

Convolutional neural networks

In this notebook, we'll put together our convolutional layers to implement a 3-layer CNN. Then, we'll ask you to implement a CNN that can achieve $> 65\%$ validation error on CIFAR-10.

CS231n has built a solid API for building these modular frameworks and training them, and we will use their very well implemented framework as opposed to "reinventing the wheel." This includes using their Solver, various utility functions, their layer structure, and their implementation of fast CNN layers. This also includes `nndl.fc_net`, `nndl.layers`, and `nndl.layer_utils`. As in prior assignments, we thank Serena Yeung & Justin Johnson for permission to use code written for the CS 231n class (cs231n.stanford.edu).

If you have not completed the Spatial BatchNorm Notebook, please see the following description from that notebook:

Please copy and paste your prior implemented code from HW #4 to start this assignment. If you did not correctly implement the layers in HW #4, you may collaborate with a classmate to use their layer implementations from HW #4. You may also visit TA or Prof OH to correct your implementation.

You'll want to copy and paste from HW #4:

- `layers.py` for your FC network layers, as well as batchnorm and dropout.
- `layer_utils.py` for your combined FC network layers.
- `optim.py` for your optimizers.

Be sure to place these in the `nndl/` directory so they're imported correctly. Note, as announced in class, we will not be releasing our solutions.

```
In [1]: # As usual, a bit of setup

import numpy as np
import matplotlib.pyplot as plt
from nndl.cnn import *
from cs231n.data_utils import get_CIFAR10_data
from cs231n.gradient_check import eval_numerical_gradient_array, eval_numerical_gradient
from nndl.layers import *
from nndl.conv_layers import *
from cs231n.fast_layers import *
from cs231n.solver import Solver

%matplotlib inline
plt.rcParams['figure.figsize'] = (10.0, 8.0) # set default size of plots
plt.rcParams['image.interpolation'] = 'nearest'
plt.rcParams['image.cmap'] = 'gray'

# for auto-reloading external modules
# see http://stackoverflow.com/questions/1907993/autoreload-of-modules-in-ipython
%load_ext autoreload
%autoreload 2

def rel_error(x, y):
    """ returns relative error """
    return np.max(np.abs(x - y) / (np.maximum(1e-8, np.abs(x) + np.abs(y))))
```

```
In [2]: # Load the (preprocessed) CIFAR10 data.

data = get_CIFAR10_data()
for k in data.keys():
    print('{}: {}'.format(k, data[k].shape))
```

```
X_train: (49000, 3, 32, 32)
y_train: (49000,)
X_val: (1000, 3, 32, 32)
y_val: (1000,)
X_test: (1000, 3, 32, 32)
y_test: (1000,)
```

Three layer CNN

In this notebook, you will implement a three layer CNN. The `ThreeLayerConvNet` class is in `nndl/cnn.py`. You'll need to modify that code for this section, including the initialization, as well as the calculation of the loss and gradients. You should be able to use the building blocks you have either earlier coded or that we have provided. Be sure to use the fast layers.

The architecture of this CNN will be:

conv - relu - 2x2 max pool - affine - relu - affine - softmax

We won't use batchnorm yet. You've also done enough of these to know how to debug; use the cells below.

Note: As we are implementing several layers CNN networks. The gradient error can be expected for the `eval_numerical_gradient()` function. If your `W1` max relative error and `W2` max relative error are around or below 0.01, they should be acceptable. Other errors should be less than $1e-5$.

```
In [3]: num_inputs = 2
input_dim = (3, 16, 16)
reg = 0.0
num_classes = 10
X = np.random.randn(num_inputs, *input_dim)
y = np.random.randint(num_classes, size=num_inputs)

model = ThreeLayerConvNet(num_filters=3, filter_size=3,
                           input_dim=input_dim, hidden_dim=7,
                           dtype=np.float64)
loss, grads = model.loss(X, y)
for param_name in sorted(grads):
    f = lambda _: model.loss(X, y)[0]
    param_grad_num = eval_numerical_gradient(f, model.params[param_name], verbose=False, h=1e-6)
    e = rel_error(param_grad_num, grads[param_name])
    print('{} max relative error: {}'.format(param_name, rel_error(param_grad_num, grads[param_name])))
```

```
W1 max relative error: 0.17369254221108352
W2 max relative error: 0.01003010098195593
W3 max relative error: 0.002829701265554436
b1 max relative error: 0.00011525595307646323
b2 max relative error: 0.1446326467843803
b3 max relative error: 1.0669031439138003e-09
```

Overfit small dataset

To check your CNN implementation, let's overfit a small dataset.

```
In [4]: num_train = 100
small_data = {
    'X_train': data['X_train'][:num_train],
    'y_train': data['y_train'][:num_train],
    'X_val': data['X_val'],
    'y_val': data['y_val'],
}

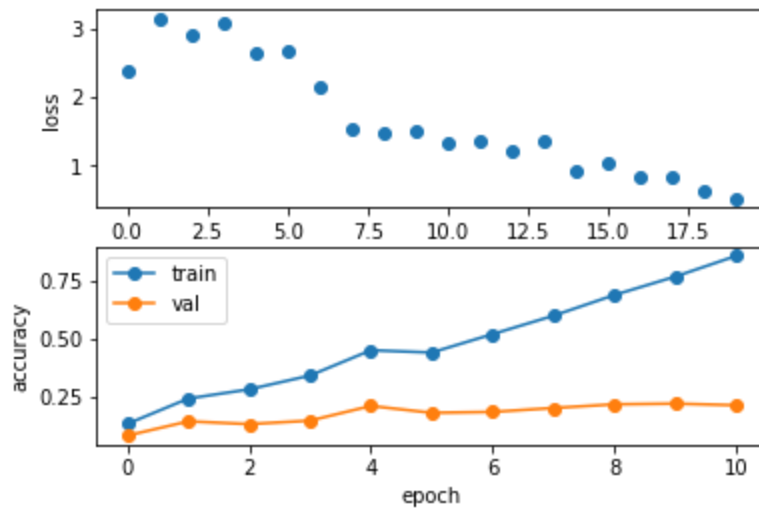
model = ThreeLayerConvNet(weight_scale=1e-2)

solver = Solver(model, small_data,
                num_epochs=10, batch_size=50,
                update_rule='adam',
                optim_config={
                    'learning_rate': 1e-3,
                },
                verbose=True, print_every=1)
solver.train()

(Iteration 1 / 20) loss: 2.371367
(Epoch 0 / 10) train acc: 0.130000; val_acc: 0.079000
(Iteration 2 / 20) loss: 3.147996
(Epoch 1 / 10) train acc: 0.240000; val_acc: 0.141000
(Iteration 3 / 20) loss: 2.916641
(Iteration 4 / 20) loss: 3.084006
(Epoch 2 / 10) train acc: 0.280000; val_acc: 0.129000
(Iteration 5 / 20) loss: 2.636065
(Iteration 6 / 20) loss: 2.684105
(Epoch 3 / 10) train acc: 0.340000; val_acc: 0.144000
(Iteration 7 / 20) loss: 2.146016
(Iteration 8 / 20) loss: 1.541894
(Epoch 4 / 10) train acc: 0.450000; val_acc: 0.208000
(Iteration 9 / 20) loss: 1.465960
(Iteration 10 / 20) loss: 1.501722
(Epoch 5 / 10) train acc: 0.440000; val_acc: 0.178000
(Iteration 11 / 20) loss: 1.341185
(Iteration 12 / 20) loss: 1.363934
(Epoch 6 / 10) train acc: 0.520000; val_acc: 0.182000
(Iteration 13 / 20) loss: 1.216340
(Iteration 14 / 20) loss: 1.352318
(Epoch 7 / 10) train acc: 0.600000; val_acc: 0.199000
(Iteration 15 / 20) loss: 0.921658
(Iteration 16 / 20) loss: 1.045481
(Epoch 8 / 10) train acc: 0.690000; val_acc: 0.215000
(Iteration 17 / 20) loss: 0.823893
(Iteration 18 / 20) loss: 0.841351
(Epoch 9 / 10) train acc: 0.770000; val_acc: 0.218000
(Iteration 19 / 20) loss: 0.620718
(Iteration 20 / 20) loss: 0.519454
(Epoch 10 / 10) train acc: 0.860000; val_acc: 0.211000
```

```
In [5]: plt.subplot(2, 1, 1)
plt.plot(solver.loss_history, 'o')
plt.xlabel('iteration')
plt.ylabel('loss')

plt.subplot(2, 1, 2)
plt.plot(solver.train_acc_history, '-o')
plt.plot(solver.val_acc_history, '-o')
plt.legend(['train', 'val'], loc='upper left')
plt.xlabel('epoch')
plt.ylabel('accuracy')
plt.show()
```



Train the network

Now we train the 3 layer CNN on CIFAR-10 and assess its accuracy.

```
In [6]: model = ThreeLayerConvNet(weight_scale=0.001, hidden_dim=500, reg=0.001)

solver = Solver(model, data,
                 num_epochs=1, batch_size=50,
                 update_rule='adam',
                 optim_config={
                     'learning_rate': 1e-3,
                 },
                 verbose=True, print_every=20)
solver.train()
```

(Iteration 1 / 980) loss: 2.304605
(Epoch 0 / 1) train acc: 0.080000; val_acc: 0.078000
(Iteration 21 / 980) loss: 2.129291
(Iteration 41 / 980) loss: 1.937440
(Iteration 61 / 980) loss: 1.765984
(Iteration 81 / 980) loss: 1.875893
(Iteration 101 / 980) loss: 1.720115
(Iteration 121 / 980) loss: 2.073529
(Iteration 141 / 980) loss: 1.888670
(Iteration 161 / 980) loss: 1.735924
(Iteration 181 / 980) loss: 1.727734
(Iteration 201 / 980) loss: 1.699182
(Iteration 221 / 980) loss: 1.604645
(Iteration 241 / 980) loss: 1.873202
(Iteration 261 / 980) loss: 1.836574
(Iteration 281 / 980) loss: 1.695972
(Iteration 301 / 980) loss: 1.710535
(Iteration 321 / 980) loss: 1.457101
(Iteration 341 / 980) loss: 1.891809
(Iteration 361 / 980) loss: 1.584709
(Iteration 381 / 980) loss: 1.751821
(Iteration 401 / 980) loss: 1.244619
(Iteration 421 / 980) loss: 1.701515
(Iteration 441 / 980) loss: 1.715431
(Iteration 461 / 980) loss: 1.451432
(Iteration 481 / 980) loss: 1.639757
(Iteration 501 / 980) loss: 1.773881
(Iteration 521 / 980) loss: 1.417885
(Iteration 541 / 980) loss: 1.646714
(Iteration 561 / 980) loss: 1.520109
(Iteration 581 / 980) loss: 1.651191
(Iteration 601 / 980) loss: 1.525304
(Iteration 621 / 980) loss: 1.363232
(Iteration 641 / 980) loss: 1.405605
(Iteration 661 / 980) loss: 1.607109
(Iteration 681 / 980) loss: 1.285568
(Iteration 701 / 980) loss: 1.607308
(Iteration 721 / 980) loss: 1.563374
(Iteration 741 / 980) loss: 1.450425
(Iteration 761 / 980) loss: 1.275599
(Iteration 781 / 980) loss: 1.471156
(Iteration 801 / 980) loss: 1.397873
(Iteration 821 / 980) loss: 1.592969
(Iteration 841 / 980) loss: 1.640301
(Iteration 861 / 980) loss: 1.553780
(Iteration 881 / 980) loss: 1.568807
(Iteration 901 / 980) loss: 1.459975
(Iteration 921 / 980) loss: 1.774200
(Iteration 941 / 980) loss: 1.544449
(Iteration 961 / 980) loss: 1.625979
(Epoch 1 / 1) train acc: 0.456000; val_acc: 0.456000

Get > 65% validation accuracy on CIFAR-10.

In the last part of the assignment, we'll now ask you to train a CNN to get better than 65% validation accuracy on CIFAR-10.

Things you should try:

- Filter size: Above we used 7x7; but VGGNet and onwards showed stacks of 3x3 filters are good.
- Number of filters: Above we used 32 filters. Do more or fewer do better?
- Batch normalization: Try adding spatial batch normalization after convolution layers and vanilla batch normalization after affine layers. Do your networks train faster?
- Network architecture: Can a deeper CNN do better? Consider these architectures:
 - [conv-relu-pool]xN - conv - relu - [affine]xM - [softmax or SVM]
 - [conv-relu-pool]xN - [affine]xM - [softmax or SVM]
 - [conv-relu-conv-relu-pool]xN - [affine]xM - [softmax or SVM]

Tips for training

For each network architecture that you try, you should tune the learning rate and regularization strength. When doing this there are a couple important things to keep in mind:

- If the parameters are working well, you should see improvement within a few hundred iterations
- Remember the coarse-to-fine approach for hyperparameter tuning: start by testing a large range of hyperparameters for just a few training iterations to find the combinations of parameters that are working at all.
- Once you have found some sets of parameters that seem to work, search more finely around these parameters. You may need to train for more epochs.

```
In [7]: # ===== #
# YOUR CODE HERE:
#   Implement a CNN to achieve greater than 65% validation accuracy
#   on CIFAR-10.
# ===== #
model = ThreeLayerConvNet( filter_size=3,
                           num_filters=64,
                           weight_scale=0.003,
                           hidden_dim=512,
                           reg=0.002)

solver = Solver(model, data,
                num_epochs=10, batch_size=64,
                update_rule='adam',
                optim_config={
                    'learning_rate': 5e-4,
                },
                verbose=True, print_every=5)
solver.train()

# ===== #
# END YOUR CODE HERE
# ===== #
```


(Iteration 1 / 7650) loss: 2.375404
(Epoch 0 / 10) train acc: 0.119000; val_acc: 0.145000
(Iteration 6 / 7650) loss: 2.381252
(Iteration 11 / 7650) loss: 2.120171
(Iteration 16 / 7650) loss: 2.017179
(Iteration 21 / 7650) loss: 2.195607
(Iteration 26 / 7650) loss: 2.096921
(Iteration 31 / 7650) loss: 1.891293
(Iteration 36 / 7650) loss: 1.975608
(Iteration 41 / 7650) loss: 1.865667
(Iteration 46 / 7650) loss: 2.002808
(Iteration 51 / 7650) loss: 1.801819
(Iteration 56 / 7650) loss: 1.716044
(Iteration 61 / 7650) loss: 1.925032
(Iteration 66 / 7650) loss: 1.760886
(Iteration 71 / 7650) loss: 1.726783
(Iteration 76 / 7650) loss: 1.653626
(Iteration 81 / 7650) loss: 1.695570
(Iteration 86 / 7650) loss: 1.615822
(Iteration 91 / 7650) loss: 1.593690
(Iteration 96 / 7650) loss: 1.802100
(Iteration 101 / 7650) loss: 1.633719
(Iteration 106 / 7650) loss: 1.619523
(Iteration 111 / 7650) loss: 1.703927
(Iteration 116 / 7650) loss: 1.728776
(Iteration 121 / 7650) loss: 1.615352
(Iteration 126 / 7650) loss: 1.840884
(Iteration 131 / 7650) loss: 1.540247
(Iteration 136 / 7650) loss: 1.550069
(Iteration 141 / 7650) loss: 1.772074
(Iteration 146 / 7650) loss: 1.623844
(Iteration 151 / 7650) loss: 1.560213
(Iteration 156 / 7650) loss: 1.933874
(Iteration 161 / 7650) loss: 1.459333
(Iteration 166 / 7650) loss: 1.525070
(Iteration 171 / 7650) loss: 1.882932
(Iteration 176 / 7650) loss: 1.554650
(Iteration 181 / 7650) loss: 1.670744
(Iteration 186 / 7650) loss: 1.480088
(Iteration 191 / 7650) loss: 1.522078
(Iteration 196 / 7650) loss: 1.703526
(Iteration 201 / 7650) loss: 1.643115
(Iteration 206 / 7650) loss: 1.460049
(Iteration 211 / 7650) loss: 1.923548
(Iteration 216 / 7650) loss: 1.392906
(Iteration 221 / 7650) loss: 1.457751
(Iteration 226 / 7650) loss: 1.623576
(Iteration 231 / 7650) loss: 1.640046
(Iteration 236 / 7650) loss: 1.511859
(Iteration 241 / 7650) loss: 1.522858
(Iteration 246 / 7650) loss: 1.535287
(Iteration 251 / 7650) loss: 1.484968
(Iteration 256 / 7650) loss: 1.426736
(Iteration 261 / 7650) loss: 1.455502
(Iteration 266 / 7650) loss: 1.429950
(Iteration 271 / 7650) loss: 1.380375
(Iteration 276 / 7650) loss: 1.518550
(Iteration 281 / 7650) loss: 1.567545
(Iteration 286 / 7650) loss: 1.238830
(Iteration 291 / 7650) loss: 1.445629
(Iteration 296 / 7650) loss: 1.457910

(Iteration 301 / 7650) loss: 1.549622
(Iteration 306 / 7650) loss: 1.519667
(Iteration 311 / 7650) loss: 1.668039
(Iteration 316 / 7650) loss: 1.698929
(Iteration 321 / 7650) loss: 1.520792
(Iteration 326 / 7650) loss: 1.415297
(Iteration 331 / 7650) loss: 1.450703
(Iteration 336 / 7650) loss: 1.476154
(Iteration 341 / 7650) loss: 1.432166
(Iteration 346 / 7650) loss: 1.410417
(Iteration 351 / 7650) loss: 1.416161
(Iteration 356 / 7650) loss: 1.248444
(Iteration 361 / 7650) loss: 1.394517
(Iteration 366 / 7650) loss: 1.497947
(Iteration 371 / 7650) loss: 1.516114
(Iteration 376 / 7650) loss: 1.297164
(Iteration 381 / 7650) loss: 1.520393
(Iteration 386 / 7650) loss: 1.473067
(Iteration 391 / 7650) loss: 1.545345
(Iteration 396 / 7650) loss: 1.441910
(Iteration 401 / 7650) loss: 1.317155
(Iteration 406 / 7650) loss: 1.392173
(Iteration 411 / 7650) loss: 1.358100
(Iteration 416 / 7650) loss: 1.189245
(Iteration 421 / 7650) loss: 1.132642
(Iteration 426 / 7650) loss: 1.585379
(Iteration 431 / 7650) loss: 1.618765
(Iteration 436 / 7650) loss: 1.486816
(Iteration 441 / 7650) loss: 1.573074
(Iteration 446 / 7650) loss: 1.345059
(Iteration 451 / 7650) loss: 1.314734
(Iteration 456 / 7650) loss: 1.420065
(Iteration 461 / 7650) loss: 1.511526
(Iteration 466 / 7650) loss: 1.471766
(Iteration 471 / 7650) loss: 1.301032
(Iteration 476 / 7650) loss: 1.361857
(Iteration 481 / 7650) loss: 1.254953
(Iteration 486 / 7650) loss: 1.241131
(Iteration 491 / 7650) loss: 1.340847
(Iteration 496 / 7650) loss: 1.572579
(Iteration 501 / 7650) loss: 1.590191
(Iteration 506 / 7650) loss: 1.346362
(Iteration 511 / 7650) loss: 1.339800
(Iteration 516 / 7650) loss: 1.360269
(Iteration 521 / 7650) loss: 1.424104
(Iteration 526 / 7650) loss: 1.293776
(Iteration 531 / 7650) loss: 1.232933
(Iteration 536 / 7650) loss: 1.476707
(Iteration 541 / 7650) loss: 1.582495
(Iteration 546 / 7650) loss: 1.386911
(Iteration 551 / 7650) loss: 1.209217
(Iteration 556 / 7650) loss: 1.160716
(Iteration 561 / 7650) loss: 1.166313
(Iteration 566 / 7650) loss: 1.142204
(Iteration 571 / 7650) loss: 1.575638
(Iteration 576 / 7650) loss: 1.526949
(Iteration 581 / 7650) loss: 1.523456
(Iteration 586 / 7650) loss: 1.515171
(Iteration 591 / 7650) loss: 1.464246
(Iteration 596 / 7650) loss: 1.322605
(Iteration 601 / 7650) loss: 1.553033
(Iteration 606 / 7650) loss: 1.218889

(Iteration 611 / 7650) loss: 1.099775
(Iteration 616 / 7650) loss: 1.512513
(Iteration 621 / 7650) loss: 1.100293
(Iteration 626 / 7650) loss: 1.167698
(Iteration 631 / 7650) loss: 1.507277
(Iteration 636 / 7650) loss: 1.187071
(Iteration 641 / 7650) loss: 1.326051
(Iteration 646 / 7650) loss: 1.368890
(Iteration 651 / 7650) loss: 1.340888
(Iteration 656 / 7650) loss: 1.236634
(Iteration 661 / 7650) loss: 1.519877
(Iteration 666 / 7650) loss: 1.711923
(Iteration 671 / 7650) loss: 1.360022
(Iteration 676 / 7650) loss: 1.550958
(Iteration 681 / 7650) loss: 1.400504
(Iteration 686 / 7650) loss: 1.624957
(Iteration 691 / 7650) loss: 1.118548
(Iteration 696 / 7650) loss: 1.209236
(Iteration 701 / 7650) loss: 1.134515
(Iteration 706 / 7650) loss: 1.583019
(Iteration 711 / 7650) loss: 1.366197
(Iteration 716 / 7650) loss: 1.220449
(Iteration 721 / 7650) loss: 1.518120
(Iteration 726 / 7650) loss: 1.495330
(Iteration 731 / 7650) loss: 1.402939
(Iteration 736 / 7650) loss: 1.595109
(Iteration 741 / 7650) loss: 1.513344
(Iteration 746 / 7650) loss: 1.421319
(Iteration 751 / 7650) loss: 1.150230
(Iteration 756 / 7650) loss: 1.160614
(Iteration 761 / 7650) loss: 1.102766
(Epoch 1 / 10) train acc: 0.567000; val_acc: 0.529000
(Iteration 766 / 7650) loss: 1.287275
(Iteration 771 / 7650) loss: 1.236410
(Iteration 776 / 7650) loss: 1.248851
(Iteration 781 / 7650) loss: 1.389969
(Iteration 786 / 7650) loss: 1.372728
(Iteration 791 / 7650) loss: 1.185078
(Iteration 796 / 7650) loss: 1.684234
(Iteration 801 / 7650) loss: 1.513874
(Iteration 806 / 7650) loss: 1.519563
(Iteration 811 / 7650) loss: 1.349280
(Iteration 816 / 7650) loss: 1.282468
(Iteration 821 / 7650) loss: 1.276529
(Iteration 826 / 7650) loss: 1.150856
(Iteration 831 / 7650) loss: 1.685941
(Iteration 836 / 7650) loss: 1.484115
(Iteration 841 / 7650) loss: 1.257338
(Iteration 846 / 7650) loss: 1.222756
(Iteration 851 / 7650) loss: 1.448713
(Iteration 856 / 7650) loss: 1.338826
(Iteration 861 / 7650) loss: 1.378435
(Iteration 866 / 7650) loss: 1.252024
(Iteration 871 / 7650) loss: 1.268984
(Iteration 876 / 7650) loss: 1.377200
(Iteration 881 / 7650) loss: 1.434672
(Iteration 886 / 7650) loss: 1.356351
(Iteration 891 / 7650) loss: 1.544971
(Iteration 896 / 7650) loss: 1.163800
(Iteration 901 / 7650) loss: 1.387905
(Iteration 906 / 7650) loss: 1.206827
(Iteration 911 / 7650) loss: 1.061513

(Iteration 916 / 7650) loss: 1.122370
(Iteration 921 / 7650) loss: 1.308596
(Iteration 926 / 7650) loss: 1.289988
(Iteration 931 / 7650) loss: 1.211290
(Iteration 936 / 7650) loss: 1.300536
(Iteration 941 / 7650) loss: 1.245318
(Iteration 946 / 7650) loss: 1.345550
(Iteration 951 / 7650) loss: 1.394880
(Iteration 956 / 7650) loss: 1.481462
(Iteration 961 / 7650) loss: 1.585424
(Iteration 966 / 7650) loss: 1.185483
(Iteration 971 / 7650) loss: 1.324891
(Iteration 976 / 7650) loss: 1.463105
(Iteration 981 / 7650) loss: 1.451366
(Iteration 986 / 7650) loss: 1.145399
(Iteration 991 / 7650) loss: 1.237403
(Iteration 996 / 7650) loss: 1.128637
(Iteration 1001 / 7650) loss: 1.161483
(Iteration 1006 / 7650) loss: 1.552966
(Iteration 1011 / 7650) loss: 1.374469
(Iteration 1016 / 7650) loss: 1.403682
(Iteration 1021 / 7650) loss: 1.378902
(Iteration 1026 / 7650) loss: 1.322583
(Iteration 1031 / 7650) loss: 1.368373
(Iteration 1036 / 7650) loss: 1.211848
(Iteration 1041 / 7650) loss: 1.293859
(Iteration 1046 / 7650) loss: 1.392876
(Iteration 1051 / 7650) loss: 1.237761
(Iteration 1056 / 7650) loss: 1.401176
(Iteration 1061 / 7650) loss: 1.378087
(Iteration 1066 / 7650) loss: 1.373522
(Iteration 1071 / 7650) loss: 1.154970
(Iteration 1076 / 7650) loss: 1.312085
(Iteration 1081 / 7650) loss: 1.141953
(Iteration 1086 / 7650) loss: 1.211608
(Iteration 1091 / 7650) loss: 1.139052
(Iteration 1096 / 7650) loss: 1.406756
(Iteration 1101 / 7650) loss: 1.409238
(Iteration 1106 / 7650) loss: 1.215847
(Iteration 1111 / 7650) loss: 1.262547
(Iteration 1116 / 7650) loss: 1.262257
(Iteration 1121 / 7650) loss: 1.404782
(Iteration 1126 / 7650) loss: 1.101103
(Iteration 1131 / 7650) loss: 1.113717
(Iteration 1136 / 7650) loss: 1.344757
(Iteration 1141 / 7650) loss: 1.468514
(Iteration 1146 / 7650) loss: 1.196354
(Iteration 1151 / 7650) loss: 1.221937
(Iteration 1156 / 7650) loss: 1.207940
(Iteration 1161 / 7650) loss: 1.285113
(Iteration 1166 / 7650) loss: 1.286871
(Iteration 1171 / 7650) loss: 1.168635
(Iteration 1176 / 7650) loss: 1.477423
(Iteration 1181 / 7650) loss: 1.243895
(Iteration 1186 / 7650) loss: 1.201585
(Iteration 1191 / 7650) loss: 1.466383
(Iteration 1196 / 7650) loss: 1.430389
(Iteration 1201 / 7650) loss: 1.262316
(Iteration 1206 / 7650) loss: 1.192513
(Iteration 1211 / 7650) loss: 1.506640
(Iteration 1216 / 7650) loss: 1.152213
(Iteration 1221 / 7650) loss: 1.229144

(Iteration 1226 / 7650) loss: 1.214342
(Iteration 1231 / 7650) loss: 1.380804
(Iteration 1236 / 7650) loss: 1.349223
(Iteration 1241 / 7650) loss: 1.289804
(Iteration 1246 / 7650) loss: 1.326106
(Iteration 1251 / 7650) loss: 1.491270
(Iteration 1256 / 7650) loss: 1.136388
(Iteration 1261 / 7650) loss: 1.504285
(Iteration 1266 / 7650) loss: 1.489975
(Iteration 1271 / 7650) loss: 1.373026
(Iteration 1276 / 7650) loss: 1.103823
(Iteration 1281 / 7650) loss: 1.213054
(Iteration 1286 / 7650) loss: 1.159515
(Iteration 1291 / 7650) loss: 1.323245
(Iteration 1296 / 7650) loss: 1.392215
(Iteration 1301 / 7650) loss: 1.250193
(Iteration 1306 / 7650) loss: 1.279439
(Iteration 1311 / 7650) loss: 1.045455
(Iteration 1316 / 7650) loss: 1.015152
(Iteration 1321 / 7650) loss: 1.266039
(Iteration 1326 / 7650) loss: 1.259720
(Iteration 1331 / 7650) loss: 1.081595
(Iteration 1336 / 7650) loss: 1.256857
(Iteration 1341 / 7650) loss: 1.321061
(Iteration 1346 / 7650) loss: 1.278328
(Iteration 1351 / 7650) loss: 1.144807
(Iteration 1356 / 7650) loss: 1.485163
(Iteration 1361 / 7650) loss: 1.326993
(Iteration 1366 / 7650) loss: 1.387003
(Iteration 1371 / 7650) loss: 1.232158
(Iteration 1376 / 7650) loss: 1.346923
(Iteration 1381 / 7650) loss: 1.157563
(Iteration 1386 / 7650) loss: 1.090910
(Iteration 1391 / 7650) loss: 1.230247
(Iteration 1396 / 7650) loss: 1.326974
(Iteration 1401 / 7650) loss: 0.905055
(Iteration 1406 / 7650) loss: 1.539698
(Iteration 1411 / 7650) loss: 1.373597
(Iteration 1416 / 7650) loss: 1.223957
(Iteration 1421 / 7650) loss: 1.134753
(Iteration 1426 / 7650) loss: 1.305815
(Iteration 1431 / 7650) loss: 1.305983
(Iteration 1436 / 7650) loss: 1.470798
(Iteration 1441 / 7650) loss: 1.271812
(Iteration 1446 / 7650) loss: 1.132949
(Iteration 1451 / 7650) loss: 1.387401
(Iteration 1456 / 7650) loss: 1.009688
(Iteration 1461 / 7650) loss: 1.122642
(Iteration 1466 / 7650) loss: 1.146548
(Iteration 1471 / 7650) loss: 1.080333
(Iteration 1476 / 7650) loss: 1.212916
(Iteration 1481 / 7650) loss: 1.657279
(Iteration 1486 / 7650) loss: 1.419555
(Iteration 1491 / 7650) loss: 1.237609
(Iteration 1496 / 7650) loss: 1.434508
(Iteration 1501 / 7650) loss: 1.401233
(Iteration 1506 / 7650) loss: 1.231421
(Iteration 1511 / 7650) loss: 1.260346
(Iteration 1516 / 7650) loss: 1.287610
(Iteration 1521 / 7650) loss: 1.202495
(Iteration 1526 / 7650) loss: 1.318126
(Epoch 2 / 10) train acc: 0.595000; val_acc: 0.585000

(Iteration 1531 / 7650) loss: 1.141751
(Iteration 1536 / 7650) loss: 1.217275
(Iteration 1541 / 7650) loss: 1.229369
(Iteration 1546 / 7650) loss: 0.893589
(Iteration 1551 / 7650) loss: 1.443570
(Iteration 1556 / 7650) loss: 1.175604
(Iteration 1561 / 7650) loss: 1.538328
(Iteration 1566 / 7650) loss: 1.409811
(Iteration 1571 / 7650) loss: 1.330735
(Iteration 1576 / 7650) loss: 1.236685
(Iteration 1581 / 7650) loss: 1.271454
(Iteration 1586 / 7650) loss: 1.118203
(Iteration 1591 / 7650) loss: 1.542707
(Iteration 1596 / 7650) loss: 1.395810
(Iteration 1601 / 7650) loss: 1.381102
(Iteration 1606 / 7650) loss: 0.925183
(Iteration 1611 / 7650) loss: 1.176593
(Iteration 1616 / 7650) loss: 1.151910
(Iteration 1621 / 7650) loss: 1.083715
(Iteration 1626 / 7650) loss: 1.446853
(Iteration 1631 / 7650) loss: 1.104148
(Iteration 1636 / 7650) loss: 1.499093
(Iteration 1641 / 7650) loss: 1.319322
(Iteration 1646 / 7650) loss: 1.238586
(Iteration 1651 / 7650) loss: 1.227501
(Iteration 1656 / 7650) loss: 1.069758
(Iteration 1661 / 7650) loss: 1.153331
(Iteration 1666 / 7650) loss: 1.395295
(Iteration 1671 / 7650) loss: 1.076109
(Iteration 1676 / 7650) loss: 1.259550
(Iteration 1681 / 7650) loss: 1.015840
(Iteration 1686 / 7650) loss: 1.052440
(Iteration 1691 / 7650) loss: 1.278808
(Iteration 1696 / 7650) loss: 1.284523
(Iteration 1701 / 7650) loss: 1.086230
(Iteration 1706 / 7650) loss: 1.124735
(Iteration 1711 / 7650) loss: 1.399850
(Iteration 1716 / 7650) loss: 1.071046
(Iteration 1721 / 7650) loss: 1.163681
(Iteration 1726 / 7650) loss: 1.207589
(Iteration 1731 / 7650) loss: 1.041809
(Iteration 1736 / 7650) loss: 1.082617
(Iteration 1741 / 7650) loss: 1.205055
(Iteration 1746 / 7650) loss: 1.444534
(Iteration 1751 / 7650) loss: 1.174946
(Iteration 1756 / 7650) loss: 1.158094
(Iteration 1761 / 7650) loss: 1.209477
(Iteration 1766 / 7650) loss: 1.179872
(Iteration 1771 / 7650) loss: 1.079653
(Iteration 1776 / 7650) loss: 1.167513
(Iteration 1781 / 7650) loss: 1.222531
(Iteration 1786 / 7650) loss: 0.983213
(Iteration 1791 / 7650) loss: 1.125855
(Iteration 1796 / 7650) loss: 1.192924
(Iteration 1801 / 7650) loss: 1.267595
(Iteration 1806 / 7650) loss: 0.792247
(Iteration 1811 / 7650) loss: 1.178695
(Iteration 1816 / 7650) loss: 1.272754
(Iteration 1821 / 7650) loss: 1.028227
(Iteration 1826 / 7650) loss: 1.230707
(Iteration 1831 / 7650) loss: 1.207479
(Iteration 1836 / 7650) loss: 1.474924

(Iteration 1841 / 7650) loss: 1.187187
(Iteration 1846 / 7650) loss: 1.110509
(Iteration 1851 / 7650) loss: 1.033334
(Iteration 1856 / 7650) loss: 1.359105
(Iteration 1861 / 7650) loss: 1.281690
(Iteration 1866 / 7650) loss: 1.093891
(Iteration 1871 / 7650) loss: 1.160881
(Iteration 1876 / 7650) loss: 1.158154
(Iteration 1881 / 7650) loss: 0.981155
(Iteration 1886 / 7650) loss: 1.187854
(Iteration 1891 / 7650) loss: 0.888369
(Iteration 1896 / 7650) loss: 1.097645
(Iteration 1901 / 7650) loss: 1.075217
(Iteration 1906 / 7650) loss: 0.951580
(Iteration 1911 / 7650) loss: 1.159625
(Iteration 1916 / 7650) loss: 0.889478
(Iteration 1921 / 7650) loss: 1.065364
(Iteration 1926 / 7650) loss: 1.307873
(Iteration 1931 / 7650) loss: 1.271440
(Iteration 1936 / 7650) loss: 1.083492
(Iteration 1941 / 7650) loss: 1.109733
(Iteration 1946 / 7650) loss: 1.567912
(Iteration 1951 / 7650) loss: 1.180957
(Iteration 1956 / 7650) loss: 1.076408
(Iteration 1961 / 7650) loss: 1.238759
(Iteration 1966 / 7650) loss: 1.033110
(Iteration 1971 / 7650) loss: 0.801362
(Iteration 1976 / 7650) loss: 1.198918
(Iteration 1981 / 7650) loss: 1.067103
(Iteration 1986 / 7650) loss: 0.947905
(Iteration 1991 / 7650) loss: 1.026179
(Iteration 1996 / 7650) loss: 1.207072
(Iteration 2001 / 7650) loss: 0.994997
(Iteration 2006 / 7650) loss: 1.001645
(Iteration 2011 / 7650) loss: 1.344866
(Iteration 2016 / 7650) loss: 1.429101
(Iteration 2021 / 7650) loss: 1.074737
(Iteration 2026 / 7650) loss: 0.961077
(Iteration 2031 / 7650) loss: 1.400884
(Iteration 2036 / 7650) loss: 1.038015
(Iteration 2041 / 7650) loss: 0.945563
(Iteration 2046 / 7650) loss: 1.345094
(Iteration 2051 / 7650) loss: 1.260719
(Iteration 2056 / 7650) loss: 1.319633
(Iteration 2061 / 7650) loss: 1.229115
(Iteration 2066 / 7650) loss: 1.004552
(Iteration 2071 / 7650) loss: 1.058406
(Iteration 2076 / 7650) loss: 0.951366
(Iteration 2081 / 7650) loss: 0.948819
(Iteration 2086 / 7650) loss: 0.995032
(Iteration 2091 / 7650) loss: 1.021543
(Iteration 2096 / 7650) loss: 1.608754
(Iteration 2101 / 7650) loss: 1.240050
(Iteration 2106 / 7650) loss: 1.139369
(Iteration 2111 / 7650) loss: 1.067664
(Iteration 2116 / 7650) loss: 1.165359
(Iteration 2121 / 7650) loss: 1.210336
(Iteration 2126 / 7650) loss: 1.379931
(Iteration 2131 / 7650) loss: 0.985329
(Iteration 2136 / 7650) loss: 1.030987
(Iteration 2141 / 7650) loss: 1.018374
(Iteration 2146 / 7650) loss: 1.033842

```
(Iteration 2151 / 7650) loss: 0.775469
(Iteration 2156 / 7650) loss: 1.237425
(Iteration 2161 / 7650) loss: 1.271160
(Iteration 2166 / 7650) loss: 1.178857
(Iteration 2171 / 7650) loss: 1.061437
(Iteration 2176 / 7650) loss: 1.165683
(Iteration 2181 / 7650) loss: 1.216204
(Iteration 2186 / 7650) loss: 1.026641
(Iteration 2191 / 7650) loss: 1.068539
(Iteration 2196 / 7650) loss: 1.053423
(Iteration 2201 / 7650) loss: 1.127587
(Iteration 2206 / 7650) loss: 1.099447
(Iteration 2211 / 7650) loss: 1.130363
(Iteration 2216 / 7650) loss: 1.042315
(Iteration 2221 / 7650) loss: 0.951079
(Iteration 2226 / 7650) loss: 1.320554
(Iteration 2231 / 7650) loss: 0.990163
(Iteration 2236 / 7650) loss: 1.138530
(Iteration 2241 / 7650) loss: 1.356667
(Iteration 2246 / 7650) loss: 1.407050
(Iteration 2251 / 7650) loss: 1.142687
(Iteration 2256 / 7650) loss: 1.188623
(Iteration 2261 / 7650) loss: 1.268388
(Iteration 2266 / 7650) loss: 1.197960
(Iteration 2271 / 7650) loss: 1.183959
(Iteration 2276 / 7650) loss: 1.084637
(Iteration 2281 / 7650) loss: 1.127329
(Iteration 2286 / 7650) loss: 1.043330
(Iteration 2291 / 7650) loss: 1.143472
(Epoch 3 / 10) train acc: 0.645000; val_acc: 0.627000
(Iteration 2296 / 7650) loss: 1.058702
(Iteration 2301 / 7650) loss: 1.074035
(Iteration 2306 / 7650) loss: 1.211971
(Iteration 2311 / 7650) loss: 1.086771
(Iteration 2316 / 7650) loss: 1.268818
(Iteration 2321 / 7650) loss: 1.080612
(Iteration 2326 / 7650) loss: 1.204467
(Iteration 2331 / 7650) loss: 1.278505
(Iteration 2336 / 7650) loss: 1.148992
(Iteration 2341 / 7650) loss: 1.038374
(Iteration 2346 / 7650) loss: 1.096963
(Iteration 2351 / 7650) loss: 1.132965
(Iteration 2356 / 7650) loss: 1.018590
(Iteration 2361 / 7650) loss: 1.062053
(Iteration 2366 / 7650) loss: 1.237584
(Iteration 2371 / 7650) loss: 1.076855
(Iteration 2376 / 7650) loss: 1.196138
(Iteration 2381 / 7650) loss: 1.309393
(Iteration 2386 / 7650) loss: 1.009853
(Iteration 2391 / 7650) loss: 1.033332
(Iteration 2396 / 7650) loss: 1.058203
(Iteration 2401 / 7650) loss: 1.085179
(Iteration 2406 / 7650) loss: 1.160828
(Iteration 2411 / 7650) loss: 1.201271
(Iteration 2416 / 7650) loss: 1.157460
(Iteration 2421 / 7650) loss: 1.084207
(Iteration 2426 / 7650) loss: 1.023660
(Iteration 2431 / 7650) loss: 1.343450
(Iteration 2436 / 7650) loss: 1.308118
(Iteration 2441 / 7650) loss: 1.034779
(Iteration 2446 / 7650) loss: 1.157877
(Iteration 2451 / 7650) loss: 1.222042
```


(Iteration 2456 / 7650) loss: 1.090559
(Iteration 2461 / 7650) loss: 1.356768
(Iteration 2466 / 7650) loss: 1.037442
(Iteration 2471 / 7650) loss: 1.147850
(Iteration 2476 / 7650) loss: 1.023458
(Iteration 2481 / 7650) loss: 0.997238
(Iteration 2486 / 7650) loss: 1.250184
(Iteration 2491 / 7650) loss: 1.141657
(Iteration 2496 / 7650) loss: 1.114382
(Iteration 2501 / 7650) loss: 1.166611
(Iteration 2506 / 7650) loss: 1.072644
(Iteration 2511 / 7650) loss: 1.124918
(Iteration 2516 / 7650) loss: 1.020647
(Iteration 2521 / 7650) loss: 1.194685
(Iteration 2526 / 7650) loss: 1.294606
(Iteration 2531 / 7650) loss: 1.149028
(Iteration 2536 / 7650) loss: 1.191885
(Iteration 2541 / 7650) loss: 1.307097
(Iteration 2546 / 7650) loss: 0.999466
(Iteration 2551 / 7650) loss: 1.050003
(Iteration 2556 / 7650) loss: 0.927072
(Iteration 2561 / 7650) loss: 1.238440
(Iteration 2566 / 7650) loss: 1.202060
(Iteration 2571 / 7650) loss: 1.080104
(Iteration 2576 / 7650) loss: 1.109120
(Iteration 2581 / 7650) loss: 1.304584
(Iteration 2586 / 7650) loss: 1.025634
(Iteration 2591 / 7650) loss: 1.012282
(Iteration 2596 / 7650) loss: 1.089440
(Iteration 2601 / 7650) loss: 1.079803
(Iteration 2606 / 7650) loss: 1.163080
(Iteration 2611 / 7650) loss: 1.381645
(Iteration 2616 / 7650) loss: 1.152684
(Iteration 2621 / 7650) loss: 0.985922
(Iteration 2626 / 7650) loss: 1.039961
(Iteration 2631 / 7650) loss: 0.952380
(Iteration 2636 / 7650) loss: 1.160222
(Iteration 2641 / 7650) loss: 0.809588
(Iteration 2646 / 7650) loss: 1.093748
(Iteration 2651 / 7650) loss: 1.192671
(Iteration 2656 / 7650) loss: 1.008389
(Iteration 2661 / 7650) loss: 1.272457
(Iteration 2666 / 7650) loss: 0.899237
(Iteration 2671 / 7650) loss: 1.071578
(Iteration 2676 / 7650) loss: 1.007883
(Iteration 2681 / 7650) loss: 1.247309
(Iteration 2686 / 7650) loss: 0.963975
(Iteration 2691 / 7650) loss: 1.008360
(Iteration 2696 / 7650) loss: 1.077825
(Iteration 2701 / 7650) loss: 1.044406
(Iteration 2706 / 7650) loss: 1.120359
(Iteration 2711 / 7650) loss: 0.975582
(Iteration 2716 / 7650) loss: 0.894732
(Iteration 2721 / 7650) loss: 1.242861
(Iteration 2726 / 7650) loss: 1.002045
(Iteration 2731 / 7650) loss: 0.821566
(Iteration 2736 / 7650) loss: 0.843965
(Iteration 2741 / 7650) loss: 0.913128
(Iteration 2746 / 7650) loss: 0.957770
(Iteration 2751 / 7650) loss: 1.021722
(Iteration 2756 / 7650) loss: 1.168507
(Iteration 2761 / 7650) loss: 1.353875

(Iteration 2766 / 7650) loss: 1.220874
(Iteration 2771 / 7650) loss: 0.929521
(Iteration 2776 / 7650) loss: 1.134993
(Iteration 2781 / 7650) loss: 1.258452
(Iteration 2786 / 7650) loss: 1.320597
(Iteration 2791 / 7650) loss: 1.016173
(Iteration 2796 / 7650) loss: 1.060385
(Iteration 2801 / 7650) loss: 1.590611
(Iteration 2806 / 7650) loss: 0.956177
(Iteration 2811 / 7650) loss: 0.933605
(Iteration 2816 / 7650) loss: 1.162163
(Iteration 2821 / 7650) loss: 1.038963
(Iteration 2826 / 7650) loss: 1.019189
(Iteration 2831 / 7650) loss: 1.023802
(Iteration 2836 / 7650) loss: 1.231355
(Iteration 2841 / 7650) loss: 1.049852
(Iteration 2846 / 7650) loss: 1.018657
(Iteration 2851 / 7650) loss: 0.987311
(Iteration 2856 / 7650) loss: 0.933151
(Iteration 2861 / 7650) loss: 1.212414
(Iteration 2866 / 7650) loss: 0.973806
(Iteration 2871 / 7650) loss: 1.191222
(Iteration 2876 / 7650) loss: 0.870379
(Iteration 2881 / 7650) loss: 0.990416
(Iteration 2886 / 7650) loss: 1.136231
(Iteration 2891 / 7650) loss: 0.902905
(Iteration 2896 / 7650) loss: 1.088447
(Iteration 2901 / 7650) loss: 1.282413
(Iteration 2906 / 7650) loss: 1.271526
(Iteration 2911 / 7650) loss: 0.957049
(Iteration 2916 / 7650) loss: 1.084652
(Iteration 2921 / 7650) loss: 1.011786
(Iteration 2926 / 7650) loss: 1.221206
(Iteration 2931 / 7650) loss: 0.893893
(Iteration 2936 / 7650) loss: 0.988184
(Iteration 2941 / 7650) loss: 0.859413
(Iteration 2946 / 7650) loss: 1.059265
(Iteration 2951 / 7650) loss: 0.844607
(Iteration 2956 / 7650) loss: 1.377576
(Iteration 2961 / 7650) loss: 1.222666
(Iteration 2966 / 7650) loss: 0.937122
(Iteration 2971 / 7650) loss: 1.194014
(Iteration 2976 / 7650) loss: 0.835030
(Iteration 2981 / 7650) loss: 1.147894
(Iteration 2986 / 7650) loss: 1.215365
(Iteration 2991 / 7650) loss: 1.119782
(Iteration 2996 / 7650) loss: 1.028927
(Iteration 3001 / 7650) loss: 1.203995
(Iteration 3006 / 7650) loss: 0.990253
(Iteration 3011 / 7650) loss: 1.005157
(Iteration 3016 / 7650) loss: 1.144725
(Iteration 3021 / 7650) loss: 0.956591
(Iteration 3026 / 7650) loss: 0.897234
(Iteration 3031 / 7650) loss: 1.200504
(Iteration 3036 / 7650) loss: 1.083169
(Iteration 3041 / 7650) loss: 1.242543
(Iteration 3046 / 7650) loss: 0.915531
(Iteration 3051 / 7650) loss: 0.961981
(Iteration 3056 / 7650) loss: 1.224002
(Epoch 4 / 10) train acc: 0.702000; val_acc: 0.606000
(Iteration 3061 / 7650) loss: 0.988814
(Iteration 3066 / 7650) loss: 1.226141

(Iteration 3071 / 7650) loss: 1.380902
(Iteration 3076 / 7650) loss: 1.126113
(Iteration 3081 / 7650) loss: 1.162086
(Iteration 3086 / 7650) loss: 1.035033
(Iteration 3091 / 7650) loss: 0.915229
(Iteration 3096 / 7650) loss: 1.027369
(Iteration 3101 / 7650) loss: 1.036532
(Iteration 3106 / 7650) loss: 1.145467
(Iteration 3111 / 7650) loss: 1.017399
(Iteration 3116 / 7650) loss: 0.888312
(Iteration 3121 / 7650) loss: 0.941765
(Iteration 3126 / 7650) loss: 0.811893
(Iteration 3131 / 7650) loss: 1.288258
(Iteration 3136 / 7650) loss: 1.028547
(Iteration 3141 / 7650) loss: 1.105953
(Iteration 3146 / 7650) loss: 0.939300
(Iteration 3151 / 7650) loss: 1.134297
(Iteration 3156 / 7650) loss: 1.017644
(Iteration 3161 / 7650) loss: 1.310795
(Iteration 3166 / 7650) loss: 1.070345
(Iteration 3171 / 7650) loss: 1.129236
(Iteration 3176 / 7650) loss: 1.090240
(Iteration 3181 / 7650) loss: 1.010279
(Iteration 3186 / 7650) loss: 1.020441
(Iteration 3191 / 7650) loss: 1.003695
(Iteration 3196 / 7650) loss: 1.072564
(Iteration 3201 / 7650) loss: 1.250229
(Iteration 3206 / 7650) loss: 1.101404
(Iteration 3211 / 7650) loss: 0.994911
(Iteration 3216 / 7650) loss: 0.922980
(Iteration 3221 / 7650) loss: 1.237880
(Iteration 3226 / 7650) loss: 1.001883
(Iteration 3231 / 7650) loss: 1.193487
(Iteration 3236 / 7650) loss: 1.038907
(Iteration 3241 / 7650) loss: 0.963766
(Iteration 3246 / 7650) loss: 0.899504
(Iteration 3251 / 7650) loss: 1.115458
(Iteration 3256 / 7650) loss: 1.028225
(Iteration 3261 / 7650) loss: 0.998592
(Iteration 3266 / 7650) loss: 1.112409
(Iteration 3271 / 7650) loss: 0.968008
(Iteration 3276 / 7650) loss: 1.078761
(Iteration 3281 / 7650) loss: 1.361964
(Iteration 3286 / 7650) loss: 0.933318
(Iteration 3291 / 7650) loss: 0.799229
(Iteration 3296 / 7650) loss: 0.846767
(Iteration 3301 / 7650) loss: 1.154060
(Iteration 3306 / 7650) loss: 1.100481
(Iteration 3311 / 7650) loss: 1.254655
(Iteration 3316 / 7650) loss: 1.063586
(Iteration 3321 / 7650) loss: 1.217040
(Iteration 3326 / 7650) loss: 0.970940
(Iteration 3331 / 7650) loss: 1.160027
(Iteration 3336 / 7650) loss: 1.074571
(Iteration 3341 / 7650) loss: 1.308852
(Iteration 3346 / 7650) loss: 1.153187
(Iteration 3351 / 7650) loss: 1.007908
(Iteration 3356 / 7650) loss: 1.150420
(Iteration 3361 / 7650) loss: 1.512215
(Iteration 3366 / 7650) loss: 1.078629
(Iteration 3371 / 7650) loss: 0.792685
(Iteration 3376 / 7650) loss: 0.907218

(Iteration 3381 / 7650) loss: 0.992005
(Iteration 3386 / 7650) loss: 1.094194
(Iteration 3391 / 7650) loss: 0.969678
(Iteration 3396 / 7650) loss: 0.902276
(Iteration 3401 / 7650) loss: 1.046948
(Iteration 3406 / 7650) loss: 1.132624
(Iteration 3411 / 7650) loss: 0.973115
(Iteration 3416 / 7650) loss: 1.045027
(Iteration 3421 / 7650) loss: 1.131821
(Iteration 3426 / 7650) loss: 1.125337
(Iteration 3431 / 7650) loss: 0.987418
(Iteration 3436 / 7650) loss: 1.112054
(Iteration 3441 / 7650) loss: 1.043095
(Iteration 3446 / 7650) loss: 0.982582
(Iteration 3451 / 7650) loss: 1.093402
(Iteration 3456 / 7650) loss: 0.943491
(Iteration 3461 / 7650) loss: 0.985861
(Iteration 3466 / 7650) loss: 1.461994
(Iteration 3471 / 7650) loss: 1.005577
(Iteration 3476 / 7650) loss: 0.934558
(Iteration 3481 / 7650) loss: 1.174123
(Iteration 3486 / 7650) loss: 0.956844
(Iteration 3491 / 7650) loss: 1.032630
(Iteration 3496 / 7650) loss: 0.921083
(Iteration 3501 / 7650) loss: 1.263830
(Iteration 3506 / 7650) loss: 1.036538
(Iteration 3511 / 7650) loss: 0.958373
(Iteration 3516 / 7650) loss: 1.069023
(Iteration 3521 / 7650) loss: 0.972558
(Iteration 3526 / 7650) loss: 0.872660
(Iteration 3531 / 7650) loss: 1.117606
(Iteration 3536 / 7650) loss: 1.091181
(Iteration 3541 / 7650) loss: 1.025514
(Iteration 3546 / 7650) loss: 1.071683
(Iteration 3551 / 7650) loss: 1.073133
(Iteration 3556 / 7650) loss: 0.792353
(Iteration 3561 / 7650) loss: 1.048979
(Iteration 3566 / 7650) loss: 0.697136
(Iteration 3571 / 7650) loss: 1.049222
(Iteration 3576 / 7650) loss: 1.131728
(Iteration 3581 / 7650) loss: 0.936757
(Iteration 3586 / 7650) loss: 1.156954
(Iteration 3591 / 7650) loss: 1.018006
(Iteration 3596 / 7650) loss: 1.235166
(Iteration 3601 / 7650) loss: 1.014663
(Iteration 3606 / 7650) loss: 1.104612
(Iteration 3611 / 7650) loss: 0.752908
(Iteration 3616 / 7650) loss: 0.933723
(Iteration 3621 / 7650) loss: 1.059774
(Iteration 3626 / 7650) loss: 0.964656
(Iteration 3631 / 7650) loss: 0.981113
(Iteration 3636 / 7650) loss: 1.151733
(Iteration 3641 / 7650) loss: 0.866082
(Iteration 3646 / 7650) loss: 1.094979
(Iteration 3651 / 7650) loss: 1.397625
(Iteration 3656 / 7650) loss: 0.937994
(Iteration 3661 / 7650) loss: 1.158444
(Iteration 3666 / 7650) loss: 1.067450
(Iteration 3671 / 7650) loss: 1.048342
(Iteration 3676 / 7650) loss: 1.077456
(Iteration 3681 / 7650) loss: 1.130343
(Iteration 3686 / 7650) loss: 1.059170

(Iteration 3691 / 7650) loss: 1.067135
(Iteration 3696 / 7650) loss: 1.073876
(Iteration 3701 / 7650) loss: 1.090909
(Iteration 3706 / 7650) loss: 0.797184
(Iteration 3711 / 7650) loss: 0.865825
(Iteration 3716 / 7650) loss: 0.969578
(Iteration 3721 / 7650) loss: 0.881349
(Iteration 3726 / 7650) loss: 0.986980
(Iteration 3731 / 7650) loss: 1.007591
(Iteration 3736 / 7650) loss: 1.125828
(Iteration 3741 / 7650) loss: 0.852052
(Iteration 3746 / 7650) loss: 0.991665
(Iteration 3751 / 7650) loss: 1.032088
(Iteration 3756 / 7650) loss: 0.919773
(Iteration 3761 / 7650) loss: 1.005680
(Iteration 3766 / 7650) loss: 0.875921
(Iteration 3771 / 7650) loss: 1.151447
(Iteration 3776 / 7650) loss: 1.034292
(Iteration 3781 / 7650) loss: 0.894014
(Iteration 3786 / 7650) loss: 0.992966
(Iteration 3791 / 7650) loss: 1.259974
(Iteration 3796 / 7650) loss: 1.030607
(Iteration 3801 / 7650) loss: 1.002846
(Iteration 3806 / 7650) loss: 1.107178
(Iteration 3811 / 7650) loss: 1.048618
(Iteration 3816 / 7650) loss: 0.787994
(Iteration 3821 / 7650) loss: 1.132724
(Epoch 5 / 10) train acc: 0.719000; val_acc: 0.624000
(Iteration 3826 / 7650) loss: 1.119531
(Iteration 3831 / 7650) loss: 1.319071
(Iteration 3836 / 7650) loss: 0.918526
(Iteration 3841 / 7650) loss: 1.232046
(Iteration 3846 / 7650) loss: 1.209099
(Iteration 3851 / 7650) loss: 1.111681
(Iteration 3856 / 7650) loss: 0.996845
(Iteration 3861 / 7650) loss: 1.111032
(Iteration 3866 / 7650) loss: 0.948995
(Iteration 3871 / 7650) loss: 0.929361
(Iteration 3876 / 7650) loss: 0.778753
(Iteration 3881 / 7650) loss: 0.783553
(Iteration 3886 / 7650) loss: 0.786826
(Iteration 3891 / 7650) loss: 1.151493
(Iteration 3896 / 7650) loss: 1.099025
(Iteration 3901 / 7650) loss: 0.963151
(Iteration 3906 / 7650) loss: 0.973819
(Iteration 3911 / 7650) loss: 0.960535
(Iteration 3916 / 7650) loss: 1.015008
(Iteration 3921 / 7650) loss: 0.939431
(Iteration 3926 / 7650) loss: 1.219633
(Iteration 3931 / 7650) loss: 1.129300
(Iteration 3936 / 7650) loss: 0.985728
(Iteration 3941 / 7650) loss: 0.972272
(Iteration 3946 / 7650) loss: 0.669733
(Iteration 3951 / 7650) loss: 1.092937
(Iteration 3956 / 7650) loss: 1.091940
(Iteration 3961 / 7650) loss: 0.899984
(Iteration 3966 / 7650) loss: 0.984982
(Iteration 3971 / 7650) loss: 0.995393
(Iteration 3976 / 7650) loss: 0.754848
(Iteration 3981 / 7650) loss: 1.361934
(Iteration 3986 / 7650) loss: 0.975032
(Iteration 3991 / 7650) loss: 1.169376

(Iteration 3996 / 7650) loss: 1.063999
(Iteration 4001 / 7650) loss: 1.120854
(Iteration 4006 / 7650) loss: 0.856781
(Iteration 4011 / 7650) loss: 1.081724
(Iteration 4016 / 7650) loss: 0.897660
(Iteration 4021 / 7650) loss: 1.029078
(Iteration 4026 / 7650) loss: 0.948562
(Iteration 4031 / 7650) loss: 0.955184
(Iteration 4036 / 7650) loss: 0.981693
(Iteration 4041 / 7650) loss: 0.923352
(Iteration 4046 / 7650) loss: 0.840840
(Iteration 4051 / 7650) loss: 0.977203
(Iteration 4056 / 7650) loss: 1.131611
(Iteration 4061 / 7650) loss: 0.927604
(Iteration 4066 / 7650) loss: 0.904566
(Iteration 4071 / 7650) loss: 1.083830
(Iteration 4076 / 7650) loss: 1.103302
(Iteration 4081 / 7650) loss: 1.074792
(Iteration 4086 / 7650) loss: 0.798757
(Iteration 4091 / 7650) loss: 1.034902
(Iteration 4096 / 7650) loss: 1.047397
(Iteration 4101 / 7650) loss: 1.215508
(Iteration 4106 / 7650) loss: 1.034837
(Iteration 4111 / 7650) loss: 0.943984
(Iteration 4116 / 7650) loss: 0.983753
(Iteration 4121 / 7650) loss: 1.398468
(Iteration 4126 / 7650) loss: 1.050323
(Iteration 4131 / 7650) loss: 1.045875
(Iteration 4136 / 7650) loss: 1.149072
(Iteration 4141 / 7650) loss: 1.066228
(Iteration 4146 / 7650) loss: 1.116435
(Iteration 4151 / 7650) loss: 0.822380
(Iteration 4156 / 7650) loss: 1.176067
(Iteration 4161 / 7650) loss: 0.867875
(Iteration 4166 / 7650) loss: 1.199875
(Iteration 4171 / 7650) loss: 0.949896
(Iteration 4176 / 7650) loss: 1.118947
(Iteration 4181 / 7650) loss: 1.071119
(Iteration 4186 / 7650) loss: 0.907706
(Iteration 4191 / 7650) loss: 0.947114
(Iteration 4196 / 7650) loss: 1.213632
(Iteration 4201 / 7650) loss: 1.119240
(Iteration 4206 / 7650) loss: 1.002885
(Iteration 4211 / 7650) loss: 0.674520
(Iteration 4216 / 7650) loss: 0.979471
(Iteration 4221 / 7650) loss: 0.962823
(Iteration 4226 / 7650) loss: 0.965958
(Iteration 4231 / 7650) loss: 1.132832
(Iteration 4236 / 7650) loss: 0.831937
(Iteration 4241 / 7650) loss: 1.106615
(Iteration 4246 / 7650) loss: 0.997692
(Iteration 4251 / 7650) loss: 0.873365
(Iteration 4256 / 7650) loss: 0.998082
(Iteration 4261 / 7650) loss: 0.805264
(Iteration 4266 / 7650) loss: 1.127554
(Iteration 4271 / 7650) loss: 1.139359
(Iteration 4276 / 7650) loss: 0.878264
(Iteration 4281 / 7650) loss: 1.119810
(Iteration 4286 / 7650) loss: 1.122383
(Iteration 4291 / 7650) loss: 0.817973
(Iteration 4296 / 7650) loss: 0.993323
(Iteration 4301 / 7650) loss: 0.799945

(Iteration 4306 / 7650) loss: 0.801079
(Iteration 4311 / 7650) loss: 0.714231
(Iteration 4316 / 7650) loss: 0.809688
(Iteration 4321 / 7650) loss: 1.102052
(Iteration 4326 / 7650) loss: 0.802835
(Iteration 4331 / 7650) loss: 0.922868
(Iteration 4336 / 7650) loss: 0.945698
(Iteration 4341 / 7650) loss: 0.959859
(Iteration 4346 / 7650) loss: 0.795896
(Iteration 4351 / 7650) loss: 1.120713
(Iteration 4356 / 7650) loss: 0.773450
(Iteration 4361 / 7650) loss: 0.837210
(Iteration 4366 / 7650) loss: 0.922244
(Iteration 4371 / 7650) loss: 0.716723
(Iteration 4376 / 7650) loss: 0.972347
(Iteration 4381 / 7650) loss: 1.231900
(Iteration 4386 / 7650) loss: 1.073658
(Iteration 4391 / 7650) loss: 0.904367
(Iteration 4396 / 7650) loss: 1.012479
(Iteration 4401 / 7650) loss: 0.951562
(Iteration 4406 / 7650) loss: 0.953178
(Iteration 4411 / 7650) loss: 0.983359
(Iteration 4416 / 7650) loss: 0.835405
(Iteration 4421 / 7650) loss: 0.920105
(Iteration 4426 / 7650) loss: 0.812572
(Iteration 4431 / 7650) loss: 0.847125
(Iteration 4436 / 7650) loss: 0.901186
(Iteration 4441 / 7650) loss: 0.795389
(Iteration 4446 / 7650) loss: 0.912203
(Iteration 4451 / 7650) loss: 0.990513
(Iteration 4456 / 7650) loss: 1.058266
(Iteration 4461 / 7650) loss: 0.695528
(Iteration 4466 / 7650) loss: 0.822700
(Iteration 4471 / 7650) loss: 0.801469
(Iteration 4476 / 7650) loss: 1.142673
(Iteration 4481 / 7650) loss: 0.951789
(Iteration 4486 / 7650) loss: 1.017409
(Iteration 4491 / 7650) loss: 1.045165
(Iteration 4496 / 7650) loss: 0.978172
(Iteration 4501 / 7650) loss: 1.094497
(Iteration 4506 / 7650) loss: 1.090825
(Iteration 4511 / 7650) loss: 1.007913
(Iteration 4516 / 7650) loss: 0.728817
(Iteration 4521 / 7650) loss: 0.920223
(Iteration 4526 / 7650) loss: 0.926656
(Iteration 4531 / 7650) loss: 1.130290
(Iteration 4536 / 7650) loss: 0.851966
(Iteration 4541 / 7650) loss: 0.970117
(Iteration 4546 / 7650) loss: 0.959411
(Iteration 4551 / 7650) loss: 0.814596
(Iteration 4556 / 7650) loss: 1.302119
(Iteration 4561 / 7650) loss: 1.104005
(Iteration 4566 / 7650) loss: 0.930908
(Iteration 4571 / 7650) loss: 0.854833
(Iteration 4576 / 7650) loss: 1.053771
(Iteration 4581 / 7650) loss: 0.910819
(Iteration 4586 / 7650) loss: 0.906284
(Epoch 6 / 10) train acc: 0.749000; val_acc: 0.655000
(Iteration 4591 / 7650) loss: 1.034530
(Iteration 4596 / 7650) loss: 1.037974
(Iteration 4601 / 7650) loss: 1.146596
(Iteration 4606 / 7650) loss: 0.880627

(Iteration 4611 / 7650) loss: 1.114253
(Iteration 4616 / 7650) loss: 1.040068
(Iteration 4621 / 7650) loss: 0.753328
(Iteration 4626 / 7650) loss: 0.918830
(Iteration 4631 / 7650) loss: 1.003143
(Iteration 4636 / 7650) loss: 0.937593
(Iteration 4641 / 7650) loss: 1.090657
(Iteration 4646 / 7650) loss: 0.949098
(Iteration 4651 / 7650) loss: 0.911336
(Iteration 4656 / 7650) loss: 1.050970
(Iteration 4661 / 7650) loss: 0.981368
(Iteration 4666 / 7650) loss: 1.118431
(Iteration 4671 / 7650) loss: 1.004181
(Iteration 4676 / 7650) loss: 1.283425
(Iteration 4681 / 7650) loss: 1.009340
(Iteration 4686 / 7650) loss: 1.122911
(Iteration 4691 / 7650) loss: 0.844424
(Iteration 4696 / 7650) loss: 0.814014
(Iteration 4701 / 7650) loss: 0.732410
(Iteration 4706 / 7650) loss: 0.883654
(Iteration 4711 / 7650) loss: 1.004272
(Iteration 4716 / 7650) loss: 0.661620
(Iteration 4721 / 7650) loss: 1.108447
(Iteration 4726 / 7650) loss: 1.113001
(Iteration 4731 / 7650) loss: 0.914178
(Iteration 4736 / 7650) loss: 0.872357
(Iteration 4741 / 7650) loss: 0.937024
(Iteration 4746 / 7650) loss: 0.864100
(Iteration 4751 / 7650) loss: 0.975750
(Iteration 4756 / 7650) loss: 0.806148
(Iteration 4761 / 7650) loss: 0.942822
(Iteration 4766 / 7650) loss: 0.907516
(Iteration 4771 / 7650) loss: 1.161338
(Iteration 4776 / 7650) loss: 0.934137
(Iteration 4781 / 7650) loss: 1.053592
(Iteration 4786 / 7650) loss: 0.942416
(Iteration 4791 / 7650) loss: 0.837450
(Iteration 4796 / 7650) loss: 1.179631
(Iteration 4801 / 7650) loss: 1.055681
(Iteration 4806 / 7650) loss: 0.978699
(Iteration 4811 / 7650) loss: 0.902664
(Iteration 4816 / 7650) loss: 1.269640
(Iteration 4821 / 7650) loss: 1.107321
(Iteration 4826 / 7650) loss: 0.882100
(Iteration 4831 / 7650) loss: 1.218091
(Iteration 4836 / 7650) loss: 1.104362
(Iteration 4841 / 7650) loss: 0.962383
(Iteration 4846 / 7650) loss: 0.850810
(Iteration 4851 / 7650) loss: 1.068747
(Iteration 4856 / 7650) loss: 0.824147
(Iteration 4861 / 7650) loss: 0.935876
(Iteration 4866 / 7650) loss: 0.919246
(Iteration 4871 / 7650) loss: 0.864930
(Iteration 4876 / 7650) loss: 0.945641
(Iteration 4881 / 7650) loss: 0.881856
(Iteration 4886 / 7650) loss: 1.103199
(Iteration 4891 / 7650) loss: 0.962862
(Iteration 4896 / 7650) loss: 0.774639
(Iteration 4901 / 7650) loss: 0.859330
(Iteration 4906 / 7650) loss: 0.957979
(Iteration 4911 / 7650) loss: 0.884047
(Iteration 4916 / 7650) loss: 0.994283

(Iteration 4921 / 7650) loss: 1.275586
(Iteration 4926 / 7650) loss: 1.329400
(Iteration 4931 / 7650) loss: 1.014047
(Iteration 4936 / 7650) loss: 0.890193
(Iteration 4941 / 7650) loss: 0.856904
(Iteration 4946 / 7650) loss: 0.887784
(Iteration 4951 / 7650) loss: 0.724929
(Iteration 4956 / 7650) loss: 0.776992
(Iteration 4961 / 7650) loss: 0.935056
(Iteration 4966 / 7650) loss: 1.107600
(Iteration 4971 / 7650) loss: 0.734219
(Iteration 4976 / 7650) loss: 0.876062
(Iteration 4981 / 7650) loss: 1.050311
(Iteration 4986 / 7650) loss: 0.874001
(Iteration 4991 / 7650) loss: 1.042894
(Iteration 4996 / 7650) loss: 0.999891
(Iteration 5001 / 7650) loss: 1.098141
(Iteration 5006 / 7650) loss: 0.940936
(Iteration 5011 / 7650) loss: 1.144342
(Iteration 5016 / 7650) loss: 1.126635
(Iteration 5021 / 7650) loss: 0.901201
(Iteration 5026 / 7650) loss: 0.727981
(Iteration 5031 / 7650) loss: 0.929783
(Iteration 5036 / 7650) loss: 0.795858
(Iteration 5041 / 7650) loss: 1.215855
(Iteration 5046 / 7650) loss: 0.799818
(Iteration 5051 / 7650) loss: 1.047088
(Iteration 5056 / 7650) loss: 1.002952
(Iteration 5061 / 7650) loss: 1.179395
(Iteration 5066 / 7650) loss: 0.761305
(Iteration 5071 / 7650) loss: 0.785186
(Iteration 5076 / 7650) loss: 0.699591
(Iteration 5081 / 7650) loss: 1.037474
(Iteration 5086 / 7650) loss: 1.206541
(Iteration 5091 / 7650) loss: 1.058228
(Iteration 5096 / 7650) loss: 1.020658
(Iteration 5101 / 7650) loss: 0.705285
(Iteration 5106 / 7650) loss: 1.002631
(Iteration 5111 / 7650) loss: 0.820246
(Iteration 5116 / 7650) loss: 0.772493
(Iteration 5121 / 7650) loss: 0.624291
(Iteration 5126 / 7650) loss: 0.846109
(Iteration 5131 / 7650) loss: 1.034328
(Iteration 5136 / 7650) loss: 0.887927
(Iteration 5141 / 7650) loss: 0.807928
(Iteration 5146 / 7650) loss: 0.978068
(Iteration 5151 / 7650) loss: 0.749252
(Iteration 5156 / 7650) loss: 0.850072
(Iteration 5161 / 7650) loss: 0.872567
(Iteration 5166 / 7650) loss: 1.160183
(Iteration 5171 / 7650) loss: 0.816041
(Iteration 5176 / 7650) loss: 0.804997
(Iteration 5181 / 7650) loss: 0.810281
(Iteration 5186 / 7650) loss: 0.848384
(Iteration 5191 / 7650) loss: 0.894287
(Iteration 5196 / 7650) loss: 0.866775
(Iteration 5201 / 7650) loss: 0.850290
(Iteration 5206 / 7650) loss: 0.797132
(Iteration 5211 / 7650) loss: 1.063247
(Iteration 5216 / 7650) loss: 0.898258
(Iteration 5221 / 7650) loss: 0.821341
(Iteration 5226 / 7650) loss: 1.090518

(Iteration 5231 / 7650) loss: 0.852370
(Iteration 5236 / 7650) loss: 1.131319
(Iteration 5241 / 7650) loss: 0.817538
(Iteration 5246 / 7650) loss: 0.871199
(Iteration 5251 / 7650) loss: 0.854653
(Iteration 5256 / 7650) loss: 0.683949
(Iteration 5261 / 7650) loss: 0.895571
(Iteration 5266 / 7650) loss: 0.891792
(Iteration 5271 / 7650) loss: 0.976546
(Iteration 5276 / 7650) loss: 1.204397
(Iteration 5281 / 7650) loss: 1.142124
(Iteration 5286 / 7650) loss: 0.617870
(Iteration 5291 / 7650) loss: 1.091273
(Iteration 5296 / 7650) loss: 0.789452
(Iteration 5301 / 7650) loss: 0.884881
(Iteration 5306 / 7650) loss: 0.947815
(Iteration 5311 / 7650) loss: 1.001278
(Iteration 5316 / 7650) loss: 1.024247
(Iteration 5321 / 7650) loss: 1.157923
(Iteration 5326 / 7650) loss: 0.757644
(Iteration 5331 / 7650) loss: 0.689988
(Iteration 5336 / 7650) loss: 0.775844
(Iteration 5341 / 7650) loss: 1.024965
(Iteration 5346 / 7650) loss: 0.834142
(Iteration 5351 / 7650) loss: 1.127271
(Epoch 7 / 10) train acc: 0.744000; val_acc: 0.641000
(Iteration 5356 / 7650) loss: 0.847653
(Iteration 5361 / 7650) loss: 1.174722
(Iteration 5366 / 7650) loss: 0.874137
(Iteration 5371 / 7650) loss: 1.084393
(Iteration 5376 / 7650) loss: 0.881804
(Iteration 5381 / 7650) loss: 1.125402
(Iteration 5386 / 7650) loss: 1.047553
(Iteration 5391 / 7650) loss: 0.803034
(Iteration 5396 / 7650) loss: 0.759249
(Iteration 5401 / 7650) loss: 1.112946
(Iteration 5406 / 7650) loss: 1.095683
(Iteration 5411 / 7650) loss: 1.102651
(Iteration 5416 / 7650) loss: 0.961539
(Iteration 5421 / 7650) loss: 0.967932
(Iteration 5426 / 7650) loss: 1.033955
(Iteration 5431 / 7650) loss: 0.930418
(Iteration 5436 / 7650) loss: 1.092703
(Iteration 5441 / 7650) loss: 1.022916
(Iteration 5446 / 7650) loss: 0.902029
(Iteration 5451 / 7650) loss: 0.955619
(Iteration 5456 / 7650) loss: 1.255713
(Iteration 5461 / 7650) loss: 1.008829
(Iteration 5466 / 7650) loss: 0.742583
(Iteration 5471 / 7650) loss: 0.856546
(Iteration 5476 / 7650) loss: 0.804726
(Iteration 5481 / 7650) loss: 0.941465
(Iteration 5486 / 7650) loss: 0.812943
(Iteration 5491 / 7650) loss: 1.110736
(Iteration 5496 / 7650) loss: 0.897096
(Iteration 5501 / 7650) loss: 1.043620
(Iteration 5506 / 7650) loss: 0.793986
(Iteration 5511 / 7650) loss: 0.885176
(Iteration 5516 / 7650) loss: 0.862641
(Iteration 5521 / 7650) loss: 1.057606
(Iteration 5526 / 7650) loss: 0.839601
(Iteration 5531 / 7650) loss: 1.033437

(Iteration 5536 / 7650) loss: 1.079661
(Iteration 5541 / 7650) loss: 1.004549
(Iteration 5546 / 7650) loss: 0.980076
(Iteration 5551 / 7650) loss: 1.180764
(Iteration 5556 / 7650) loss: 0.925918
(Iteration 5561 / 7650) loss: 1.005378
(Iteration 5566 / 7650) loss: 0.913331
(Iteration 5571 / 7650) loss: 1.202896
(Iteration 5576 / 7650) loss: 0.904018
(Iteration 5581 / 7650) loss: 0.791878
(Iteration 5586 / 7650) loss: 1.146078
(Iteration 5591 / 7650) loss: 0.823149
(Iteration 5596 / 7650) loss: 1.031429
(Iteration 5601 / 7650) loss: 0.794190
(Iteration 5606 / 7650) loss: 0.909533
(Iteration 5611 / 7650) loss: 0.793819
(Iteration 5616 / 7650) loss: 1.023496
(Iteration 5621 / 7650) loss: 0.978255
(Iteration 5626 / 7650) loss: 1.081334
(Iteration 5631 / 7650) loss: 0.843245
(Iteration 5636 / 7650) loss: 0.837736
(Iteration 5641 / 7650) loss: 0.809896
(Iteration 5646 / 7650) loss: 1.090820
(Iteration 5651 / 7650) loss: 0.938732
(Iteration 5656 / 7650) loss: 0.989972
(Iteration 5661 / 7650) loss: 0.961149
(Iteration 5666 / 7650) loss: 1.346638
(Iteration 5671 / 7650) loss: 1.045412
(Iteration 5676 / 7650) loss: 1.111767
(Iteration 5681 / 7650) loss: 0.797939
(Iteration 5686 / 7650) loss: 0.744400
(Iteration 5691 / 7650) loss: 1.096192
(Iteration 5696 / 7650) loss: 1.042614
(Iteration 5701 / 7650) loss: 0.632439
(Iteration 5706 / 7650) loss: 1.014177
(Iteration 5711 / 7650) loss: 1.281049
(Iteration 5716 / 7650) loss: 0.821826
(Iteration 5721 / 7650) loss: 0.952409
(Iteration 5726 / 7650) loss: 0.794481
(Iteration 5731 / 7650) loss: 0.983021
(Iteration 5736 / 7650) loss: 0.863976
(Iteration 5741 / 7650) loss: 0.941988
(Iteration 5746 / 7650) loss: 1.146642
(Iteration 5751 / 7650) loss: 1.007078
(Iteration 5756 / 7650) loss: 0.767256
(Iteration 5761 / 7650) loss: 1.010224
(Iteration 5766 / 7650) loss: 0.765008
(Iteration 5771 / 7650) loss: 0.885441
(Iteration 5776 / 7650) loss: 0.736470
(Iteration 5781 / 7650) loss: 1.071183
(Iteration 5786 / 7650) loss: 1.250667
(Iteration 5791 / 7650) loss: 0.881653
(Iteration 5796 / 7650) loss: 1.051918
(Iteration 5801 / 7650) loss: 0.929951
(Iteration 5806 / 7650) loss: 0.830579
(Iteration 5811 / 7650) loss: 0.813488
(Iteration 5816 / 7650) loss: 0.734540
(Iteration 5821 / 7650) loss: 0.855702
(Iteration 5826 / 7650) loss: 0.806663
(Iteration 5831 / 7650) loss: 0.847400
(Iteration 5836 / 7650) loss: 0.745070
(Iteration 5841 / 7650) loss: 1.004854

(Iteration 5846 / 7650) loss: 0.824121
(Iteration 5851 / 7650) loss: 1.003410
(Iteration 5856 / 7650) loss: 1.021688
(Iteration 5861 / 7650) loss: 0.975350
(Iteration 5866 / 7650) loss: 0.787144
(Iteration 5871 / 7650) loss: 1.003697
(Iteration 5876 / 7650) loss: 1.037274
(Iteration 5881 / 7650) loss: 1.097536
(Iteration 5886 / 7650) loss: 1.023650
(Iteration 5891 / 7650) loss: 1.007140
(Iteration 5896 / 7650) loss: 0.961851
(Iteration 5901 / 7650) loss: 0.818600
(Iteration 5906 / 7650) loss: 0.974332
(Iteration 5911 / 7650) loss: 0.953236
(Iteration 5916 / 7650) loss: 1.121268
(Iteration 5921 / 7650) loss: 0.848476
(Iteration 5926 / 7650) loss: 0.884801
(Iteration 5931 / 7650) loss: 1.059191
(Iteration 5936 / 7650) loss: 1.098799
(Iteration 5941 / 7650) loss: 0.888124
(Iteration 5946 / 7650) loss: 0.706747
(Iteration 5951 / 7650) loss: 0.897210
(Iteration 5956 / 7650) loss: 0.730707
(Iteration 5961 / 7650) loss: 1.078671
(Iteration 5966 / 7650) loss: 1.000980
(Iteration 5971 / 7650) loss: 1.055734
(Iteration 5976 / 7650) loss: 1.004905
(Iteration 5981 / 7650) loss: 0.987444
(Iteration 5986 / 7650) loss: 0.718054
(Iteration 5991 / 7650) loss: 0.918194
(Iteration 5996 / 7650) loss: 0.837709
(Iteration 6001 / 7650) loss: 1.093720
(Iteration 6006 / 7650) loss: 1.010523
(Iteration 6011 / 7650) loss: 0.849212
(Iteration 6016 / 7650) loss: 0.931662
(Iteration 6021 / 7650) loss: 0.748531
(Iteration 6026 / 7650) loss: 0.889718
(Iteration 6031 / 7650) loss: 0.897993
(Iteration 6036 / 7650) loss: 0.926982
(Iteration 6041 / 7650) loss: 0.821571
(Iteration 6046 / 7650) loss: 0.855060
(Iteration 6051 / 7650) loss: 0.974855
(Iteration 6056 / 7650) loss: 1.172303
(Iteration 6061 / 7650) loss: 1.194977
(Iteration 6066 / 7650) loss: 0.712328
(Iteration 6071 / 7650) loss: 0.831021
(Iteration 6076 / 7650) loss: 1.151566
(Iteration 6081 / 7650) loss: 1.018957
(Iteration 6086 / 7650) loss: 0.754916
(Iteration 6091 / 7650) loss: 0.850507
(Iteration 6096 / 7650) loss: 0.930577
(Iteration 6101 / 7650) loss: 1.188617
(Iteration 6106 / 7650) loss: 0.925805
(Iteration 6111 / 7650) loss: 0.909608
(Iteration 6116 / 7650) loss: 0.786739
(Epoch 8 / 10) train acc: 0.750000; val_acc: 0.641000
(Iteration 6121 / 7650) loss: 0.968910
(Iteration 6126 / 7650) loss: 0.764008
(Iteration 6131 / 7650) loss: 1.047404
(Iteration 6136 / 7650) loss: 0.790796
(Iteration 6141 / 7650) loss: 0.743568
(Iteration 6146 / 7650) loss: 0.893426

(Iteration 6151 / 7650) loss: 0.973601
(Iteration 6156 / 7650) loss: 1.065623
(Iteration 6161 / 7650) loss: 0.756333
(Iteration 6166 / 7650) loss: 0.929663
(Iteration 6171 / 7650) loss: 0.860881
(Iteration 6176 / 7650) loss: 0.687132
(Iteration 6181 / 7650) loss: 0.984157
(Iteration 6186 / 7650) loss: 0.838385
(Iteration 6191 / 7650) loss: 0.721745
(Iteration 6196 / 7650) loss: 0.926667
(Iteration 6201 / 7650) loss: 1.067619
(Iteration 6206 / 7650) loss: 0.761957
(Iteration 6211 / 7650) loss: 0.743149
(Iteration 6216 / 7650) loss: 0.814465
(Iteration 6221 / 7650) loss: 0.997282
(Iteration 6226 / 7650) loss: 0.513990
(Iteration 6231 / 7650) loss: 0.894327
(Iteration 6236 / 7650) loss: 1.030080
(Iteration 6241 / 7650) loss: 0.963654
(Iteration 6246 / 7650) loss: 0.729998
(Iteration 6251 / 7650) loss: 1.052758
(Iteration 6256 / 7650) loss: 0.806410
(Iteration 6261 / 7650) loss: 1.116019
(Iteration 6266 / 7650) loss: 1.039943
(Iteration 6271 / 7650) loss: 1.003045
(Iteration 6276 / 7650) loss: 0.810958
(Iteration 6281 / 7650) loss: 0.769106
(Iteration 6286 / 7650) loss: 1.072090
(Iteration 6291 / 7650) loss: 0.790915
(Iteration 6296 / 7650) loss: 0.802934
(Iteration 6301 / 7650) loss: 0.961291
(Iteration 6306 / 7650) loss: 0.984209
(Iteration 6311 / 7650) loss: 0.816225
(Iteration 6316 / 7650) loss: 0.873543
(Iteration 6321 / 7650) loss: 0.989796
(Iteration 6326 / 7650) loss: 0.790876
(Iteration 6331 / 7650) loss: 0.702976
(Iteration 6336 / 7650) loss: 0.701438
(Iteration 6341 / 7650) loss: 0.985738
(Iteration 6346 / 7650) loss: 0.946820
(Iteration 6351 / 7650) loss: 0.815861
(Iteration 6356 / 7650) loss: 0.863079
(Iteration 6361 / 7650) loss: 0.758395
(Iteration 6366 / 7650) loss: 0.817906
(Iteration 6371 / 7650) loss: 0.932472
(Iteration 6376 / 7650) loss: 1.010971
(Iteration 6381 / 7650) loss: 1.018193
(Iteration 6386 / 7650) loss: 1.037951
(Iteration 6391 / 7650) loss: 0.960297
(Iteration 6396 / 7650) loss: 0.882195
(Iteration 6401 / 7650) loss: 0.787210
(Iteration 6406 / 7650) loss: 0.925728
(Iteration 6411 / 7650) loss: 1.040486
(Iteration 6416 / 7650) loss: 0.891415
(Iteration 6421 / 7650) loss: 1.168179
(Iteration 6426 / 7650) loss: 0.969718
(Iteration 6431 / 7650) loss: 0.862338
(Iteration 6436 / 7650) loss: 0.838186
(Iteration 6441 / 7650) loss: 0.899441
(Iteration 6446 / 7650) loss: 0.879075
(Iteration 6451 / 7650) loss: 0.939537
(Iteration 6456 / 7650) loss: 0.893248

(Iteration 6461 / 7650) loss: 0.933758
(Iteration 6466 / 7650) loss: 0.733723
(Iteration 6471 / 7650) loss: 0.731828
(Iteration 6476 / 7650) loss: 0.837188
(Iteration 6481 / 7650) loss: 0.979112
(Iteration 6486 / 7650) loss: 0.783179
(Iteration 6491 / 7650) loss: 0.996579
(Iteration 6496 / 7650) loss: 1.130551
(Iteration 6501 / 7650) loss: 0.867764
(Iteration 6506 / 7650) loss: 0.915004
(Iteration 6511 / 7650) loss: 0.698659
(Iteration 6516 / 7650) loss: 0.794851
(Iteration 6521 / 7650) loss: 1.029955
(Iteration 6526 / 7650) loss: 0.681788
(Iteration 6531 / 7650) loss: 0.826876
(Iteration 6536 / 7650) loss: 0.697807
(Iteration 6541 / 7650) loss: 0.884881
(Iteration 6546 / 7650) loss: 0.766150
(Iteration 6551 / 7650) loss: 0.952183
(Iteration 6556 / 7650) loss: 0.913798
(Iteration 6561 / 7650) loss: 1.364332
(Iteration 6566 / 7650) loss: 0.632178
(Iteration 6571 / 7650) loss: 0.776308
(Iteration 6576 / 7650) loss: 0.933718
(Iteration 6581 / 7650) loss: 0.829758
(Iteration 6586 / 7650) loss: 0.928771
(Iteration 6591 / 7650) loss: 0.709342
(Iteration 6596 / 7650) loss: 0.829610
(Iteration 6601 / 7650) loss: 0.951887
(Iteration 6606 / 7650) loss: 0.846299
(Iteration 6611 / 7650) loss: 0.766542
(Iteration 6616 / 7650) loss: 0.952757
(Iteration 6621 / 7650) loss: 0.761650
(Iteration 6626 / 7650) loss: 0.857484
(Iteration 6631 / 7650) loss: 0.803973
(Iteration 6636 / 7650) loss: 1.088276
(Iteration 6641 / 7650) loss: 1.051005
(Iteration 6646 / 7650) loss: 0.784539
(Iteration 6651 / 7650) loss: 0.832603
(Iteration 6656 / 7650) loss: 0.838715
(Iteration 6661 / 7650) loss: 0.890883
(Iteration 6666 / 7650) loss: 0.949051
(Iteration 6671 / 7650) loss: 0.697968
(Iteration 6676 / 7650) loss: 1.031010
(Iteration 6681 / 7650) loss: 0.862416
(Iteration 6686 / 7650) loss: 0.849737
(Iteration 6691 / 7650) loss: 0.921891
(Iteration 6696 / 7650) loss: 0.853455
(Iteration 6701 / 7650) loss: 1.058424
(Iteration 6706 / 7650) loss: 0.799920
(Iteration 6711 / 7650) loss: 0.789857
(Iteration 6716 / 7650) loss: 0.832347
(Iteration 6721 / 7650) loss: 0.854852
(Iteration 6726 / 7650) loss: 0.872292
(Iteration 6731 / 7650) loss: 1.046295
(Iteration 6736 / 7650) loss: 1.371388
(Iteration 6741 / 7650) loss: 1.042325
(Iteration 6746 / 7650) loss: 0.706692
(Iteration 6751 / 7650) loss: 0.815128
(Iteration 6756 / 7650) loss: 0.772008
(Iteration 6761 / 7650) loss: 0.929504
(Iteration 6766 / 7650) loss: 0.824441

(Iteration 6771 / 7650) loss: 0.939831
(Iteration 6776 / 7650) loss: 0.877077
(Iteration 6781 / 7650) loss: 0.667176
(Iteration 6786 / 7650) loss: 0.781347
(Iteration 6791 / 7650) loss: 0.910064
(Iteration 6796 / 7650) loss: 0.669835
(Iteration 6801 / 7650) loss: 0.979942
(Iteration 6806 / 7650) loss: 0.873475
(Iteration 6811 / 7650) loss: 0.819158
(Iteration 6816 / 7650) loss: 0.863716
(Iteration 6821 / 7650) loss: 0.698598
(Iteration 6826 / 7650) loss: 0.808015
(Iteration 6831 / 7650) loss: 0.920493
(Iteration 6836 / 7650) loss: 0.723782
(Iteration 6841 / 7650) loss: 0.906566
(Iteration 6846 / 7650) loss: 0.886715
(Iteration 6851 / 7650) loss: 0.836407
(Iteration 6856 / 7650) loss: 0.890443
(Iteration 6861 / 7650) loss: 1.000430
(Iteration 6866 / 7650) loss: 0.801113
(Iteration 6871 / 7650) loss: 0.786138
(Iteration 6876 / 7650) loss: 1.086929
(Iteration 6881 / 7650) loss: 0.648797
(Epoch 9 / 10) train acc: 0.736000; val_acc: 0.659000
(Iteration 6886 / 7650) loss: 0.923479
(Iteration 6891 / 7650) loss: 0.756088
(Iteration 6896 / 7650) loss: 1.102457
(Iteration 6901 / 7650) loss: 0.854649
(Iteration 6906 / 7650) loss: 1.084361
(Iteration 6911 / 7650) loss: 1.004428
(Iteration 6916 / 7650) loss: 1.007691
(Iteration 6921 / 7650) loss: 0.935492
(Iteration 6926 / 7650) loss: 0.966669
(Iteration 6931 / 7650) loss: 0.912309
(Iteration 6936 / 7650) loss: 0.878685
(Iteration 6941 / 7650) loss: 1.113317
(Iteration 6946 / 7650) loss: 0.754423
(Iteration 6951 / 7650) loss: 0.928566
(Iteration 6956 / 7650) loss: 0.748616
(Iteration 6961 / 7650) loss: 0.803493
(Iteration 6966 / 7650) loss: 1.055440
(Iteration 6971 / 7650) loss: 0.984673
(Iteration 6976 / 7650) loss: 1.108104
(Iteration 6981 / 7650) loss: 0.966659
(Iteration 6986 / 7650) loss: 0.927594
(Iteration 6991 / 7650) loss: 0.887696
(Iteration 6996 / 7650) loss: 1.002929
(Iteration 7001 / 7650) loss: 0.945741
(Iteration 7006 / 7650) loss: 0.896464
(Iteration 7011 / 7650) loss: 1.016592
(Iteration 7016 / 7650) loss: 0.695354
(Iteration 7021 / 7650) loss: 0.814057
(Iteration 7026 / 7650) loss: 0.925694
(Iteration 7031 / 7650) loss: 0.795410
(Iteration 7036 / 7650) loss: 1.187458
(Iteration 7041 / 7650) loss: 0.807720
(Iteration 7046 / 7650) loss: 0.564385
(Iteration 7051 / 7650) loss: 0.690284
(Iteration 7056 / 7650) loss: 0.670266
(Iteration 7061 / 7650) loss: 0.614490
(Iteration 7066 / 7650) loss: 0.796880
(Iteration 7071 / 7650) loss: 0.919717

(Iteration 7076 / 7650) loss: 0.880772
(Iteration 7081 / 7650) loss: 1.032464
(Iteration 7086 / 7650) loss: 0.962422
(Iteration 7091 / 7650) loss: 1.018697
(Iteration 7096 / 7650) loss: 0.737704
(Iteration 7101 / 7650) loss: 0.796735
(Iteration 7106 / 7650) loss: 1.016400
(Iteration 7111 / 7650) loss: 0.780009
(Iteration 7116 / 7650) loss: 0.940736
(Iteration 7121 / 7650) loss: 0.951924
(Iteration 7126 / 7650) loss: 0.711871
(Iteration 7131 / 7650) loss: 0.999363
(Iteration 7136 / 7650) loss: 0.746478
(Iteration 7141 / 7650) loss: 0.733494
(Iteration 7146 / 7650) loss: 0.764567
(Iteration 7151 / 7650) loss: 0.813701
(Iteration 7156 / 7650) loss: 0.957625
(Iteration 7161 / 7650) loss: 0.905827
(Iteration 7166 / 7650) loss: 0.675281
(Iteration 7171 / 7650) loss: 0.910644
(Iteration 7176 / 7650) loss: 0.796410
(Iteration 7181 / 7650) loss: 0.849823
(Iteration 7186 / 7650) loss: 0.746318
(Iteration 7191 / 7650) loss: 0.909943
(Iteration 7196 / 7650) loss: 0.784883
(Iteration 7201 / 7650) loss: 0.900365
(Iteration 7206 / 7650) loss: 0.784788
(Iteration 7211 / 7650) loss: 0.766659
(Iteration 7216 / 7650) loss: 0.857364
(Iteration 7221 / 7650) loss: 0.692567
(Iteration 7226 / 7650) loss: 0.782678
(Iteration 7231 / 7650) loss: 0.838976
(Iteration 7236 / 7650) loss: 0.785890
(Iteration 7241 / 7650) loss: 1.204396
(Iteration 7246 / 7650) loss: 0.741334
(Iteration 7251 / 7650) loss: 1.007311
(Iteration 7256 / 7650) loss: 0.979191
(Iteration 7261 / 7650) loss: 0.724051
(Iteration 7266 / 7650) loss: 0.844154
(Iteration 7271 / 7650) loss: 0.814851
(Iteration 7276 / 7650) loss: 0.832461
(Iteration 7281 / 7650) loss: 1.037421
(Iteration 7286 / 7650) loss: 0.895831
(Iteration 7291 / 7650) loss: 0.782810
(Iteration 7296 / 7650) loss: 1.091444
(Iteration 7301 / 7650) loss: 0.825727
(Iteration 7306 / 7650) loss: 1.253874
(Iteration 7311 / 7650) loss: 0.848480
(Iteration 7316 / 7650) loss: 0.768647
(Iteration 7321 / 7650) loss: 0.816646
(Iteration 7326 / 7650) loss: 1.036183
(Iteration 7331 / 7650) loss: 0.936061
(Iteration 7336 / 7650) loss: 0.855017
(Iteration 7341 / 7650) loss: 0.671844
(Iteration 7346 / 7650) loss: 0.647321
(Iteration 7351 / 7650) loss: 0.847113
(Iteration 7356 / 7650) loss: 0.913386
(Iteration 7361 / 7650) loss: 1.056947
(Iteration 7366 / 7650) loss: 0.798233
(Iteration 7371 / 7650) loss: 0.698391
(Iteration 7376 / 7650) loss: 0.830800
(Iteration 7381 / 7650) loss: 0.858760

(Iteration 7386 / 7650) loss: 1.061773
(Iteration 7391 / 7650) loss: 0.968609
(Iteration 7396 / 7650) loss: 1.050600
(Iteration 7401 / 7650) loss: 0.842013
(Iteration 7406 / 7650) loss: 1.038919
(Iteration 7411 / 7650) loss: 1.060614
(Iteration 7416 / 7650) loss: 0.763195
(Iteration 7421 / 7650) loss: 0.973167
(Iteration 7426 / 7650) loss: 1.169730
(Iteration 7431 / 7650) loss: 0.771601
(Iteration 7436 / 7650) loss: 0.928651
(Iteration 7441 / 7650) loss: 0.848787
(Iteration 7446 / 7650) loss: 1.061931
(Iteration 7451 / 7650) loss: 0.842870
(Iteration 7456 / 7650) loss: 0.783769
(Iteration 7461 / 7650) loss: 0.892387
(Iteration 7466 / 7650) loss: 0.925750
(Iteration 7471 / 7650) loss: 1.018064
(Iteration 7476 / 7650) loss: 0.805533
(Iteration 7481 / 7650) loss: 0.817141
(Iteration 7486 / 7650) loss: 0.914933
(Iteration 7491 / 7650) loss: 0.984035
(Iteration 7496 / 7650) loss: 0.769249
(Iteration 7556 / 7650) loss: 0.819417
(Iteration 7561 / 7650) loss: 1.000325
(Iteration 7566 / 7650) loss: 0.864301
(Iteration 7571 / 7650) loss: 0.934848
(Iteration 7576 / 7650) loss: 0.910115
(Iteration 7581 / 7650) loss: 0.754571
(Iteration 7586 / 7650) loss: 0.764230
(Iteration 7591 / 7650) loss: 0.909455
(Iteration 7596 / 7650) loss: 0.846797
(Iteration 7601 / 7650) loss: 0.810347
(Iteration 7606 / 7650) loss: 0.695130
(Iteration 7611 / 7650) loss: 1.096688
(Iteration 7616 / 7650) loss: 1.054058
(Iteration 7621 / 7650) loss: 1.194617
(Iteration 7626 / 7650) loss: 0.915988
(Iteration 7631 / 7650) loss: 0.820150
(Iteration 7636 / 7650) loss: 0.722024
(Iteration 7641 / 7650) loss: 0.920464
(Iteration 7646 / 7650) loss: 1.321097
(Epoch 10 / 10) train acc: 0.783000; val_acc: 0.669000