# Ex9

June 22, 2021

## 1 Task 1: Classification

- define a Neural Network
- def

[1]:
```
import torch
import torch.nn as nn
import torch.nn.functional as F
import torch.optim as optim
import torchvision
from torchvision import datasets, transforms
import matplotlib.pyplot as plt
import numpy as np
from utils import NoisyFashionMNIST

%matplotlib inline
def show(img):
    npimg = img.numpy()
    plt.imshow(np.transpose(npimg, (1,2,0)), interpolation='nearest')
```

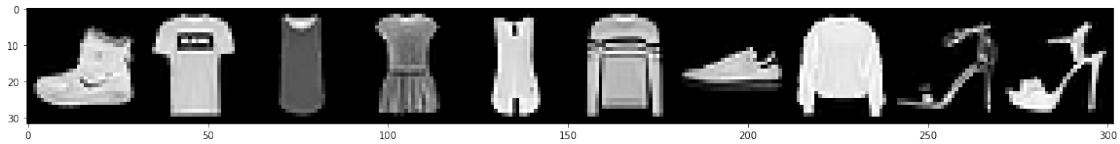
### 1.1 Dataset

[2]:
```
transform=transforms.Compose([
        transforms.ToTensor()])

train_dataset = datasets.FashionMNIST("./data", train = True, download=True,␣
 ↪transform=transform)
test_dataset = datasets.FashionMNIST("./data", train = False, download=True,␣
 ↪transform=transform)


idx_to_class = {v: k for k, v in train_dataset.class_to_idx.items()}
```

[3]:
```
x = [train_dataset[i][0] for i in range(10)]
labels = [idx_to_class[train_dataset[i][1]] for i in range(10)]
print(labels)

plt.figure(figsize=(20,10))
```

```
show(torchvision.utils.make_grid(x, nrow=10))
plt.show()
```

```
['Ankle boot', 'T-shirt/top', 'T-shirt/top', 'Dress', 'T-shirt/top', 'Pullover',
'Sneaker', 'Pullover', 'Sandal', 'Sandal']
```




[4]:
```
# use cuda if available
device = 'cuda' if torch.cuda.is_available() else 'cpu'
print('Using {} device'.format(device))
```

```
Using cuda device
```

[5]:
```
# class defining neural network
# input layer is linear, takes 28*28 input parameters, outputs 512 uses ReLU
# 1 hidden layer input 512, output 512 uses ReLU()
# output layer, input 512, output 10, one for each class, uses ReLU()
# ReLU is widely used as it solves the problem of vanishing gradients
# nn uses bias term throughout network
class NeuralNetwork(nn.Module):
    def __init__(self):
        super(NeuralNetwork, self).__init__()
        self.flatten = nn.Flatten()
        self.linear_relu_stack = nn.Sequential(
            nn.Linear(28*28, 512),
            nn.ReLU(),
            nn.Linear(512, 512),
            nn.ReLU(),
            nn.Linear(512, 28*28),
            nn.ReLU()
        )

    def forward(self, x):
        x = self.flatten(x)
        logits = self.linear_relu_stack(x)
        return logits
```

We also tested some other activation functions such as sigmoid for the output layer, however, we achieved better results just using ReLU

[6]:
```
# obtain and print the model
# execute model on device --> cuda if available
model = NeuralNetwork().to(device)
print(model)
```

```
NeuralNetwork(
  (flatten): Flatten(start_dim=1, end_dim=-1)
  (linear_relu_stack): Sequential(
    (0): Linear(in_features=784, out_features=512, bias=True)
    (1): ReLU()
    (2): Linear(in_features=512, out_features=512, bias=True)
    (3): ReLU()
    (4): Linear(in_features=512, out_features=784, bias=True)
    (5): ReLU()
  )
)
```

[7]:
```
# function for training
# @param dataloader: the data loader for the training set
# @param model: the previously defined neural network model
# @param loss_fn: defined loss function
# @param optimizer: how we calculate the loss
def train_loop(dataloader, model, loss_fn, optimizer):
    size = len(dataloader.dataset)
    for batch, (X, y) in enumerate(dataloader):
        # move X and y to gpu if available
        X = X.to(device)
        y = y.to(device)
        # Compute prediction and loss
        pred = model(X)
        loss = loss_fn(pred, y)

        # Backpropagation
        optimizer.zero_grad()
        loss.backward()
        optimizer.step()

        if batch % 100 == 0:
            loss, current = loss.item(), batch * len(X)
            print(f"loss: {loss:>7f}  [{current:>5d}/{size:>5d}]")

# function for testing
# @param dataloader: the data loader for the training set
# @param model: the previously defined neural network model after training
# @param loss_fn: defined loss function
def test_loop(dataloader, model, loss_fn):
    size = len(dataloader.dataset)
```

```
    test_loss, correct = 0, 0

    with torch.no_grad():
        for X, y in dataloader:
            # move X and y to gpu if available
            X = X.to(device)
            y = y.to(device)
            pred = model(X)
            test_loss += loss_fn(pred, y).item()
            correct += (pred.argmax(1) == y).type(torch.float).sum().item()

    test_loss /= size
    correct /= size
    print(f"Test Error: \n Accuracy: {(100*correct):>0.1f}%, Avg loss:␣
 ↪{test_loss:>8f} \n")
```

[8]:
```
# define hyper params
# updating rate for each batch/epoch
# empirical evaluation found that 1e-3 works better than other values
learning_rate = 1e-3
# number of data samples to propagate through network before updating params
batch_size = 64
# iteration times
# We tested with different epoch sizes, however, due to the pdf form, we␣
 ↪decided to
# stay at 50 to not have a really large pdf
epochs = 50
# using CrossEntropy for numeric stability, combines LogSoftmax and NLLLoss
# useful when training a problem with C classes
loss_fn = nn.CrossEntropyLoss()
# defining the optimizer
optimizer = torch.optim.SGD(model.parameters(), lr=learning_rate)

# use data loader to obtain batch training and test data
train_loader = torch.utils.data.DataLoader(train_dataset, batch_size = 20,␣
 ↪shuffle=True)
test_loader = torch.utils.data.DataLoader(test_dataset, batch_size = 20,␣
 ↪shuffle=True)


for t in range(epochs):
    print(f"Epoch {t+1}\n-------------------------------")
    train_loop(train_loader, model, loss_fn, optimizer)
    test_loop(test_loader, model, loss_fn)
print("Done!")
```

```
Epoch 1
```

```
-------------------------------
loss: 6.659163  [    0/60000]
loss: 6.634620  [ 2000/60000]
loss: 6.604107  [ 4000/60000]
loss: 6.500693  [ 6000/60000]
loss: 6.501466  [ 8000/60000]
loss: 6.285739  [10000/60000]
loss: 6.181007  [12000/60000]
loss: 6.048857  [14000/60000]
loss: 5.470643  [16000/60000]
loss: 3.688171  [18000/60000]
loss: 4.683969  [20000/60000]
loss: 3.433100  [22000/60000]
loss: 3.865809  [24000/60000]
loss: 4.424966  [26000/60000]
loss: 3.915787  [28000/60000]
loss: 4.042671  [30000/60000]
loss: 3.264639  [32000/60000]
loss: 3.542874  [34000/60000]
loss: 4.438230  [36000/60000]
loss: 3.105624  [38000/60000]
loss: 4.270744  [40000/60000]
loss: 2.936469  [42000/60000]
loss: 3.731812  [44000/60000]
loss: 3.576220  [46000/60000]
loss: 3.143201  [48000/60000]
loss: 3.769495  [50000/60000]
loss: 2.770670  [52000/60000]
loss: 3.665986  [54000/60000]
loss: 4.063390  [56000/60000]
loss: 4.017674  [58000/60000]
Test Error:
 Accuracy: 37.7%, Avg loss: 0.182739

Epoch 2
-------------------------------
loss: 4.685852  [    0/60000]
loss: 4.417016  [ 2000/60000]
loss: 2.793880  [ 4000/60000]
loss: 4.996428  [ 6000/60000]
loss: 4.294638  [ 8000/60000]
loss: 3.010256  [10000/60000]
loss: 3.161332  [12000/60000]
loss: 5.036667  [14000/60000]
loss: 3.260948  [16000/60000]
loss: 3.643913  [18000/60000]
loss: 2.514807  [20000/60000]
loss: 2.422271  [22000/60000]
```

```
loss: 2.879000  [24000/60000]
loss: 3.661248  [26000/60000]
loss: 3.055619  [28000/60000]
loss: 3.098306  [30000/60000]
loss: 2.917189  [32000/60000]
loss: 4.337924  [34000/60000]
loss: 3.255198  [36000/60000]
loss: 2.893723  [38000/60000]
loss: 3.799011  [40000/60000]
loss: 2.672043  [42000/60000]
loss: 3.074779  [44000/60000]
loss: 4.114913  [46000/60000]
loss: 3.122439  [48000/60000]
loss: 2.245150  [50000/60000]
loss: 3.437981  [52000/60000]
loss: 2.459142  [54000/60000]
loss: 2.295113  [56000/60000]
loss: 2.538028  [58000/60000]
Test Error:
 Accuracy: 49.2%, Avg loss: 0.158911

Epoch 3
-------------------------------
loss: 3.468908  [    0/60000]
loss: 2.444246  [ 2000/60000]
loss: 3.369915  [ 4000/60000]
loss: 3.340917  [ 6000/60000]
loss: 3.941503  [ 8000/60000]
loss: 3.341870  [10000/60000]
loss: 2.492937  [12000/60000]
loss: 2.197646  [14000/60000]
loss: 3.120433  [16000/60000]
loss: 2.312072  [18000/60000]
loss: 3.233260  [20000/60000]
loss: 2.803998  [22000/60000]
loss: 3.754359  [24000/60000]
loss: 3.544811  [26000/60000]
loss: 2.995533  [28000/60000]
loss: 2.563748  [30000/60000]
loss: 3.804260  [32000/60000]
loss: 2.757863  [34000/60000]
loss: 3.205398  [36000/60000]
loss: 3.746204  [38000/60000]
loss: 5.157423  [40000/60000]
loss: 2.317360  [42000/60000]
loss: 3.225624  [44000/60000]
loss: 3.496303  [46000/60000]
loss: 3.184761  [48000/60000]
```

```
loss: 2.767407  [50000/60000]
loss: 3.208905  [52000/60000]
loss: 2.966935  [54000/60000]
loss: 2.397077  [56000/60000]
loss: 3.354623  [58000/60000]
Test Error:
 Accuracy: 51.4%, Avg loss: 0.147699

Epoch 4
-------------------------------
loss: 2.921476  [    0/60000]
loss: 2.693045  [ 2000/60000]
loss: 2.309947  [ 4000/60000]
loss: 3.942859  [ 6000/60000]
loss: 3.172477  [ 8000/60000]
loss: 2.195727  [10000/60000]
loss: 2.189579  [12000/60000]
loss: 1.348711  [14000/60000]
loss: 2.771563  [16000/60000]
loss: 3.647937  [18000/60000]
loss: 2.266284  [20000/60000]
loss: 3.360425  [22000/60000]
loss: 3.511155  [24000/60000]
loss: 4.325378  [26000/60000]
loss: 1.310665  [28000/60000]
loss: 2.536285  [30000/60000]
loss: 1.916466  [32000/60000]
loss: 1.926596  [34000/60000]
loss: 1.726348  [36000/60000]
loss: 2.354888  [38000/60000]
loss: 1.837492  [40000/60000]
loss: 1.488634  [42000/60000]
loss: 2.044516  [44000/60000]
loss: 3.203298  [46000/60000]
loss: 1.982810  [48000/60000]
loss: 2.569319  [50000/60000]
loss: 3.150255  [52000/60000]
loss: 1.307605  [54000/60000]
loss: 1.841227  [56000/60000]
loss: 1.592426  [58000/60000]
Test Error:
 Accuracy: 67.9%, Avg loss: 0.079563

Epoch 5
-------------------------------
loss: 1.610508  [    0/60000]
loss: 1.386135  [ 2000/60000]
loss: 2.021617  [ 4000/60000]
```

```
loss: 1.661016  [ 6000/60000]
loss: 1.957520  [ 8000/60000]
loss: 1.234498  [10000/60000]
loss: 1.960788  [12000/60000]
loss: 1.031879  [14000/60000]
loss: 2.080177  [16000/60000]
loss: 2.640392  [18000/60000]
loss: 1.560687  [20000/60000]
loss: 2.088563  [22000/60000]
loss: 2.402857  [24000/60000]
loss: 1.055825  [26000/60000]
loss: 1.256927  [28000/60000]
loss: 1.838809  [30000/60000]
loss: 1.786149  [32000/60000]
loss: 1.498445  [34000/60000]
loss: 1.150294  [36000/60000]
loss: 0.996086  [38000/60000]
loss: 1.411007  [40000/60000]
loss: 2.463927  [42000/60000]
loss: 1.982298  [44000/60000]
loss: 0.533722  [46000/60000]
loss: 1.140635  [48000/60000]
loss: 0.604442  [50000/60000]
loss: 1.633607  [52000/60000]
loss: 1.008184  [54000/60000]
loss: 1.685809  [56000/60000]
loss: 0.940951  [58000/60000]
Test Error:
 Accuracy: 68.8%, Avg loss: 0.070477

Epoch 6
-------------------------------
loss: 1.352678  [    0/60000]
loss: 1.774660  [ 2000/60000]
loss: 2.022779  [ 4000/60000]
loss: 1.151147  [ 6000/60000]
loss: 1.394450  [ 8000/60000]
loss: 1.003740  [10000/60000]
loss: 1.237940  [12000/60000]
loss: 1.292591  [14000/60000]
loss: 0.505913  [16000/60000]
loss: 1.548932  [18000/60000]
loss: 0.905898  [20000/60000]
loss: 0.734348  [22000/60000]
loss: 1.693594  [24000/60000]
loss: 0.607318  [26000/60000]
loss: 0.921061  [28000/60000]
loss: 1.283383  [30000/60000]
```

```
loss: 1.556962  [32000/60000]
loss: 1.588507  [34000/60000]
loss: 0.887730  [36000/60000]
loss: 0.774513  [38000/60000]
loss: 1.461372  [40000/60000]
loss: 0.736030  [42000/60000]
loss: 1.772682  [44000/60000]
loss: 1.352482  [46000/60000]
loss: 2.640409  [48000/60000]
loss: 1.527078  [50000/60000]
loss: 1.268715  [52000/60000]
loss: 1.263400  [54000/60000]
loss: 1.670193  [56000/60000]
loss: 1.851696  [58000/60000]
Test Error:
 Accuracy: 70.5%, Avg loss: 0.066652

Epoch 7
-------------------------------
loss: 0.903608  [    0/60000]
loss: 1.540961  [ 2000/60000]
loss: 0.611077  [ 4000/60000]
loss: 1.843069  [ 6000/60000]
loss: 1.627776  [ 8000/60000]
loss: 0.846823  [10000/60000]
loss: 1.480331  [12000/60000]
loss: 1.348107  [14000/60000]
loss: 1.625764  [16000/60000]
loss: 1.350903  [18000/60000]
loss: 1.397542  [20000/60000]
loss: 1.316731  [22000/60000]
loss: 1.122718  [24000/60000]
loss: 0.903823  [26000/60000]
loss: 1.899338  [28000/60000]
loss: 0.891912  [30000/60000]
loss: 1.004997  [32000/60000]
loss: 0.860628  [34000/60000]
loss: 2.341786  [36000/60000]
loss: 0.928937  [38000/60000]
loss: 2.118868  [40000/60000]
loss: 1.101820  [42000/60000]
loss: 0.973972  [44000/60000]
loss: 2.158295  [46000/60000]
loss: 1.284381  [48000/60000]
loss: 1.905996  [50000/60000]
loss: 1.219850  [52000/60000]
loss: 1.462900  [54000/60000]
loss: 1.360865  [56000/60000]
```

```
loss: 0.739559  [58000/60000]
Test Error:
 Accuracy: 71.2%, Avg loss: 0.064673

Epoch 8
-------------------------------
loss: 1.809494  [    0/60000]
loss: 0.767017  [ 2000/60000]
loss: 1.376169  [ 4000/60000]
loss: 1.958833  [ 6000/60000]
loss: 1.103237  [ 8000/60000]
loss: 1.189915  [10000/60000]
loss: 1.441446  [12000/60000]
loss: 1.221204  [14000/60000]
loss: 1.189496  [16000/60000]
loss: 2.954329  [18000/60000]
loss: 1.251656  [20000/60000]
loss: 1.722936  [22000/60000]
loss: 1.835952  [24000/60000]
loss: 0.727421  [26000/60000]
loss: 1.246783  [28000/60000]
loss: 0.361259  [30000/60000]
loss: 1.389363  [32000/60000]
loss: 1.014249  [34000/60000]
loss: 0.505057  [36000/60000]
loss: 1.846012  [38000/60000]
loss: 1.461501  [40000/60000]
loss: 1.007898  [42000/60000]
loss: 1.529718  [44000/60000]
loss: 1.437172  [46000/60000]
loss: 0.618914  [48000/60000]
loss: 1.470695  [50000/60000]
loss: 0.765718  [52000/60000]
loss: 1.531708  [54000/60000]
loss: 0.443997  [56000/60000]
loss: 1.770550  [58000/60000]
Test Error:
 Accuracy: 71.3%, Avg loss: 0.063286

Epoch 9
-------------------------------
loss: 1.699882  [    0/60000]
loss: 0.997431  [ 2000/60000]
loss: 0.751554  [ 4000/60000]
loss: 0.722580  [ 6000/60000]
loss: 1.282469  [ 8000/60000]
loss: 1.497511  [10000/60000]
loss: 1.661094  [12000/60000]
```

```
loss: 0.968048  [14000/60000]
loss: 2.045938  [16000/60000]
loss: 1.697475  [18000/60000]
loss: 1.259605  [20000/60000]
loss: 1.630184  [22000/60000]
loss: 1.104853  [24000/60000]
loss: 1.057830  [26000/60000]
loss: 0.711312  [28000/60000]
loss: 1.676739  [30000/60000]
loss: 1.329903  [32000/60000]
loss: 0.592808  [34000/60000]
loss: 0.550494  [36000/60000]
loss: 1.048864  [38000/60000]
loss: 1.016789  [40000/60000]
loss: 0.766682  [42000/60000]
loss: 0.686866  [44000/60000]
loss: 0.676689  [46000/60000]
loss: 0.731731  [48000/60000]
loss: 0.301070  [50000/60000]
loss: 0.443153  [52000/60000]
loss: 0.909241  [54000/60000]
loss: 0.863136  [56000/60000]
loss: 0.331706  [58000/60000]
Test Error:
 Accuracy: 79.1%, Avg loss: 0.030398

Epoch 10
-------------------------------
loss: 0.417514  [    0/60000]
loss: 0.578139  [ 2000/60000]
loss: 0.420469  [ 4000/60000]
loss: 0.659393  [ 6000/60000]
loss: 0.586010  [ 8000/60000]
loss: 0.578119  [10000/60000]
loss: 0.971575  [12000/60000]
loss: 0.550236  [14000/60000]
loss: 0.504797  [16000/60000]
loss: 0.859542  [18000/60000]
loss: 0.658412  [20000/60000]
loss: 0.620058  [22000/60000]
loss: 0.582601  [24000/60000]
loss: 0.488871  [26000/60000]
loss: 0.764865  [28000/60000]
loss: 0.555597  [30000/60000]
loss: 0.635234  [32000/60000]
loss: 0.532016  [34000/60000]
loss: 0.563043  [36000/60000]
loss: 0.294142  [38000/60000]
```

```
loss: 0.422371  [40000/60000]
loss: 0.828342  [42000/60000]
loss: 0.377879  [44000/60000]
loss: 0.709101  [46000/60000]
loss: 0.324084  [48000/60000]
loss: 0.582540  [50000/60000]
loss: 0.450960  [52000/60000]
loss: 0.487268  [54000/60000]
loss: 0.411693  [56000/60000]
loss: 0.403189  [58000/60000]
Test Error:
 Accuracy: 79.7%, Avg loss: 0.028681

Epoch 11
-------------------------------
loss: 0.514471  [    0/60000]
loss: 0.647337  [ 2000/60000]
loss: 0.445183  [ 4000/60000]
loss: 0.454401  [ 6000/60000]
loss: 0.456248  [ 8000/60000]
loss: 0.494867  [10000/60000]
loss: 0.637364  [12000/60000]
loss: 0.451287  [14000/60000]
loss: 0.798879  [16000/60000]
loss: 0.500858  [18000/60000]
loss: 0.536289  [20000/60000]
loss: 0.907154  [22000/60000]
loss: 0.516603  [24000/60000]
loss: 0.499381  [26000/60000]
loss: 0.446181  [28000/60000]
loss: 0.304153  [30000/60000]
loss: 0.393301  [32000/60000]
loss: 0.677752  [34000/60000]
loss: 0.267985  [36000/60000]
loss: 0.595498  [38000/60000]
loss: 0.575077  [40000/60000]
loss: 0.484349  [42000/60000]
loss: 0.607485  [44000/60000]
loss: 0.453379  [46000/60000]
loss: 0.534378  [48000/60000]
loss: 0.269032  [50000/60000]
loss: 0.538898  [52000/60000]
loss: 0.194856  [54000/60000]
loss: 0.635707  [56000/60000]
loss: 0.403146  [58000/60000]
Test Error:
 Accuracy: 80.9%, Avg loss: 0.026928
```

```
Epoch 12
-------------------------------
loss: 0.308463  [    0/60000]
loss: 0.480179  [ 2000/60000]
loss: 0.579240  [ 4000/60000]
loss: 0.448752  [ 6000/60000]
loss: 0.536774  [ 8000/60000]
loss: 0.691817  [10000/60000]
loss: 0.663361  [12000/60000]
loss: 0.329575  [14000/60000]
loss: 0.765114  [16000/60000]
loss: 0.639854  [18000/60000]
loss: 0.341706  [20000/60000]
loss: 0.534562  [22000/60000]
loss: 0.233914  [24000/60000]
loss: 0.383646  [26000/60000]
loss: 0.574317  [28000/60000]
loss: 0.591364  [30000/60000]
loss: 0.576217  [32000/60000]
loss: 0.479694  [34000/60000]
loss: 0.651666  [36000/60000]
loss: 0.472085  [38000/60000]
loss: 0.553525  [40000/60000]
loss: 0.498700  [42000/60000]
loss: 0.261596  [44000/60000]
loss: 0.576181  [46000/60000]
loss: 0.398138  [48000/60000]
loss: 0.398030  [50000/60000]
loss: 0.591326  [52000/60000]
loss: 0.519146  [54000/60000]
loss: 0.277123  [56000/60000]
loss: 0.449934  [58000/60000]
Test Error:
 Accuracy: 81.0%, Avg loss: 0.026359

Epoch 13
-------------------------------
loss: 0.545991  [    0/60000]
loss: 0.185759  [ 2000/60000]
loss: 0.908423  [ 4000/60000]
loss: 0.443340  [ 6000/60000]
loss: 0.475143  [ 8000/60000]
loss: 0.401196  [10000/60000]
loss: 0.301060  [12000/60000]
loss: 0.624574  [14000/60000]
loss: 0.519346  [16000/60000]
loss: 0.651235  [18000/60000]
loss: 0.765740  [20000/60000]
```

```
loss: 0.308110  [22000/60000]
loss: 0.408021  [24000/60000]
loss: 0.567086  [26000/60000]
loss: 0.732476  [28000/60000]
loss: 0.409096  [30000/60000]
loss: 0.676318  [32000/60000]
loss: 0.526769  [34000/60000]
loss: 0.398010  [36000/60000]
loss: 0.506954  [38000/60000]
loss: 0.484482  [40000/60000]
loss: 0.540755  [42000/60000]
loss: 0.356620  [44000/60000]
loss: 0.614389  [46000/60000]
loss: 0.467455  [48000/60000]
loss: 0.311631  [50000/60000]
loss: 0.545393  [52000/60000]
loss: 0.408112  [54000/60000]
loss: 0.524086  [56000/60000]
loss: 0.513635  [58000/60000]
Test Error:
 Accuracy: 81.4%, Avg loss: 0.025848

Epoch 14
-------------------------------
loss: 0.327068  [    0/60000]
loss: 0.511139  [ 2000/60000]
loss: 0.494225  [ 4000/60000]
loss: 0.600983  [ 6000/60000]
loss: 0.517713  [ 8000/60000]
loss: 0.771407  [10000/60000]
loss: 0.676152  [12000/60000]
loss: 0.546794  [14000/60000]
loss: 0.327251  [16000/60000]
loss: 0.552048  [18000/60000]
loss: 0.695032  [20000/60000]
loss: 0.301325  [22000/60000]
loss: 0.321873  [24000/60000]
loss: 0.633078  [26000/60000]
loss: 0.331901  [28000/60000]
loss: 0.608378  [30000/60000]
loss: 0.357368  [32000/60000]
loss: 1.023990  [34000/60000]
loss: 0.424955  [36000/60000]
loss: 0.998428  [38000/60000]
loss: 0.544664  [40000/60000]
loss: 0.425959  [42000/60000]
loss: 0.317561  [44000/60000]
loss: 0.519035  [46000/60000]
```

```
loss: 0.471822  [48000/60000]
loss: 0.630540  [50000/60000]
loss: 0.345888  [52000/60000]
loss: 0.295669  [54000/60000]
loss: 0.431723  [56000/60000]
loss: 0.369320  [58000/60000]
Test Error:
 Accuracy: 81.8%, Avg loss: 0.025450

Epoch 15
-------------------------------
loss: 0.366451  [    0/60000]
loss: 0.613301  [ 2000/60000]
loss: 0.484831  [ 4000/60000]
loss: 0.539231  [ 6000/60000]
loss: 0.522514  [ 8000/60000]
loss: 0.381062  [10000/60000]
loss: 0.320783  [12000/60000]
loss: 0.657533  [14000/60000]
loss: 0.278460  [16000/60000]
loss: 0.186041  [18000/60000]
loss: 0.306383  [20000/60000]
loss: 0.705173  [22000/60000]
loss: 0.457502  [24000/60000]
loss: 0.279727  [26000/60000]
loss: 0.347057  [28000/60000]
loss: 0.282076  [30000/60000]
loss: 0.580039  [32000/60000]
loss: 0.373644  [34000/60000]
loss: 0.521778  [36000/60000]
loss: 0.571253  [38000/60000]
loss: 0.479007  [40000/60000]
loss: 0.438458  [42000/60000]
loss: 0.496105  [44000/60000]
loss: 0.325542  [46000/60000]
loss: 0.253039  [48000/60000]
loss: 0.646575  [50000/60000]
loss: 0.507299  [52000/60000]
loss: 0.276004  [54000/60000]
loss: 0.829070  [56000/60000]
loss: 0.434837  [58000/60000]
Test Error:
 Accuracy: 81.9%, Avg loss: 0.025090

Epoch 16
-------------------------------
loss: 0.821456  [    0/60000]
loss: 0.558854  [ 2000/60000]
```

```
loss: 0.764975  [ 4000/60000]
loss: 0.314048  [ 6000/60000]
loss: 0.285533  [ 8000/60000]
loss: 0.701928  [10000/60000]
loss: 0.207781  [12000/60000]
loss: 0.538243  [14000/60000]
loss: 0.557073  [16000/60000]
loss: 0.235158  [18000/60000]
loss: 1.258302  [20000/60000]
loss: 0.319974  [22000/60000]
loss: 0.282701  [24000/60000]
loss: 0.579575  [26000/60000]
loss: 0.300805  [28000/60000]
loss: 0.463270  [30000/60000]
loss: 0.589668  [32000/60000]
loss: 0.566088  [34000/60000]
loss: 0.500857  [36000/60000]
loss: 0.185012  [38000/60000]
loss: 0.616291  [40000/60000]
loss: 0.373380  [42000/60000]
loss: 0.519098  [44000/60000]
loss: 0.801677  [46000/60000]
loss: 0.579322  [48000/60000]
loss: 0.492157  [50000/60000]
loss: 0.269469  [52000/60000]
loss: 0.370691  [54000/60000]
loss: 0.766141  [56000/60000]
loss: 0.432741  [58000/60000]
Test Error:
 Accuracy: 82.1%, Avg loss: 0.024863

Epoch 17
-------------------------------
loss: 0.367931  [    0/60000]
loss: 0.442729  [ 2000/60000]
loss: 0.386182  [ 4000/60000]
loss: 0.416819  [ 6000/60000]
loss: 0.272124  [ 8000/60000]
loss: 0.350277  [10000/60000]
loss: 0.300819  [12000/60000]
loss: 0.357076  [14000/60000]
loss: 0.368194  [16000/60000]
loss: 0.632234  [18000/60000]
loss: 0.839941  [20000/60000]
loss: 0.970526  [22000/60000]
loss: 0.359606  [24000/60000]
loss: 0.359596  [26000/60000]
loss: 0.444325  [28000/60000]
```

```
loss: 0.323579  [30000/60000]
loss: 0.633728  [32000/60000]
loss: 0.412980  [34000/60000]
loss: 0.342917  [36000/60000]
loss: 0.326689  [38000/60000]
loss: 0.556963  [40000/60000]
loss: 0.695602  [42000/60000]
loss: 0.543009  [44000/60000]
loss: 0.650551  [46000/60000]
loss: 0.559271  [48000/60000]
loss: 0.324879  [50000/60000]
loss: 0.290214  [52000/60000]
loss: 0.272712  [54000/60000]
loss: 0.296236  [56000/60000]
loss: 0.562023  [58000/60000]
Test Error:
 Accuracy: 82.3%, Avg loss: 0.024696

Epoch 18
-------------------------------
loss: 0.794506  [    0/60000]
loss: 0.578484  [ 2000/60000]
loss: 0.446716  [ 4000/60000]
loss: 0.558437  [ 6000/60000]
loss: 0.466792  [ 8000/60000]
loss: 1.072482  [10000/60000]
loss: 0.416468  [12000/60000]
loss: 0.369850  [14000/60000]
loss: 0.424504  [16000/60000]
loss: 0.837162  [18000/60000]
loss: 0.260128  [20000/60000]
loss: 0.446862  [22000/60000]
loss: 0.456297  [24000/60000]
loss: 0.715349  [26000/60000]
loss: 0.218237  [28000/60000]
loss: 0.187034  [30000/60000]
loss: 0.485019  [32000/60000]
loss: 0.147568  [34000/60000]
loss: 0.407512  [36000/60000]
loss: 0.159754  [38000/60000]
loss: 0.576079  [40000/60000]
loss: 0.245402  [42000/60000]
loss: 0.752368  [44000/60000]
loss: 0.184519  [46000/60000]
loss: 0.350219  [48000/60000]
loss: 0.550156  [50000/60000]
loss: 0.146192  [52000/60000]
loss: 0.137929  [54000/60000]
```

```
loss: 0.315584  [56000/60000]
loss: 0.656634  [58000/60000]
Test Error:
 Accuracy: 82.2%, Avg loss: 0.024444

Epoch 19
-------------------------------
loss: 0.400612  [    0/60000]
loss: 0.399072  [ 2000/60000]
loss: 0.718432  [ 4000/60000]
loss: 0.260686  [ 6000/60000]
loss: 0.433173  [ 8000/60000]
loss: 0.212783  [10000/60000]
loss: 0.489535  [12000/60000]
loss: 0.635278  [14000/60000]
loss: 0.408390  [16000/60000]
loss: 0.613539  [18000/60000]
loss: 0.256765  [20000/60000]
loss: 0.652719  [22000/60000]
loss: 0.368266  [24000/60000]
loss: 0.565605  [26000/60000]
loss: 0.418680  [28000/60000]
loss: 0.268719  [30000/60000]
loss: 0.562465  [32000/60000]
loss: 0.293228  [34000/60000]
loss: 0.446637  [36000/60000]
loss: 0.560051  [38000/60000]
loss: 0.470871  [40000/60000]
loss: 0.298039  [42000/60000]
loss: 0.413555  [44000/60000]
loss: 0.410613  [46000/60000]
loss: 0.321063  [48000/60000]
loss: 0.291497  [50000/60000]
loss: 0.305111  [52000/60000]
loss: 0.372376  [54000/60000]
loss: 0.484115  [56000/60000]
loss: 0.313354  [58000/60000]
Test Error:
 Accuracy: 82.4%, Avg loss: 0.024323

Epoch 20
-------------------------------
loss: 0.708393  [    0/60000]
loss: 0.427134  [ 2000/60000]
loss: 0.718070  [ 4000/60000]
loss: 0.702810  [ 6000/60000]
loss: 0.378853  [ 8000/60000]
loss: 0.385039  [10000/60000]
```

```
loss: 0.387367  [12000/60000]
loss: 0.213991  [14000/60000]
loss: 0.343427  [16000/60000]
loss: 0.193605  [18000/60000]
loss: 0.296274  [20000/60000]
loss: 0.335871  [22000/60000]
loss: 0.531581  [24000/60000]
loss: 0.446219  [26000/60000]
loss: 0.358838  [28000/60000]
loss: 0.506640  [30000/60000]
loss: 0.372478  [32000/60000]
loss: 0.650899  [34000/60000]
loss: 0.351922  [36000/60000]
loss: 0.707902  [38000/60000]
loss: 0.572810  [40000/60000]
loss: 0.537572  [42000/60000]
loss: 0.265368  [44000/60000]
loss: 0.806479  [46000/60000]
loss: 0.189738  [48000/60000]
loss: 0.321943  [50000/60000]
loss: 0.460488  [52000/60000]
loss: 0.418275  [54000/60000]
loss: 0.231005  [56000/60000]
loss: 0.736386  [58000/60000]
Test Error:
 Accuracy: 82.7%, Avg loss: 0.024303

Epoch 21
-------------------------------
loss: 0.265574  [    0/60000]
loss: 0.496688  [ 2000/60000]
loss: 0.336331  [ 4000/60000]
loss: 0.385174  [ 6000/60000]
loss: 0.597944  [ 8000/60000]
loss: 0.311012  [10000/60000]
loss: 0.574866  [12000/60000]
loss: 0.393066  [14000/60000]
loss: 0.347843  [16000/60000]
loss: 0.576090  [18000/60000]
loss: 0.756715  [20000/60000]
loss: 0.168701  [22000/60000]
loss: 0.910966  [24000/60000]
loss: 0.453626  [26000/60000]
loss: 0.409163  [28000/60000]
loss: 0.319241  [30000/60000]
loss: 0.423435  [32000/60000]
loss: 0.374152  [34000/60000]
loss: 0.692995  [36000/60000]
```

```
loss: 0.458535  [38000/60000]
loss: 0.336049  [40000/60000]
loss: 0.498855  [42000/60000]
loss: 0.379962  [44000/60000]
loss: 0.268175  [46000/60000]
loss: 0.620157  [48000/60000]
loss: 0.961076  [50000/60000]
loss: 0.283377  [52000/60000]
loss: 0.396545  [54000/60000]
loss: 0.294484  [56000/60000]
loss: 0.287311  [58000/60000]
Test Error:
 Accuracy: 83.2%, Avg loss: 0.023673

Epoch 22
-------------------------------
loss: 0.313401  [    0/60000]
loss: 0.540704  [ 2000/60000]
loss: 0.270607  [ 4000/60000]
loss: 0.563021  [ 6000/60000]
loss: 0.869435  [ 8000/60000]
loss: 0.335853  [10000/60000]
loss: 0.770454  [12000/60000]
loss: 0.759857  [14000/60000]
loss: 0.378567  [16000/60000]
loss: 0.295045  [18000/60000]
loss: 0.673591  [20000/60000]
loss: 0.395221  [22000/60000]
loss: 0.308524  [24000/60000]
loss: 0.746887  [26000/60000]
loss: 0.456909  [28000/60000]
loss: 0.412710  [30000/60000]
loss: 0.352799  [32000/60000]
loss: 0.280713  [34000/60000]
loss: 0.414905  [36000/60000]
loss: 0.510313  [38000/60000]
loss: 0.110715  [40000/60000]
loss: 0.691948  [42000/60000]
loss: 0.729476  [44000/60000]
loss: 0.349746  [46000/60000]
loss: 0.240993  [48000/60000]
loss: 0.220569  [50000/60000]
loss: 0.341337  [52000/60000]
loss: 0.408379  [54000/60000]
loss: 0.720527  [56000/60000]
loss: 0.317904  [58000/60000]
Test Error:
 Accuracy: 83.5%, Avg loss: 0.023341
```

```
Epoch 23
-------------------------------
loss: 0.577251  [    0/60000]
loss: 0.411940  [ 2000/60000]
loss: 0.266194  [ 4000/60000]
loss: 0.541625  [ 6000/60000]
loss: 0.893436  [ 8000/60000]
loss: 0.426033  [10000/60000]
loss: 0.701208  [12000/60000]
loss: 0.398461  [14000/60000]
loss: 0.601084  [16000/60000]
loss: 0.629062  [18000/60000]
loss: 0.596812  [20000/60000]
loss: 1.021389  [22000/60000]
loss: 0.326255  [24000/60000]
loss: 0.283282  [26000/60000]
loss: 0.323964  [28000/60000]
loss: 0.751345  [30000/60000]
loss: 0.725712  [32000/60000]
loss: 0.247599  [34000/60000]
loss: 0.366887  [36000/60000]
loss: 0.385055  [38000/60000]
loss: 0.710139  [40000/60000]
loss: 0.331079  [42000/60000]
loss: 0.670560  [44000/60000]
loss: 0.374340  [46000/60000]
loss: 0.367473  [48000/60000]
loss: 0.196883  [50000/60000]
loss: 0.183368  [52000/60000]
loss: 0.332280  [54000/60000]
loss: 0.432793  [56000/60000]
loss: 0.199286  [58000/60000]
Test Error:
 Accuracy: 83.4%, Avg loss: 0.023212

Epoch 24
-------------------------------
loss: 0.302090  [    0/60000]
loss: 0.513892  [ 2000/60000]
loss: 0.346810  [ 4000/60000]
loss: 0.314754  [ 6000/60000]
loss: 0.673256  [ 8000/60000]
loss: 0.742183  [10000/60000]
loss: 0.368682  [12000/60000]
loss: 0.355731  [14000/60000]
loss: 0.555570  [16000/60000]
loss: 0.330246  [18000/60000]
```

```
loss: 0.203635  [20000/60000]
loss: 0.489226  [22000/60000]
loss: 0.560378  [24000/60000]
loss: 0.396795  [26000/60000]
loss: 0.284099  [28000/60000]
loss: 0.229002  [30000/60000]
loss: 0.829184  [32000/60000]
loss: 0.272116  [34000/60000]
loss: 0.478283  [36000/60000]
loss: 0.599197  [38000/60000]
loss: 0.548377  [40000/60000]
loss: 0.365689  [42000/60000]
loss: 0.198245  [44000/60000]
loss: 0.149113  [46000/60000]
loss: 0.462795  [48000/60000]
loss: 0.679903  [50000/60000]
loss: 0.456957  [52000/60000]
loss: 0.366381  [54000/60000]
loss: 0.348866  [56000/60000]
loss: 0.310377  [58000/60000]
Test Error:
 Accuracy: 83.6%, Avg loss: 0.022956

Epoch 25
-------------------------------
loss: 0.455795  [    0/60000]
loss: 0.899028  [ 2000/60000]
loss: 0.642447  [ 4000/60000]
loss: 0.208164  [ 6000/60000]
loss: 0.404077  [ 8000/60000]
loss: 0.562301  [10000/60000]
loss: 0.243904  [12000/60000]
loss: 0.931277  [14000/60000]
loss: 0.351438  [16000/60000]
loss: 0.240119  [18000/60000]
loss: 0.218891  [20000/60000]
loss: 0.557128  [22000/60000]
loss: 0.560854  [24000/60000]
loss: 0.845935  [26000/60000]
loss: 0.252310  [28000/60000]
loss: 0.277959  [30000/60000]
loss: 0.590519  [32000/60000]
loss: 0.219271  [34000/60000]
loss: 0.531998  [36000/60000]
loss: 0.268418  [38000/60000]
loss: 0.235886  [40000/60000]
loss: 0.173768  [42000/60000]
loss: 0.543642  [44000/60000]
```

```
loss: 0.419588  [46000/60000]
loss: 0.366731  [48000/60000]
loss: 0.360841  [50000/60000]
loss: 0.300287  [52000/60000]
loss: 0.280213  [54000/60000]
loss: 0.460975  [56000/60000]
loss: 0.419864  [58000/60000]
Test Error:
 Accuracy: 83.7%, Avg loss: 0.022937

Epoch 26
-------------------------------
loss: 0.329790  [    0/60000]
loss: 0.661417  [ 2000/60000]
loss: 0.307133  [ 4000/60000]
loss: 0.230443  [ 6000/60000]
loss: 0.243392  [ 8000/60000]
loss: 0.341349  [10000/60000]
loss: 0.279817  [12000/60000]
loss: 0.335640  [14000/60000]
loss: 0.274711  [16000/60000]
loss: 0.589418  [18000/60000]
loss: 0.302777  [20000/60000]
loss: 0.393761  [22000/60000]
loss: 0.375301  [24000/60000]
loss: 0.541300  [26000/60000]
loss: 0.382885  [28000/60000]
loss: 0.398809  [30000/60000]
loss: 0.372435  [32000/60000]
loss: 0.406709  [34000/60000]
loss: 0.474806  [36000/60000]
loss: 0.296364  [38000/60000]
loss: 0.261549  [40000/60000]
loss: 0.555476  [42000/60000]
loss: 0.523683  [44000/60000]
loss: 0.425478  [46000/60000]
loss: 0.287648  [48000/60000]
loss: 0.499335  [50000/60000]
loss: 0.121157  [52000/60000]
loss: 0.491877  [54000/60000]
loss: 0.246780  [56000/60000]
loss: 0.352122  [58000/60000]
Test Error:
 Accuracy: 83.9%, Avg loss: 0.022833

Epoch 27
-------------------------------
loss: 0.171080  [    0/60000]
```

```
loss: 0.261338  [ 2000/60000]
loss: 0.149445  [ 4000/60000]
loss: 0.222239  [ 6000/60000]
loss: 0.192809  [ 8000/60000]
loss: 0.555222  [10000/60000]
loss: 0.224043  [12000/60000]
loss: 0.562406  [14000/60000]
loss: 0.459651  [16000/60000]
loss: 0.643603  [18000/60000]
loss: 0.293847  [20000/60000]
loss: 0.326901  [22000/60000]
loss: 0.344484  [24000/60000]
loss: 0.225724  [26000/60000]
loss: 0.370726  [28000/60000]
loss: 0.409030  [30000/60000]
loss: 0.337083  [32000/60000]
loss: 0.349988  [34000/60000]
loss: 0.874935  [36000/60000]
loss: 0.288535  [38000/60000]
loss: 0.390747  [40000/60000]
loss: 0.769629  [42000/60000]
loss: 0.526652  [44000/60000]
loss: 0.648369  [46000/60000]
loss: 0.351275  [48000/60000]
loss: 0.158654  [50000/60000]
loss: 0.247581  [52000/60000]
loss: 0.463424  [54000/60000]
loss: 0.182854  [56000/60000]
loss: 0.338945  [58000/60000]
Test Error:
 Accuracy: 84.0%, Avg loss: 0.022359

Epoch 28
-------------------------------
loss: 0.314766  [    0/60000]
loss: 0.457515  [ 2000/60000]
loss: 0.636549  [ 4000/60000]
loss: 0.228395  [ 6000/60000]
loss: 0.557397  [ 8000/60000]
loss: 0.212325  [10000/60000]
loss: 0.258437  [12000/60000]
loss: 0.600765  [14000/60000]
loss: 0.575039  [16000/60000]
loss: 0.387526  [18000/60000]
loss: 0.415915  [20000/60000]
loss: 0.387059  [22000/60000]
loss: 0.169372  [24000/60000]
loss: 0.512745  [26000/60000]
```

```
loss: 0.288933  [28000/60000]
loss: 0.251637  [30000/60000]
loss: 0.299689  [32000/60000]
loss: 0.429992  [34000/60000]
loss: 0.481846  [36000/60000]
loss: 0.308690  [38000/60000]
loss: 0.538768  [40000/60000]
loss: 0.750468  [42000/60000]
loss: 0.499672  [44000/60000]
loss: 0.618239  [46000/60000]
loss: 0.720131  [48000/60000]
loss: 0.150248  [50000/60000]
loss: 0.251059  [52000/60000]
loss: 0.554178  [54000/60000]
loss: 0.401043  [56000/60000]
loss: 0.849558  [58000/60000]
Test Error:
 Accuracy: 84.1%, Avg loss: 0.022274

Epoch 29
-------------------------------
loss: 0.598980  [    0/60000]
loss: 0.247162  [ 2000/60000]
loss: 0.321313  [ 4000/60000]
loss: 0.575302  [ 6000/60000]
loss: 0.609683  [ 8000/60000]
loss: 0.386257  [10000/60000]
loss: 0.243701  [12000/60000]
loss: 1.164218  [14000/60000]
loss: 0.260695  [16000/60000]
loss: 0.454024  [18000/60000]
loss: 0.157289  [20000/60000]
loss: 0.364262  [22000/60000]
loss: 0.363659  [24000/60000]
loss: 0.694221  [26000/60000]
loss: 0.420782  [28000/60000]
loss: 0.307974  [30000/60000]
loss: 0.349566  [32000/60000]
loss: 0.618652  [34000/60000]
loss: 0.183334  [36000/60000]
loss: 0.405223  [38000/60000]
loss: 0.444249  [40000/60000]
loss: 0.587481  [42000/60000]
loss: 0.434036  [44000/60000]
loss: 0.444477  [46000/60000]
loss: 0.350548  [48000/60000]
loss: 0.310543  [50000/60000]
loss: 0.305269  [52000/60000]
```

```
loss: 0.385675  [54000/60000]
loss: 0.511756  [56000/60000]
loss: 0.334432  [58000/60000]
Test Error:
 Accuracy: 83.2%, Avg loss: 0.023177

Epoch 30
-------------------------------
loss: 0.339647  [    0/60000]
loss: 0.697195  [ 2000/60000]
loss: 0.556763  [ 4000/60000]
loss: 0.546387  [ 6000/60000]
loss: 0.308644  [ 8000/60000]
loss: 0.254990  [10000/60000]
loss: 0.272854  [12000/60000]
loss: 0.309550  [14000/60000]
loss: 0.408569  [16000/60000]
loss: 0.354552  [18000/60000]
loss: 0.234108  [20000/60000]
loss: 0.436584  [22000/60000]
loss: 0.423819  [24000/60000]
loss: 0.415022  [26000/60000]
loss: 0.378934  [28000/60000]
loss: 0.274815  [30000/60000]
loss: 0.479053  [32000/60000]
loss: 0.514944  [34000/60000]
loss: 0.586388  [36000/60000]
loss: 0.334611  [38000/60000]
loss: 0.456582  [40000/60000]
loss: 0.378607  [42000/60000]
loss: 0.433923  [44000/60000]
loss: 0.707766  [46000/60000]
loss: 0.515546  [48000/60000]
loss: 0.366249  [50000/60000]
loss: 0.628518  [52000/60000]
loss: 0.400445  [54000/60000]
loss: 0.581407  [56000/60000]
loss: 0.264231  [58000/60000]
Test Error:
 Accuracy: 84.3%, Avg loss: 0.022028

Epoch 31
-------------------------------
loss: 0.409415  [    0/60000]
loss: 0.230068  [ 2000/60000]
loss: 0.271399  [ 4000/60000]
loss: 0.669796  [ 6000/60000]
loss: 0.244543  [ 8000/60000]
```

```
loss: 0.134768  [10000/60000]
loss: 0.280281  [12000/60000]
loss: 0.293945  [14000/60000]
loss: 0.159334  [16000/60000]
loss: 0.176883  [18000/60000]
loss: 0.481323  [20000/60000]
loss: 0.347279  [22000/60000]
loss: 0.237550  [24000/60000]
loss: 0.378889  [26000/60000]
loss: 0.428188  [28000/60000]
loss: 0.483310  [30000/60000]
loss: 0.661657  [32000/60000]
loss: 0.410817  [34000/60000]
loss: 0.104303  [36000/60000]
loss: 0.195048  [38000/60000]
loss: 0.475504  [40000/60000]
loss: 0.427547  [42000/60000]
loss: 0.526557  [44000/60000]
loss: 0.214758  [46000/60000]
loss: 0.251048  [48000/60000]
loss: 0.398603  [50000/60000]
loss: 0.498168  [52000/60000]
loss: 0.536997  [54000/60000]
loss: 0.433380  [56000/60000]
loss: 0.181143  [58000/60000]
Test Error:
 Accuracy: 84.3%, Avg loss: 0.022049

Epoch 32
-------------------------------
loss: 0.381197  [    0/60000]
loss: 0.487164  [ 2000/60000]
loss: 0.446712  [ 4000/60000]
loss: 0.304958  [ 6000/60000]
loss: 0.185393  [ 8000/60000]
loss: 0.329161  [10000/60000]
loss: 0.532347  [12000/60000]
loss: 0.332050  [14000/60000]
loss: 0.159974  [16000/60000]
loss: 0.300386  [18000/60000]
loss: 0.592033  [20000/60000]
loss: 0.410123  [22000/60000]
loss: 0.218056  [24000/60000]
loss: 0.228406  [26000/60000]
loss: 0.235715  [28000/60000]
loss: 0.346333  [30000/60000]
loss: 0.346611  [32000/60000]
loss: 0.308105  [34000/60000]
```

```
loss: 0.204373  [36000/60000]
loss: 0.577086  [38000/60000]
loss: 0.725445  [40000/60000]
loss: 0.115334  [42000/60000]
loss: 0.627927  [44000/60000]
loss: 0.285324  [46000/60000]
loss: 0.262488  [48000/60000]
loss: 0.145952  [50000/60000]
loss: 0.314055  [52000/60000]
loss: 0.101043  [54000/60000]
loss: 0.591889  [56000/60000]
loss: 0.258002  [58000/60000]
Test Error:
 Accuracy: 84.5%, Avg loss: 0.021844

Epoch 33
-------------------------------
loss: 0.276071  [    0/60000]
loss: 0.227993  [ 2000/60000]
loss: 0.455401  [ 4000/60000]
loss: 0.428134  [ 6000/60000]
loss: 0.444575  [ 8000/60000]
loss: 0.182776  [10000/60000]
loss: 0.158441  [12000/60000]
loss: 0.549197  [14000/60000]
loss: 0.695230  [16000/60000]
loss: 0.221137  [18000/60000]
loss: 0.226240  [20000/60000]
loss: 0.608959  [22000/60000]
loss: 0.584306  [24000/60000]
loss: 0.774720  [26000/60000]
loss: 0.461752  [28000/60000]
loss: 0.782512  [30000/60000]
loss: 0.371295  [32000/60000]
loss: 0.398954  [34000/60000]
loss: 0.215198  [36000/60000]
loss: 0.435970  [38000/60000]
loss: 0.286564  [40000/60000]
loss: 0.627392  [42000/60000]
loss: 0.360459  [44000/60000]
loss: 0.288456  [46000/60000]
loss: 0.291543  [48000/60000]
loss: 0.299041  [50000/60000]
loss: 0.409462  [52000/60000]
loss: 0.465088  [54000/60000]
loss: 0.553827  [56000/60000]
loss: 0.348717  [58000/60000]
Test Error:
```

```
 Accuracy: 84.9%, Avg loss: 0.021632

Epoch 34
-------------------------------
loss: 0.679451  [    0/60000]
loss: 0.631088  [ 2000/60000]
loss: 0.241526  [ 4000/60000]
loss: 0.497705  [ 6000/60000]
loss: 0.412923  [ 8000/60000]
loss: 0.855169  [10000/60000]
loss: 0.302219  [12000/60000]
loss: 0.320686  [14000/60000]
loss: 0.601021  [16000/60000]
loss: 0.518540  [18000/60000]
loss: 0.172240  [20000/60000]
loss: 0.606621  [22000/60000]
loss: 0.545365  [24000/60000]
loss: 0.442323  [26000/60000]
loss: 0.341450  [28000/60000]
loss: 0.387920  [30000/60000]
loss: 0.436618  [32000/60000]
loss: 0.529196  [34000/60000]
loss: 0.105415  [36000/60000]
loss: 0.281269  [38000/60000]
loss: 0.190055  [40000/60000]
loss: 0.486571  [42000/60000]
loss: 0.313179  [44000/60000]
loss: 0.536952  [46000/60000]
loss: 0.261443  [48000/60000]
loss: 0.269510  [50000/60000]
loss: 0.519705  [52000/60000]
loss: 0.368519  [54000/60000]
loss: 0.173580  [56000/60000]
loss: 0.179707  [58000/60000]
Test Error:
 Accuracy: 84.5%, Avg loss: 0.021807

Epoch 35
-------------------------------
loss: 0.195148  [    0/60000]
loss: 0.491105  [ 2000/60000]
loss: 0.433730  [ 4000/60000]
loss: 0.467674  [ 6000/60000]
loss: 0.278630  [ 8000/60000]
loss: 0.313493  [10000/60000]
loss: 0.319261  [12000/60000]
loss: 0.177389  [14000/60000]
loss: 0.393024  [16000/60000]
```

```
loss: 0.371352  [18000/60000]
loss: 0.497103  [20000/60000]
loss: 0.610452  [22000/60000]
loss: 0.353437  [24000/60000]
loss: 0.577930  [26000/60000]
loss: 0.481185  [28000/60000]
loss: 0.167947  [30000/60000]
loss: 0.237912  [32000/60000]
loss: 0.348220  [34000/60000]
loss: 0.213302  [36000/60000]
loss: 0.419622  [38000/60000]
loss: 0.682178  [40000/60000]
loss: 0.609873  [42000/60000]
loss: 0.273753  [44000/60000]
loss: 0.276036  [46000/60000]
loss: 0.301765  [48000/60000]
loss: 0.494475  [50000/60000]
loss: 0.274217  [52000/60000]
loss: 0.533928  [54000/60000]
loss: 0.410422  [56000/60000]
loss: 0.470909  [58000/60000]
Test Error:
 Accuracy: 84.3%, Avg loss: 0.021899

Epoch 36
-------------------------------
loss: 0.454392  [    0/60000]
loss: 0.423872  [ 2000/60000]
loss: 0.596158  [ 4000/60000]
loss: 0.767811  [ 6000/60000]
loss: 0.549884  [ 8000/60000]
loss: 0.382153  [10000/60000]
loss: 0.382708  [12000/60000]
loss: 0.063219  [14000/60000]
loss: 0.558187  [16000/60000]
loss: 0.522580  [18000/60000]
loss: 0.575977  [20000/60000]
loss: 0.081882  [22000/60000]
loss: 0.435366  [24000/60000]
loss: 0.189791  [26000/60000]
loss: 0.135821  [28000/60000]
loss: 0.733120  [30000/60000]
loss: 0.146787  [32000/60000]
loss: 0.255251  [34000/60000]
loss: 0.211552  [36000/60000]
loss: 0.640578  [38000/60000]
loss: 0.675465  [40000/60000]
loss: 0.484889  [42000/60000]
```

```
loss: 0.244178  [44000/60000]
loss: 0.257799  [46000/60000]
loss: 0.272370  [48000/60000]
loss: 0.247934  [50000/60000]
loss: 0.187465  [52000/60000]
loss: 0.237526  [54000/60000]
loss: 0.354069  [56000/60000]
loss: 0.357834  [58000/60000]
Test Error:
 Accuracy: 84.6%, Avg loss: 0.021535

Epoch 37
-------------------------------
loss: 0.471344  [    0/60000]
loss: 0.180059  [ 2000/60000]
loss: 0.243386  [ 4000/60000]
loss: 0.565850  [ 6000/60000]
loss: 0.320589  [ 8000/60000]
loss: 0.444606  [10000/60000]
loss: 0.200231  [12000/60000]
loss: 0.338843  [14000/60000]
loss: 0.361842  [16000/60000]
loss: 0.439745  [18000/60000]
loss: 0.164282  [20000/60000]
loss: 0.396585  [22000/60000]
loss: 0.280305  [24000/60000]
loss: 0.547507  [26000/60000]
loss: 0.687372  [28000/60000]
loss: 0.436462  [30000/60000]
loss: 0.447351  [32000/60000]
loss: 0.468164  [34000/60000]
loss: 0.387829  [36000/60000]
loss: 0.775903  [38000/60000]
loss: 0.331903  [40000/60000]
loss: 0.588532  [42000/60000]
loss: 0.314588  [44000/60000]
loss: 0.270326  [46000/60000]
loss: 0.431449  [48000/60000]
loss: 0.140096  [50000/60000]
loss: 0.179387  [52000/60000]
loss: 0.352432  [54000/60000]
loss: 0.736220  [56000/60000]
loss: 0.316145  [58000/60000]
Test Error:
 Accuracy: 85.1%, Avg loss: 0.021215

Epoch 38
-------------------------------
```

```
loss: 0.422995  [    0/60000]
loss: 0.741234  [ 2000/60000]
loss: 0.380755  [ 4000/60000]
loss: 0.401698  [ 6000/60000]
loss: 0.247908  [ 8000/60000]
loss: 0.617148  [10000/60000]
loss: 0.477416  [12000/60000]
loss: 0.546854  [14000/60000]
loss: 0.213535  [16000/60000]
loss: 0.399783  [18000/60000]
loss: 0.438010  [20000/60000]
loss: 0.245601  [22000/60000]
loss: 0.614253  [24000/60000]
loss: 0.091529  [26000/60000]
loss: 0.379128  [28000/60000]
loss: 0.480661  [30000/60000]
loss: 0.398782  [32000/60000]
loss: 0.471299  [34000/60000]
loss: 0.137514  [36000/60000]
loss: 0.350200  [38000/60000]
loss: 0.683704  [40000/60000]
loss: 0.447474  [42000/60000]
loss: 0.547199  [44000/60000]
loss: 0.244961  [46000/60000]
loss: 0.652186  [48000/60000]
loss: 0.470830  [50000/60000]
loss: 0.531528  [52000/60000]
loss: 0.183895  [54000/60000]
loss: 0.377244  [56000/60000]
loss: 0.289575  [58000/60000]
Test Error:
 Accuracy: 84.9%, Avg loss: 0.021087

Epoch 39
-------------------------------
loss: 0.369175  [    0/60000]
loss: 0.650166  [ 2000/60000]
loss: 0.204393  [ 4000/60000]
loss: 0.676076  [ 6000/60000]
loss: 0.245698  [ 8000/60000]
loss: 0.563068  [10000/60000]
loss: 0.207756  [12000/60000]
loss: 0.522316  [14000/60000]
loss: 0.817075  [16000/60000]
loss: 0.179746  [18000/60000]
loss: 0.294534  [20000/60000]
loss: 0.308137  [22000/60000]
loss: 0.419592  [24000/60000]
```

```
loss: 0.474044  [26000/60000]
loss: 0.445327  [28000/60000]
loss: 0.276829  [30000/60000]
loss: 0.679084  [32000/60000]
loss: 0.293374  [34000/60000]
loss: 0.646757  [36000/60000]
loss: 0.207762  [38000/60000]
loss: 0.235276  [40000/60000]
loss: 0.194488  [42000/60000]
loss: 0.236745  [44000/60000]
loss: 0.234480  [46000/60000]
loss: 0.311641  [48000/60000]
loss: 0.440247  [50000/60000]
loss: 0.481913  [52000/60000]
loss: 0.464090  [54000/60000]
loss: 0.120067  [56000/60000]
loss: 0.259038  [58000/60000]
Test Error:
 Accuracy: 85.1%, Avg loss: 0.021292

Epoch 40
-------------------------------
loss: 0.153597  [    0/60000]
loss: 0.142387  [ 2000/60000]
loss: 0.205221  [ 4000/60000]
loss: 0.422757  [ 6000/60000]
loss: 0.370416  [ 8000/60000]
loss: 0.391883  [10000/60000]
loss: 0.306231  [12000/60000]
loss: 0.504396  [14000/60000]
loss: 0.271652  [16000/60000]
loss: 0.242809  [18000/60000]
loss: 0.647556  [20000/60000]
loss: 0.290452  [22000/60000]
loss: 0.480230  [24000/60000]
loss: 0.180095  [26000/60000]
loss: 0.270124  [28000/60000]
loss: 0.582530  [30000/60000]
loss: 0.252465  [32000/60000]
loss: 0.414486  [34000/60000]
loss: 0.228927  [36000/60000]
loss: 0.141152  [38000/60000]
loss: 0.247601  [40000/60000]
loss: 0.402203  [42000/60000]
loss: 0.325863  [44000/60000]
loss: 0.413172  [46000/60000]
loss: 0.828125  [48000/60000]
loss: 0.396913  [50000/60000]
```

```
loss: 0.268203  [52000/60000]
loss: 0.240989  [54000/60000]
loss: 0.572848  [56000/60000]
loss: 0.165154  [58000/60000]
Test Error:
 Accuracy: 85.3%, Avg loss: 0.020934

Epoch 41
-------------------------------
loss: 0.151066  [    0/60000]
loss: 0.344647  [ 2000/60000]
loss: 0.341661  [ 4000/60000]
loss: 0.309685  [ 6000/60000]
loss: 0.450709  [ 8000/60000]
loss: 0.595809  [10000/60000]
loss: 0.335920  [12000/60000]
loss: 0.695773  [14000/60000]
loss: 0.644232  [16000/60000]
loss: 0.525449  [18000/60000]
loss: 0.480619  [20000/60000]
loss: 0.188839  [22000/60000]
loss: 0.357795  [24000/60000]
loss: 0.290170  [26000/60000]
loss: 0.469459  [28000/60000]
loss: 0.375639  [30000/60000]
loss: 0.169225  [32000/60000]
loss: 0.167970  [34000/60000]
loss: 0.497196  [36000/60000]
loss: 0.153716  [38000/60000]
loss: 0.363005  [40000/60000]
loss: 0.554586  [42000/60000]
loss: 0.307205  [44000/60000]
loss: 0.325669  [46000/60000]
loss: 0.988584  [48000/60000]
loss: 0.210753  [50000/60000]
loss: 0.371393  [52000/60000]
loss: 0.177849  [54000/60000]
loss: 0.319071  [56000/60000]
loss: 0.309970  [58000/60000]
Test Error:
 Accuracy: 84.9%, Avg loss: 0.021559

Epoch 42
-------------------------------
loss: 0.407941  [    0/60000]
loss: 0.879170  [ 2000/60000]
loss: 0.205125  [ 4000/60000]
loss: 0.213695  [ 6000/60000]
```

```
loss: 0.652273  [ 8000/60000]
loss: 0.267508  [10000/60000]
loss: 0.131343  [12000/60000]
loss: 0.258628  [14000/60000]
loss: 0.321543  [16000/60000]
loss: 0.401838  [18000/60000]
loss: 0.706083  [20000/60000]
loss: 0.182090  [22000/60000]
loss: 0.384950  [24000/60000]
loss: 0.415713  [26000/60000]
loss: 0.440088  [28000/60000]
loss: 0.356864  [30000/60000]
loss: 0.240613  [32000/60000]
loss: 0.301331  [34000/60000]
loss: 0.297611  [36000/60000]
loss: 0.220604  [38000/60000]
loss: 0.212342  [40000/60000]
loss: 0.888635  [42000/60000]
loss: 0.618507  [44000/60000]
loss: 0.577854  [46000/60000]
loss: 0.164763  [48000/60000]
loss: 0.073755  [50000/60000]
loss: 0.374439  [52000/60000]
loss: 0.200528  [54000/60000]
loss: 0.237258  [56000/60000]
loss: 0.277098  [58000/60000]
Test Error:
 Accuracy: 85.1%, Avg loss: 0.020860

Epoch 43
-------------------------------
loss: 0.457037  [    0/60000]
loss: 0.169292  [ 2000/60000]
loss: 0.354759  [ 4000/60000]
loss: 0.268487  [ 6000/60000]
loss: 0.169033  [ 8000/60000]
loss: 0.175171  [10000/60000]
loss: 0.512869  [12000/60000]
loss: 0.376474  [14000/60000]
loss: 0.265118  [16000/60000]
loss: 0.405198  [18000/60000]
loss: 0.337347  [20000/60000]
loss: 0.320326  [22000/60000]
loss: 0.792143  [24000/60000]
loss: 0.161875  [26000/60000]
loss: 0.343902  [28000/60000]
loss: 0.316491  [30000/60000]
loss: 0.542517  [32000/60000]
```

```
loss: 0.115512  [34000/60000]
loss: 0.176047  [36000/60000]
loss: 0.470724  [38000/60000]
loss: 0.203816  [40000/60000]
loss: 0.272240  [42000/60000]
loss: 0.766715  [44000/60000]
loss: 0.464577  [46000/60000]
loss: 0.698509  [48000/60000]
loss: 0.253653  [50000/60000]
loss: 0.099310  [52000/60000]
loss: 0.251614  [54000/60000]
loss: 0.423305  [56000/60000]
loss: 0.425169  [58000/60000]
Test Error:
 Accuracy: 85.4%, Avg loss: 0.020744

Epoch 44
-------------------------------
loss: 0.401534  [    0/60000]
loss: 0.379846  [ 2000/60000]
loss: 0.444873  [ 4000/60000]
loss: 0.440076  [ 6000/60000]
loss: 0.320743  [ 8000/60000]
loss: 0.363262  [10000/60000]
loss: 0.545316  [12000/60000]
loss: 0.578762  [14000/60000]
loss: 0.192469  [16000/60000]
loss: 0.646665  [18000/60000]
loss: 0.609672  [20000/60000]
loss: 0.289964  [22000/60000]
loss: 0.194512  [24000/60000]
loss: 0.406185  [26000/60000]
loss: 0.488025  [28000/60000]
loss: 0.393986  [30000/60000]
loss: 0.325888  [32000/60000]
loss: 0.510546  [34000/60000]
loss: 0.668197  [36000/60000]
loss: 0.285262  [38000/60000]
loss: 0.692627  [40000/60000]
loss: 0.265130  [42000/60000]
loss: 0.132196  [44000/60000]
loss: 0.202607  [46000/60000]
loss: 0.669520  [48000/60000]
loss: 0.232245  [50000/60000]
loss: 0.454523  [52000/60000]
loss: 0.114332  [54000/60000]
loss: 0.451992  [56000/60000]
loss: 0.267415  [58000/60000]
```

```
Test Error:
 Accuracy: 85.4%, Avg loss: 0.020539

Epoch 45
-------------------------------
loss: 0.196099  [    0/60000]
loss: 0.141413  [ 2000/60000]
loss: 0.674957  [ 4000/60000]
loss: 0.398325  [ 6000/60000]
loss: 0.350467  [ 8000/60000]
loss: 0.353690  [10000/60000]
loss: 0.132402  [12000/60000]
loss: 0.294035  [14000/60000]
loss: 0.319088  [16000/60000]
loss: 0.371041  [18000/60000]
loss: 0.319217  [20000/60000]
loss: 0.487280  [22000/60000]
loss: 0.502080  [24000/60000]
loss: 0.353284  [26000/60000]
loss: 0.274680  [28000/60000]
loss: 0.205790  [30000/60000]
loss: 0.122578  [32000/60000]
loss: 0.415471  [34000/60000]
loss: 0.269053  [36000/60000]
loss: 0.348591  [38000/60000]
loss: 0.171575  [40000/60000]
loss: 0.375241  [42000/60000]
loss: 0.292875  [44000/60000]
loss: 0.227642  [46000/60000]
loss: 0.531569  [48000/60000]
loss: 0.259183  [50000/60000]
loss: 0.348855  [52000/60000]
loss: 0.584723  [54000/60000]
loss: 0.292214  [56000/60000]
loss: 0.282895  [58000/60000]
Test Error:
 Accuracy: 85.1%, Avg loss: 0.021291

Epoch 46
-------------------------------
loss: 0.720522  [    0/60000]
loss: 0.329464  [ 2000/60000]
loss: 0.421902  [ 4000/60000]
loss: 0.667356  [ 6000/60000]
loss: 0.279202  [ 8000/60000]
loss: 0.165110  [10000/60000]
loss: 0.516503  [12000/60000]
loss: 0.169593  [14000/60000]
```

```
loss: 0.180269  [16000/60000]
loss: 0.454534  [18000/60000]
loss: 0.448411  [20000/60000]
loss: 0.586522  [22000/60000]
loss: 0.400147  [24000/60000]
loss: 0.442246  [26000/60000]
loss: 0.525110  [28000/60000]
loss: 0.497048  [30000/60000]
loss: 0.445419  [32000/60000]
loss: 0.210297  [34000/60000]
loss: 0.450996  [36000/60000]
loss: 0.200949  [38000/60000]
loss: 0.138999  [40000/60000]
loss: 0.595855  [42000/60000]
loss: 0.212821  [44000/60000]
loss: 0.444723  [46000/60000]
loss: 0.644623  [48000/60000]
loss: 0.253559  [50000/60000]
loss: 0.501052  [52000/60000]
loss: 0.397219  [54000/60000]
loss: 0.221925  [56000/60000]
loss: 0.169910  [58000/60000]
Test Error:
 Accuracy: 85.5%, Avg loss: 0.020535

Epoch 47
-------------------------------
loss: 0.264516  [    0/60000]
loss: 0.518591  [ 2000/60000]
loss: 0.391601  [ 4000/60000]
loss: 0.345367  [ 6000/60000]
loss: 0.247443  [ 8000/60000]
loss: 0.168589  [10000/60000]
loss: 0.294344  [12000/60000]
loss: 0.651169  [14000/60000]
loss: 0.297461  [16000/60000]
loss: 0.286524  [18000/60000]
loss: 0.421977  [20000/60000]
loss: 0.686190  [22000/60000]
loss: 0.393794  [24000/60000]
loss: 0.237770  [26000/60000]
loss: 0.382262  [28000/60000]
loss: 0.136875  [30000/60000]
loss: 0.168227  [32000/60000]
loss: 0.385370  [34000/60000]
loss: 0.168809  [36000/60000]
loss: 0.466111  [38000/60000]
loss: 0.559046  [40000/60000]
```

```
loss: 0.255439  [42000/60000]
loss: 0.460974  [44000/60000]
loss: 0.393586  [46000/60000]
loss: 0.375484  [48000/60000]
loss: 0.327131  [50000/60000]
loss: 0.462512  [52000/60000]
loss: 0.548310  [54000/60000]
loss: 0.174852  [56000/60000]
loss: 0.283525  [58000/60000]
Test Error:
 Accuracy: 85.7%, Avg loss: 0.020263

Epoch 48
-------------------------------
loss: 0.529546  [    0/60000]
loss: 0.711456  [ 2000/60000]
loss: 0.190235  [ 4000/60000]
loss: 0.146634  [ 6000/60000]
loss: 0.414020  [ 8000/60000]
loss: 0.193823  [10000/60000]
loss: 0.155631  [12000/60000]
loss: 0.248444  [14000/60000]
loss: 0.136052  [16000/60000]
loss: 0.476589  [18000/60000]
loss: 0.277821  [20000/60000]
loss: 0.143992  [22000/60000]
loss: 0.291148  [24000/60000]
loss: 0.299686  [26000/60000]
loss: 0.528786  [28000/60000]
loss: 0.232656  [30000/60000]
loss: 0.277121  [32000/60000]
loss: 0.305679  [34000/60000]
loss: 0.277148  [36000/60000]
loss: 0.248426  [38000/60000]
loss: 0.187178  [40000/60000]
loss: 0.456407  [42000/60000]
loss: 0.335112  [44000/60000]
loss: 0.370089  [46000/60000]
loss: 0.389177  [48000/60000]
loss: 0.560646  [50000/60000]
loss: 0.568503  [52000/60000]
loss: 0.310773  [54000/60000]
loss: 0.646816  [56000/60000]
loss: 0.147702  [58000/60000]
Test Error:
 Accuracy: 85.3%, Avg loss: 0.020439

Epoch 49
```

```
-------------------------------
loss: 0.401721  [    0/60000]
loss: 0.243639  [ 2000/60000]
loss: 0.337102  [ 4000/60000]
loss: 0.491471  [ 6000/60000]
loss: 0.556118  [ 8000/60000]
loss: 0.313565  [10000/60000]
loss: 0.482663  [12000/60000]
loss: 0.105854  [14000/60000]
loss: 0.417358  [16000/60000]
loss: 0.062081  [18000/60000]
loss: 0.122578  [20000/60000]
loss: 0.499368  [22000/60000]
loss: 0.412000  [24000/60000]
loss: 0.430714  [26000/60000]
loss: 0.287181  [28000/60000]
loss: 0.283320  [30000/60000]
loss: 0.804142  [32000/60000]
loss: 0.305113  [34000/60000]
loss: 0.186209  [36000/60000]
loss: 0.282997  [38000/60000]
loss: 0.380261  [40000/60000]
loss: 0.325967  [42000/60000]
loss: 0.468961  [44000/60000]
loss: 0.558956  [46000/60000]
loss: 0.275116  [48000/60000]
loss: 0.580058  [50000/60000]
loss: 0.289445  [52000/60000]
loss: 0.591416  [54000/60000]
loss: 0.529931  [56000/60000]
loss: 0.142492  [58000/60000]
Test Error:
 Accuracy: 85.6%, Avg loss: 0.020253

Epoch 50
-------------------------------
loss: 0.243540  [    0/60000]
loss: 0.216742  [ 2000/60000]
loss: 0.437545  [ 4000/60000]
loss: 0.747316  [ 6000/60000]
loss: 0.225040  [ 8000/60000]
loss: 0.271205  [10000/60000]
loss: 0.264643  [12000/60000]
loss: 0.083286  [14000/60000]
loss: 0.211620  [16000/60000]
loss: 0.384806  [18000/60000]
loss: 0.281321  [20000/60000]
loss: 0.509731  [22000/60000]
```

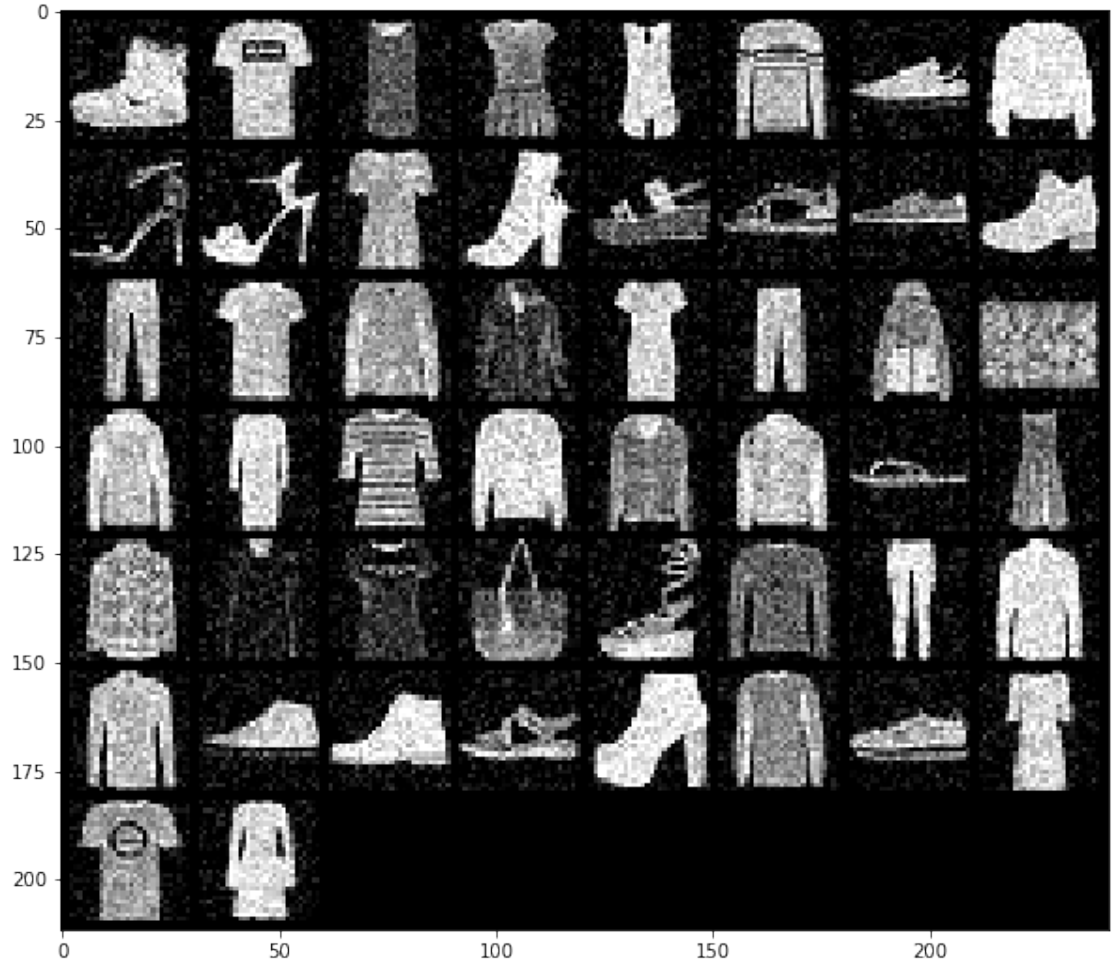
```
loss: 0.408460  [24000/60000]
loss: 0.296143  [26000/60000]
loss: 0.295677  [28000/60000]
loss: 0.241329  [30000/60000]
loss: 0.133101  [32000/60000]
loss: 0.333134  [34000/60000]
loss: 0.216544  [36000/60000]
loss: 0.435272  [38000/60000]
loss: 0.120267  [40000/60000]
loss: 0.355821  [42000/60000]
loss: 0.194458  [44000/60000]
loss: 0.334823  [46000/60000]
loss: 0.271909  [48000/60000]
loss: 0.333106  [50000/60000]
loss: 0.970565  [52000/60000]
loss: 0.574639  [54000/60000]
loss: 0.396736  [56000/60000]
loss: 0.744688  [58000/60000]
Test Error:
 Accuracy: 85.7%, Avg loss: 0.020144

Done!
```

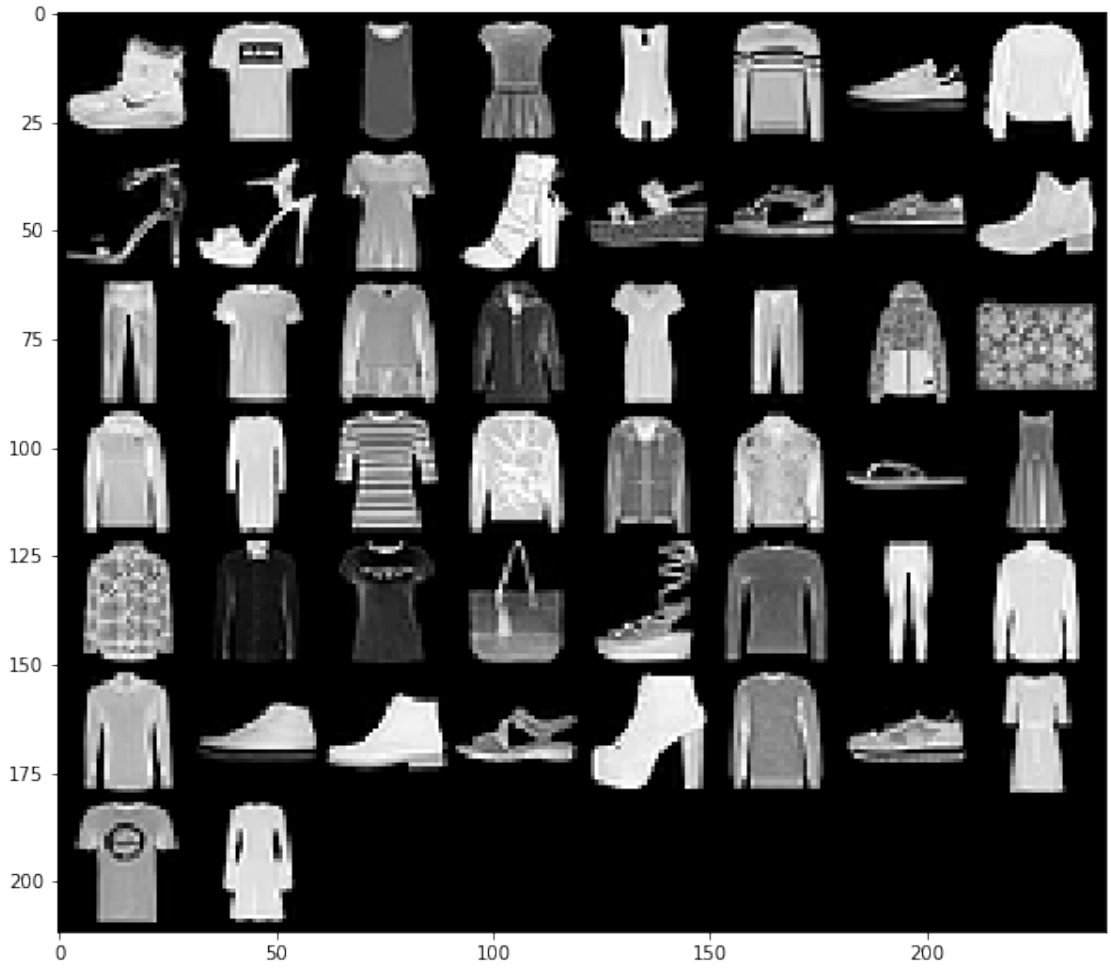
## 2 Task 2:

[9]:
```
train_dataset = NoisyFashionMNIST("./data", True)
test_dataset = NoisyFashionMNIST("./data", False)
```

[10]:
```
x = [train_dataset[i][0] for i in range(50)]
y = [train_dataset[i][1] for i in range(50)]

plt.figure(figsize=(10,10))
show(torchvision.utils.make_grid(x))
plt.show()

plt.figure(figsize=(10,10))
show(torchvision.utils.make_grid(y))
plt.show()
```

```
Clipping input data to the valid range for imshow with RGB data ([0..1] for
floats or [0..255] for integers).
```


0
25
50
75
100
125
150
175
200
0
50
100
150
200




## 2.1 Autoencoder

An autoencoder consists of an encoder and a decoder part. In the encoder part data is compressed to e.g. enable better identification of important data. We half the size of the input features until we reach 16. The decoder part decodes that encoded data again and in our case, transforms it back to its original input size by doubling the feature size in each layer. We found that linear layers together with relus are sufficient, to quickly and efficiently achieve good results.

[11]:
```
class Autoencoder(nn.Module):
    def __init__(self):
        super(Autoencoder, self).__init__()
        # We found that simple linear layers are sufficient for the network to␣
↪denoise
        self.flatten = nn.Flatten()
        # encoder
        self.enc1 = nn.Linear(in_features=28*28, out_features=256)
        self.enc2 = nn.Linear(in_features=256, out_features=128)
```

```
        self.enc3 = nn.Linear(in_features=128, out_features=64)
        self.enc4 = nn.Linear(in_features=64, out_features=32)
        self.enc5 = nn.Linear(in_features=32, out_features=16)
        # decoder
        self.dec1 = nn.Linear(in_features=16, out_features=32)
        self.dec2 = nn.Linear(in_features=32, out_features=64)
        self.dec3 = nn.Linear(in_features=64, out_features=128)
        self.dec4 = nn.Linear(in_features=128, out_features=256)
        self.dec5 = nn.Linear(in_features=256, out_features=28*28)

    def forward(self, x):
        x = self.flatten(x)
        # relu works fast and reliable as activiation function
        x = F.relu(self.enc1(x))
        x = F.relu(self.enc2(x))
        x = F.relu(self.enc3(x))
        x = F.relu(self.enc4(x))
        x = F.relu(self.enc5(x))
        x = F.relu(self.dec1(x))
        x = F.relu(self.dec2(x))
        x = F.relu(self.dec3(x))
        x = F.relu(self.dec4(x))
        x = F.relu(self.dec5(x))
        return x

model = Autoencoder()
model.to(device)
print(model)
```

```
Autoencoder(
  (flatten): Flatten(start_dim=1, end_dim=-1)
  (enc1): Linear(in_features=784, out_features=256, bias=True)
  (enc2): Linear(in_features=256, out_features=128, bias=True)
  (enc3): Linear(in_features=128, out_features=64, bias=True)
  (enc4): Linear(in_features=64, out_features=32, bias=True)
  (enc5): Linear(in_features=32, out_features=16, bias=True)
  (dec1): Linear(in_features=16, out_features=32, bias=True)
  (dec2): Linear(in_features=32, out_features=64, bias=True)
  (dec3): Linear(in_features=64, out_features=128, bias=True)
  (dec4): Linear(in_features=128, out_features=256, bias=True)
  (dec5): Linear(in_features=256, out_features=784, bias=True)
)
```

[12]:
```
# function for training
# @param dataloader: the data loader for the training set
# @param model: the previously defined neural network model
# @param loss_fn: defined loss function
```

```
# @param optimizer: how we calculate the loss
def train_loop(dataloader, model, loss_fn, optimizer):
    size = len(dataloader.dataset)
    for batch, (X, y) in enumerate(dataloader):
        # move X and y to gpu if available
        X = X.to(device)
        y = y.to(device)
        y = torch.flatten(y, start_dim=1)
        # Compute prediction and loss
        pred = model(X)
        loss = loss_fn(pred, y)

        # Backpropagation
        optimizer.zero_grad()
        loss.backward()
        optimizer.step()

        if batch % 100 == 0:
            loss, current = loss.item(), batch * len(X)
            print(f"loss: {loss:>7f}  [{current:>5d}/{size:>5d}]")

# function for testing, here we only do empirical testing --> we print the␣
 ↪predictions, i.e. the denoised
# pictures and the labels
# @param dataloader: the data loader for the training set
# @param model: the previously defined neural network model after training
def test_loop(dataloader, model):

    for X, y in dataloader:
        # move X and y to gpu if available
        X = X.to(device)
        y = y.to(device)
        img = y
        y = torch.flatten(y, start_dim=1)
        pred = model(X)
        pred = torch.reshape(pred, (20, 1, 28,28))
        pred = pred.cpu()
        img = img.cpu()

    x = [pred[i] for i in range(19)]
    y = [img[i] for i in range(19)]
    plt.figure(figsize=(10,10))
    show(torchvision.utils.make_grid(x))
    plt.show()
    plt.figure(figsize=(10,10))
    show(torchvision.utils.make_grid(y))
    plt.show()
```

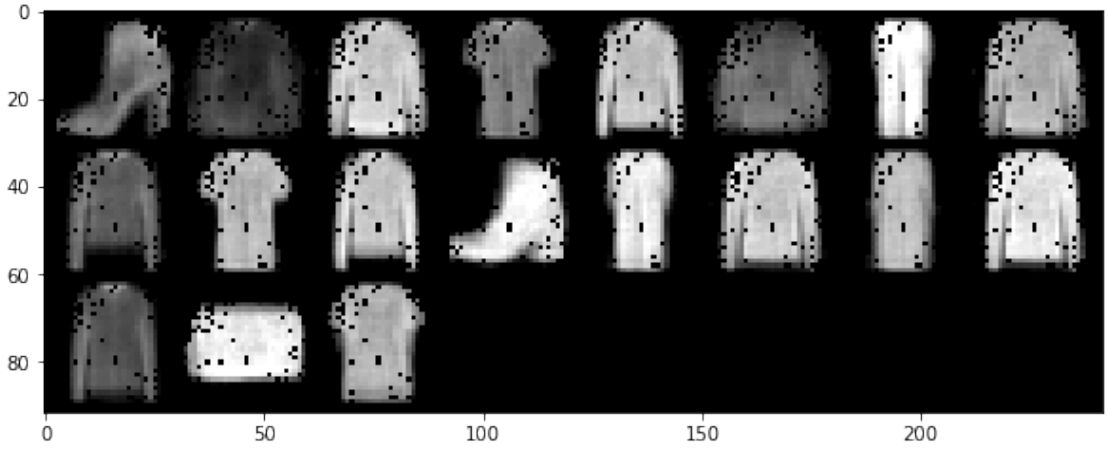
After conducting some internet research, we found that denoising autoencoders are mostly evaluated by reviewing the test errors and comparing denoised pictures to labels. Hence, we refrain from calculating accuracy and instead display said pictures after each training epoch.

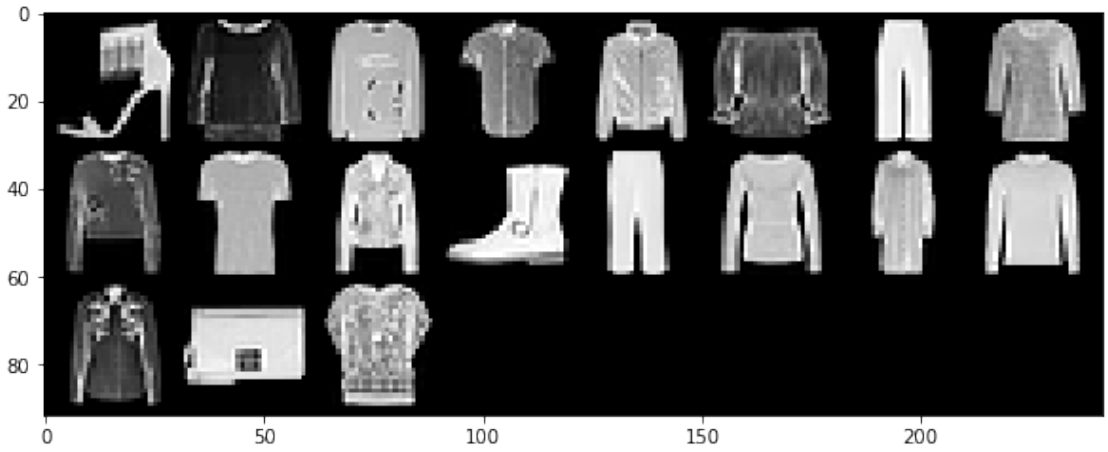
[13]:
```
# define hyper params
# updating rate for each batch/epoch
learning_rate = 1e-3
# number of data samples to propagate through network before updating params
batch_size = 64
# iteration times, we used 50 to, on the one hand, test our gpu performance out␣
 ↪of curiosity and, on the other
# hand, see, how much impact the number of epochs has on the performance of the␣
 ↪net.
epochs = 50
# using CrossEntropy for numeric stability, combines LogSoftmax and NLLLoss
# useful when training a problem with C classes

# We investigated multiple loss functions, including but not limited to␣
 ↪CrossEntropyLoss, KLDivLoss, and BCELoss
# Through empirical tests, we found, that the MSELoss works well for denoising␣
 ↪in our network.
# The denoising task, in contrast to the classification task, is a regression␣
 ↪task, hence, common regression
# losses like MSE work well in this case
loss_fn = nn.MSELoss()
# defining the optimizer
optimizer = torch.optim.Adam(model.parameters(), lr=learning_rate)

# use data loader to obtain batch training and test data
train_loader = torch.utils.data.DataLoader(train_dataset, batch_size = 20,␣
 ↪shuffle=True)
test_loader = torch.utils.data.DataLoader(test_dataset, batch_size = 20,␣
 ↪shuffle=True)

for t in range(epochs):
    print(f"Epoch {t+1}\n-------------------------------")
    train_loop(train_loader, model, loss_fn, optimizer)
    test_loop(test_loader, model)
print("Done!")
```

```
Epoch 1
-------------------------------
loss: 0.241252  [    0/60000]
loss: 0.117466  [ 2000/60000]
loss: 0.120310  [ 4000/60000]
loss: 0.062071  [ 6000/60000]
loss: 0.057809  [ 8000/60000]
```

```
loss: 0.053941  [10000/60000]
loss: 0.040293  [12000/60000]
loss: 0.049407  [14000/60000]
loss: 0.057688  [16000/60000]
loss: 0.072280  [18000/60000]
loss: 0.050666  [20000/60000]
loss: 0.053689  [22000/60000]
loss: 0.053847  [24000/60000]
loss: 0.062665  [26000/60000]
loss: 0.053468  [28000/60000]
loss: 0.044271  [30000/60000]
loss: 0.049811  [32000/60000]
loss: 0.041836  [34000/60000]
loss: 0.035506  [36000/60000]
loss: 0.051145  [38000/60000]
loss: 0.039259  [40000/60000]
loss: 0.035055  [42000/60000]
loss: 0.037451  [44000/60000]
loss: 0.040996  [46000/60000]
loss: 0.043236  [48000/60000]
loss: 0.031809  [50000/60000]
loss: 0.035170  [52000/60000]
loss: 0.042376  [54000/60000]
loss: 0.041742  [56000/60000]
loss: 0.038022  [58000/60000]
```

```
Clipping input data to the valid range for imshow with RGB data ([0..1] for
floats or [0..255] for integers).
```

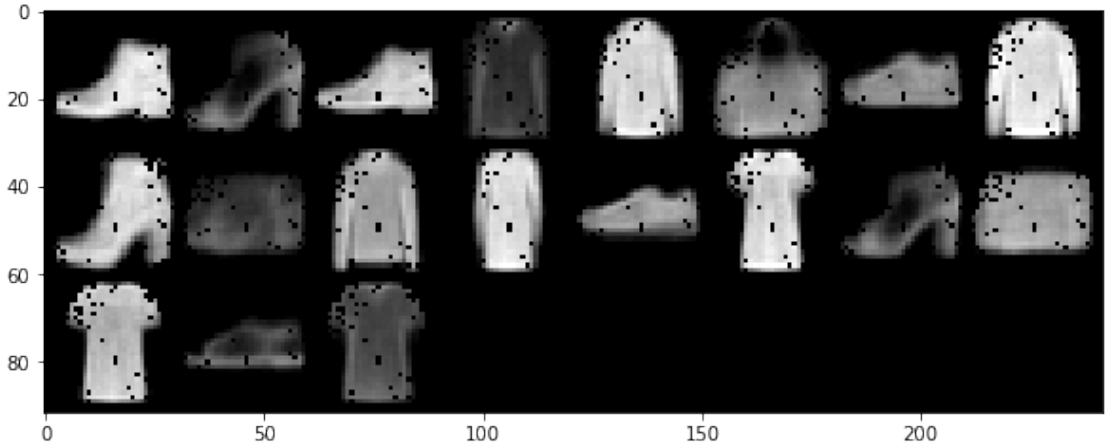


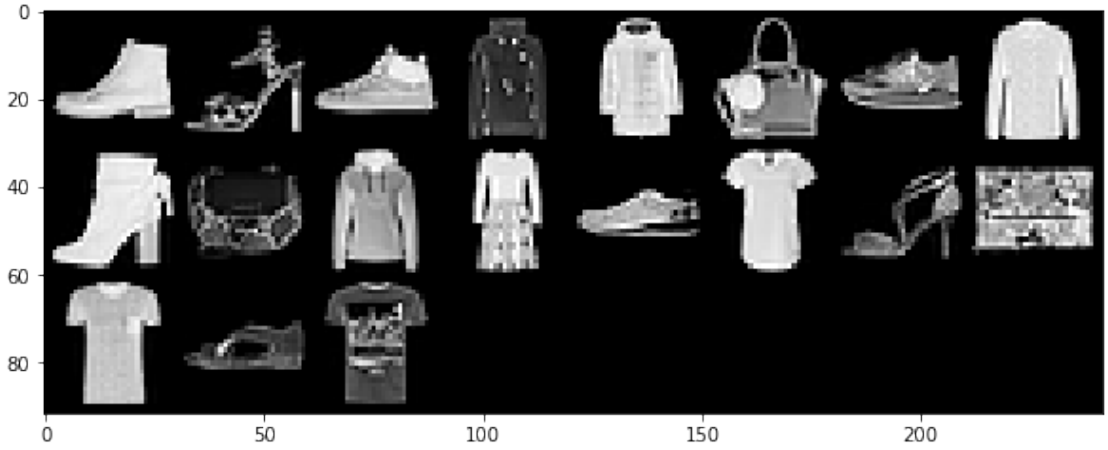



```
Epoch 2
-------------------------------
loss: 0.035346  [    0/60000]
loss: 0.030390  [ 2000/60000]
loss: 0.039968  [ 4000/60000]
loss: 0.032057  [ 6000/60000]
loss: 0.032130  [ 8000/60000]
loss: 0.034007  [10000/60000]
loss: 0.038569  [12000/60000]
loss: 0.036707  [14000/60000]
loss: 0.034573  [16000/60000]
loss: 0.037444  [18000/60000]
loss: 0.032280  [20000/60000]
loss: 0.032230  [22000/60000]
loss: 0.037252  [24000/60000]
loss: 0.032721  [26000/60000]
loss: 0.037024  [28000/60000]
loss: 0.027674  [30000/60000]
loss: 0.037107  [32000/60000]
loss: 0.032482  [34000/60000]
loss: 0.034344  [36000/60000]
loss: 0.030397  [38000/60000]
loss: 0.031307  [40000/60000]
loss: 0.033655  [42000/60000]
loss: 0.031323  [44000/60000]
loss: 0.036870  [46000/60000]
loss: 0.034712  [48000/60000]
loss: 0.025827  [50000/60000]
loss: 0.028278  [52000/60000]
loss: 0.039902  [54000/60000]
loss: 0.031779  [56000/60000]
```

```
loss: 0.032610  [58000/60000]
```

```
Clipping input data to the valid range for imshow with RGB data ([0..1] for
floats or [0..255] for integers).
```

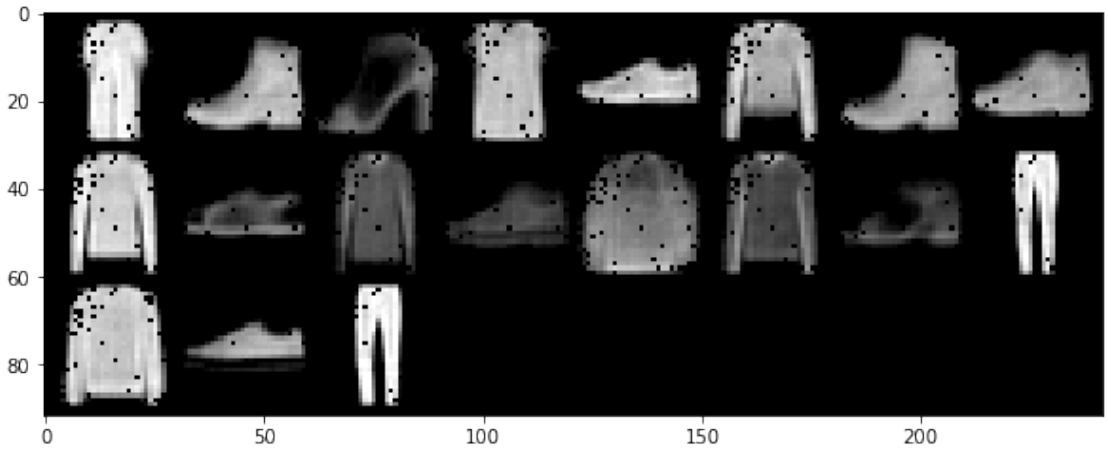


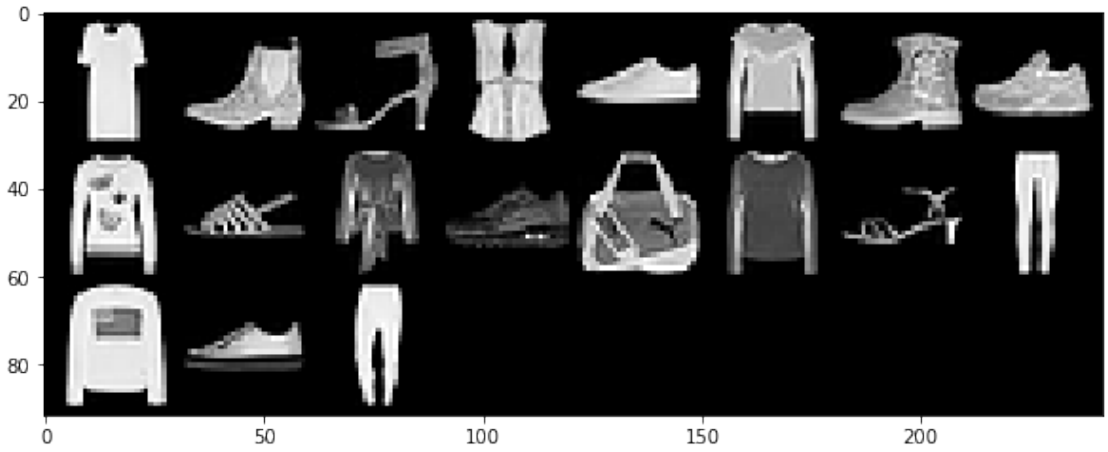




```
Epoch 3
-------------------------------
loss: 0.037479  [    0/60000]
loss: 0.030603  [ 2000/60000]
loss: 0.028677  [ 4000/60000]
loss: 0.033095  [ 6000/60000]
loss: 0.034482  [ 8000/60000]
loss: 0.029600  [10000/60000]
loss: 0.028973  [12000/60000]
loss: 0.028893  [14000/60000]
loss: 0.034826  [16000/60000]
```

```
loss: 0.026534  [18000/60000]
loss: 0.025768  [20000/60000]
loss: 0.030954  [22000/60000]
loss: 0.031142  [24000/60000]
loss: 0.032867  [26000/60000]
loss: 0.031377  [28000/60000]
loss: 0.027562  [30000/60000]
loss: 0.029668  [32000/60000]
loss: 0.023990  [34000/60000]
loss: 0.027427  [36000/60000]
loss: 0.030931  [38000/60000]
loss: 0.028382  [40000/60000]
loss: 0.032808  [42000/60000]
loss: 0.035520  [44000/60000]
loss: 0.029352  [46000/60000]
loss: 0.029733  [48000/60000]
loss: 0.023594  [50000/60000]
loss: 0.034532  [52000/60000]
loss: 0.021927  [54000/60000]
loss: 0.028524  [56000/60000]
loss: 0.027136  [58000/60000]
```

```
Clipping input data to the valid range for imshow with RGB data ([0..1] for
floats or [0..255] for integers).
```

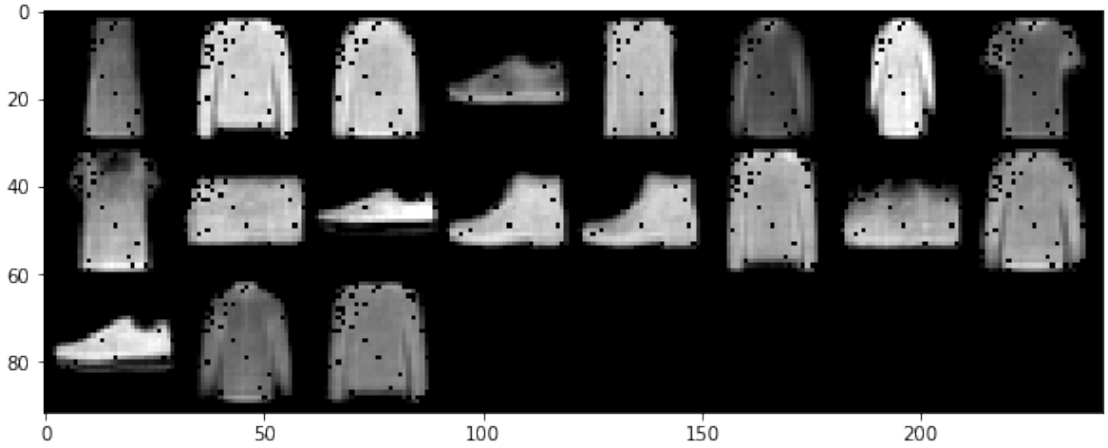


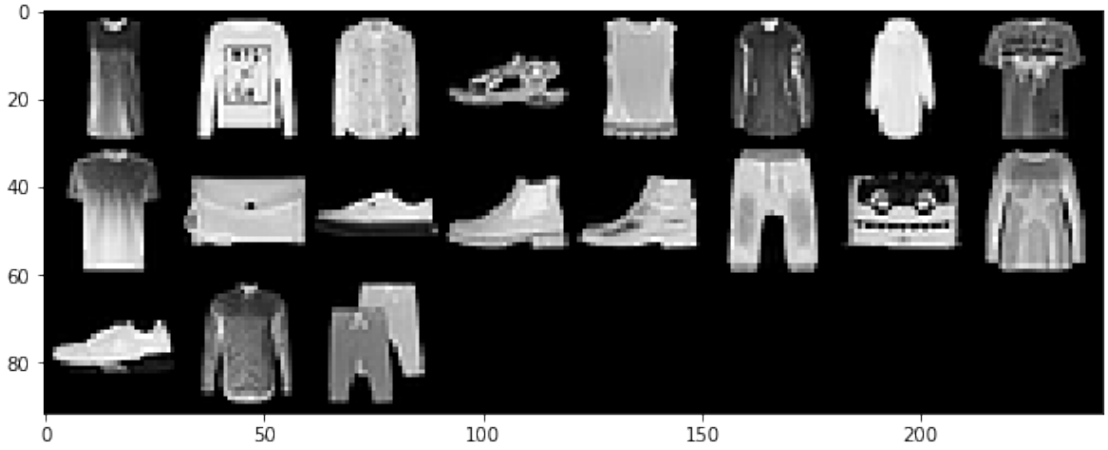



```
Epoch 4
-------------------------------
loss: 0.034060  [    0/60000]
loss: 0.032846  [ 2000/60000]
loss: 0.029287  [ 4000/60000]
loss: 0.030928  [ 6000/60000]
loss: 0.035637  [ 8000/60000]
loss: 0.032839  [10000/60000]
loss: 0.022392  [12000/60000]
loss: 0.030029  [14000/60000]
loss: 0.028122  [16000/60000]
loss: 0.028625  [18000/60000]
loss: 0.028614  [20000/60000]
loss: 0.030194  [22000/60000]
loss: 0.030133  [24000/60000]
loss: 0.023145  [26000/60000]
loss: 0.025595  [28000/60000]
loss: 0.025010  [30000/60000]
loss: 0.025825  [32000/60000]
loss: 0.029084  [34000/60000]
loss: 0.026619  [36000/60000]
loss: 0.025824  [38000/60000]
loss: 0.033863  [40000/60000]
loss: 0.026065  [42000/60000]
loss: 0.027539  [44000/60000]
loss: 0.028956  [46000/60000]
loss: 0.027586  [48000/60000]
loss: 0.033614  [50000/60000]
loss: 0.030132  [52000/60000]
loss: 0.032789  [54000/60000]
loss: 0.029327  [56000/60000]
```

```
loss: 0.028483  [58000/60000]
```

```
Clipping input data to the valid range for imshow with RGB data ([0..1] for
floats or [0..255] for integers).
```

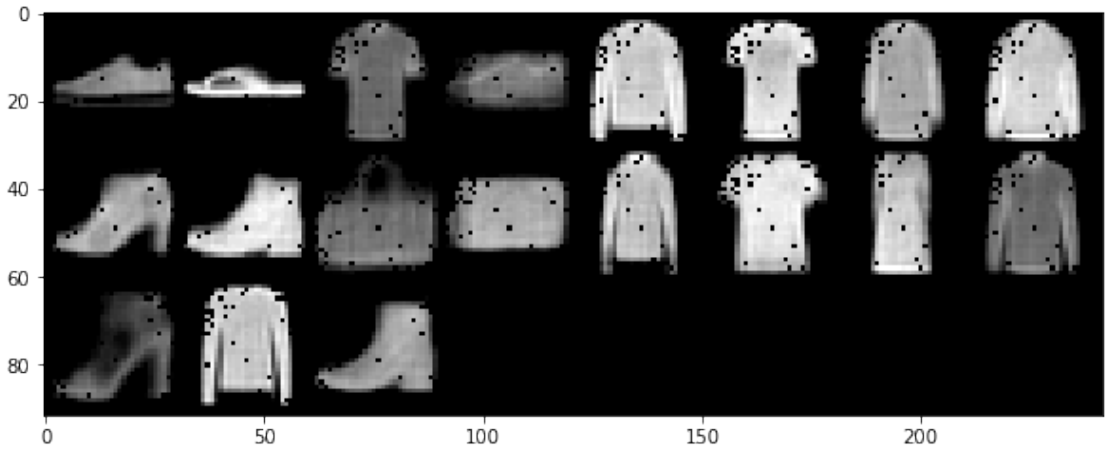


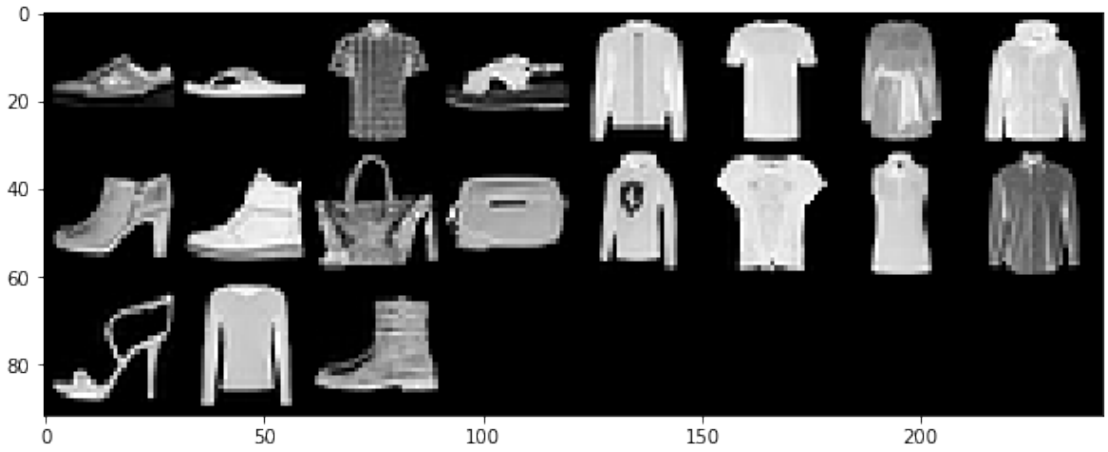




```
Epoch 5
-------------------------------
loss: 0.035256  [    0/60000]
loss: 0.027384  [ 2000/60000]
loss: 0.028907  [ 4000/60000]
loss: 0.028388  [ 6000/60000]
loss: 0.030026  [ 8000/60000]
loss: 0.030815  [10000/60000]
loss: 0.022639  [12000/60000]
loss: 0.026696  [14000/60000]
loss: 0.026019  [16000/60000]
```

```
loss: 0.025647  [18000/60000]
loss: 0.023991  [20000/60000]
loss: 0.028975  [22000/60000]
loss: 0.024594  [24000/60000]
loss: 0.028410  [26000/60000]
loss: 0.022159  [28000/60000]
loss: 0.034530  [30000/60000]
loss: 0.023504  [32000/60000]
loss: 0.023308  [34000/60000]
loss: 0.026284  [36000/60000]
loss: 0.032946  [38000/60000]
loss: 0.031572  [40000/60000]
loss: 0.021836  [42000/60000]
loss: 0.027742  [44000/60000]
loss: 0.020884  [46000/60000]
loss: 0.021634  [48000/60000]
loss: 0.025180  [50000/60000]
loss: 0.025904  [52000/60000]
loss: 0.025265  [54000/60000]
loss: 0.029393  [56000/60000]
loss: 0.029835  [58000/60000]
```

```
Clipping input data to the valid range for imshow with RGB data ([0..1] for
floats or [0..255] for integers).
```

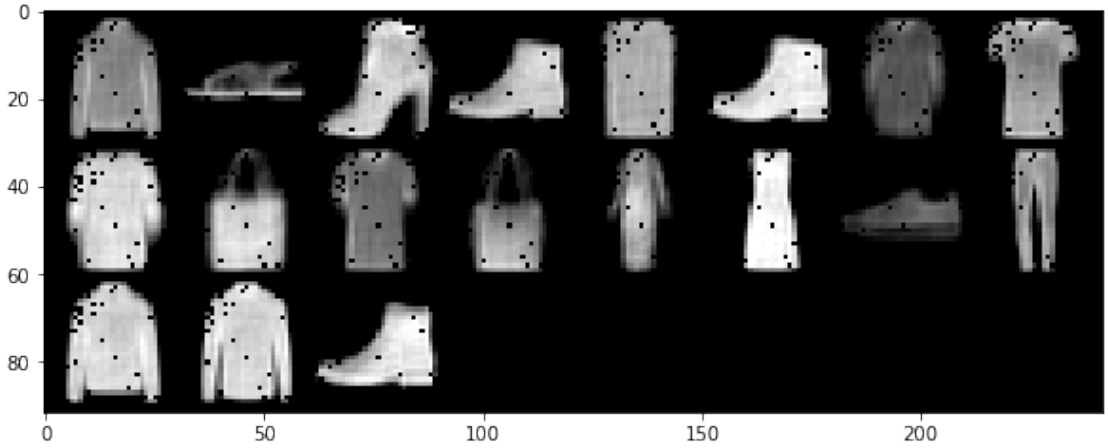


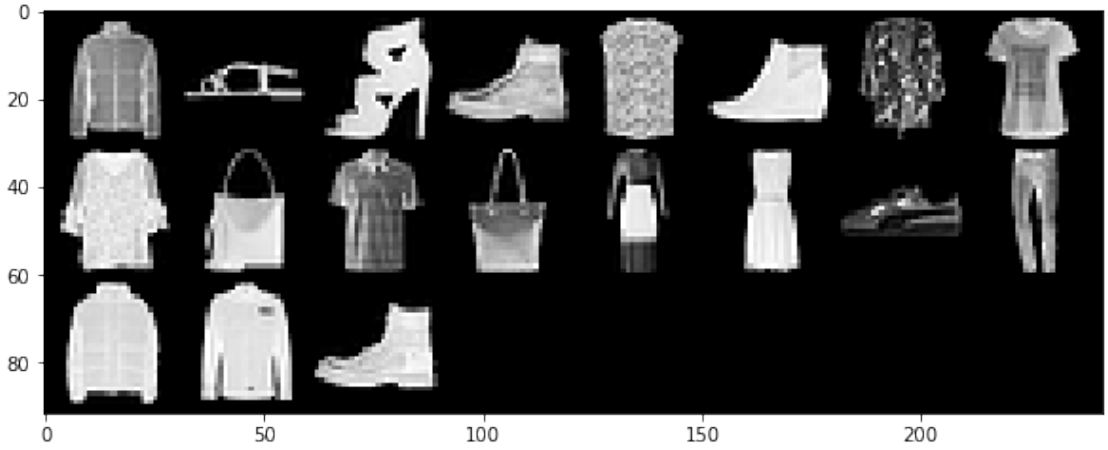



```
Epoch 6
-------------------------------
loss: 0.021054  [    0/60000]
loss: 0.023184  [ 2000/60000]
loss: 0.022062  [ 4000/60000]
loss: 0.027175  [ 6000/60000]
loss: 0.021638  [ 8000/60000]
loss: 0.027894  [10000/60000]
loss: 0.033709  [12000/60000]
loss: 0.023465  [14000/60000]
loss: 0.029766  [16000/60000]
loss: 0.023322  [18000/60000]
loss: 0.027227  [20000/60000]
loss: 0.024504  [22000/60000]
loss: 0.023582  [24000/60000]
loss: 0.018874  [26000/60000]
loss: 0.025584  [28000/60000]
loss: 0.022851  [30000/60000]
loss: 0.030582  [32000/60000]
loss: 0.023798  [34000/60000]
loss: 0.024023  [36000/60000]
loss: 0.023399  [38000/60000]
loss: 0.030700  [40000/60000]
loss: 0.023372  [42000/60000]
loss: 0.022106  [44000/60000]
loss: 0.030943  [46000/60000]
loss: 0.021335  [48000/60000]
loss: 0.029395  [50000/60000]
loss: 0.029763  [52000/60000]
loss: 0.028427  [54000/60000]
loss: 0.028166  [56000/60000]
```

```
loss: 0.023195  [58000/60000]
```

```
Clipping input data to the valid range for imshow with RGB data ([0..1] for
floats or [0..255] for integers).
```

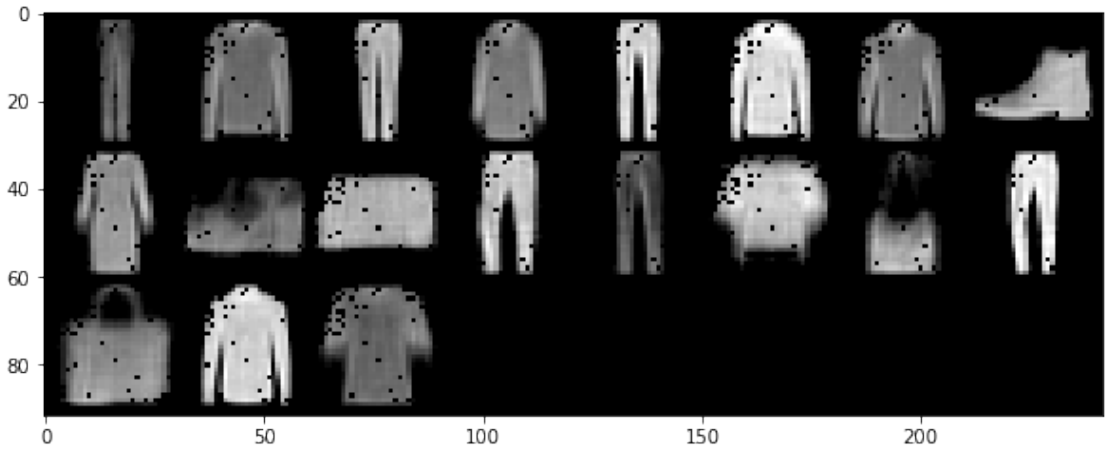


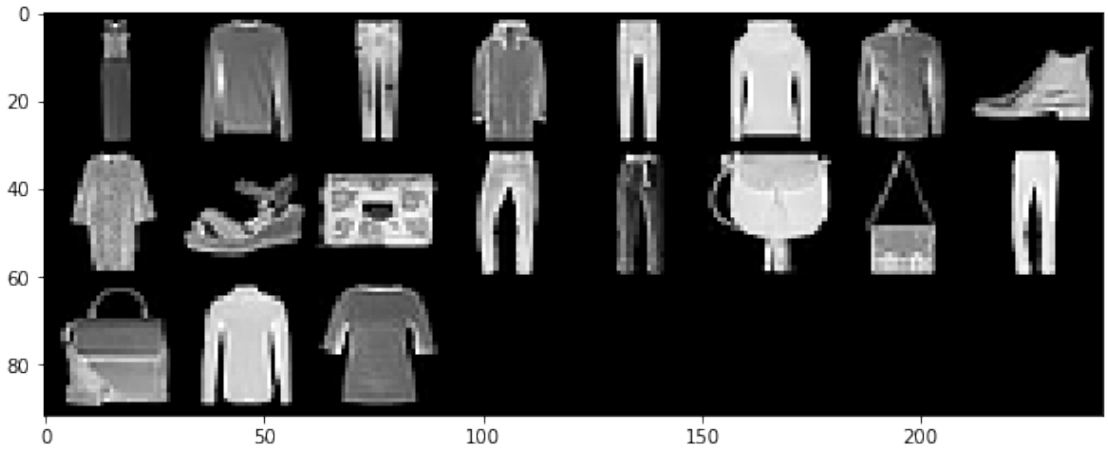




```
Epoch 7
-------------------------------
loss: 0.025378  [    0/60000]
loss: 0.024401  [ 2000/60000]
loss: 0.027090  [ 4000/60000]
loss: 0.025220  [ 6000/60000]
loss: 0.022628  [ 8000/60000]
loss: 0.026051  [10000/60000]
loss: 0.025213  [12000/60000]
loss: 0.023286  [14000/60000]
loss: 0.029799  [16000/60000]
```

```
loss: 0.024283  [18000/60000]
loss: 0.022422  [20000/60000]
loss: 0.028861  [22000/60000]
loss: 0.028137  [24000/60000]
loss: 0.022924  [26000/60000]
loss: 0.025808  [28000/60000]
loss: 0.025611  [30000/60000]
loss: 0.029543  [32000/60000]
loss: 0.020559  [34000/60000]
loss: 0.026421  [36000/60000]
loss: 0.024773  [38000/60000]
loss: 0.030364  [40000/60000]
loss: 0.024767  [42000/60000]
loss: 0.023114  [44000/60000]
loss: 0.026334  [46000/60000]
loss: 0.022024  [48000/60000]
loss: 0.024665  [50000/60000]
loss: 0.028681  [52000/60000]
loss: 0.021577  [54000/60000]
loss: 0.026708  [56000/60000]
loss: 0.023996  [58000/60000]
```

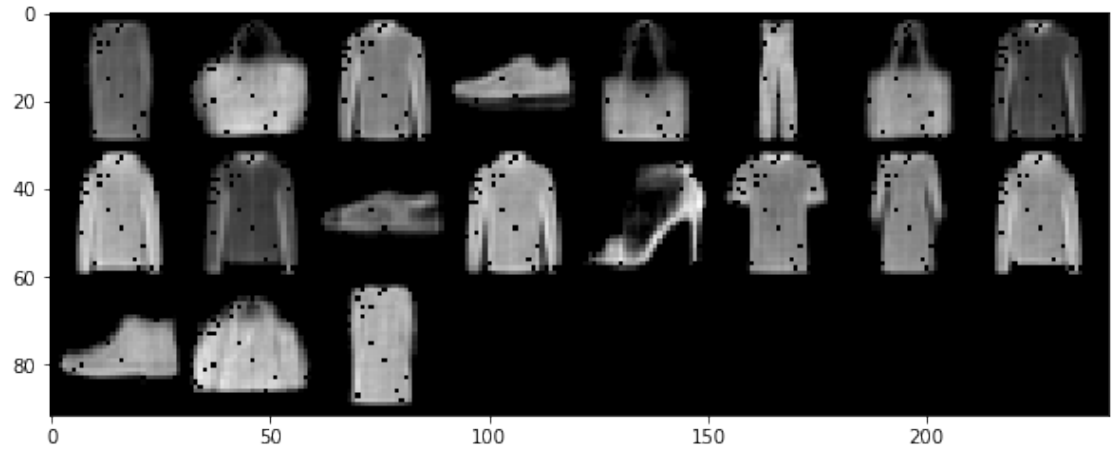
```
Clipping input data to the valid range for imshow with RGB data ([0..1] for
floats or [0..255] for integers).
```

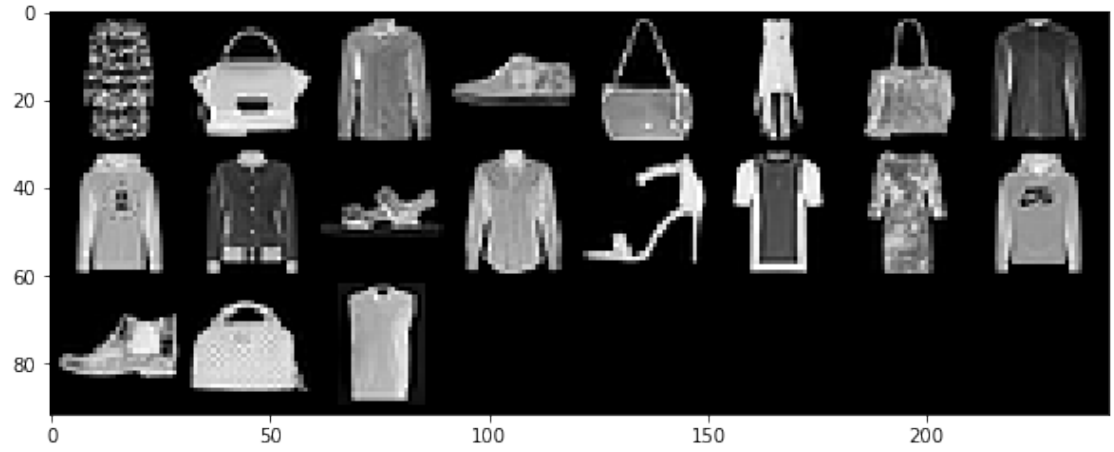





```
Epoch 8
-------------------------------
loss: 0.031895  [    0/60000]
loss: 0.028209  [ 2000/60000]
loss: 0.027662  [ 4000/60000]
loss: 0.027049  [ 6000/60000]
loss: 0.029013  [ 8000/60000]
loss: 0.022144  [10000/60000]
loss: 0.025314  [12000/60000]
loss: 0.024726  [14000/60000]
loss: 0.025141  [16000/60000]
loss: 0.018459  [18000/60000]
loss: 0.021444  [20000/60000]
loss: 0.023956  [22000/60000]
loss: 0.028628  [24000/60000]
loss: 0.024303  [26000/60000]
loss: 0.026596  [28000/60000]
loss: 0.027155  [30000/60000]
loss: 0.026925  [32000/60000]
loss: 0.028573  [34000/60000]
loss: 0.029556  [36000/60000]
loss: 0.024174  [38000/60000]
loss: 0.029469  [40000/60000]
loss: 0.025039  [42000/60000]
loss: 0.026329  [44000/60000]
loss: 0.025980  [46000/60000]
loss: 0.022778  [48000/60000]
loss: 0.027704  [50000/60000]
loss: 0.029671  [52000/60000]
loss: 0.023024  [54000/60000]
loss: 0.021338  [56000/60000]
```

```
loss: 0.021682  [58000/60000]
```

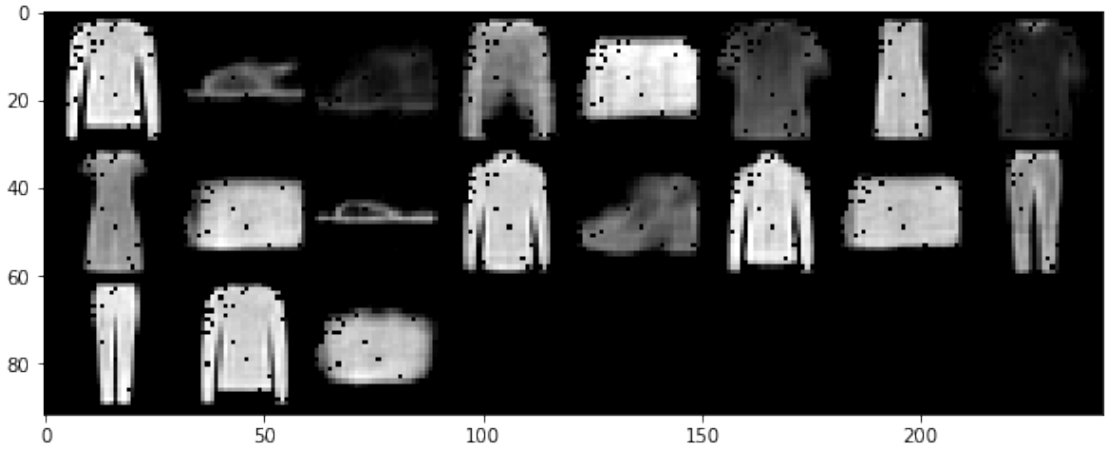


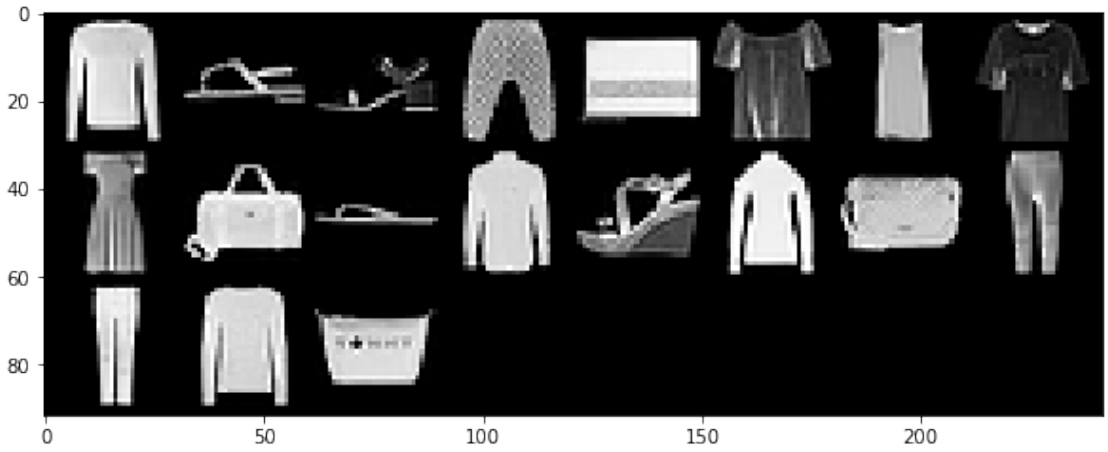




```
Epoch 9
-------------------------------
loss: 0.025304  [    0/60000]
loss: 0.024287  [ 2000/60000]
loss: 0.029313  [ 4000/60000]
loss: 0.025266  [ 6000/60000]
loss: 0.027315  [ 8000/60000]
loss: 0.025048  [10000/60000]
loss: 0.030731  [12000/60000]
loss: 0.023652  [14000/60000]
loss: 0.027232  [16000/60000]
loss: 0.020292  [18000/60000]
loss: 0.026228  [20000/60000]
loss: 0.022230  [22000/60000]
```

```
loss: 0.023608  [24000/60000]
loss: 0.025557  [26000/60000]
loss: 0.023935  [28000/60000]
loss: 0.025098  [30000/60000]
loss: 0.022888  [32000/60000]
loss: 0.025729  [34000/60000]
loss: 0.026705  [36000/60000]
loss: 0.023930  [38000/60000]
loss: 0.021936  [40000/60000]
loss: 0.026107  [42000/60000]
loss: 0.023152  [44000/60000]
loss: 0.026240  [46000/60000]
loss: 0.021213  [48000/60000]
loss: 0.034330  [50000/60000]
loss: 0.023427  [52000/60000]
loss: 0.023649  [54000/60000]
loss: 0.022452  [56000/60000]
loss: 0.022797  [58000/60000]
```

```
Clipping input data to the valid range for imshow with RGB data ([0..1] for
floats or [0..255] for integers).
```

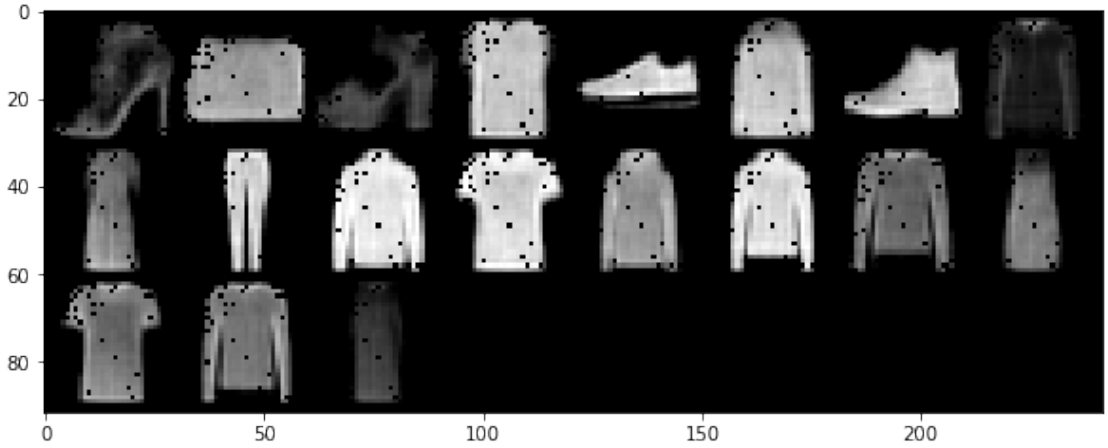


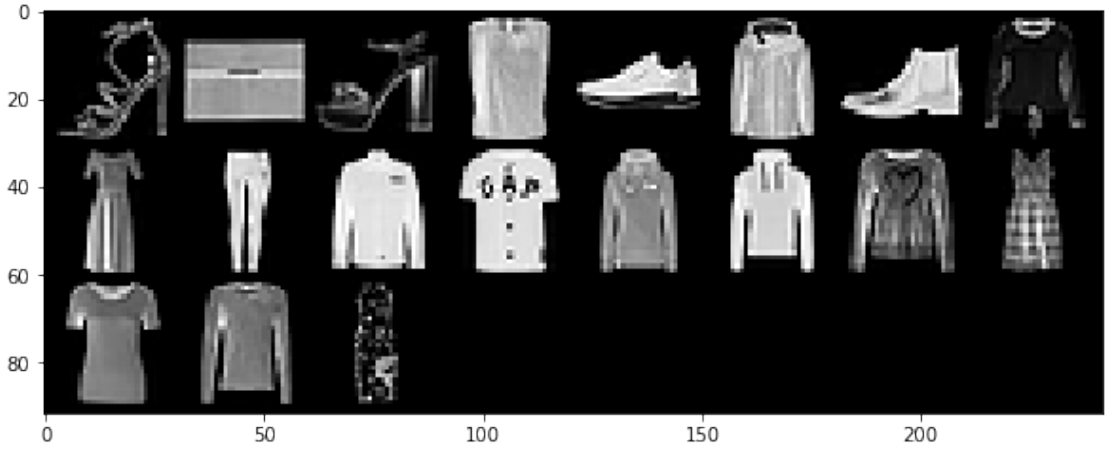



```
Epoch 10
-------------------------------
loss: 0.023576  [    0/60000]
loss: 0.029186  [ 2000/60000]
loss: 0.021372  [ 4000/60000]
loss: 0.020742  [ 6000/60000]
loss: 0.022265  [ 8000/60000]
loss: 0.022207  [10000/60000]
loss: 0.021140  [12000/60000]
loss: 0.026746  [14000/60000]
loss: 0.026350  [16000/60000]
loss: 0.021825  [18000/60000]
loss: 0.023388  [20000/60000]
loss: 0.026253  [22000/60000]
loss: 0.026019  [24000/60000]
loss: 0.026197  [26000/60000]
loss: 0.022887  [28000/60000]
loss: 0.020524  [30000/60000]
loss: 0.022532  [32000/60000]
loss: 0.022739  [34000/60000]
loss: 0.025715  [36000/60000]
loss: 0.025056  [38000/60000]
loss: 0.023100  [40000/60000]
loss: 0.021090  [42000/60000]
loss: 0.022591  [44000/60000]
loss: 0.023859  [46000/60000]
loss: 0.024683  [48000/60000]
loss: 0.022937  [50000/60000]
loss: 0.024772  [52000/60000]
loss: 0.020063  [54000/60000]
loss: 0.024558  [56000/60000]
```

```
loss: 0.026947  [58000/60000]
```

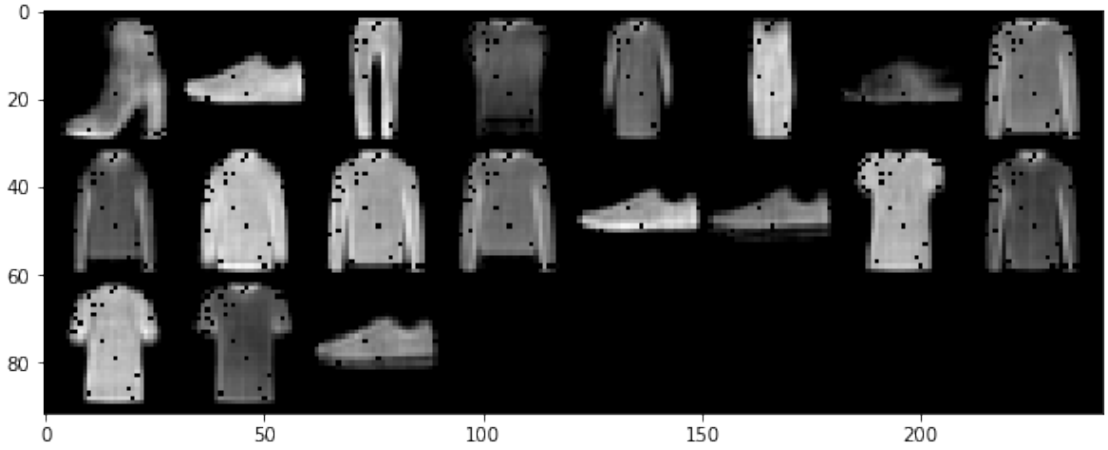
```
Clipping input data to the valid range for imshow with RGB data ([0..1] for
floats or [0..255] for integers).
```

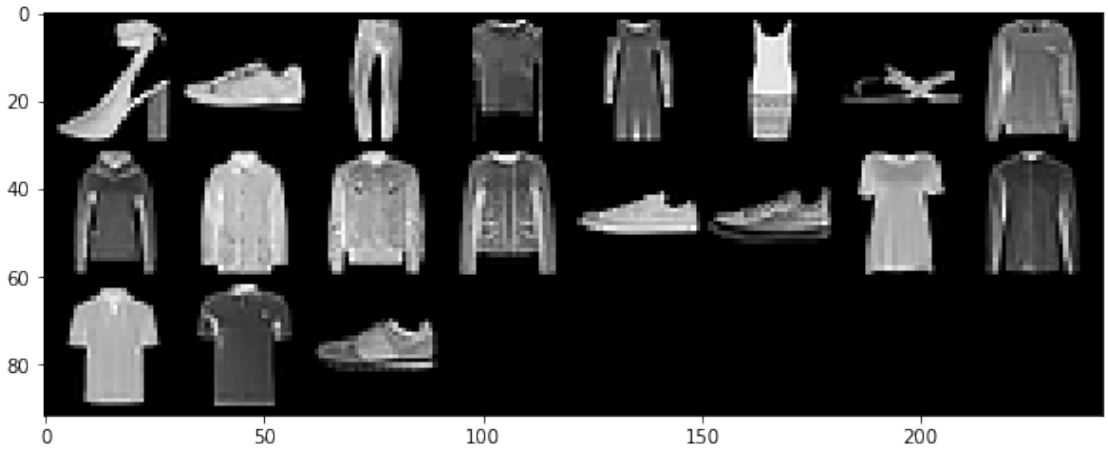



```
Epoch 11
-------------------------------
loss: 0.018467  [    0/60000]
loss: 0.031149  [ 2000/60000]
loss: 0.022506  [ 4000/60000]
loss: 0.020029  [ 6000/60000]
loss: 0.021655  [ 8000/60000]
loss: 0.028726  [10000/60000]
loss: 0.026683  [12000/60000]
loss: 0.026515  [14000/60000]
loss: 0.023883  [16000/60000]
```

```
loss: 0.022716  [18000/60000]
loss: 0.020458  [20000/60000]
loss: 0.020192  [22000/60000]
loss: 0.025270  [24000/60000]
loss: 0.024837  [26000/60000]
loss: 0.029189  [28000/60000]
loss: 0.032403  [30000/60000]
loss: 0.018371  [32000/60000]
loss: 0.023419  [34000/60000]
loss: 0.024040  [36000/60000]
loss: 0.023054  [38000/60000]
loss: 0.022205  [40000/60000]
loss: 0.028117  [42000/60000]
loss: 0.021075  [44000/60000]
loss: 0.026784  [46000/60000]
loss: 0.022800  [48000/60000]
loss: 0.026799  [50000/60000]
loss: 0.022490  [52000/60000]
loss: 0.025412  [54000/60000]
loss: 0.021645  [56000/60000]
loss: 0.027179  [58000/60000]
```

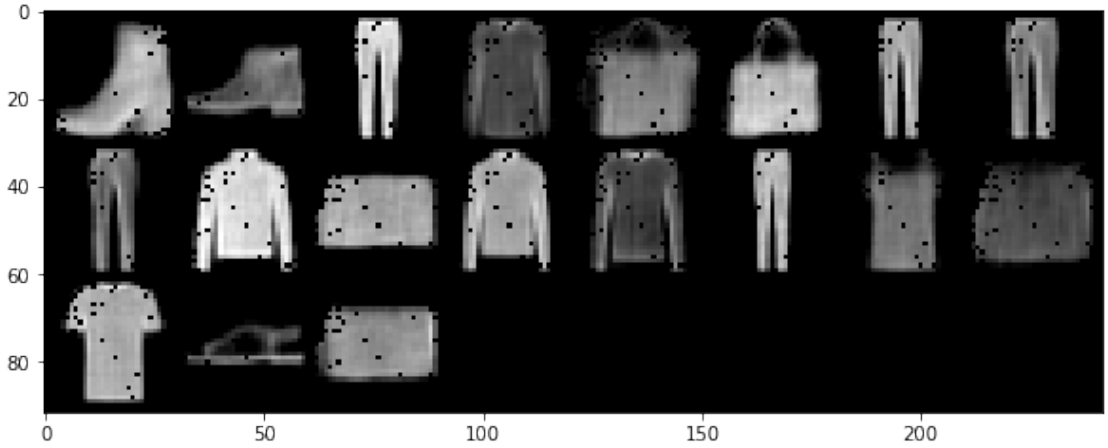


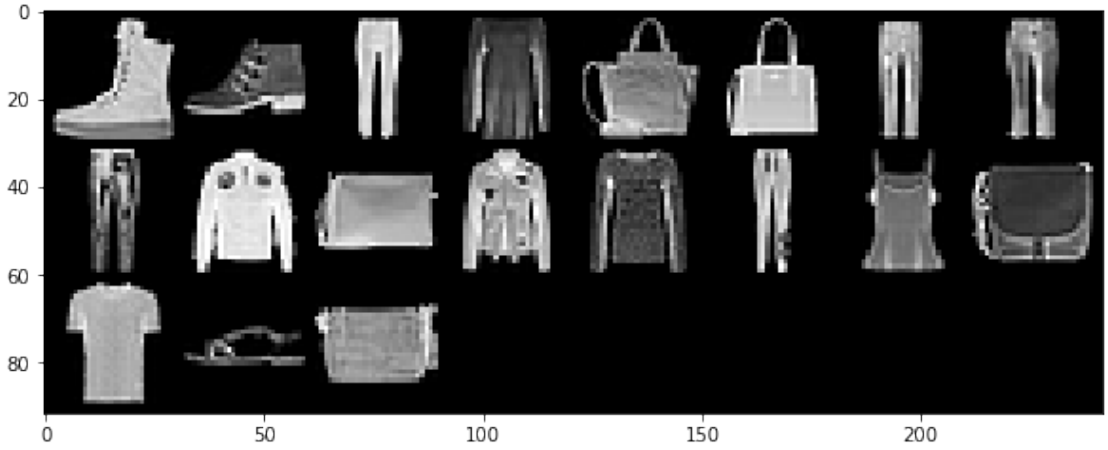



```
Epoch 12
-------------------------------
loss: 0.023322  [    0/60000]
loss: 0.020005  [ 2000/60000]
loss: 0.020314  [ 4000/60000]
loss: 0.029388  [ 6000/60000]
loss: 0.024259  [ 8000/60000]
loss: 0.025065  [10000/60000]
loss: 0.019326  [12000/60000]
loss: 0.024645  [14000/60000]
loss: 0.019306  [16000/60000]
loss: 0.025927  [18000/60000]
loss: 0.024100  [20000/60000]
loss: 0.020407  [22000/60000]
loss: 0.019021  [24000/60000]
loss: 0.022557  [26000/60000]
loss: 0.025768  [28000/60000]
loss: 0.020560  [30000/60000]
loss: 0.027895  [32000/60000]
loss: 0.024811  [34000/60000]
loss: 0.023873  [36000/60000]
loss: 0.024194  [38000/60000]
loss: 0.024393  [40000/60000]
loss: 0.019209  [42000/60000]
loss: 0.020802  [44000/60000]
loss: 0.018255  [46000/60000]
loss: 0.019352  [48000/60000]
loss: 0.022334  [50000/60000]
loss: 0.025974  [52000/60000]
loss: 0.020772  [54000/60000]
loss: 0.021313  [56000/60000]
```

```
loss: 0.033079  [58000/60000]
```

```
Clipping input data to the valid range for imshow with RGB data ([0..1] for
floats or [0..255] for integers).
```

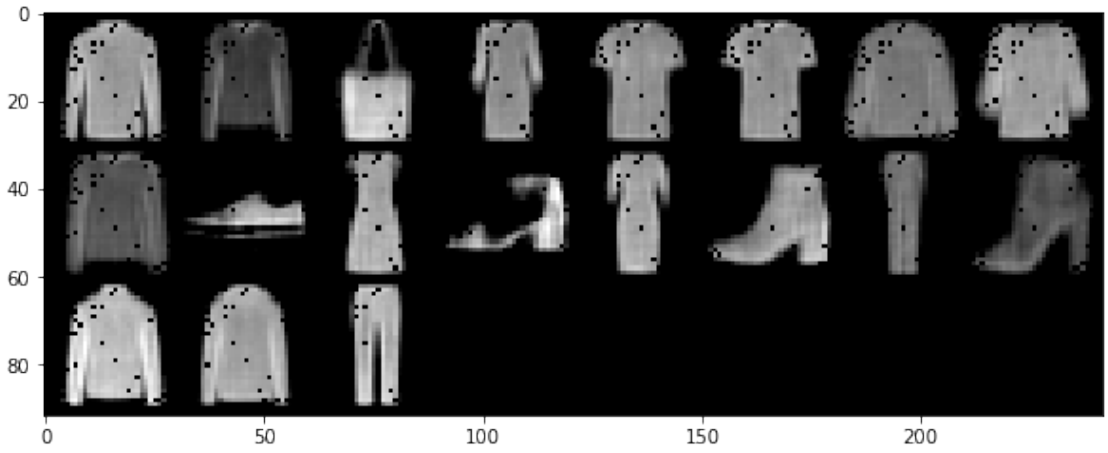


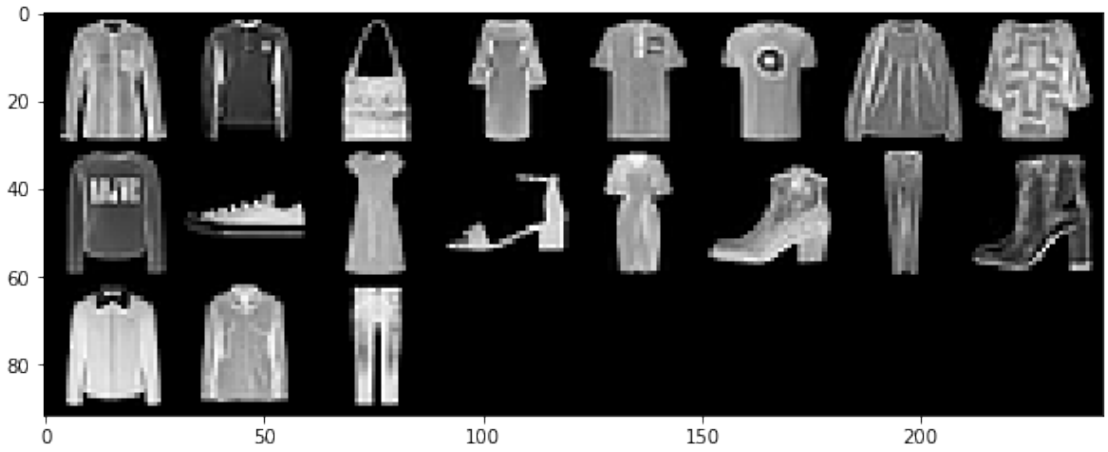




```
Epoch 13
-------------------------------
loss: 0.022265  [    0/60000]
loss: 0.017451  [ 2000/60000]
loss: 0.023229  [ 4000/60000]
loss: 0.023635  [ 6000/60000]
loss: 0.025277  [ 8000/60000]
loss: 0.024382  [10000/60000]
loss: 0.021704  [12000/60000]
loss: 0.018963  [14000/60000]
loss: 0.024145  [16000/60000]
```

```
loss: 0.019780  [18000/60000]
loss: 0.020668  [20000/60000]
loss: 0.022427  [22000/60000]
loss: 0.025930  [24000/60000]
loss: 0.020624  [26000/60000]
loss: 0.024596  [28000/60000]
loss: 0.016319  [30000/60000]
loss: 0.020708  [32000/60000]
loss: 0.026303  [34000/60000]
loss: 0.019630  [36000/60000]
loss: 0.021499  [38000/60000]
loss: 0.026976  [40000/60000]
loss: 0.025345  [42000/60000]
loss: 0.022611  [44000/60000]
loss: 0.026223  [46000/60000]
loss: 0.022832  [48000/60000]
loss: 0.027757  [50000/60000]
loss: 0.026637  [52000/60000]
loss: 0.025530  [54000/60000]
loss: 0.019185  [56000/60000]
loss: 0.020581  [58000/60000]
```

```
Clipping input data to the valid range for imshow with RGB data ([0..1] for
floats or [0..255] for integers).
```

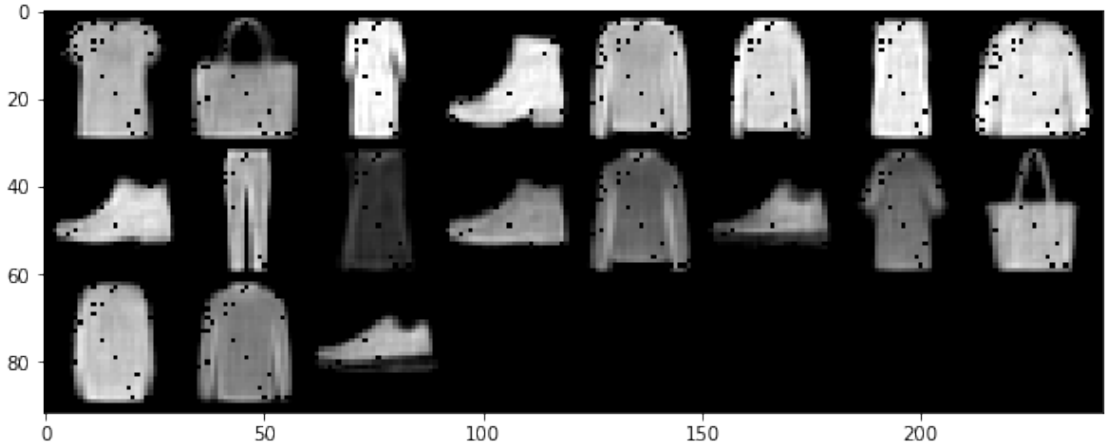


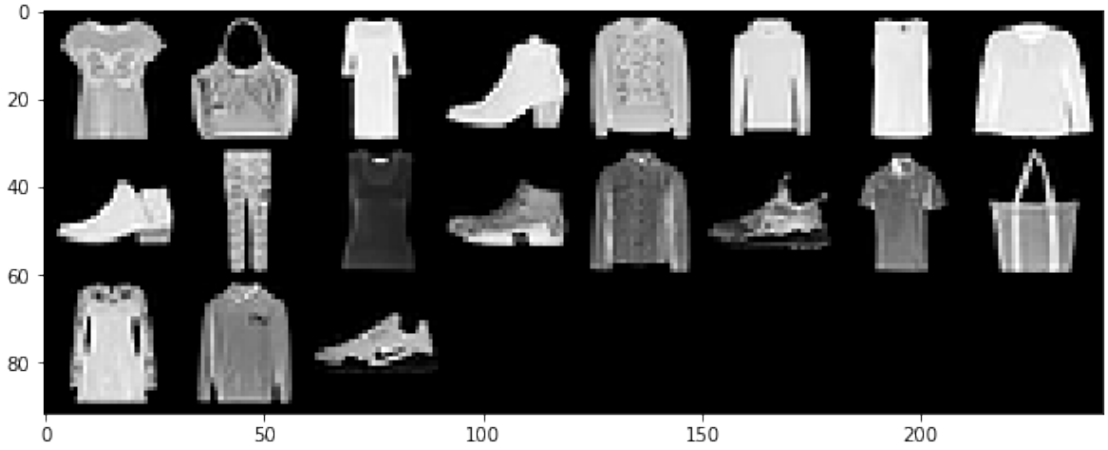



```
Epoch 14
-------------------------------
loss: 0.023956  [    0/60000]
loss: 0.021337  [ 2000/60000]
loss: 0.025904  [ 4000/60000]
loss: 0.024676  [ 6000/60000]
loss: 0.026619  [ 8000/60000]
loss: 0.022056  [10000/60000]
loss: 0.021847  [12000/60000]
loss: 0.021369  [14000/60000]
loss: 0.020373  [16000/60000]
loss: 0.025601  [18000/60000]
loss: 0.022009  [20000/60000]
loss: 0.023960  [22000/60000]
loss: 0.019851  [24000/60000]
loss: 0.023801  [26000/60000]
loss: 0.024724  [28000/60000]
loss: 0.017547  [30000/60000]
loss: 0.026178  [32000/60000]
loss: 0.021982  [34000/60000]
loss: 0.028675  [36000/60000]
loss: 0.022356  [38000/60000]
loss: 0.020443  [40000/60000]
loss: 0.021172  [42000/60000]
loss: 0.023647  [44000/60000]
loss: 0.027729  [46000/60000]
loss: 0.021210  [48000/60000]
loss: 0.020600  [50000/60000]
loss: 0.024517  [52000/60000]
loss: 0.028916  [54000/60000]
loss: 0.026357  [56000/60000]
```

```
loss: 0.024036  [58000/60000]
```

```
Clipping input data to the valid range for imshow with RGB data ([0..1] for
floats or [0..255] for integers).
```

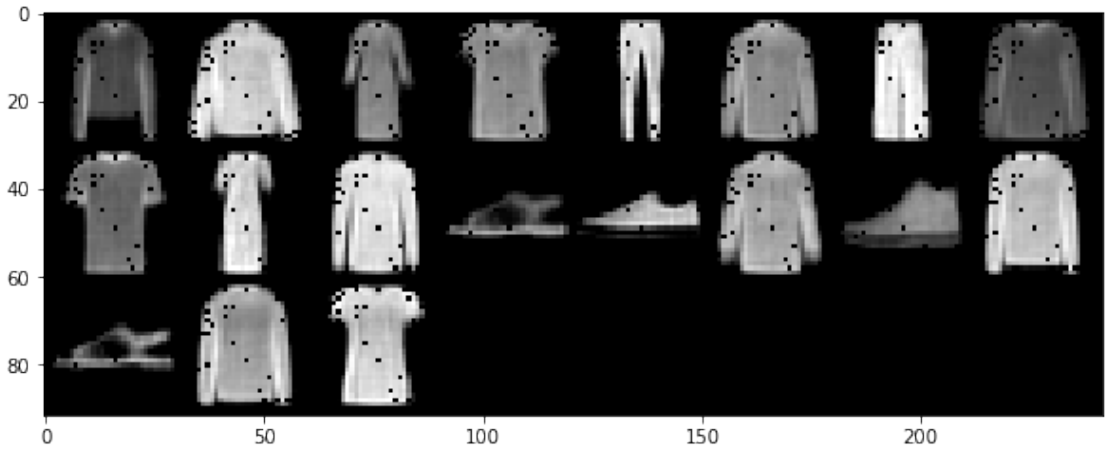


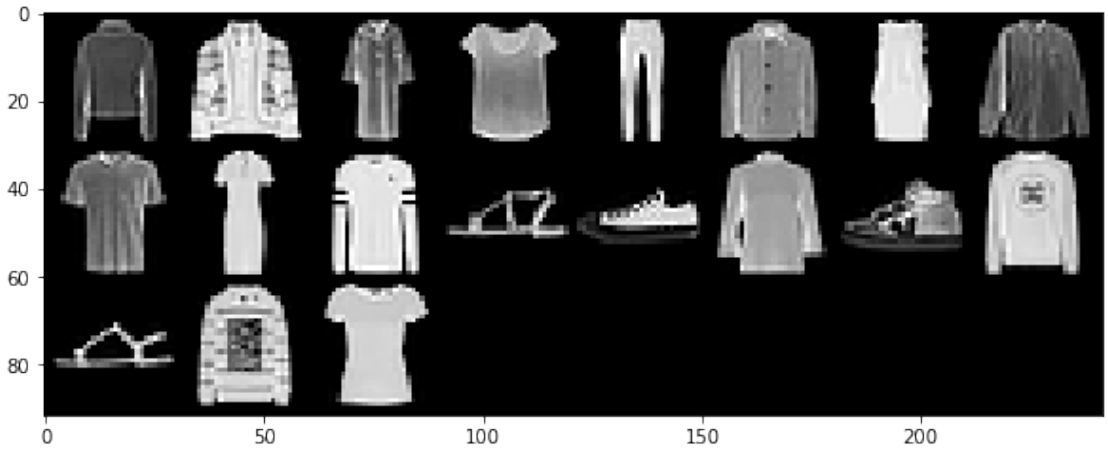




```
Epoch 15
-------------------------------
loss: 0.022814  [    0/60000]
loss: 0.026634  [ 2000/60000]
loss: 0.027422  [ 4000/60000]
loss: 0.020283  [ 6000/60000]
loss: 0.021997  [ 8000/60000]
loss: 0.021269  [10000/60000]
loss: 0.023428  [12000/60000]
loss: 0.023293  [14000/60000]
loss: 0.018492  [16000/60000]
```

```
loss: 0.024893  [18000/60000]
loss: 0.023426  [20000/60000]
loss: 0.021715  [22000/60000]
loss: 0.024915  [24000/60000]
loss: 0.019301  [26000/60000]
loss: 0.021287  [28000/60000]
loss: 0.021748  [30000/60000]
loss: 0.021278  [32000/60000]
loss: 0.023577  [34000/60000]
loss: 0.022058  [36000/60000]
loss: 0.019562  [38000/60000]
loss: 0.025405  [40000/60000]
loss: 0.021455  [42000/60000]
loss: 0.021883  [44000/60000]
loss: 0.019739  [46000/60000]
loss: 0.021858  [48000/60000]
loss: 0.016807  [50000/60000]
loss: 0.021063  [52000/60000]
loss: 0.019996  [54000/60000]
loss: 0.023567  [56000/60000]
loss: 0.028701  [58000/60000]
```

```
Clipping input data to the valid range for imshow with RGB data ([0..1] for
floats or [0..255] for integers).
```

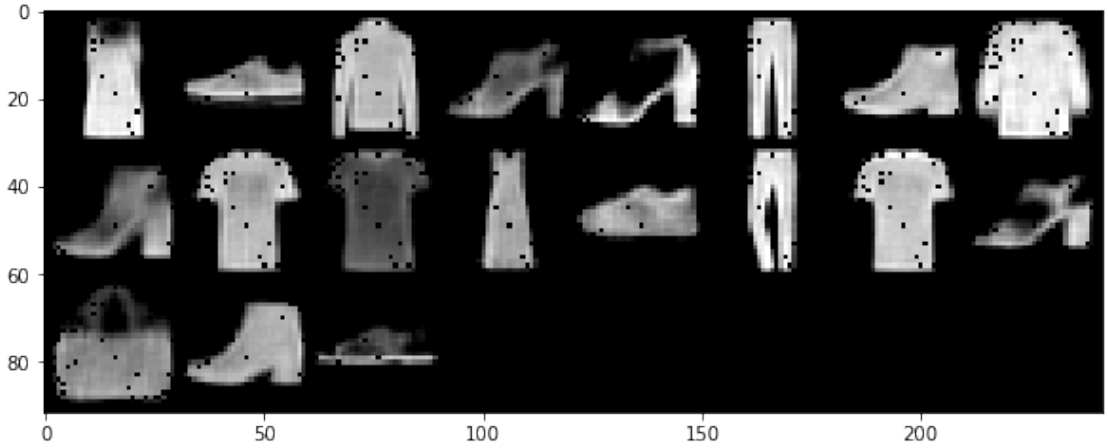


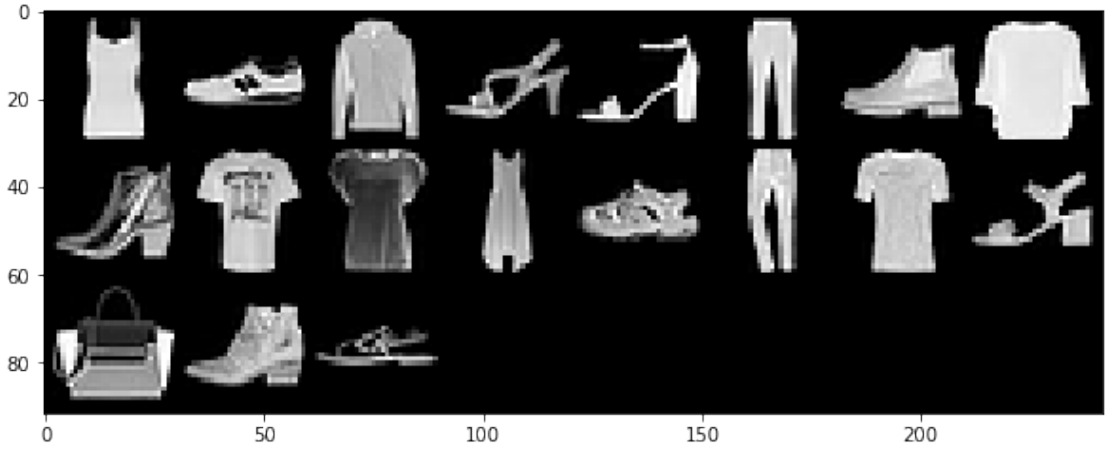



```
Epoch 16
-------------------------------
loss: 0.028799  [    0/60000]
loss: 0.026897  [ 2000/60000]
loss: 0.018647  [ 4000/60000]
loss: 0.022400  [ 6000/60000]
loss: 0.024144  [ 8000/60000]
loss: 0.022988  [10000/60000]
loss: 0.022509  [12000/60000]
loss: 0.019457  [14000/60000]
loss: 0.018453  [16000/60000]
loss: 0.018695  [18000/60000]
loss: 0.022941  [20000/60000]
loss: 0.015517  [22000/60000]
loss: 0.026541  [24000/60000]
loss: 0.024048  [26000/60000]
loss: 0.020284  [28000/60000]
loss: 0.017455  [30000/60000]
loss: 0.019456  [32000/60000]
loss: 0.020900  [34000/60000]
loss: 0.019948  [36000/60000]
loss: 0.024803  [38000/60000]
loss: 0.020237  [40000/60000]
loss: 0.019256  [42000/60000]
loss: 0.020648  [44000/60000]
loss: 0.022662  [46000/60000]
loss: 0.019323  [48000/60000]
loss: 0.023933  [50000/60000]
loss: 0.024780  [52000/60000]
loss: 0.019716  [54000/60000]
loss: 0.016630  [56000/60000]
```

```
loss: 0.020769  [58000/60000]
```

```
Clipping input data to the valid range for imshow with RGB data ([0..1] for
floats or [0..255] for integers).
```

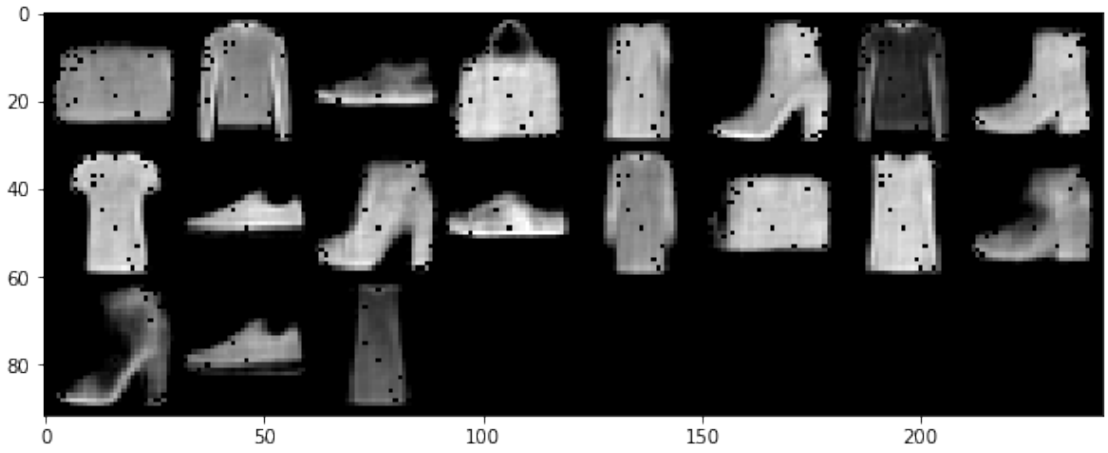


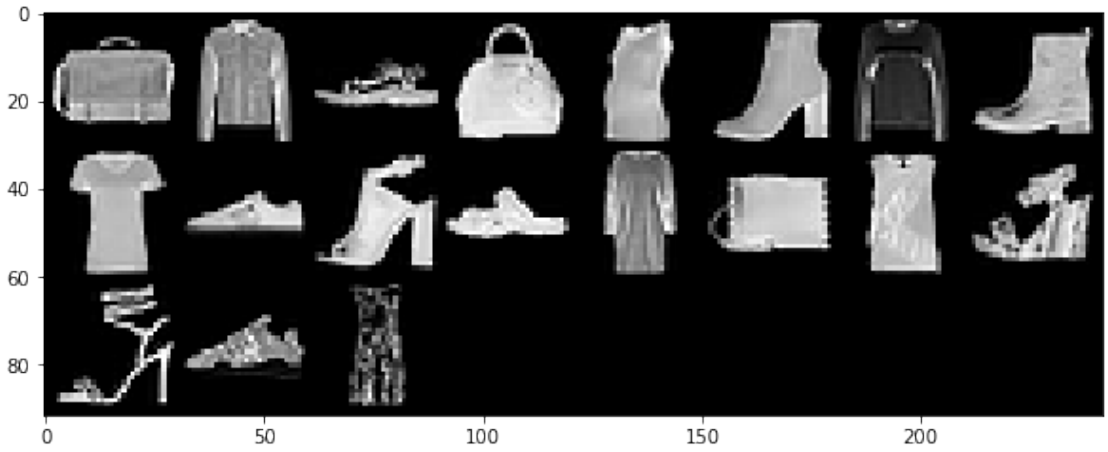




```
Epoch 17
-------------------------------
loss: 0.019580  [    0/60000]
loss: 0.021197  [ 2000/60000]
loss: 0.023372  [ 4000/60000]
loss: 0.020428  [ 6000/60000]
loss: 0.022693  [ 8000/60000]
loss: 0.024833  [10000/60000]
loss: 0.020473  [12000/60000]
loss: 0.020393  [14000/60000]
loss: 0.022313  [16000/60000]
```

```
loss: 0.021555  [18000/60000]
loss: 0.023008  [20000/60000]
loss: 0.020521  [22000/60000]
loss: 0.024333  [24000/60000]
loss: 0.023163  [26000/60000]
loss: 0.022619  [28000/60000]
loss: 0.022277  [30000/60000]
loss: 0.021637  [32000/60000]
loss: 0.019375  [34000/60000]
loss: 0.021309  [36000/60000]
loss: 0.017431  [38000/60000]
loss: 0.021916  [40000/60000]
loss: 0.022885  [42000/60000]
loss: 0.020103  [44000/60000]
loss: 0.020657  [46000/60000]
loss: 0.018510  [48000/60000]
loss: 0.026156  [50000/60000]
loss: 0.018438  [52000/60000]
loss: 0.020145  [54000/60000]
loss: 0.022538  [56000/60000]
loss: 0.019520  [58000/60000]
```

```
Clipping input data to the valid range for imshow with RGB data ([0..1] for
floats or [0..255] for integers).
```

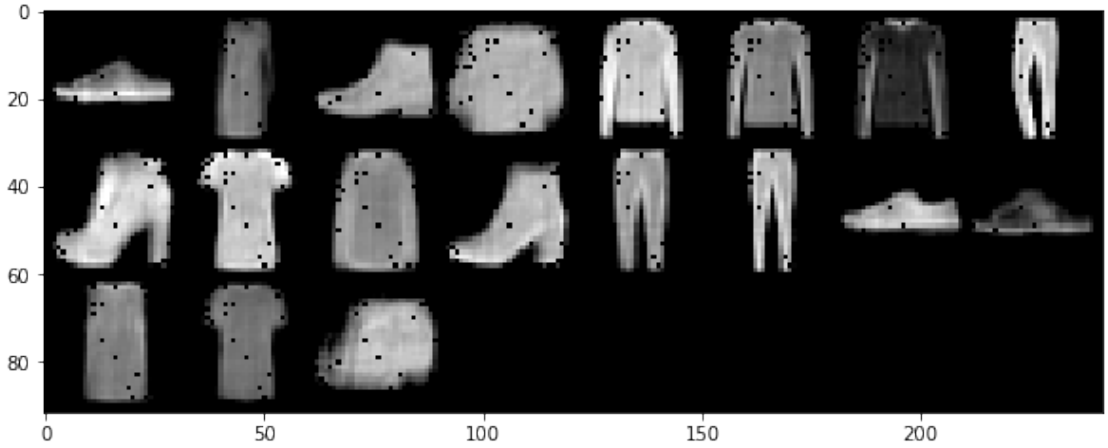


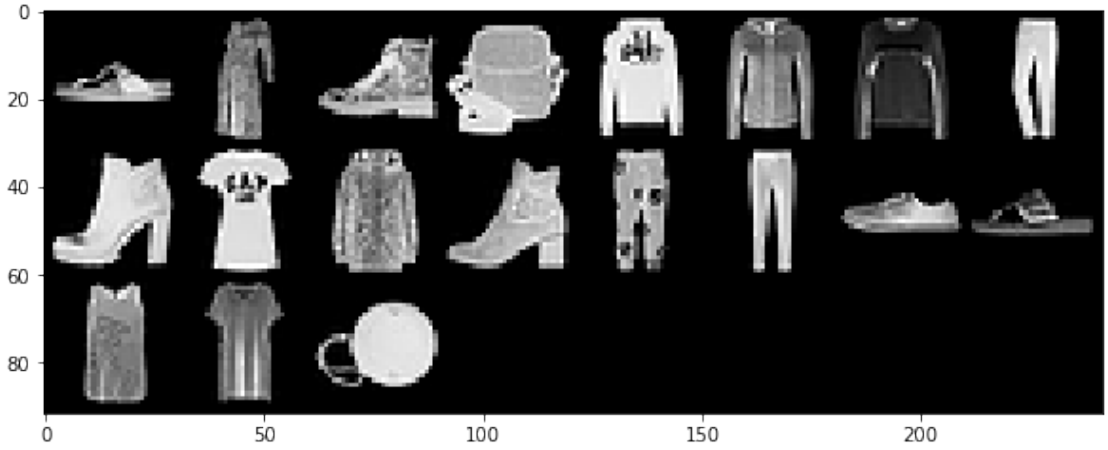



```
Epoch 18
-------------------------------
loss: 0.019089  [    0/60000]
loss: 0.022652  [ 2000/60000]
loss: 0.024257  [ 4000/60000]
loss: 0.022798  [ 6000/60000]
loss: 0.017988  [ 8000/60000]
loss: 0.024448  [10000/60000]
loss: 0.020788  [12000/60000]
loss: 0.022102  [14000/60000]
loss: 0.025712  [16000/60000]
loss: 0.017707  [18000/60000]
loss: 0.022769  [20000/60000]
loss: 0.020695  [22000/60000]
loss: 0.018510  [24000/60000]
loss: 0.022032  [26000/60000]
loss: 0.019966  [28000/60000]
loss: 0.017801  [30000/60000]
loss: 0.027880  [32000/60000]
loss: 0.019788  [34000/60000]
loss: 0.023270  [36000/60000]
loss: 0.023256  [38000/60000]
loss: 0.021230  [40000/60000]
loss: 0.021153  [42000/60000]
loss: 0.023424  [44000/60000]
loss: 0.022302  [46000/60000]
loss: 0.019543  [48000/60000]
loss: 0.017300  [50000/60000]
loss: 0.021750  [52000/60000]
loss: 0.022123  [54000/60000]
loss: 0.019410  [56000/60000]
```

```
loss: 0.027939  [58000/60000]
```

```
Clipping input data to the valid range for imshow with RGB data ([0..1] for
floats or [0..255] for integers).
```

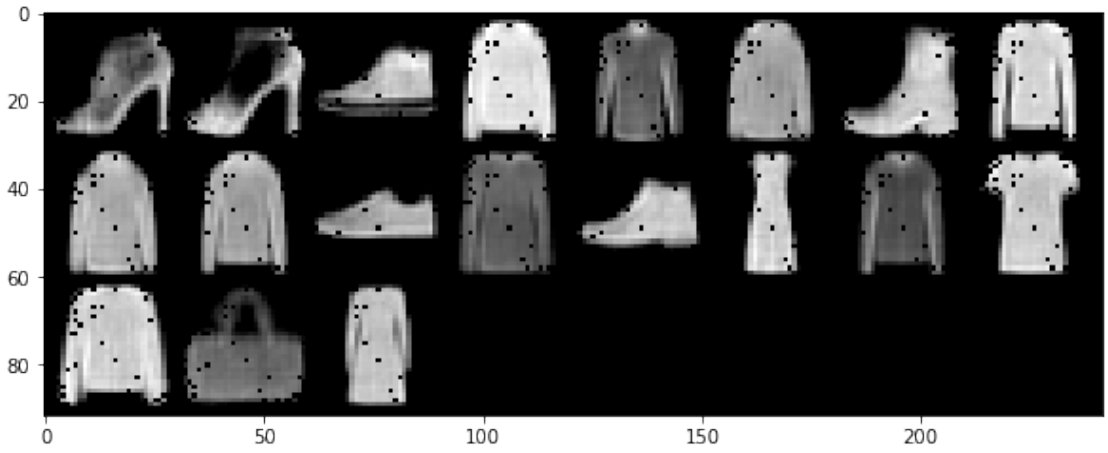


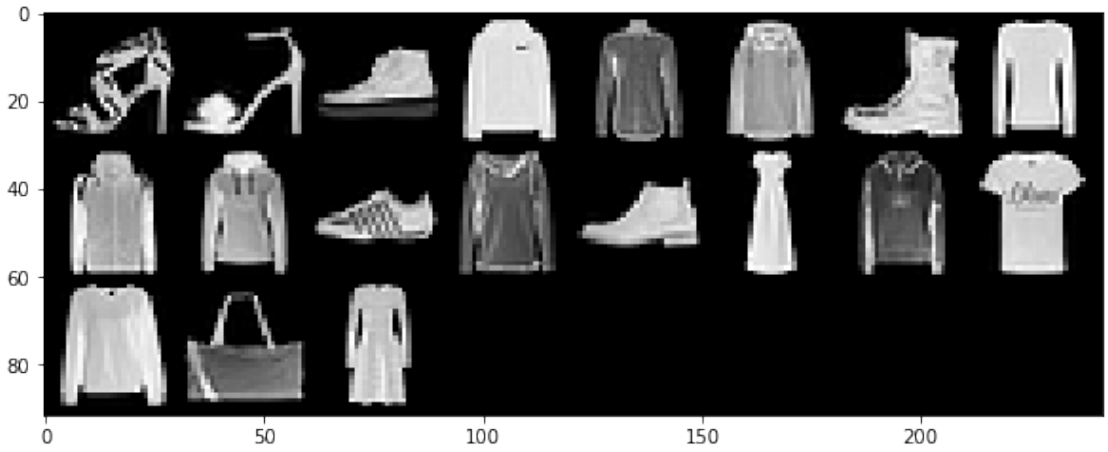




```
Epoch 19
-------------------------------
loss: 0.021250  [    0/60000]
loss: 0.016875  [ 2000/60000]
loss: 0.023632  [ 4000/60000]
loss: 0.020441  [ 6000/60000]
loss: 0.025433  [ 8000/60000]
loss: 0.019460  [10000/60000]
loss: 0.023267  [12000/60000]
loss: 0.022114  [14000/60000]
loss: 0.022654  [16000/60000]
```

```
loss: 0.020916  [18000/60000]
loss: 0.019740  [20000/60000]
loss: 0.017379  [22000/60000]
loss: 0.020337  [24000/60000]
loss: 0.020316  [26000/60000]
loss: 0.022633  [28000/60000]
loss: 0.016792  [30000/60000]
loss: 0.018153  [32000/60000]
loss: 0.021913  [34000/60000]
loss: 0.021887  [36000/60000]
loss: 0.021107  [38000/60000]
loss: 0.024988  [40000/60000]
loss: 0.023238  [42000/60000]
loss: 0.020664  [44000/60000]
loss: 0.026457  [46000/60000]
loss: 0.025328  [48000/60000]
loss: 0.021554  [50000/60000]
loss: 0.020909  [52000/60000]
loss: 0.019226  [54000/60000]
loss: 0.019272  [56000/60000]
loss: 0.019421  [58000/60000]
```

```
Clipping input data to the valid range for imshow with RGB data ([0..1] for
floats or [0..255] for integers).
```

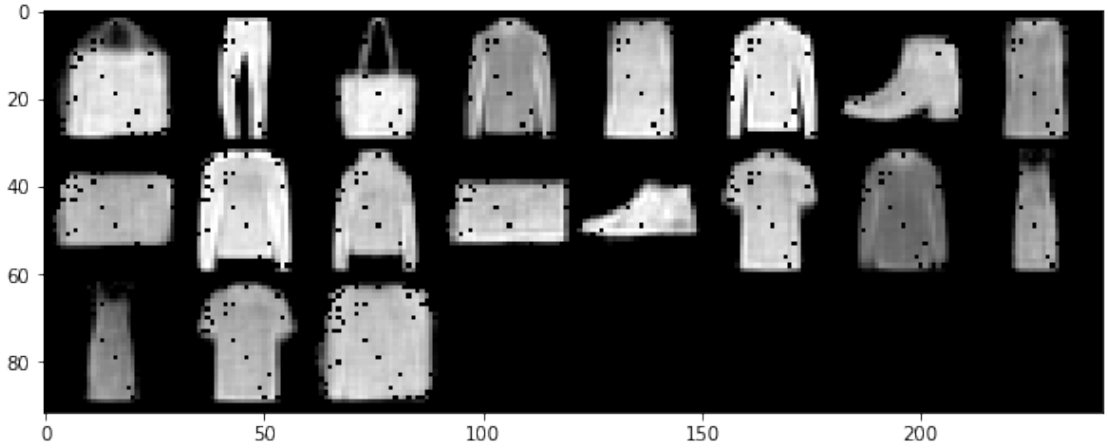


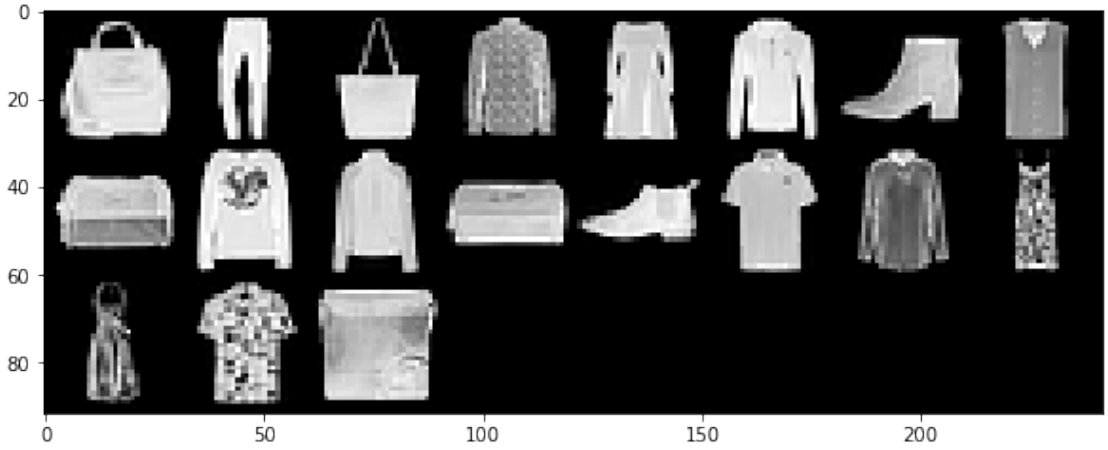



```
Epoch 20
-------------------------------
loss: 0.021731  [    0/60000]
loss: 0.021480  [ 2000/60000]
loss: 0.021695  [ 4000/60000]
loss: 0.022534  [ 6000/60000]
loss: 0.023293  [ 8000/60000]
loss: 0.025424  [10000/60000]
loss: 0.021013  [12000/60000]
loss: 0.019040  [14000/60000]
loss: 0.023896  [16000/60000]
loss: 0.023257  [18000/60000]
loss: 0.020986  [20000/60000]
loss: 0.016583  [22000/60000]
loss: 0.017099  [24000/60000]
loss: 0.020012  [26000/60000]
loss: 0.018128  [28000/60000]
loss: 0.023939  [30000/60000]
loss: 0.020062  [32000/60000]
loss: 0.017673  [34000/60000]
loss: 0.023062  [36000/60000]
loss: 0.026980  [38000/60000]
loss: 0.020066  [40000/60000]
loss: 0.023125  [42000/60000]
loss: 0.021361  [44000/60000]
loss: 0.018504  [46000/60000]
loss: 0.024693  [48000/60000]
loss: 0.020919  [50000/60000]
loss: 0.015643  [52000/60000]
loss: 0.021666  [54000/60000]
loss: 0.018729  [56000/60000]
```

```
loss: 0.023030  [58000/60000]
```

```
Clipping input data to the valid range for imshow with RGB data ([0..1] for
floats or [0..255] for integers).
```

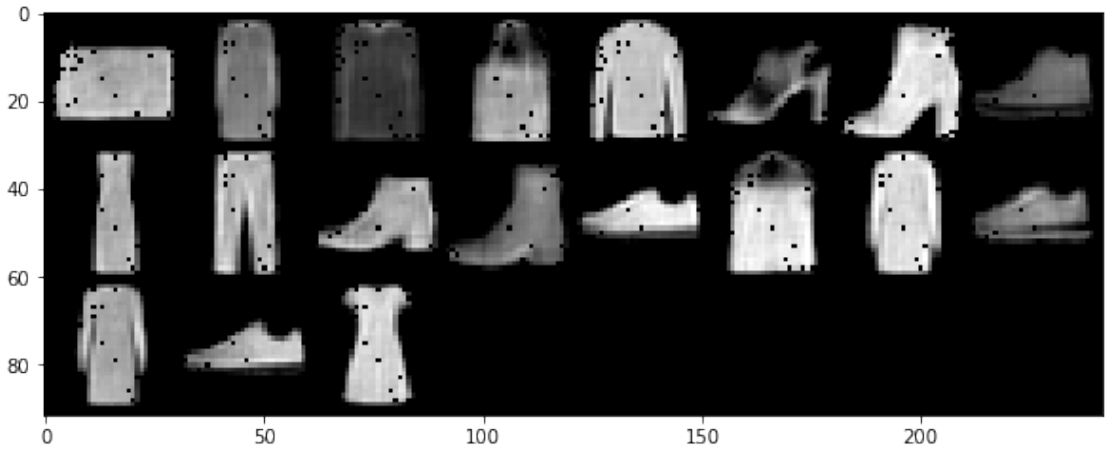


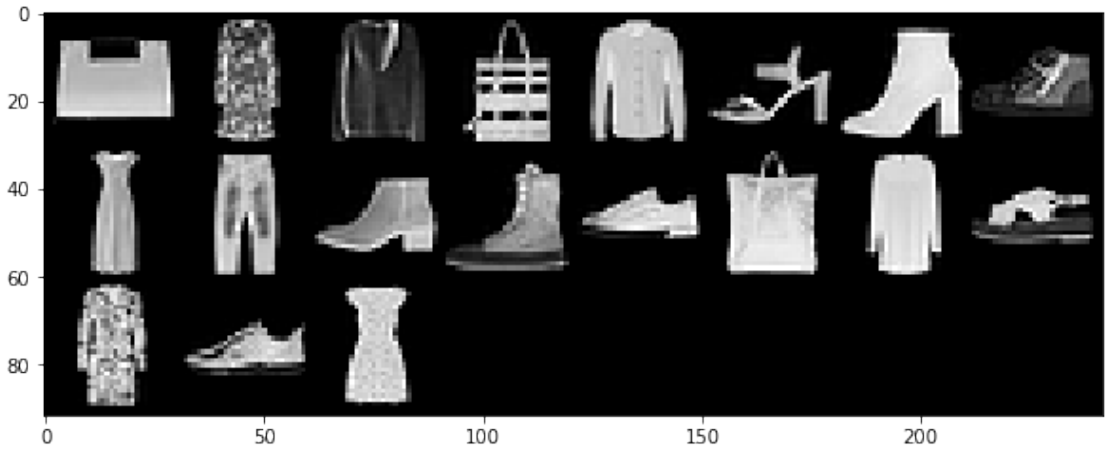




```
Epoch 21
-------------------------------
loss: 0.022299  [    0/60000]
loss: 0.017752  [ 2000/60000]
loss: 0.020800  [ 4000/60000]
loss: 0.023463  [ 6000/60000]
loss: 0.020896  [ 8000/60000]
loss: 0.022061  [10000/60000]
loss: 0.019739  [12000/60000]
loss: 0.022727  [14000/60000]
loss: 0.019066  [16000/60000]
```

```
loss: 0.021918  [18000/60000]
loss: 0.020450  [20000/60000]
loss: 0.022592  [22000/60000]
loss: 0.022077  [24000/60000]
loss: 0.021726  [26000/60000]
loss: 0.021007  [28000/60000]
loss: 0.018010  [30000/60000]
loss: 0.024713  [32000/60000]
loss: 0.022340  [34000/60000]
loss: 0.024855  [36000/60000]
loss: 0.021635  [38000/60000]
loss: 0.023974  [40000/60000]
loss: 0.018750  [42000/60000]
loss: 0.023781  [44000/60000]
loss: 0.022866  [46000/60000]
loss: 0.022676  [48000/60000]
loss: 0.024581  [50000/60000]
loss: 0.017801  [52000/60000]
loss: 0.026907  [54000/60000]
loss: 0.020367  [56000/60000]
loss: 0.017473  [58000/60000]
```

```
Clipping input data to the valid range for imshow with RGB data ([0..1] for
floats or [0..255] for integers).
```

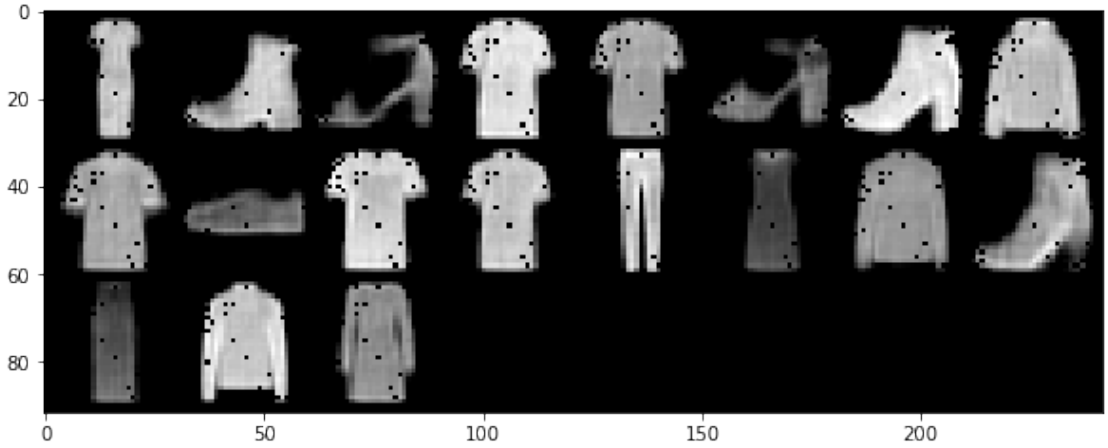


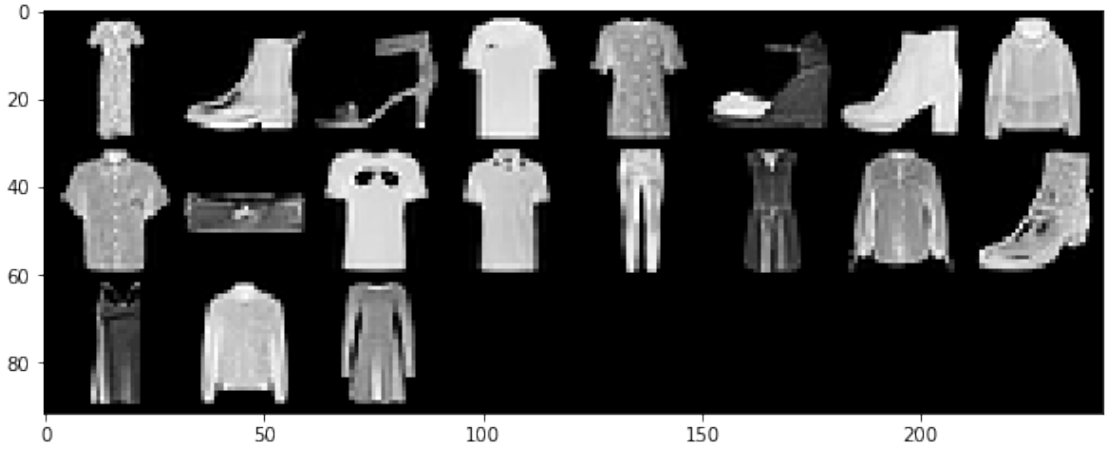



```
Epoch 22
-------------------------------
loss: 0.020257  [    0/60000]
loss: 0.023136  [ 2000/60000]
loss: 0.020322  [ 4000/60000]
loss: 0.017327  [ 6000/60000]
loss: 0.027206  [ 8000/60000]
loss: 0.020209  [10000/60000]
loss: 0.021790  [12000/60000]
loss: 0.022951  [14000/60000]
loss: 0.024600  [16000/60000]
loss: 0.023938  [18000/60000]
loss: 0.023642  [20000/60000]
loss: 0.022046  [22000/60000]
loss: 0.022831  [24000/60000]
loss: 0.018650  [26000/60000]
loss: 0.023152  [28000/60000]
loss: 0.020257  [30000/60000]
loss: 0.018411  [32000/60000]
loss: 0.017564  [34000/60000]
loss: 0.023547  [36000/60000]
loss: 0.020564  [38000/60000]
loss: 0.021148  [40000/60000]
loss: 0.019571  [42000/60000]
loss: 0.018949  [44000/60000]
loss: 0.017199  [46000/60000]
loss: 0.020427  [48000/60000]
loss: 0.019722  [50000/60000]
loss: 0.020441  [52000/60000]
loss: 0.029056  [54000/60000]
loss: 0.021677  [56000/60000]
```

```
loss: 0.022960  [58000/60000]
```

```
Clipping input data to the valid range for imshow with RGB data ([0..1] for
floats or [0..255] for integers).
```

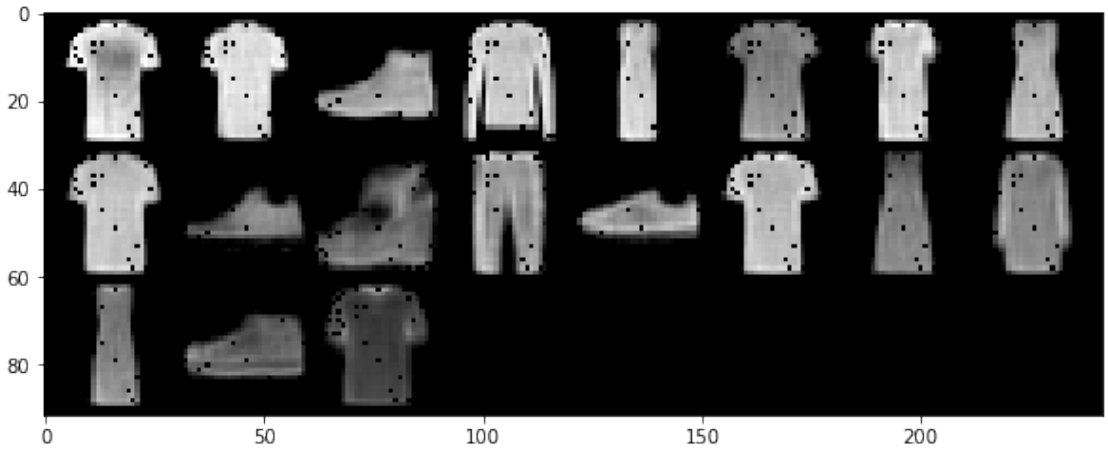


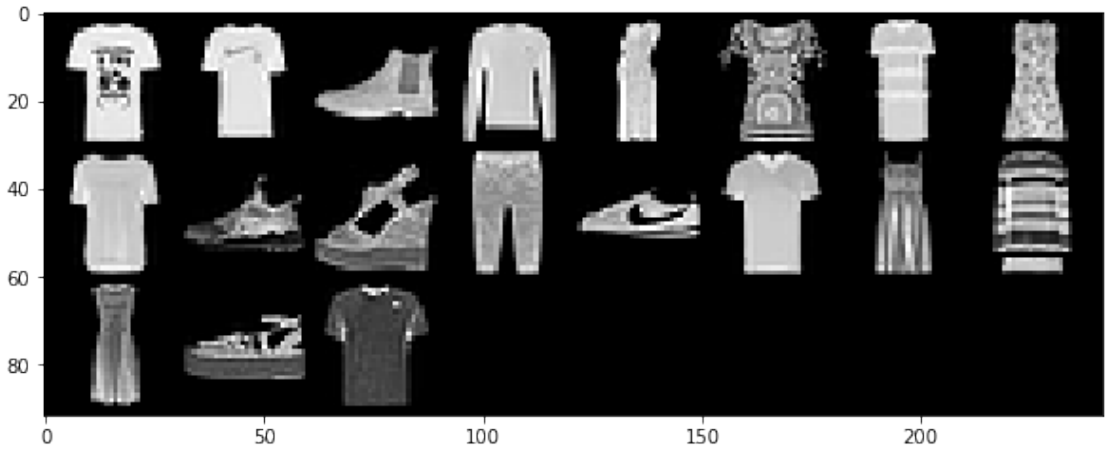




```
Epoch 23
-------------------------------
loss: 0.017747  [    0/60000]
loss: 0.019029  [ 2000/60000]
loss: 0.028038  [ 4000/60000]
loss: 0.018618  [ 6000/60000]
loss: 0.021730  [ 8000/60000]
loss: 0.022938  [10000/60000]
loss: 0.021655  [12000/60000]
loss: 0.027780  [14000/60000]
loss: 0.024875  [16000/60000]
```

```
loss: 0.018542  [18000/60000]
loss: 0.018901  [20000/60000]
loss: 0.020263  [22000/60000]
loss: 0.017400  [24000/60000]
loss: 0.028894  [26000/60000]
loss: 0.018588  [28000/60000]
loss: 0.018474  [30000/60000]
loss: 0.019276  [32000/60000]
loss: 0.020650  [34000/60000]
loss: 0.020782  [36000/60000]
loss: 0.019038  [38000/60000]
loss: 0.022182  [40000/60000]
loss: 0.017848  [42000/60000]
loss: 0.022550  [44000/60000]
loss: 0.017781  [46000/60000]
loss: 0.025300  [48000/60000]
loss: 0.017864  [50000/60000]
loss: 0.018689  [52000/60000]
loss: 0.015455  [54000/60000]
loss: 0.020335  [56000/60000]
loss: 0.018407  [58000/60000]
```

```
Clipping input data to the valid range for imshow with RGB data ([0..1] for
floats or [0..255] for integers).
```

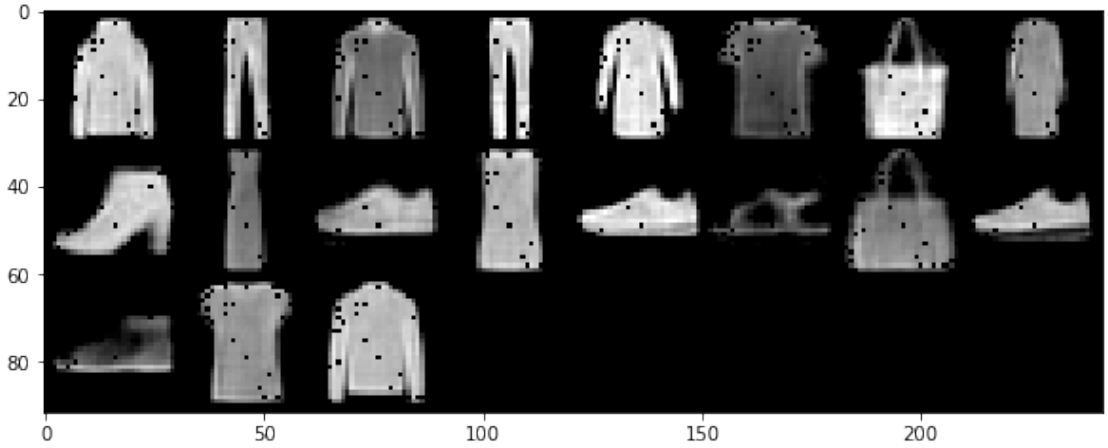


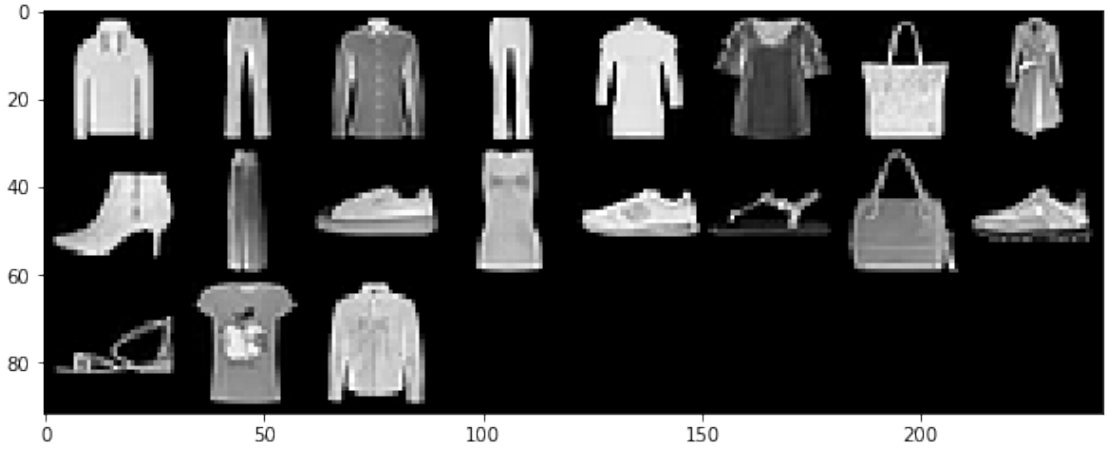



```
Epoch 24
-------------------------------
loss: 0.024060  [    0/60000]
loss: 0.023420  [ 2000/60000]
loss: 0.024757  [ 4000/60000]
loss: 0.019553  [ 6000/60000]
loss: 0.019786  [ 8000/60000]
loss: 0.023212  [10000/60000]
loss: 0.020194  [12000/60000]
loss: 0.020397  [14000/60000]
loss: 0.025528  [16000/60000]
loss: 0.022839  [18000/60000]
loss: 0.018276  [20000/60000]
loss: 0.019497  [22000/60000]
loss: 0.025640  [24000/60000]
loss: 0.018714  [26000/60000]
loss: 0.019638  [28000/60000]
loss: 0.021050  [30000/60000]
loss: 0.022744  [32000/60000]
loss: 0.025154  [34000/60000]
loss: 0.022396  [36000/60000]
loss: 0.021865  [38000/60000]
loss: 0.019257  [40000/60000]
loss: 0.021982  [42000/60000]
loss: 0.018208  [44000/60000]
loss: 0.020531  [46000/60000]
loss: 0.019389  [48000/60000]
loss: 0.019049  [50000/60000]
loss: 0.020885  [52000/60000]
loss: 0.020687  [54000/60000]
loss: 0.017557  [56000/60000]
```

```
loss: 0.017784  [58000/60000]
```

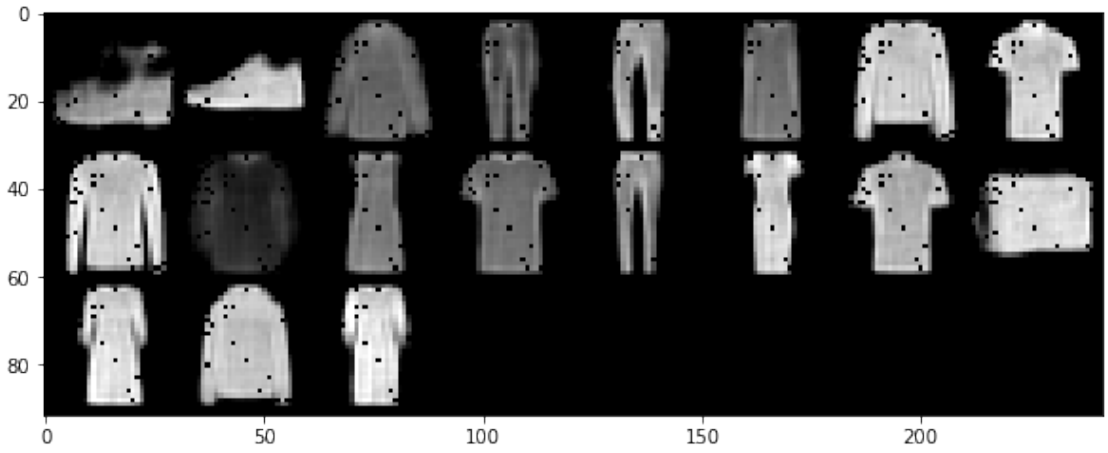
```
Clipping input data to the valid range for imshow with RGB data ([0..1] for
floats or [0..255] for integers).
```

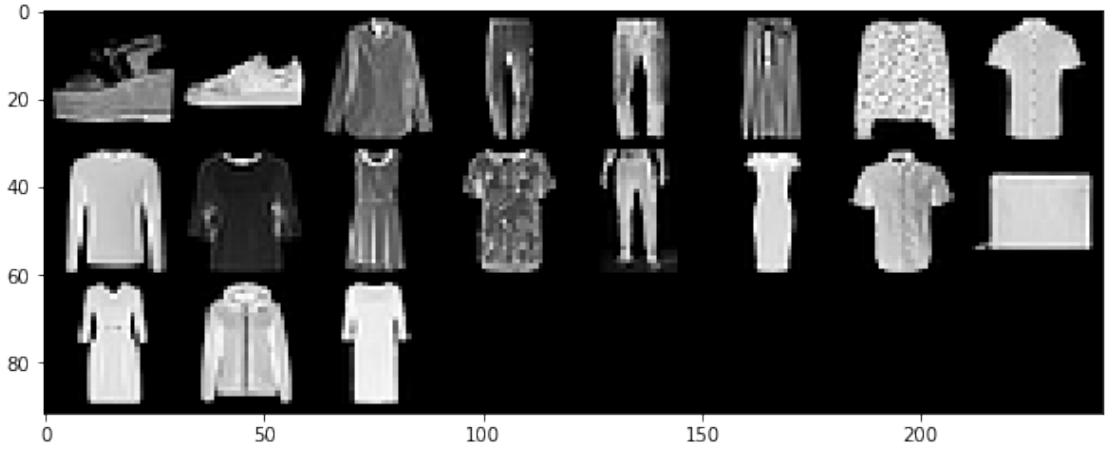



```
Epoch 25
------------------------------
loss: 0.021917  [    0/60000]
loss: 0.020263  [ 2000/60000]
loss: 0.024755  [ 4000/60000]
loss: 0.022547  [ 6000/60000]
loss: 0.018531  [ 8000/60000]
loss: 0.022593  [10000/60000]
loss: 0.023106  [12000/60000]
loss: 0.023882  [14000/60000]
loss: 0.015740  [16000/60000]
```

```
loss: 0.018511  [18000/60000]
loss: 0.020694  [20000/60000]
loss: 0.020990  [22000/60000]
loss: 0.023040  [24000/60000]
loss: 0.016919  [26000/60000]
loss: 0.022353  [28000/60000]
loss: 0.018557  [30000/60000]
loss: 0.019328  [32000/60000]
loss: 0.018806  [34000/60000]
loss: 0.023588  [36000/60000]
loss: 0.018188  [38000/60000]
loss: 0.021520  [40000/60000]
loss: 0.021411  [42000/60000]
loss: 0.022975  [44000/60000]
loss: 0.022445  [46000/60000]
loss: 0.021001  [48000/60000]
loss: 0.020139  [50000/60000]
loss: 0.019220  [52000/60000]
loss: 0.022745  [54000/60000]
loss: 0.020113  [56000/60000]
loss: 0.021176  [58000/60000]
Clipping input data to the valid range for imshow with RGB data ([0..1] for
floats or [0..255] for integers).
```

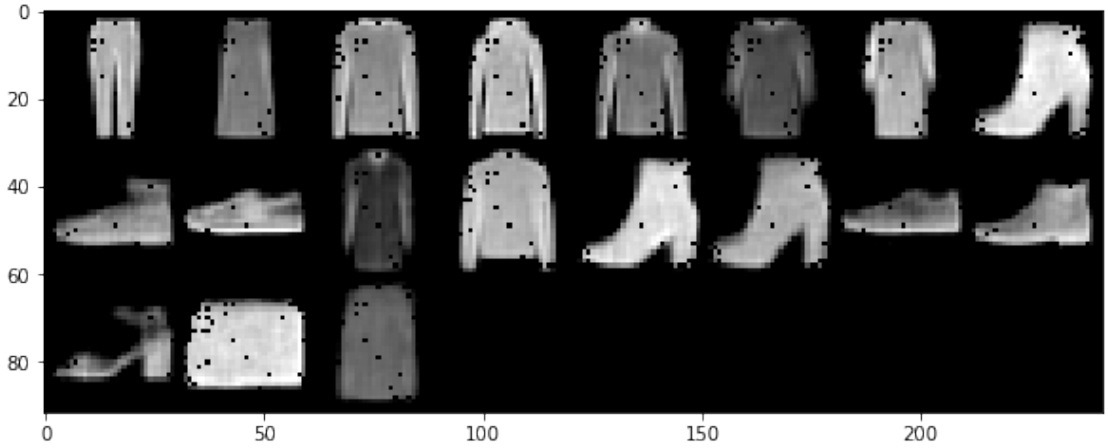


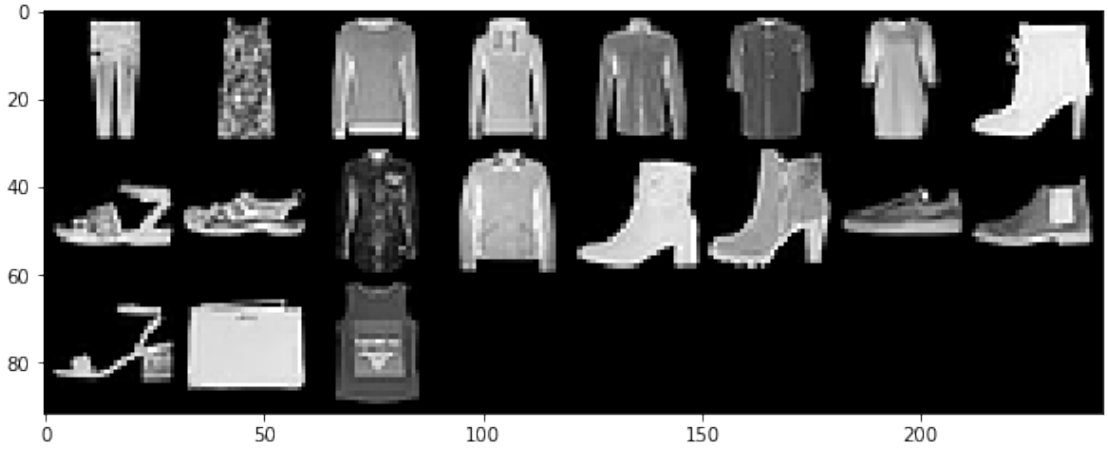



```
Epoch 26
-------------------------------
loss: 0.023705  [    0/60000]
loss: 0.021389  [ 2000/60000]
loss: 0.025313  [ 4000/60000]
loss: 0.024878  [ 6000/60000]
loss: 0.021656  [ 8000/60000]
loss: 0.020917  [10000/60000]
loss: 0.019293  [12000/60000]
loss: 0.022937  [14000/60000]
loss: 0.019205  [16000/60000]
loss: 0.024545  [18000/60000]
loss: 0.018798  [20000/60000]
loss: 0.022756  [22000/60000]
loss: 0.016691  [24000/60000]
loss: 0.023425  [26000/60000]
loss: 0.020392  [28000/60000]
loss: 0.018741  [30000/60000]
loss: 0.021957  [32000/60000]
loss: 0.022431  [34000/60000]
loss: 0.020052  [36000/60000]
loss: 0.027324  [38000/60000]
loss: 0.023297  [40000/60000]
loss: 0.027831  [42000/60000]
loss: 0.022375  [44000/60000]
loss: 0.020746  [46000/60000]
loss: 0.018230  [48000/60000]
loss: 0.022312  [50000/60000]
loss: 0.021101  [52000/60000]
loss: 0.022164  [54000/60000]
loss: 0.023092  [56000/60000]
```

```
loss: 0.022069  [58000/60000]
```

```
Clipping input data to the valid range for imshow with RGB data ([0..1] for
floats or [0..255] for integers).
```

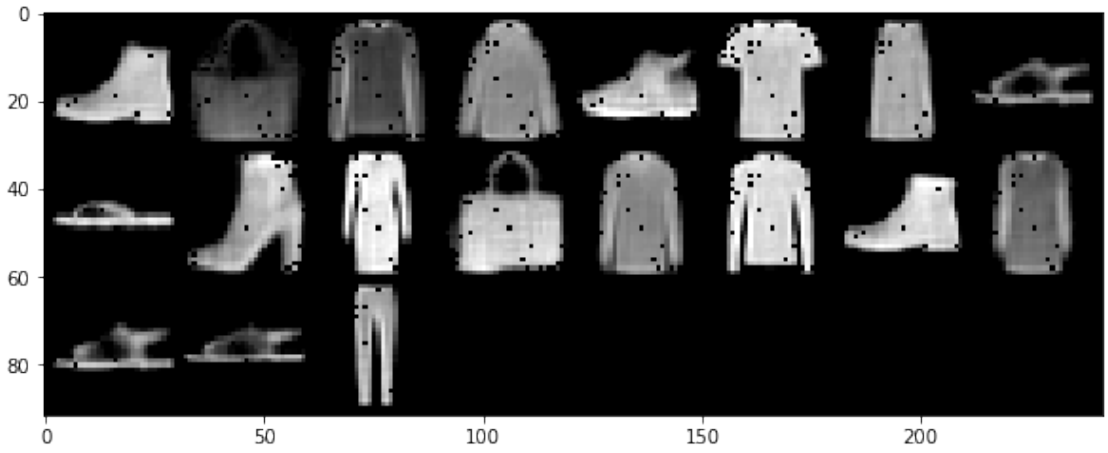


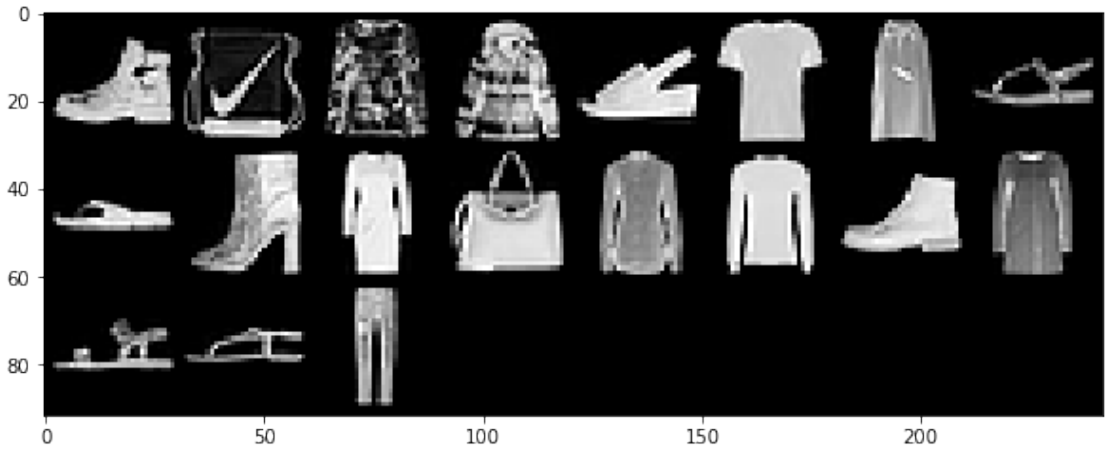




```
Epoch 27
-------------------------------
loss: 0.018842  [    0/60000]
loss: 0.017427  [ 2000/60000]
loss: 0.018310  [ 4000/60000]
loss: 0.019344  [ 6000/60000]
loss: 0.023484  [ 8000/60000]
loss: 0.020453  [10000/60000]
loss: 0.018942  [12000/60000]
loss: 0.019159  [14000/60000]
loss: 0.023246  [16000/60000]
```

```
loss: 0.019237  [18000/60000]
loss: 0.023108  [20000/60000]
loss: 0.021501  [22000/60000]
loss: 0.022873  [24000/60000]
loss: 0.021025  [26000/60000]
loss: 0.021732  [28000/60000]
loss: 0.016686  [30000/60000]
loss: 0.020397  [32000/60000]
loss: 0.020218  [34000/60000]
loss: 0.021383  [36000/60000]
loss: 0.023595  [38000/60000]
loss: 0.020614  [40000/60000]
loss: 0.021944  [42000/60000]
loss: 0.022082  [44000/60000]
loss: 0.021277  [46000/60000]
loss: 0.022535  [48000/60000]
loss: 0.020621  [50000/60000]
loss: 0.022609  [52000/60000]
loss: 0.018962  [54000/60000]
loss: 0.022153  [56000/60000]
loss: 0.021334  [58000/60000]
```

```
Clipping input data to the valid range for imshow with RGB data ([0..1] for
floats or [0..255] for integers).
```

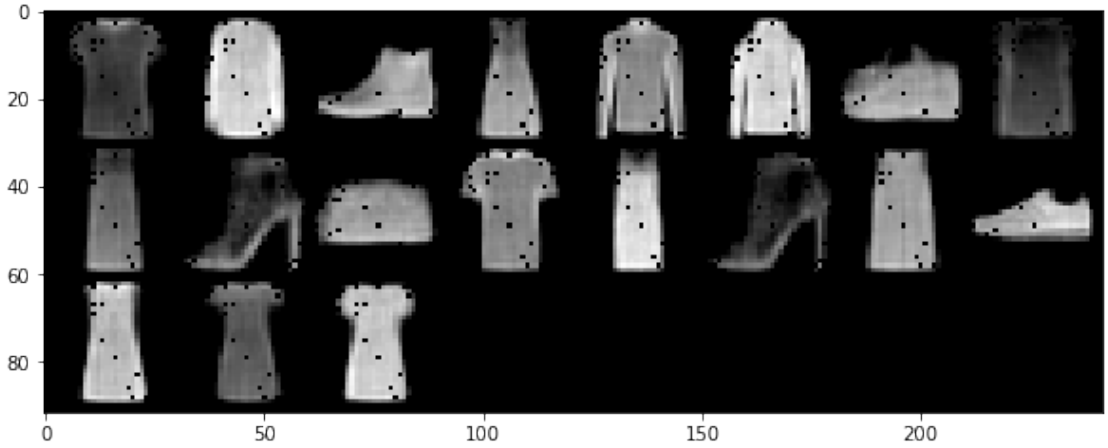


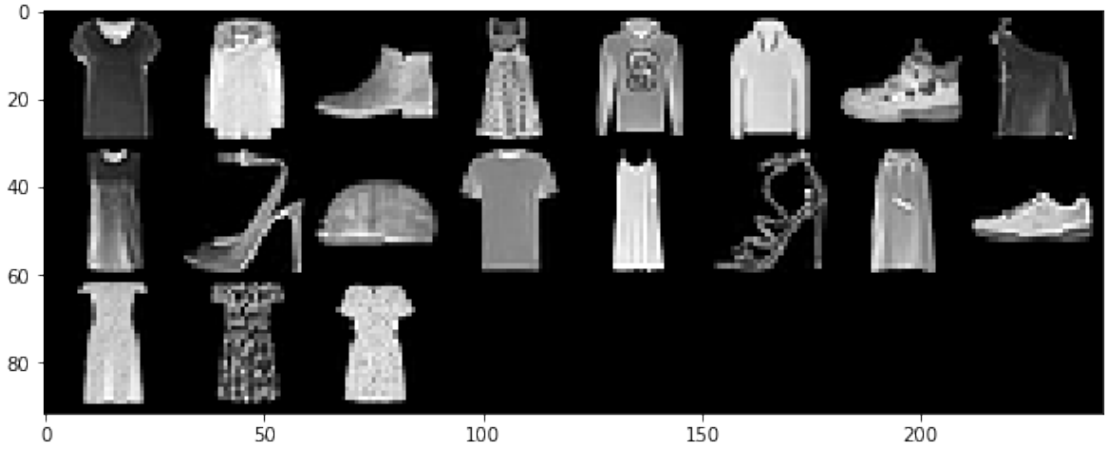



```
Epoch 28
-------------------------------
loss: 0.019269  [    0/60000]
loss: 0.017227  [ 2000/60000]
loss: 0.022898  [ 4000/60000]
loss: 0.022468  [ 6000/60000]
loss: 0.024100  [ 8000/60000]
loss: 0.017827  [10000/60000]
loss: 0.020717  [12000/60000]
loss: 0.020021  [14000/60000]
loss: 0.017899  [16000/60000]
loss: 0.018965  [18000/60000]
loss: 0.017631  [20000/60000]
loss: 0.019743  [22000/60000]
loss: 0.017133  [24000/60000]
loss: 0.021560  [26000/60000]
loss: 0.019704  [28000/60000]
loss: 0.023693  [30000/60000]
loss: 0.021161  [32000/60000]
loss: 0.019415  [34000/60000]
loss: 0.020706  [36000/60000]
loss: 0.018045  [38000/60000]
loss: 0.018719  [40000/60000]
loss: 0.020371  [42000/60000]
loss: 0.022478  [44000/60000]
loss: 0.023063  [46000/60000]
loss: 0.020904  [48000/60000]
loss: 0.020953  [50000/60000]
loss: 0.021850  [52000/60000]
loss: 0.023385  [54000/60000]
loss: 0.019934  [56000/60000]
```

```
loss: 0.025322  [58000/60000]
```

```
Clipping input data to the valid range for imshow with RGB data ([0..1] for
floats or [0..255] for integers).
```

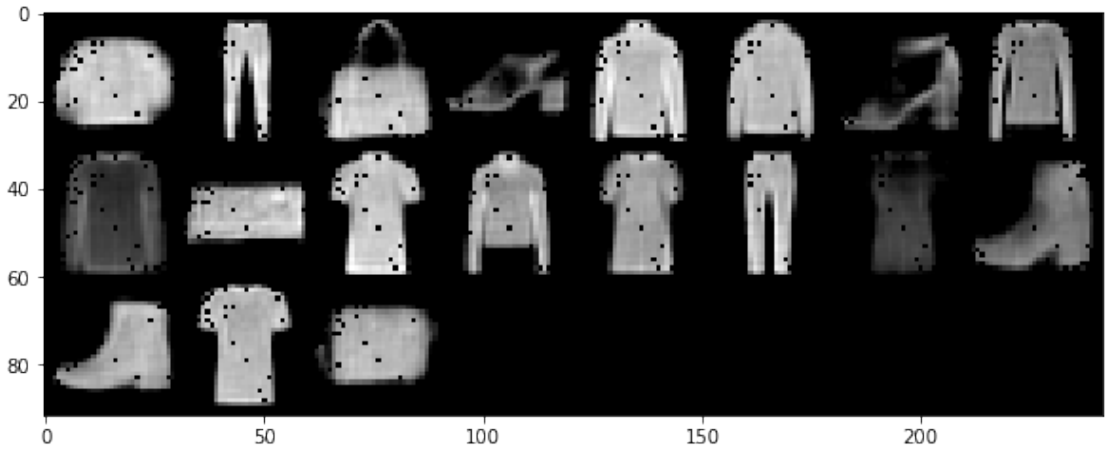


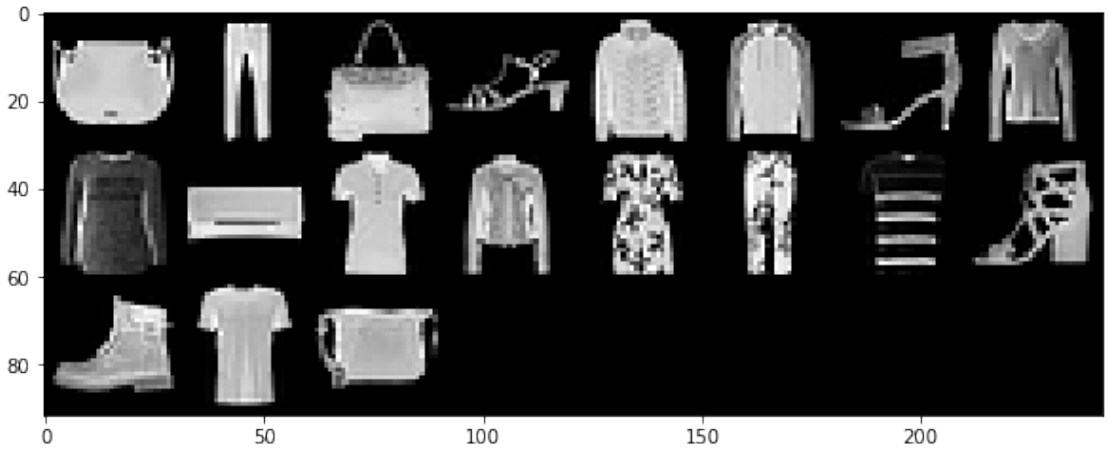




```
Epoch 29
------------------------------
loss: 0.020544  [    0/60000]
loss: 0.023939  [ 2000/60000]
loss: 0.016221  [ 4000/60000]
loss: 0.022914  [ 6000/60000]
loss: 0.022517  [ 8000/60000]
loss: 0.021368  [10000/60000]
loss: 0.020082  [12000/60000]
loss: 0.020604  [14000/60000]
loss: 0.025604  [16000/60000]
```

```
loss: 0.020283  [18000/60000]
loss: 0.017833  [20000/60000]
loss: 0.024912  [22000/60000]
loss: 0.019371  [24000/60000]
loss: 0.022541  [26000/60000]
loss: 0.018577  [28000/60000]
loss: 0.020352  [30000/60000]
loss: 0.021004  [32000/60000]
loss: 0.025274  [34000/60000]
loss: 0.021456  [36000/60000]
loss: 0.021319  [38000/60000]
loss: 0.020424  [40000/60000]
loss: 0.024907  [42000/60000]
loss: 0.020094  [44000/60000]
loss: 0.016684  [46000/60000]
loss: 0.022386  [48000/60000]
loss: 0.019849  [50000/60000]
loss: 0.019935  [52000/60000]
loss: 0.022122  [54000/60000]
loss: 0.019469  [56000/60000]
loss: 0.026282  [58000/60000]
```

```
Clipping input data to the valid range for imshow with RGB data ([0..1] for
floats or [0..255] for integers).
```

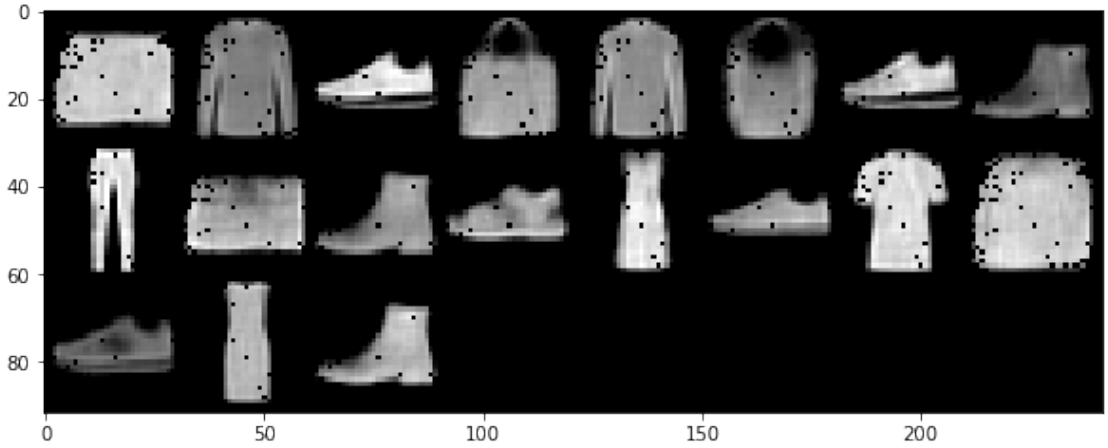


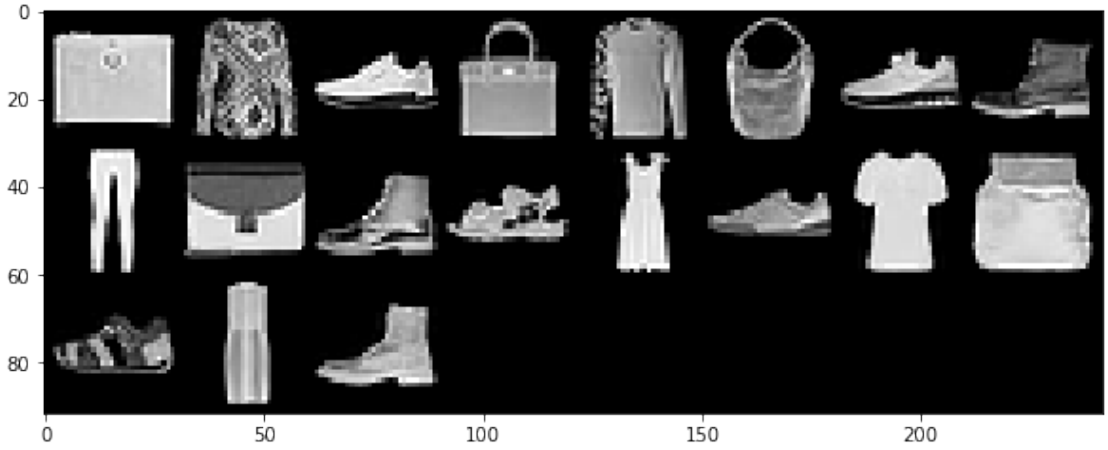



```
Epoch 30
-------------------------------
loss: 0.021801  [    0/60000]
loss: 0.020059  [ 2000/60000]
loss: 0.016526  [ 4000/60000]
loss: 0.023063  [ 6000/60000]
loss: 0.024389  [ 8000/60000]
loss: 0.017959  [10000/60000]
loss: 0.019572  [12000/60000]
loss: 0.013984  [14000/60000]
loss: 0.023156  [16000/60000]
loss: 0.021960  [18000/60000]
loss: 0.018441  [20000/60000]
loss: 0.020258  [22000/60000]
loss: 0.017454  [24000/60000]
loss: 0.018707  [26000/60000]
loss: 0.017811  [28000/60000]
loss: 0.022712  [30000/60000]
loss: 0.022148  [32000/60000]
loss: 0.021792  [34000/60000]
loss: 0.019071  [36000/60000]
loss: 0.021924  [38000/60000]
loss: 0.018382  [40000/60000]
loss: 0.026592  [42000/60000]
loss: 0.020494  [44000/60000]
loss: 0.022102  [46000/60000]
loss: 0.022176  [48000/60000]
loss: 0.019583  [50000/60000]
loss: 0.022791  [52000/60000]
loss: 0.017540  [54000/60000]
loss: 0.020371  [56000/60000]
```

```
loss: 0.017451  [58000/60000]
```

```
Clipping input data to the valid range for imshow with RGB data ([0..1] for
floats or [0..255] for integers).
```

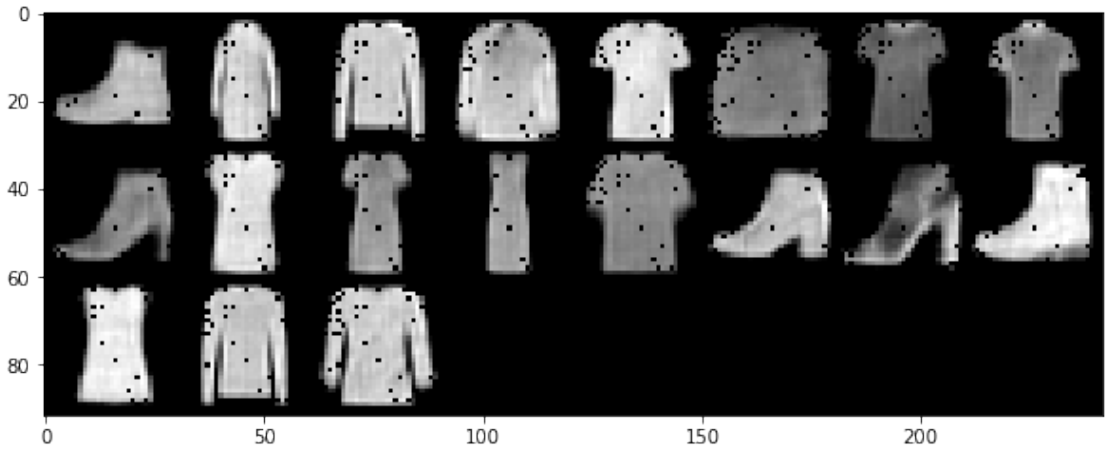


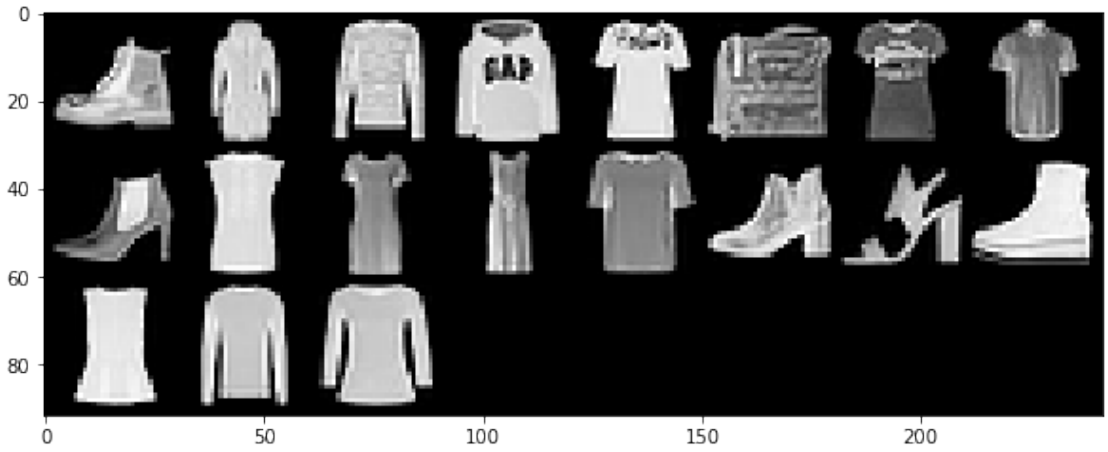




```
Epoch 31
-------------------------------
loss: 0.018855  [    0/60000]
loss: 0.022475  [ 2000/60000]
loss: 0.019305  [ 4000/60000]
loss: 0.016545  [ 6000/60000]
loss: 0.021712  [ 8000/60000]
loss: 0.020113  [10000/60000]
loss: 0.020711  [12000/60000]
loss: 0.016972  [14000/60000]
loss: 0.019864  [16000/60000]
```

```
loss: 0.021393  [18000/60000]
loss: 0.016756  [20000/60000]
loss: 0.021035  [22000/60000]
loss: 0.018081  [24000/60000]
loss: 0.020015  [26000/60000]
loss: 0.017792  [28000/60000]
loss: 0.023753  [30000/60000]
loss: 0.018171  [32000/60000]
loss: 0.016624  [34000/60000]
loss: 0.020484  [36000/60000]
loss: 0.019466  [38000/60000]
loss: 0.022712  [40000/60000]
loss: 0.020414  [42000/60000]
loss: 0.023010  [44000/60000]
loss: 0.017713  [46000/60000]
loss: 0.019050  [48000/60000]
loss: 0.021550  [50000/60000]
loss: 0.021470  [52000/60000]
loss: 0.019555  [54000/60000]
loss: 0.022807  [56000/60000]
loss: 0.024282  [58000/60000]
```

```
Clipping input data to the valid range for imshow with RGB data ([0..1] for
floats or [0..255] for integers).
```

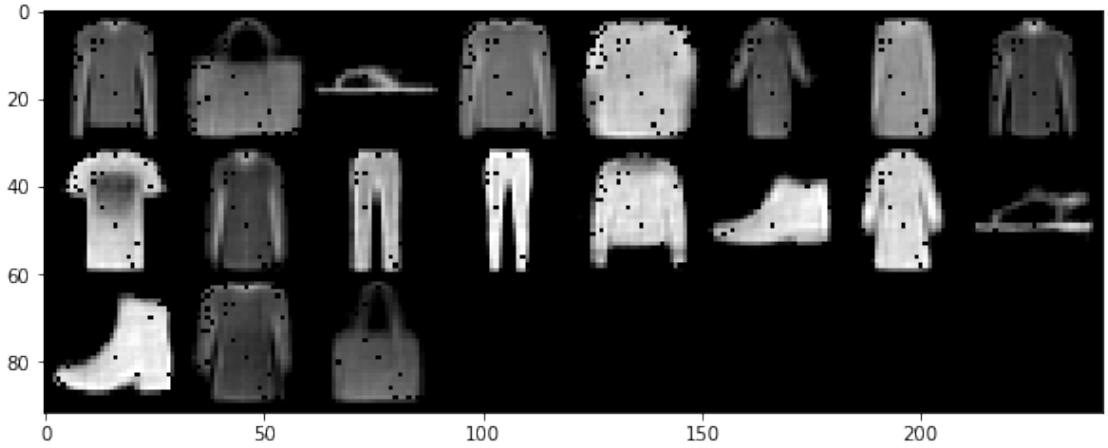


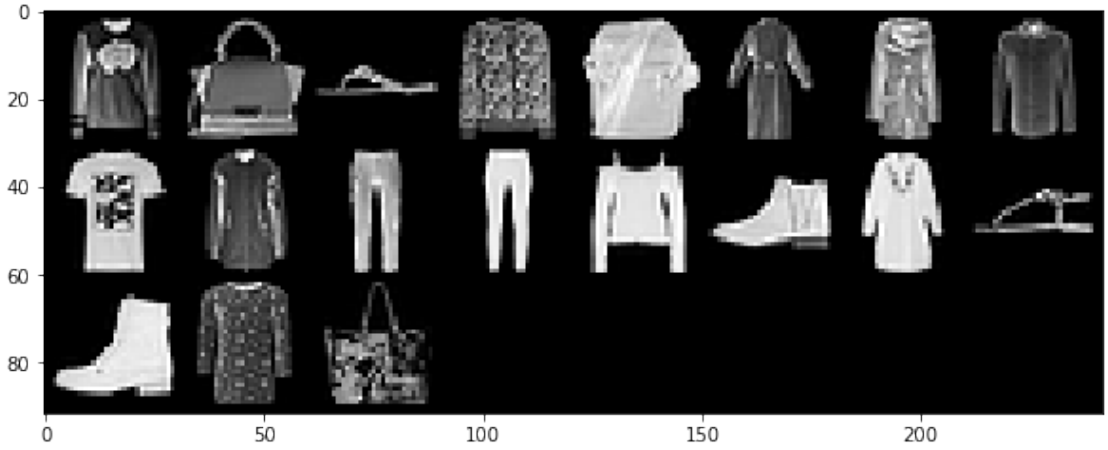



```
Epoch 32
-------------------------------
loss: 0.018675  [    0/60000]
loss: 0.023696  [ 2000/60000]
loss: 0.022529  [ 4000/60000]
loss: 0.018647  [ 6000/60000]
loss: 0.017409  [ 8000/60000]
loss: 0.021592  [10000/60000]
loss: 0.020956  [12000/60000]
loss: 0.020675  [14000/60000]
loss: 0.020307  [16000/60000]
loss: 0.018763  [18000/60000]
loss: 0.018096  [20000/60000]
loss: 0.021311  [22000/60000]
loss: 0.022746  [24000/60000]
loss: 0.019683  [26000/60000]
loss: 0.019287  [28000/60000]
loss: 0.019753  [30000/60000]
loss: 0.022590  [32000/60000]
loss: 0.020964  [34000/60000]
loss: 0.018508  [36000/60000]
loss: 0.021186  [38000/60000]
loss: 0.019899  [40000/60000]
loss: 0.016175  [42000/60000]
loss: 0.024695  [44000/60000]
loss: 0.021037  [46000/60000]
loss: 0.022291  [48000/60000]
loss: 0.017703  [50000/60000]
loss: 0.021202  [52000/60000]
loss: 0.024477  [54000/60000]
loss: 0.017704  [56000/60000]
```

```
loss: 0.018905  [58000/60000]
```

```
Clipping input data to the valid range for imshow with RGB data ([0..1] for
floats or [0..255] for integers).
```

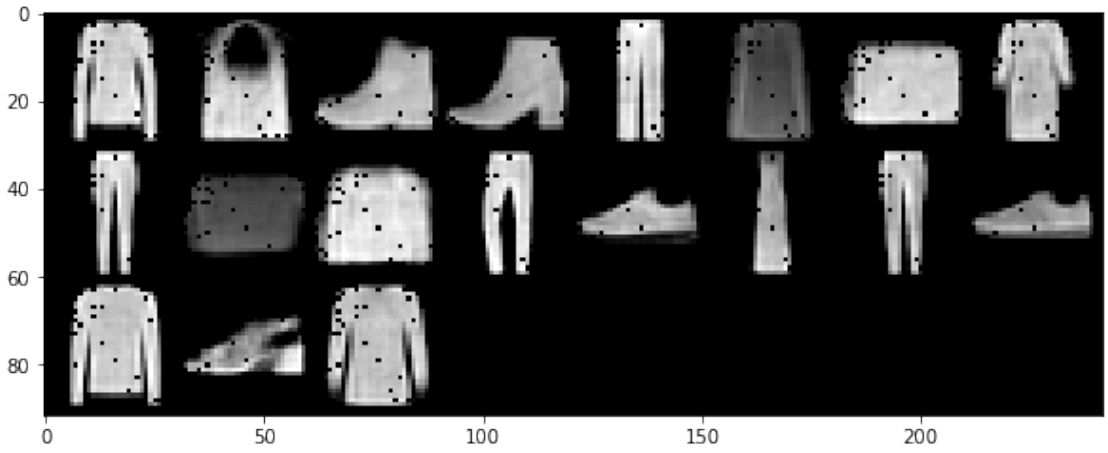


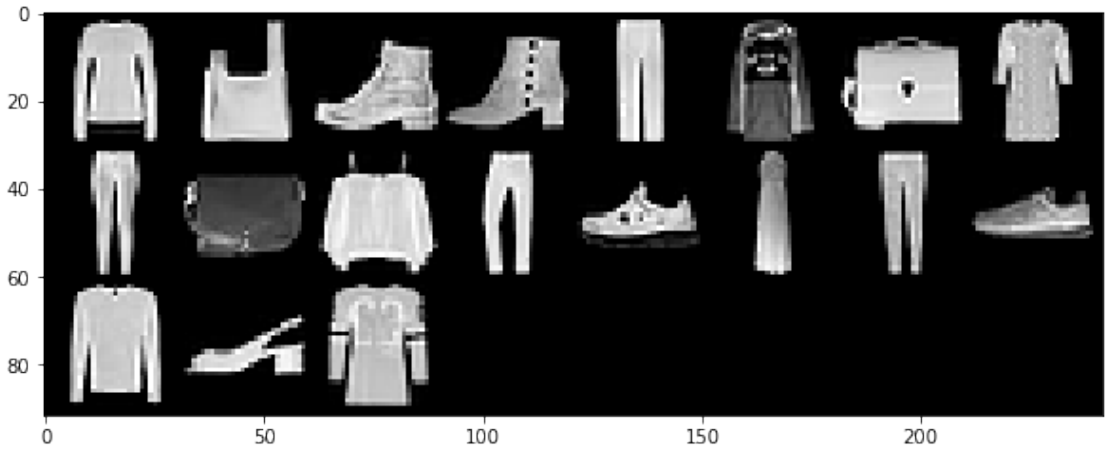




```
Epoch 33
-------------------------------
loss: 0.020905  [    0/60000]
loss: 0.020290  [ 2000/60000]
loss: 0.022538  [ 4000/60000]
loss: 0.021705  [ 6000/60000]
loss: 0.023228  [ 8000/60000]
loss: 0.015865  [10000/60000]
loss: 0.027602  [12000/60000]
loss: 0.020880  [14000/60000]
loss: 0.020082  [16000/60000]
```

```
loss: 0.023972  [18000/60000]
loss: 0.020381  [20000/60000]
loss: 0.018778  [22000/60000]
loss: 0.021601  [24000/60000]
loss: 0.019104  [26000/60000]
loss: 0.021478  [28000/60000]
loss: 0.020523  [30000/60000]
loss: 0.019963  [32000/60000]
loss: 0.020592  [34000/60000]
loss: 0.021212  [36000/60000]
loss: 0.017867  [38000/60000]
loss: 0.022426  [40000/60000]
loss: 0.020375  [42000/60000]
loss: 0.017016  [44000/60000]
loss: 0.016937  [46000/60000]
loss: 0.024881  [48000/60000]
loss: 0.024601  [50000/60000]
loss: 0.019427  [52000/60000]
loss: 0.020814  [54000/60000]
loss: 0.027241  [56000/60000]
loss: 0.021794  [58000/60000]
```

```
Clipping input data to the valid range for imshow with RGB data ([0..1] for
floats or [0..255] for integers).
```

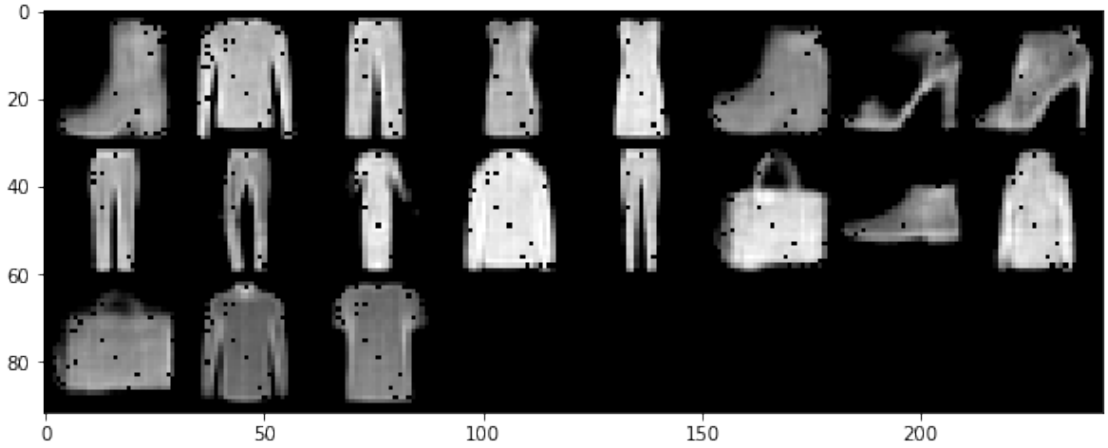


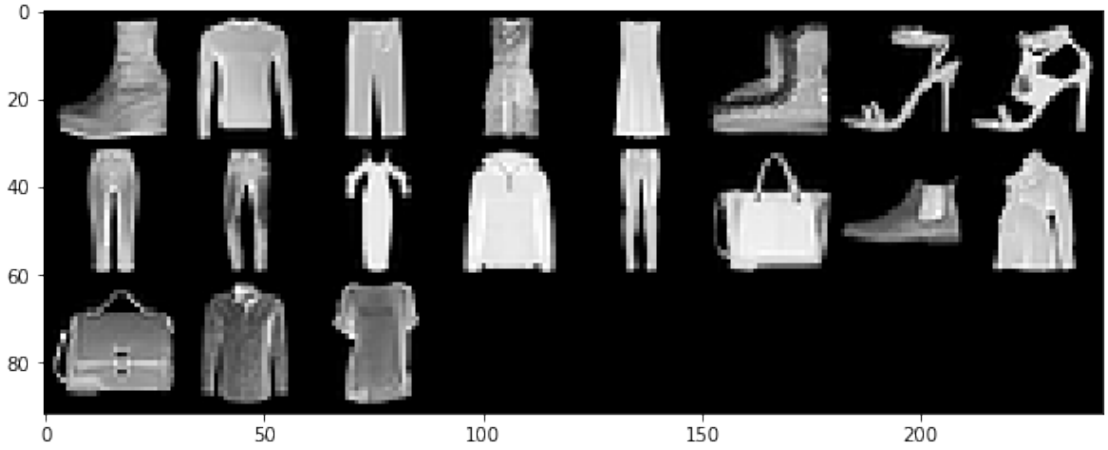



```
Epoch 34
-------------------------------
loss: 0.017030  [    0/60000]
loss: 0.022073  [ 2000/60000]
loss: 0.020186  [ 4000/60000]
loss: 0.020750  [ 6000/60000]
loss: 0.021708  [ 8000/60000]
loss: 0.017122  [10000/60000]
loss: 0.022898  [12000/60000]
loss: 0.019677  [14000/60000]
loss: 0.022013  [16000/60000]
loss: 0.015424  [18000/60000]
loss: 0.021292  [20000/60000]
loss: 0.023948  [22000/60000]
loss: 0.021393  [24000/60000]
loss: 0.018221  [26000/60000]
loss: 0.018375  [28000/60000]
loss: 0.021687  [30000/60000]
loss: 0.020502  [32000/60000]
loss: 0.025205  [34000/60000]
loss: 0.016993  [36000/60000]
loss: 0.022503  [38000/60000]
loss: 0.019727  [40000/60000]
loss: 0.017471  [42000/60000]
loss: 0.021432  [44000/60000]
loss: 0.018624  [46000/60000]
loss: 0.023352  [48000/60000]
loss: 0.019384  [50000/60000]
loss: 0.016160  [52000/60000]
loss: 0.022673  [54000/60000]
loss: 0.022241  [56000/60000]
```

```
loss: 0.020617  [58000/60000]
```

```
Clipping input data to the valid range for imshow with RGB data ([0..1] for
floats or [0..255] for integers).
```

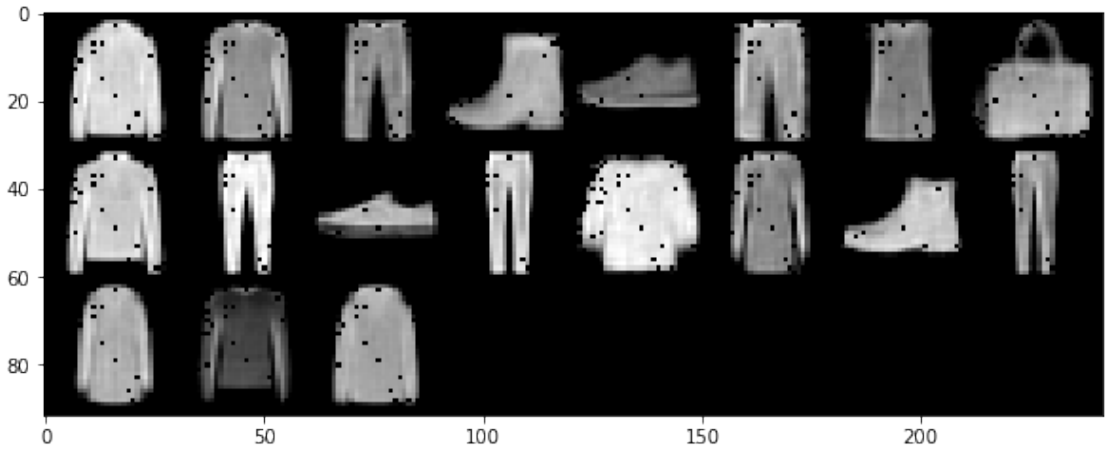


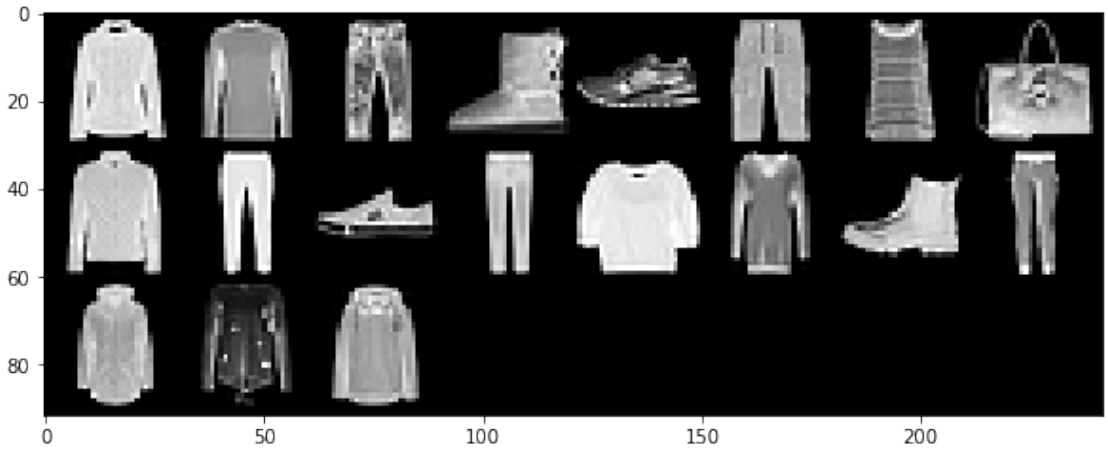




```
Epoch 35
-------------------------------
loss: 0.020993  [    0/60000]
loss: 0.020889  [ 2000/60000]
loss: 0.022460  [ 4000/60000]
loss: 0.022561  [ 6000/60000]
loss: 0.023659  [ 8000/60000]
loss: 0.020735  [10000/60000]
loss: 0.019977  [12000/60000]
loss: 0.021619  [14000/60000]
loss: 0.020136  [16000/60000]
```

```
loss: 0.021934  [18000/60000]
loss: 0.024822  [20000/60000]
loss: 0.021750  [22000/60000]
loss: 0.020178  [24000/60000]
loss: 0.018983  [26000/60000]
loss: 0.022783  [28000/60000]
loss: 0.021040  [30000/60000]
loss: 0.025416  [32000/60000]
loss: 0.020936  [34000/60000]
loss: 0.021624  [36000/60000]
loss: 0.020638  [38000/60000]
loss: 0.022186  [40000/60000]
loss: 0.021651  [42000/60000]
loss: 0.021987  [44000/60000]
loss: 0.015479  [46000/60000]
loss: 0.018531  [48000/60000]
loss: 0.022982  [50000/60000]
loss: 0.022364  [52000/60000]
loss: 0.019284  [54000/60000]
loss: 0.025861  [56000/60000]
loss: 0.022978  [58000/60000]
```

```
Clipping input data to the valid range for imshow with RGB data ([0..1] for
floats or [0..255] for integers).
```

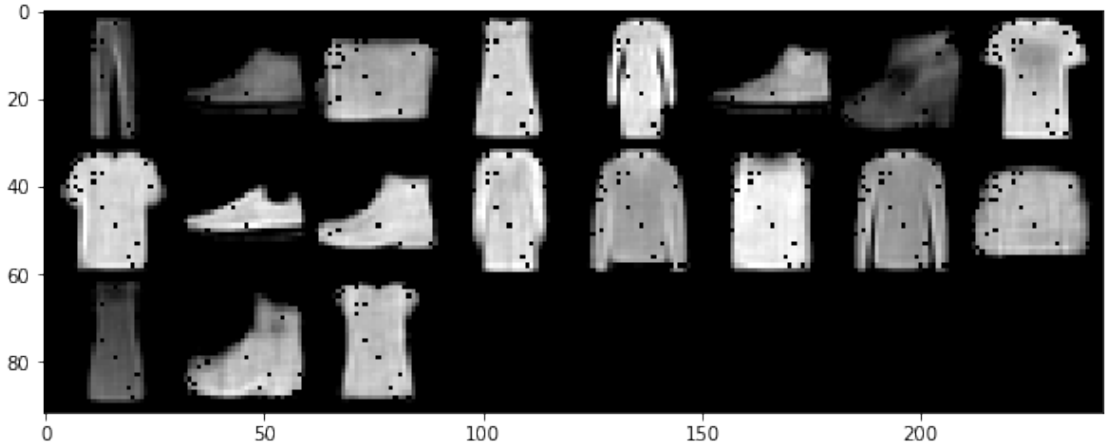


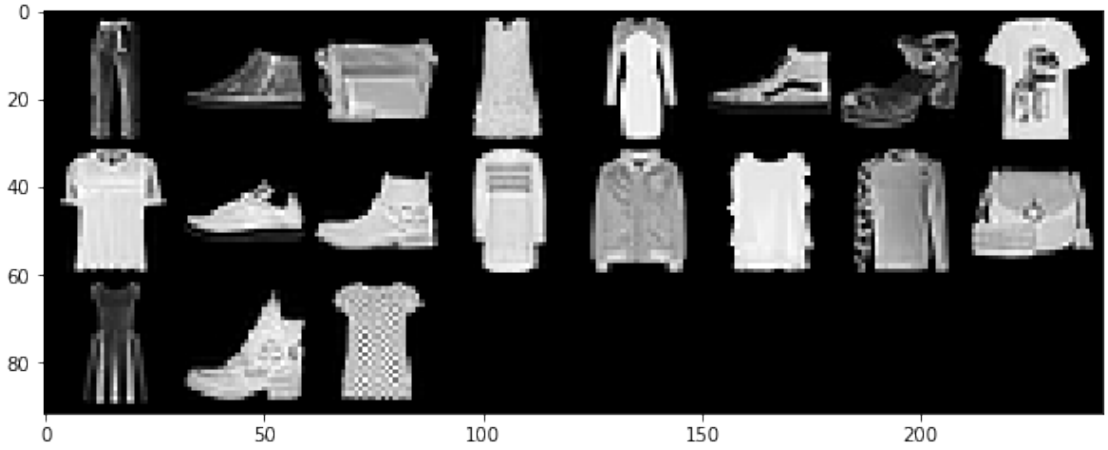



```
Epoch 36
-------------------------------
loss: 0.023326  [    0/60000]
loss: 0.020345  [ 2000/60000]
loss: 0.020342  [ 4000/60000]
loss: 0.021351  [ 6000/60000]
loss: 0.020761  [ 8000/60000]
loss: 0.020366  [10000/60000]
loss: 0.018747  [12000/60000]
loss: 0.018815  [14000/60000]
loss: 0.026383  [16000/60000]
loss: 0.018341  [18000/60000]
loss: 0.024944  [20000/60000]
loss: 0.019630  [22000/60000]
loss: 0.021416  [24000/60000]
loss: 0.023009  [26000/60000]
loss: 0.022984  [28000/60000]
loss: 0.022545  [30000/60000]
loss: 0.021506  [32000/60000]
loss: 0.017933  [34000/60000]
loss: 0.023747  [36000/60000]
loss: 0.022597  [38000/60000]
loss: 0.019002  [40000/60000]
loss: 0.017005  [42000/60000]
loss: 0.017695  [44000/60000]
loss: 0.017723  [46000/60000]
loss: 0.019665  [48000/60000]
loss: 0.018417  [50000/60000]
loss: 0.016982  [52000/60000]
loss: 0.018674  [54000/60000]
loss: 0.016742  [56000/60000]
```

```
loss: 0.022401  [58000/60000]
```

```
Clipping input data to the valid range for imshow with RGB data ([0..1] for
floats or [0..255] for integers).
```

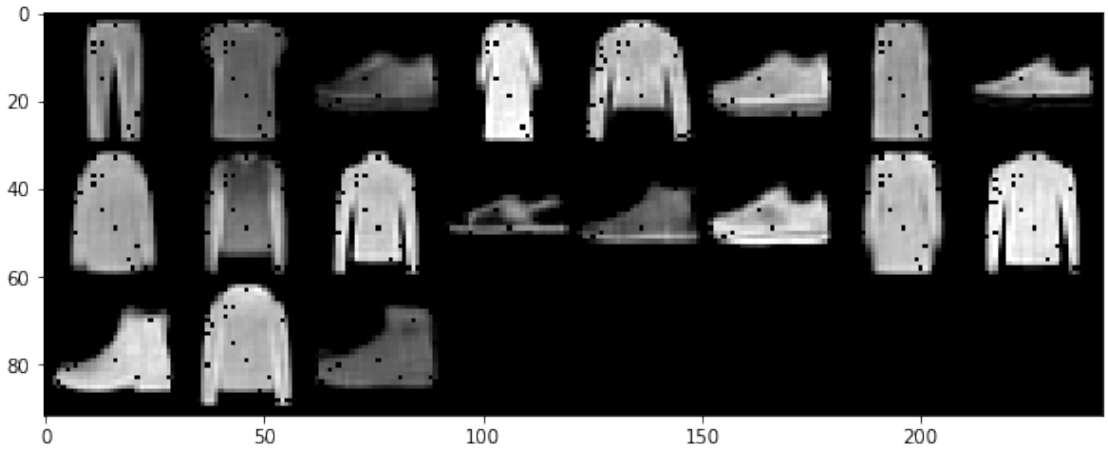


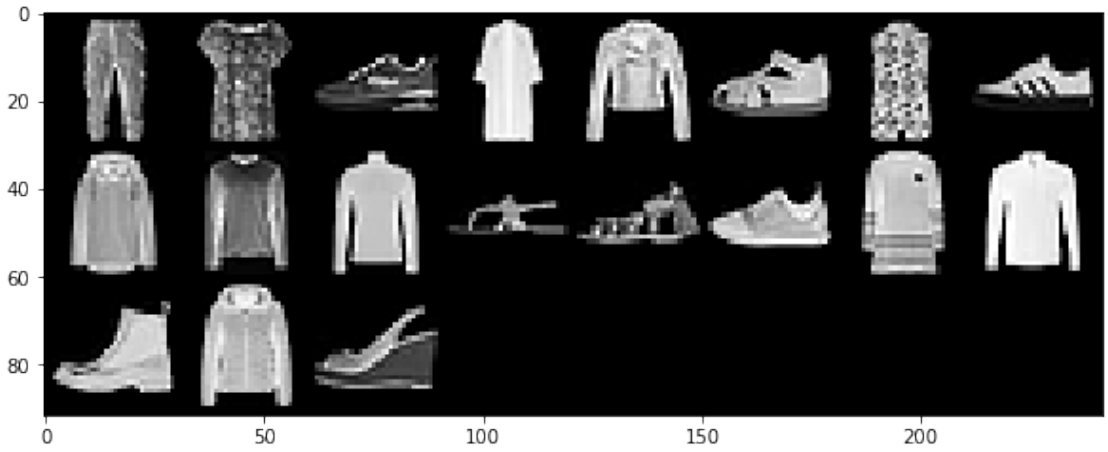




```
Epoch 37
-------------------------------
loss: 0.019904  [    0/60000]
loss: 0.018826  [ 2000/60000]
loss: 0.015590  [ 4000/60000]
loss: 0.021732  [ 6000/60000]
loss: 0.021877  [ 8000/60000]
loss: 0.021505  [10000/60000]
loss: 0.019464  [12000/60000]
loss: 0.018368  [14000/60000]
loss: 0.021374  [16000/60000]
```

```
loss: 0.015898  [18000/60000]
loss: 0.019797  [20000/60000]
loss: 0.021754  [22000/60000]
loss: 0.016995  [24000/60000]
loss: 0.016895  [26000/60000]
loss: 0.019254  [28000/60000]
loss: 0.018784  [30000/60000]
loss: 0.020964  [32000/60000]
loss: 0.016746  [34000/60000]
loss: 0.018840  [36000/60000]
loss: 0.019024  [38000/60000]
loss: 0.024364  [40000/60000]
loss: 0.021765  [42000/60000]
loss: 0.020725  [44000/60000]
loss: 0.020192  [46000/60000]
loss: 0.021934  [48000/60000]
loss: 0.020486  [50000/60000]
loss: 0.026202  [52000/60000]
loss: 0.022768  [54000/60000]
loss: 0.016115  [56000/60000]
loss: 0.016098  [58000/60000]
```

```
Clipping input data to the valid range for imshow with RGB data ([0..1] for
floats or [0..255] for integers).
```

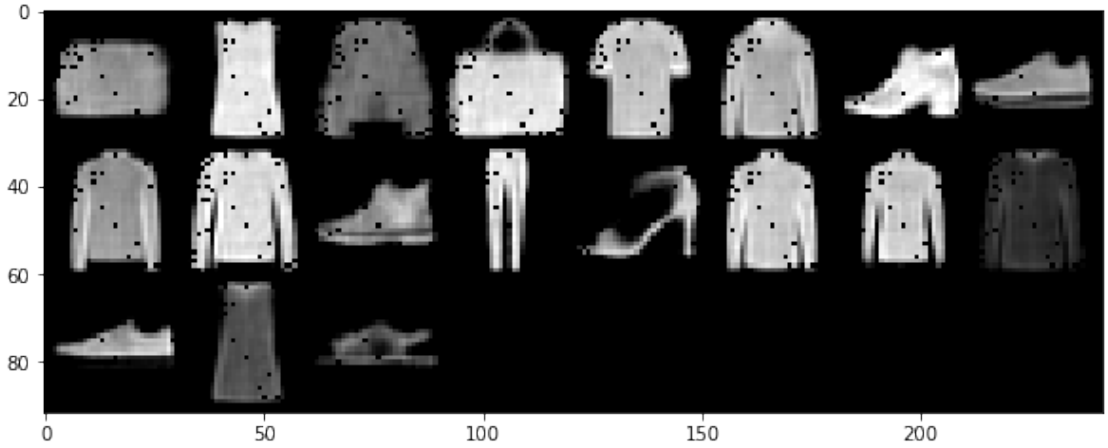


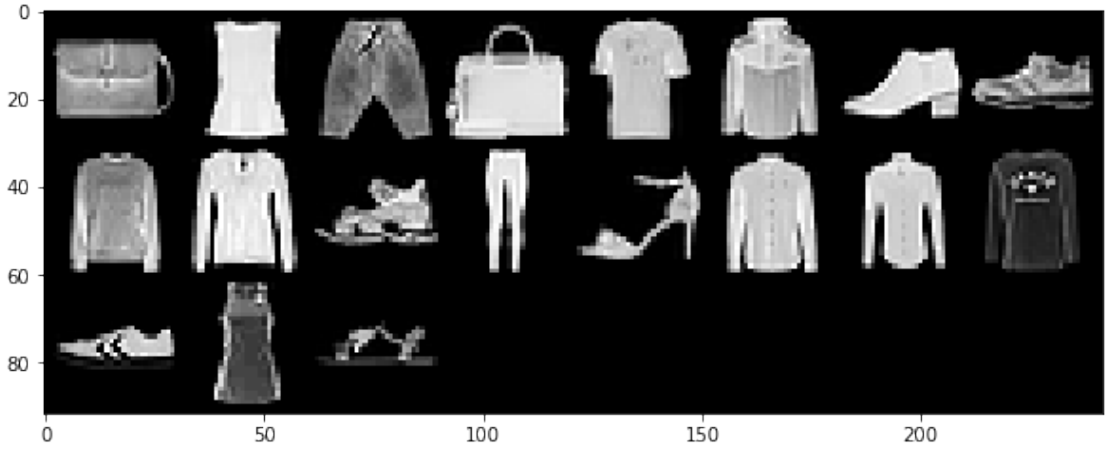



```
Epoch 38
-------------------------------
loss: 0.018655  [    0/60000]
loss: 0.021669  [ 2000/60000]
loss: 0.016499  [ 4000/60000]
loss: 0.020438  [ 6000/60000]
loss: 0.015414  [ 8000/60000]
loss: 0.020468  [10000/60000]
loss: 0.021086  [12000/60000]
loss: 0.019925  [14000/60000]
loss: 0.021110  [16000/60000]
loss: 0.019201  [18000/60000]
loss: 0.021972  [20000/60000]
loss: 0.023051  [22000/60000]
loss: 0.020421  [24000/60000]
loss: 0.017368  [26000/60000]
loss: 0.016481  [28000/60000]
loss: 0.021697  [30000/60000]
loss: 0.020210  [32000/60000]
loss: 0.018858  [34000/60000]
loss: 0.020174  [36000/60000]
loss: 0.022178  [38000/60000]
loss: 0.022560  [40000/60000]
loss: 0.022638  [42000/60000]
loss: 0.021915  [44000/60000]
loss: 0.023116  [46000/60000]
loss: 0.022344  [48000/60000]
loss: 0.019801  [50000/60000]
loss: 0.021418  [52000/60000]
loss: 0.019676  [54000/60000]
loss: 0.026148  [56000/60000]
```

```
loss: 0.019323  [58000/60000]
```

```
Clipping input data to the valid range for imshow with RGB data ([0..1] for
floats or [0..255] for integers).
```

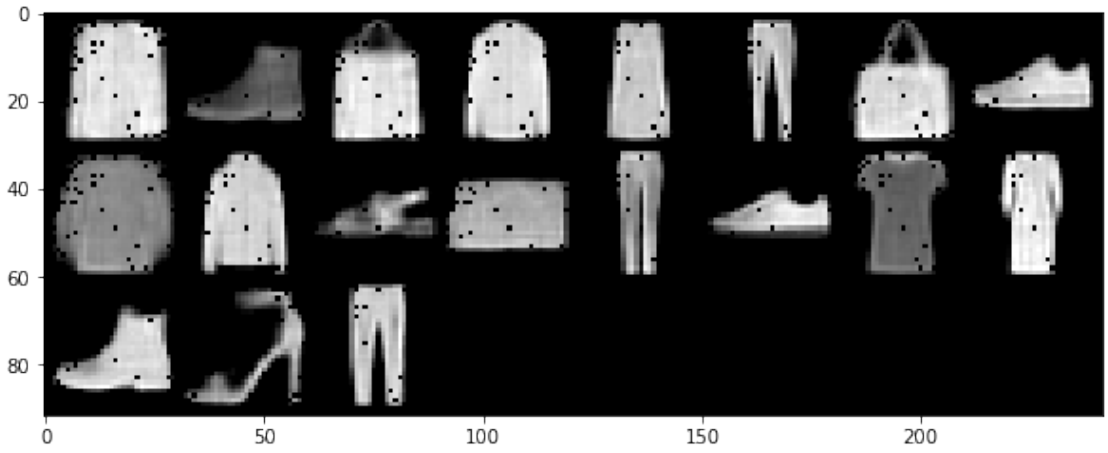


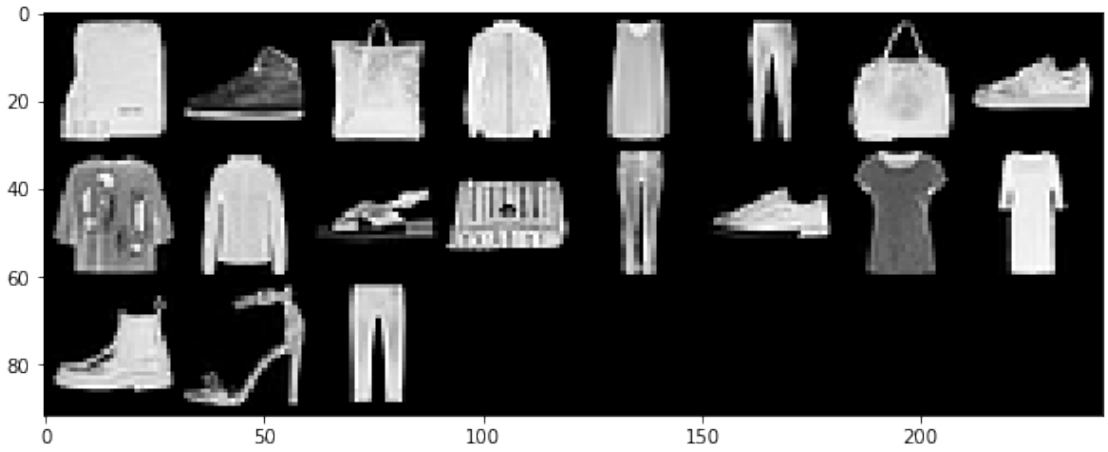




```
Epoch 39
-------------------------------
loss: 0.019403  [    0/60000]
loss: 0.019581  [ 2000/60000]
loss: 0.019670  [ 4000/60000]
loss: 0.016726  [ 6000/60000]
loss: 0.017980  [ 8000/60000]
loss: 0.019704  [10000/60000]
loss: 0.023238  [12000/60000]
loss: 0.016354  [14000/60000]
loss: 0.027597  [16000/60000]
```

```
loss: 0.025917  [18000/60000]
loss: 0.024370  [20000/60000]
loss: 0.022165  [22000/60000]
loss: 0.021091  [24000/60000]
loss: 0.017851  [26000/60000]
loss: 0.018652  [28000/60000]
loss: 0.016726  [30000/60000]
loss: 0.018297  [32000/60000]
loss: 0.020137  [34000/60000]
loss: 0.020412  [36000/60000]
loss: 0.021998  [38000/60000]
loss: 0.020502  [40000/60000]
loss: 0.022155  [42000/60000]
loss: 0.022200  [44000/60000]
loss: 0.015834  [46000/60000]
loss: 0.017855  [48000/60000]
loss: 0.022767  [50000/60000]
loss: 0.018497  [52000/60000]
loss: 0.019268  [54000/60000]
loss: 0.021769  [56000/60000]
loss: 0.020299  [58000/60000]
```

```
Clipping input data to the valid range for imshow with RGB data ([0..1] for
floats or [0..255] for integers).
```

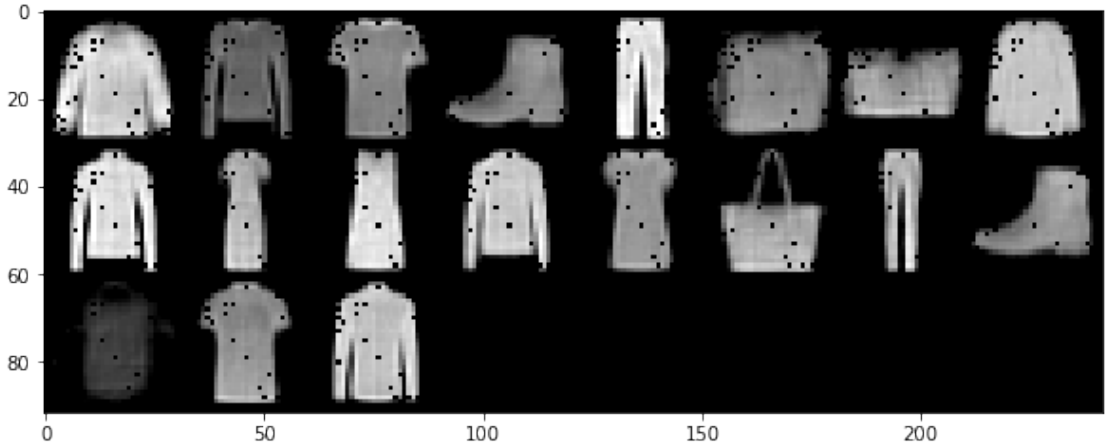


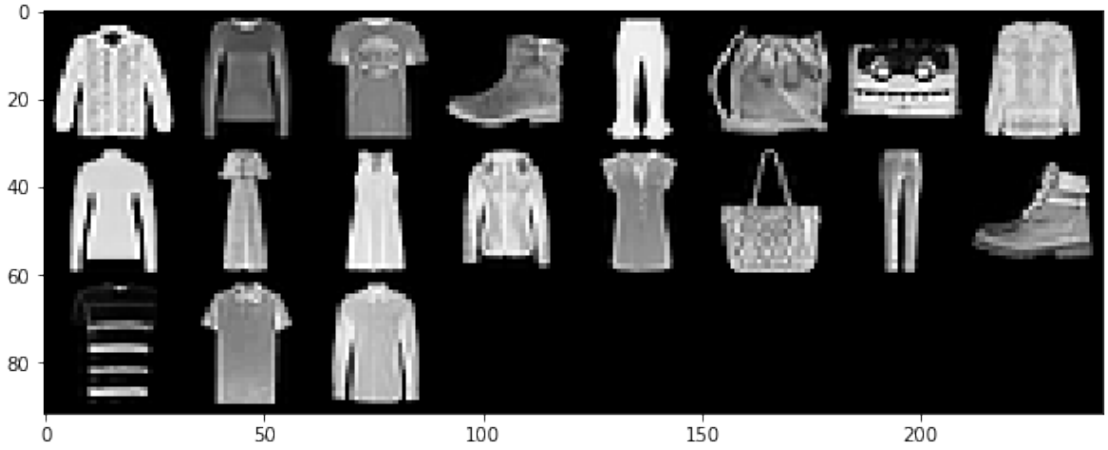



```
Epoch 40
-------------------------------
loss: 0.018506  [    0/60000]
loss: 0.020098  [ 2000/60000]
loss: 0.021390  [ 4000/60000]
loss: 0.020058  [ 6000/60000]
loss: 0.018553  [ 8000/60000]
loss: 0.020519  [10000/60000]
loss: 0.020639  [12000/60000]
loss: 0.020523  [14000/60000]
loss: 0.018812  [16000/60000]
loss: 0.022085  [18000/60000]
loss: 0.022092  [20000/60000]
loss: 0.023047  [22000/60000]
loss: 0.020042  [24000/60000]
loss: 0.019650  [26000/60000]
loss: 0.021517  [28000/60000]
loss: 0.022814  [30000/60000]
loss: 0.021350  [32000/60000]
loss: 0.019845  [34000/60000]
loss: 0.023808  [36000/60000]
loss: 0.019395  [38000/60000]
loss: 0.020171  [40000/60000]
loss: 0.024702  [42000/60000]
loss: 0.022150  [44000/60000]
loss: 0.027668  [46000/60000]
loss: 0.021203  [48000/60000]
loss: 0.021291  [50000/60000]
loss: 0.016714  [52000/60000]
loss: 0.016976  [54000/60000]
loss: 0.022181  [56000/60000]
```

```
loss: 0.023806  [58000/60000]
```

```
Clipping input data to the valid range for imshow with RGB data ([0..1] for
floats or [0..255] for integers).
```

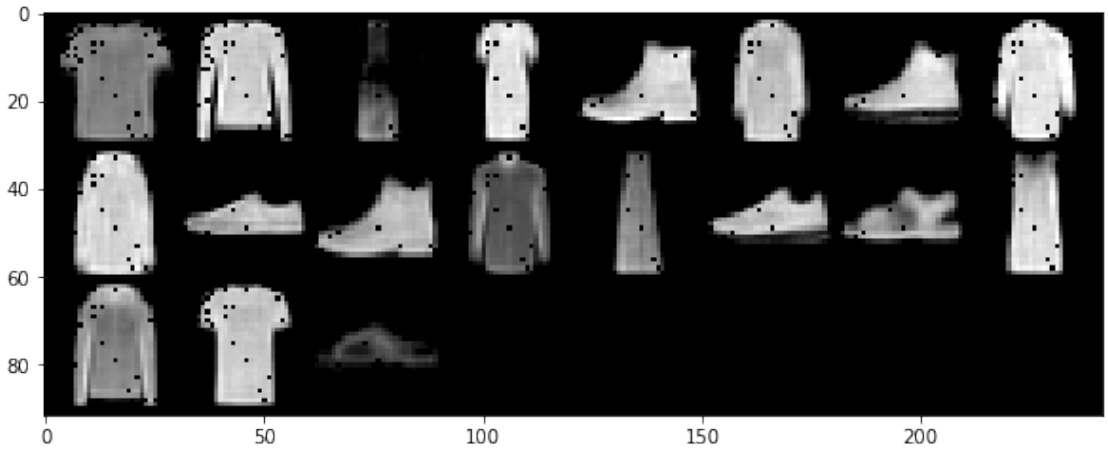


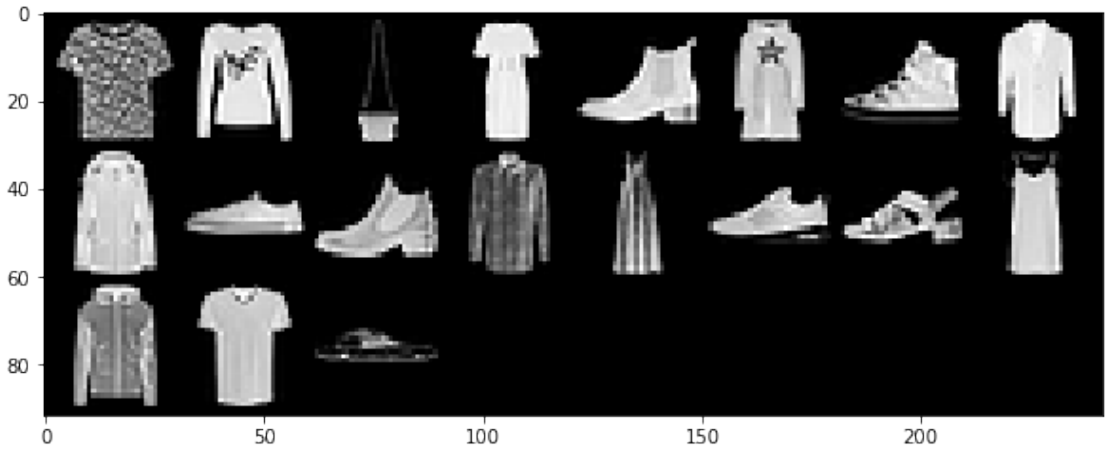




```
Epoch 41
-------------------------------
loss: 0.022632  [    0/60000]
loss: 0.022138  [ 2000/60000]
loss: 0.021411  [ 4000/60000]
loss: 0.017176  [ 6000/60000]
loss: 0.023380  [ 8000/60000]
loss: 0.025110  [10000/60000]
loss: 0.020919  [12000/60000]
loss: 0.017205  [14000/60000]
loss: 0.023631  [16000/60000]
```

```
loss: 0.021229  [18000/60000]
loss: 0.024504  [20000/60000]
loss: 0.020706  [22000/60000]
loss: 0.019479  [24000/60000]
loss: 0.021276  [26000/60000]
loss: 0.023997  [28000/60000]
loss: 0.021158  [30000/60000]
loss: 0.019097  [32000/60000]
loss: 0.019697  [34000/60000]
loss: 0.018298  [36000/60000]
loss: 0.020078  [38000/60000]
loss: 0.017030  [40000/60000]
loss: 0.020063  [42000/60000]
loss: 0.018749  [44000/60000]
loss: 0.019836  [46000/60000]
loss: 0.020321  [48000/60000]
loss: 0.015801  [50000/60000]
loss: 0.023479  [52000/60000]
loss: 0.018787  [54000/60000]
loss: 0.019305  [56000/60000]
loss: 0.022090  [58000/60000]
```

```
Clipping input data to the valid range for imshow with RGB data ([0..1] for
floats or [0..255] for integers).
```

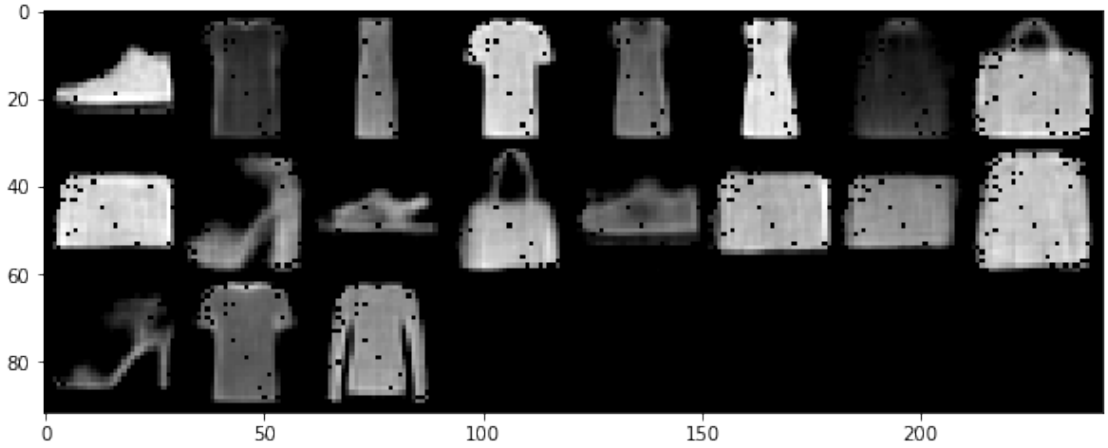


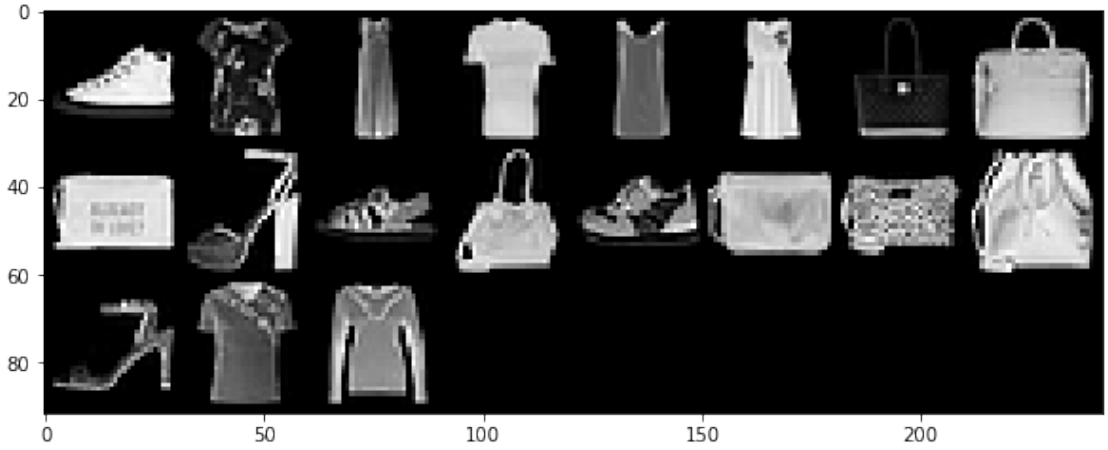



```
Epoch 42
-------------------------------
loss: 0.018878  [    0/60000]
loss: 0.025007  [ 2000/60000]
loss: 0.019187  [ 4000/60000]
loss: 0.025165  [ 6000/60000]
loss: 0.017584  [ 8000/60000]
loss: 0.019908  [10000/60000]
loss: 0.015867  [12000/60000]
loss: 0.017465  [14000/60000]
loss: 0.019521  [16000/60000]
loss: 0.020419  [18000/60000]
loss: 0.019645  [20000/60000]
loss: 0.020285  [22000/60000]
loss: 0.019425  [24000/60000]
loss: 0.019977  [26000/60000]
loss: 0.019300  [28000/60000]
loss: 0.019969  [30000/60000]
loss: 0.021110  [32000/60000]
loss: 0.018856  [34000/60000]
loss: 0.020687  [36000/60000]
loss: 0.021969  [38000/60000]
loss: 0.019272  [40000/60000]
loss: 0.022640  [42000/60000]
loss: 0.017958  [44000/60000]
loss: 0.015882  [46000/60000]
loss: 0.018446  [48000/60000]
loss: 0.018547  [50000/60000]
loss: 0.019449  [52000/60000]
loss: 0.018886  [54000/60000]
loss: 0.021505  [56000/60000]
```

```
loss: 0.018393  [58000/60000]
```

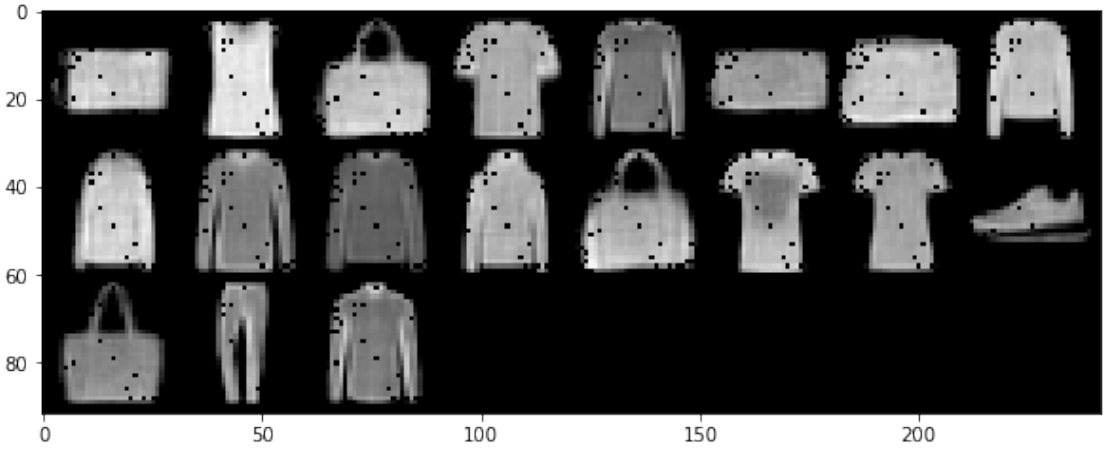
```
Clipping input data to the valid range for imshow with RGB data ([0..1] for
floats or [0..255] for integers).
```

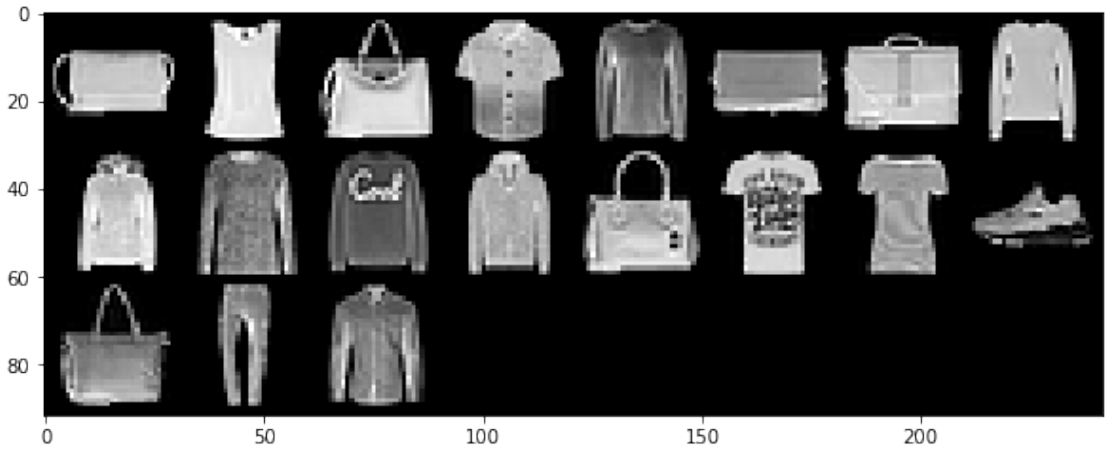



```
Epoch 43
-------------------------------
loss: 0.020619  [    0/60000]
loss: 0.021926  [ 2000/60000]
loss: 0.019719  [ 4000/60000]
loss: 0.025310  [ 6000/60000]
loss: 0.016176  [ 8000/60000]
loss: 0.023019  [10000/60000]
loss: 0.023004  [12000/60000]
loss: 0.019621  [14000/60000]
loss: 0.019266  [16000/60000]
```

```
loss: 0.024505  [18000/60000]
loss: 0.023781  [20000/60000]
loss: 0.023198  [22000/60000]
loss: 0.017011  [24000/60000]
loss: 0.024297  [26000/60000]
loss: 0.020213  [28000/60000]
loss: 0.021891  [30000/60000]
loss: 0.020792  [32000/60000]
loss: 0.020339  [34000/60000]
loss: 0.022277  [36000/60000]
loss: 0.018763  [38000/60000]
loss: 0.022313  [40000/60000]
loss: 0.023352  [42000/60000]
loss: 0.018274  [44000/60000]
loss: 0.027846  [46000/60000]
loss: 0.016117  [48000/60000]
loss: 0.021073  [50000/60000]
loss: 0.019483  [52000/60000]
loss: 0.015896  [54000/60000]
loss: 0.019390  [56000/60000]
loss: 0.019297  [58000/60000]
```

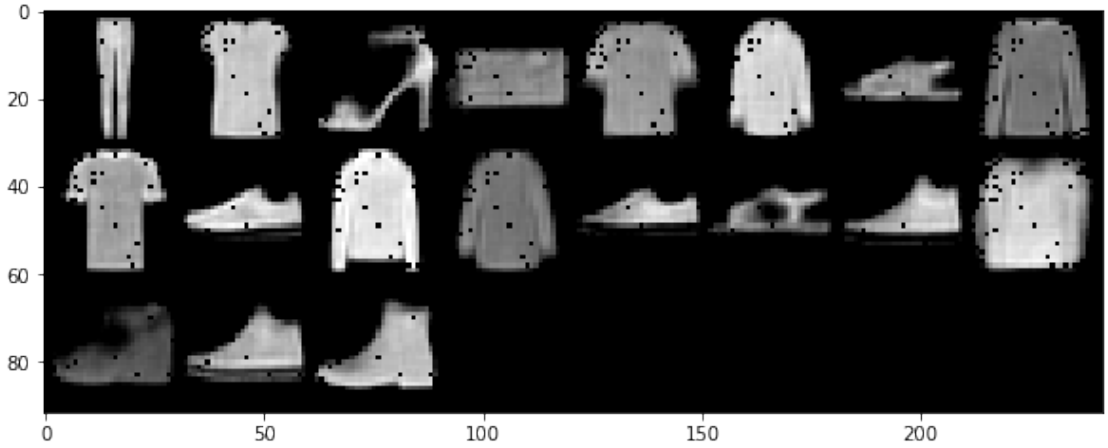


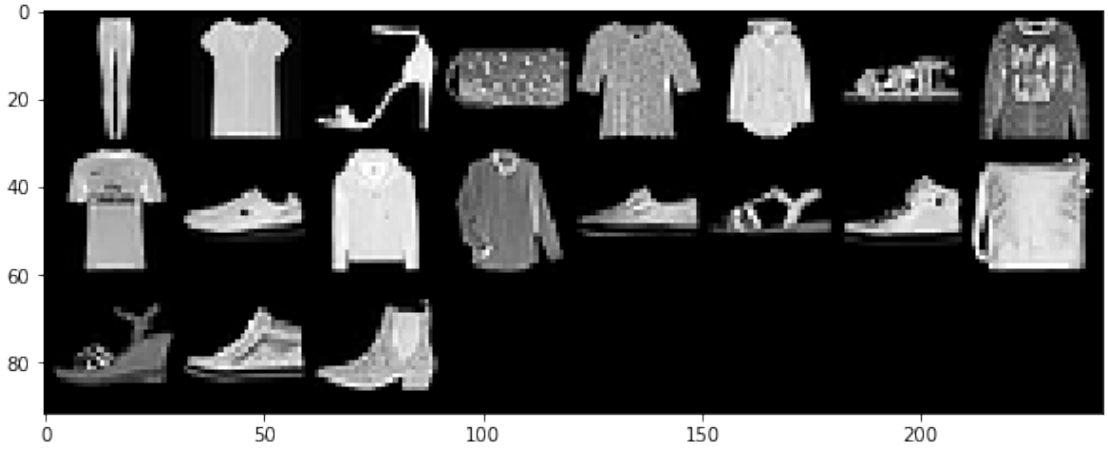



```
Epoch 44
-------------------------------
loss: 0.021824  [    0/60000]
loss: 0.021115  [ 2000/60000]
loss: 0.019348  [ 4000/60000]
loss: 0.023272  [ 6000/60000]
loss: 0.021823  [ 8000/60000]
loss: 0.020300  [10000/60000]
loss: 0.018484  [12000/60000]
loss: 0.021549  [14000/60000]
loss: 0.022298  [16000/60000]
loss: 0.017843  [18000/60000]
loss: 0.022263  [20000/60000]
loss: 0.020754  [22000/60000]
loss: 0.021195  [24000/60000]
loss: 0.024629  [26000/60000]
loss: 0.023288  [28000/60000]
loss: 0.017348  [30000/60000]
loss: 0.020584  [32000/60000]
loss: 0.023830  [34000/60000]
loss: 0.020126  [36000/60000]
loss: 0.018919  [38000/60000]
loss: 0.022283  [40000/60000]
loss: 0.021183  [42000/60000]
loss: 0.018813  [44000/60000]
loss: 0.022027  [46000/60000]
loss: 0.021766  [48000/60000]
loss: 0.019365  [50000/60000]
loss: 0.020046  [52000/60000]
loss: 0.015078  [54000/60000]
loss: 0.024833  [56000/60000]
```

```
loss: 0.018470  [58000/60000]
```

```
Clipping input data to the valid range for imshow with RGB data ([0..1] for
floats or [0..255] for integers).
```

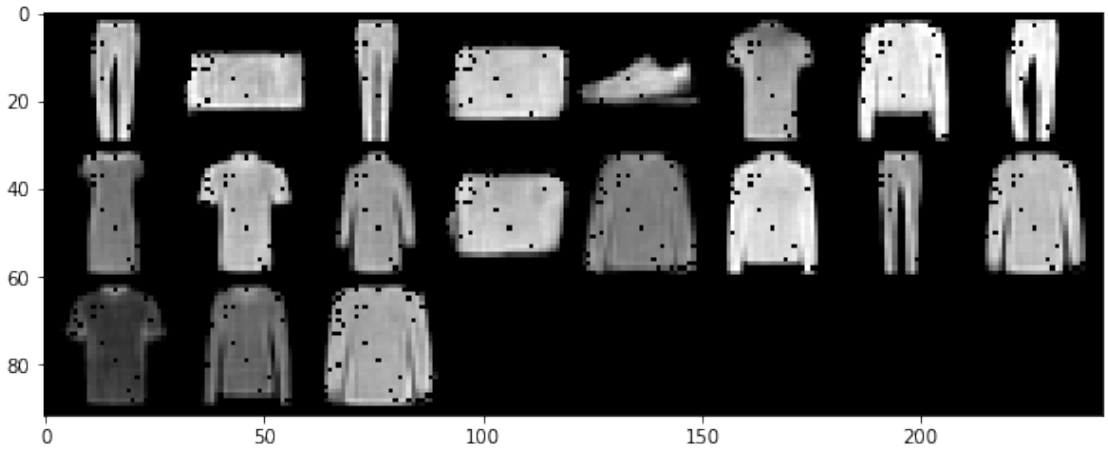


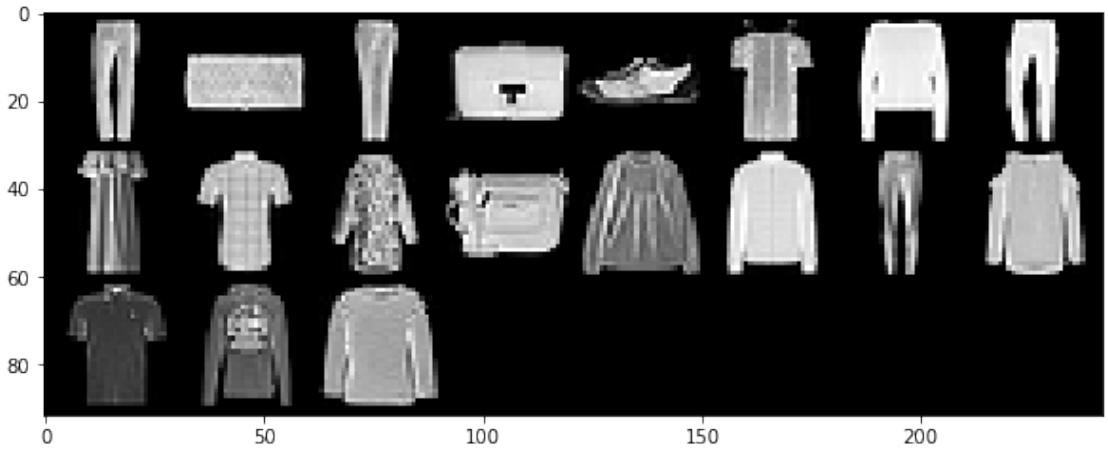




```
Epoch 45
-------------------------------
loss: 0.020394  [    0/60000]
loss: 0.017965  [ 2000/60000]
loss: 0.019894  [ 4000/60000]
loss: 0.023183  [ 6000/60000]
loss: 0.020460  [ 8000/60000]
loss: 0.019179  [10000/60000]
loss: 0.022600  [12000/60000]
loss: 0.022672  [14000/60000]
loss: 0.022673  [16000/60000]
```

```
loss: 0.018469  [18000/60000]
loss: 0.021874  [20000/60000]
loss: 0.023799  [22000/60000]
loss: 0.017960  [24000/60000]
loss: 0.019380  [26000/60000]
loss: 0.021766  [28000/60000]
loss: 0.023100  [30000/60000]
loss: 0.021277  [32000/60000]
loss: 0.020221  [34000/60000]
loss: 0.020684  [36000/60000]
loss: 0.021356  [38000/60000]
loss: 0.021382  [40000/60000]
loss: 0.017517  [42000/60000]
loss: 0.016474  [44000/60000]
loss: 0.019728  [46000/60000]
loss: 0.020826  [48000/60000]
loss: 0.024820  [50000/60000]
loss: 0.019257  [52000/60000]
loss: 0.017883  [54000/60000]
loss: 0.023421  [56000/60000]
loss: 0.023231  [58000/60000]
```

```
Clipping input data to the valid range for imshow with RGB data ([0..1] for
floats or [0..255] for integers).
```

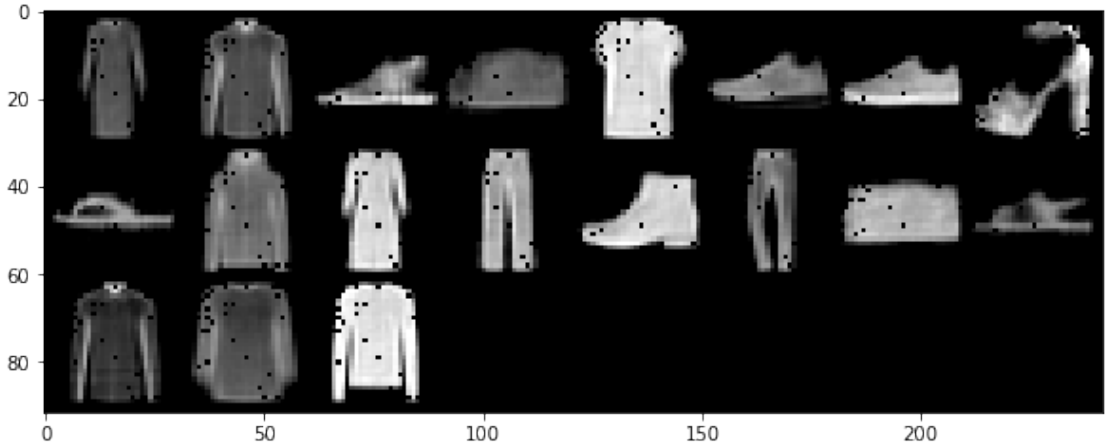


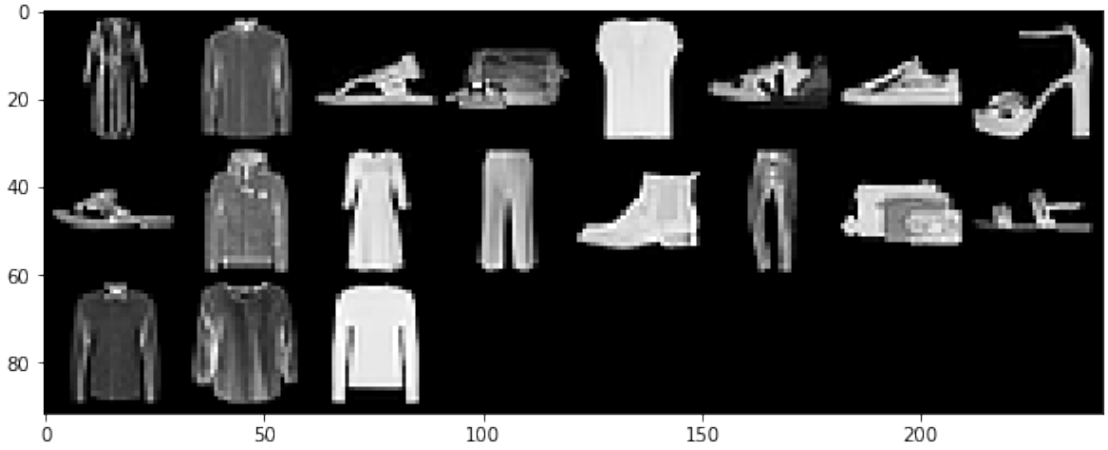



```
Epoch 46
-------------------------------
loss: 0.017607  [    0/60000]
loss: 0.018837  [ 2000/60000]
loss: 0.018152  [ 4000/60000]
loss: 0.018018  [ 6000/60000]
loss: 0.019979  [ 8000/60000]
loss: 0.018226  [10000/60000]
loss: 0.019717  [12000/60000]
loss: 0.017973  [14000/60000]
loss: 0.021881  [16000/60000]
loss: 0.016772  [18000/60000]
loss: 0.021468  [20000/60000]
loss: 0.016289  [22000/60000]
loss: 0.018489  [24000/60000]
loss: 0.018502  [26000/60000]
loss: 0.020534  [28000/60000]
loss: 0.020111  [30000/60000]
loss: 0.020441  [32000/60000]
loss: 0.023965  [34000/60000]
loss: 0.018810  [36000/60000]
loss: 0.025836  [38000/60000]
loss: 0.021926  [40000/60000]
loss: 0.019289  [42000/60000]
loss: 0.020701  [44000/60000]
loss: 0.020931  [46000/60000]
loss: 0.022099  [48000/60000]
loss: 0.016331  [50000/60000]
loss: 0.019379  [52000/60000]
loss: 0.017443  [54000/60000]
loss: 0.017317  [56000/60000]
```

```
loss: 0.019787  [58000/60000]
```

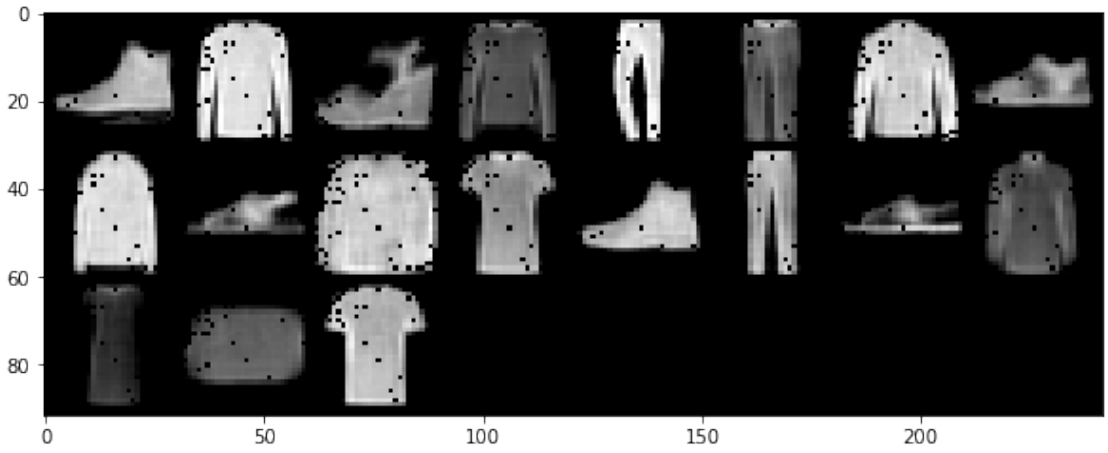
```
Clipping input data to the valid range for imshow with RGB data ([0..1] for
floats or [0..255] for integers).
```

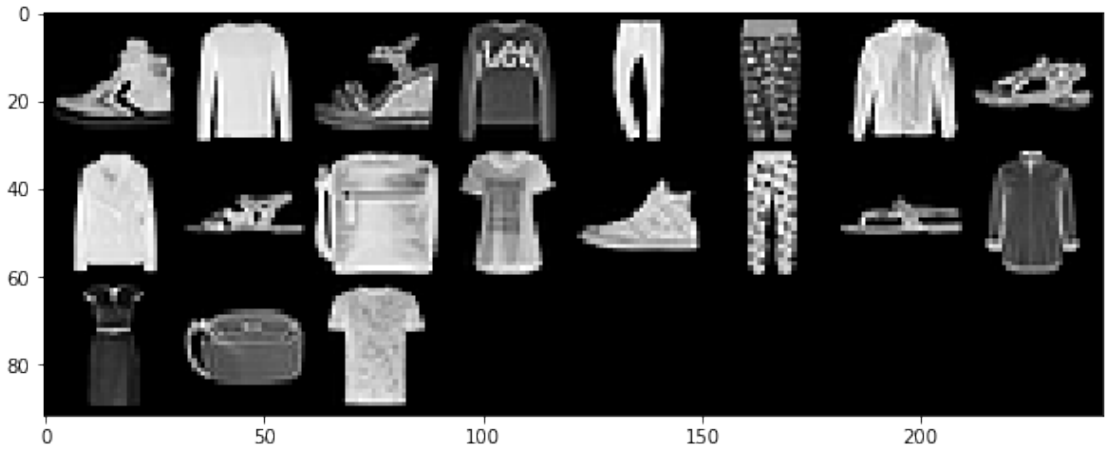



```
Epoch 47
-------------------------------
loss: 0.021529  [    0/60000]
loss: 0.020044  [ 2000/60000]
loss: 0.018913  [ 4000/60000]
loss: 0.024467  [ 6000/60000]
loss: 0.020925  [ 8000/60000]
loss: 0.023977  [10000/60000]
loss: 0.018969  [12000/60000]
loss: 0.015348  [14000/60000]
loss: 0.024129  [16000/60000]
```

```
loss: 0.021634  [18000/60000]
loss: 0.024689  [20000/60000]
loss: 0.020363  [22000/60000]
loss: 0.021758  [24000/60000]
loss: 0.025806  [26000/60000]
loss: 0.017076  [28000/60000]
loss: 0.019597  [30000/60000]
loss: 0.018516  [32000/60000]
loss: 0.017610  [34000/60000]
loss: 0.028584  [36000/60000]
loss: 0.020211  [38000/60000]
loss: 0.021587  [40000/60000]
loss: 0.021631  [42000/60000]
loss: 0.020087  [44000/60000]
loss: 0.017441  [46000/60000]
loss: 0.020233  [48000/60000]
loss: 0.021682  [50000/60000]
loss: 0.019019  [52000/60000]
loss: 0.017866  [54000/60000]
loss: 0.024142  [56000/60000]
loss: 0.020596  [58000/60000]
Clipping input data to the valid range for imshow with RGB data ([0..1] for
floats or [0..255] for integers).
```

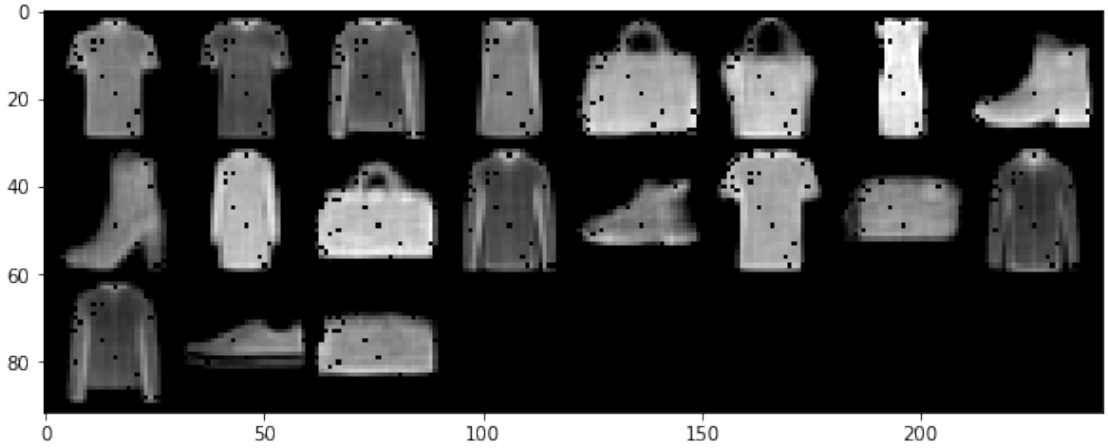


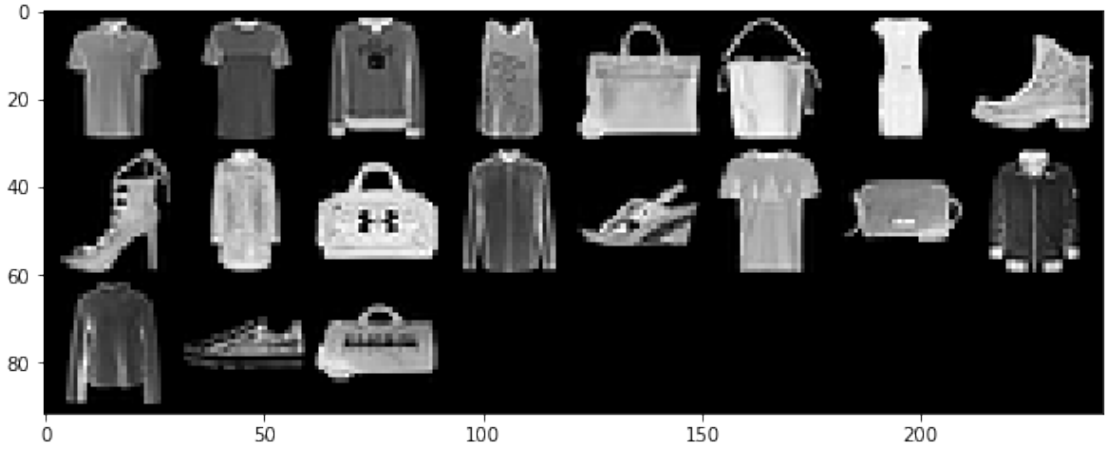



```
Epoch 48
-------------------------------
loss: 0.018205  [    0/60000]
loss: 0.018859  [ 2000/60000]
loss: 0.023561  [ 4000/60000]
loss: 0.020285  [ 6000/60000]
loss: 0.019228  [ 8000/60000]
loss: 0.022633  [10000/60000]
loss: 0.021385  [12000/60000]
loss: 0.020606  [14000/60000]
loss: 0.018464  [16000/60000]
loss: 0.017924  [18000/60000]
loss: 0.017743  [20000/60000]
loss: 0.023514  [22000/60000]
loss: 0.021984  [24000/60000]
loss: 0.020688  [26000/60000]
loss: 0.016557  [28000/60000]
loss: 0.019406  [30000/60000]
loss: 0.019472  [32000/60000]
loss: 0.019781  [34000/60000]
loss: 0.017252  [36000/60000]
loss: 0.016685  [38000/60000]
loss: 0.017790  [40000/60000]
loss: 0.017028  [42000/60000]
loss: 0.025919  [44000/60000]
loss: 0.020951  [46000/60000]
loss: 0.019198  [48000/60000]
loss: 0.021411  [50000/60000]
loss: 0.026309  [52000/60000]
loss: 0.022057  [54000/60000]
loss: 0.018656  [56000/60000]
```

```
loss: 0.019899  [58000/60000]
```

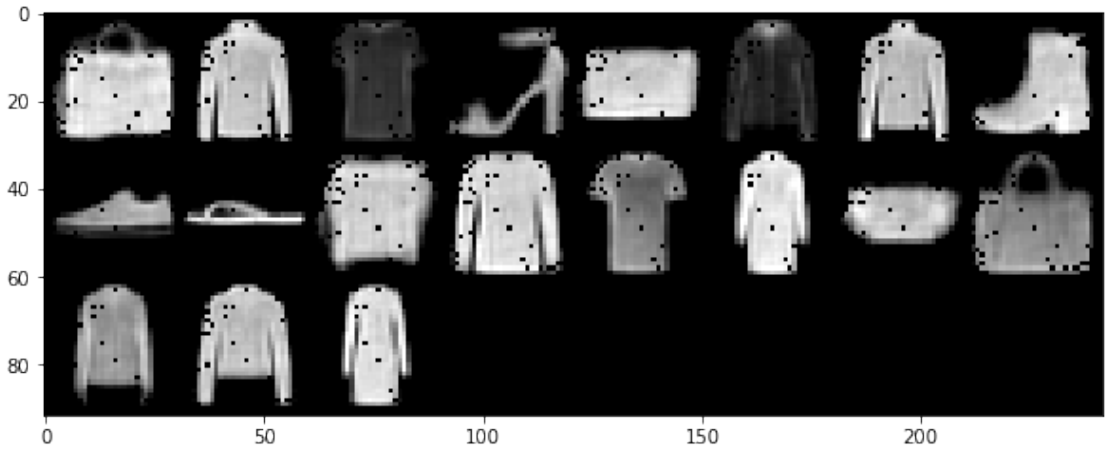
```
Clipping input data to the valid range for imshow with RGB data ([0..1] for
floats or [0..255] for integers).
```

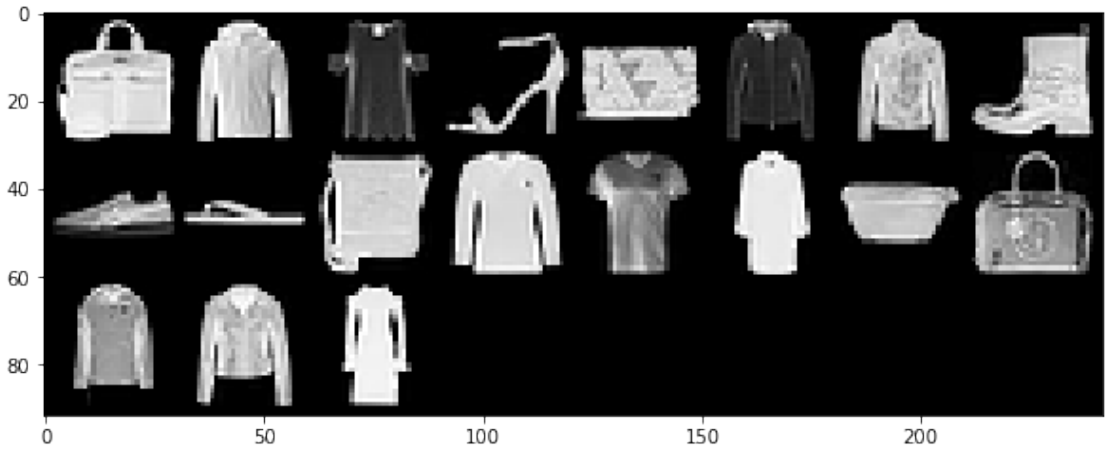



```
Epoch 49
-------------------------------
loss: 0.021558  [    0/60000]
loss: 0.017509  [ 2000/60000]
loss: 0.021441  [ 4000/60000]
loss: 0.025564  [ 6000/60000]
loss: 0.017623  [ 8000/60000]
loss: 0.018052  [10000/60000]
loss: 0.019773  [12000/60000]
loss: 0.019950  [14000/60000]
loss: 0.021478  [16000/60000]
```

```
loss: 0.027499  [18000/60000]
loss: 0.019883  [20000/60000]
loss: 0.019887  [22000/60000]
loss: 0.017078  [24000/60000]
loss: 0.018332  [26000/60000]
loss: 0.020318  [28000/60000]
loss: 0.017559  [30000/60000]
loss: 0.020596  [32000/60000]
loss: 0.017945  [34000/60000]
loss: 0.022808  [36000/60000]
loss: 0.015904  [38000/60000]
loss: 0.019011  [40000/60000]
loss: 0.022442  [42000/60000]
loss: 0.019464  [44000/60000]
loss: 0.022041  [46000/60000]
loss: 0.023036  [48000/60000]
loss: 0.025156  [50000/60000]
loss: 0.021985  [52000/60000]
loss: 0.019208  [54000/60000]
loss: 0.015586  [56000/60000]
loss: 0.022053  [58000/60000]
Clipping input data to the valid range for imshow with RGB data ([0..1] for
floats or [0..255] for integers).
```

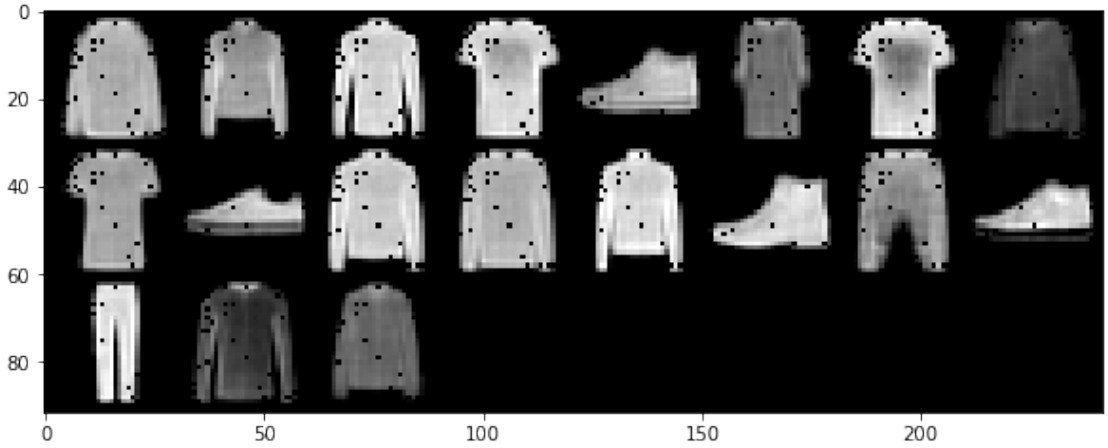


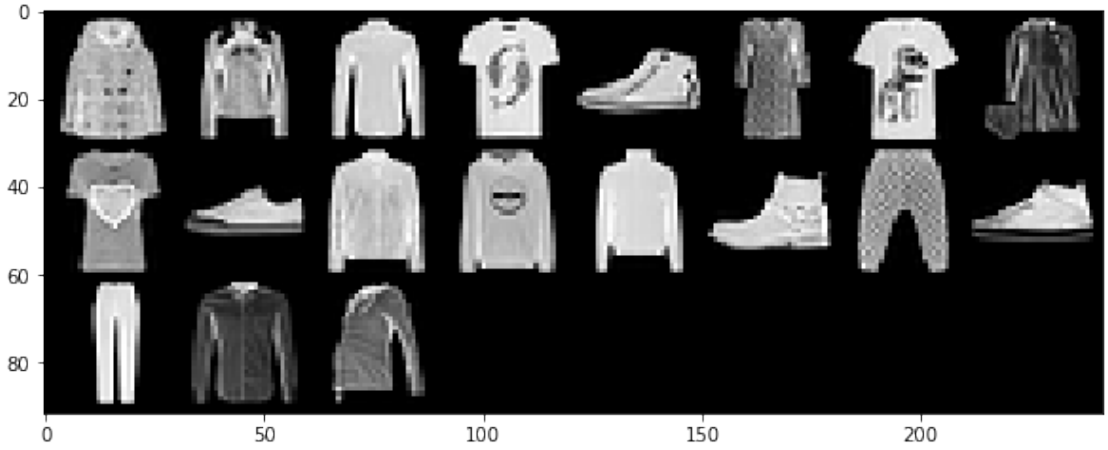



```
Epoch 50
-------------------------------
loss: 0.018151  [    0/60000]
loss: 0.016759  [ 2000/60000]
loss: 0.019554  [ 4000/60000]
loss: 0.018184  [ 6000/60000]
loss: 0.022422  [ 8000/60000]
loss: 0.018607  [10000/60000]
loss: 0.020977  [12000/60000]
loss: 0.024846  [14000/60000]
loss: 0.017212  [16000/60000]
loss: 0.022620  [18000/60000]
loss: 0.014591  [20000/60000]
loss: 0.017303  [22000/60000]
loss: 0.017820  [24000/60000]
loss: 0.018997  [26000/60000]
loss: 0.022222  [28000/60000]
loss: 0.026583  [30000/60000]
loss: 0.018152  [32000/60000]
loss: 0.020181  [34000/60000]
loss: 0.016965  [36000/60000]
loss: 0.016973  [38000/60000]
loss: 0.023406  [40000/60000]
loss: 0.019934  [42000/60000]
loss: 0.018806  [44000/60000]
loss: 0.024410  [46000/60000]
loss: 0.022426  [48000/60000]
loss: 0.021310  [50000/60000]
loss: 0.017159  [52000/60000]
loss: 0.023559  [54000/60000]
loss: 0.021230  [56000/60000]
```

```
loss: 0.019309  [58000/60000]
```

```
Clipping input data to the valid range for imshow with RGB data ([0..1] for
floats or [0..255] for integers).
```




```
Done!
```

```
[ ]:
```