

# fashion\_mnist

February 5, 2020

```
In [0]: #from google.colab import drive
        #drive.mount('/content/drive')

        data_dir = '/content/drive/My Drive/cs461/data'
```

```
In [0]: import matplotlib.pyplot as plt
        %matplotlib inline

        import torch
        import torch.nn as nn

        import torch.nn.functional as F
        import torchvision
        import torchvision.transforms as transforms
```

```
In [0]: class Args:
        def __init__(self):
            self.use_cuda = True
            self.log_interval = 20
            self.train_batch_size = 64
            self.test_batch_size = 64
            self.lr = 0.01
            self.momentum = 0.9
            self.num_epochs = 3
```

```
args = Args()
```

```
In [11]: device = torch.device('cpu')
         if args.use_cuda and torch.cuda.is_available():
             device = torch.device('cuda:0')

         print('Using {}'.format(device))
```

Using cuda:0.

Load dataset and create data loaders.

```

In [0]: def prepare_data(args):
    kwargs = {}
    if args.use_cuda and torch.cuda.is_available():
        kwargs = {'num_workers': 1, 'pin_memory': True}

    transform = transforms.Compose([
        transforms.ToTensor(),
        transforms.Normalize((0.5,), (0.5,)),
    ])

    # Load original training data.
    trainval_set = torchvision.datasets.FashionMNIST(
        root=data_dir, train=True, download=True, transform=transform)

    # Split original training data into training set and validation set
    # and create data loaders.
    train_set = torch.utils.data.Subset(trainval_set, range(50000))
    train_loader = torch.utils.data.DataLoader(
        train_set, batch_size=args.train_batch_size, shuffle=True, **kwargs)

    val_set = torch.utils.data.Subset(trainval_set, range(50000, 60000))
    val_loader = torch.utils.data.DataLoader(
        val_set, batch_size=args.test_batch_size, shuffle=False, **kwargs)

    # Load testing data and create data loader.
    test_set = torchvision.datasets.FashionMNIST(
        root=data_dir, train=False, download=True, transform=transform)
    test_loader = torch.utils.data.DataLoader(
        test_set, batch_size=args.test_batch_size, shuffle=False, **kwargs)

    return train_set, val_set, test_set, train_loader, val_loader, test_loader

```

Visualize some training samples.

```

In [13]: train_set, val_set, test_set, train_loader, val_loader, test_loader = prepare_data(args)

fig, axes = plt.subplots(1, 10)
for idx, ax in enumerate(axes):
    img, lbl = train_set[idx]
    ax.imshow(img[0], cmap='gray')
    ax.set_title(lbl)
    ax.set_xticklabels([])
    ax.set_yticklabels([])
plt.show()

```

Out [00:00, ?it/s]

Downloading <http://fashion-mnist.s3-website.eu-central-1.amazonaws.com/train-images-idx3-ubyte.g>

26427392it [00:04, 5806910.29it/s]

Extracting /content/drive/My Drive/cs461/data/FashionMNIST/raw/train-images-idx3-ubyte.gz to /co

0it [00:00, ?it/s]

Downloading http://fashion-mnist.s3-website.eu-central-1.amazonaws.com/train-labels-idx1-ubyte.g

32768it [00:00, 41275.66it/s]

0it [00:00, ?it/s]

Extracting /content/drive/My Drive/cs461/data/FashionMNIST/raw/train-labels-idx1-ubyte.gz to /co

Downloading http://fashion-mnist.s3-website.eu-central-1.amazonaws.com/t10k-images-idx3-ubyte.gz

4423680it [00:02, 1661211.92it/s]

0it [00:00, ?it/s]

Extracting /content/drive/My Drive/cs461/data/FashionMNIST/raw/t10k-images-idx3-ubyte.gz to /con

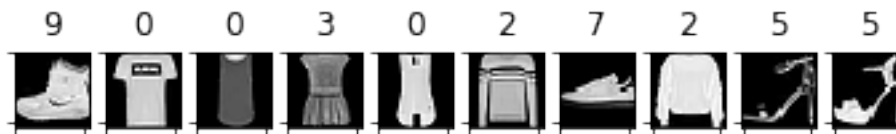
Downloading http://fashion-mnist.s3-website.eu-central-1.amazonaws.com/t10k-labels-idx1-ubyte.gz

8192it [00:00, 13927.94it/s]

Extracting /content/drive/My Drive/cs461/data/FashionMNIST/raw/t10k-labels-idx1-ubyte.gz to /con

Processing...

Done!



Define the model.

```
In [0]: class Net(nn.Module):
        def __init__(self):
            super(Net, self).__init__()
            self.conv1 = nn.Conv2d(1, 6, 5)
            self.conv2 = nn.Conv2d(6, 16, 5)
            self.fc3 = nn.Linear(256, 10)
```

```

        self._initialize_weights()

def _initialize_weights(self):
    pass

def forward(self, x):
    x = F.relu(self.conv1(x))
    x = F.max_pool2d(x, 2, stride=2)
    x = F.relu(self.conv2(x))
    x = F.max_pool2d(x, 2, stride=2)
    x = torch.flatten(x, start_dim=1)
    x = self.fc3(x)
    return x

```

Define the training function.

```

In [0]: def train(args, model, criterion, train_loader, optimizer, device):
    model.train()
    total_loss = 0.
    for i, data in enumerate(train_loader):
        imgs, lbls = data[0].to(device), data[1].to(device)

        optimizer.zero_grad()
        outputs = model(imgs)
        loss = criterion(outputs, lbls)
        loss.backward()
        optimizer.step()

        total_loss += loss.item()
        if (i + 1) % args.log_interval == 0:
            mean_loss = total_loss / args.log_interval
            print(' batch {:4d}: loss={:.3f}'.format(i + 1, mean_loss))
            total_loss = 0.

```

Define the testing function.

```

In [0]: def test(args, model, test_loader, device):
    model.eval()
    total, correct = 0, 0
    with torch.no_grad():
        for data in test_loader:
            imgs, lbls = data[0].to(device), data[1].to(device)
            outputs = model(imgs)
            _, preds = torch.max(outputs.data, 1)
            total += lbls.shape[0]
            correct += (preds == lbls).sum().item()

    acc = correct / total
    print(' acc={:.3f}'.format(acc))

```

Run training and validation.

```
In [0]: model = Net()
        model.to(device)

        criterion = nn.CrossEntropyLoss()
        optimizer = torch.optim.SGD(
            model.parameters(),
            lr=args.lr,
            momentum=args.momentum
        )

        for e in range(args.num_epochs):
            print('Training epoch {}'.format(e))
            train(args, model, criterion, train_loader, optimizer, device)
            print('Testing on validation set')
            test(args, model, val_loader, device)
```

Training epoch 0

```
batch 20: loss=2.220
batch 40: loss=1.519
batch 60: loss=0.880
batch 80: loss=0.774
batch 100: loss=0.773
batch 120: loss=0.721
batch 140: loss=0.658
batch 160: loss=0.667
batch 180: loss=0.613
batch 200: loss=0.633
batch 220: loss=0.609
batch 240: loss=0.612
batch 260: loss=0.609
batch 280: loss=0.585
batch 300: loss=0.533
batch 320: loss=0.515
batch 340: loss=0.611
batch 360: loss=0.556
batch 380: loss=0.553
batch 400: loss=0.486
batch 420: loss=0.490
batch 440: loss=0.482
batch 460: loss=0.514
batch 480: loss=0.469
batch 500: loss=0.474
batch 520: loss=0.483
batch 540: loss=0.451
batch 560: loss=0.490
batch 580: loss=0.500
```

batch 600: loss=0.433  
batch 620: loss=0.515  
batch 640: loss=0.449  
batch 660: loss=0.473  
batch 680: loss=0.418  
batch 700: loss=0.464  
batch 720: loss=0.440  
batch 740: loss=0.415  
batch 760: loss=0.436  
batch 780: loss=0.409

Testing on validation set  
acc=0.846

Training epoch 1

batch 20: loss=0.360  
batch 40: loss=0.464  
batch 60: loss=0.448  
batch 80: loss=0.405  
batch 100: loss=0.431  
batch 120: loss=0.416  
batch 140: loss=0.384  
batch 160: loss=0.400  
batch 180: loss=0.404  
batch 200: loss=0.365  
batch 220: loss=0.425  
batch 240: loss=0.397  
batch 260: loss=0.412  
batch 280: loss=0.447  
batch 300: loss=0.425  
batch 320: loss=0.401  
batch 340: loss=0.388  
batch 360: loss=0.404  
batch 380: loss=0.405  
batch 400: loss=0.419  
batch 420: loss=0.356  
batch 440: loss=0.393  
batch 460: loss=0.390  
batch 480: loss=0.376  
batch 500: loss=0.349  
batch 520: loss=0.381  
batch 540: loss=0.365  
batch 560: loss=0.391  
batch 580: loss=0.404  
batch 600: loss=0.379  
batch 620: loss=0.365  
batch 640: loss=0.399  
batch 660: loss=0.362  
batch 680: loss=0.401  
batch 700: loss=0.416

```
batch 720: loss=0.409
batch 740: loss=0.366
batch 760: loss=0.379
batch 780: loss=0.360
```

```
Testing on validation set
acc=0.863
```

```
Training epoch 2
```

```
batch 20: loss=0.383
batch 40: loss=0.352
batch 60: loss=0.378
batch 80: loss=0.353
batch 100: loss=0.333
batch 120: loss=0.351
batch 140: loss=0.364
batch 160: loss=0.388
batch 180: loss=0.364
batch 200: loss=0.347
batch 220: loss=0.393
batch 240: loss=0.342
batch 260: loss=0.340
batch 280: loss=0.348
batch 300: loss=0.314
batch 320: loss=0.378
batch 340: loss=0.367
batch 360: loss=0.380
batch 380: loss=0.332
batch 400: loss=0.368
batch 420: loss=0.337
batch 440: loss=0.342
batch 460: loss=0.351
batch 480: loss=0.342
batch 500: loss=0.316
batch 520: loss=0.374
batch 540: loss=0.320
batch 560: loss=0.356
batch 580: loss=0.325
batch 600: loss=0.334
batch 620: loss=0.362
batch 640: loss=0.410
batch 660: loss=0.342
batch 680: loss=0.372
batch 700: loss=0.363
batch 720: loss=0.328
batch 740: loss=0.340
batch 760: loss=0.328
batch 780: loss=0.324
```

```
Testing on validation set
acc=0.875
```

Question 1: Evaluating trained model on test set

```
In [0]: test(args, model, test_loader, device)
```

```
acc=0.870
```

Question 2: Modifying model & training scheme for improvement

Part 1: Smaller batch size

```
In [0]: model1 = Net()
        model1.to(device)

        criterion = nn.CrossEntropyLoss()
        optimizer = torch.optim.SGD(
            model1.parameters(),
            lr=args.lr,
            momentum=args.momentum
        )

        args.train_batch_size = 32

        for e in range(args.num_epochs):
            print('Training epoch {}'.format(e))
            train(args, model1, criterion, train_loader, optimizer, device)
            print('Testing on validation set')
            test(args, model1, val_loader, device)
```

Training epoch 0

```
batch 20: loss=2.214
batch 40: loss=1.488
batch 60: loss=1.027
batch 80: loss=0.858
batch 100: loss=0.797
batch 120: loss=0.744
batch 140: loss=0.680
batch 160: loss=0.659
batch 180: loss=0.640
batch 200: loss=0.660
batch 220: loss=0.611
batch 240: loss=0.599
batch 260: loss=0.603
batch 280: loss=0.582
batch 300: loss=0.547
batch 320: loss=0.566
batch 340: loss=0.545
batch 360: loss=0.566
```



```
batch 380: loss=0.530
batch 400: loss=0.518
batch 420: loss=0.526
batch 440: loss=0.506
batch 460: loss=0.464
batch 480: loss=0.469
batch 500: loss=0.466
batch 520: loss=0.435
batch 540: loss=0.472
batch 560: loss=0.494
batch 580: loss=0.443
batch 600: loss=0.510
batch 620: loss=0.434
batch 640: loss=0.441
batch 660: loss=0.450
batch 680: loss=0.432
batch 700: loss=0.474
batch 720: loss=0.451
batch 740: loss=0.411
batch 760: loss=0.421
batch 780: loss=0.407
Testing on validation set
acc=0.844
```

Training epoch 1

```
batch 20: loss=0.405
batch 40: loss=0.410
batch 60: loss=0.428
batch 80: loss=0.436
batch 100: loss=0.414
batch 120: loss=0.385
batch 140: loss=0.405
batch 160: loss=0.424
batch 180: loss=0.415
batch 200: loss=0.398
batch 220: loss=0.384
batch 240: loss=0.448
batch 260: loss=0.388
batch 280: loss=0.423
batch 300: loss=0.364
batch 320: loss=0.409
batch 340: loss=0.390
batch 360: loss=0.417
batch 380: loss=0.377
batch 400: loss=0.411
batch 420: loss=0.417
batch 440: loss=0.431
batch 460: loss=0.439
batch 480: loss=0.408
```

batch 500: loss=0.377  
batch 520: loss=0.366  
batch 540: loss=0.388  
batch 560: loss=0.425  
batch 580: loss=0.351  
batch 600: loss=0.414  
batch 620: loss=0.375  
batch 640: loss=0.402  
batch 660: loss=0.379  
batch 680: loss=0.345  
batch 700: loss=0.411  
batch 720: loss=0.359  
batch 740: loss=0.403  
batch 760: loss=0.326  
batch 780: loss=0.368

Testing on validation set  
acc=0.865

Training epoch 2

batch 20: loss=0.385  
batch 40: loss=0.353  
batch 60: loss=0.401  
batch 80: loss=0.320  
batch 100: loss=0.378  
batch 120: loss=0.328  
batch 140: loss=0.357  
batch 160: loss=0.348  
batch 180: loss=0.359  
batch 200: loss=0.370  
batch 220: loss=0.376  
batch 240: loss=0.340  
batch 260: loss=0.326  
batch 280: loss=0.348  
batch 300: loss=0.345  
batch 320: loss=0.363  
batch 340: loss=0.344  
batch 360: loss=0.343  
batch 380: loss=0.334  
batch 400: loss=0.365  
batch 420: loss=0.351  
batch 440: loss=0.382  
batch 460: loss=0.377  
batch 480: loss=0.356  
batch 500: loss=0.348  
batch 520: loss=0.383  
batch 540: loss=0.345  
batch 560: loss=0.344  
batch 580: loss=0.314  
batch 600: loss=0.354

```
batch 620: loss=0.319
batch 640: loss=0.375
batch 660: loss=0.362
batch 680: loss=0.349
batch 700: loss=0.339
batch 720: loss=0.384
batch 740: loss=0.366
batch 760: loss=0.315
batch 780: loss=0.379
Testing on validation set
acc=0.874
```

A smaller batch size of 32 decreased accuracy very slightly.  
Part 2: More epochs (original batch size)

```
In [0]: model2 = Net()
        model2.to(device)

        criterion = nn.CrossEntropyLoss()
        optimizer = torch.optim.SGD(
            model2.parameters(),
            lr=args.lr,
            momentum=args.momentum
        )

        args.num_epochs = 10
        args.train_batch_size = 64

        for e in range(args.num_epochs):
            print('Training epoch {}'.format(e))
            train(args, model2, criterion, train_loader, optimizer, device)
            print('Testing on validation set')
            test(args, model2, val_loader, device)
```

```
Training epoch 0
batch 20: loss=2.184
batch 40: loss=1.409
batch 60: loss=0.972
batch 80: loss=0.875
batch 100: loss=0.827
batch 120: loss=0.731
batch 140: loss=0.642
batch 160: loss=0.642
batch 180: loss=0.583
batch 200: loss=0.618
batch 220: loss=0.651
batch 240: loss=0.583
```

```
batch 260: loss=0.569
batch 280: loss=0.580
batch 300: loss=0.532
batch 320: loss=0.526
batch 340: loss=0.554
batch 360: loss=0.480
batch 380: loss=0.477
batch 400: loss=0.489
batch 420: loss=0.497
batch 440: loss=0.534
batch 460: loss=0.513
batch 480: loss=0.504
batch 500: loss=0.437
batch 520: loss=0.472
batch 540: loss=0.463
batch 560: loss=0.448
batch 580: loss=0.464
batch 600: loss=0.491
batch 620: loss=0.437
batch 640: loss=0.414
batch 660: loss=0.467
batch 680: loss=0.477
batch 700: loss=0.449
batch 720: loss=0.427
batch 740: loss=0.406
batch 760: loss=0.410
batch 780: loss=0.426
Testing on validation set
acc=0.841
```

Training epoch 1

```
batch 20: loss=0.432
batch 40: loss=0.447
batch 60: loss=0.412
batch 80: loss=0.365
batch 100: loss=0.425
batch 120: loss=0.425
batch 140: loss=0.425
batch 160: loss=0.382
batch 180: loss=0.417
batch 200: loss=0.398
batch 220: loss=0.378
batch 240: loss=0.408
batch 260: loss=0.390
batch 280: loss=0.396
batch 300: loss=0.406
batch 320: loss=0.431
batch 340: loss=0.414
batch 360: loss=0.387
```

```
batch 380: loss=0.350
batch 400: loss=0.373
batch 420: loss=0.411
batch 440: loss=0.394
batch 460: loss=0.378
batch 480: loss=0.360
batch 500: loss=0.385
batch 520: loss=0.380
batch 540: loss=0.368
batch 560: loss=0.420
batch 580: loss=0.353
batch 600: loss=0.357
batch 620: loss=0.394
batch 640: loss=0.387
batch 660: loss=0.392
batch 680: loss=0.361
batch 700: loss=0.393
batch 720: loss=0.379
batch 740: loss=0.368
batch 760: loss=0.393
batch 780: loss=0.354
Testing on validation set
acc=0.854
```

Training epoch 2

```
batch 20: loss=0.359
batch 40: loss=0.350
batch 60: loss=0.347
batch 80: loss=0.351
batch 100: loss=0.366
batch 120: loss=0.380
batch 140: loss=0.379
batch 160: loss=0.344
batch 180: loss=0.352
batch 200: loss=0.370
batch 220: loss=0.343
batch 240: loss=0.371
batch 260: loss=0.345
batch 280: loss=0.342
batch 300: loss=0.365
batch 320: loss=0.388
batch 340: loss=0.312
batch 360: loss=0.352
batch 380: loss=0.341
batch 400: loss=0.351
batch 420: loss=0.367
batch 440: loss=0.364
batch 460: loss=0.361
batch 480: loss=0.323
```

```
batch 500: loss=0.360
batch 520: loss=0.353
batch 540: loss=0.363
batch 560: loss=0.347
batch 580: loss=0.331
batch 600: loss=0.306
batch 620: loss=0.349
batch 640: loss=0.340
batch 660: loss=0.352
batch 680: loss=0.320
batch 700: loss=0.344
batch 720: loss=0.361
batch 740: loss=0.307
batch 760: loss=0.388
batch 780: loss=0.357
```

```
Testing on validation set
acc=0.870
```

```
Training epoch 3
```

```
batch 20: loss=0.330
batch 40: loss=0.346
batch 60: loss=0.306
batch 80: loss=0.325
batch 100: loss=0.302
batch 120: loss=0.308
batch 140: loss=0.354
batch 160: loss=0.318
batch 180: loss=0.307
batch 200: loss=0.325
batch 220: loss=0.318
batch 240: loss=0.337
batch 260: loss=0.338
batch 280: loss=0.323
batch 300: loss=0.292
batch 320: loss=0.344
batch 340: loss=0.335
batch 360: loss=0.332
batch 380: loss=0.345
batch 400: loss=0.325
batch 420: loss=0.349
batch 440: loss=0.323
batch 460: loss=0.334
batch 480: loss=0.345
batch 500: loss=0.328
batch 520: loss=0.318
batch 540: loss=0.298
batch 560: loss=0.306
batch 580: loss=0.368
batch 600: loss=0.347
```

```
batch 620: loss=0.339
batch 640: loss=0.313
batch 660: loss=0.352
batch 680: loss=0.331
batch 700: loss=0.286
batch 720: loss=0.367
batch 740: loss=0.311
batch 760: loss=0.307
batch 780: loss=0.338
Testing on validation set
acc=0.873
```

Training epoch 4

```
batch 20: loss=0.307
batch 40: loss=0.306
batch 60: loss=0.315
batch 80: loss=0.340
batch 100: loss=0.312
batch 120: loss=0.308
batch 140: loss=0.281
batch 160: loss=0.284
batch 180: loss=0.329
batch 200: loss=0.315
batch 220: loss=0.321
batch 240: loss=0.294
batch 260: loss=0.291
batch 280: loss=0.320
batch 300: loss=0.331
batch 320: loss=0.320
batch 340: loss=0.320
batch 360: loss=0.296
batch 380: loss=0.288
batch 400: loss=0.336
batch 420: loss=0.327
batch 440: loss=0.325
batch 460: loss=0.340
batch 480: loss=0.325
batch 500: loss=0.294
batch 520: loss=0.292
batch 540: loss=0.334
batch 560: loss=0.335
batch 580: loss=0.297
batch 600: loss=0.313
batch 620: loss=0.325
batch 640: loss=0.323
batch 660: loss=0.327
batch 680: loss=0.275
batch 700: loss=0.292
batch 720: loss=0.326
```

```
batch 740: loss=0.335
batch 760: loss=0.293
batch 780: loss=0.284
Testing on validation set
acc=0.875
```

Training epoch 5

```
batch 20: loss=0.318
batch 40: loss=0.334
batch 60: loss=0.308
batch 80: loss=0.288
batch 100: loss=0.300
batch 120: loss=0.310
batch 140: loss=0.289
batch 160: loss=0.291
batch 180: loss=0.289
batch 200: loss=0.321
batch 220: loss=0.307
batch 240: loss=0.285
batch 260: loss=0.313
batch 280: loss=0.313
batch 300: loss=0.320
batch 320: loss=0.263
batch 340: loss=0.266
batch 360: loss=0.329
batch 380: loss=0.316
batch 400: loss=0.308
batch 420: loss=0.295
batch 440: loss=0.324
batch 460: loss=0.322
batch 480: loss=0.281
batch 500: loss=0.303
batch 520: loss=0.302
batch 540: loss=0.334
batch 560: loss=0.281
batch 580: loss=0.291
batch 600: loss=0.276
batch 620: loss=0.272
batch 640: loss=0.300
batch 660: loss=0.319
batch 680: loss=0.300
batch 700: loss=0.305
batch 720: loss=0.289
batch 740: loss=0.310
batch 760: loss=0.295
batch 780: loss=0.282
Testing on validation set
acc=0.876
```

Training epoch 6



```
batch 20: loss=0.301
batch 40: loss=0.294
batch 60: loss=0.265
batch 80: loss=0.282
batch 100: loss=0.296
batch 120: loss=0.314
batch 140: loss=0.329
batch 160: loss=0.276
batch 180: loss=0.268
batch 200: loss=0.312
batch 220: loss=0.306
batch 240: loss=0.281
batch 260: loss=0.325
batch 280: loss=0.289
batch 300: loss=0.331
batch 320: loss=0.283
batch 340: loss=0.334
batch 360: loss=0.302
batch 380: loss=0.282
batch 400: loss=0.268
batch 420: loss=0.311
batch 440: loss=0.308
batch 460: loss=0.281
batch 480: loss=0.287
batch 500: loss=0.297
batch 520: loss=0.316
batch 540: loss=0.246
batch 560: loss=0.287
batch 580: loss=0.296
batch 600: loss=0.221
batch 620: loss=0.282
batch 640: loss=0.259
batch 660: loss=0.316
batch 680: loss=0.289
batch 700: loss=0.287
batch 720: loss=0.320
batch 740: loss=0.313
batch 760: loss=0.307
batch 780: loss=0.311
Testing on validation set
acc=0.871
Training epoch 7
batch 20: loss=0.288
batch 40: loss=0.280
batch 60: loss=0.255
batch 80: loss=0.294
batch 100: loss=0.276
batch 120: loss=0.262
```

```
batch 140: loss=0.282
batch 160: loss=0.305
batch 180: loss=0.276
batch 200: loss=0.309
batch 220: loss=0.308
batch 240: loss=0.272
batch 260: loss=0.326
batch 280: loss=0.291
batch 300: loss=0.301
batch 320: loss=0.294
batch 340: loss=0.265
batch 360: loss=0.261
batch 380: loss=0.301
batch 400: loss=0.297
batch 420: loss=0.283
batch 440: loss=0.332
batch 460: loss=0.258
batch 480: loss=0.319
batch 500: loss=0.279
batch 520: loss=0.273
batch 540: loss=0.286
batch 560: loss=0.267
batch 580: loss=0.277
batch 600: loss=0.278
batch 620: loss=0.259
batch 640: loss=0.270
batch 660: loss=0.321
batch 680: loss=0.282
batch 700: loss=0.296
batch 720: loss=0.283
batch 740: loss=0.290
batch 760: loss=0.282
batch 780: loss=0.289
Testing on validation set
acc=0.885
```

Training epoch 8

```
batch 20: loss=0.276
batch 40: loss=0.282
batch 60: loss=0.271
batch 80: loss=0.261
batch 100: loss=0.247
batch 120: loss=0.295
batch 140: loss=0.275
batch 160: loss=0.270
batch 180: loss=0.285
batch 200: loss=0.263
batch 220: loss=0.313
batch 240: loss=0.275
```

```
batch 260: loss=0.274
batch 280: loss=0.273
batch 300: loss=0.314
batch 320: loss=0.297
batch 340: loss=0.276
batch 360: loss=0.267
batch 380: loss=0.258
batch 400: loss=0.271
batch 420: loss=0.304
batch 440: loss=0.276
batch 460: loss=0.287
batch 480: loss=0.301
batch 500: loss=0.271
batch 520: loss=0.266
batch 540: loss=0.283
batch 560: loss=0.307
batch 580: loss=0.280
batch 600: loss=0.261
batch 620: loss=0.248
batch 640: loss=0.266
batch 660: loss=0.267
batch 680: loss=0.294
batch 700: loss=0.317
batch 720: loss=0.317
batch 740: loss=0.268
batch 760: loss=0.283
batch 780: loss=0.279
Testing on validation set
acc=0.887
```

Training epoch 9

```
batch 20: loss=0.271
batch 40: loss=0.274
batch 60: loss=0.270
batch 80: loss=0.261
batch 100: loss=0.285
batch 120: loss=0.258
batch 140: loss=0.288
batch 160: loss=0.279
batch 180: loss=0.269
batch 200: loss=0.272
batch 220: loss=0.269
batch 240: loss=0.292
batch 260: loss=0.244
batch 280: loss=0.255
batch 300: loss=0.295
batch 320: loss=0.275
batch 340: loss=0.293
batch 360: loss=0.264
```

```

batch 380: loss=0.268
batch 400: loss=0.282
batch 420: loss=0.270
batch 440: loss=0.269
batch 460: loss=0.284
batch 480: loss=0.248
batch 500: loss=0.255
batch 520: loss=0.308
batch 540: loss=0.249
batch 560: loss=0.256
batch 580: loss=0.327
batch 600: loss=0.266
batch 620: loss=0.255
batch 640: loss=0.295
batch 660: loss=0.272
batch 680: loss=0.274
batch 700: loss=0.280
batch 720: loss=0.286
batch 740: loss=0.285
batch 760: loss=0.292
batch 780: loss=0.288
Testing on validation set
acc=0.883

```

Increasing the number of epochs increased accuracy by over 1%.  
Part 3: Lower learning rate (increased number of epochs)

```
In [0]: args.lr = 0.001
```

```

model3 = Net()
model3.to(device)

criterion = nn.CrossEntropyLoss()
optimizer = torch.optim.SGD(
    model3.parameters(),
    lr=args.lr,
    momentum=args.momentum
)

for e in range(args.num_epochs):
    print('Training epoch {}'.format(e))
    train(args, model3, criterion, train_loader, optimizer, device)
    print('Testing on validation set')
    test(args, model3, val_loader, device)

```

```

Training epoch 0
batch 20: loss=2.296

```

```
batch 40: loss=2.268
batch 60: loss=2.220
batch 80: loss=2.173
batch 100: loss=2.078
batch 120: loss=1.946
batch 140: loss=1.767
batch 160: loss=1.592
batch 180: loss=1.390
batch 200: loss=1.207
batch 220: loss=1.086
batch 240: loss=1.018
batch 260: loss=0.966
batch 280: loss=0.947
batch 300: loss=0.884
batch 320: loss=0.873
batch 340: loss=0.865
batch 360: loss=0.787
batch 380: loss=0.813
batch 400: loss=0.798
batch 420: loss=0.828
batch 440: loss=0.728
batch 460: loss=0.762
batch 480: loss=0.757
batch 500: loss=0.772
batch 520: loss=0.761
batch 540: loss=0.743
batch 560: loss=0.668
batch 580: loss=0.709
batch 600: loss=0.706
batch 620: loss=0.715
batch 640: loss=0.670
batch 660: loss=0.713
batch 680: loss=0.719
batch 700: loss=0.668
batch 720: loss=0.696
batch 740: loss=0.646
batch 760: loss=0.681
batch 780: loss=0.660
Testing on validation set
acc=0.759
Training epoch 1
batch 20: loss=0.629
batch 40: loss=0.653
batch 60: loss=0.669
batch 80: loss=0.641
batch 100: loss=0.632
batch 120: loss=0.666
batch 140: loss=0.662
```

```
batch 160: loss=0.687
batch 180: loss=0.640
batch 200: loss=0.633
batch 220: loss=0.625
batch 240: loss=0.626
batch 260: loss=0.624
batch 280: loss=0.640
batch 300: loss=0.636
batch 320: loss=0.697
batch 340: loss=0.589
batch 360: loss=0.624
batch 380: loss=0.628
batch 400: loss=0.607
batch 420: loss=0.605
batch 440: loss=0.585
batch 460: loss=0.577
batch 480: loss=0.629
batch 500: loss=0.562
batch 520: loss=0.582
batch 540: loss=0.570
batch 560: loss=0.559
batch 580: loss=0.559
batch 600: loss=0.599
batch 620: loss=0.536
batch 640: loss=0.594
batch 660: loss=0.590
batch 680: loss=0.591
batch 700: loss=0.558
batch 720: loss=0.582
batch 740: loss=0.581
batch 760: loss=0.584
batch 780: loss=0.601
Testing on validation set
acc=0.796
```

Training epoch 2

```
batch 20: loss=0.587
batch 40: loss=0.540
batch 60: loss=0.523
batch 80: loss=0.549
batch 100: loss=0.521
batch 120: loss=0.559
batch 140: loss=0.527
batch 160: loss=0.504
batch 180: loss=0.567
batch 200: loss=0.509
batch 220: loss=0.541
batch 240: loss=0.548
batch 260: loss=0.594
```

```
batch 280: loss=0.534
batch 300: loss=0.574
batch 320: loss=0.531
batch 340: loss=0.528
batch 360: loss=0.530
batch 380: loss=0.537
batch 400: loss=0.525
batch 420: loss=0.535
batch 440: loss=0.496
batch 460: loss=0.551
batch 480: loss=0.548
batch 500: loss=0.545
batch 520: loss=0.527
batch 540: loss=0.554
batch 560: loss=0.515
batch 580: loss=0.495
batch 600: loss=0.529
batch 620: loss=0.529
batch 640: loss=0.523
batch 660: loss=0.528
batch 680: loss=0.500
batch 700: loss=0.535
batch 720: loss=0.492
batch 740: loss=0.527
batch 760: loss=0.542
batch 780: loss=0.485
Testing on validation set
acc=0.819
```

Training epoch 3

```
batch 20: loss=0.529
batch 40: loss=0.566
batch 60: loss=0.497
batch 80: loss=0.486
batch 100: loss=0.492
batch 120: loss=0.510
batch 140: loss=0.487
batch 160: loss=0.456
batch 180: loss=0.486
batch 200: loss=0.503
batch 220: loss=0.490
batch 240: loss=0.531
batch 260: loss=0.482
batch 280: loss=0.543
batch 300: loss=0.472
batch 320: loss=0.515
batch 340: loss=0.481
batch 360: loss=0.553
batch 380: loss=0.522
```

```
batch 400: loss=0.472
batch 420: loss=0.504
batch 440: loss=0.470
batch 460: loss=0.520
batch 480: loss=0.486
batch 500: loss=0.483
batch 520: loss=0.488
batch 540: loss=0.502
batch 560: loss=0.506
batch 580: loss=0.430
batch 600: loss=0.467
batch 620: loss=0.477
batch 640: loss=0.477
batch 660: loss=0.491
batch 680: loss=0.452
batch 700: loss=0.491
batch 720: loss=0.471
batch 740: loss=0.500
batch 760: loss=0.416
batch 780: loss=0.478
Testing on validation set
acc=0.827
```

Training epoch 4

```
batch 20: loss=0.450
batch 40: loss=0.479
batch 60: loss=0.506
batch 80: loss=0.491
batch 100: loss=0.465
batch 120: loss=0.470
batch 140: loss=0.496
batch 160: loss=0.437
batch 180: loss=0.498
batch 200: loss=0.446
batch 220: loss=0.455
batch 240: loss=0.478
batch 260: loss=0.512
batch 280: loss=0.495
batch 300: loss=0.455
batch 320: loss=0.470
batch 340: loss=0.472
batch 360: loss=0.469
batch 380: loss=0.476
batch 400: loss=0.462
batch 420: loss=0.457
batch 440: loss=0.447
batch 460: loss=0.479
batch 480: loss=0.413
batch 500: loss=0.420
```



```
batch 520: loss=0.460
batch 540: loss=0.477
batch 560: loss=0.477
batch 580: loss=0.464
batch 600: loss=0.458
batch 620: loss=0.444
batch 640: loss=0.441
batch 660: loss=0.451
batch 680: loss=0.481
batch 700: loss=0.483
batch 720: loss=0.429
batch 740: loss=0.456
batch 760: loss=0.487
batch 780: loss=0.463
Testing on validation set
acc=0.833
```

Training epoch 5

```
batch 20: loss=0.417
batch 40: loss=0.456
batch 60: loss=0.472
batch 80: loss=0.445
batch 100: loss=0.465
batch 120: loss=0.433
batch 140: loss=0.469
batch 160: loss=0.479
batch 180: loss=0.470
batch 200: loss=0.430
batch 220: loss=0.452
batch 240: loss=0.481
batch 260: loss=0.437
batch 280: loss=0.437
batch 300: loss=0.444
batch 320: loss=0.442
batch 340: loss=0.448
batch 360: loss=0.455
batch 380: loss=0.475
batch 400: loss=0.438
batch 420: loss=0.471
batch 440: loss=0.470
batch 460: loss=0.488
batch 480: loss=0.429
batch 500: loss=0.480
batch 520: loss=0.415
batch 540: loss=0.439
batch 560: loss=0.433
batch 580: loss=0.406
batch 600: loss=0.441
batch 620: loss=0.445
```

```
batch 640: loss=0.452
batch 660: loss=0.448
batch 680: loss=0.413
batch 700: loss=0.427
batch 720: loss=0.429
batch 740: loss=0.426
batch 760: loss=0.419
batch 780: loss=0.424
Testing on validation set
acc=0.831
```

Training epoch 6

```
batch 20: loss=0.444
batch 40: loss=0.443
batch 60: loss=0.500
batch 80: loss=0.443
batch 100: loss=0.423
batch 120: loss=0.435
batch 140: loss=0.412
batch 160: loss=0.414
batch 180: loss=0.440
batch 200: loss=0.428
batch 220: loss=0.433
batch 240: loss=0.406
batch 260: loss=0.459
batch 280: loss=0.401
batch 300: loss=0.443
batch 320: loss=0.441
batch 340: loss=0.419
batch 360: loss=0.443
batch 380: loss=0.435
batch 400: loss=0.455
batch 420: loss=0.388
batch 440: loss=0.450
batch 460: loss=0.410
batch 480: loss=0.437
batch 500: loss=0.472
batch 520: loss=0.491
batch 540: loss=0.417
batch 560: loss=0.433
batch 580: loss=0.419
batch 600: loss=0.413
batch 620: loss=0.451
batch 640: loss=0.407
batch 660: loss=0.400
batch 680: loss=0.399
batch 700: loss=0.439
batch 720: loss=0.414
batch 740: loss=0.406
```

batch 760: loss=0.393  
batch 780: loss=0.410  
Testing on validation set  
acc=0.847

Training epoch 7

batch 20: loss=0.406  
batch 40: loss=0.403  
batch 60: loss=0.423  
batch 80: loss=0.425  
batch 100: loss=0.401  
batch 120: loss=0.422  
batch 140: loss=0.389  
batch 160: loss=0.457  
batch 180: loss=0.435  
batch 200: loss=0.379  
batch 220: loss=0.399  
batch 240: loss=0.425  
batch 260: loss=0.434  
batch 280: loss=0.405  
batch 300: loss=0.417  
batch 320: loss=0.410  
batch 340: loss=0.413  
batch 360: loss=0.447  
batch 380: loss=0.422  
batch 400: loss=0.392  
batch 420: loss=0.421  
batch 440: loss=0.413  
batch 460: loss=0.422  
batch 480: loss=0.357  
batch 500: loss=0.381  
batch 520: loss=0.428  
batch 540: loss=0.438  
batch 560: loss=0.423  
batch 580: loss=0.439  
batch 600: loss=0.430  
batch 620: loss=0.390  
batch 640: loss=0.394  
batch 660: loss=0.375  
batch 680: loss=0.435  
batch 700: loss=0.427  
batch 720: loss=0.429  
batch 740: loss=0.413  
batch 760: loss=0.425  
batch 780: loss=0.389

Testing on validation set  
acc=0.852

Training epoch 8

batch 20: loss=0.424

```
batch 40: loss=0.438
batch 60: loss=0.423
batch 80: loss=0.420
batch 100: loss=0.385
batch 120: loss=0.376
batch 140: loss=0.367
batch 160: loss=0.380
batch 180: loss=0.413
batch 200: loss=0.403
batch 220: loss=0.358
batch 240: loss=0.392
batch 260: loss=0.445
batch 280: loss=0.424
batch 300: loss=0.400
batch 320: loss=0.401
batch 340: loss=0.386
batch 360: loss=0.421
batch 380: loss=0.430
batch 400: loss=0.389
batch 420: loss=0.427
batch 440: loss=0.444
batch 460: loss=0.393
batch 480: loss=0.396
batch 500: loss=0.368
batch 520: loss=0.400
batch 540: loss=0.372
batch 560: loss=0.403
batch 580: loss=0.420
batch 600: loss=0.401
batch 620: loss=0.395
batch 640: loss=0.364
batch 660: loss=0.423
batch 680: loss=0.449
batch 700: loss=0.378
batch 720: loss=0.396
batch 740: loss=0.379
batch 760: loss=0.364
batch 780: loss=0.397
Testing on validation set
acc=0.848
Training epoch 9
batch 20: loss=0.408
batch 40: loss=0.393
batch 60: loss=0.371
batch 80: loss=0.398
batch 100: loss=0.377
batch 120: loss=0.396
batch 140: loss=0.423
```

```
batch 160: loss=0.389
batch 180: loss=0.376
batch 200: loss=0.430
batch 220: loss=0.398
batch 240: loss=0.370
batch 260: loss=0.371
batch 280: loss=0.404
batch 300: loss=0.448
batch 320: loss=0.405
batch 340: loss=0.388
batch 360: loss=0.381
batch 380: loss=0.365
batch 400: loss=0.406
batch 420: loss=0.406
batch 440: loss=0.412
batch 460: loss=0.370
batch 480: loss=0.351
batch 500: loss=0.386
batch 520: loss=0.405
batch 540: loss=0.419
batch 560: loss=0.382
batch 580: loss=0.343
batch 600: loss=0.384
batch 620: loss=0.386
batch 640: loss=0.377
batch 660: loss=0.409
batch 680: loss=0.378
batch 700: loss=0.380
batch 720: loss=0.380
batch 740: loss=0.398
batch 760: loss=0.332
batch 780: loss=0.398
Testing on validation set
acc=0.859
```

The lower learning rate slightly decreases the validation accuracy.  
Part 4: Adding more channels from layer 1 to 2

```
In [0]: class Net2(nn.Module):
        def __init__(self):
            super(Net2, self).__init__()
            self.conv1 = nn.Conv2d(1, 32, 5)
            self.conv2 = nn.Conv2d(32, 16, 5)
            self.fc3 = nn.Linear(256, 10)

            self._initialize_weights()
```

```

def _initialize_weights(self):
    pass

def forward(self, x):
    x = F.relu(self.conv1(x))
    x = F.max_pool2d(x, 2, stride=2)
    x = F.relu(self.conv2(x))
    x = F.max_pool2d(x, 2, stride=2)
    x = torch.flatten(x, start_dim=1)
    x = self.fc3(x)
    return x

args.lr = 0.01

model4 = Net2()
model4.to(device)

criterion = nn.CrossEntropyLoss()
optimizer = torch.optim.SGD(
    model4.parameters(),
    lr=args.lr,
    momentum=args.momentum
)

for e in range(args.num_epochs):
    print('Training epoch {}'.format(e))
    train(args, model4, criterion, train_loader, optimizer, device)
    print('Testing on validation set')
    test(args, model4, val_loader, device)

```

Training epoch 0

```

batch 20: loss=2.113
batch 40: loss=1.215
batch 60: loss=0.864
batch 80: loss=0.770
batch 100: loss=0.762
batch 120: loss=0.706
batch 140: loss=0.684
batch 160: loss=0.641
batch 180: loss=0.572
batch 200: loss=0.573
batch 220: loss=0.580
batch 240: loss=0.548
batch 260: loss=0.473
batch 280: loss=0.506
batch 300: loss=0.533
batch 320: loss=0.503
batch 340: loss=0.508

```

batch 360: loss=0.550  
batch 380: loss=0.474  
batch 400: loss=0.440  
batch 420: loss=0.460  
batch 440: loss=0.458  
batch 460: loss=0.423  
batch 480: loss=0.478  
batch 500: loss=0.481  
batch 520: loss=0.438  
batch 540: loss=0.444  
batch 560: loss=0.397  
batch 580: loss=0.409  
batch 600: loss=0.449  
batch 620: loss=0.436  
batch 640: loss=0.425  
batch 660: loss=0.388  
batch 680: loss=0.447  
batch 700: loss=0.371  
batch 720: loss=0.458  
batch 740: loss=0.416  
batch 760: loss=0.391  
batch 780: loss=0.399

Testing on validation set  
acc=0.845

Training epoch 1

batch 20: loss=0.394  
batch 40: loss=0.356  
batch 60: loss=0.387  
batch 80: loss=0.433  
batch 100: loss=0.402  
batch 120: loss=0.331  
batch 140: loss=0.347  
batch 160: loss=0.346  
batch 180: loss=0.431  
batch 200: loss=0.354  
batch 220: loss=0.374  
batch 240: loss=0.353  
batch 260: loss=0.329  
batch 280: loss=0.408  
batch 300: loss=0.422  
batch 320: loss=0.363  
batch 340: loss=0.373  
batch 360: loss=0.373  
batch 380: loss=0.378  
batch 400: loss=0.383  
batch 420: loss=0.387  
batch 440: loss=0.360  
batch 460: loss=0.367

batch 480: loss=0.374  
batch 500: loss=0.339  
batch 520: loss=0.359  
batch 540: loss=0.366  
batch 560: loss=0.370  
batch 580: loss=0.364  
batch 600: loss=0.392  
batch 620: loss=0.341  
batch 640: loss=0.356  
batch 660: loss=0.363  
batch 680: loss=0.350  
batch 700: loss=0.357  
batch 720: loss=0.375  
batch 740: loss=0.347  
batch 760: loss=0.329  
batch 780: loss=0.338

Testing on validation set  
acc=0.870

Training epoch 2

batch 20: loss=0.373  
batch 40: loss=0.341  
batch 60: loss=0.325  
batch 80: loss=0.305  
batch 100: loss=0.323  
batch 120: loss=0.347  
batch 140: loss=0.357  
batch 160: loss=0.306  
batch 180: loss=0.349  
batch 200: loss=0.326  
batch 220: loss=0.315  
batch 240: loss=0.385  
batch 260: loss=0.324  
batch 280: loss=0.309  
batch 300: loss=0.312  
batch 320: loss=0.295  
batch 340: loss=0.316  
batch 360: loss=0.352  
batch 380: loss=0.371  
batch 400: loss=0.331  
batch 420: loss=0.336  
batch 440: loss=0.326  
batch 460: loss=0.338  
batch 480: loss=0.321  
batch 500: loss=0.335  
batch 520: loss=0.361  
batch 540: loss=0.314  
batch 560: loss=0.319  
batch 580: loss=0.331



batch 600: loss=0.325  
batch 620: loss=0.342  
batch 640: loss=0.326  
batch 660: loss=0.282  
batch 680: loss=0.298  
batch 700: loss=0.304  
batch 720: loss=0.323  
batch 740: loss=0.316  
batch 760: loss=0.321  
batch 780: loss=0.363

Testing on validation set  
acc=0.880

Training epoch 3

batch 20: loss=0.337  
batch 40: loss=0.320  
batch 60: loss=0.319  
batch 80: loss=0.254  
batch 100: loss=0.314  
batch 120: loss=0.311  
batch 140: loss=0.303  
batch 160: loss=0.284  
batch 180: loss=0.291  
batch 200: loss=0.270  
batch 220: loss=0.317  
batch 240: loss=0.295  
batch 260: loss=0.300  
batch 280: loss=0.320  
batch 300: loss=0.351  
batch 320: loss=0.284  
batch 340: loss=0.251  
batch 360: loss=0.307  
batch 380: loss=0.314  
batch 400: loss=0.328  
batch 420: loss=0.302  
batch 440: loss=0.328  
batch 460: loss=0.311  
batch 480: loss=0.303  
batch 500: loss=0.279  
batch 520: loss=0.335  
batch 540: loss=0.270  
batch 560: loss=0.336  
batch 580: loss=0.298  
batch 600: loss=0.316  
batch 620: loss=0.299  
batch 640: loss=0.295  
batch 660: loss=0.325  
batch 680: loss=0.313  
batch 700: loss=0.303

```
batch 720: loss=0.324
batch 740: loss=0.325
batch 760: loss=0.293
batch 780: loss=0.283
Testing on validation set
acc=0.890
Training epoch 4
batch 20: loss=0.289
batch 40: loss=0.241
batch 60: loss=0.327
batch 80: loss=0.286
batch 100: loss=0.280
batch 120: loss=0.293
batch 140: loss=0.260
batch 160: loss=0.295
batch 180: loss=0.328
batch 200: loss=0.263
batch 220: loss=0.288
batch 240: loss=0.266
batch 260: loss=0.281
batch 280: loss=0.284
batch 300: loss=0.288
batch 320: loss=0.287
batch 340: loss=0.285
batch 360: loss=0.245
batch 380: loss=0.318
batch 400: loss=0.296
batch 420: loss=0.313
batch 440: loss=0.330
batch 460: loss=0.280
batch 480: loss=0.327
batch 500: loss=0.291
batch 520: loss=0.283
batch 540: loss=0.329
batch 560: loss=0.305
batch 580: loss=0.266
batch 600: loss=0.275
batch 620: loss=0.279
batch 640: loss=0.284
batch 660: loss=0.307
batch 680: loss=0.260
batch 700: loss=0.272
batch 720: loss=0.281
batch 740: loss=0.280
batch 760: loss=0.259
batch 780: loss=0.318
Testing on validation set
acc=0.882
```

Training epoch 5

batch 20: loss=0.253  
batch 40: loss=0.298  
batch 60: loss=0.269  
batch 80: loss=0.281  
batch 100: loss=0.274  
batch 120: loss=0.279  
batch 140: loss=0.269  
batch 160: loss=0.264  
batch 180: loss=0.271  
batch 200: loss=0.300  
batch 220: loss=0.274  
batch 240: loss=0.262  
batch 260: loss=0.286  
batch 280: loss=0.288  
batch 300: loss=0.253  
batch 320: loss=0.234  
batch 340: loss=0.265  
batch 360: loss=0.272  
batch 380: loss=0.289  
batch 400: loss=0.281  
batch 420: loss=0.285  
batch 440: loss=0.307  
batch 460: loss=0.305  
batch 480: loss=0.255  
batch 500: loss=0.262  
batch 520: loss=0.288  
batch 540: loss=0.250  
batch 560: loss=0.296  
batch 580: loss=0.235  
batch 600: loss=0.290  
batch 620: loss=0.255  
batch 640: loss=0.270  
batch 660: loss=0.277  
batch 680: loss=0.298  
batch 700: loss=0.289  
batch 720: loss=0.273  
batch 740: loss=0.277  
batch 760: loss=0.264  
batch 780: loss=0.276

Testing on validation set

acc=0.892

Training epoch 6

batch 20: loss=0.294  
batch 40: loss=0.290  
batch 60: loss=0.246  
batch 80: loss=0.270  
batch 100: loss=0.260

batch 120: loss=0.254  
batch 140: loss=0.275  
batch 160: loss=0.246  
batch 180: loss=0.237  
batch 200: loss=0.241  
batch 220: loss=0.268  
batch 240: loss=0.269  
batch 260: loss=0.262  
batch 280: loss=0.238  
batch 300: loss=0.242  
batch 320: loss=0.268  
batch 340: loss=0.249  
batch 360: loss=0.216  
batch 380: loss=0.297  
batch 400: loss=0.251  
batch 420: loss=0.299  
batch 440: loss=0.289  
batch 460: loss=0.262  
batch 480: loss=0.238  
batch 500: loss=0.260  
batch 520: loss=0.222  
batch 540: loss=0.263  
batch 560: loss=0.287  
batch 580: loss=0.270  
batch 600: loss=0.268  
batch 620: loss=0.242  
batch 640: loss=0.256  
batch 660: loss=0.276  
batch 680: loss=0.262  
batch 700: loss=0.231  
batch 720: loss=0.236  
batch 740: loss=0.303  
batch 760: loss=0.262  
batch 780: loss=0.283

Testing on validation set

acc=0.889

Training epoch 7

batch 20: loss=0.270  
batch 40: loss=0.251  
batch 60: loss=0.249  
batch 80: loss=0.243  
batch 100: loss=0.235  
batch 120: loss=0.257  
batch 140: loss=0.268  
batch 160: loss=0.271  
batch 180: loss=0.250  
batch 200: loss=0.253  
batch 220: loss=0.206

```
batch 240: loss=0.267
batch 260: loss=0.277
batch 280: loss=0.231
batch 300: loss=0.255
batch 320: loss=0.271
batch 340: loss=0.263
batch 360: loss=0.240
batch 380: loss=0.258
batch 400: loss=0.236
batch 420: loss=0.281
batch 440: loss=0.278
batch 460: loss=0.235
batch 480: loss=0.240
batch 500: loss=0.196
batch 520: loss=0.305
batch 540: loss=0.263
batch 560: loss=0.221
batch 580: loss=0.233
batch 600: loss=0.242
batch 620: loss=0.262
batch 640: loss=0.280
batch 660: loss=0.266
batch 680: loss=0.291
batch 700: loss=0.251
batch 720: loss=0.247
batch 740: loss=0.240
batch 760: loss=0.272
batch 780: loss=0.220
Testing on validation set
acc=0.899
```

Training epoch 8

```
batch 20: loss=0.273
batch 40: loss=0.261
batch 60: loss=0.218
batch 80: loss=0.259
batch 100: loss=0.245
batch 120: loss=0.256
batch 140: loss=0.193
batch 160: loss=0.257
batch 180: loss=0.216
batch 200: loss=0.287
batch 220: loss=0.256
batch 240: loss=0.207
batch 260: loss=0.236
batch 280: loss=0.235
batch 300: loss=0.220
batch 320: loss=0.251
batch 340: loss=0.231
```

```
batch 360: loss=0.216
batch 380: loss=0.223
batch 400: loss=0.259
batch 420: loss=0.255
batch 440: loss=0.252
batch 460: loss=0.237
batch 480: loss=0.242
batch 500: loss=0.270
batch 520: loss=0.223
batch 540: loss=0.236
batch 560: loss=0.210
batch 580: loss=0.259
batch 600: loss=0.253
batch 620: loss=0.253
batch 640: loss=0.250
batch 660: loss=0.248
batch 680: loss=0.245
batch 700: loss=0.230
batch 720: loss=0.239
batch 740: loss=0.301
batch 760: loss=0.257
batch 780: loss=0.256
```

Testing on validation set  
acc=0.898

Training epoch 9

```
batch 20: loss=0.229
batch 40: loss=0.252
batch 60: loss=0.226
batch 80: loss=0.244
batch 100: loss=0.239
batch 120: loss=0.230
batch 140: loss=0.229
batch 160: loss=0.219
batch 180: loss=0.257
batch 200: loss=0.248
batch 220: loss=0.230
batch 240: loss=0.224
batch 260: loss=0.193
batch 280: loss=0.231
batch 300: loss=0.236
batch 320: loss=0.232
batch 340: loss=0.244
batch 360: loss=0.250
batch 380: loss=0.207
batch 400: loss=0.234
batch 420: loss=0.223
batch 440: loss=0.261
batch 460: loss=0.167
```

```

batch 480: loss=0.233
batch 500: loss=0.240
batch 520: loss=0.230
batch 540: loss=0.256
batch 560: loss=0.226
batch 580: loss=0.269
batch 600: loss=0.228
batch 620: loss=0.247
batch 640: loss=0.247
batch 660: loss=0.264
batch 680: loss=0.243
batch 700: loss=0.238
batch 720: loss=0.245
batch 740: loss=0.259
batch 760: loss=0.248
batch 780: loss=0.208
Testing on validation set
acc=0.901

```

Increasing the number of channels for the convolutional layers increases the accuracy on the validation set by a good amount.

Part 5: Dropout layer after the first convolutional layer for regularization

```

In [0]: class Net3(nn.Module):
        def __init__(self):
            super(Net3, self).__init__()
            self.conv1 = nn.Conv2d(1, 6, 5)
            self.dropout1 = nn.Dropout(p=0.3)
            self.conv2 = nn.Conv2d(6, 16, 5)
            self.fc3 = nn.Linear(256, 10)

            self._initialize_weights()

        def _initialize_weights(self):
            pass

        def forward(self, x):
            x = F.relu(self.conv1(x))
            if self.training:
                x = self.dropout1(x)
            x = F.max_pool2d(x, 2, stride=2)
            x = F.relu(self.conv2(x))
            x = F.max_pool2d(x, 2, stride=2)
            x = torch.flatten(x, start_dim=1)
            x = self.fc3(x)
            return x

```

```

model5 = Net3()
model5.to(device)

criterion = nn.CrossEntropyLoss()
optimizer = torch.optim.SGD(
    model5.parameters(),
    lr=args.lr,
    momentum=args.momentum
)

for e in range(args.num_epochs):
    print('Training epoch {}'.format(e))
    train(args, model5, criterion, train_loader, optimizer, device)
    print('Testing on validation set')
    test(args, model5, val_loader, device)

```

Training epoch 0

```

batch 20: loss=2.127
batch 40: loss=1.233
batch 60: loss=0.976
batch 80: loss=0.865
batch 100: loss=0.712
batch 120: loss=0.643
batch 140: loss=0.619
batch 160: loss=0.682
batch 180: loss=0.621
batch 200: loss=0.607
batch 220: loss=0.584
batch 240: loss=0.616
batch 260: loss=0.586
batch 280: loss=0.589
batch 300: loss=0.515
batch 320: loss=0.546
batch 340: loss=0.527
batch 360: loss=0.550
batch 380: loss=0.517
batch 400: loss=0.493
batch 420: loss=0.517
batch 440: loss=0.508
batch 460: loss=0.466
batch 480: loss=0.468
batch 500: loss=0.492
batch 520: loss=0.465
batch 540: loss=0.440
batch 560: loss=0.463
batch 580: loss=0.446

```



batch 600: loss=0.448  
batch 620: loss=0.466  
batch 640: loss=0.415  
batch 660: loss=0.473  
batch 680: loss=0.426  
batch 700: loss=0.454  
batch 720: loss=0.458  
batch 740: loss=0.474  
batch 760: loss=0.433  
batch 780: loss=0.430

Testing on validation set  
acc=0.842

Training epoch 1

batch 20: loss=0.407  
batch 40: loss=0.446  
batch 60: loss=0.476  
batch 80: loss=0.437  
batch 100: loss=0.430  
batch 120: loss=0.409  
batch 140: loss=0.469  
batch 160: loss=0.419  
batch 180: loss=0.418  
batch 200: loss=0.350  
batch 220: loss=0.357  
batch 240: loss=0.422  
batch 260: loss=0.476  
batch 280: loss=0.443  
batch 300: loss=0.419  
batch 320: loss=0.400  
batch 340: loss=0.388  
batch 360: loss=0.402  
batch 380: loss=0.388  
batch 400: loss=0.382  
batch 420: loss=0.358  
batch 440: loss=0.419  
batch 460: loss=0.376  
batch 480: loss=0.383  
batch 500: loss=0.395  
batch 520: loss=0.393  
batch 540: loss=0.390  
batch 560: loss=0.387  
batch 580: loss=0.360  
batch 600: loss=0.416  
batch 620: loss=0.398  
batch 640: loss=0.368  
batch 660: loss=0.344  
batch 680: loss=0.388  
batch 700: loss=0.376

```
batch 720: loss=0.371
batch 740: loss=0.380
batch 760: loss=0.394
batch 780: loss=0.409
Testing on validation set
acc=0.870
Training epoch 2
batch 20: loss=0.360
batch 40: loss=0.377
batch 60: loss=0.365
batch 80: loss=0.351
batch 100: loss=0.373
batch 120: loss=0.383
batch 140: loss=0.346
batch 160: loss=0.367
batch 180: loss=0.348
batch 200: loss=0.402
batch 220: loss=0.316
batch 240: loss=0.330
batch 260: loss=0.386
batch 280: loss=0.362
batch 300: loss=0.320
batch 320: loss=0.375
batch 340: loss=0.377
batch 360: loss=0.398
batch 380: loss=0.351
batch 400: loss=0.382
batch 420: loss=0.409
batch 440: loss=0.332
batch 460: loss=0.328
batch 480: loss=0.392
batch 500: loss=0.348
batch 520: loss=0.338
batch 540: loss=0.335
batch 560: loss=0.393
batch 580: loss=0.360
batch 600: loss=0.332
batch 620: loss=0.367
batch 640: loss=0.404
batch 660: loss=0.378
batch 680: loss=0.347
batch 700: loss=0.353
batch 720: loss=0.367
batch 740: loss=0.420
batch 760: loss=0.383
batch 780: loss=0.346
Testing on validation set
acc=0.875
```

Training epoch 3

batch 20: loss=0.308  
batch 40: loss=0.319  
batch 60: loss=0.339  
batch 80: loss=0.352  
batch 100: loss=0.310  
batch 120: loss=0.378  
batch 140: loss=0.340  
batch 160: loss=0.356  
batch 180: loss=0.317  
batch 200: loss=0.335  
batch 220: loss=0.394  
batch 240: loss=0.360  
batch 260: loss=0.335  
batch 280: loss=0.350  
batch 300: loss=0.324  
batch 320: loss=0.321  
batch 340: loss=0.391  
batch 360: loss=0.420  
batch 380: loss=0.338  
batch 400: loss=0.357  
batch 420: loss=0.338  
batch 440: loss=0.345  
batch 460: loss=0.334  
batch 480: loss=0.330  
batch 500: loss=0.356  
batch 520: loss=0.361  
batch 540: loss=0.366  
batch 560: loss=0.311  
batch 580: loss=0.332  
batch 600: loss=0.313  
batch 620: loss=0.363  
batch 640: loss=0.345  
batch 660: loss=0.362  
batch 680: loss=0.338  
batch 700: loss=0.362  
batch 720: loss=0.348  
batch 740: loss=0.351  
batch 760: loss=0.367  
batch 780: loss=0.341

Testing on validation set

acc=0.872

Training epoch 4

batch 20: loss=0.330  
batch 40: loss=0.313  
batch 60: loss=0.378  
batch 80: loss=0.362  
batch 100: loss=0.325

batch 120: loss=0.349  
batch 140: loss=0.323  
batch 160: loss=0.316  
batch 180: loss=0.330  
batch 200: loss=0.300  
batch 220: loss=0.343  
batch 240: loss=0.368  
batch 260: loss=0.339  
batch 280: loss=0.304  
batch 300: loss=0.353  
batch 320: loss=0.358  
batch 340: loss=0.327  
batch 360: loss=0.326  
batch 380: loss=0.368  
batch 400: loss=0.330  
batch 420: loss=0.298  
batch 440: loss=0.324  
batch 460: loss=0.350  
batch 480: loss=0.328  
batch 500: loss=0.328  
batch 520: loss=0.325  
batch 540: loss=0.346  
batch 560: loss=0.340  
batch 580: loss=0.280  
batch 600: loss=0.322  
batch 620: loss=0.339  
batch 640: loss=0.361  
batch 660: loss=0.339  
batch 680: loss=0.310  
batch 700: loss=0.329  
batch 720: loss=0.326  
batch 740: loss=0.334  
batch 760: loss=0.341  
batch 780: loss=0.357

Testing on validation set

acc=0.868

Training epoch 5

batch 20: loss=0.311  
batch 40: loss=0.332  
batch 60: loss=0.314  
batch 80: loss=0.314  
batch 100: loss=0.336  
batch 120: loss=0.327  
batch 140: loss=0.314  
batch 160: loss=0.319  
batch 180: loss=0.338  
batch 200: loss=0.321  
batch 220: loss=0.366

```
batch 240: loss=0.357
batch 260: loss=0.342
batch 280: loss=0.346
batch 300: loss=0.322
batch 320: loss=0.295
batch 340: loss=0.350
batch 360: loss=0.318
batch 380: loss=0.315
batch 400: loss=0.311
batch 420: loss=0.342
batch 440: loss=0.321
batch 460: loss=0.288
batch 480: loss=0.319
batch 500: loss=0.315
batch 520: loss=0.280
batch 540: loss=0.367
batch 560: loss=0.325
batch 580: loss=0.286
batch 600: loss=0.344
batch 620: loss=0.284
batch 640: loss=0.332
batch 660: loss=0.369
batch 680: loss=0.321
batch 700: loss=0.325
batch 720: loss=0.350
batch 740: loss=0.291
batch 760: loss=0.333
batch 780: loss=0.320
Testing on validation set
acc=0.875
```

Training epoch 6

```
batch 20: loss=0.348
batch 40: loss=0.349
batch 60: loss=0.367
batch 80: loss=0.311
batch 100: loss=0.288
batch 120: loss=0.301
batch 140: loss=0.325
batch 160: loss=0.382
batch 180: loss=0.323
batch 200: loss=0.312
batch 220: loss=0.302
batch 240: loss=0.298
batch 260: loss=0.286
batch 280: loss=0.290
batch 300: loss=0.335
batch 320: loss=0.374
batch 340: loss=0.305
```

batch 360: loss=0.299  
batch 380: loss=0.349  
batch 400: loss=0.334  
batch 420: loss=0.322  
batch 440: loss=0.343  
batch 460: loss=0.287  
batch 480: loss=0.343  
batch 500: loss=0.307  
batch 520: loss=0.316  
batch 540: loss=0.281  
batch 560: loss=0.322  
batch 580: loss=0.321  
batch 600: loss=0.293  
batch 620: loss=0.287  
batch 640: loss=0.330  
batch 660: loss=0.309  
batch 680: loss=0.310  
batch 700: loss=0.339  
batch 720: loss=0.358  
batch 740: loss=0.278  
batch 760: loss=0.311  
batch 780: loss=0.328

Testing on validation set  
acc=0.872

Training epoch 7

batch 20: loss=0.317  
batch 40: loss=0.297  
batch 60: loss=0.292  
batch 80: loss=0.290  
batch 100: loss=0.314  
batch 120: loss=0.336  
batch 140: loss=0.303  
batch 160: loss=0.308  
batch 180: loss=0.326  
batch 200: loss=0.269  
batch 220: loss=0.296  
batch 240: loss=0.289  
batch 260: loss=0.280  
batch 280: loss=0.275  
batch 300: loss=0.323  
batch 320: loss=0.274  
batch 340: loss=0.282  
batch 360: loss=0.321  
batch 380: loss=0.336  
batch 400: loss=0.339  
batch 420: loss=0.331  
batch 440: loss=0.325  
batch 460: loss=0.305

```
batch 480: loss=0.321
batch 500: loss=0.320
batch 520: loss=0.326
batch 540: loss=0.302
batch 560: loss=0.322
batch 580: loss=0.323
batch 600: loss=0.318
batch 620: loss=0.304
batch 640: loss=0.325
batch 660: loss=0.353
batch 680: loss=0.283
batch 700: loss=0.296
batch 720: loss=0.354
batch 740: loss=0.300
batch 760: loss=0.315
batch 780: loss=0.284
```

Testing on validation set  
acc=0.871

Training epoch 8

```
batch 20: loss=0.345
batch 40: loss=0.281
batch 60: loss=0.312
batch 80: loss=0.264
batch 100: loss=0.287
batch 120: loss=0.309
batch 140: loss=0.275
batch 160: loss=0.287
batch 180: loss=0.325
batch 200: loss=0.323
batch 220: loss=0.301
batch 240: loss=0.268
batch 260: loss=0.317
batch 280: loss=0.304
batch 300: loss=0.284
batch 320: loss=0.285
batch 340: loss=0.350
batch 360: loss=0.339
batch 380: loss=0.323
batch 400: loss=0.306
batch 420: loss=0.313
batch 440: loss=0.339
batch 460: loss=0.303
batch 480: loss=0.295
batch 500: loss=0.310
batch 520: loss=0.288
batch 540: loss=0.339
batch 560: loss=0.295
batch 580: loss=0.342
```

```
batch 600: loss=0.293
batch 620: loss=0.285
batch 640: loss=0.280
batch 660: loss=0.294
batch 680: loss=0.330
batch 700: loss=0.286
batch 720: loss=0.319
batch 740: loss=0.292
batch 760: loss=0.319
batch 780: loss=0.316
```

```
Testing on validation set
acc=0.880
```

```
Training epoch 9
```

```
batch 20: loss=0.325
batch 40: loss=0.319
batch 60: loss=0.308
batch 80: loss=0.259
batch 100: loss=0.307
batch 120: loss=0.283
batch 140: loss=0.288
batch 160: loss=0.286
batch 180: loss=0.322
batch 200: loss=0.311
batch 220: loss=0.274
batch 240: loss=0.309
batch 260: loss=0.330
batch 280: loss=0.306
batch 300: loss=0.283
batch 320: loss=0.255
batch 340: loss=0.308
batch 360: loss=0.325
batch 380: loss=0.290
batch 400: loss=0.304
batch 420: loss=0.335
batch 440: loss=0.315
batch 460: loss=0.303
batch 480: loss=0.303
batch 500: loss=0.286
batch 520: loss=0.298
batch 540: loss=0.292
batch 560: loss=0.303
batch 580: loss=0.327
batch 600: loss=0.297
batch 620: loss=0.295
batch 640: loss=0.278
batch 660: loss=0.318
batch 680: loss=0.319
batch 700: loss=0.317
```



```
batch 720: loss=0.324
batch 740: loss=0.336
batch 760: loss=0.331
batch 780: loss=0.333
Testing on validation set
acc=0.887
```

The dropout layer increase the accuracy slightly

Question 3 - Best network: Increased number of epochs, dropout layer, and increased number of channels

```
In [15]: class BestNet(nn.Module):
        def __init__(self):
            super(BestNet, self).__init__()
            self.conv1 = nn.Conv2d(1, 32, 5)
            self.dropout1 = nn.Dropout(p=0.3)
            self.conv2 = nn.Conv2d(32, 16, 5)
            self.fc3 = nn.Linear(256, 10)

            self._initialize_weights()

        def _initialize_weights(self):
            pass

        def forward(self, x):
            x = F.relu(self.conv1(x))
            if self.training:
                x = self.dropout1(x)
            x = F.max_pool2d(x, 2, stride=2)
            x = F.relu(self.conv2(x))
            x = F.max_pool2d(x, 2, stride=2)
            x = torch.flatten(x, start_dim=1)
            x = self.fc3(x)
            return x

        best_model = BestNet()
        best_model.to(device)

        criterion = nn.CrossEntropyLoss()
        optimizer = torch.optim.SGD(
            best_model.parameters(),
            lr=args.lr,
            momentum=args.momentum
        )

        for e in range(args.num_epochs):
            print('Training epoch {}'.format(e))
```

```
train(args, best_model, criterion, train_loader, optimizer, device)
print('Testing on validation set')
test(args, best_model, val_loader, device)
```

Training epoch 0

```
batch 20: loss=2.058
batch 40: loss=1.118
batch 60: loss=0.813
batch 80: loss=0.748
batch 100: loss=0.703
batch 120: loss=0.655
batch 140: loss=0.623
batch 160: loss=0.581
batch 180: loss=0.558
batch 200: loss=0.541
batch 220: loss=0.493
batch 240: loss=0.525
batch 260: loss=0.532
batch 280: loss=0.504
batch 300: loss=0.476
batch 320: loss=0.525
batch 340: loss=0.485
batch 360: loss=0.504
batch 380: loss=0.469
batch 400: loss=0.420
batch 420: loss=0.447
batch 440: loss=0.413
batch 460: loss=0.432
batch 480: loss=0.444
batch 500: loss=0.452
batch 520: loss=0.428
batch 540: loss=0.431
batch 560: loss=0.413
batch 580: loss=0.387
batch 600: loss=0.407
batch 620: loss=0.411
batch 640: loss=0.393
batch 660: loss=0.385
batch 680: loss=0.378
batch 700: loss=0.388
batch 720: loss=0.424
batch 740: loss=0.414
batch 760: loss=0.388
batch 780: loss=0.422
```

Testing on validation set  
acc=0.852

Training epoch 1

```
batch 20: loss=0.426
```

```
batch 40: loss=0.352
batch 60: loss=0.406
batch 80: loss=0.362
batch 100: loss=0.391
batch 120: loss=0.417
batch 140: loss=0.407
batch 160: loss=0.372
batch 180: loss=0.373
batch 200: loss=0.366
batch 220: loss=0.374
batch 240: loss=0.354
batch 260: loss=0.389
batch 280: loss=0.425
batch 300: loss=0.347
batch 320: loss=0.363
batch 340: loss=0.314
batch 360: loss=0.375
batch 380: loss=0.343
batch 400: loss=0.316
batch 420: loss=0.317
batch 440: loss=0.384
batch 460: loss=0.338
batch 480: loss=0.364
batch 500: loss=0.339
batch 520: loss=0.377
batch 540: loss=0.371
batch 560: loss=0.359
batch 580: loss=0.344
batch 600: loss=0.335
batch 620: loss=0.341
batch 640: loss=0.348
batch 660: loss=0.369
batch 680: loss=0.343
batch 700: loss=0.342
batch 720: loss=0.287
batch 740: loss=0.329
batch 760: loss=0.336
batch 780: loss=0.351
Testing on validation set
acc=0.879
Training epoch 2
batch 20: loss=0.349
batch 40: loss=0.313
batch 60: loss=0.315
batch 80: loss=0.359
batch 100: loss=0.349
batch 120: loss=0.322
batch 140: loss=0.326
```

```
batch 160: loss=0.315
batch 180: loss=0.376
batch 200: loss=0.318
batch 220: loss=0.352
batch 240: loss=0.299
batch 260: loss=0.346
batch 280: loss=0.324
batch 300: loss=0.327
batch 320: loss=0.352
batch 340: loss=0.301
batch 360: loss=0.330
batch 380: loss=0.366
batch 400: loss=0.259
batch 420: loss=0.353
batch 440: loss=0.321
batch 460: loss=0.309
batch 480: loss=0.335
batch 500: loss=0.359
batch 520: loss=0.309
batch 540: loss=0.356
batch 560: loss=0.323
batch 580: loss=0.322
batch 600: loss=0.330
batch 620: loss=0.328
batch 640: loss=0.317
batch 660: loss=0.310
batch 680: loss=0.317
batch 700: loss=0.301
batch 720: loss=0.319
batch 740: loss=0.324
batch 760: loss=0.303
batch 780: loss=0.326
Testing on validation set
acc=0.880
```

Evaluating the best network on the test set

```
In [16]: test(args, best_model, val_loader, device)

acc=0.880
```

The best network was able to reach 88% accuracy, which probably can be outperformed with further optimization of hyperparameters and architecture