoch: 0 | train loss: 0.2871 | test accuracy: 0.92



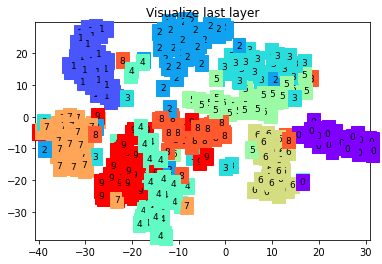
Epoch: 0 | train loss: 0.0689 | test accuracy: 0.94



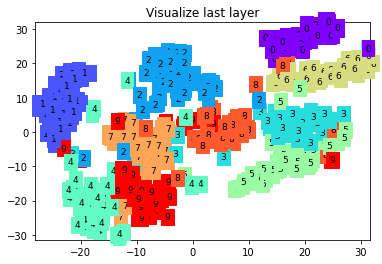
Epoch: 0 | train loss: 0.0240 | test accuracy: 0.95



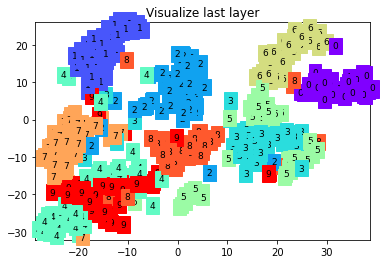
Epoch: 0 | train loss: 0.0603 | test accuracy: 0.97



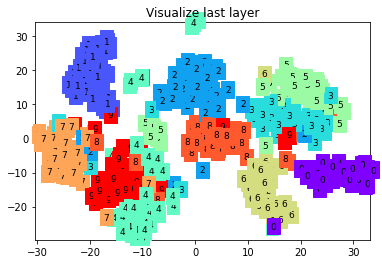
Epoch: 0 | train loss: 0.0435 | test accuracy: 0.96



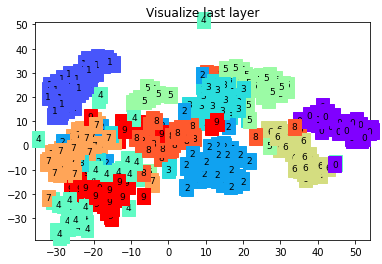
Epoch: 0 | train loss: 0.0337 | test accuracy: 0.97



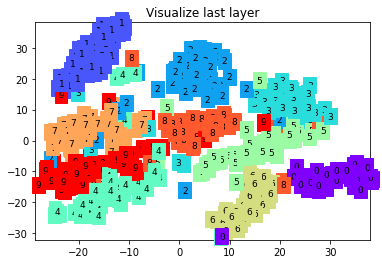
Epoch: 0 | train loss: 0.2206 | test accuracy: 0.96



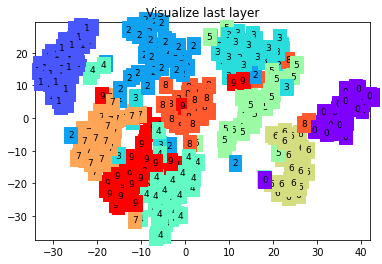
Epoch: 0 | train loss: 0.0511 | test accuracy: 0.97



Epoch: 0 | train loss: 0.0077 | test accuracy: 0.97



Epoch: 0 | train loss: 0.0286 | test accuracy: 0.98



1. Add dropout layers to the hidden layer and compare the results.

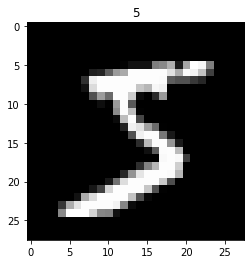
<Program Code>



<Result>

torch.Size([60000, 28, 28])

torch.Size([60000])



torch.Size([28, 28])

torch.Size([1, 28, 28])

tensor(7)

CNN(

(conv1): Sequential(

(0): Conv2d(1, 16, kernel\_size=(5, 5), stride=(1, 1), padding=(2, 2))

(1): ReLU()

(2): MaxPool2d(kernel\_size=2, stride=2, padding=0, dilation=1, ceil\_mode=False)

)

(conv2): Sequential(

(0): Conv2d(16, 32, kernel\_size=(5, 5), stride=(1, 1), padding=(2, 2))

(1): ReLU()

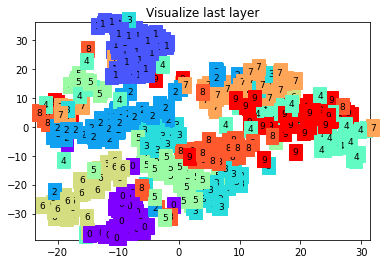
(2): MaxPool2d(kernel\_size=2, stride=2, padding=0, dilation=1, ceil\_mode=False)

)

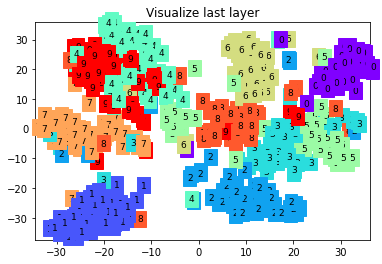
(out): Linear(in\_features=1568, out\_features=10, bias=True)

)

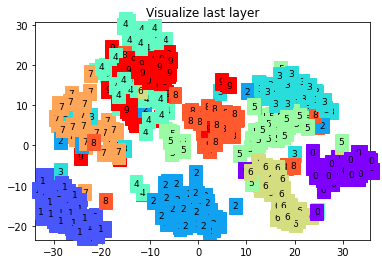
Epoch: 0 | train loss: 2.3125 | test accuracy: 0.18



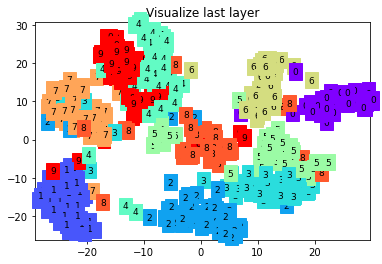
Epoch: 0 | train loss: 0.3737 | test accuracy: 0.90



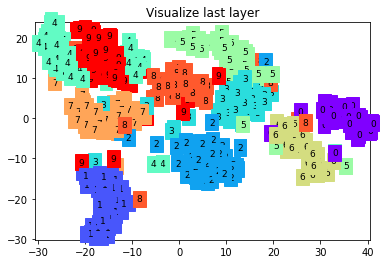
Epoch: 0 | train loss: 0.2347 | test accuracy: 0.93



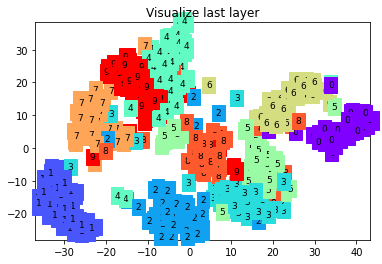
Epoch: 0 | train loss: 0.0560 | test accuracy: 0.95



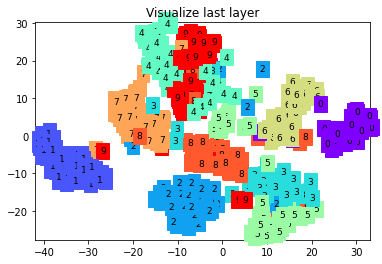
Epoch: 0 | train loss: 0.0692 | test accuracy: 0.96



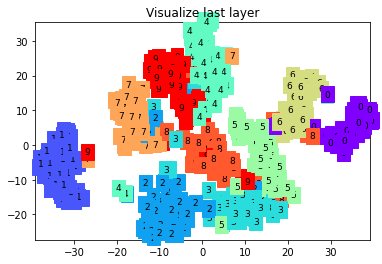
Epoch: 0 | train loss: 0.0858 | test accuracy: 0.97



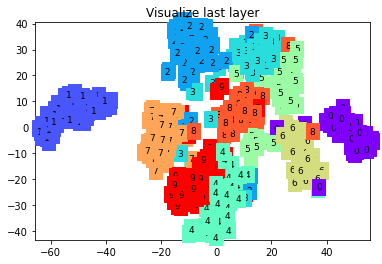
Epoch: 0 | train loss: 0.0406 | test accuracy: 0.96



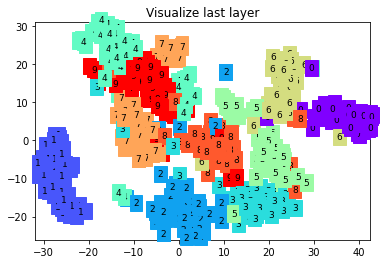
Epoch: 0 | train loss: 0.0192 | test accuracy: 0.96



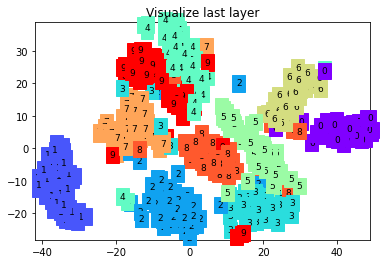
Epoch: 0 | train loss: 0.2302 | test accuracy: 0.96



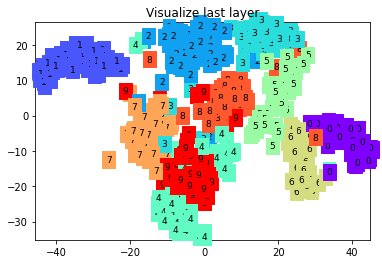
Epoch: 0 | train loss: 0.0435 | test accuracy: 0.97



Epoch: 0 | train loss: 0.0116 | test accuracy: 0.97



Epoch: 0 | train loss: 0.0254 | test accuracy: 0.97



1. Change the current optimization method to other optimization methods (e.g. adam, adaGrad, RMSProp, adaDelta, etc). You can use torch.optim.Adam, etc. Repeat this three times and compare the results.

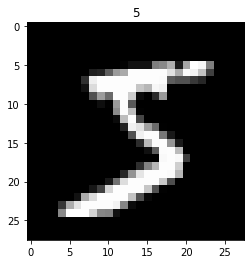
<Program Code>  
Adagrad



<Result>

torch.Size([60000, 28, 28])

torch.Size([60000])



torch.Size([28, 28])

torch.Size([1, 28, 28])

tensor(7)

CNN(

(conv1): Sequential(

(0): Conv2d(1, 16, kernel\_size=(5, 5), stride=(1, 1), padding=(2, 2))

(1): ReLU()

(2): MaxPool2d(kernel\_size=2, stride=2, padding=0, dilation=1, ceil\_mode=False)

)

(conv2): Sequential(

(0): Conv2d(16, 32, kernel\_size=(5, 5), stride=(1, 1), padding=(2, 2))

(1): ReLU()

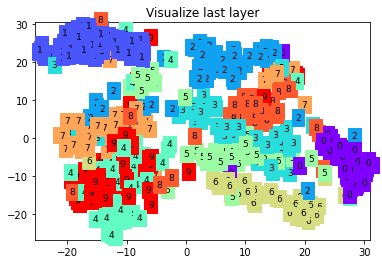
(2): MaxPool2d(kernel\_size=2, stride=2, padding=0, dilation=1, ceil\_mode=False)

)

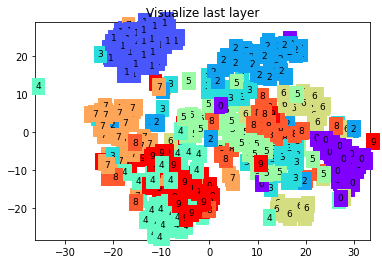
(out): Linear(in\_features=1568, out\_features=10, bias=True)

)

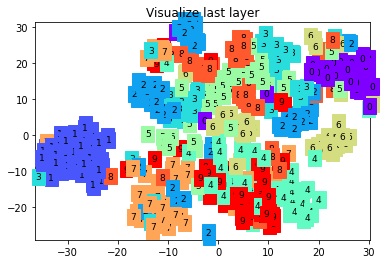
Epoch: 0 | train loss: 2.3125 | test accuracy: 0.18



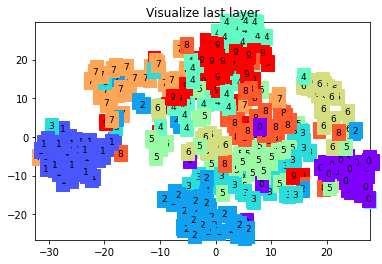
Epoch: 0 | train loss: 0.9193 | test accuracy: 0.77



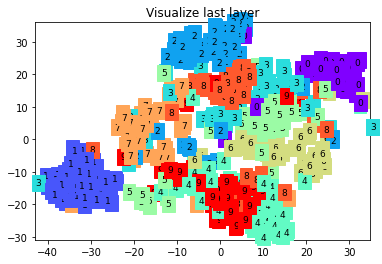
Epoch: 0 | train loss: 0.7785 | test accuracy: 0.80



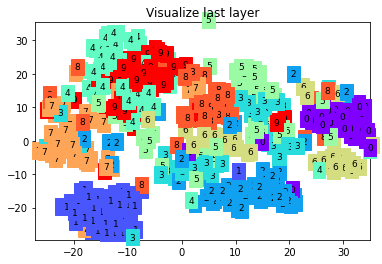
Epoch: 0 | train loss: 0.3836 | test accuracy: 0.82



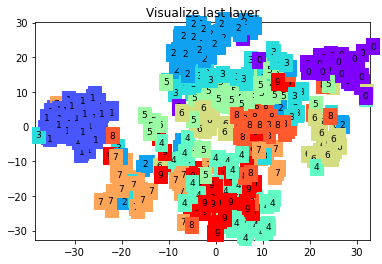
Epoch: 0 | train loss: 0.4267 | test accuracy: 0.84



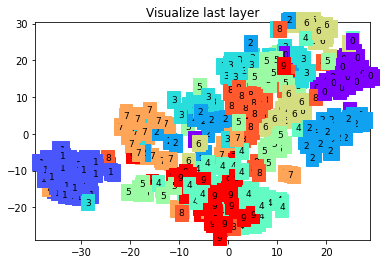
Epoch: 0 | train loss: 0.4352 | test accuracy: 0.86



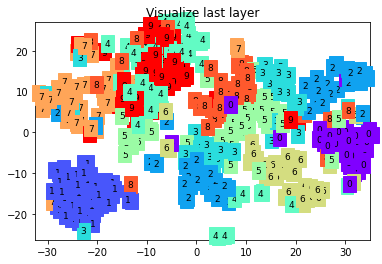
Epoch: 0 | train loss: 0.2942 | test accuracy: 0.86



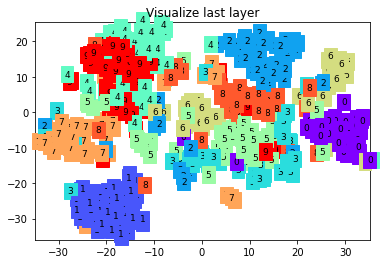
Epoch: 0 | train loss: 0.1927 | test accuracy: 0.86



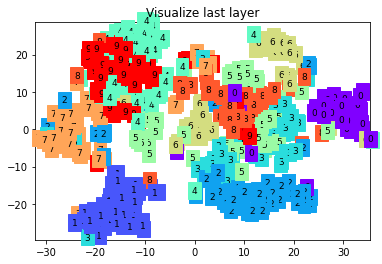
Epoch: 0 | train loss: 0.3950 | test accuracy: 0.87



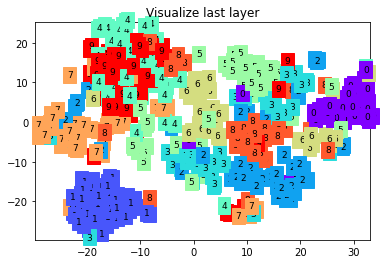
Epoch: 0 | train loss: 0.3412 | test accuracy: 0.87



Epoch: 0 | train loss: 0.1926 | test accuracy: 0.88



Epoch: 0 | train loss: 0.2971 | test accuracy: 0.88



<Program Code>

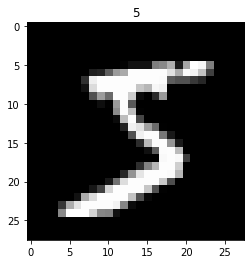
RMSprop



<Result>

torch.Size([60000, 28, 28])

torch.Size([60000])



torch.Size([28, 28])

torch.Size([1, 28, 28])

tensor(7)

CNN(

(conv1): Sequential(

(0): Conv2d(1, 16, kernel\_size=(5, 5), stride=(1, 1), padding=(2, 2))

(1): ReLU()

(2): MaxPool2d(kernel\_size=2, stride=2, padding=0, dilation=1, ceil\_mode=False)

)

(conv2): Sequential(

(0): Conv2d(16, 32, kernel\_size=(5, 5), stride=(1, 1), padding=(2, 2))

(1): ReLU()

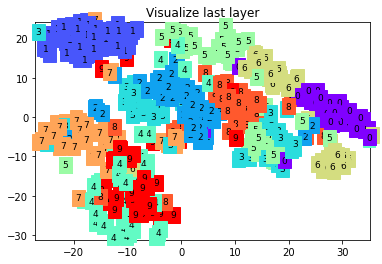
(2): MaxPool2d(kernel\_size=2, stride=2, padding=0, dilation=1, ceil\_mode=False)

)

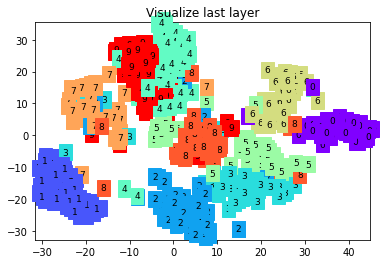
(out): Linear(in\_features=1568, out\_features=10, bias=True)

)

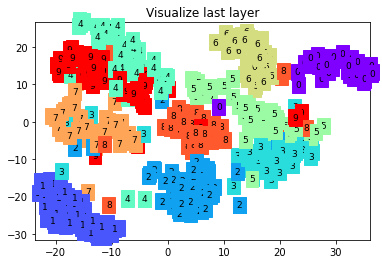
Epoch: 0 | train loss: 2.3125 | test accuracy: 0.14



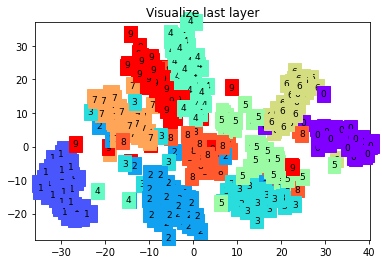
Epoch: 0 | train loss: 0.2791 | test accuracy: 0.94



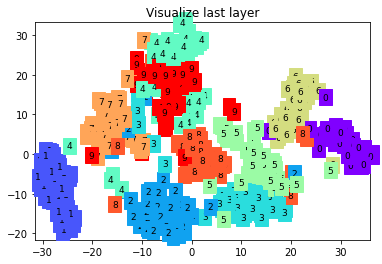
Epoch: 0 | train loss: 0.1765 | test accuracy: 0.95



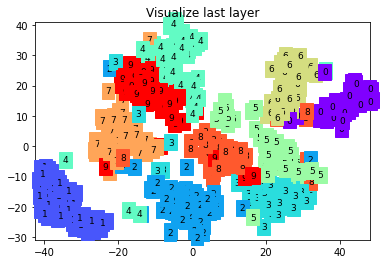
Epoch: 0 | train loss: 0.0379 | test accuracy: 0.95



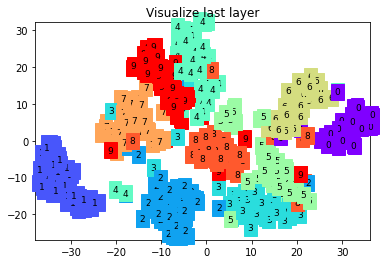
Epoch: 0 | train loss: 0.0800 | test accuracy: 0.96



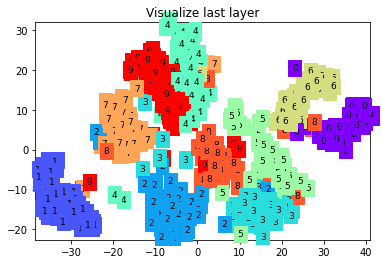
Epoch: 0 | train loss: 0.0419 | test accuracy: 0.97



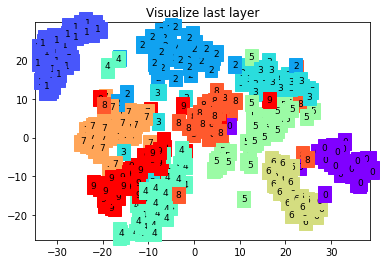
Epoch: 0 | train loss: 0.0439 | test accuracy: 0.97



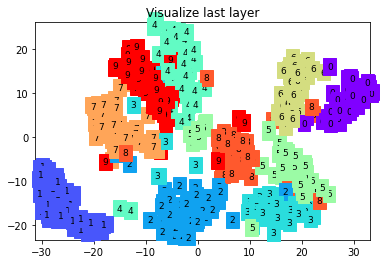
Epoch: 0 | train loss: 0.0172 | test accuracy: 0.97



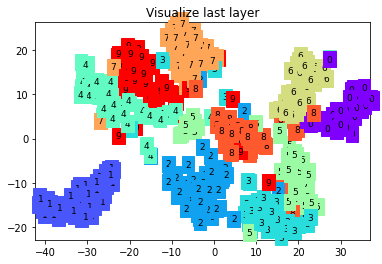
Epoch: 0 | train loss: 0.1743 | test accuracy: 0.97



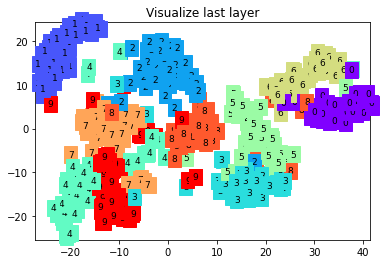
Epoch: 0 | train loss: 0.0251 | test accuracy: 0.97



Epoch: 0 | train loss: 0.0120 | test accuracy: 0.98

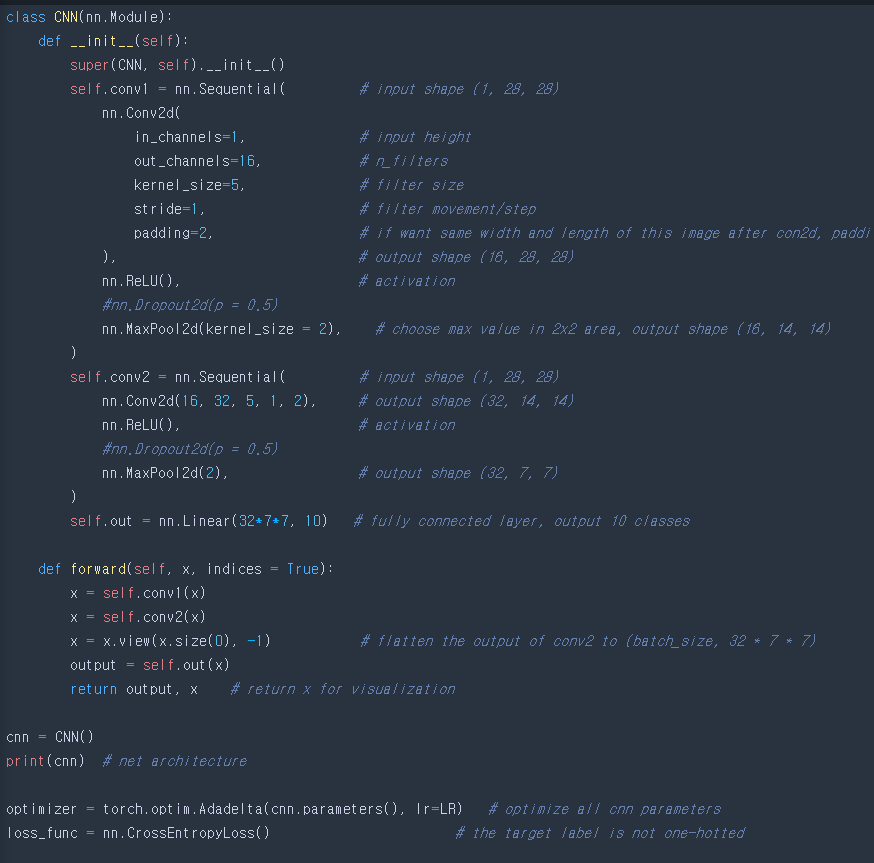


Epoch: 0 | train loss: 0.0476 | test accuracy: 0.98



<Program Code>

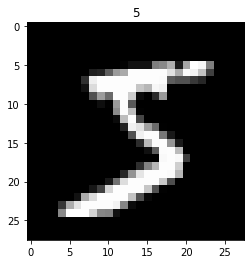
Adadelta



<Result>

torch.Size([60000, 28, 28])

torch.Size([60000])



torch.Size([28, 28])

torch.Size([1, 28, 28])

tensor(7)

CNN(

(conv1): Sequential(

(0): Conv2d(1, 16, kernel\_size=(5, 5), stride=(1, 1), padding=(2, 2))

(1): ReLU()

(2): MaxPool2d(kernel\_size=2, stride=2, padding=0, dilation=1, ceil\_mode=False)

)

(conv2): Sequential(

(0): Conv2d(16, 32, kernel\_size=(5, 5), stride=(1, 1), padding=(2, 2))

(1): ReLU()

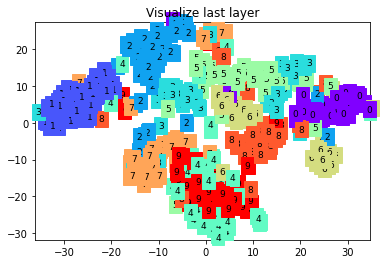
(2): MaxPool2d(kernel\_size=2, stride=2, padding=0, dilation=1, ceil\_mode=False)

)

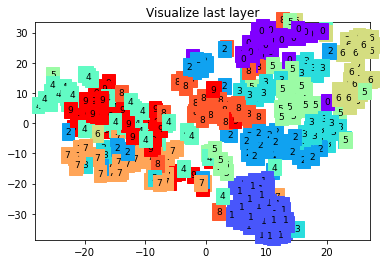
(out): Linear(in\_features=1568, out\_features=10, bias=True)

)

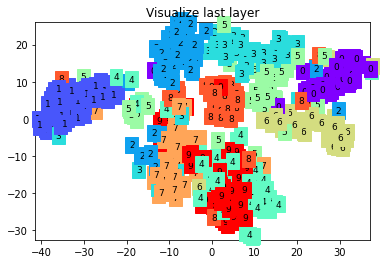
Epoch: 0 | train loss: 2.3125 | test accuracy: 0.05



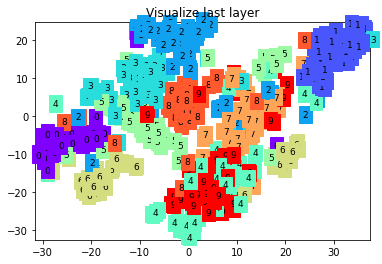
Epoch: 0 | train loss: 2.2913 | test accuracy: 0.09



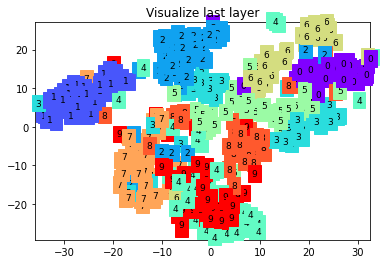
Epoch: 0 | train loss: 2.2929 | test accuracy: 0.15



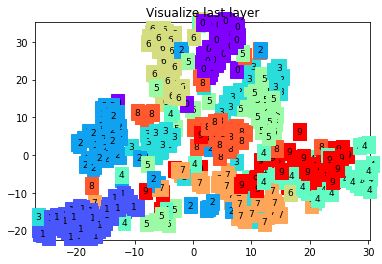
Epoch: 0 | train loss: 2.2995 | test accuracy: 0.18



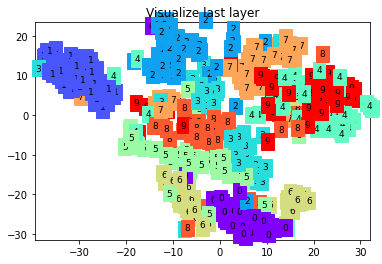
Epoch: 0 | train loss: 2.2861 | test accuracy: 0.22



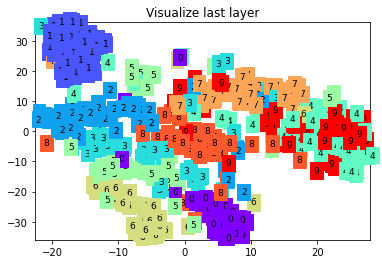
Epoch: 0 | train loss: 2.2717 | test accuracy: 0.25



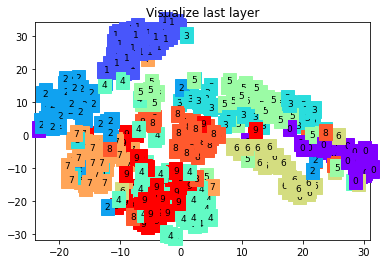
Epoch: 0 | train loss: 2.2604 | test accuracy: 0.28



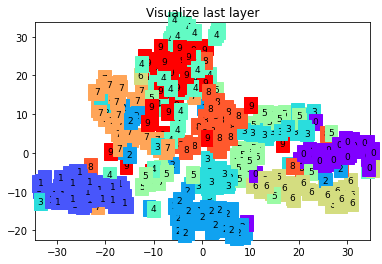
Epoch: 0 | train loss: 2.2678 | test accuracy: 0.29



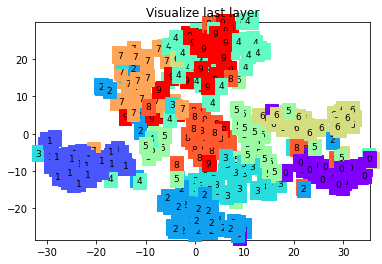
Epoch: 0 | train loss: 2.2498 | test accuracy: 0.30



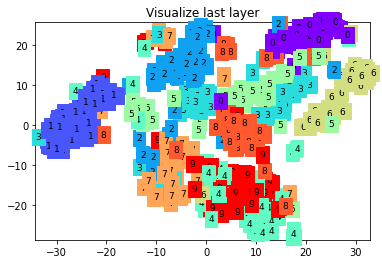
Epoch: 0 | train loss: 2.2456 | test accuracy: 0.32



Epoch: 0 | train loss: 2.2370 | test accuracy: 0.33



Epoch: 0 | train loss: 2.2435 | test accuracy: 0.36



1. Now add the Xavier weight initialization method and compare the results. (use torch.nn.init.xavier\_uniform)

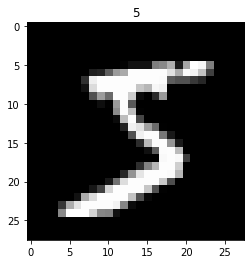
<Program Code>



<Result>

torch.Size([60000, 28, 28])

torch.Size([60000])



torch.Size([28, 28])

torch.Size([1, 28, 28])

tensor(7)

CNN(

(conv1): Sequential(

(0): Conv2d(1, 16, kernel\_size=(5, 5), stride=(1, 1), padding=(2, 2))

(1): ReLU()

(2): MaxPool2d(kernel\_size=2, stride=2, padding=0, dilation=1, ceil\_mode=False)

)

(conv2): Sequential(

(0): Conv2d(16, 32, kernel\_size=(5, 5), stride=(1, 1), padding=(2, 2))

(1): ReLU()

(2): MaxPool2d(kernel\_size=2, stride=2, padding=0, dilation=1, ceil\_mode=False)

)

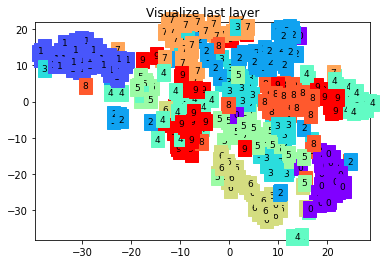
(out): Linear(in\_features=1568, out\_features=10, bias=True)

)

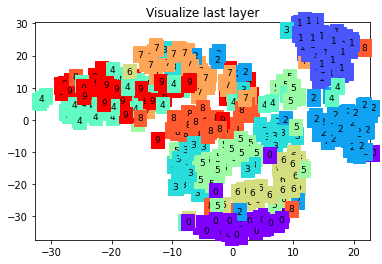
<ipython-input-49-ad16757697b6>:65: UserWarning: nn.init.xavier\_uniform is now deprecated in favor of nn.init.xavier\_uniform\_.

torch.nn.init.xavier\_uniform(self.out.weight) #1-7

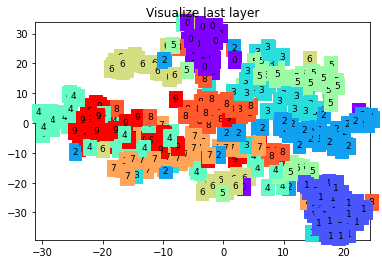
Epoch: 0 | train loss: 2.3213 | test accuracy: 0.10



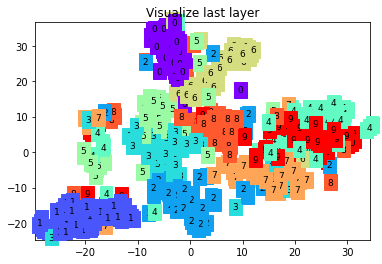
Epoch: 0 | train loss: 2.3319 | test accuracy: 0.10



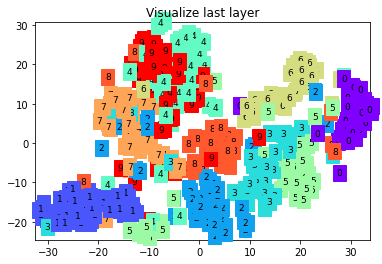
Epoch: 0 | train loss: 2.2720 | test accuracy: 0.10



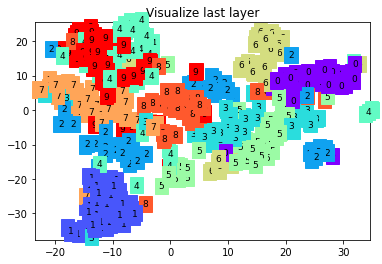
Epoch: 0 | train loss: 2.2824 | test accuracy: 0.11



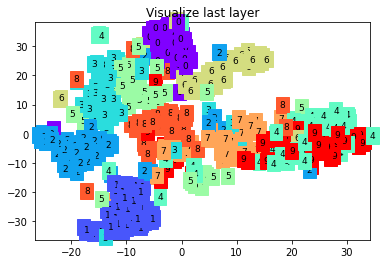
Epoch: 0 | train loss: 2.3246 | test accuracy: 0.12



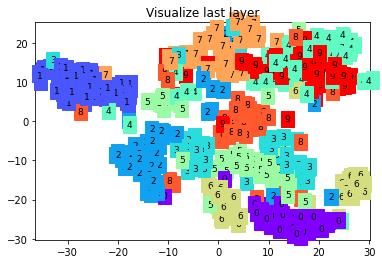
Epoch: 0 | train loss: 2.2991 | test accuracy: 0.16



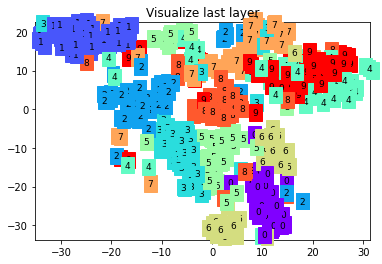
Epoch: 0 | train loss: 2.2431 | test accuracy: 0.21



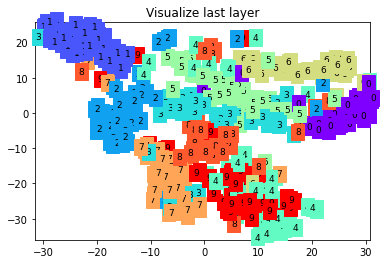
Epoch: 0 | train loss: 2.2772 | test accuracy: 0.25



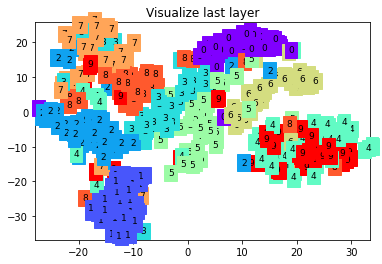
Epoch: 0 | train loss: 2.2023 | test accuracy: 0.28



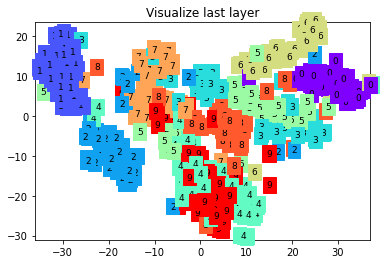
Epoch: 0 | train loss: 2.2551 | test accuracy: 0.31



Epoch: 0 | train loss: 2.2684 | test accuracy: 0.33



Epoch: 0 | train loss: 2.1974 | test accuracy: 0.36



1. Choose ONE other parameters of CNN program (e.g. number of hidden nodes, epochs, batch normalization, etc). Change the value of this parameter and compare the results.

<Program Code>



<Result>

torch.Size([60000, 28, 28])

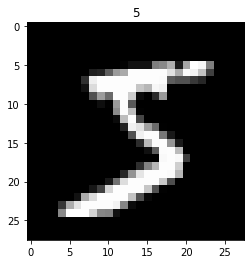
torch.Size([60000])

c:\users\hhshi\appdata\local\programs\python\python39\lib\site-packages\torchvision\datasets\mnist.py:64: UserWarning: train\_data has been renamed data

warnings.warn("train\_data has been renamed data")

c:\users\hhshi\appdata\local\programs\python\python39\lib\site-packages\torchvision\datasets\mnist.py:54: UserWarning: train\_labels has been renamed targets

warnings.warn("train\_labels has been renamed targets")



c:\users\hhshi\appdata\local\programs\python\python39\lib\site-packages\torchvision\datasets\mnist.py:69: UserWarning: test\_data has been renamed data

warnings.warn("test\_data has been renamed data")

torch.Size([28, 28])

torch.Size([1, 28, 28])

tensor(7)

CNN(

(conv1): Sequential(

(0): Conv2d(1, 16, kernel\_size=(5, 5), stride=(1, 1), padding=(2, 2))

(1): ReLU()

(2): MaxPool2d(kernel\_size=2, stride=2, padding=0, dilation=1, ceil\_mode=False)

)

(conv2): Sequential(

(0): Conv2d(16, 32, kernel\_size=(5, 5), stride=(1, 1), padding=(2, 2))

(1): ReLU()

(2): MaxPool2d(kernel\_size=2, stride=2, padding=0, dilation=1, ceil\_mode=False)

)

(out): Linear(in\_features=1568, out\_features=10, bias=True)

)

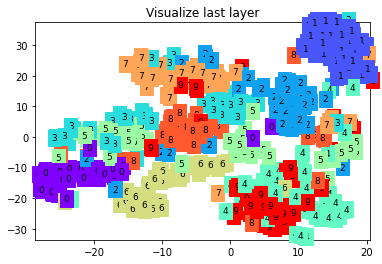
c:\users\hhshi\appdata\local\programs\python\python39\lib\site-packages\torchvision\datasets\mnist.py:59: UserWarning: test\_labels has been renamed targets

warnings.warn("test\_labels has been renamed targets")

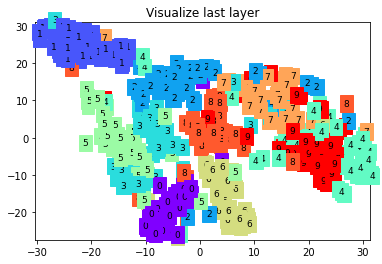
<ipython-input-50-d6b2cef14fcb>:65: UserWarning: nn.init.xavier\_uniform is now deprecated in favor of nn.init.xavier\_uniform\_.

torch.nn.init.xavier\_uniform(self.out.weight) #1-7

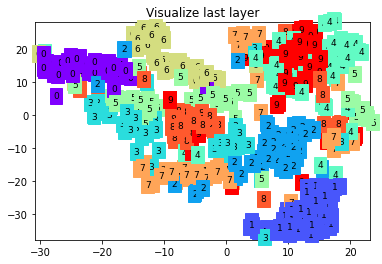
Epoch: 0 | train loss: 2.3213 | test accuracy: 0.10



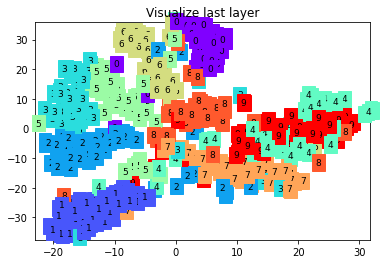
Epoch: 0 | train loss: 2.3319 | test accuracy: 0.10



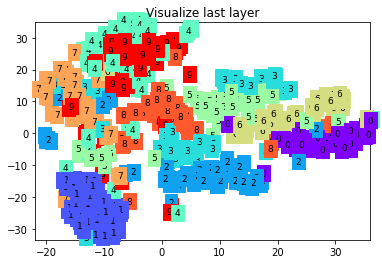
Epoch: 0 | train loss: 2.2720 | test accuracy: 0.10



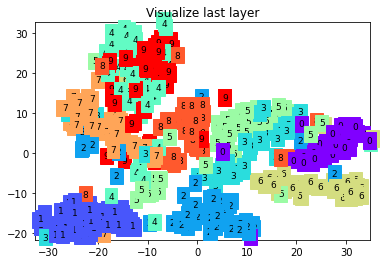
Epoch: 0 | train loss: 2.2824 | test accuracy: 0.11



Epoch: 0 | train loss: 2.3246 | test accuracy: 0.12



Epoch: 0 | train loss: 2.2991 | test accuracy: 0.16



1. (\*optional\*) choose Adam optimization method and use L2 (ridge) regularization method this time. You can do so by setting ‘weight\_decay’ value in optimization method (torch.optim.Adam) to a tiny number (e.g. 1e-5). Compare the results of using regularization.

<Code>



<Result>

torch.Size([60000, 28, 28])

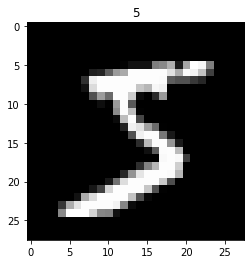
torch.Size([60000])

c:\users\hhshi\appdata\local\programs\python\python39\lib\site-packages\torchvision\datasets\mnist.py:64: UserWarning: train\_data has been renamed data

warnings.warn("train\_data has been renamed data")

c:\users\hhshi\appdata\local\programs\python\python39\lib\site-packages\torchvision\datasets\mnist.py:54: UserWarning: train\_labels has been renamed targets

warnings.warn("train\_labels has been renamed targets")



c:\users\hhshi\appdata\local\programs\python\python39\lib\site-packages\torchvision\datasets\mnist.py:69: UserWarning: test\_data has been renamed data

warnings.warn("test\_data has been renamed data")

c:\users\hhshi\appdata\local\programs\python\python39\lib\site-packages\torchvision\datasets\mnist.py:59: UserWarning: test\_labels has been renamed targets

warnings.warn("test\_labels has been renamed targets")

torch.Size([28, 28])

torch.Size([1, 28, 28])

tensor(7)

CNN(

(conv1): Sequential(

(0): Conv2d(1, 16, kernel\_size=(5, 5), stride=(1, 1), padding=(2, 2))

(1): ReLU()

(2): MaxPool2d(kernel\_size=2, stride=2, padding=0, dilation=1, ceil\_mode=False)

)

(conv2): Sequential(

(0): Conv2d(16, 32, kernel\_size=(5, 5), stride=(1, 1), padding=(2, 2))

(1): ReLU()

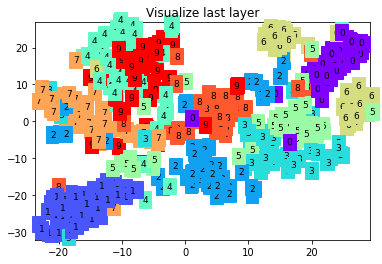
(2): MaxPool2d(kernel\_size=2, stride=2, padding=0, dilation=1, ceil\_mode=False)

)

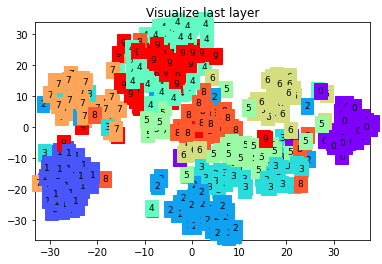
(out): Linear(in\_features=1568, out\_features=10, bias=True)

)

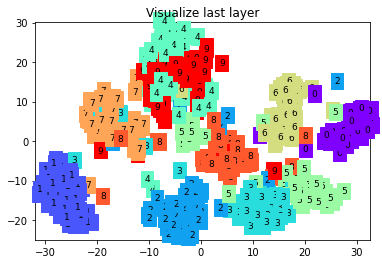
Epoch: 0 | train loss: 2.3125 | test accuracy: 0.18



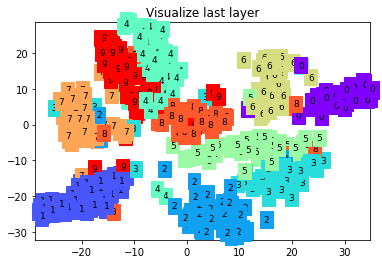
Epoch: 0 | train loss: 0.3734 | test accuracy: 0.90



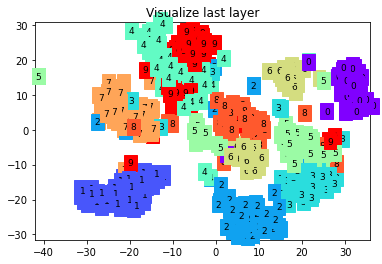
Epoch: 0 | train loss: 0.2327 | test accuracy: 0.93



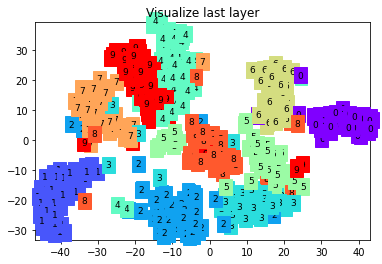
Epoch: 0 | train loss: 0.0571 | test accuracy: 0.95



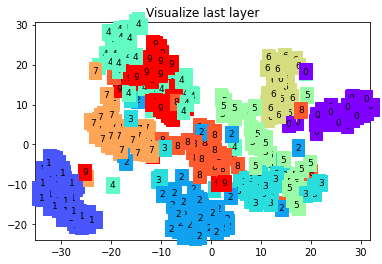
Epoch: 0 | train loss: 0.0704 | test accuracy: 0.96



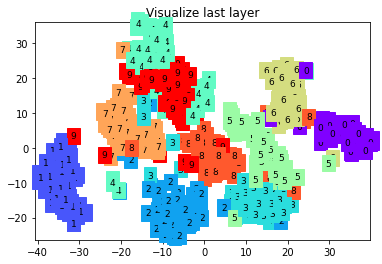
Epoch: 0 | train loss: 0.0842 | test accuracy: 0.96



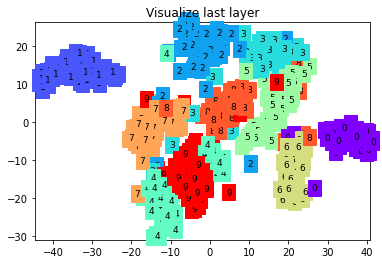
Epoch: 0 | train loss: 0.0389 | test accuracy: 0.96



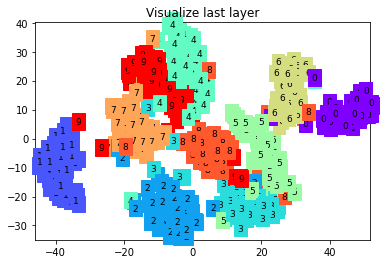
Epoch: 0 | train loss: 0.0193 | test accuracy: 0.96



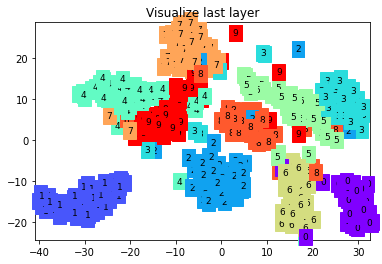
Epoch: 0 | train loss: 0.2377 | test accuracy: 0.96



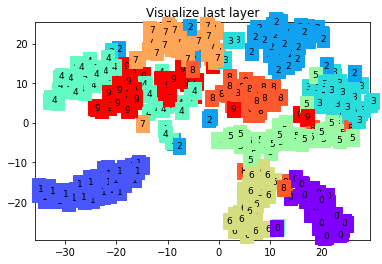
Epoch: 0 | train loss: 0.0447 | test accuracy: 0.97



Epoch: 0 | train loss: 0.0108 | test accuracy: 0.97



Epoch: 0 | train loss: 0.0247 | test accuracy: 0.97



1. CNN & CIFAR-10