

# Machine Learning: Assignment #3

## Fall 2019

**Due:** Oct 31th, 23:59:59 CST (UTC +8).

### 1. Neural Networks

The codes of this section are in the *neural\_networks* folder.

In this problem, we will implement the feedforward and backpropagation process of the neural networks. We will use *digital.mat* as our experiment data. Finish *fullyconnect\_feedforward*, *fullyconnect\_backprop*, *relu\_feedforward*, *relu\_backprop* and the testing part in *run.m/run.ipynb*. Then we can train three layer (data, hidden-relu, loss) neural networks and report test accuracy.

Supplementary Knowledges:

- i) Instead of using MSE loss function, we adopt softmax loss function here. Recall that in Assignment #2, we used logistic regression to classify two classes. And softmax regression model is a model that extends logistic regression to classify more classes than two <sup>1</sup>. In this problem, we have 10 classes. The softmax loss part codes are done.
- ii) We can use *gradient\_check.m/gradient\_check.py* to check the correctness your computation. If  $\frac{d}{d\theta}J(\theta) = \lim_{\epsilon \rightarrow 0} \frac{J(\theta+\epsilon) - J(\theta)}{\epsilon}$  holds, then we are in the right way. <sup>2</sup>
- iii) Weight decay and momentum are used to update weight paramters in *get\_new\_weight\_inc.m/get\_new\_weight\_inc.py*.

### 2. K-Nearest Neighbor

The codes of this section are in the *knn* folder.

In this problem, we will play with K-Nearest Neighbor (KNN) algorithm and try it on real-world data. Implement KNN algorithm (in *knn.m/knn.py*), then answer the following questions.

- (a) In *knn\_exp.m/knn\_exp.ipynb*, try KNN with different K (you should at least experiment  $K = 1, 10$  and  $100$ ) and plot the decision boundary.

You are encouraged to vectorize<sup>34</sup> your code, otherwise the experiment time might be extremely long. You may find the MATLAB build-in functions *pdist2*, *sort*, *max* and *hist* useful. Also, you can use the function *eudist2*<sup>5</sup> written by Prof. Deng Cai<sup>6</sup>.

<sup>1</sup>[http://ufldl.stanford.edu/wiki/index.php/Softmax\\_Regression](http://ufldl.stanford.edu/wiki/index.php/Softmax_Regression)

<sup>2</sup>[http://ufldl.stanford.edu/wiki/index.php/Gradient\\_checking\\_and\\_advanced\\_optimization](http://ufldl.stanford.edu/wiki/index.php/Gradient_checking_and_advanced_optimization)

<sup>3</sup>[http://www.mathworks.cn/cn/help/matlab/matlab\\_prog/vectorization.html](http://www.mathworks.cn/cn/help/matlab/matlab_prog/vectorization.html)

<sup>4</sup><https://stackoverflow.com/questions/47755442/what-is-vectorization>

<sup>5</sup><http://www.cad.zju.edu.cn/home/dengcai/Data/code/EuDist2.m>

<sup>6</sup>Prof. Deng Cai is an expert on MATLAB, you can find all his code at <http://www.cad.zju.edu.cn/home/dengcai/Data/data.html>. You can learn how to write fast MATLAB code by reading his code.

- (b) We have seen the effects of different choices of  $K$ . How can you choose a proper  $K$  when dealing with real-world data ?
- (c) Now let us use KNN algorithm to hack the CAPTCHA of a website<sup>7</sup> that we are all familiar with:



Finish *hack.m/hack.py* to recognize the CAPTCHA image using KNN algorithm.

You should label some training data yourself, and store the training data in *hack\_data.mat/hack\_data.npz*. Helper functions *extract\_image* and *show\_image* are give for your convenience.

Remember to submit *hack\_data.mat/hack\_data.npz* along with your code and report.

### 3. Decision Tree and ID3

Consider the scholarship evaluation problem: selecting scholarship recipients based on gender and GPA. Given the following training data:

Draw the decision tree that would be learned by ID3 algorithm and annotate each non-leaf node in the tree with the information gain attained by the respective split.

<sup>7</sup><http://cwcx.zju.edu.cn/WFManager/login.jsp>

Gender	GPA	Scholarship	Count
F	Low	+	10
F	High	+	95
M	Low	+	5
M	High	+	90
F	Low	-	80
F	High	-	20
M	Low	-	120
M	High	-	30

## 4. K-Means Clustering

The codes of this section are in the *kmeans* folder.

Finally, we will run our first unsupervised algorithm – k-means clustering. Implement k-means algorithm (in *kmeans.m/kmeans.py*), then answer the following questions.

Note that there are different kind of methods to setup initial cluster centers for k-means algorithm, we will use a simple one – randomly choose  $K$  samples from dataset as initial cluster centers.

- Run your k-means algorithm on *kmeans\_data.mat* with the number of clusters  $K$  set to 2. Repeat the experiment 1000 times. Use *kmeans\_plot.m/kmeans\_plot.py* to visualize the process of k-means algorithm for the two trials with largest and smallest SD (sum of distances from each point to its respective centroid).
- You should observe the issue that the outcome of k-means algorithm is very sensitive to cluster centroids initialization form the above experiment. How can we get a stable result using k-means?
- Run your k-means algorithm on the digit dataset *digit\_data.mat* with the number of clusters  $K$  set to 10, 20 and 50. Visualize the centroids using *show\_digit.m/show\_digit.py*. You should be able to observe that k-means algorithm can discover the patterns in dataset without any label information.
- Another important application of k-means is Vector quantization<sup>8</sup>. Vector quantization is a classical quantization technique from signal processing. It works by dividing a large set of points (vectors) into groups, then representing the data points by their group centroid points, as in k-means and some other clustering algorithms.

Here we will use vector quantization to do image compression. By clustering image pixel value into  $K$  groups, we can represent each pixel with  $\log(K)$  bits, instead of 24 bits (RGB, each channel has 8bit depth).

Finish *vq.m/vq.ipynb*. Compress images with  $K$  set to 8, 16, 32 and 64. I have provided you some sample images, however use your own photos is encouraged.

What is the compress ratio if we set  $K$  to 64 (Optionally, you can compute the compress ratio using Huffman encoding) ?

---

<sup>8</sup>[https://en.wikipedia.org/wiki/Vector\\_quantization](https://en.wikipedia.org/wiki/Vector_quantization)

---

Please submit your homework report to at <http://assignment.zjulearning.org:8081> in pdf format, with all your code in a zip archive.