ECE657, Spring 2020, Assignment 2

Zhijie Wang, [redacted]

June 16, 2020

Problem 1

First we rewrite $\sum_{i=1}^{N} w_k(n)\phi\{\mathbf{x}(n), \mathbf{c}_k(n), \sigma_k(n)\}$ as $\mathbf{w}_n^T \Phi(n)$, where

$$\mathbf{w}_n \quad \begin{bmatrix} w_1(n) \\ w_2(n) \\ \dots \\ w_N(n) \end{bmatrix} \quad \Phi(n) \quad \begin{bmatrix} \phi\{\mathbf{x}(n), \mathbf{c}_1, \sigma_1\} \\ \phi\{\mathbf{x}(n), \mathbf{c}_2, \sigma_2\} \\ \dots \\ \phi\{\mathbf{x}(n), \mathbf{c}_N, \sigma_N\} \end{bmatrix} \tag{1}$$

Therefore,

$$\begin{aligned} \frac{\partial}{\partial \mathbf{w}} J(n) &\quad \left[y_d(n) - \mathbf{w}_n^T \Phi(n)\right] \cdot [-\Phi(n)] \quad -e(n)\Phi(n) \\ \mathbf{w}(n+1) &\quad \mathbf{w}(n) + \mu_w e(n)\Phi(n) \end{aligned} \tag{2}$$

Then,

$$\begin{aligned} \frac{\partial}{\partial \mathbf{c}_k} J(n) &\quad -e(n) \cdot \frac{\partial}{\partial \mathbf{c}_k}\left[\sum_{i=1}^{N} w_k(n) \exp\left(-\frac{\|\mathbf{x}(n) - \mathbf{c}_k(n)\|^2}{2\sigma_k^2(n)}\right)\right] \\ &\quad -e(n) \cdot \frac{\partial}{\partial \mathbf{c}_k}\left[w_k(n) \exp\left(-\frac{\|\mathbf{x}(n) - \mathbf{c}_k(n)\|^2}{2\sigma_k^2(n)}\right)\right] \\ &\quad -e(n) w_k(n) \exp\left(-\frac{\|\mathbf{x}(n) - \mathbf{c}_k(n)\|^2}{2\sigma_k^2(n)}\right) \cdot \frac{\partial}{\partial \mathbf{c}_k}\left(-\frac{\|\mathbf{x}(n) - \mathbf{c}_k(n)\|^2}{2\sigma_k^2(n)}\right) \\ &\quad -e(n) w_k(n) \phi\{\mathbf{x}(n), \mathbf{c}_k(n), \sigma_k(n)\} \cdot \frac{\partial}{\partial \mathbf{c}_k}\left(-\frac{\mathbf{x}(n)^T\mathbf{x}(n) - 2\mathbf{c}_k(n)^T\mathbf{x}(n) + \mathbf{c}_k(n)^T\mathbf{c}_k(n)}{2\sigma_k^2(n)}\right) \\ &\quad -e(n) w_k(n) \phi\{\mathbf{x}(n), \mathbf{c}_k(n), \sigma_k(n)\} \frac{-[\mathbf{x}(n) - \mathbf{c}_k(n)]}{\sigma_k^2(n)} \end{aligned} \tag{3}$$

$$\mathbf{c}_k(n+1) \quad \mathbf{c}_k(n) + \mu_c \frac{e(n) w_k(n)}{\sigma_k^2(n)} \phi\{\mathbf{x}(n), \mathbf{c}_k(n), \sigma_k(n)\}[\mathbf{x}(n) - \mathbf{c}_k(n)]$$

Similarly,

$$\begin{aligned}\frac{\partial}{\partial \sigma_k} J(n) \quad & -e(n)w_k(n)\phi\{\mathbf{x}(n), \mathbf{c}_k(n), \sigma_k(n)\} \cdot \frac{\partial}{\partial \sigma_k}\left(-\frac{\|\mathbf{x}(n)-\mathbf{c}_k(n)\|^2}{2\sigma_k^2(n)}\right) \\ & -e(n)w_k(n)\phi\{\mathbf{x}(n), \mathbf{c}_k(n), \sigma_k(n)\} \cdot \left(\frac{\|\mathbf{x}(n)-\mathbf{c}_k(n)\|^2}{\sigma_k^3(n)}\right)\end{aligned} \tag{4}$$

$$\sigma_k(n+1) \quad \sigma_k(n) - \mu_\sigma \frac{e(n)w_k(n)}{\sigma_k^3(n)}\phi\{\mathbf{x}(n), \mathbf{c}_k(n), \sigma_k(n)\}\|\mathbf{x}(n)-\mathbf{c}_k(n)\|^2$$

Since μ_c and μ_σ are both appropriate learning rate, we can say that

$$\begin{aligned}\mathbf{c}_k(n+1) \quad & \mathbf{c}_k(n) + \mu_c \frac{e(n)w_k(n)}{\sigma_k^2(n)}\phi\{\mathbf{x}(n), \mathbf{c}_k(n), \sigma_k(n)\}[\mathbf{x}(n)-\mathbf{c}_k(n)] \\ \sigma_k(n+1) \quad & \sigma_k(n) + \mu_\sigma \frac{e(n)w_k(n)}{\sigma_k^3(n)}\phi\{\mathbf{x}(n), \mathbf{c}_k(n), \sigma_k(n)\}\|\mathbf{x}(n)-\mathbf{c}_k(n)\|^2\end{aligned} \tag{5}$$

Problem 2

1. Research Subjects

   The associate memory as a network which is attractive to many applications have been demonstrated by Hopfiled, therefore, in this paper, the authors mainly researched in the techniques on how to analysis Hopfield-memories. And in detail, the authors found that such techniques are quite similar to methods in coding theory. We can think of the associative memory as a kind of decoder for a code consisting of the m fundamental memories as codewords. However, such coding have very low rate, thus, the authors think of an associative memory as a basket of m memories. Then, under these considerations, the authors proposed that the memory encoding rule is essentially the algorithm for finding appropriate weights connections between two states in the network.

2. Major Contributions

   First, the authors introduced how sum-of-outer products connection matrix works in Hopfiled network, which will finally make the memories be stable. One feature of sum-of-outer products is that once connection matrix T have been calculated, all other information about memory x^{α} will be forgotten, which is important when the network have to learn. The authors also proved that the outer product algorithm behaves well with regard to stability of the memories provided that the number of memories m is small enough compared to the number of components n in the memory vectors.

   Second, comparing to sum-of-outer products matrix connection, the authors also provide some examples of other connection methods, which might have higher capacity, however, such methods might be harder to build. Thus, sum-of-outer products seem to be the most suitable method.

   Third, the authors discussed about the stability of memories. One is the error-correcting or said "pull-in" capability. Suppose there are some components which are unknown or incorrect, we want a method to correct them. In general, the authors said there are three possibilities of convergence. Most probe vectors will reach the fundamental memory at the center of the sphere as a stable or fixed point, in both the asynchronous and synchronous models, if there are not too many fundamental memories at the start. And the fundamental memories will be reached from within the spheres, too.

   Forth, the authors discussed about the capacity of the coding method. Instead of talking about an exact number, the capacity will be a growth rate. If direct attraction is desired and all the fundamental memories must be recallable correctly, the capacity is

   $$\frac{(1-2\rho)^2}{4} n/\log n \tag{6}$$

   where $0 < \rho \leqslant 1$. A simplified derivation of capacity is that the maximum number of memories that can be stored in a Hopfield matrix is asymptotically at most $n/(2\log n)$.

   Fifth, the authors also had a discussion about the extension for capacity when nondirect convergence happens.

3. Conclusions

The main conclusion in this paper is about the capacity m of a Hopfiled associative memory of length n when it is to be presented with a number m of random independent ± 1 probability 1/2 fundamental memories to store and when probing with a probe n-tuple at most ρn away from a fundamental memory ($0 \leqslant \rho < 1$) is

$$\frac{(1-2\rho)^2}{2} n/\log n \tag{7}$$

If no fundamental memory can be exceptional, the m should be

$$\frac{(1-2\rho)^2}{4} n/\log n \tag{8}$$

if ρ given, and some wrong moves are permitted, and we can have as above a small fraction of exceptional fundamental memories

$$n/(2\log n) \tag{9}$$

And if no fundamental memory can be exceptional, m should be

$$n/(4\log n) \tag{10}$$

4. Commentary

In this paper, the authors mainly discussed about why sum-of-outer products connection matrix used and the stability of Hopfield, then follows the performance of Hopfield network, especially the capacity m a Hopfiled associative memory of length n.

From the lecture we've known that with the Hopfield networks' special structure, therefore, can be used in solving problems like optimization and pattern/speech recognition. However, from the paper we can know the capacity of Hopfield network is limited in storage, hence, cannot be used in a large application.

Problem 3

1. From the plot Fig.1 we can find that, when sigma is small, MSE is high, however, when sigma is increasing, the MSE first decrease and then increase. We choose the sigma σ 2 with smallest MSE 0.118889.

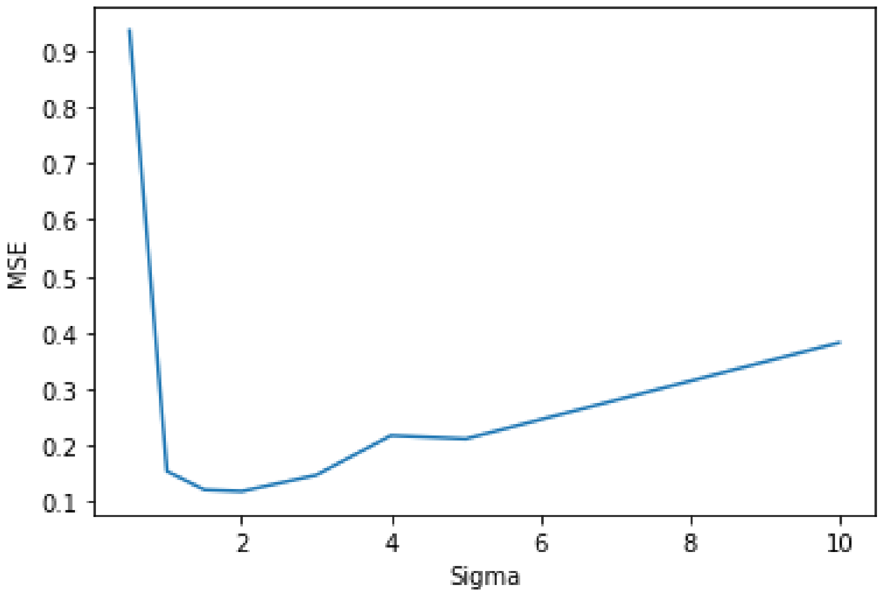

Figure 1: Sigma vs MSE

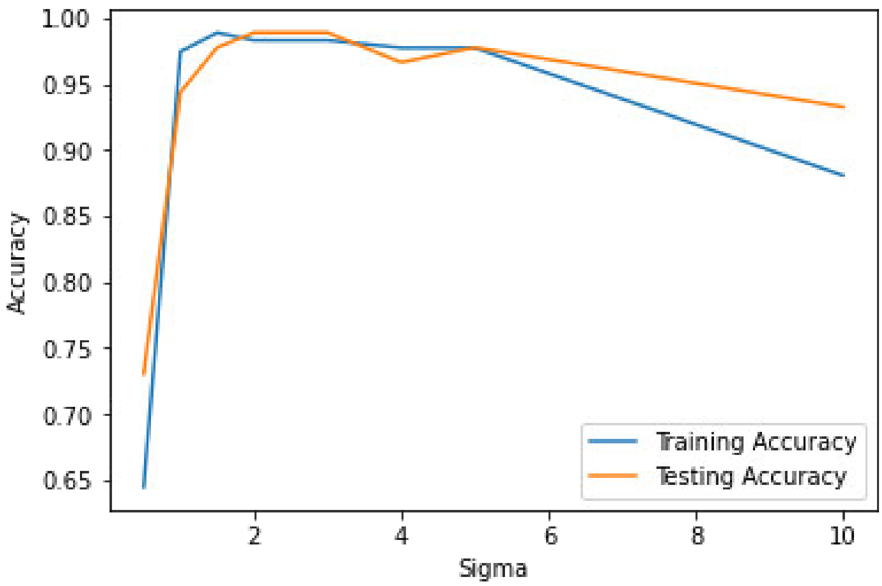

Figure 2: Sigma vs Accuracy

From the plot Fig. 2 we can find that, when sigma is small, the accuracy is low, when sigma is increasing, the accuracy first increase and then decrease. When sigma = 2, which we've chose according to MSE, the accuracy for training and testing data are both nearly the highest, which is 0.983 and 0.989.

2. (a) When we choose randomly 150 centers for RBF, train accuracy is 0.986, test accuracy is 0.989.

(b) When use 150 K-means center, train Accuracy is 0.989, test accuracy is 0.989.

We can find that both these two methods have similar result comparing to using all the training samples as centers.

Problem 4

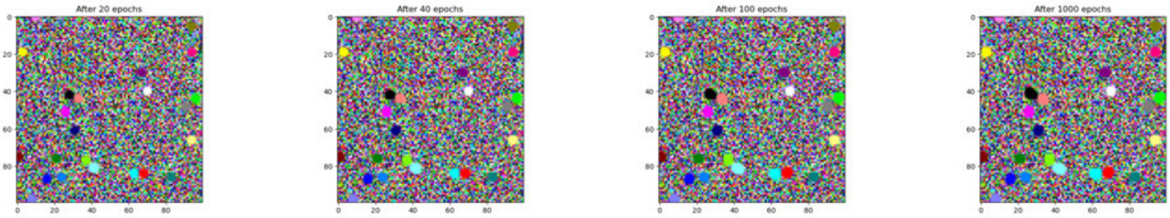

Figure 3: σ 1

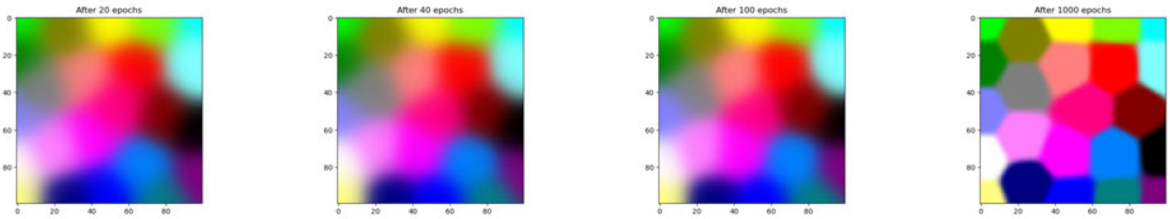

Figure 4: σ 10

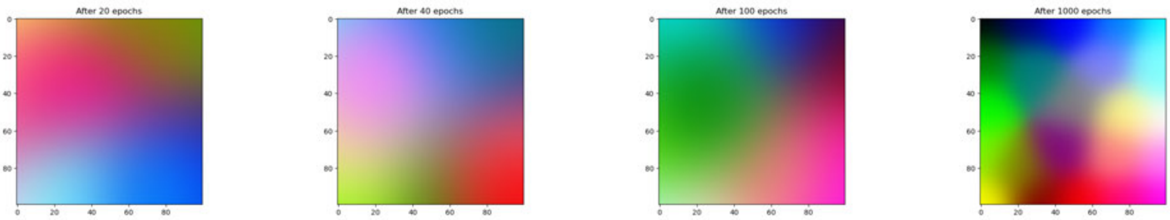

Figure 5: σ 30

1.

2. When σ_0 is small, the boundary between different colours are clear, and only a small cluster for each color. When σ_0 is big, the boundary between different colours become smooth, we can see a smooth change from one colour to another colour on the map.

   However, when σ_0 is small, the training process is fast, only need a few epochs to achieve the final result. When σ_0 is bigger, the algorithm takes more epochs to achieve the final result. We can see that when $\sigma_0 \geqslant 50$, only after 1000 epochs we can see a good map.

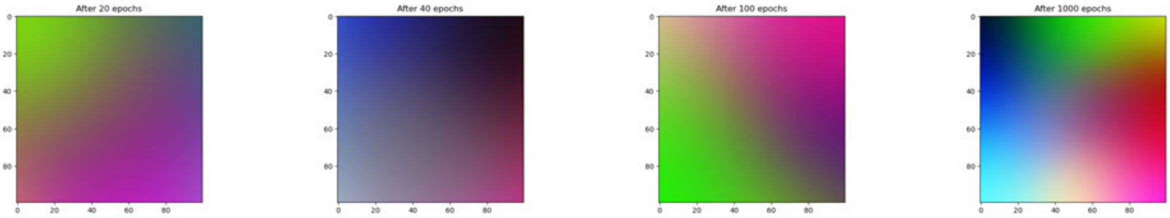

Figure 6: σ 50

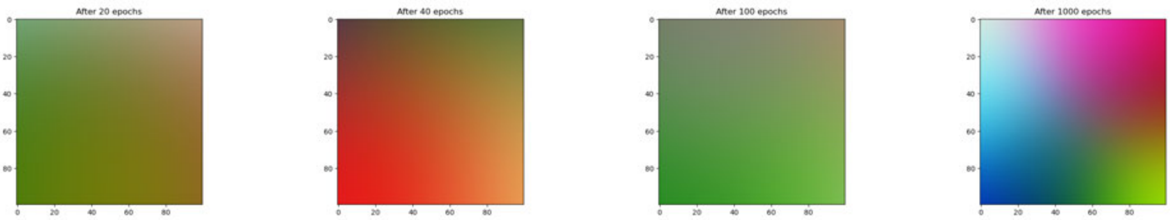

Figure 7: σ 70