

JichenDai-HW3

Problem1: (Code is in file [kmeans.py](#))

(1) [5.171, 3.171]

(2) [5.3, 4.0]

(3) [6.2, 3.025]

(4) 2 iterations

Problem2:

$$(1) \frac{\partial L}{\partial \mu_1} = \frac{\partial}{\partial \mu_1} \sum_{x_i \in S_1} (x_i - \mu_1)^T (x_i - \mu_1)$$
$$= \sum_{x_i \in S_1} 2(\mu_1 - x_i)$$

$$\text{So, } \mu_1 \leftarrow \mu_1 + 2\epsilon \sum_{x_i \in S_1} (x_i - \mu_1)$$

$$(2) \mu_1 \leftarrow \mu_1 + \epsilon (x_i - \mu_1) \text{ when } x_i \in S_1$$

$$(3) \text{In standard k-means: } \mu_1 \leftarrow \sum_{x_i \in S_1} \frac{1}{|S_1|} x_i$$

$$\text{In part (1), we got: } \mu_1 \leftarrow \mu_1 + 2\epsilon \sum_{x_i \in S_1} (x_i - \mu_1)$$

$$\text{So, } \sum_{x_i \in S_1} \frac{1}{|S_1|} x_i = \mu_1 + 2\epsilon \sum_{x_i \in S_1} (x_i - \mu_1)$$

$$\text{Since } \mu_1 = \sum_{x_i \in S_1} \frac{1}{|S_1|} \mu_1$$

$$\sum_{x_i \in S_1} \frac{1}{|S_1|} (x_i - \mu_1) = 2\epsilon \sum_{x_i \in S_1} (x_i - \mu_1)$$

$$\epsilon = \frac{1}{2|S_1|}$$

Problem3:

(1) Since $\sum_{k=1}^K \pi_k = 1$

$$P(\mathbf{z}) = \prod_{k=1}^K \pi_k^{z_k}$$

Then, $p(x|\mathbf{z}) = \prod_{k=1}^K N(x|\mu_k, \Sigma_k)^{z_k}$

$$(2) p(x) = \sum_{k=1}^K \pi_k N(x|\mu_k, \Sigma_k)$$

$$= \sum_{\mathbf{z}} P(\mathbf{z}) p(x|\mathbf{z})$$

So, according to (1), we can get:

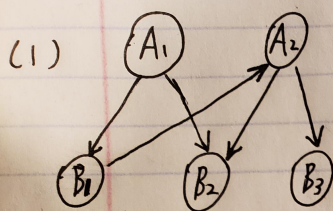
$$p(x) = \sum_{k=1}^K \pi_k N(x|\mu_k, \Sigma_k)$$

(3) Expectation-Maximization (EM) can be used,

Difference:

- a. k-means use L2 Norm, while EM doesn't use L2 norm.
- b. k-means hard assign a data point to a cluster, while EM's result is based on probability.

Problem4:



$$(2) P(A_1, A_2, B_1, B_2, B_3) \\ = P(A_1) P(A_2|B_1) P(B_1|A_1) P(B_2|A_1, A_2) P(B_3|A_2)$$

$$(3) \text{parameters needed} = 1 + 2 + 2 + 4 + 2 = 11$$

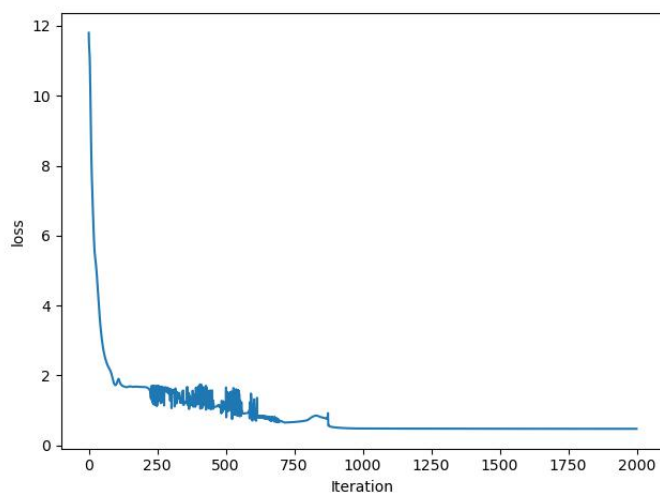
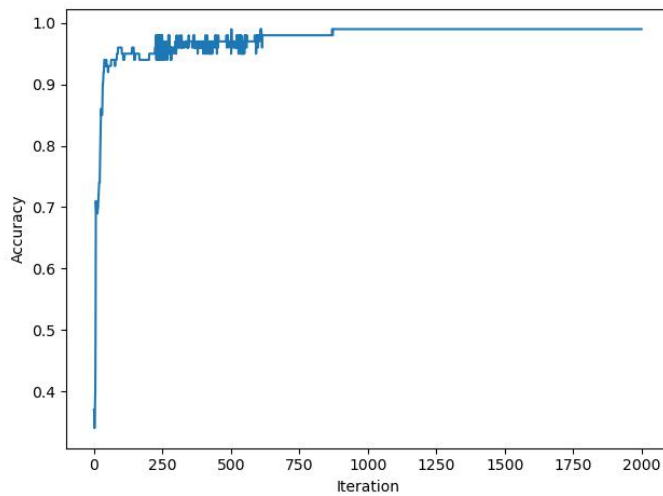
(4) One possible factorization is

$$P(A_1) P(A_2) P(B_1) P(B_2) P(B_3), \text{ only 5 independent parameter is needed}$$

Problem5: (code is in the file [NN.py](#))

(1) Number of neuron: **4**

The test accuracy: 0.9591836734693877



(2) I created two network that has **different number of neurons** in hidden layer. One of them has **2** hidden neuron, another has **8** hidden neuron.

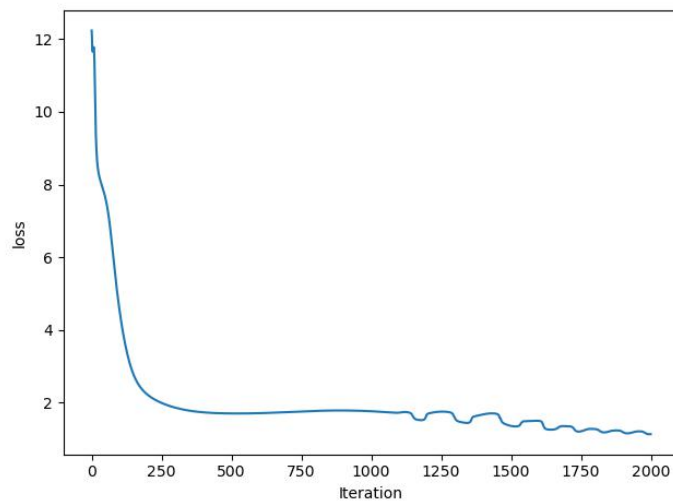
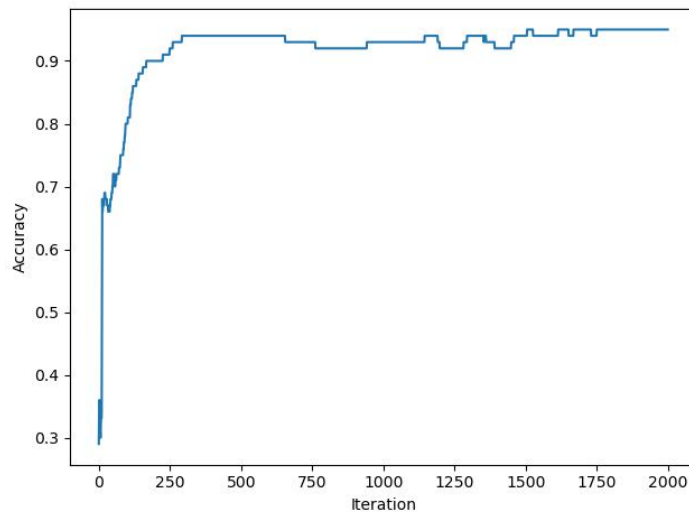
Comparison: As shown in images below. **First**, the image of 2 neuron model is more tortuous, while the lines of 8 neuron model is much smoother.

Additionally, as the number of hidden neuron grow, the accuracy is also increasing.

However, this conclusion is not always valid, if you run this code for many times, you may encounter a situation that the line of 2 neuron is smoother.

Number of neuron: **2**

The test accuracy: 0.9387755102040817



Number of neuron: **8**

The test accuracy: 0.9795918367346939

