Assignment 2 Solution

Problem Q1:

Derive formula for Stochastic based gradient-based method for training an RBF NN

Solution:

Given

The first step in the development of the stochastic gradient-based supervised learning algorithm is to define the instantaneous error cost function as

$$\mathrm{J(n)} = \frac{1}{2}|e(n)|^2 = \frac{1}{2}[y_d(n) - y(n)]^2 \qquad \text{------ 1}$$

Where $y(n)$ is the actual output and $y_d(n)$ is the desired output at iteration (n) with

$$y(n) = \sum_{k=1}^{N} w_k(n) \ \emptyset\{x(n), c_k, \sigma_k\} \qquad \text{------ 2}$$

Where $c_k(n)$ being the center vector for the k-th radial function $\emptyset\{x(n), c_k, \sigma_k\}$ and σ_k is the spread parameter.

$$\mathrm{J(n)} = \frac{1}{2}|e(n)|^2 = \frac{1}{2}\left[y_d(n) - \sum_{k=1}^{N} w_k(n) \ \emptyset\{x(n), c_k(n), \sigma_k(n)\}\right]^2 \qquad \text{------ 3}$$

If the RBF seed function $\emptyset\{x(n), c_k(n), \sigma_k(n)\}$ is chosen to be Gaussian kernel, $\mathrm{J(n)}$ becomes

$$\mathrm{J(n)} = \frac{1}{2}\left[y_d(n) - \sum_{k=1}^{N} w_k(n) \ exp\left(-\frac{\|x(n)-c_k(n)\|^2}{2{\sigma_k}^2(n)}\right)\right]^2 \qquad \text{------ 4}$$

The updated equation for the network parameters are then given by

$$w(n+1) = w(n) - \mu_w \frac{d}{dw} J(n)|_{w=w(n)} \qquad \text{------ 5}$$

$$c_k(n+1) = c_k(n) - \mu_c \frac{d}{dc_k} J(n)|_{c_k=c_k(n)} \qquad \text{------ 6}$$

$$\sigma_k(n+1) = \sigma_k(n) - \mu_\sigma \frac{d}{d\sigma_k} J(n)|_{\sigma_k=\sigma_k(n)} \qquad \text{------ 7}$$

μ_w,μ_c,μ_σ are appropriate learning rate parameters.

To Show,

$$\boldsymbol{w(n+1) = w(n) + \mu_w e(n) \varphi(n)}$$

$$\boldsymbol{c_k(n+1) = c_k(n) + \mu_c \frac{e(n) w_k(n)}{{\sigma_k}^2(n)} \emptyset\{x(n), c_k(n), \sigma_k\}[x(n) - c_k(n)]}$$

$$\boldsymbol{\sigma_k(n+1) = \sigma_k(n) + \mu_\sigma \frac{e(n) w_k(n)}{{\sigma_k}^3(n)} \emptyset\{x(n), c_k(n), \sigma_k\}\|x(n) - c_k(n)\|^2}$$

Where,

$$\varphi(n) = [\emptyset\{x(n), c_1, \sigma_1\} + \emptyset\{x(n), c_2, \sigma_2\} + \cdots \dots \emptyset\{x(n), c_N, \sigma_N\}]^T$$

1. To show

$$\boldsymbol{w(n+1) = w(n) + \mu_w e(n)\varphi(n)}$$

Using the chain rule of derivative, from Equation 1, we can write.

$$\mathrm{J(n)} = \frac{1}{2}|e(n)|^2$$

$$= \frac{d}{de(n)}\frac{1}{2}|e(n)|^2 . \frac{d\,e(n)}{dw}$$

$$= \left[\frac{1}{2}(2).\frac{d\,e(n)}{dw}\right]$$

$$\mathrm{J(n)} = e(n).\frac{d\,e(n)}{dw} \qquad \text{------ 8}$$

Substitute the value of $\mathrm{J(n)}$ in equation 5.

$$w(n+1) = w(n) - \mu_w e(n)\frac{d\,e(n)}{dw}$$

$$w(n+1) = w(n) + \mu_w e(n)\left(-\frac{d\,e(n)}{dw}\right)$$

$$w(n+1) = w(n) + \mu_w e(n)\frac{d}{dw}\left[-\left(y_d(n) - \sum_{k=1}^{N} w_k(n)\ \emptyset\{x(n), c_k, \sigma_k\}\right)\right]$$

$$= w(n) + \mu_w e(n)\frac{d}{dw}[-y_d(n) + w_1(n)\ \emptyset\{x(n), c_1, \sigma_1\} + w_2(n)\ \emptyset\{x(n), c_2, \sigma_2\} + \cdots + w_N(n)\ \emptyset\{x(n), c_N, \sigma_N\}]$$

Since $y_d(n)$ is constant

$$= w(n) + \mu_w e(n)\frac{d}{dw}[w_1(n)\ \emptyset\{x(n), c_1, \sigma_1\} + w_2(n)\ \emptyset\{x(n), c_2, \sigma_2\} + \cdots + w_N(n)\ \emptyset\{x(n), c_N, \sigma_N\}]$$

$$w(n+1) = \begin{bmatrix} w_1(n+1) \\ w_2(n+1) \\ \\ w_N(n+1) \end{bmatrix}$$

Since it is a partial derivative w.r.t w, $w_1, w_2, \ldots \ldots . w_N$ are constant & its partial derivative w.r.t w_1 is 0.

Also $\emptyset\{x, c, \sigma\}$ is constant. Since input x is constant.

$$w_1(n+1) = w_1(n) + \mu_w e(n)\emptyset\{x(n), c_1, \sigma_1\}$$

Similarly,

$$w_2(n+1) = w_2(n) + \mu_w e(n)\emptyset\{x(n), c_2, \sigma_2\}$$

$$w(n+1) = w(n) + \mu_w e(n)\varphi(n)$$

$$w(n+1) = \begin{bmatrix} w_1(n+1) \\ w_2(n+1) \\ \\ w_N(n+1) \end{bmatrix} = \begin{bmatrix} w_1(n) \\ w_2(n) \\ \\ w_N(n) \end{bmatrix} + \mu_w e(n) \begin{bmatrix} \emptyset\{x(n), c_1, \sigma_1\} \\ \emptyset\{x(n), c_2, \sigma_2\} \\ \\ \emptyset\{x(n), c_N, \sigma_N\} \end{bmatrix}$$

$$\boldsymbol{w(n+1) = w(n) + \mu_w e(n)\varphi(n)}$$

Where,

$$\varphi(n) = [\emptyset\{x(n), c_1, \sigma_1\} + \emptyset\{x(n), c_2, \sigma_2\} + \cdots \ldots \emptyset\{x(n), c_N, \sigma_N\}]^T$$

2. To show

$$\boldsymbol{c_k(n+1) = c_k(n) + \mu_c \frac{e(n)w_k(n)}{{\sigma_k}^2(n)} \emptyset\{x(n), c_k(n), \sigma_k\}[x(n) - c_k(n)]}$$

Using the chain rule of derivative, from Equation 1, we can write.

$$\mathrm{J(n)} = \frac{1}{2}|e(n)|^2$$

$$= \frac{d}{de(n)}\frac{1}{2}|e(n)|^2 . \frac{d\,e(n)}{dw}$$

$$\mathrm{J(n)} = e(n) . \frac{d\,e(n)}{dw}$$

$$c_k(n+1) = c_k(n) + \mu_c e(n)\left[-\frac{de(n)}{dc_k}\right]$$

$$c_k(n+1) = c_k(n) + \mu_c e(n)\frac{d}{dc_k}\left[-\left(y_d(n) - \textstyle\sum_{k=1}^{N} w_k(n)\ \emptyset\{x(n), c_k(n), \sigma_k\}\right)\right]$$

$$= c_k(n) + \mu_c e(n)\frac{d}{dc_k}[w_1\ \emptyset\{x(n), c_k(n), \sigma_1\} + w_2\ \emptyset\{x(n), c_k(n), \sigma_2\} + \cdots + w_N\ \emptyset\{x(n), c_k(n), \sigma_N\}]$$

Expanding $\emptyset$ we get,

$$= c_k(n) + \mu_c e(n)\frac{d}{dc_k}\left[w_1 exp\left(-\frac{\|x(n)-c_1(n)\|^2}{2{\sigma_1}^2(n)}\right) + w_2 exp\left(-\frac{\|x(n)-c_2(n)\|^2}{2{\sigma_2}^2(n)}\right) + \cdots + w_N exp\left(-\frac{\|x(n)-c_N(n)\|^2}{2{\sigma_N}^2(n)}\right)\right]$$

Since it is a partial derivative w.r.t c_1, $w_1, w_2, \ldots \ldots, w_N$ are constant x is also constant.

$$c_1(n+1) = c_1(n) + \mu_c e(n)\frac{d}{dc_1}\left[w_1 exp\left(-\frac{\|x(n)-c_1(n)\|^2}{2{\sigma_1}^2}\right)\right]$$

$$= c_1(n) + \mu_c e(n). w_1 exp\left(-\frac{\|x(n)-c_1(n)\|^2}{2{\sigma_1}^2(n)}\right)\frac{(-2)(x(n)-c_1(n))}{2{\sigma_1}^2(n)}(-1)$$

$$= c_1(n) + \mu_c e(n). w_1\emptyset\{x(n), c_1(n), \sigma_1\}\frac{(x(n)-c_1(n))}{{\sigma_1}^2(n)}$$

Similarly, we can write the same Equation for $c_k(n+1)$

$$c_k(n+1) = c_k(n) + \mu_c e(n)\frac{d}{dc_k}\left[w_1 exp\left(-\frac{\|x(n)-c_1(n)\|^2}{2{\sigma_k}^2}\right)\right]$$

$$= c_k(n) + \mu_c e(n). w_k exp\left(-\frac{\|x(n)-c_k(n)\|^2}{2{\sigma_k}^2(n)}\right)\frac{(-2)(x(n)-c_k(n))}{2{\sigma_k}^2(n)}(-1)$$

$$= c_k(n) + \mu_c e(n). w_k\emptyset\{x(n), c_k(n), \sigma_k\}\frac{(x(n)-c_k(n))}{{\sigma_k}^2(n)}$$

$$\boldsymbol{c_k(n+1) = c_k(n) + \mu_c \frac{e(n)w_k(n)}{{\sigma_k}^2(n)} \emptyset\{x(n), c_k(n), \sigma_k\}[x(n) - c_k(n)]}$$

3. To show

$$\boldsymbol{\sigma_k(n+1) = \sigma_k(n) + \mu_\sigma \frac{e(n)w_k(n)}{{\sigma_k}^3(n)} \emptyset\{x(n), c_k(n), \sigma_k\} \| x(n) - c_k(n) \|^2}$$

From Equation 7, we can write.

$$\sigma_k(n+1) = \sigma_k(n) - \mu_\sigma \frac{d}{d\sigma_k} J(n)|_{\sigma_k=\sigma_k(n)}$$

Using the chain rule of derivative, from Equation 1, we can write.

$$\mathrm{J(n)} = \frac{1}{2}|e(n)|^2$$

$$= \frac{d}{de(n)} \frac{1}{2} |e(n)|^2 . \frac{d\, e(n)}{dw}$$

$$\mathrm{J(n)} = e(n) . \frac{d\, e(n)}{dw}$$

$$\sigma_k(n+1) = \sigma_k(n) + \mu_\sigma e(n) \left[-\frac{de(n)}{d\sigma_k} \right]$$

$$= \sigma_k(n) + \mu_c e(n) \frac{d}{d\sigma_k} \left[-\left(y_d(n) - \sum_{k=1}^{N} w_k(n)\ \emptyset\{x(n), c_k(n), \sigma_k\} \right) \right]$$

$$= \sigma_k(n) + \mu_c e(n) \frac{d}{d\sigma_k} [w_1 \emptyset\{x(n), c_k(n), \sigma_1\} + w_2 \emptyset\{x(n), c_k(n), \sigma_2\} + \cdots + w_N \emptyset\{x(n), c_k(n), \sigma_N\}]$$

Expanding ∅ we get,

$$= \sigma_k(n) + \mu_c e(n) \frac{d}{d\sigma_k} \left[w_1 exp\left(-\frac{\|x(n)-c_1(n)\|^2}{2{\sigma_1}^2(n)} \right) + w_2 exp\left(-\frac{\|x(n)-c_2(n)\|^2}{2{\sigma_2}^2(n)} \right) + \cdots + w_N exp\left(-\frac{\|x(n)-c_N(n)\|^2}{2{\sigma_N}^2(n)} \right) \right]$$

$$\sigma_1(n+1) = \sigma_1(n) + \mu_c e(n) \frac{d}{d\sigma_1} \left[w_1 exp\left(-\frac{\|x(n)-c_1(n)\|^2}{2{\sigma_1}^2(n)} \right) \right]$$

$$= \sigma_1(n) + \mu_c e(n). w_1 exp\left(-\frac{\|x(n)-c_1(n)\|^2}{2{\sigma_1}^2(n)} \right) \frac{(-2)\|x(n)-c_1(n)\|^2}{2{\sigma_1}^3(n)} (-1)$$

$$= \sigma_1(n) + \mu_c e(n). w_1 exp\left(-\frac{\|x(n)-c_1(n)\|^2}{2{\sigma_1}^2(n)} \right) \frac{\|x(n)-c_1(n)\|^2}{{\sigma_1}^3(n)}$$

Similarly, we can write the same Equation for $\sigma_k(n+1)$

$$\sigma_k(n+1) = \sigma_k(n) + \mu_c e(n) \frac{d}{d\sigma_k} \left[w_k exp\left(-\frac{\|x(n)-c_k(n)\|^2}{2{\sigma_k}^2(n)} \right) \right]$$

$$= \sigma_k(n) + \mu_c e(n). w_k exp\left(-\frac{\|x(n)-c_k(n)\|^2}{2{\sigma_k}^2(n)} \right) \frac{(-2)\|x(n)-c_k(n)\|^2}{2{\sigma_k}^3(n)} (-1)$$

$$= \sigma_k(n) + \mu_c e(n). w_k exp\left(-\frac{\|x(n)-c_k(n)\|^2}{2{\sigma_k}^2(n)} \right) \frac{\|x(n)-c_k(n)\|^2}{{\sigma_k}^3(n)}$$

$$\boldsymbol{\sigma_k(n+1) = \sigma_k(n) + \mu_\sigma \frac{e(n)w_k(n)}{{\sigma_k}^3(n)} \emptyset\{x(n), c_k(n), \sigma_k\} \| x(n) - c_k(n) \|^2}$$

Problem Q2:

The paper titled "The Capacity of the Hopfield Associative Memory". Summarize the paper thoroughly heightening the subject of research, the major contributions, and the conclusions

Solution:

Subject of Research:

Capacity of Hopfield associative memory can be improved with the help of coding theory. These memory units store sequences in the form of +1 and -1. Hopfield network relies on the network formed with the assistance of symmetric connections connecting fundamental memory blocks which are either in excited or inhibited state at one time. These fundamental memories are capable of retaining pattern with the help of n initial components with hamming distance of n/2 from fundamental memory.

Significant Contribution:

Hopfield networks are evocative of coding theory and more specifically in random coding and sphere hardening. Also these networks are often related to "Neuroanatomical models" for the brain, which represents memory as interconnection of dynamic cellular clusters by dense connecting mediums of linear synaptic conduits. Similarly mathematical representation of associative memory can be perceived as a neuron acting as a memory cell with two possible states (on and off) which follows the principle of neural networks. The interconnection between these neurons are considered to have fixed weights and it is claimed that it has symmetric structure with all diagonal elements as zero. Unlike neural networks, decisions are taken with the assistance of a simple threshold method, which results in 1 if the weighted sum of states at a particular node crosses the threshold. The same will result into off state if weighted sum is below threshold. Usually this threshold is considered to be zero.

$$x'_i = sgn\{\textstyle\sum_{j=1}^{n} T_{ij}\ x_j\} = \begin{cases} +1 & if \sum T_{ij}\ x_j\ \geq 0 \\ -1 & if \sum T_{ij}\ x_j\ < 0 \end{cases} \qquad \text{--Eq(1)}$$

This learning rule can be summarized using equations above, where x' represents the new state of ith node. This transition of x to x' can take place either using synchronous or asynchronous mode. In synchronous mode, each neuron changes state at a time with the help of learning rule stated above. However, in asynchronous manner all neurons are update one at a time. It's worth noting that, despite simple operations like weighted sum (linear operation) and nonlinear logical thresholding, these network structures requires high computations in order to reach stable state.

Key features of this system are: 1) powerful information processing, 2) simple processing (learning logic) at individual nodes, 3) Concurrent information processing. Along with this, memory is distributed and thus overall system becomes robust and demands much less complexity.

Associative memory works as decoder which can decode some fundamental memory as code words. These memory units acts as attractors such that similar patterns are mapped together to create region of influence. In order to determine similarity 'hamming distance' is used. Because of inclusion of association and memory inside NN structure we face issue of memory encoding rule which defines structure of association and recall capacity of overall system with acceptable level of error correction. Since threshold is fix, we can only tune the weights as desired referring the approaches described below:

Process of information retrieval can be explained with the help of example, for any n dimensional vector, another vector x having least hamming distance, update x asynchronously as per weight update rule described in equation 1. Since these updates are random and independent for any network having symmetric connections, this process will converge after finite number of iterations. In other words, if we start with random pattern vector x, it is guaranteed to achieve a fix vector y such that y = sgn(Ty). In asynchronous Hopfield model, associative recall is workable by means of error correction. Nakano defined it as "associatron" and proved that with the support of chain of associations recall and error correction is achievable. To summarize this, outer product approach can gain stability of memories if and only if memories are small enough with respect to the number of components in memory vectors.

Basic design of Hopfield network does not consider non-symmetric connections, and in fact these fix points may or may not exist in reality. Asynchronous model changes state of nodes one at a time and has no memory. Contrarily, in synchronous models all nodes are changed simultaneously, as a result a fixed point in a model may not be achieved. Nevertheless, if connections are same the resultant fix points will be same in different models. The capacity results are constant irrespective of changes in memory. Which means that change in ith component depends on generalized average of last k values.

Apart from sum of outer products, other approaches can be practiced if the fundamental memory patter is eigenvector of connections with all positive values, which yields fixed points. Let the memories $x^{(\propto)}$ be eigenvectors of T with positive eigenvalues$\lambda^{(\propto)}$. Then $sgn\left(\left(Tx^{(\propto)}\right)i\right) = \left(\lambda^{(\propto)}x_i{}^{(\propto)}\right) = x_i{}^{(\propto)}$. Thus the fundamental memories $x_i{}^{(\propto)}$ will be fixed points.

Stability can be viewed in terms of convergence. Convergence can occur in three ways, in first case there is monotonic change in correct direction (for synchronous it will reach fundamental memory in one step). Another possibility is random steps in right direction (for synchronous model convergence will be achieved in two iterations). In third approach, states may oscillate back and forth however the resultant error is minimized as compare to previous step and ultimately after finite number of steps model will converge. As a result, a fix point will exist in spite of synchronous nature of model. It suggests there is a region of attraction around fundamental memories, with radius of approximately equal to n/2. Both approaches are stable for smaller number of fundamental memories.

Capacity of Hopfield network is defined as rate of growth. First, every one of the m fundamental memories may be fixed with a high probability, with practically its full pn-sphere being immediately attracted. Second, conceptually every memory is likely to be excellent, but not necessarily, as stated above and it doubles the memory capacity.

In convergence methodology where there are few wrong moves but error is still minimized to reach fundamental pattern in order to achieve convergence. This is helpful in long memory retrievals with small fractions (approx. ½).

$$\frac{(1-2p)^2}{4}\ n/\log n$$

As previously said, if we can have a little fraction of extraordinary core memories, our capacity is doubled. If we allow the second sort of convergence, where we can make a few incorrect moves but still get close enough to the fundamental memory to have direct convergence, then we eliminate the component $(1 – 2p)^2$ above (for any fixed p, 0 < p < 1/2). This improvement is crucial when we need to reconstruct long memories while only being certain of a minute portion of them. As we observed in the last section, p is close to 1/2 in this case.

We have utterly allowed a small part of the m fundamental memories to be exceptional, that is, not fixed points, in the previous. If we desire to correct all m fundamental memories, we will have to cut m in half as n grows larger. When we reduce m by $(1 - 2p)^2$, the probe will proceed directly to the fundamental memory in only one synchronous step, however since the probe was pn or less away from a fundamental memory and 0 ≤ p < 1/2.

Summary:

By recapitulating above discussion we can conclude that given a number m of random independent ±1 probability 1/2 fundamental memories to store and probing with a probe n-tuple at most pn distant from a fundamental memory (0 ≤ p < 1/2.) the (asymptotic) capacity m of a Hopfield associative memory of length n is

$$\frac{(1-2p)^2}{2}\ n/\log n$$

Except for a remarkably small fraction of the fundamental memories, if the unique fundamental memory may be recovered with a high probability via direct convergence to the fundamental memory:

$$\frac{(1-2p)^2}{4}\ n/\log n$$

If no fundamental memory can be established in the above case:

$$\frac{n}{2\log n}$$

If 0 ≤ p < 12, p is given, and some incorrect moves are allowed (two steps adequate), we get a small percentage of extraordinary fundamental memories:

$$\frac{n}{4\log n}$$

If, as stated before, incorrect moves are allowed (two synchronous moves are adequate), but no fundamental memory can be exceptional.