ECE 417 Multimedia Signal Processing
Homework 3

UNIVERSITY OF ILLINOIS
Department of Electrical and Computer Engineering

Assigned: Tuesday, 9/22/2020; Due: Monday, 10/5/2020
Reading: Chap. 6, **Neural Networks for Pattern Recognition**, Bishop 1995

Problem 3.1

Your training database contains matched pairs $\{(\vec{x}_1, \vec{y}_1), \ldots, (\vec{x}_n, \vec{y}_n)\}$ where $\vec{x}_i = [x_{i1}, \ldots, x_{ip}]$ is the i^{th} observation vector, and $\vec{y}_i = [y_{i1}, \ldots, y_{iq}]$ is the i^{th} label vector. For some initial weight matrix $W = \begin{bmatrix} w_{11} & \cdots \\ \cdots & w_{qp} \end{bmatrix}$, you have already computed the following two quantities:

$$\hat{y}_{i\ell} = f_\ell(\vec{x}_i, W) \qquad 1 \le \ell \le q,\ 1 \le i \le n \tag{3.1-1}$$

$$\frac{\partial f_\ell(\vec{x}_i, W)}{\partial w_{kj}} \qquad 1 \le \ell \le q,\ 1 \le k \le q,\ 1 \le j \le p,\ 1 \le i \le n \tag{3.1-2}$$

You want to find a new matrix $W' = \begin{bmatrix} w'_{11} & \cdots \\ \cdots & w'_{qp} \end{bmatrix}$ such that $\mathcal{J}(W') \ge \mathcal{J}(W)$ (that is, you want to **maximize** $\mathcal{J}$), where

$$\mathcal{J}(W) = \sum_{i=1}^{n} \sum_{\ell=1}^{q} y_{i\ell} \ln\left(f_\ell(\vec{x}_i, W)\right)$$

Give a formula for w'_{kj} in terms of w_{kj}, $f_\ell(\vec{x}_i, W)$, $\frac{\partial f_\ell(\vec{x}_i, W)}{\partial w_{kj}}$, and in terms of a step size, η, such that for suitable values of η, $\mathcal{J}(W') \ge \mathcal{J}(W)$.

Solution:

$$w'_{kj} = w_{kj} + \eta \sum_{i=1}^{n} \sum_{\ell=1}^{q} \frac{y_{i\ell}}{f_\ell(\vec{x}_i, W)} \frac{\partial f_\ell(\vec{x}_i, W)}{\partial w_{kj}}$$

Problem 3.2

A particular two-layer neural network accepts a two-dimensional input vector $\vec{x} = [x_1, x_2]^T$, and generates an output $z = h(\vec{v}^T g(U\vec{x}))$. Choose network weights $\vec{v}$ and U, and element-wise scalar nonlinearities $h()$ and $g()$, that will generate the following output:

$$z = \begin{cases} 1 & |x_1| < 2 \text{ and } |x_2| < 2 \\ -1 & \text{otherwise} \end{cases}$$

Note: your nonlinearities don't have to be chosen from among the ones introduced in class. You can use any scalar nonlinearity that you like. In particular, since this network doesn't have any biases, you can build the biases into the nonlinearity if you like.

Solution: One possible solution is to use the hidden units to compute $|x_1| < 2$ and $|x_2| < 2$:

$$U = \left[\begin{array}{cc} 1 & 0 \\ 0 & 1 \end{array}\right], \quad g(x) = \begin{cases} 1 & |x| < 2 \\ 0 & \text{otherwise} \end{cases}$$

Then the output layer computes the AND operation:

$$\vec{v} = [1, 1], \quad h(x) = \begin{cases} 1 & x > 1.5 \\ -1 & x \le 1.5 \end{cases}$$

Problem 3.3

A "spiral network" is a brand new category of neural network, invented just for this homework. It is a network with a scalar input variable x_i, a scalar target variable t_i, and with the following architecture:

$$h_{ij} = \left\{ \begin{array}{ll} x_i & j = 1 \\ g\left(e_{ij}\right) & 2 \le j \le M \end{array} \right. , \quad e_{ij} = \sum_{k=1}^{j-1} w_{jk} h_{ik}$$

Suppose that the network is trained to minimize the sum of the per-token squared errors $\mathcal{E} = \frac{1}{2}\sum_{i=1}^{n}(h_{iM} - t_i)^2$. The error gradient can be written as

$$\frac{d\mathcal{E}}{dw_{jk}} = \sum_{i=1}^{n} \delta_{ij} h_{ik}$$

Find a formula that can be used to compute δ_{ij}, for all $2 \le j \le M$, in terms of t_i, $h_{ij} = g(e_{ij})$, and/or $g'(e_{ij}) = \frac{dg}{de_{ij}}$.

Solution:

$$\delta_{ij} = \left\{ \begin{array}{ll} (h_{iM} - t_i) g'(e_{iM}) & j = M \\ \sum_{k=j+1}^{M} \delta_{ik} w_{kj} g'(e_{ij}) & \text{otherwise} \end{array} \right.$$

Problem 3.4

A two-layer neural net has MSE error criterion

$$\mathcal{E} = \frac{1}{2n} \sum_{i=1}^{n} \|\bar{z}_i - \bar{t}_i\|^2$$

where $\bar{t}_i = [t_{i1}, \ldots, t_{ir}]$ is the target vector, and $\bar{z}_i = [z_{i1}, \ldots, z_{ir}]$ is the network output. z_i is computed as

$$z_{ki} = g_{ReLU}(\bar{w}_k \bar{h}_i^T)$$

where $\bar{w}_k = [w_{k1}, \ldots, w_{kq}]$ is a weight vector, and $\bar{h}_i = [h_{i1}, \ldots, h_{iq}]$ is the hidden layer. $\bar{h}_i$ is computed as

$$h_{ij} = g_{ReLU}(\bar{v}_j \bar{x}_i^T)$$

where $\bar{v}_j = [v_{j1}, \ldots, v_{jp}]$ is a weight vector, and $\bar{x}_i = [x_{i1}, \ldots, x_{ip}]^T$ is the network input. The rectified linear units are defined by

$$g_{ReLU}(a) = \max(0, a)$$

Notice that, with these definitions,

$$\frac{d\mathcal{E}}{dw_{kj}} = \frac{1}{n}\sum_{i\in\mathcal{S}} \delta_{ik} h_{ij}$$

for some set of indices $\mathcal{S}$ which is a subset of $\{1, \ldots, n\}$. Find a definition of $\mathcal{S}$, in terms of the network inputs, excitations, activations, and/or weight vectors, that permits you to write $\delta_{ki} = (z_{ik} - t_{ik})$.

Solution:

$$\mathcal{S} = \{i : \; z_{ki} > 0\}$$