

Deep Learning HW2 Report

Kaho Chan (kc3137)

Yu Wang (yw3025)

Jingxi Xu (jx2324)

October 2017

Part 1

Architecture

An overview of our *ConvNet* architecture is as follows (batch size = 50):

$$input \rightarrow \underbrace{\text{Conv} \rightarrow \text{ReLU} \rightarrow \text{MaxPool}} \rightarrow \underbrace{\text{Conv} \rightarrow \text{ReLU} \rightarrow \text{MaxPool}} \rightarrow \text{Dense} \rightarrow output$$

See table 1 for details of each layer

Convolution layer	Kernel $5 \times 5 \times 32$, padding: same, activation: ReLu, stride: 1
Batch normalization layer	
Max Pooling	Pool size: 2×2 , stride: 2
Convolution layer	Kernel $5 \times 5 \times 64$, padding: same, activation: ReLu, stride: 1
Batch normalization layer	
Max Pooling	Pool size: 2×2 , strides: 2
Dense layer	Units = 1024
Output	Units = 10

Table 1: Details of each layer in the ConvNet

Optimizer

Gradient descent (learning rate = 0.03) .

Train and test accuracy

We split the data into 9(train):1(validation). The following result is performed with Intel(R) core(TM) i7-7700HQ CPU @ 2.80GHz. The training time is about 22 minutes as seen in the notebook.

```
trn_loss: 1.598, train_acc: 0.512, test_loss: 1.173 test_acc: 0.595(0 epoch)
trn_loss: 1.011, train_acc: 0.650, test_loss: 0.985 test_acc: 0.670(1 epoch)
trn_loss: 0.853, train_acc: 0.701, test_loss: 0.947 test_acc: 0.676(2 epoch)
trn_loss: 0.749, train_acc: 0.740, test_loss: 0.888 test_acc: 0.705(3 epoch)
trn_loss: 0.671, train_acc: 0.766, test_loss: 0.919 test_acc: 0.695(4 epoch)
trn_loss: 0.602, train_acc: 0.789, test_loss: 0.889 test_acc: 0.718(5 epoch)
trn_loss: 0.550, train_acc: 0.806, test_loss: 0.902 test_acc: 0.716(6 epoch)
```

The we use the model to predict on the test dataset and store the result in `q1-uni.ipynb`.

Part 2

As the training process is quite long, we break the experiments into different parts, assigned with different group numbers. We only submitted the `ipynb` file containing the training process of the best model. However, all the meaningful experiments are included here in this report to avoid redundancy of showing repeated codes in the `ipynb` file.

Size of training data For all the experiments on our models in Part 2, we split the data (50,000 images) into two parts: 90% (45,000 images) for training and 10% (5,000 images) for validation.

Batch size All models are using a batch size of 50.

The Best Model

Architecture

An overview of the architecture of our best models is as follows.

$input \rightarrow \text{Conv} \rightarrow \text{BatchNorm} \rightarrow \text{MaxPool} \rightarrow \text{Conv} \rightarrow \text{BatchNorm} \rightarrow \text{MaxPool} \rightarrow \text{Dense} \rightarrow \text{BatchNorm} \rightarrow \text{Dropout} \rightarrow output$

The **duration of training** is approximately 2 hours for 30 epochs. Table 2 indicates the architectures of each layer in our best model:

Convolution layer	Kernel $5 \times 5 \times 64$, padding: same, activation: ReLu, stride: 1
Batch normalization layer	
Max Pooling	Pool size: 3×3 , stride: 2
Convolution layer	Kernel $5 \times 5 \times 64$, padding: same, activation: ReLu, stride: 1
Batch normalization layer	
Max Pooling	Pool size: 3×3 , strides: 2
Dense layer	Units = 1024
Batch normalization layer	
Dropout layer	dropout rate = 0.5
Output	Units = 10

Table 2: Details of each layer in the best model

Optimizer

Gradient Descent (learning rate = 0.03)

TensorBoard Screenshots

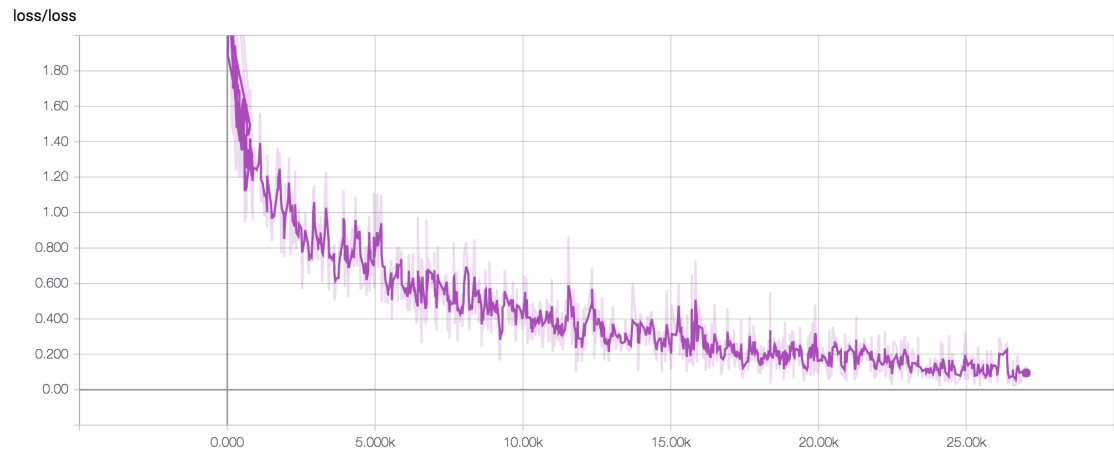


Figure 1: Total Training Loss

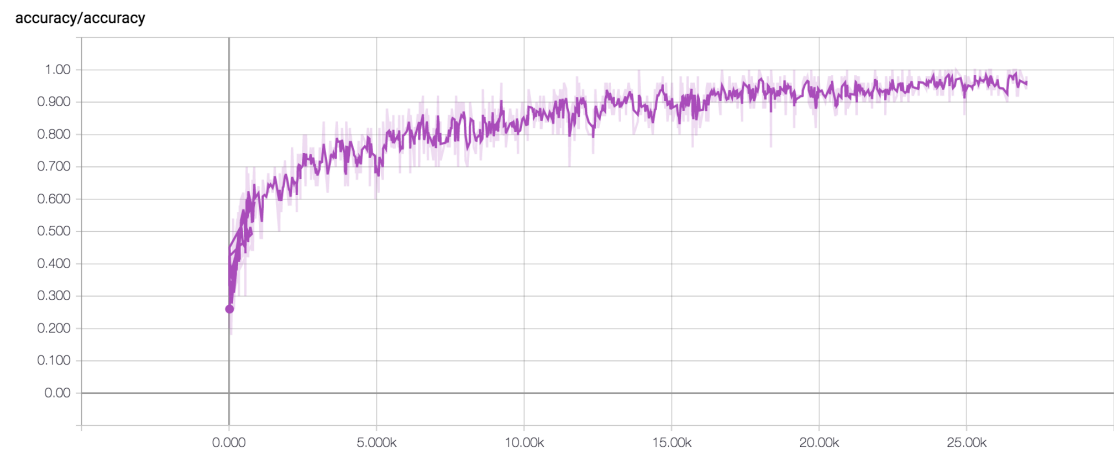


Figure 2: Training Accuracy

Gradient/conv1
train

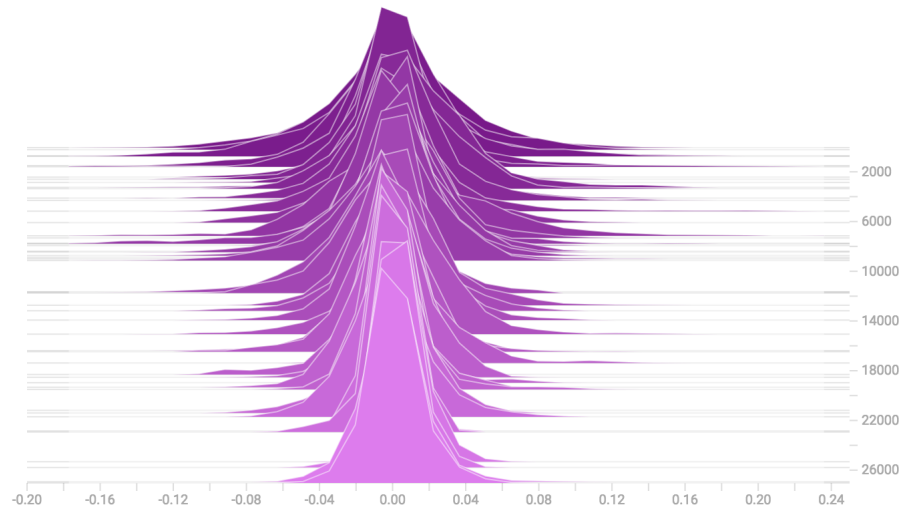


Figure 3: Histogram of weight gradient of the convolution layer #1

Gradient/conv2
train

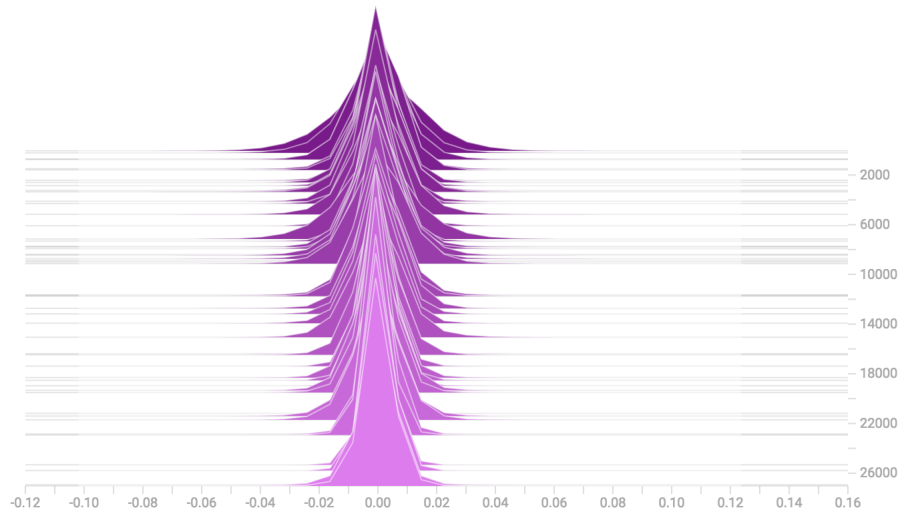


Figure 4: Histogram of weight gradient of the convolution layer #2

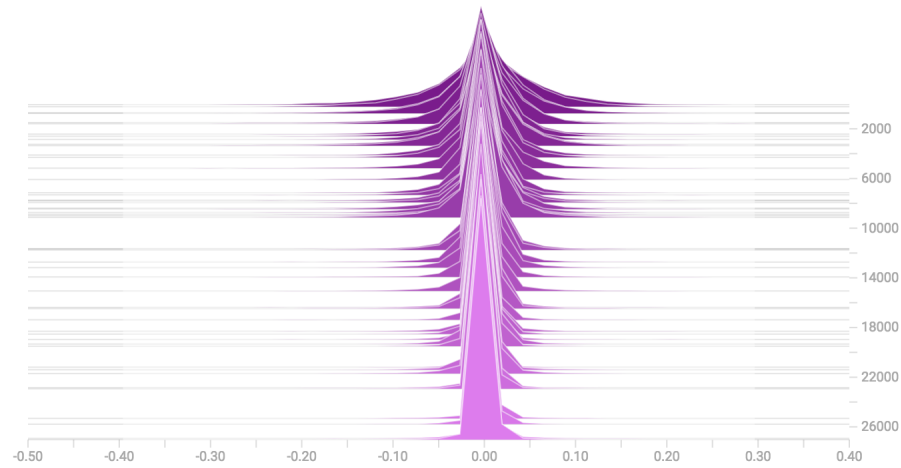


Figure 5: Histogram of weight gradient of the last fully connected layer

Other Models #1

Modification w.r.t. Base Model

- Add one more batch normalization layer in the end before last dense layer, and change pool size from 2×2 to 3×3
- Use *exponential decay* on learning rate

Training and Test Accuracy

trn: 0.988, val: 0.732 (10 epochs)

trn: 1.000, val: 0.747 (20 epochs)

trn: 1.000, val: 0.746 (30 epochs)

Other Models #2

Modification w.r.t. Base Model

- Add one more batch normalization layer in the end before last dense layer, and change pool size from 2×2 to 3×3
- Regularization

Training and Test Accuracy

trn: 0.829, val: 0.764 (10 epochs)

trn: 0.857, val: 0.777 (15 epochs)

Best Model (Documented Above)

Modification w.r.t. Base Model

- Add one more batch normalization layer in the end before last dense layer, and change pool size from 2×2 to 3×3
- Regularization
- Dropout = 0.5

Training and Test Accuracy

trn: 0.960, val: 0.787 (30 epochs)

NOTE

For the training processes of all the models mentioned in Part 2, please refer to the appendix at the end of this document. In the appendix, we achieved a best test accuracy of 0.802, which might be slightly different from that of the `ipynb` file we submitted.

Appendix1

October 30, 2017

0.1 Model Improvement 1

change architecture(add one more batch normalization layer in the end before last dense layer) + exponential_decay of learning rate, test accuracy = 0.746 - base (0005) - modify the CNN class function (with global step as default, maybe?) - trn: 0.988, val: 0.732 (10 epochs) - trn: 1.000, val: 0.747 (20 epochs) - trn: 1.000, val: 0.746 (30 epochs)

```
In [52]: def model_fn(layer_input, labels, is_training):
        # conv - bn - max_pool
        F1 = 32
        layer = tf.layers.conv2d(
            inputs=layer_input, filters=F1, kernel_size=[5, 5], padding='same', activation=tf.nn.relu)
        layer = tf.layers.batch_normalization(
            inputs=layer, training=is_training)
        layer = tf.layers.max_pooling2d(inputs=layer, pool_size=[2, 2], strides=2)
        # conv - bn - max_pool
        F2 = 64
        layer = tf.layers.conv2d(
            inputs=layer, filters=F2, kernel_size=[5, 5], padding='same', activation=tf.nn.relu)
        layer = tf.layers.batch_normalization(
            inputs=layer, training=is_training)
        layer = tf.layers.max_pooling2d(inputs=layer, pool_size=[2, 2], strides=2)
        # dense1 - bn - softmax
        flat_size = int(H / 4 * W / 4 * F2)
        print(flat_size)
        layer = tf.reshape(layer, [-1, flat_size])
        layer = tf.layers.dense(
            inputs=layer, units=1024, activation=tf.nn.relu)
        layer = tf.layers.batch_normalization(
            inputs=layer, training=is_training)
        # layer_dropout = tf.layers.dropout(inputs=layer_dense, rate=0.4)
        # no activation here for logit, as it will be calculated in loss
        logit = tf.layers.dense(
            inputs=layer, units=T, activation=None)
        onehot_labels = tf.one_hot(indices=tf.cast(labels, tf.int64), depth=10)
        loss = tf.losses.softmax_cross_entropy(onehot_labels=onehot_labels, logits=logit)
        return logit, loss
```

```

tf.reset_default_graph()

starter_learning_rate = 0.1
global_step = tf.Variable(0, trainable=False)
learning_rate = tf.train.exponential_decay(
    learning_rate=starter_learning_rate, global_step=global_step,
    decay_steps=1000, decay_rate=0.96, staircase=True)
trainer = tf.train.GradientDescentOptimizer(learning_rate=learning_rate)
cnn = CNN(model_fn, trainer, global_step=global_step)
cnns['0020'] = cnn

```

4096

```

In [53]: print('train')
cnn.train(X_trn, y_trn, epochs=10, batch_size=50, print_every=100, plot_losses=True)
print('validation')
cnn.validate(X_val, y_val, epochs=1, batch_size=y_val.shape[0])

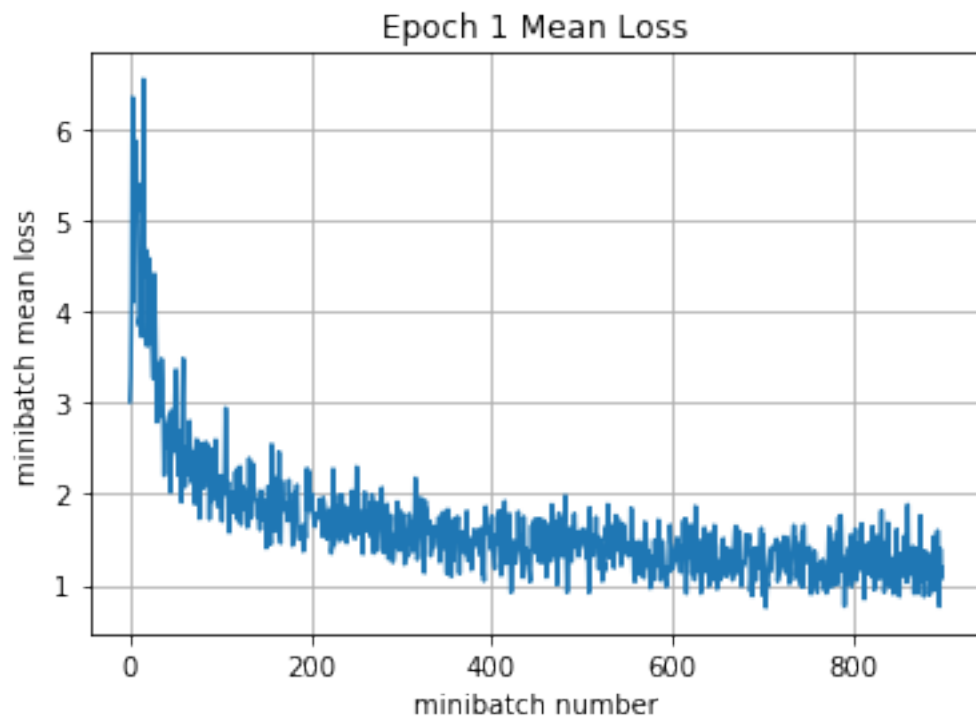
```

train

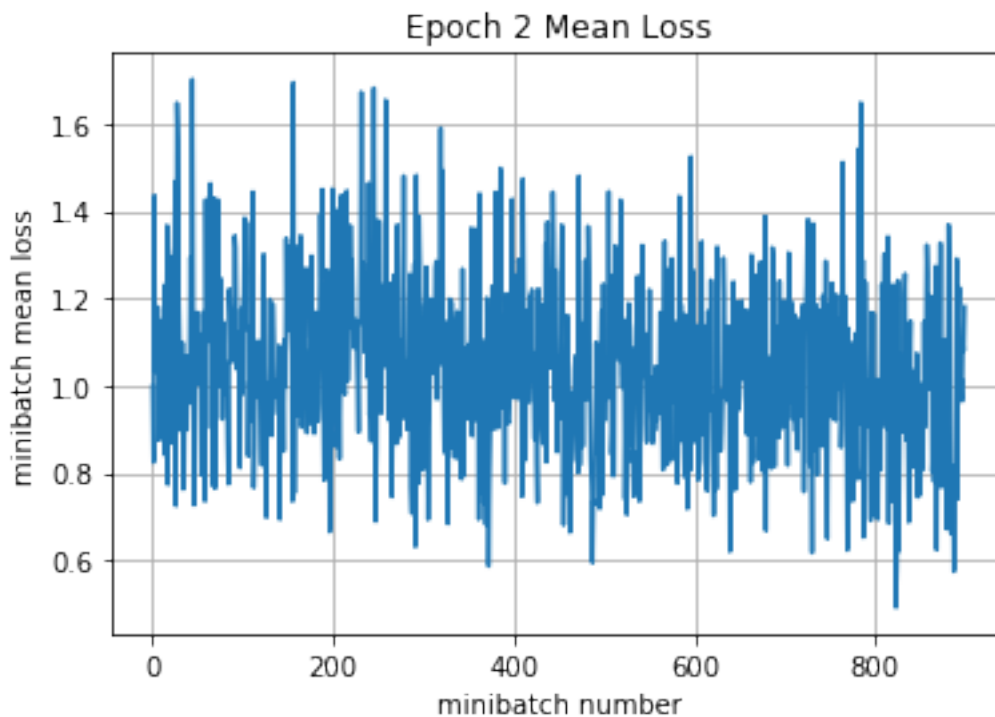
```

2017-10-27 01:25:18 Iter 0: batch trn loss = 3.011, accuracy = 0.160
2017-10-27 01:25:40 Iter 100: batch trn loss = 1.954, accuracy = 0.460
2017-10-27 01:26:01 Iter 200: batch trn loss = 1.821, accuracy = 0.360
2017-10-27 01:26:21 Iter 300: batch trn loss = 1.775, accuracy = 0.500
2017-10-27 01:26:42 Iter 400: batch trn loss = 1.536, accuracy = 0.560
2017-10-27 01:27:02 Iter 500: batch trn loss = 1.605, accuracy = 0.520
2017-10-27 01:27:23 Iter 600: batch trn loss = 1.412, accuracy = 0.540
2017-10-27 01:27:44 Iter 700: batch trn loss = 0.897, accuracy = 0.640
2017-10-27 01:28:06 Iter 800: batch trn loss = 1.232, accuracy = 0.680
Epoch 0: mean loss = 1.646, accuracy = 0.487

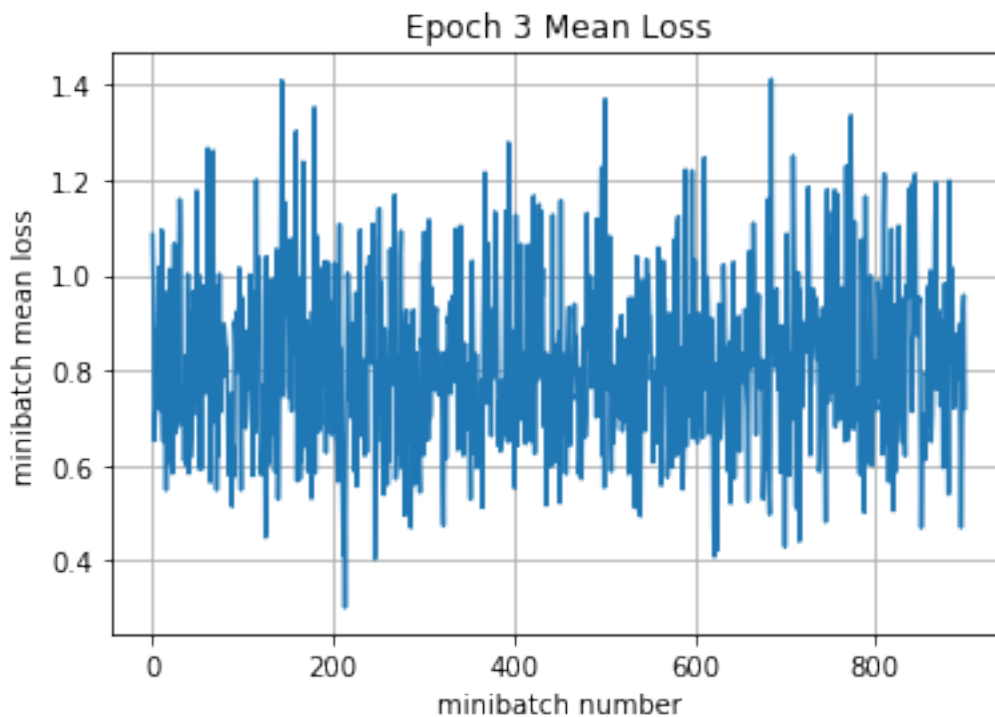
```

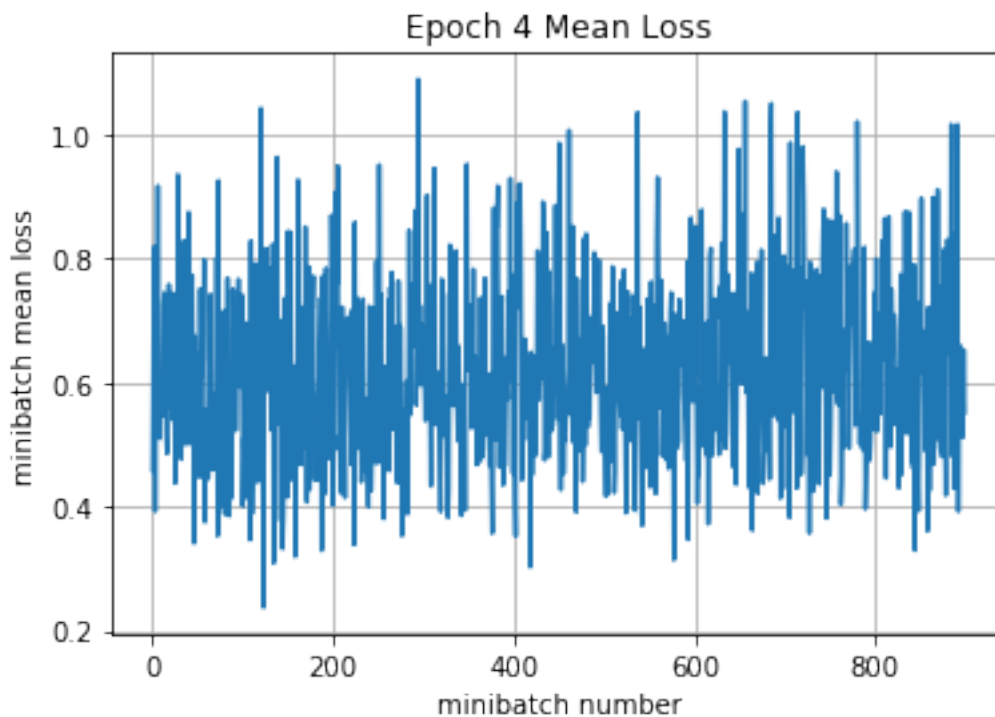
```
2017-10-27 01:28:28 Iter 900: batch trn loss = 0.998, accuracy = 0.660
2017-10-27 01:28:48 Iter 1000: batch trn loss = 1.191, accuracy = 0.540
2017-10-27 01:29:11 Iter 1100: batch trn loss = 0.946, accuracy = 0.700
2017-10-27 01:29:31 Iter 1200: batch trn loss = 0.940, accuracy = 0.700
2017-10-27 01:29:52 Iter 1300: batch trn loss = 0.990, accuracy = 0.600
2017-10-27 01:30:14 Iter 1400: batch trn loss = 0.919, accuracy = 0.680
2017-10-27 01:30:35 Iter 1500: batch trn loss = 0.871, accuracy = 0.660
2017-10-27 01:30:57 Iter 1600: batch trn loss = 1.098, accuracy = 0.600
2017-10-27 01:31:19 Iter 1700: batch trn loss = 0.729, accuracy = 0.660
Epoch 1: mean loss = 1.039, accuracy = 0.643
```



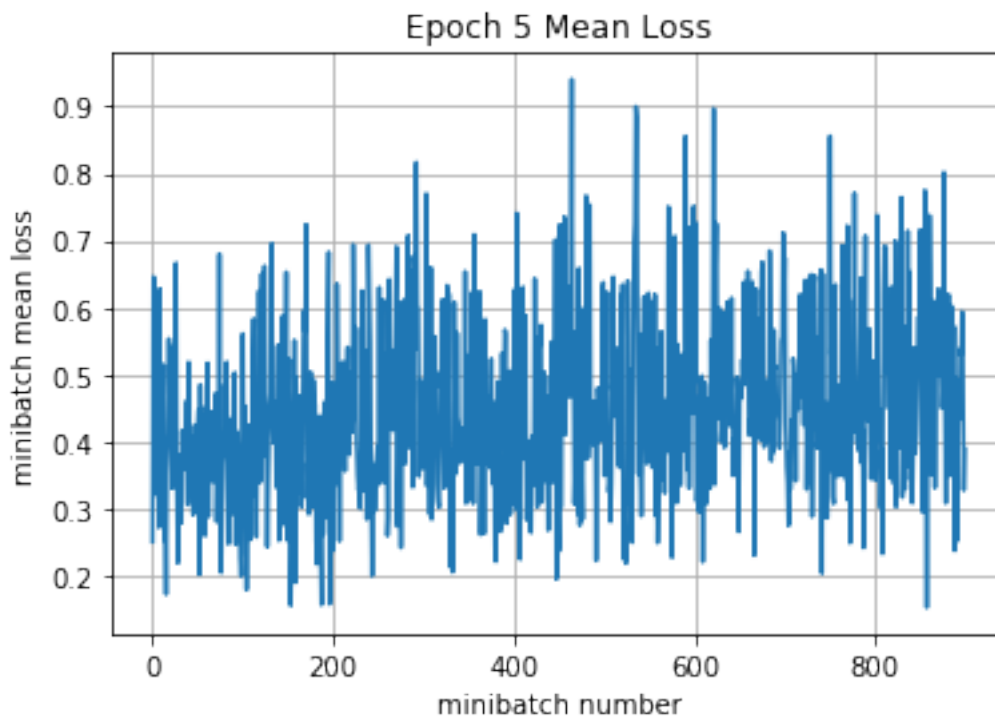
```
2017-10-27 01:31:40 Iter 1800: batch trn loss = 1.085, accuracy = 0.660
2017-10-27 01:32:02 Iter 1900: batch trn loss = 0.892, accuracy = 0.680
2017-10-27 01:32:23 Iter 2000: batch trn loss = 0.859, accuracy = 0.740
2017-10-27 01:32:46 Iter 2100: batch trn loss = 0.621, accuracy = 0.740
2017-10-27 01:33:10 Iter 2200: batch trn loss = 0.795, accuracy = 0.780
2017-10-27 01:33:35 Iter 2300: batch trn loss = 0.554, accuracy = 0.840
2017-10-27 01:33:59 Iter 2400: batch trn loss = 1.030, accuracy = 0.680
2017-10-27 01:34:21 Iter 2500: batch trn loss = 0.429, accuracy = 0.840
2017-10-27 01:34:42 Iter 2600: batch trn loss = 0.734, accuracy = 0.800
Epoch 2: mean loss = 0.811, accuracy = 0.722
```



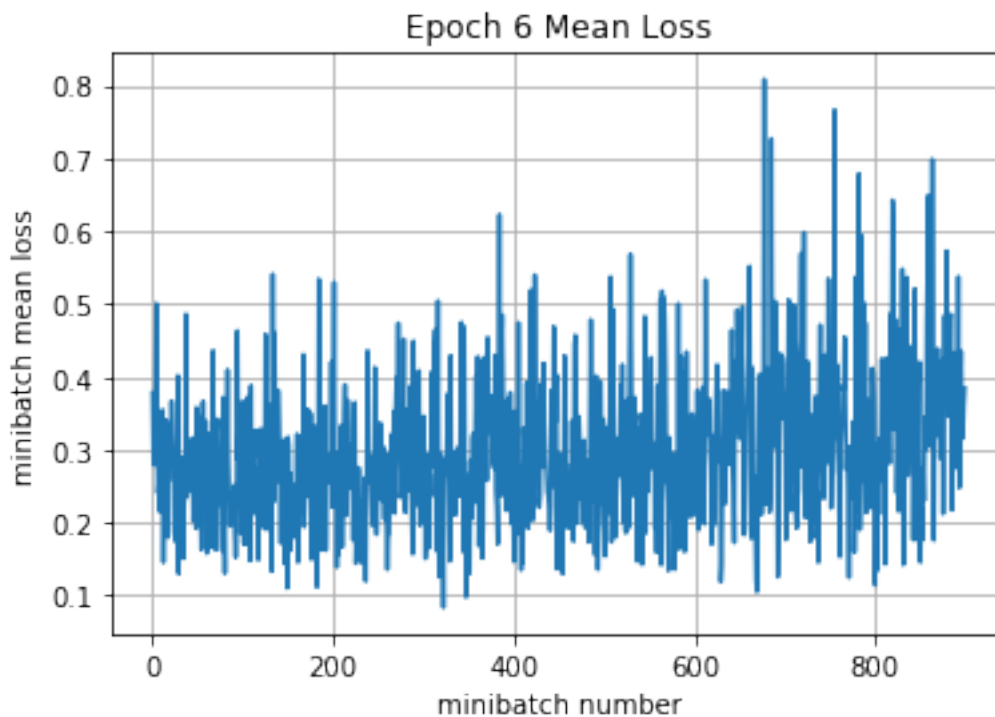
```
2017-10-27 01:35:04 Iter 2700: batch trn loss = 0.459, accuracy = 0.820
2017-10-27 01:35:26 Iter 2800: batch trn loss = 0.400, accuracy = 0.860
2017-10-27 01:35:51 Iter 2900: batch trn loss = 0.402, accuracy = 0.880
2017-10-27 01:36:16 Iter 3000: batch trn loss = 0.674, accuracy = 0.720
2017-10-27 01:36:37 Iter 3100: batch trn loss = 0.774, accuracy = 0.760
2017-10-27 01:36:59 Iter 3200: batch trn loss = 0.525, accuracy = 0.840
2017-10-27 01:37:21 Iter 3300: batch trn loss = 0.617, accuracy = 0.760
2017-10-27 01:37:42 Iter 3400: batch trn loss = 0.611, accuracy = 0.800
2017-10-27 01:38:03 Iter 3500: batch trn loss = 0.523, accuracy = 0.840
Epoch 3: mean loss = 0.618, accuracy = 0.786
```



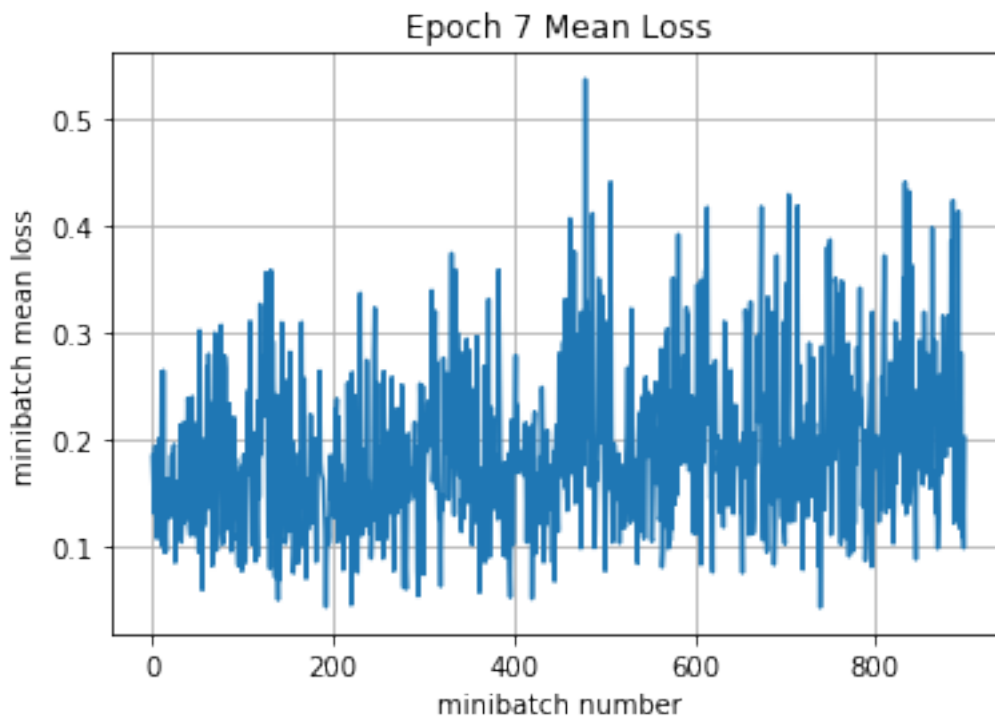
```
2017-10-27 01:38:23 Iter 3600: batch trn loss = 0.251, accuracy = 0.940
2017-10-27 01:38:43 Iter 3700: batch trn loss = 0.561, accuracy = 0.760
2017-10-27 01:39:04 Iter 3800: batch trn loss = 0.238, accuracy = 0.900
2017-10-27 01:39:24 Iter 3900: batch trn loss = 0.383, accuracy = 0.860
2017-10-27 01:39:44 Iter 4000: batch trn loss = 0.626, accuracy = 0.800
2017-10-27 01:40:05 Iter 4100: batch trn loss = 0.350, accuracy = 0.860
2017-10-27 01:40:25 Iter 4200: batch trn loss = 0.488, accuracy = 0.820
2017-10-27 01:40:45 Iter 4300: batch trn loss = 0.674, accuracy = 0.740
2017-10-27 01:41:05 Iter 4400: batch trn loss = 0.343, accuracy = 0.880
Epoch 4: mean loss = 0.448, accuracy = 0.844
```



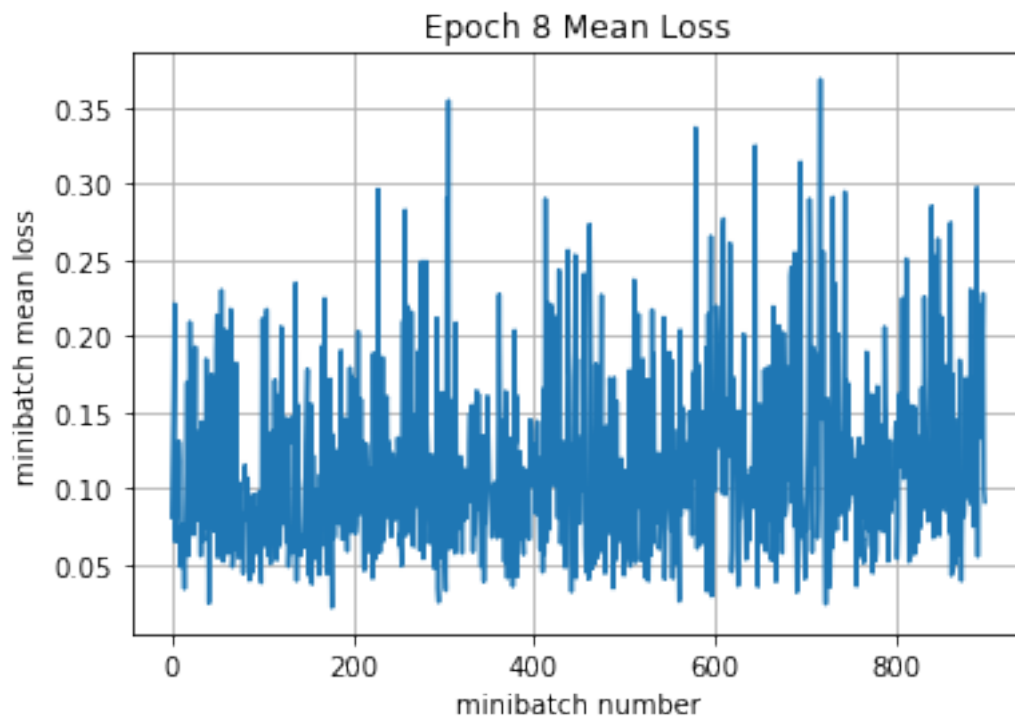
```
2017-10-27 01:41:26 Iter 4500: batch trn loss = 0.378, accuracy = 0.860
2017-10-27 01:41:46 Iter 4600: batch trn loss = 0.366, accuracy = 0.880
2017-10-27 01:42:07 Iter 4700: batch trn loss = 0.352, accuracy = 0.900
2017-10-27 01:42:28 Iter 4800: batch trn loss = 0.346, accuracy = 0.880
2017-10-27 01:42:48 Iter 4900: batch trn loss = 0.147, accuracy = 0.980
2017-10-27 01:43:08 Iter 5000: batch trn loss = 0.214, accuracy = 0.920
2017-10-27 01:43:29 Iter 5100: batch trn loss = 0.286, accuracy = 0.880
2017-10-27 01:43:49 Iter 5200: batch trn loss = 0.348, accuracy = 0.840
2017-10-27 01:44:11 Iter 5300: batch trn loss = 0.114, accuracy = 0.960
Epoch 5: mean loss = 0.297, accuracy = 0.897
```



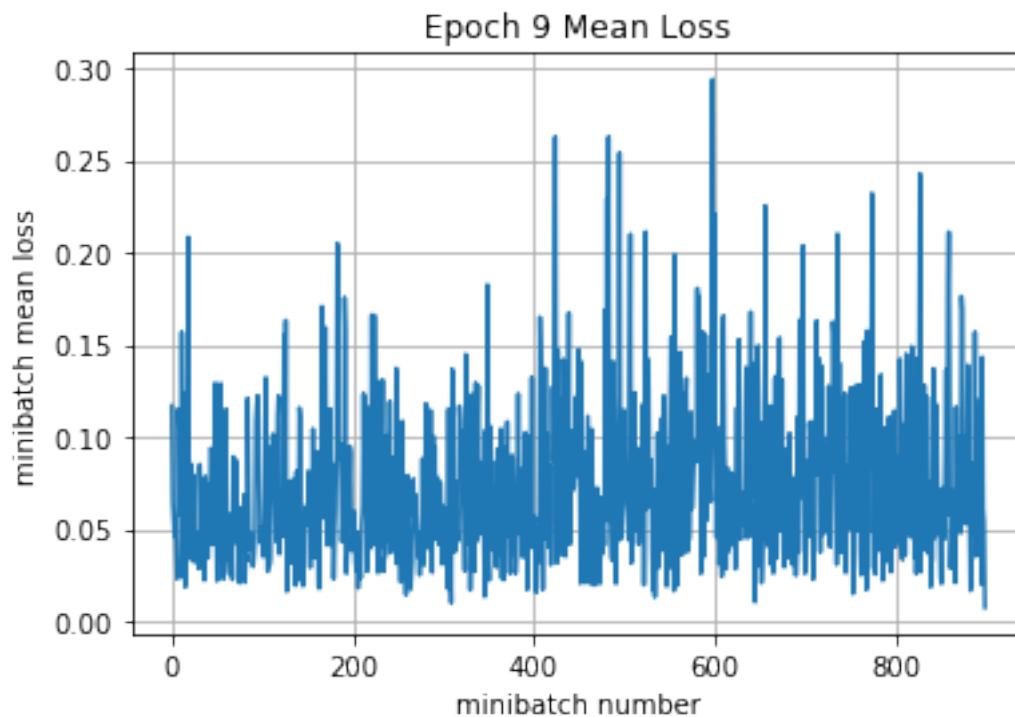
```
2017-10-27 01:44:31 Iter 5400: batch trn loss = 0.184, accuracy = 0.920
2017-10-27 01:44:51 Iter 5500: batch trn loss = 0.156, accuracy = 0.940
2017-10-27 01:45:11 Iter 5600: batch trn loss = 0.148, accuracy = 0.920
2017-10-27 01:45:32 Iter 5700: batch trn loss = 0.073, accuracy = 0.980
2017-10-27 01:45:52 Iter 5800: batch trn loss = 0.192, accuracy = 0.940
2017-10-27 01:46:14 Iter 5900: batch trn loss = 0.142, accuracy = 0.960
2017-10-27 01:46:34 Iter 6000: batch trn loss = 0.159, accuracy = 0.960
2017-10-27 01:46:55 Iter 6100: batch trn loss = 0.102, accuracy = 0.960
2017-10-27 01:47:15 Iter 6200: batch trn loss = 0.173, accuracy = 0.920
Epoch 6: mean loss = 0.189, accuracy = 0.936
```



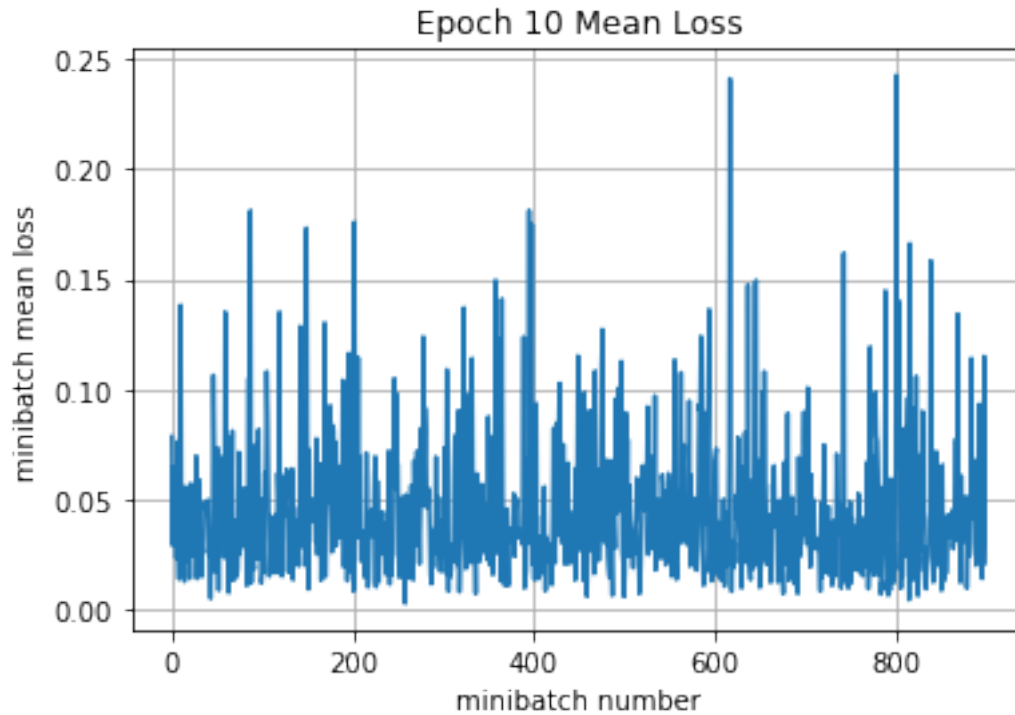
```
2017-10-27 01:47:35 Iter 6300: batch trn loss = 0.081, accuracy = 0.980
2017-10-27 01:47:55 Iter 6400: batch trn loss = 0.211, accuracy = 0.940
2017-10-27 01:48:16 Iter 6500: batch trn loss = 0.129, accuracy = 0.960
2017-10-27 01:48:36 Iter 6600: batch trn loss = 0.072, accuracy = 1.000
2017-10-27 01:48:56 Iter 6700: batch trn loss = 0.107, accuracy = 0.960
2017-10-27 01:49:16 Iter 6800: batch trn loss = 0.043, accuracy = 1.000
2017-10-27 01:49:37 Iter 6900: batch trn loss = 0.098, accuracy = 0.980
2017-10-27 01:49:57 Iter 7000: batch trn loss = 0.041, accuracy = 1.000
2017-10-27 01:50:17 Iter 7100: batch trn loss = 0.140, accuracy = 0.980
Epoch 7: mean loss = 0.115, accuracy = 0.963
```



```
2017-10-27 01:50:37 Iter 7200: batch trn loss = 0.117, accuracy = 0.920
2017-10-27 01:50:58 Iter 7300: batch trn loss = 0.041, accuracy = 1.000
2017-10-27 01:51:18 Iter 7400: batch trn loss = 0.046, accuracy = 0.980
2017-10-27 01:51:38 Iter 7500: batch trn loss = 0.062, accuracy = 0.980
2017-10-27 01:51:59 Iter 7600: batch trn loss = 0.051, accuracy = 0.980
2017-10-27 01:52:19 Iter 7700: batch trn loss = 0.115, accuracy = 0.940
2017-10-27 01:52:39 Iter 7800: batch trn loss = 0.221, accuracy = 0.940
2017-10-27 01:52:59 Iter 7900: batch trn loss = 0.026, accuracy = 1.000
2017-10-27 01:53:19 Iter 8000: batch trn loss = 0.040, accuracy = 0.980
Epoch 8: mean loss = 0.073, accuracy = 0.978
```

```
2017-10-27 01:53:40 Iter 8100: batch trn loss = 0.079, accuracy = 0.980
2017-10-27 01:54:00 Iter 8200: batch trn loss = 0.017, accuracy = 1.000
2017-10-27 01:54:21 Iter 8300: batch trn loss = 0.039, accuracy = 0.980
2017-10-27 01:54:41 Iter 8400: batch trn loss = 0.035, accuracy = 1.000
2017-10-27 01:55:01 Iter 8500: batch trn loss = 0.040, accuracy = 0.980
2017-10-27 01:55:21 Iter 8600: batch trn loss = 0.006, accuracy = 1.000
2017-10-27 01:55:41 Iter 8700: batch trn loss = 0.068, accuracy = 0.980
2017-10-27 01:56:02 Iter 8800: batch trn loss = 0.073, accuracy = 0.980
2017-10-27 01:56:22 Iter 8900: batch trn loss = 0.018, accuracy = 1.000
Epoch 9: mean loss = 0.044, accuracy = 0.988
```



validation

Epoch 0: mean loss = 1.212, accuracy = 0.732

Out[53]: (1.2121452, 0.73240000000000005)

In [54]: print('train')

cnn.train(X_trn, y_trn, epochs=10, batch_size=50, print_every=100, plot_losses=True)

print('validation')

cnn.validate(X_val, y_val, epochs=1, batch_size=y_val.shape[0])

train

2017-10-27 01:56:51 Iter 0: batch trn loss = 0.011, accuracy = 1.000

2017-10-27 01:57:12 Iter 100: batch trn loss = 0.021, accuracy = 1.000

2017-10-27 01:57:32 Iter 200: batch trn loss = 0.038, accuracy = 0.980

2017-10-27 01:57:52 Iter 300: batch trn loss = 0.009, accuracy = 1.000

2017-10-27 01:58:13 Iter 400: batch trn loss = 0.070, accuracy = 0.960

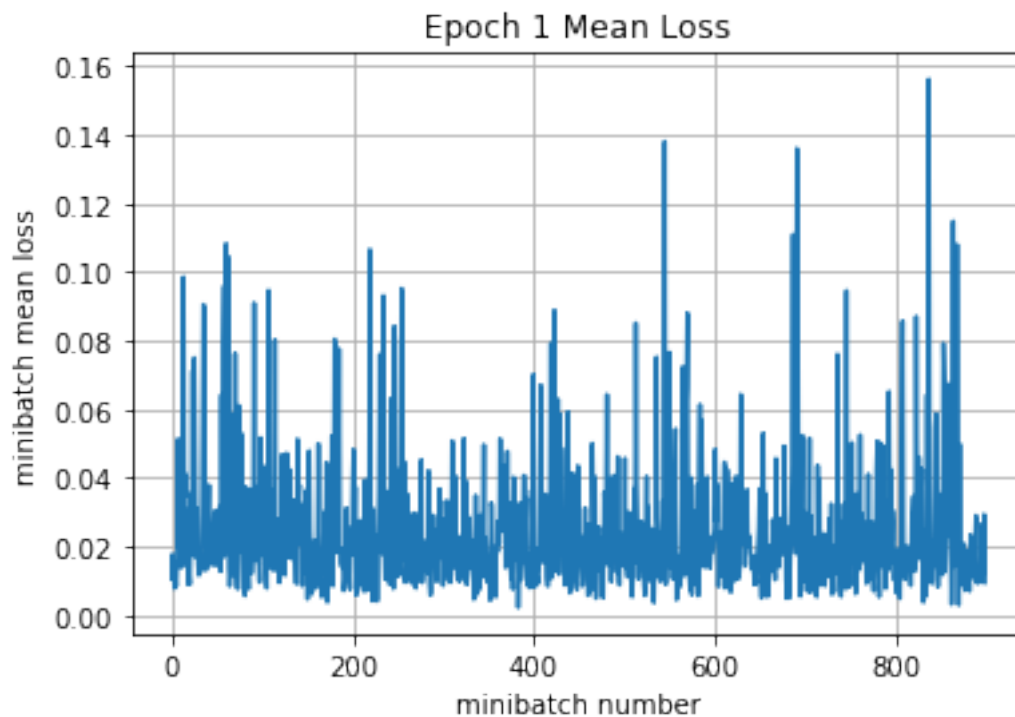
2017-10-27 01:58:33 Iter 500: batch trn loss = 0.016, accuracy = 1.000

2017-10-27 01:58:53 Iter 600: batch trn loss = 0.025, accuracy = 1.000

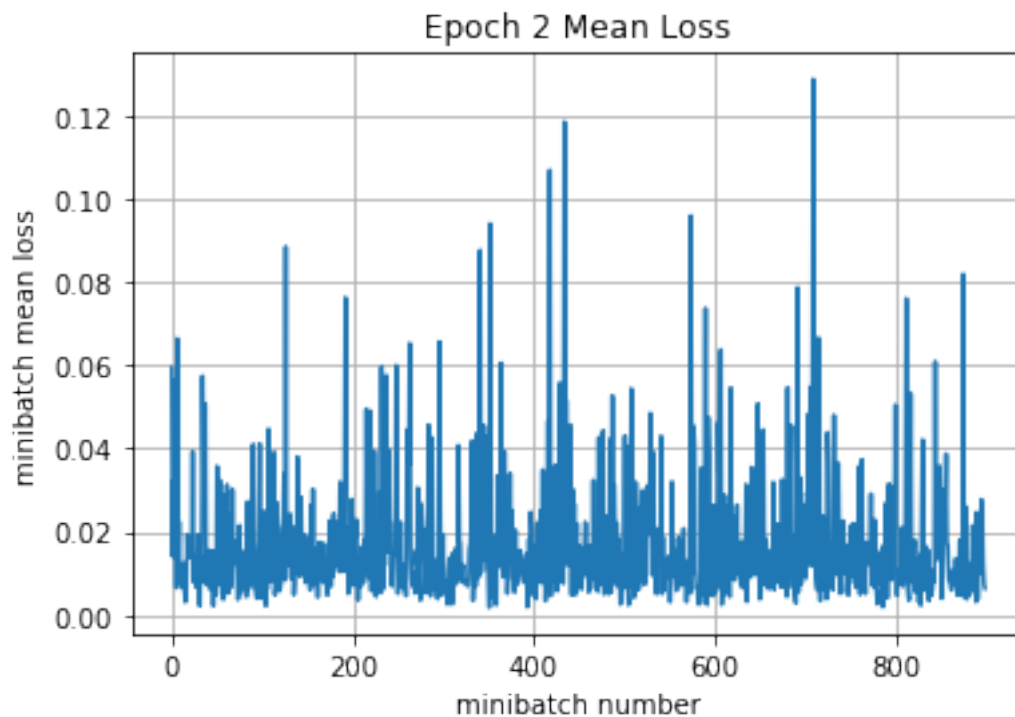
2017-10-27 01:59:13 Iter 700: batch trn loss = 0.010, accuracy = 1.000

2017-10-27 01:59:34 Iter 800: batch trn loss = 0.018, accuracy = 1.000

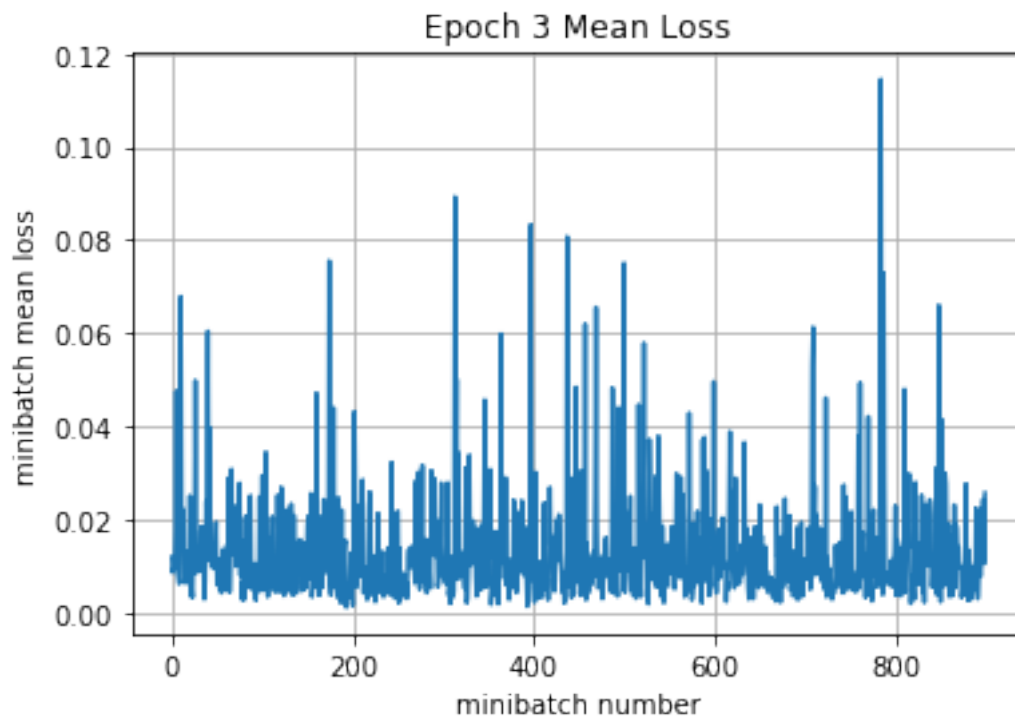
Epoch 0: mean loss = 0.025, accuracy = 0.995



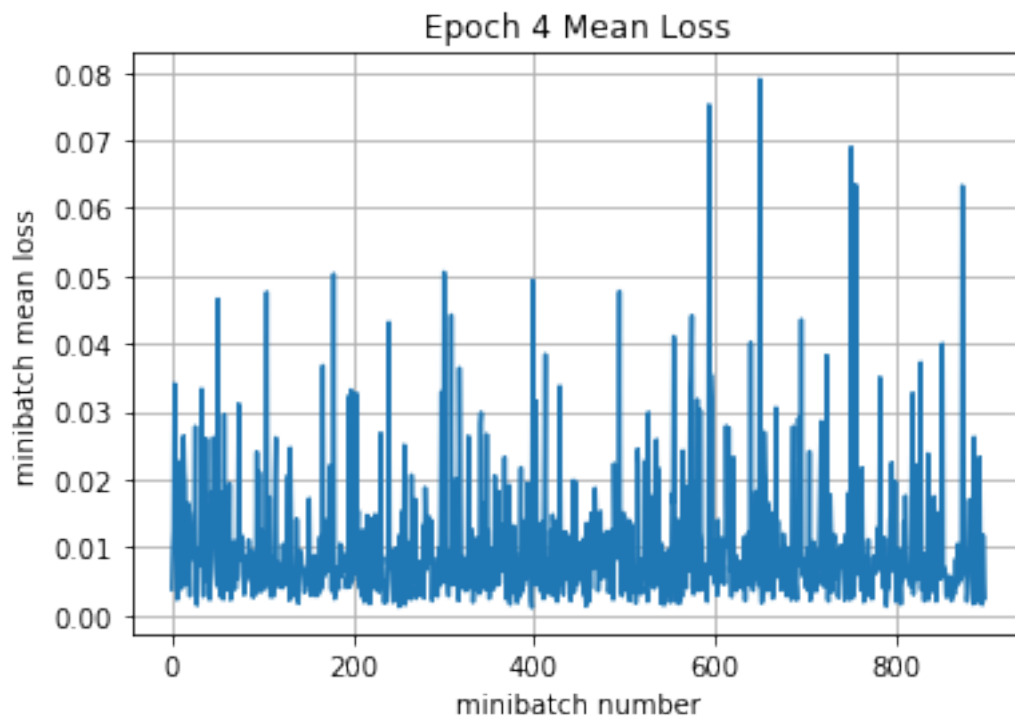
```
2017-10-27 01:59:54 Iter 900: batch trn loss = 0.059, accuracy = 0.980
2017-10-27 02:00:14 Iter 1000: batch trn loss = 0.015, accuracy = 1.000
2017-10-27 02:00:35 Iter 1100: batch trn loss = 0.015, accuracy = 1.000
2017-10-27 02:00:55 Iter 1200: batch trn loss = 0.019, accuracy = 1.000
2017-10-27 02:01:15 Iter 1300: batch trn loss = 0.005, accuracy = 1.000
2017-10-27 02:01:35 Iter 1400: batch trn loss = 0.021, accuracy = 0.980
2017-10-27 02:01:56 Iter 1500: batch trn loss = 0.005, accuracy = 1.000
2017-10-27 02:02:16 Iter 1600: batch trn loss = 0.014, accuracy = 1.000
2017-10-27 02:02:36 Iter 1700: batch trn loss = 0.050, accuracy = 0.980
Epoch 1: mean loss = 0.017, accuracy = 0.997
```



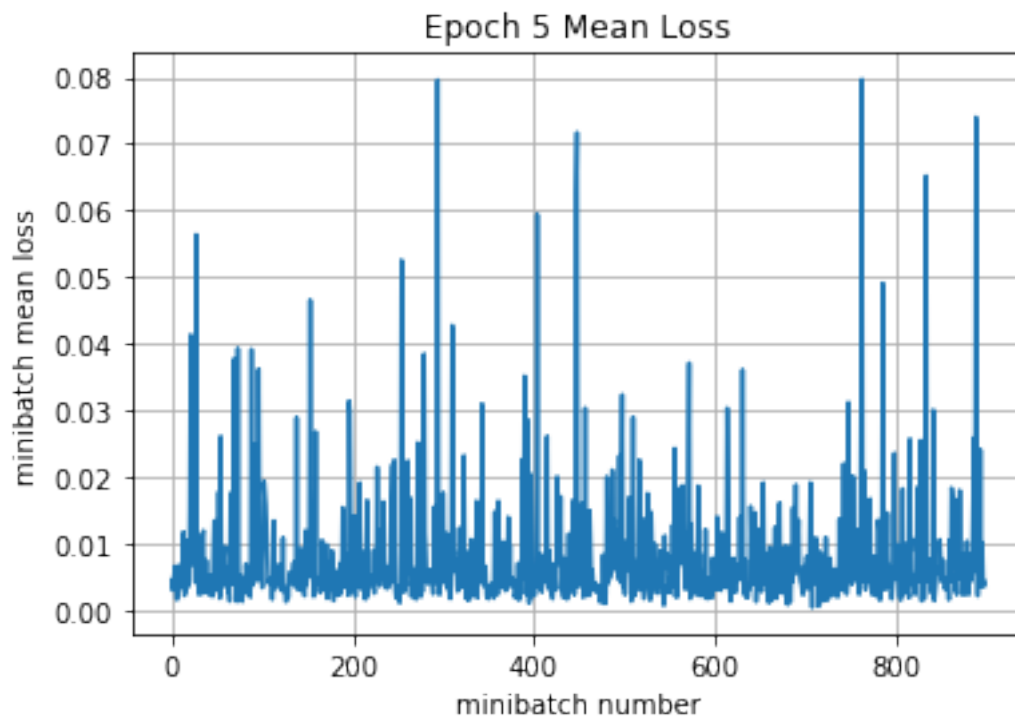
```
2017-10-27 02:02:57 Iter 1800: batch trn loss = 0.012, accuracy = 1.000
2017-10-27 02:03:17 Iter 1900: batch trn loss = 0.029, accuracy = 1.000
2017-10-27 02:03:38 Iter 2000: batch trn loss = 0.001, accuracy = 1.000
2017-10-27 02:03:58 Iter 2100: batch trn loss = 0.008, accuracy = 1.000
2017-10-27 02:04:19 Iter 2200: batch trn loss = 0.005, accuracy = 1.000
2017-10-27 02:04:39 Iter 2300: batch trn loss = 0.075, accuracy = 0.960
2017-10-27 02:04:59 Iter 2400: batch trn loss = 0.010, accuracy = 1.000
2017-10-27 02:05:19 Iter 2500: batch trn loss = 0.008, accuracy = 1.000
2017-10-27 02:05:42 Iter 2600: batch trn loss = 0.023, accuracy = 0.980
Epoch 2: mean loss = 0.013, accuracy = 0.998
```



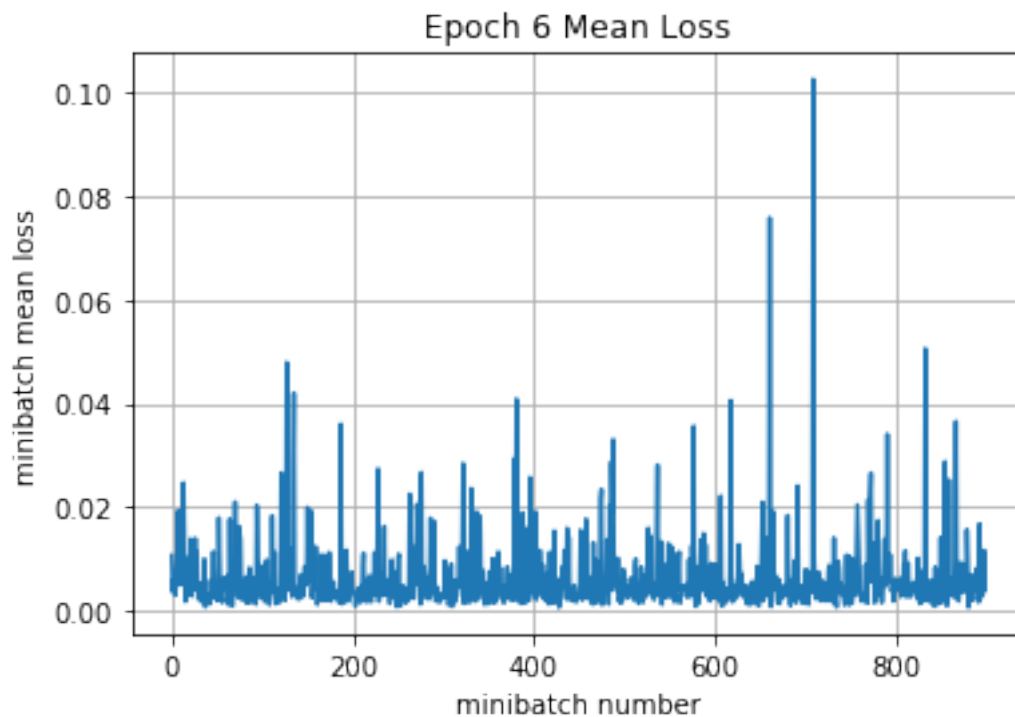
```
2017-10-27 02:06:03 Iter 2700: batch trn loss = 0.004, accuracy = 1.000
2017-10-27 02:06:23 Iter 2800: batch trn loss = 0.006, accuracy = 1.000
2017-10-27 02:06:44 Iter 2900: batch trn loss = 0.005, accuracy = 1.000
2017-10-27 02:07:04 Iter 3000: batch trn loss = 0.005, accuracy = 1.000
2017-10-27 02:07:24 Iter 3100: batch trn loss = 0.022, accuracy = 1.000
2017-10-27 02:07:45 Iter 3200: batch trn loss = 0.004, accuracy = 1.000
2017-10-27 02:08:05 Iter 3300: batch trn loss = 0.005, accuracy = 1.000
2017-10-27 02:08:26 Iter 3400: batch trn loss = 0.008, accuracy = 1.000
2017-10-27 02:08:46 Iter 3500: batch trn loss = 0.020, accuracy = 0.980
Epoch 3: mean loss = 0.010, accuracy = 0.999
```



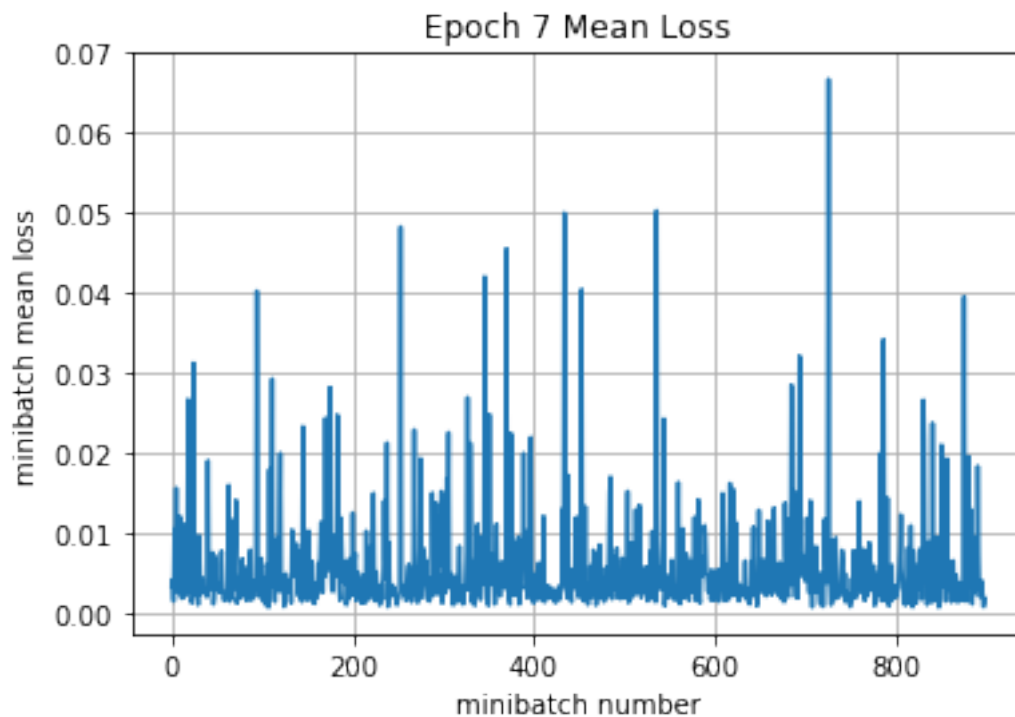
```
2017-10-27 02:09:06 Iter 3600: batch trn loss = 0.005, accuracy = 1.000
2017-10-27 02:09:27 Iter 3700: batch trn loss = 0.003, accuracy = 1.000
2017-10-27 02:09:47 Iter 3800: batch trn loss = 0.003, accuracy = 1.000
2017-10-27 02:10:07 Iter 3900: batch trn loss = 0.018, accuracy = 1.000
2017-10-27 02:10:27 Iter 4000: batch trn loss = 0.006, accuracy = 1.000
2017-10-27 02:10:48 Iter 4100: batch trn loss = 0.007, accuracy = 1.000
2017-10-27 02:11:08 Iter 4200: batch trn loss = 0.005, accuracy = 1.000
2017-10-27 02:11:28 Iter 4300: batch trn loss = 0.005, accuracy = 1.000
2017-10-27 02:11:48 Iter 4400: batch trn loss = 0.005, accuracy = 1.000
Epoch 4: mean loss = 0.008, accuracy = 0.999
```



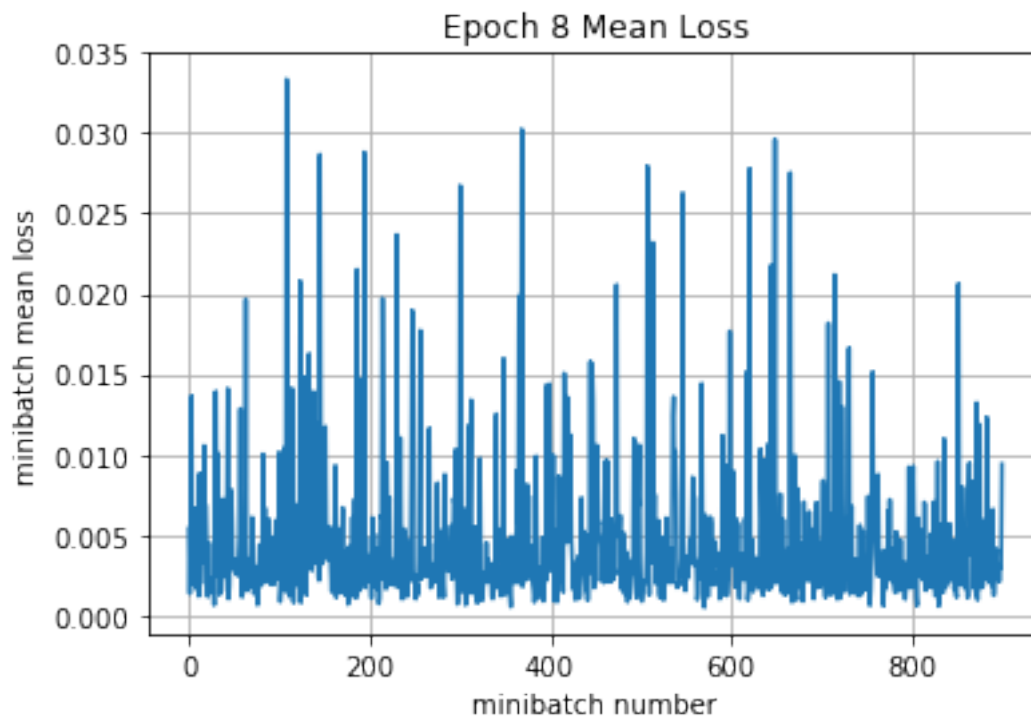
```
2017-10-27 02:12:09 Iter 4500: batch trn loss = 0.011, accuracy = 1.000
2017-10-27 02:12:29 Iter 4600: batch trn loss = 0.003, accuracy = 1.000
2017-10-27 02:12:50 Iter 4700: batch trn loss = 0.003, accuracy = 1.000
2017-10-27 02:13:10 Iter 4800: batch trn loss = 0.003, accuracy = 1.000
2017-10-27 02:13:30 Iter 4900: batch trn loss = 0.003, accuracy = 1.000
2017-10-27 02:13:51 Iter 5000: batch trn loss = 0.005, accuracy = 1.000
2017-10-27 02:14:11 Iter 5100: batch trn loss = 0.005, accuracy = 1.000
2017-10-27 02:14:31 Iter 5200: batch trn loss = 0.006, accuracy = 1.000
2017-10-27 02:14:51 Iter 5300: batch trn loss = 0.006, accuracy = 1.000
Epoch 5: mean loss = 0.006, accuracy = 0.999
```



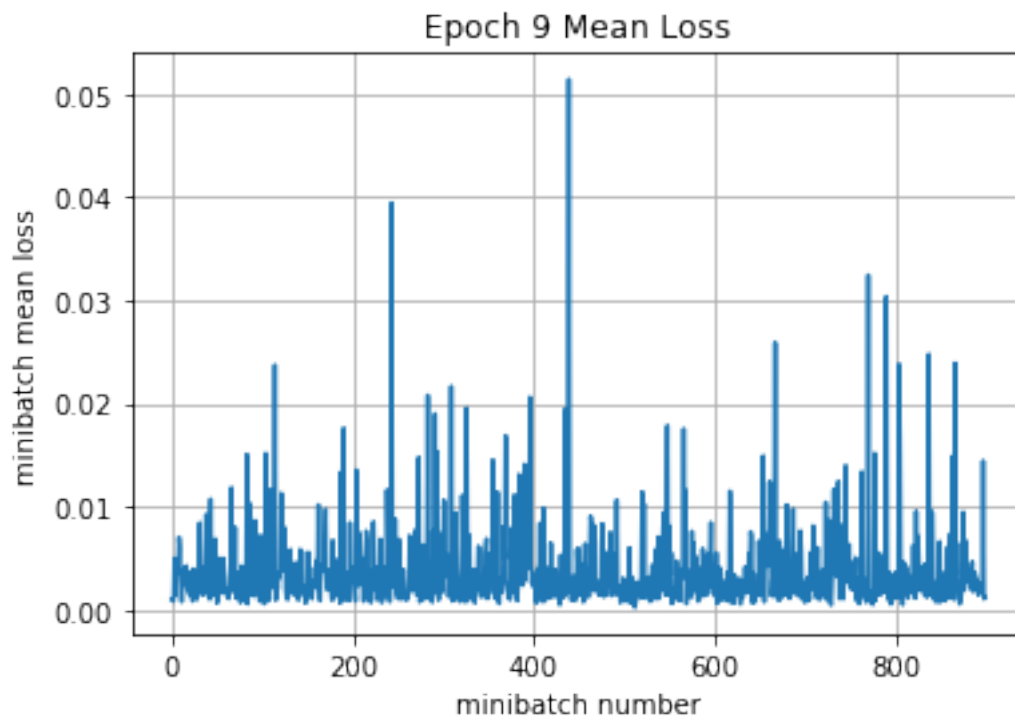
```
2017-10-27 02:15:12 Iter 5400: batch trn loss = 0.003, accuracy = 1.000
2017-10-27 02:15:32 Iter 5500: batch trn loss = 0.003, accuracy = 1.000
2017-10-27 02:15:52 Iter 5600: batch trn loss = 0.012, accuracy = 1.000
2017-10-27 02:16:13 Iter 5700: batch trn loss = 0.005, accuracy = 1.000
2017-10-27 02:16:33 Iter 5800: batch trn loss = 0.001, accuracy = 1.000
2017-10-27 02:16:53 Iter 5900: batch trn loss = 0.002, accuracy = 1.000
2017-10-27 02:17:13 Iter 6000: batch trn loss = 0.002, accuracy = 1.000
2017-10-27 02:17:33 Iter 6100: batch trn loss = 0.004, accuracy = 1.000
2017-10-27 02:17:54 Iter 6200: batch trn loss = 0.003, accuracy = 1.000
Epoch 6: mean loss = 0.006, accuracy = 0.999
```

```
2017-10-27 02:18:14 Iter 6300: batch trn loss = 0.005, accuracy = 1.000
2017-10-27 02:18:35 Iter 6400: batch trn loss = 0.010, accuracy = 1.000
2017-10-27 02:18:55 Iter 6500: batch trn loss = 0.003, accuracy = 1.000
2017-10-27 02:19:15 Iter 6600: batch trn loss = 0.027, accuracy = 1.000
2017-10-27 02:19:35 Iter 6700: batch trn loss = 0.004, accuracy = 1.000
2017-10-27 02:19:56 Iter 6800: batch trn loss = 0.001, accuracy = 1.000
2017-10-27 02:20:16 Iter 6900: batch trn loss = 0.004, accuracy = 1.000
2017-10-27 02:20:36 Iter 7000: batch trn loss = 0.005, accuracy = 1.000
2017-10-27 02:20:56 Iter 7100: batch trn loss = 0.006, accuracy = 1.000
Epoch 7: mean loss = 0.005, accuracy = 1.000
```



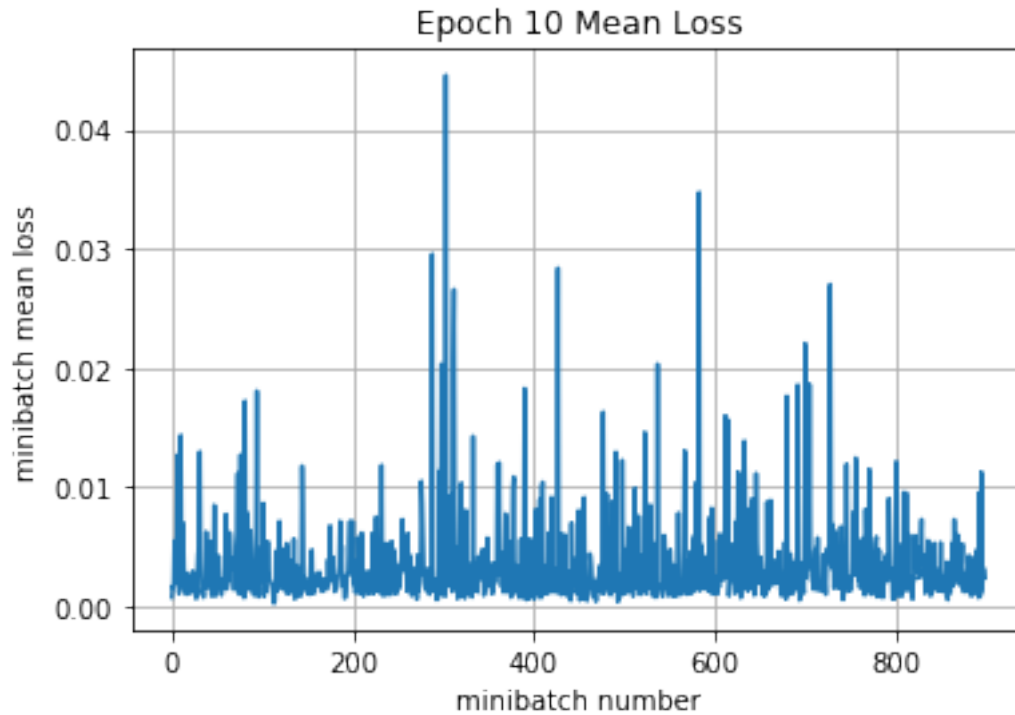
```
2017-10-27 02:21:16 Iter 7200: batch trn loss = 0.001, accuracy = 1.000
2017-10-27 02:21:37 Iter 7300: batch trn loss = 0.001, accuracy = 1.000
2017-10-27 02:21:57 Iter 7400: batch trn loss = 0.003, accuracy = 1.000
2017-10-27 02:22:17 Iter 7500: batch trn loss = 0.002, accuracy = 1.000
2017-10-27 02:22:38 Iter 7600: batch trn loss = 0.002, accuracy = 1.000
2017-10-27 02:22:58 Iter 7700: batch trn loss = 0.002, accuracy = 1.000
2017-10-27 02:23:18 Iter 7800: batch trn loss = 0.004, accuracy = 1.000
2017-10-27 02:23:38 Iter 7900: batch trn loss = 0.001, accuracy = 1.000
2017-10-27 02:23:59 Iter 8000: batch trn loss = 0.004, accuracy = 1.000
Epoch 8: mean loss = 0.004, accuracy = 1.000
```



```

2017-10-27 02:24:19 Iter 8100: batch trn loss = 0.001, accuracy = 1.000
2017-10-27 02:24:39 Iter 8200: batch trn loss = 0.003, accuracy = 1.000
2017-10-27 02:25:00 Iter 8300: batch trn loss = 0.007, accuracy = 1.000
2017-10-27 02:25:20 Iter 8400: batch trn loss = 0.003, accuracy = 1.000
2017-10-27 02:25:40 Iter 8500: batch trn loss = 0.003, accuracy = 1.000
2017-10-27 02:26:00 Iter 8600: batch trn loss = 0.003, accuracy = 1.000
2017-10-27 02:26:21 Iter 8700: batch trn loss = 0.002, accuracy = 1.000
2017-10-27 02:26:41 Iter 8800: batch trn loss = 0.022, accuracy = 0.980
2017-10-27 02:27:01 Iter 8900: batch trn loss = 0.001, accuracy = 1.000
Epoch 9: mean loss = 0.004, accuracy = 1.000

```



validation

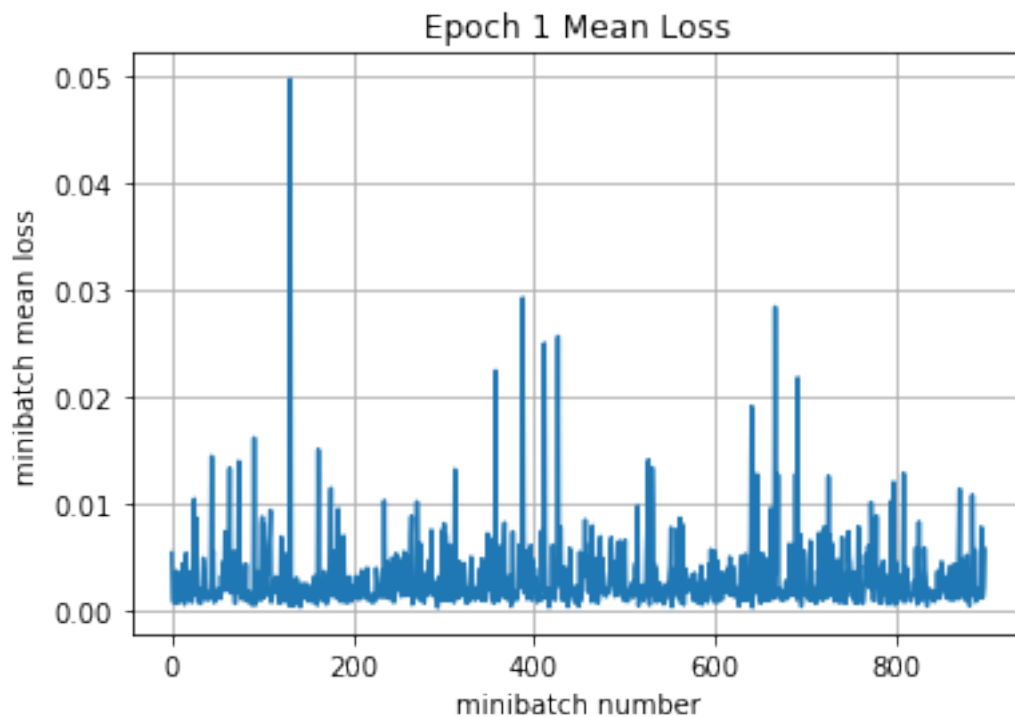
Epoch 0: mean loss = 1.271, accuracy = 0.747

Out[54]: (1.2705626, 0.74680000000000002)

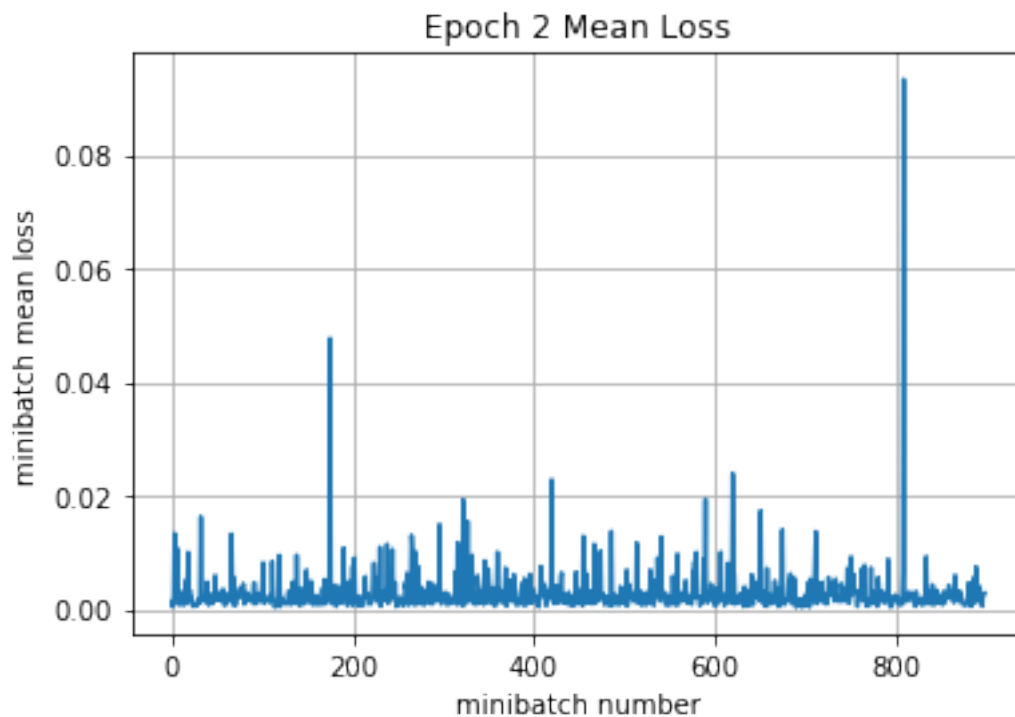
```
In [55]: print('train')
          cnn.train(X_trn, y_trn, epochs=10, batch_size=50, print_every=100, plot_losses=True)
          print('validation')
          cnn.validate(X_val, y_val, epochs=1, batch_size=y_val.shape[0])
```

train

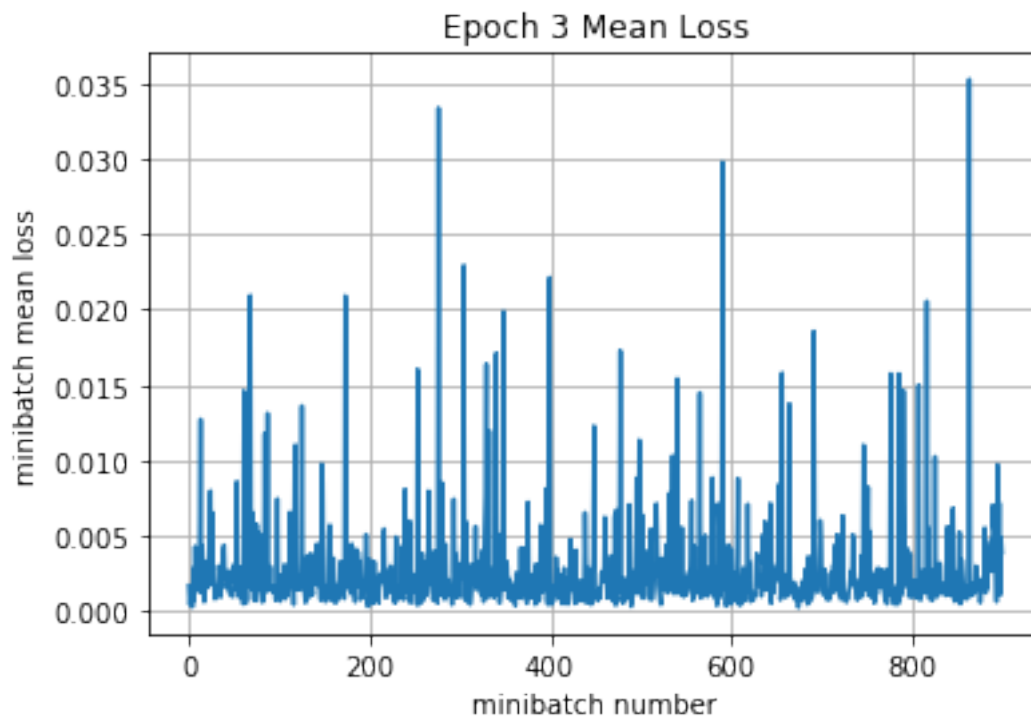
```
2017-10-27 02:27:30 Iter 0: batch trn loss = 0.005, accuracy = 1.000
2017-10-27 02:27:51 Iter 100: batch trn loss = 0.009, accuracy = 1.000
2017-10-27 02:28:11 Iter 200: batch trn loss = 0.002, accuracy = 1.000
2017-10-27 02:28:31 Iter 300: batch trn loss = 0.001, accuracy = 1.000
2017-10-27 02:28:52 Iter 400: batch trn loss = 0.002, accuracy = 1.000
2017-10-27 02:29:12 Iter 500: batch trn loss = 0.004, accuracy = 1.000
2017-10-27 02:29:32 Iter 600: batch trn loss = 0.006, accuracy = 1.000
2017-10-27 02:29:52 Iter 700: batch trn loss = 0.005, accuracy = 1.000
2017-10-27 02:30:13 Iter 800: batch trn loss = 0.001, accuracy = 1.000
Epoch 0: mean loss = 0.003, accuracy = 1.000
```



```
2017-10-27 02:30:33 Iter 900: batch trn loss = 0.001, accuracy = 1.000
2017-10-27 02:30:53 Iter 1000: batch trn loss = 0.005, accuracy = 1.000
2017-10-27 02:31:14 Iter 1100: batch trn loss = 0.008, accuracy = 1.000
2017-10-27 02:31:34 Iter 1200: batch trn loss = 0.002, accuracy = 1.000
2017-10-27 02:31:54 Iter 1300: batch trn loss = 0.004, accuracy = 1.000
2017-10-27 02:32:14 Iter 1400: batch trn loss = 0.001, accuracy = 1.000
2017-10-27 02:32:35 Iter 1500: batch trn loss = 0.001, accuracy = 1.000
2017-10-27 02:32:55 Iter 1600: batch trn loss = 0.003, accuracy = 1.000
2017-10-27 02:33:15 Iter 1700: batch trn loss = 0.003, accuracy = 1.000
Epoch 1: mean loss = 0.003, accuracy = 1.000
```



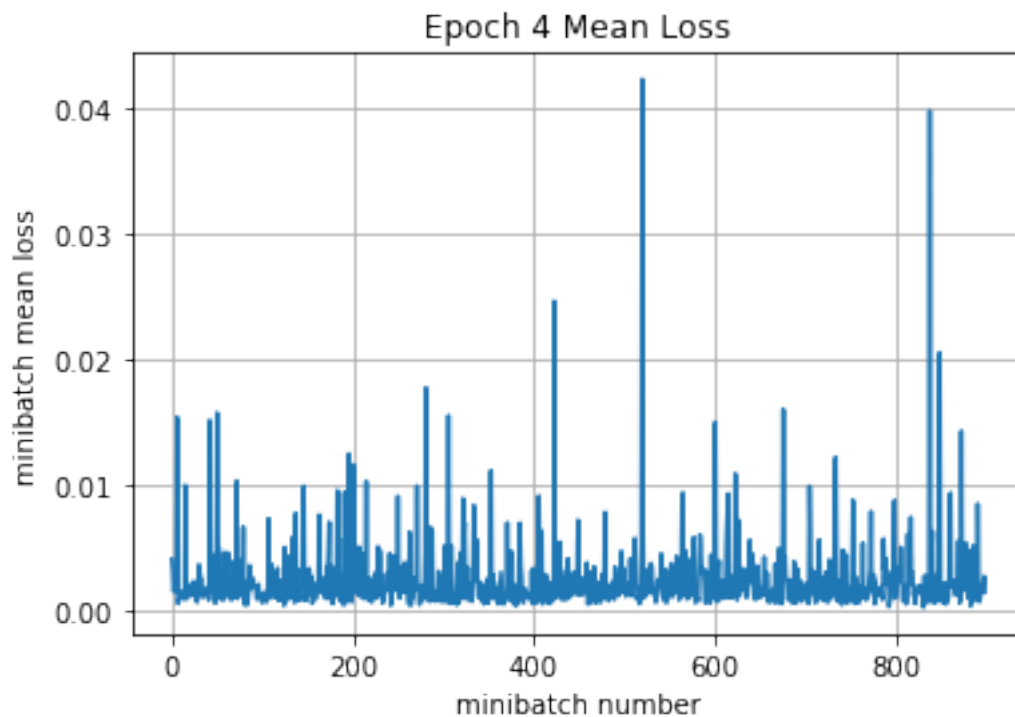
```
2017-10-27 02:33:36 Iter 1800: batch trn loss = 0.002, accuracy = 1.000
2017-10-27 02:33:56 Iter 1900: batch trn loss = 0.001, accuracy = 1.000
2017-10-27 02:34:17 Iter 2000: batch trn loss = 0.003, accuracy = 1.000
2017-10-27 02:34:37 Iter 2100: batch trn loss = 0.002, accuracy = 1.000
2017-10-27 02:34:57 Iter 2200: batch trn loss = 0.002, accuracy = 1.000
2017-10-27 02:35:18 Iter 2300: batch trn loss = 0.002, accuracy = 1.000
2017-10-27 02:35:39 Iter 2400: batch trn loss = 0.004, accuracy = 1.000
2017-10-27 02:36:02 Iter 2500: batch trn loss = 0.002, accuracy = 1.000
2017-10-27 02:36:22 Iter 2600: batch trn loss = 0.001, accuracy = 1.000
Epoch 2: mean loss = 0.003, accuracy = 1.000
```



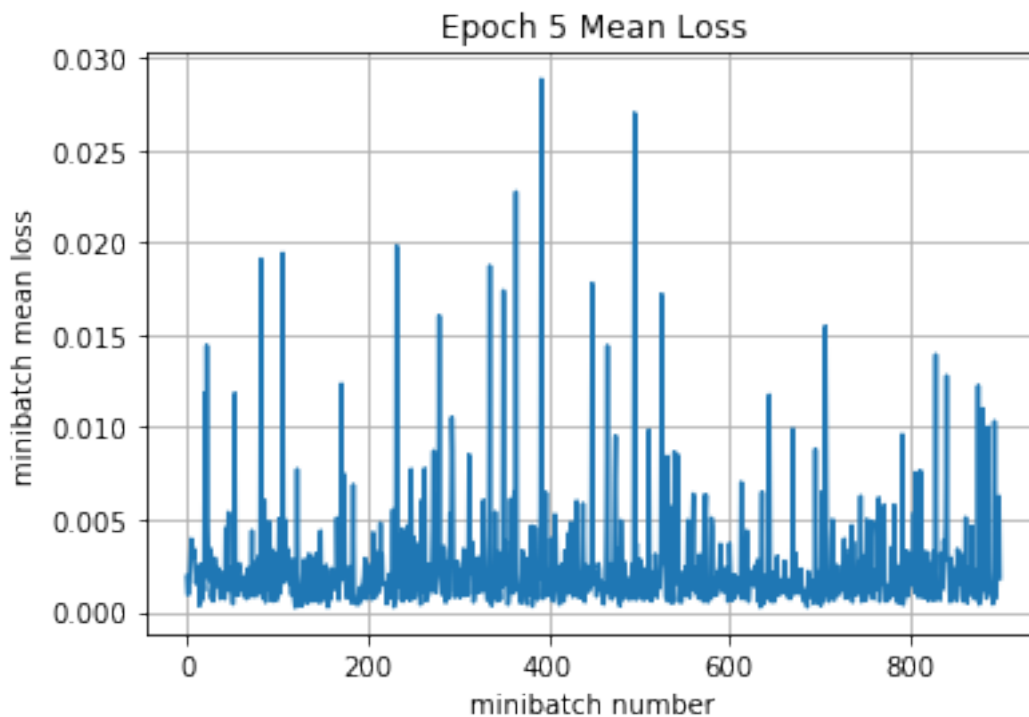
```

2017-10-27 02:36:42 Iter 2700: batch trn loss = 0.004, accuracy = 1.000
2017-10-27 02:37:02 Iter 2800: batch trn loss = 0.001, accuracy = 1.000
2017-10-27 02:37:22 Iter 2900: batch trn loss = 0.002, accuracy = 1.000
2017-10-27 02:37:43 Iter 3000: batch trn loss = 0.003, accuracy = 1.000
2017-10-27 02:38:03 Iter 3100: batch trn loss = 0.003, accuracy = 1.000
2017-10-27 02:38:23 Iter 3200: batch trn loss = 0.003, accuracy = 1.000
2017-10-27 02:38:43 Iter 3300: batch trn loss = 0.001, accuracy = 1.000
2017-10-27 02:39:04 Iter 3400: batch trn loss = 0.001, accuracy = 1.000
2017-10-27 02:39:24 Iter 3500: batch trn loss = 0.001, accuracy = 1.000
Epoch 3: mean loss = 0.002, accuracy = 1.000

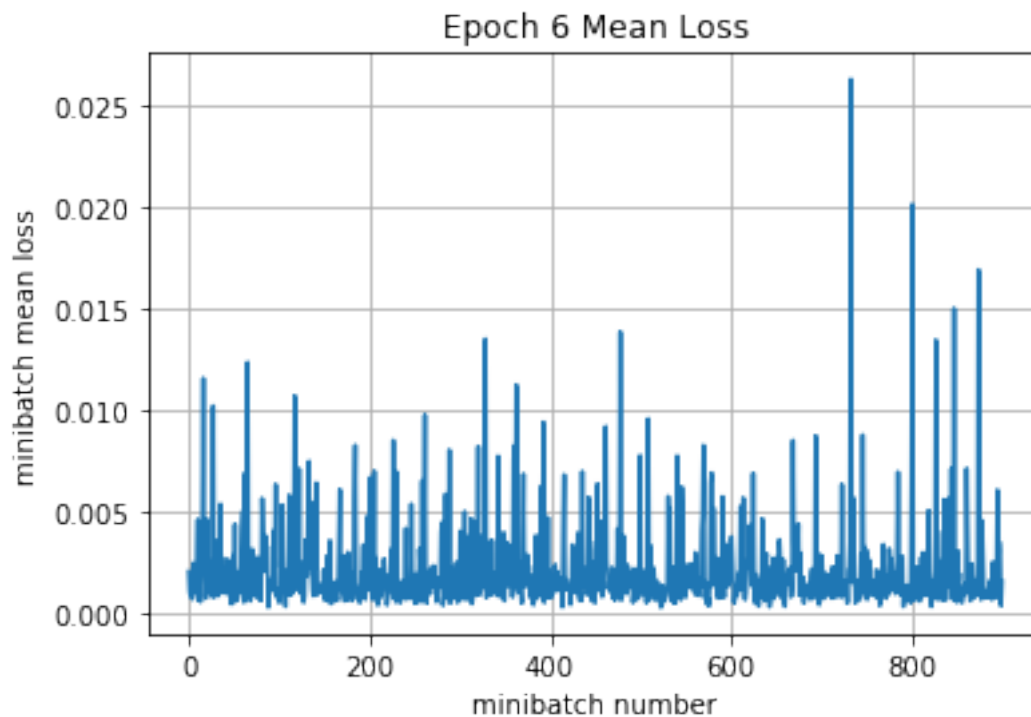
```



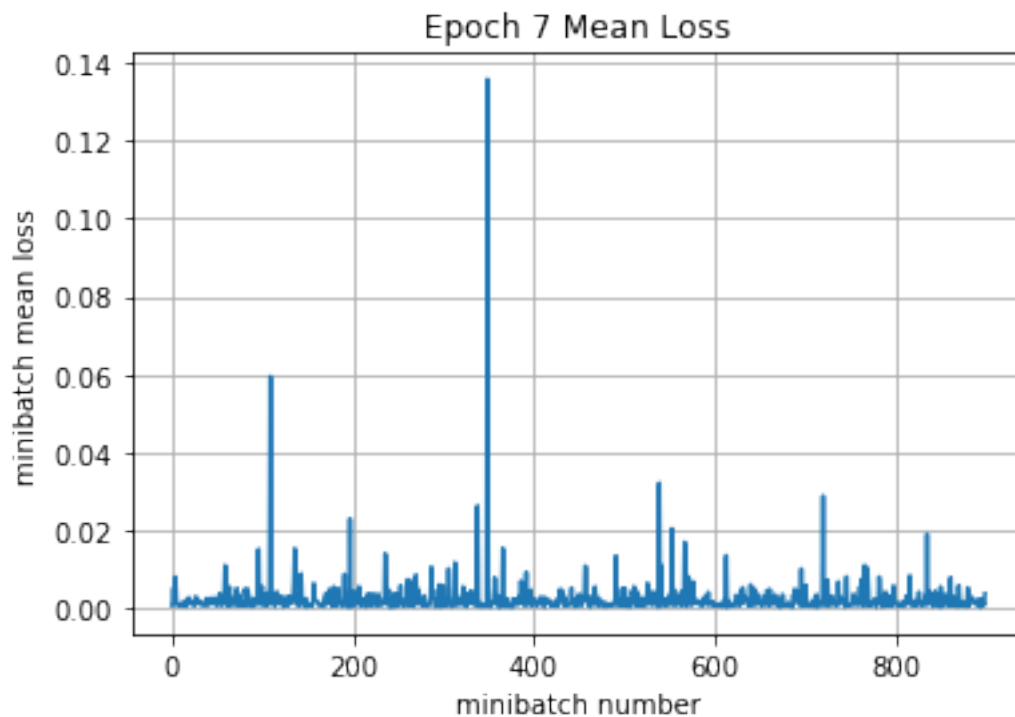
```
2017-10-27 02:39:45 Iter 3600: batch trn loss = 0.002, accuracy = 1.000
2017-10-27 02:40:05 Iter 3700: batch trn loss = 0.002, accuracy = 1.000
2017-10-27 02:40:25 Iter 3800: batch trn loss = 0.003, accuracy = 1.000
2017-10-27 02:40:45 Iter 3900: batch trn loss = 0.002, accuracy = 1.000
2017-10-27 02:41:06 Iter 4000: batch trn loss = 0.002, accuracy = 1.000
2017-10-27 02:41:26 Iter 4100: batch trn loss = 0.003, accuracy = 1.000
2017-10-27 02:41:46 Iter 4200: batch trn loss = 0.004, accuracy = 1.000
2017-10-27 02:42:07 Iter 4300: batch trn loss = 0.002, accuracy = 1.000
2017-10-27 02:42:27 Iter 4400: batch trn loss = 0.002, accuracy = 1.000
Epoch 4: mean loss = 0.002, accuracy = 1.000
```

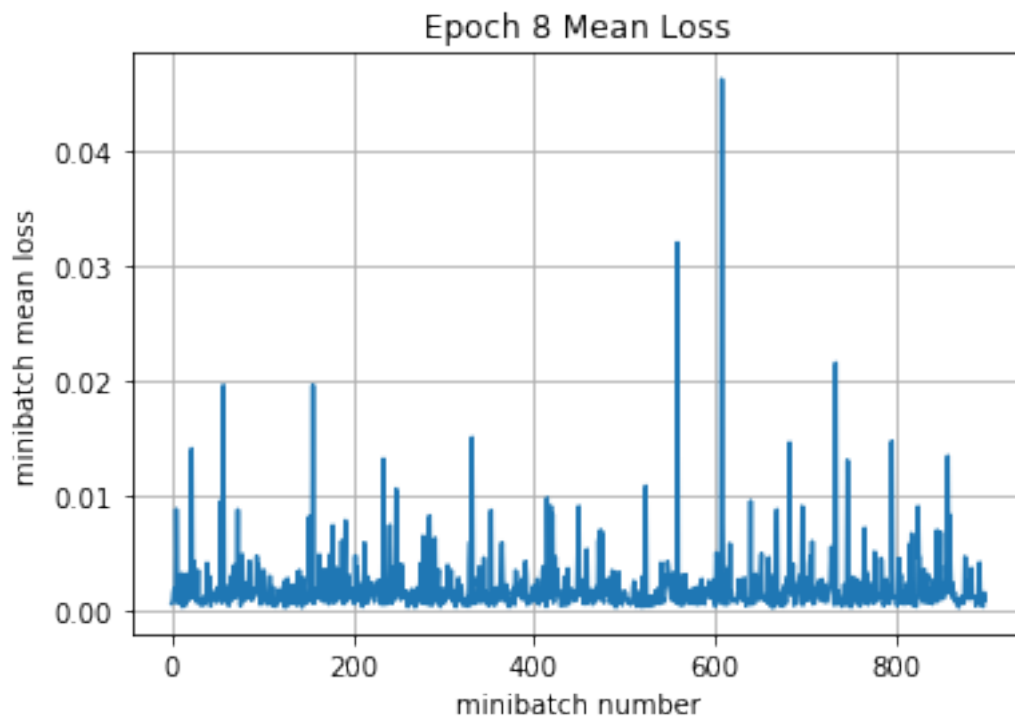
```
2017-10-27 02:42:48 Iter 4500: batch trn loss = 0.002, accuracy = 1.000
2017-10-27 02:43:08 Iter 4600: batch trn loss = 0.000, accuracy = 1.000
2017-10-27 02:43:28 Iter 4700: batch trn loss = 0.002, accuracy = 1.000
2017-10-27 02:43:48 Iter 4800: batch trn loss = 0.004, accuracy = 1.000
2017-10-27 02:44:09 Iter 4900: batch trn loss = 0.001, accuracy = 1.000
2017-10-27 02:44:29 Iter 5000: batch trn loss = 0.001, accuracy = 1.000
2017-10-27 02:44:50 Iter 5100: batch trn loss = 0.004, accuracy = 1.000
2017-10-27 02:45:10 Iter 5200: batch trn loss = 0.003, accuracy = 1.000
2017-10-27 02:45:30 Iter 5300: batch trn loss = 0.020, accuracy = 0.980
Epoch 5: mean loss = 0.002, accuracy = 1.000
```



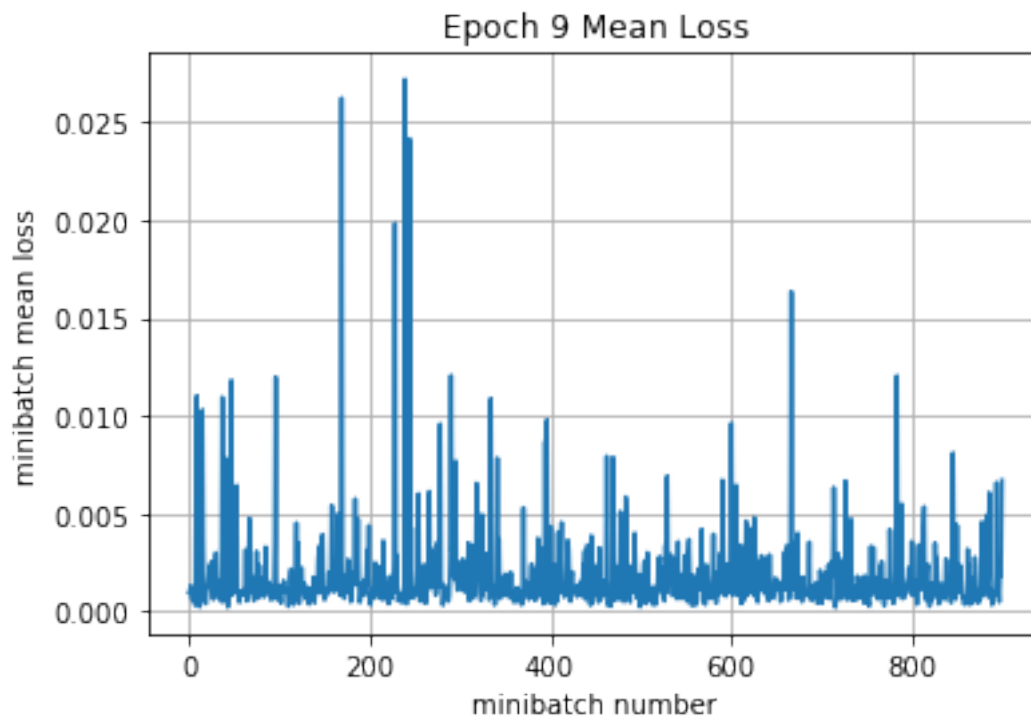
```
2017-10-27 02:45:50 Iter 5400: batch trn loss = 0.001, accuracy = 1.000
2017-10-27 02:46:11 Iter 5500: batch trn loss = 0.006, accuracy = 1.000
2017-10-27 02:46:31 Iter 5600: batch trn loss = 0.000, accuracy = 1.000
2017-10-27 02:46:51 Iter 5700: batch trn loss = 0.006, accuracy = 1.000
2017-10-27 02:47:11 Iter 5800: batch trn loss = 0.001, accuracy = 1.000
2017-10-27 02:47:32 Iter 5900: batch trn loss = 0.000, accuracy = 1.000
2017-10-27 02:47:52 Iter 6000: batch trn loss = 0.001, accuracy = 1.000
2017-10-27 02:48:12 Iter 6100: batch trn loss = 0.001, accuracy = 1.000
2017-10-27 02:48:32 Iter 6200: batch trn loss = 0.001, accuracy = 1.000
Epoch 6: mean loss = 0.002, accuracy = 1.000
```



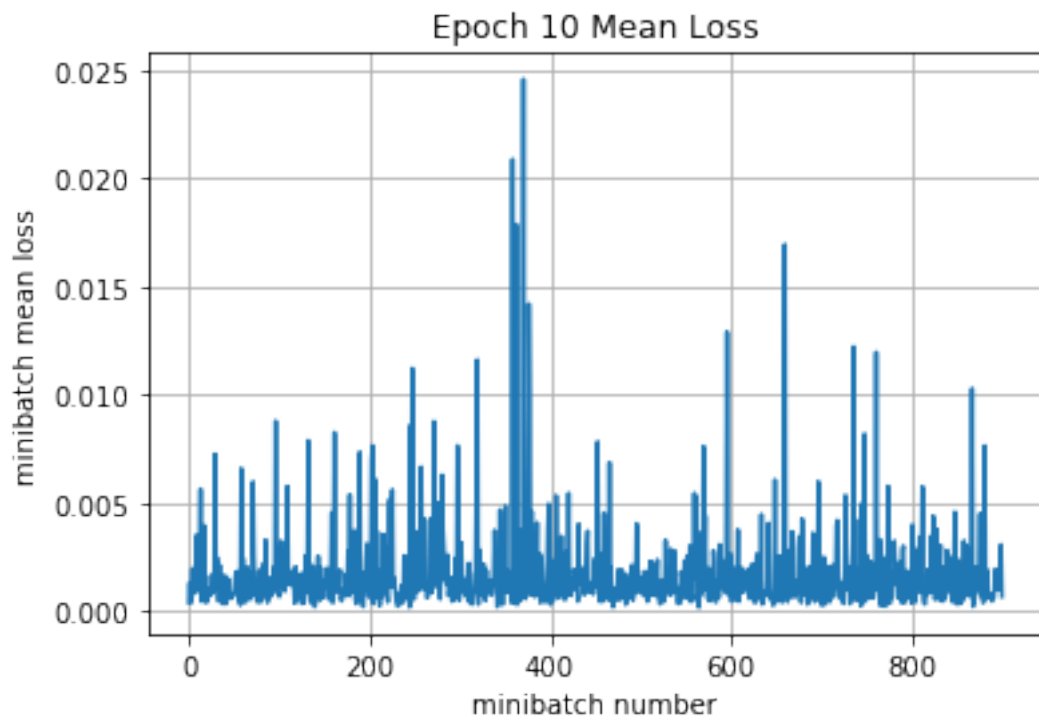
```
2017-10-27 02:48:53 Iter 6300: batch trn loss = 0.001, accuracy = 1.000
2017-10-27 02:49:13 Iter 6400: batch trn loss = 0.003, accuracy = 1.000
2017-10-27 02:49:33 Iter 6500: batch trn loss = 0.002, accuracy = 1.000
2017-10-27 02:49:54 Iter 6600: batch trn loss = 0.002, accuracy = 1.000
2017-10-27 02:50:14 Iter 6700: batch trn loss = 0.002, accuracy = 1.000
2017-10-27 02:50:34 Iter 6800: batch trn loss = 0.002, accuracy = 1.000
2017-10-27 02:50:55 Iter 6900: batch trn loss = 0.003, accuracy = 1.000
2017-10-27 02:51:15 Iter 7000: batch trn loss = 0.002, accuracy = 1.000
2017-10-27 02:51:35 Iter 7100: batch trn loss = 0.001, accuracy = 1.000
Epoch 7: mean loss = 0.002, accuracy = 1.000
```



```
2017-10-27 02:51:56 Iter 7200: batch trn loss = 0.001, accuracy = 1.000
2017-10-27 02:52:16 Iter 7300: batch trn loss = 0.001, accuracy = 1.000
2017-10-27 02:52:36 Iter 7400: batch trn loss = 0.001, accuracy = 1.000
2017-10-27 02:52:56 Iter 7500: batch trn loss = 0.002, accuracy = 1.000
2017-10-27 02:53:17 Iter 7600: batch trn loss = 0.004, accuracy = 1.000
2017-10-27 02:53:37 Iter 7700: batch trn loss = 0.001, accuracy = 1.000
2017-10-27 02:53:57 Iter 7800: batch trn loss = 0.010, accuracy = 1.000
2017-10-27 02:54:18 Iter 7900: batch trn loss = 0.000, accuracy = 1.000
2017-10-27 02:54:38 Iter 8000: batch trn loss = 0.001, accuracy = 1.000
Epoch 8: mean loss = 0.002, accuracy = 1.000
```



```
2017-10-27 02:54:58 Iter 8100: batch trn loss = 0.000, accuracy = 1.000
2017-10-27 02:55:19 Iter 8200: batch trn loss = 0.001, accuracy = 1.000
2017-10-27 02:55:39 Iter 8300: batch trn loss = 0.001, accuracy = 1.000
2017-10-27 02:55:59 Iter 8400: batch trn loss = 0.001, accuracy = 1.000
2017-10-27 02:56:19 Iter 8500: batch trn loss = 0.000, accuracy = 1.000
2017-10-27 02:56:40 Iter 8600: batch trn loss = 0.001, accuracy = 1.000
2017-10-27 02:57:00 Iter 8700: batch trn loss = 0.000, accuracy = 1.000
2017-10-27 02:57:20 Iter 8800: batch trn loss = 0.000, accuracy = 1.000
2017-10-27 02:57:40 Iter 8900: batch trn loss = 0.004, accuracy = 1.000
Epoch 9: mean loss = 0.002, accuracy = 1.000
```



validation

Epoch 0: mean loss = 1.313, accuracy = 0.746

Out[55]: (1.3129607, 0.746)

Appendix2

October 30, 2017

0.1 Model Improvement 2

change architecture(add one more batch normalization layer in the end before last dense layer) + regularization best, test accueancy = 0.777

```
In [17]: def normalization(raw_images):
    images = np.empty_like(raw_images)
    np.copyto(images, raw_images)
    for i in range(images.shape[0]):
        old = images[i]
        new = (old - np.mean(old)) / np.std(old)
        images[i, :, :, :] = new
    return images

In [19]: def model_fn(layer_input, labels, is_training):
    reg_scale= 0.1
    regularizer = tf.contrib.layers.l2_regularizer(scale=reg_scale)
    # initializer = tf.contrib.layers.xavier_initializer()
    initializer = None
    h, w = H, W
    # conv - bn - max_pool
    F1 = 64
    P1, S1 = 3, 2 # pool_size, strides
    layer = tf.layers.conv2d(
        inputs=layer_input, filters=F1, kernel_size=[5, 5], padding='same', activation=
        kernel_regularizer=regularizer, kernel_initializer=initializer)
    layer = tf.layers.batch_normalization(
        inputs=layer, training=is_training,
        beta_regularizer=regularizer, gamma_regularizer=regularizer)
    layer = tf.layers.max_pooling2d(inputs=layer, pool_size=P1, strides=S1)
    h = math.floor((h - P1 + S1) / S1)
    w = math.floor((w - P1 + S1) / S1)
    # conv - bn - max_pool
    F2 = 64
    P2, S2 = 3, 2 # pool_size, strides
    layer = tf.layers.conv2d(
        inputs=layer, filters=F2, kernel_size=[5, 5], padding='same', activation=tf.nn.
        kernel_regularizer=regularizer, kernel_initializer=initializer)
```

```

layer = tf.layers.batch_normalization(
    inputs=layer, training=is_training,
    beta_regularizer=regularizer, gamma_regularizer=regularizer)
layer = tf.layers.max_pooling2d(inputs=layer, pool_size=P2, strides=S2)
h = math.floor((h - P2 + S2) / S2)
w = math.floor((w - P2 + S2) / S2)
# dense1 - bn - dropout - fc - softmax
flat_size = F2 * h * w
print(flat_size)
layer = tf.reshape(layer, [-1, flat_size])
layer = tf.layers.dense(
    inputs=layer, units=1024, activation=tf.nn.relu,
    kernel_regularizer=regularizer, kernel_initializer=initializer)
layer = tf.layers.batch_normalization(
    inputs=layer, training=is_training,
    beta_regularizer=regularizer, gamma_regularizer=regularizer)
# no activation here for logit, as it will be calculated in loss
logit = tf.layers.dense(
    inputs=layer, units=T, activation=None,
    kernel_regularizer=regularizer, kernel_initializer=initializer)
onehot_labels = tf.one_hot(indices=tf.cast(labels, tf.int64), depth=10)
loss = tf.losses.softmax_cross_entropy(
    onehot_labels=onehot_labels, logits=logit)
return logit, loss

trainer = tf.train.GradientDescentOptimizer(learning_rate=0.03)
cnn = CNN(model_fn, trainer)
cnns['0102'] = cnn

m_X_trn = normalization(X_trn)
m_X_val = normalization(X_val)

```

3136

```

In [20]: # 5 epochs
print('train')
cnn.train(m_X_trn, y_trn, epochs=5, batch_size=100, print_every=100, plot_losses=False)
print('validation')
cnn.validate(m_X_val, y_val, epochs=1, batch_size=y_val.shape[0])

```

```

train
2017-10-29 11:20:30 Iter 0: batch trn loss = 3.250, accuracy = 0.070
2017-10-29 11:21:40 Iter 100: batch trn loss = 1.315, accuracy = 0.530
2017-10-29 11:22:51 Iter 200: batch trn loss = 1.346, accuracy = 0.510
2017-10-29 11:23:57 Iter 300: batch trn loss = 1.078, accuracy = 0.640
2017-10-29 11:25:05 Iter 400: batch trn loss = 1.171, accuracy = 0.620
Epoch 0: mean loss = 1.329, accuracy = 0.550

```



```

2017-10-29 11:26:12 Iter 500: batch trn loss = 0.779, accuracy = 0.720
2017-10-29 11:27:19 Iter 600: batch trn loss = 1.067, accuracy = 0.580
2017-10-29 11:28:26 Iter 700: batch trn loss = 0.834, accuracy = 0.690
2017-10-29 11:29:33 Iter 800: batch trn loss = 0.763, accuracy = 0.760
Epoch 1: mean loss = 0.843, accuracy = 0.710
2017-10-29 11:30:40 Iter 900: batch trn loss = 0.543, accuracy = 0.790
2017-10-29 11:31:47 Iter 1000: batch trn loss = 0.804, accuracy = 0.720
2017-10-29 11:32:54 Iter 1100: batch trn loss = 0.628, accuracy = 0.760
2017-10-29 11:34:02 Iter 1200: batch trn loss = 0.620, accuracy = 0.710
2017-10-29 11:35:07 Iter 1300: batch trn loss = 0.711, accuracy = 0.740
Epoch 2: mean loss = 0.662, accuracy = 0.773
2017-10-29 11:36:18 Iter 1400: batch trn loss = 0.481, accuracy = 0.860
2017-10-29 11:37:25 Iter 1500: batch trn loss = 0.495, accuracy = 0.800
2017-10-29 11:38:32 Iter 1600: batch trn loss = 0.645, accuracy = 0.790
2017-10-29 11:39:37 Iter 1700: batch trn loss = 0.549, accuracy = 0.820
Epoch 3: mean loss = 0.534, accuracy = 0.817
2017-10-29 11:40:42 Iter 1800: batch trn loss = 0.490, accuracy = 0.830
2017-10-29 11:41:47 Iter 1900: batch trn loss = 0.426, accuracy = 0.890
2017-10-29 11:42:58 Iter 2000: batch trn loss = 0.407, accuracy = 0.860
2017-10-29 11:44:04 Iter 2100: batch trn loss = 0.459, accuracy = 0.820
2017-10-29 11:45:09 Iter 2200: batch trn loss = 0.461, accuracy = 0.860
Epoch 4: mean loss = 0.419, accuracy = 0.859
validation
Epoch 0: mean loss = 0.811, accuracy = 0.726

```

```
Out[20]: (0.81093341, 0.72619999999999996)
```

```

In [21]: # 10 epochs
         print('train')
         cnn.train(m_X_trn, y_trn, epochs=5, batch_size=100, print_every=100, plot_losses=False)
         print('validation')
         cnn.validate(m_X_val, y_val, epochs=1, batch_size=y_val.shape[0])

```

```

train
2017-10-29 11:45:56 Iter 0: batch trn loss = 0.269, accuracy = 0.930
2017-10-29 11:47:04 Iter 100: batch trn loss = 0.330, accuracy = 0.920
2017-10-29 11:48:12 Iter 200: batch trn loss = 0.439, accuracy = 0.850
2017-10-29 11:49:19 Iter 300: batch trn loss = 0.350, accuracy = 0.880
2017-10-29 11:50:24 Iter 400: batch trn loss = 0.419, accuracy = 0.880
Epoch 0: mean loss = 0.317, accuracy = 0.897
2017-10-29 11:51:29 Iter 500: batch trn loss = 0.212, accuracy = 0.960
2017-10-29 11:52:35 Iter 600: batch trn loss = 0.160, accuracy = 0.970
2017-10-29 11:53:43 Iter 700: batch trn loss = 0.156, accuracy = 0.980
2017-10-29 11:54:50 Iter 800: batch trn loss = 0.225, accuracy = 0.920
Epoch 1: mean loss = 0.234, accuracy = 0.927
2017-10-29 11:55:56 Iter 900: batch trn loss = 0.127, accuracy = 0.980
2017-10-29 11:57:02 Iter 1000: batch trn loss = 0.171, accuracy = 0.940

```

```

2017-10-29 11:58:09 Iter 1100: batch trn loss = 0.124, accuracy = 0.980
2017-10-29 11:59:16 Iter 1200: batch trn loss = 0.311, accuracy = 0.920
2017-10-29 12:00:22 Iter 1300: batch trn loss = 0.221, accuracy = 0.950
Epoch 2: mean loss = 0.161, accuracy = 0.956
2017-10-29 12:01:30 Iter 1400: batch trn loss = 0.087, accuracy = 0.990
2017-10-29 12:02:40 Iter 1500: batch trn loss = 0.102, accuracy = 0.970
2017-10-29 12:03:54 Iter 1600: batch trn loss = 0.088, accuracy = 0.990
2017-10-29 12:05:07 Iter 1700: batch trn loss = 0.068, accuracy = 0.980
Epoch 3: mean loss = 0.101, accuracy = 0.979
2017-10-29 12:06:15 Iter 1800: batch trn loss = 0.063, accuracy = 0.990
2017-10-29 12:07:21 Iter 1900: batch trn loss = 0.051, accuracy = 0.990
2017-10-29 12:08:25 Iter 2000: batch trn loss = 0.053, accuracy = 0.990
2017-10-29 12:09:30 Iter 2100: batch trn loss = 0.069, accuracy = 1.000
2017-10-29 12:10:35 Iter 2200: batch trn loss = 0.084, accuracy = 0.980
Epoch 4: mean loss = 0.064, accuracy = 0.990
validation
Epoch 0: mean loss = 0.829, accuracy = 0.764

```

```
Out[21]: (0.82861483, 0.76400000000000001)
```

```

In [22]: # 15 epochs
         print('train')
         cnn.train(m_X_trn, y_trn, epochs=5, batch_size=100, print_every=100, plot_losses=False)
         print('validation')
         cnn.validate(m_X_val, y_val, epochs=1, batch_size=y_val.shape[0])

```

```

train
2017-10-29 12:11:20 Iter 0: batch trn loss = 0.027, accuracy = 1.000
2017-10-29 12:12:26 Iter 100: batch trn loss = 0.040, accuracy = 1.000
2017-10-29 12:13:30 Iter 200: batch trn loss = 0.061, accuracy = 0.990
2017-10-29 12:14:34 Iter 300: batch trn loss = 0.063, accuracy = 1.000
2017-10-29 12:15:37 Iter 400: batch trn loss = 0.023, accuracy = 1.000
Epoch 0: mean loss = 0.041, accuracy = 0.996
2017-10-29 12:16:41 Iter 500: batch trn loss = 0.040, accuracy = 0.990
2017-10-29 12:17:45 Iter 600: batch trn loss = 0.027, accuracy = 1.000
2017-10-29 12:18:51 Iter 700: batch trn loss = 0.025, accuracy = 1.000
2017-10-29 12:19:58 Iter 800: batch trn loss = 0.023, accuracy = 1.000
Epoch 1: mean loss = 0.028, accuracy = 0.998
2017-10-29 12:21:04 Iter 900: batch trn loss = 0.024, accuracy = 1.000
2017-10-29 12:22:11 Iter 1000: batch trn loss = 0.033, accuracy = 1.000
2017-10-29 12:23:18 Iter 1100: batch trn loss = 0.025, accuracy = 1.000
2017-10-29 12:24:25 Iter 1200: batch trn loss = 0.014, accuracy = 1.000
2017-10-29 12:25:32 Iter 1300: batch trn loss = 0.024, accuracy = 1.000
Epoch 2: mean loss = 0.019, accuracy = 1.000
2017-10-29 12:26:41 Iter 1400: batch trn loss = 0.011, accuracy = 1.000
2017-10-29 12:27:48 Iter 1500: batch trn loss = 0.017, accuracy = 1.000
2017-10-29 12:28:56 Iter 1600: batch trn loss = 0.012, accuracy = 1.000

```

```

2017-10-29 12:30:03 Iter 1700: batch trn loss = 0.011, accuracy = 1.000
Epoch 3: mean loss = 0.016, accuracy = 1.000
2017-10-29 12:31:07 Iter 1800: batch trn loss = 0.014, accuracy = 1.000
2017-10-29 12:32:12 Iter 1900: batch trn loss = 0.011, accuracy = 1.000
2017-10-29 12:33:17 Iter 2000: batch trn loss = 0.010, accuracy = 1.000
2017-10-29 12:34:22 Iter 2100: batch trn loss = 0.012, accuracy = 1.000
2017-10-29 12:35:26 Iter 2200: batch trn loss = 0.016, accuracy = 1.000
Epoch 4: mean loss = 0.012, accuracy = 1.000
validation
Epoch 0: mean loss = 0.857, accuracy = 0.777

```

```
Out[22]: (0.85706973, 0.777000000000000002)
```

1 Model improvement 3 (the best model is re-runned using this one)

change architecture(add one more batch normalization layer in the end before last dense layer) + regularization + dropout(0.5), best test accuracy: 0.8015

```

In [25]: def model_fn(layer_input, labels, is_training):
    reg_scale= 0.1
    regularizer = tf.contrib.layers.l2_regularizer(scale=reg_scale)
    # initializer = tf.contrib.layers.xavier_initializer()
    initializer = None
    h, w = H, W
    # conv - bn - max_pool
    F1 = 64
    P1, S1 = 3, 2 # pool_size, strides
    layer = tf.layers.conv2d(
        inputs=layer_input, filters=F1, kernel_size=[5, 5], padding='same', activation=
        kernel_regularizer=regularizer, kernel_initializer=initializer)
    layer = tf.layers.batch_normalization(
        inputs=layer, training=is_training,
        beta_regularizer=regularizer, gamma_regularizer=regularizer)
    layer = tf.layers.max_pooling2d(inputs=layer, pool_size=P1, strides=S1)
    h = math.floor((h - P1 + S1) / S1)
    w = math.floor((w - P1 + S1) / S1)
    # conv - bn - max_pool
    F2 = 64
    P2, S2 = 3, 2 # pool_size, strides
    layer = tf.layers.conv2d(
        inputs=layer, filters=F2, kernel_size=[5, 5], padding='same', activation=tf.nn.
        kernel_regularizer=regularizer, kernel_initializer=initializer)
    layer = tf.layers.batch_normalization(
        inputs=layer, training=is_training,
        beta_regularizer=regularizer, gamma_regularizer=regularizer)
    layer = tf.layers.max_pooling2d(inputs=layer, pool_size=P2, strides=S2)
    h = math.floor((h - P2 + S2) / S2)

```

```

w = math.floor((w - P2 + S2) / S2)
# dense1 - bn - dropout - fc - softmax
flat_size = F2 * h * w
print(flat_size)
layer = tf.reshape(layer, [-1, flat_size])
layer = tf.layers.dense(
    inputs=layer, units=1024, activation=tf.nn.relu,
    kernel_regularizer=regularizer, kernel_initializer=initializer)
layer = tf.layers.batch_normalization(
    inputs=layer, training=is_training,
    beta_regularizer=regularizer, gamma_regularizer=regularizer)
layer = tf.layers.dropout(inputs=layer, rate=0.5, training=is_training)
# no activation here for logit, as it will be calculated in loss
logit = tf.layers.dense(
    inputs=layer, units=T, activation=None,
    kernel_regularizer=regularizer, kernel_initializer=initializer)
onehot_labels = tf.one_hot(indices=tf.cast(labels, tf.int64), depth=10)
loss = tf.losses.softmax_cross_entropy(
    onehot_labels=onehot_labels, logits=logit)
return logit, loss

trainer = tf.train.GradientDescentOptimizer(learning_rate=0.03)
cnn = CNN(model_fn, trainer)
cnns['0103'] = cnn

m_X_trn = normalization(X_trn)
m_X_val = normalization(X_val)

```

3136

```

In [26]: # 5 epochs
print('train')
cnn.train(m_X_trn, y_trn, epochs=5, batch_size=50, print_every=100, plot_losses=False)
print('validation')
cnn.validate(m_X_val, y_val, epochs=1, batch_size=y_val.shape[0])

```

```

train
2017-10-29 12:55:58 Iter 0: batch trn loss = 3.171, accuracy = 0.120
2017-10-29 12:56:31 Iter 100: batch trn loss = 1.984, accuracy = 0.360
2017-10-29 12:57:05 Iter 200: batch trn loss = 1.572, accuracy = 0.360
2017-10-29 12:57:40 Iter 300: batch trn loss = 1.628, accuracy = 0.520
2017-10-29 12:58:12 Iter 400: batch trn loss = 1.474, accuracy = 0.480
2017-10-29 12:58:44 Iter 500: batch trn loss = 1.485, accuracy = 0.480
2017-10-29 12:59:17 Iter 600: batch trn loss = 1.381, accuracy = 0.580
2017-10-29 12:59:49 Iter 700: batch trn loss = 1.681, accuracy = 0.440
2017-10-29 13:00:21 Iter 800: batch trn loss = 0.820, accuracy = 0.700
Epoch 0: mean loss = 1.607, accuracy = 0.486

```

```

2017-10-29 13:00:53 Iter 900: batch trn loss = 1.120, accuracy = 0.600
2017-10-29 13:01:26 Iter 1000: batch trn loss = 1.241, accuracy = 0.640
2017-10-29 13:01:58 Iter 1100: batch trn loss = 1.265, accuracy = 0.520
2017-10-29 13:02:31 Iter 1200: batch trn loss = 0.929, accuracy = 0.700
2017-10-29 13:03:03 Iter 1300: batch trn loss = 1.136, accuracy = 0.680
2017-10-29 13:03:35 Iter 1400: batch trn loss = 1.025, accuracy = 0.680
2017-10-29 13:04:07 Iter 1500: batch trn loss = 0.905, accuracy = 0.620
2017-10-29 13:04:39 Iter 1600: batch trn loss = 0.780, accuracy = 0.720
2017-10-29 13:05:11 Iter 1700: batch trn loss = 1.112, accuracy = 0.640
Epoch 1: mean loss = 1.101, accuracy = 0.626
2017-10-29 13:05:43 Iter 1800: batch trn loss = 1.148, accuracy = 0.600
2017-10-29 13:06:15 Iter 1900: batch trn loss = 0.821, accuracy = 0.740
2017-10-29 13:06:47 Iter 2000: batch trn loss = 0.893, accuracy = 0.720
2017-10-29 13:07:19 Iter 2100: batch trn loss = 0.960, accuracy = 0.740
2017-10-29 13:07:51 Iter 2200: batch trn loss = 0.924, accuracy = 0.680
2017-10-29 13:08:25 Iter 2300: batch trn loss = 0.814, accuracy = 0.740
2017-10-29 13:09:03 Iter 2400: batch trn loss = 1.227, accuracy = 0.560
2017-10-29 13:09:40 Iter 2500: batch trn loss = 0.693, accuracy = 0.780
2017-10-29 13:10:29 Iter 2600: batch trn loss = 0.857, accuracy = 0.660
Epoch 2: mean loss = 0.939, accuracy = 0.681
2017-10-29 13:11:17 Iter 2700: batch trn loss = 0.619, accuracy = 0.800
2017-10-29 13:12:05 Iter 2800: batch trn loss = 0.767, accuracy = 0.800
2017-10-29 13:12:54 Iter 2900: batch trn loss = 0.646, accuracy = 0.800
2017-10-29 13:13:44 Iter 3000: batch trn loss = 0.709, accuracy = 0.760
2017-10-29 13:14:31 Iter 3100: batch trn loss = 0.939, accuracy = 0.720
2017-10-29 13:15:20 Iter 3200: batch trn loss = 0.743, accuracy = 0.680
2017-10-29 13:16:09 Iter 3300: batch trn loss = 0.939, accuracy = 0.720
2017-10-29 13:16:59 Iter 3400: batch trn loss = 0.552, accuracy = 0.820
2017-10-29 13:17:49 Iter 3500: batch trn loss = 1.058, accuracy = 0.700
Epoch 3: mean loss = 0.834, accuracy = 0.718
2017-10-29 13:18:39 Iter 3600: batch trn loss = 0.905, accuracy = 0.740
2017-10-29 13:19:29 Iter 3700: batch trn loss = 0.455, accuracy = 0.760
2017-10-29 13:20:20 Iter 3800: batch trn loss = 0.837, accuracy = 0.660
2017-10-29 13:21:11 Iter 3900: batch trn loss = 1.232, accuracy = 0.620
2017-10-29 13:22:00 Iter 4000: batch trn loss = 1.065, accuracy = 0.720
2017-10-29 13:22:50 Iter 4100: batch trn loss = 0.595, accuracy = 0.820
2017-10-29 13:23:39 Iter 4200: batch trn loss = 0.564, accuracy = 0.780
2017-10-29 13:24:29 Iter 4300: batch trn loss = 0.691, accuracy = 0.780
2017-10-29 13:25:33 Iter 4400: batch trn loss = 0.487, accuracy = 0.800
Epoch 4: mean loss = 0.747, accuracy = 0.743
validation
Epoch 0: mean loss = 0.773, accuracy = 0.740

```

```
Out[26]: (0.77256513, 0.73980000000000000001)
```

```
In [27]: # 10 epochs
         print('train')
```

```

cnn.train(m_X_trn, y_trn, epochs=5, batch_size=50, print_every=100, plot_losses=False)
print('validation')
cnn.validate(m_X_val, y_val, epochs=1, batch_size=y_val.shape[0])

```

train

```

2017-10-29 14:29:36 Iter 0: batch trn loss = 0.789, accuracy = 0.760
2017-10-29 14:30:11 Iter 100: batch trn loss = 0.295, accuracy = 0.900
2017-10-29 14:30:45 Iter 200: batch trn loss = 0.714, accuracy = 0.800
2017-10-29 14:31:18 Iter 300: batch trn loss = 0.900, accuracy = 0.680
2017-10-29 14:31:51 Iter 400: batch trn loss = 0.469, accuracy = 0.860
2017-10-29 14:32:26 Iter 500: batch trn loss = 0.980, accuracy = 0.720
2017-10-29 14:32:58 Iter 600: batch trn loss = 1.012, accuracy = 0.740
2017-10-29 14:33:31 Iter 700: batch trn loss = 0.525, accuracy = 0.820
2017-10-29 14:34:03 Iter 800: batch trn loss = 0.737, accuracy = 0.740
Epoch 0: mean loss = 0.683, accuracy = 0.768
2017-10-29 14:34:35 Iter 900: batch trn loss = 0.579, accuracy = 0.820
2017-10-29 14:35:08 Iter 1000: batch trn loss = 0.526, accuracy = 0.740
2017-10-29 14:35:40 Iter 1100: batch trn loss = 0.613, accuracy = 0.820
2017-10-29 14:36:12 Iter 1200: batch trn loss = 0.534, accuracy = 0.760
2017-10-29 14:36:44 Iter 1300: batch trn loss = 0.537, accuracy = 0.800
2017-10-29 14:37:17 Iter 1400: batch trn loss = 0.630, accuracy = 0.820
2017-10-29 14:37:49 Iter 1500: batch trn loss = 0.486, accuracy = 0.820
2017-10-29 14:38:21 Iter 1600: batch trn loss = 0.735, accuracy = 0.740
2017-10-29 14:38:53 Iter 1700: batch trn loss = 0.862, accuracy = 0.660
Epoch 1: mean loss = 0.627, accuracy = 0.782
2017-10-29 14:39:26 Iter 1800: batch trn loss = 0.766, accuracy = 0.760
2017-10-29 14:39:58 Iter 1900: batch trn loss = 0.464, accuracy = 0.820
2017-10-29 14:40:30 Iter 2000: batch trn loss = 0.671, accuracy = 0.780
2017-10-29 14:41:03 Iter 2100: batch trn loss = 0.487, accuracy = 0.800
2017-10-29 14:41:35 Iter 2200: batch trn loss = 0.505, accuracy = 0.820
2017-10-29 14:42:08 Iter 2300: batch trn loss = 0.281, accuracy = 0.940
2017-10-29 14:42:41 Iter 2400: batch trn loss = 0.542, accuracy = 0.840
2017-10-29 14:43:13 Iter 2500: batch trn loss = 0.531, accuracy = 0.820
2017-10-29 14:43:45 Iter 2600: batch trn loss = 0.732, accuracy = 0.700
Epoch 2: mean loss = 0.566, accuracy = 0.803
2017-10-29 14:44:17 Iter 2700: batch trn loss = 0.552, accuracy = 0.800
2017-10-29 14:44:49 Iter 2800: batch trn loss = 0.589, accuracy = 0.820
2017-10-29 14:45:20 Iter 2900: batch trn loss = 0.788, accuracy = 0.780
2017-10-29 14:45:52 Iter 3000: batch trn loss = 0.477, accuracy = 0.740
2017-10-29 14:46:24 Iter 3100: batch trn loss = 0.438, accuracy = 0.860
2017-10-29 14:46:56 Iter 3200: batch trn loss = 0.456, accuracy = 0.840
2017-10-29 14:47:28 Iter 3300: batch trn loss = 0.791, accuracy = 0.780
2017-10-29 14:48:00 Iter 3400: batch trn loss = 0.687, accuracy = 0.820
2017-10-29 14:48:32 Iter 3500: batch trn loss = 0.512, accuracy = 0.780
Epoch 3: mean loss = 0.523, accuracy = 0.818
2017-10-29 14:49:04 Iter 3600: batch trn loss = 0.430, accuracy = 0.880
2017-10-29 14:49:36 Iter 3700: batch trn loss = 0.736, accuracy = 0.780
2017-10-29 14:50:08 Iter 3800: batch trn loss = 0.442, accuracy = 0.840

```

```

2017-10-29 14:50:40 Iter 3900: batch trn loss = 0.485, accuracy = 0.840
2017-10-29 14:51:12 Iter 4000: batch trn loss = 0.614, accuracy = 0.780
2017-10-29 14:51:44 Iter 4100: batch trn loss = 0.406, accuracy = 0.900
2017-10-29 14:52:16 Iter 4200: batch trn loss = 0.443, accuracy = 0.840
2017-10-29 14:52:48 Iter 4300: batch trn loss = 0.556, accuracy = 0.800
2017-10-29 14:53:20 Iter 4400: batch trn loss = 0.636, accuracy = 0.740
Epoch 4: mean loss = 0.483, accuracy = 0.831
validation
Epoch 0: mean loss = 0.700, accuracy = 0.774

```

```
Out[27]: (0.69985479, 0.77359999999999995)
```

```

In [28]: # 15 epochs
print('train')
cnn.train(m_X_trn, y_trn, epochs=5, batch_size=50, print_every=100, plot_losses=False)
print('validation')
cnn.validate(m_X_val, y_val, epochs=1, batch_size=y_val.shape[0])

```

```

train
2017-10-29 14:54:04 Iter 0: batch trn loss = 0.442, accuracy = 0.840
2017-10-29 14:54:36 Iter 100: batch trn loss = 0.553, accuracy = 0.860
2017-10-29 14:55:08 Iter 200: batch trn loss = 0.305, accuracy = 0.920
2017-10-29 14:55:40 Iter 300: batch trn loss = 0.427, accuracy = 0.880
2017-10-29 14:56:12 Iter 400: batch trn loss = 0.415, accuracy = 0.880
2017-10-29 14:56:44 Iter 500: batch trn loss = 0.399, accuracy = 0.820
2017-10-29 14:57:16 Iter 600: batch trn loss = 0.434, accuracy = 0.880
2017-10-29 14:57:48 Iter 700: batch trn loss = 0.358, accuracy = 0.860
2017-10-29 14:58:20 Iter 800: batch trn loss = 0.412, accuracy = 0.860
Epoch 0: mean loss = 0.441, accuracy = 0.845
2017-10-29 14:58:52 Iter 900: batch trn loss = 0.365, accuracy = 0.880
2017-10-29 14:59:24 Iter 1000: batch trn loss = 0.640, accuracy = 0.800
2017-10-29 14:59:56 Iter 1100: batch trn loss = 0.383, accuracy = 0.900
2017-10-29 15:00:28 Iter 1200: batch trn loss = 0.453, accuracy = 0.840
2017-10-29 15:01:00 Iter 1300: batch trn loss = 0.578, accuracy = 0.820
2017-10-29 15:01:32 Iter 1400: batch trn loss = 0.417, accuracy = 0.860
2017-10-29 15:02:04 Iter 1500: batch trn loss = 0.381, accuracy = 0.880
2017-10-29 15:02:36 Iter 1600: batch trn loss = 0.271, accuracy = 0.900
2017-10-29 15:03:08 Iter 1700: batch trn loss = 0.327, accuracy = 0.880
Epoch 1: mean loss = 0.410, accuracy = 0.856
2017-10-29 15:03:39 Iter 1800: batch trn loss = 0.351, accuracy = 0.880
2017-10-29 15:04:11 Iter 1900: batch trn loss = 0.176, accuracy = 0.920
2017-10-29 15:04:43 Iter 2000: batch trn loss = 0.207, accuracy = 0.940
2017-10-29 15:05:15 Iter 2100: batch trn loss = 0.158, accuracy = 0.940
2017-10-29 15:05:47 Iter 2200: batch trn loss = 0.409, accuracy = 0.800
2017-10-29 15:06:19 Iter 2300: batch trn loss = 0.423, accuracy = 0.860
2017-10-29 15:06:51 Iter 2400: batch trn loss = 0.727, accuracy = 0.760
2017-10-29 15:07:23 Iter 2500: batch trn loss = 0.308, accuracy = 0.920

```



```

2017-10-29 15:07:55 Iter 2600: batch trn loss = 0.263, accuracy = 0.920
Epoch 2: mean loss = 0.373, accuracy = 0.868
2017-10-29 15:08:27 Iter 2700: batch trn loss = 0.286, accuracy = 0.880
2017-10-29 15:08:59 Iter 2800: batch trn loss = 0.620, accuracy = 0.840
2017-10-29 15:09:31 Iter 2900: batch trn loss = 0.270, accuracy = 0.920
2017-10-29 15:10:03 Iter 3000: batch trn loss = 0.331, accuracy = 0.900
2017-10-29 15:10:35 Iter 3100: batch trn loss = 0.519, accuracy = 0.840
2017-10-29 15:11:07 Iter 3200: batch trn loss = 0.365, accuracy = 0.900
2017-10-29 15:11:39 Iter 3300: batch trn loss = 0.264, accuracy = 0.880
2017-10-29 15:12:11 Iter 3400: batch trn loss = 0.233, accuracy = 0.940
2017-10-29 15:12:47 Iter 3500: batch trn loss = 0.385, accuracy = 0.860
Epoch 3: mean loss = 0.347, accuracy = 0.877
2017-10-29 15:13:19 Iter 3600: batch trn loss = 0.321, accuracy = 0.860
2017-10-29 15:13:51 Iter 3700: batch trn loss = 0.235, accuracy = 0.920
2017-10-29 15:14:22 Iter 3800: batch trn loss = 0.418, accuracy = 0.860
2017-10-29 15:14:54 Iter 3900: batch trn loss = 0.267, accuracy = 0.900
2017-10-29 15:15:26 Iter 4000: batch trn loss = 0.276, accuracy = 0.940
2017-10-29 15:15:58 Iter 4100: batch trn loss = 0.321, accuracy = 0.920
2017-10-29 15:16:30 Iter 4200: batch trn loss = 0.156, accuracy = 0.960
2017-10-29 15:17:02 Iter 4300: batch trn loss = 0.345, accuracy = 0.880
2017-10-29 15:17:34 Iter 4400: batch trn loss = 0.236, accuracy = 0.880
Epoch 4: mean loss = 0.315, accuracy = 0.889
validation
Epoch 0: mean loss = 0.706, accuracy = 0.794

```

```
Out[28]: (0.70572072, 0.79359999999999997)
```

```

In [29]: # 20 epochs
print('train')
cnn.train(m_X_trn, y_trn, epochs=5, batch_size=50, print_every=100, plot_losses=False)
print('validation')
cnn.validate(m_X_val, y_val, epochs=1, batch_size=y_val.shape[0])

```

```

train
2017-10-29 15:53:00 Iter 0: batch trn loss = 0.115, accuracy = 0.980
2017-10-29 15:53:34 Iter 100: batch trn loss = 0.162, accuracy = 0.980
2017-10-29 15:54:08 Iter 200: batch trn loss = 0.306, accuracy = 0.880
2017-10-29 15:54:41 Iter 300: batch trn loss = 0.385, accuracy = 0.840
2017-10-29 15:55:15 Iter 400: batch trn loss = 0.320, accuracy = 0.860
2017-10-29 15:55:52 Iter 500: batch trn loss = 0.191, accuracy = 0.940
2017-10-29 15:56:27 Iter 600: batch trn loss = 0.373, accuracy = 0.920
2017-10-29 15:57:00 Iter 700: batch trn loss = 0.351, accuracy = 0.880
2017-10-29 15:57:33 Iter 800: batch trn loss = 0.512, accuracy = 0.860
Epoch 0: mean loss = 0.290, accuracy = 0.896
2017-10-29 15:58:06 Iter 900: batch trn loss = 0.363, accuracy = 0.820
2017-10-29 15:58:40 Iter 1000: batch trn loss = 0.248, accuracy = 0.840
2017-10-29 15:59:13 Iter 1100: batch trn loss = 0.346, accuracy = 0.860

```



```

2017-10-29 15:59:47 Iter 1200: batch trn loss = 0.129, accuracy = 0.940
2017-10-29 16:00:21 Iter 1300: batch trn loss = 0.277, accuracy = 0.900
2017-10-29 16:00:54 Iter 1400: batch trn loss = 0.193, accuracy = 0.960
2017-10-29 16:01:29 Iter 1500: batch trn loss = 0.228, accuracy = 0.880
2017-10-29 16:02:03 Iter 1600: batch trn loss = 0.327, accuracy = 0.860
2017-10-29 16:02:38 Iter 1700: batch trn loss = 0.397, accuracy = 0.920
Epoch 1: mean loss = 0.269, accuracy = 0.904
2017-10-29 16:03:11 Iter 1800: batch trn loss = 0.218, accuracy = 0.920
2017-10-29 16:03:45 Iter 1900: batch trn loss = 0.266, accuracy = 0.900
2017-10-29 16:04:20 Iter 2000: batch trn loss = 0.149, accuracy = 0.980
2017-10-29 16:04:54 Iter 2100: batch trn loss = 0.098, accuracy = 0.980
2017-10-29 16:05:28 Iter 2200: batch trn loss = 0.083, accuracy = 0.980
2017-10-29 16:06:01 Iter 2300: batch trn loss = 0.331, accuracy = 0.900
2017-10-29 16:06:36 Iter 2400: batch trn loss = 0.197, accuracy = 0.940
2017-10-29 16:07:12 Iter 2500: batch trn loss = 0.233, accuracy = 0.860
2017-10-29 16:07:45 Iter 2600: batch trn loss = 0.373, accuracy = 0.840
Epoch 2: mean loss = 0.249, accuracy = 0.910
2017-10-29 16:08:18 Iter 2700: batch trn loss = 0.108, accuracy = 0.960
2017-10-29 16:08:51 Iter 2800: batch trn loss = 0.136, accuracy = 0.940
2017-10-29 16:09:25 Iter 2900: batch trn loss = 0.278, accuracy = 0.920
2017-10-29 16:09:59 Iter 3000: batch trn loss = 0.171, accuracy = 0.940
2017-10-29 16:10:31 Iter 3100: batch trn loss = 0.142, accuracy = 0.940
2017-10-29 16:11:06 Iter 3200: batch trn loss = 0.203, accuracy = 0.920
2017-10-29 16:11:39 Iter 3300: batch trn loss = 0.229, accuracy = 0.920
2017-10-29 16:12:13 Iter 3400: batch trn loss = 0.291, accuracy = 0.900
2017-10-29 16:12:49 Iter 3500: batch trn loss = 0.492, accuracy = 0.880
Epoch 3: mean loss = 0.225, accuracy = 0.919
2017-10-29 16:13:24 Iter 3600: batch trn loss = 0.078, accuracy = 0.980
2017-10-29 16:13:58 Iter 3700: batch trn loss = 0.162, accuracy = 0.940
2017-10-29 16:14:31 Iter 3800: batch trn loss = 0.182, accuracy = 0.940
2017-10-29 16:15:04 Iter 3900: batch trn loss = 0.540, accuracy = 0.780
2017-10-29 16:15:45 Iter 4000: batch trn loss = 0.273, accuracy = 0.880
2017-10-29 16:16:19 Iter 4100: batch trn loss = 0.240, accuracy = 0.920
2017-10-29 16:16:55 Iter 4200: batch trn loss = 0.308, accuracy = 0.900
2017-10-29 16:17:28 Iter 4300: batch trn loss = 0.147, accuracy = 0.940
2017-10-29 16:18:01 Iter 4400: batch trn loss = 0.232, accuracy = 0.900
Epoch 4: mean loss = 0.213, accuracy = 0.924
validation
Epoch 0: mean loss = 0.755, accuracy = 0.802

```

```
Out[29]: (0.7545625, 0.802200000000000002)
```

```

In [30]: # 25 epochs
         print('train')
         cnn.train(m_X_trn, y_trn, epochs=5, batch_size=50, print_every=100, plot_losses=False)
         print('validation')
         cnn.validate(m_X_val, y_val, epochs=1, batch_size=y_val.shape[0])

```

train

2017-10-29 16:18:47 Iter 0: batch trn loss = 0.268, accuracy = 0.920
2017-10-29 16:19:21 Iter 100: batch trn loss = 0.071, accuracy = 0.960
2017-10-29 16:19:54 Iter 200: batch trn loss = 0.121, accuracy = 0.940
2017-10-29 16:20:27 Iter 300: batch trn loss = 0.147, accuracy = 0.960
2017-10-29 16:21:00 Iter 400: batch trn loss = 0.162, accuracy = 0.940
2017-10-29 16:21:34 Iter 500: batch trn loss = 0.143, accuracy = 0.960
2017-10-29 16:22:08 Iter 600: batch trn loss = 0.077, accuracy = 0.980
2017-10-29 16:22:44 Iter 700: batch trn loss = 0.201, accuracy = 0.960
2017-10-29 16:23:19 Iter 800: batch trn loss = 0.207, accuracy = 0.900
Epoch 0: mean loss = 0.203, accuracy = 0.927
2017-10-29 16:23:52 Iter 900: batch trn loss = 0.252, accuracy = 0.940
2017-10-29 16:24:26 Iter 1000: batch trn loss = 0.190, accuracy = 0.940
2017-10-29 16:24:59 Iter 1100: batch trn loss = 0.207, accuracy = 0.940
2017-10-29 16:25:33 Iter 1200: batch trn loss = 0.176, accuracy = 0.900
2017-10-29 16:26:06 Iter 1300: batch trn loss = 0.109, accuracy = 0.960
2017-10-29 16:26:40 Iter 1400: batch trn loss = 0.208, accuracy = 0.940
2017-10-29 16:27:13 Iter 1500: batch trn loss = 0.095, accuracy = 1.000
2017-10-29 16:27:47 Iter 1600: batch trn loss = 0.282, accuracy = 0.900
2017-10-29 16:28:20 Iter 1700: batch trn loss = 0.217, accuracy = 0.940
Epoch 1: mean loss = 0.186, accuracy = 0.933
2017-10-29 16:28:55 Iter 1800: batch trn loss = 0.186, accuracy = 0.920
2017-10-29 16:29:30 Iter 1900: batch trn loss = 0.119, accuracy = 0.960
2017-10-29 16:30:05 Iter 2000: batch trn loss = 0.075, accuracy = 1.000
2017-10-29 16:30:39 Iter 2100: batch trn loss = 0.141, accuracy = 0.940
2017-10-29 16:31:12 Iter 2200: batch trn loss = 0.069, accuracy = 0.960
2017-10-29 16:31:46 Iter 2300: batch trn loss = 0.225, accuracy = 0.920
2017-10-29 16:32:19 Iter 2400: batch trn loss = 0.187, accuracy = 0.940
2017-10-29 16:32:53 Iter 2500: batch trn loss = 0.125, accuracy = 0.960
2017-10-29 16:33:26 Iter 2600: batch trn loss = 0.143, accuracy = 0.960
Epoch 2: mean loss = 0.174, accuracy = 0.939
2017-10-29 16:33:59 Iter 2700: batch trn loss = 0.086, accuracy = 0.980
2017-10-29 16:34:35 Iter 2800: batch trn loss = 0.200, accuracy = 0.960
2017-10-29 16:35:08 Iter 2900: batch trn loss = 0.304, accuracy = 0.920
2017-10-29 16:35:40 Iter 3000: batch trn loss = 0.074, accuracy = 0.960
2017-10-29 16:36:13 Iter 3100: batch trn loss = 0.083, accuracy = 0.980
2017-10-29 16:36:46 Iter 3200: batch trn loss = 0.296, accuracy = 0.920
2017-10-29 16:37:20 Iter 3300: batch trn loss = 0.300, accuracy = 0.900
2017-10-29 16:37:54 Iter 3400: batch trn loss = 0.180, accuracy = 0.920
2017-10-29 16:38:28 Iter 3500: batch trn loss = 0.168, accuracy = 0.920
Epoch 3: mean loss = 0.171, accuracy = 0.939
2017-10-29 16:39:02 Iter 3600: batch trn loss = 0.128, accuracy = 0.940
2017-10-29 16:39:36 Iter 3700: batch trn loss = 0.217, accuracy = 0.940
2017-10-29 16:40:09 Iter 3800: batch trn loss = 0.250, accuracy = 0.900
2017-10-29 16:40:44 Iter 3900: batch trn loss = 0.143, accuracy = 0.920
2017-10-29 16:41:17 Iter 4000: batch trn loss = 0.202, accuracy = 0.920
2017-10-29 16:41:50 Iter 4100: batch trn loss = 0.286, accuracy = 0.860
2017-10-29 16:42:23 Iter 4200: batch trn loss = 0.240, accuracy = 0.920

```
2017-10-29 16:42:57 Iter 4300: batch trn loss = 0.246, accuracy = 0.900
2017-10-29 16:43:29 Iter 4400: batch trn loss = 0.080, accuracy = 1.000
Epoch 4: mean loss = 0.162, accuracy = 0.942
validation
Epoch 0: mean loss = 0.793, accuracy = 0.797
```

```
Out[30]: (0.79273933, 0.79679999999999995)
```

```
In [31]: # 30 epochs
print('train')
cnn.train(m_X_trn, y_trn, epochs=5, batch_size=50, print_every=100, plot_losses=False)
print('validation')
cnn.validate(m_X_val, y_val, epochs=1, batch_size=y_val.shape[0])
```

```
train
2017-10-29 17:17:07 Iter 0: batch trn loss = 0.086, accuracy = 0.980
2017-10-29 17:17:40 Iter 100: batch trn loss = 0.197, accuracy = 0.940
2017-10-29 17:18:13 Iter 200: batch trn loss = 0.107, accuracy = 0.980
2017-10-29 17:18:45 Iter 300: batch trn loss = 0.257, accuracy = 0.860
2017-10-29 17:19:18 Iter 400: batch trn loss = 0.142, accuracy = 0.960
2017-10-29 17:19:50 Iter 500: batch trn loss = 0.284, accuracy = 0.900
2017-10-29 17:20:24 Iter 600: batch trn loss = 0.243, accuracy = 0.920
2017-10-29 17:20:57 Iter 700: batch trn loss = 0.089, accuracy = 0.980
2017-10-29 17:21:31 Iter 800: batch trn loss = 0.206, accuracy = 0.920
Epoch 0: mean loss = 0.147, accuracy = 0.947
2017-10-29 17:22:04 Iter 900: batch trn loss = 0.014, accuracy = 1.000
2017-10-29 17:22:37 Iter 1000: batch trn loss = 0.042, accuracy = 1.000
2017-10-29 17:23:10 Iter 1100: batch trn loss = 0.062, accuracy = 0.980
2017-10-29 17:23:43 Iter 1200: batch trn loss = 0.112, accuracy = 0.960
2017-10-29 17:24:16 Iter 1300: batch trn loss = 0.042, accuracy = 1.000
2017-10-29 17:24:49 Iter 1400: batch trn loss = 0.182, accuracy = 0.940
2017-10-29 17:25:22 Iter 1500: batch trn loss = 0.165, accuracy = 0.920
2017-10-29 17:25:56 Iter 1600: batch trn loss = 0.104, accuracy = 0.980
2017-10-29 17:26:29 Iter 1700: batch trn loss = 0.125, accuracy = 0.960
Epoch 1: mean loss = 0.137, accuracy = 0.950
2017-10-29 17:27:02 Iter 1800: batch trn loss = 0.056, accuracy = 1.000
2017-10-29 17:27:34 Iter 1900: batch trn loss = 0.079, accuracy = 0.940
2017-10-29 17:28:07 Iter 2000: batch trn loss = 0.154, accuracy = 0.920
2017-10-29 17:28:41 Iter 2100: batch trn loss = 0.048, accuracy = 0.980
2017-10-29 17:29:14 Iter 2200: batch trn loss = 0.232, accuracy = 0.940
2017-10-29 17:29:48 Iter 2300: batch trn loss = 0.036, accuracy = 1.000
2017-10-29 17:30:22 Iter 2400: batch trn loss = 0.179, accuracy = 0.920
2017-10-29 17:30:56 Iter 2500: batch trn loss = 0.199, accuracy = 0.920
2017-10-29 17:31:30 Iter 2600: batch trn loss = 0.378, accuracy = 0.860
Epoch 2: mean loss = 0.120, accuracy = 0.958
2017-10-29 17:32:06 Iter 2700: batch trn loss = 0.099, accuracy = 0.980
2017-10-29 17:32:40 Iter 2800: batch trn loss = 0.144, accuracy = 0.960
```

```

2017-10-29 17:33:13 Iter 2900: batch trn loss = 0.380, accuracy = 0.900
2017-10-29 17:33:46 Iter 3000: batch trn loss = 0.046, accuracy = 1.000
2017-10-29 17:34:20 Iter 3100: batch trn loss = 0.103, accuracy = 0.960
2017-10-29 17:34:54 Iter 3200: batch trn loss = 0.337, accuracy = 0.900
2017-10-29 17:35:28 Iter 3300: batch trn loss = 0.079, accuracy = 0.960
2017-10-29 17:36:02 Iter 3400: batch trn loss = 0.144, accuracy = 0.980
2017-10-29 17:36:35 Iter 3500: batch trn loss = 0.028, accuracy = 1.000
Epoch 3: mean loss = 0.113, accuracy = 0.960
2017-10-29 17:37:10 Iter 3600: batch trn loss = 0.210, accuracy = 0.920
2017-10-29 17:37:43 Iter 3700: batch trn loss = 0.081, accuracy = 0.960
2017-10-29 17:38:16 Iter 3800: batch trn loss = 0.109, accuracy = 0.960
2017-10-29 17:38:49 Iter 3900: batch trn loss = 0.033, accuracy = 1.000
2017-10-29 17:39:21 Iter 4000: batch trn loss = 0.030, accuracy = 0.980
2017-10-29 17:39:55 Iter 4100: batch trn loss = 0.320, accuracy = 0.900
2017-10-29 17:40:29 Iter 4200: batch trn loss = 0.098, accuracy = 0.940
2017-10-29 17:41:01 Iter 4300: batch trn loss = 0.211, accuracy = 0.940
2017-10-29 17:41:34 Iter 4400: batch trn loss = 0.078, accuracy = 1.000
Epoch 4: mean loss = 0.113, accuracy = 0.960
validation
Epoch 0: mean loss = 0.870, accuracy = 0.802

```

```
Out[31]: (0.86955673, 0.8015999999999999)
```

```

In [32]: # 35 epochs
         print('train')
         cnn.train(m_X_trn, y_trn, epochs=5, batch_size=50, print_every=100, plot_losses=False)
         print('validation')
         cnn.validate(m_X_val, y_val, epochs=1, batch_size=y_val.shape[0])

```

```

train
2017-10-29 18:43:59 Iter 0: batch trn loss = 0.073, accuracy = 0.980
2017-10-29 18:44:31 Iter 100: batch trn loss = 0.135, accuracy = 0.960
2017-10-29 18:45:03 Iter 200: batch trn loss = 0.019, accuracy = 1.000
2017-10-29 18:45:36 Iter 300: batch trn loss = 0.039, accuracy = 0.980
2017-10-29 18:46:08 Iter 400: batch trn loss = 0.072, accuracy = 0.980
2017-10-29 18:46:40 Iter 500: batch trn loss = 0.268, accuracy = 0.900
2017-10-29 18:47:12 Iter 600: batch trn loss = 0.011, accuracy = 1.000
2017-10-29 18:47:44 Iter 700: batch trn loss = 0.275, accuracy = 0.820
2017-10-29 18:48:17 Iter 800: batch trn loss = 0.061, accuracy = 0.980
Epoch 0: mean loss = 0.105, accuracy = 0.962
2017-10-29 18:48:49 Iter 900: batch trn loss = 0.059, accuracy = 0.980
2017-10-29 18:49:21 Iter 1000: batch trn loss = 0.180, accuracy = 0.900
2017-10-29 18:49:53 Iter 1100: batch trn loss = 0.089, accuracy = 0.960
2017-10-29 18:50:25 Iter 1200: batch trn loss = 0.121, accuracy = 0.940
2017-10-29 18:50:57 Iter 1300: batch trn loss = 0.042, accuracy = 0.980
2017-10-29 18:51:29 Iter 1400: batch trn loss = 0.108, accuracy = 0.960
2017-10-29 18:52:01 Iter 1500: batch trn loss = 0.130, accuracy = 0.960

```

```

2017-10-29 18:52:32 Iter 1600: batch trn loss = 0.224, accuracy = 0.920
2017-10-29 18:53:04 Iter 1700: batch trn loss = 0.167, accuracy = 0.920
Epoch 1: mean loss = 0.101, accuracy = 0.963
2017-10-29 18:53:36 Iter 1800: batch trn loss = 0.076, accuracy = 0.980
2017-10-29 18:54:08 Iter 1900: batch trn loss = 0.044, accuracy = 0.980
2017-10-29 18:54:40 Iter 2000: batch trn loss = 0.108, accuracy = 0.980
2017-10-29 18:55:12 Iter 2100: batch trn loss = 0.016, accuracy = 1.000
2017-10-29 18:55:44 Iter 2200: batch trn loss = 0.094, accuracy = 0.960
2017-10-29 18:56:16 Iter 2300: batch trn loss = 0.130, accuracy = 0.960
2017-10-29 18:56:48 Iter 2400: batch trn loss = 0.034, accuracy = 1.000
2017-10-29 18:57:20 Iter 2500: batch trn loss = 0.229, accuracy = 0.900
2017-10-29 18:57:52 Iter 2600: batch trn loss = 0.243, accuracy = 0.920
Epoch 2: mean loss = 0.095, accuracy = 0.967
2017-10-29 18:58:24 Iter 2700: batch trn loss = 0.167, accuracy = 0.960
2017-10-29 18:58:55 Iter 2800: batch trn loss = 0.119, accuracy = 0.940
2017-10-29 18:59:27 Iter 2900: batch trn loss = 0.032, accuracy = 1.000
2017-10-29 18:59:59 Iter 3000: batch trn loss = 0.056, accuracy = 0.980
2017-10-29 19:00:31 Iter 3100: batch trn loss = 0.096, accuracy = 0.980
2017-10-29 19:01:03 Iter 3200: batch trn loss = 0.059, accuracy = 0.960
2017-10-29 19:01:35 Iter 3300: batch trn loss = 0.062, accuracy = 1.000
2017-10-29 19:02:07 Iter 3400: batch trn loss = 0.237, accuracy = 0.920
2017-10-29 19:02:38 Iter 3500: batch trn loss = 0.114, accuracy = 0.980
Epoch 3: mean loss = 0.092, accuracy = 0.967
2017-10-29 19:03:10 Iter 3600: batch trn loss = 0.162, accuracy = 0.920
2017-10-29 19:03:42 Iter 3700: batch trn loss = 0.106, accuracy = 0.960
2017-10-29 19:04:14 Iter 3800: batch trn loss = 0.075, accuracy = 0.960
2017-10-29 19:04:46 Iter 3900: batch trn loss = 0.159, accuracy = 0.960
2017-10-29 19:05:18 Iter 4000: batch trn loss = 0.040, accuracy = 1.000
2017-10-29 19:05:50 Iter 4100: batch trn loss = 0.164, accuracy = 0.940
2017-10-29 19:06:21 Iter 4200: batch trn loss = 0.050, accuracy = 1.000
2017-10-29 19:06:53 Iter 4300: batch trn loss = 0.010, accuracy = 1.000
2017-10-29 19:07:25 Iter 4400: batch trn loss = 0.145, accuracy = 0.920
Epoch 4: mean loss = 0.086, accuracy = 0.970
validation
Epoch 0: mean loss = 0.881, accuracy = 0.799

```

```

Out[32]: (0.88099295, 0.7994)

```