ECE/CS 559 - Fall 2017 - Homework #4
Due: 10/16/2017, the end of class.
This is the midterm of Fall 2016.

Erdem Koyuncu

Note: All notes in the beginning of Homework #1 apply. Only a subset of the problems may be graded.

- **Q1 (24 pts):** In the following, we follow the convention that $x_0 = 1$.

  Recall that the step-activation function $u : \mathbb{R} \to \{0,1\}$ is defined as $u(v) = 1$ if $v \geq 0$, and $u(v) = 0$, otherwise. Then, for n inputs $x_1, \ldots, x_n$, a **perceptron** can be defined via the input-output relationship

  $$y' = u\left(\sum_{i=0}^{n} w_i x_i\right) = u(w_0 + w_1 x_1 + \cdots + w_n x_n),$$

  where y' is the perceptron output, $w_1, \ldots, w_n$ are the synaptic weights, and w_0 is the bias term.

  We define a new type of neuron, namely, a **sauron**, via the input-output relationship

  $$y = u\left(\prod_{i=0}^{n} (w_i + x_i)\right) = u((w_0 + 1)(w_1 + x_1) \cdots (w_n + x_n)),$$

  where y is called the sauron output.

  Let the real number 1 represent a `TRUE`, and the real number 0 represent a `FALSE`.

  (a) **(8 pts):** Let $n = 1$. Does there exist w_0, w_1 such that $y = 1 - x_1$ for $x_1 \in \{0,1\}$? In other words, can a single sauron implement the `NOT` gate? If your answer is "Yes," find specific w_0, w_1 such that the sauron implements the `NOT` gate. If your answer is "No," prove that no choice for w_0, w_1 can result in a sauron that implements the `NOT` gate.

  (b) **(8 pts):** Let $n = 2$. Does there exist w_0, w_1, w_2 such that $y = x_1 x_2$ for $x_1, x_2 \in \{0,1\}$? In other words, can a single sauron implement the `AND` gate? Justify your answer as in (a).

  (c) **(8 pts):** Let $n = 2$. Does there exist w_0, w_1, w_2 such that $y = ((x_1 + x_2) \mod 2)$ for $x_1, x_2 \in \{0,1\}$? In other words, can a single sauron implement the `XOR` gate? Justify your answer as in (a).

- **Q2 (26 pts):** In the following, consider only neurons with the step-activation function $u(\cdot)$.

  (a) **(13 pts):** Let $\mathcal{C}_0 = \{\begin{bmatrix}1\\1\end{bmatrix}, \begin{bmatrix}-1\\1\end{bmatrix}, \begin{bmatrix}0\\4\end{bmatrix}\}$ and $\mathcal{C}_1 = \{\begin{bmatrix}0\\2\end{bmatrix}, \begin{bmatrix}0\\-1\end{bmatrix}, \begin{bmatrix}0\\-2\end{bmatrix}\}$ as illustrated in Figure (i). Members of classes $\mathcal{C}_0$ and $\mathcal{C}_1$ are represented by black disks and crosses, respectively.

  [I] **(7 pts):** We wish to design a perceptron $y = u(w_0 + w_1 x_1 + w_2 x_2)$ such that $y = 0$ if $\begin{bmatrix}x_1\\x_2\end{bmatrix} \in \mathcal{C}_0$ and $y = 1$ if $\begin{bmatrix}x_1\\x_2\end{bmatrix} \in \mathcal{C}_1$. Suppose that we use the perceptron training algorithm for this purpose with initial weights $w_0 = 1$, $w_1 = 0$, $w_2 = 1$ and learning rate $\eta = 1$. Either prove that the algorithm will converge, or prove that it will not converge. You may use the perceptron convergence theorem.

  [II] **(6 pts):** Design a neural network (single-layer or multi-layer) such that the network provides an output of 0 if $\begin{bmatrix}x_1\\x_2\end{bmatrix} \in \mathcal{C}_0$ and an output of 1 if $\begin{bmatrix}x_1\\x_2\end{bmatrix} \in \mathcal{C}_1$.

  (b) **(13 pts)** Repeat (a) for classes $\mathcal{C}_0 = \{\begin{bmatrix}1\\1\end{bmatrix}, \begin{bmatrix}-1\\1\end{bmatrix}, \begin{bmatrix}0\\4\end{bmatrix}\}$ and $\mathcal{C}_1 = \{\begin{bmatrix}0\\0\end{bmatrix}, \begin{bmatrix}0\\-1\end{bmatrix}, \begin{bmatrix}0\\-2\end{bmatrix}\}$ illustrated in Figure (ii). Note that the only difference is that now, instead of the point $\begin{bmatrix}0\\2\end{bmatrix}$, we have the point $\begin{bmatrix}0\\0\end{bmatrix}$ in class $\mathcal{C}_1$.

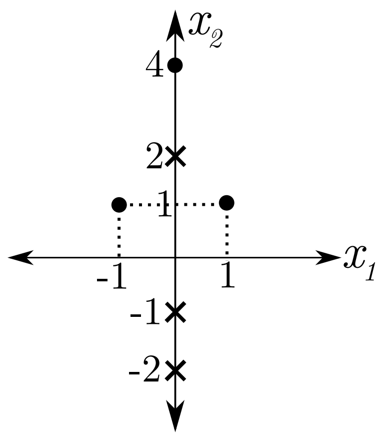


Figure (i)

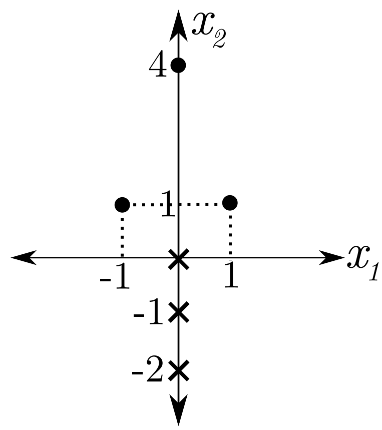


Figure (ii)

In the following, $\log(\cdot)$ is the natural logarithm, e is the base of the natural logarithm, $|\cdot|$ is the absolute value ($|x| = x$ if $x \geq 0$, and $|x| = -x$ if $x < 0$), and $\mathbb{R}$ is the set of real numbers. Recall $\frac{\partial e^x}{\partial x} = e^x$, and $e^{\log x} = x$, $x > 0$.

- **Q3 (25 pts):** Consider the activation function

$$f(v) = \begin{cases} 1 - e^{-v}, & v \geq 0, \\ -1 + e^{-|v|}, & v < 0. \end{cases}$$

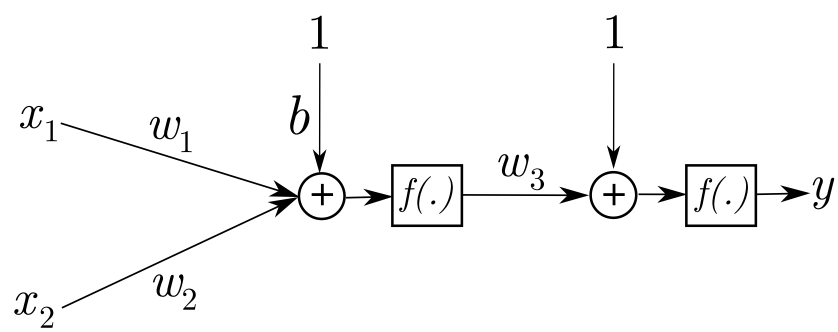
For (a)-(c), consider a single-neuron network with input-output relationship $y = f(b + \mathbf{w}^T \mathbf{x})$, where y is the network output, b is the bias term, $\mathbf{w} = \left[\begin{smallmatrix} w_1 \\ w_2 \end{smallmatrix}\right]$ are the synaptic weights, and $\mathbf{x} = \left[\begin{smallmatrix} x_1 \\ x_2 \end{smallmatrix}\right]$ is the network input.

(a) **(3 pts):** Calculate the derivative $f'(v)$ in closed form.

(b) **(7 pts):** Let $E = (d - y)^2$, where d is a constant (a generic desired output). Find the delta-learning rule (the gradient-descent update equations) for b, w_1, w_2 given learning parameter $\eta = \frac{1}{2}$. Remember that you can write a single (vectorized) update equation that can handle all variables. The final update expression(s) may however contain only the terms/functions $d, y, f', f, \mathbf{w}, w_1, w_2, b$.

(c) **(6 pts):** Consider the same delta-learning setup as in (b). Consider the training vectors $\mathbf{x}_1 = \begin{bmatrix} \log 2 \\ \log 3 \end{bmatrix}$, $\mathbf{x}_2 = \begin{bmatrix} 0 \\ \log 2 \end{bmatrix}$, with desired outputs $d_1 = \frac{2}{3}$, $d_2 = \frac{5}{2}$, respectively. Find the updated bias and the updated synaptic weights after one epoch of online learning given initial conditions $b = 0$, $\mathbf{w} = \left[\begin{smallmatrix} 0 \\ 1 \end{smallmatrix}\right]$.

(d) **(9 pts)** Consider now a multi-layer network as shown below.



Let $E = e^{(d-y)^4}$. Find the gradient-descent update equations for b, w_1, w_2, w_3 given learning parameter $\eta = \frac{1}{4}$. The final expression may only contain the terms/functions $d, y, f', f, \mathbf{w}, w_1, w_2, b, w_3$.

Hint: You do not have to apply some algorithm. Just write the input-output relationship of the network, and use chain rule. This is more of a calculus problem than a neural network problem!

- **Q4 (25 pts):** Design a (possibly multi-layer) neural network with two inputs $x_1, x_2 \in \mathbb{R}$ and a single output y such that $y = 1$ if $x_1 = 3$ and $x_2 = 2$, and $y = 0$, otherwise. In other words, the network output should assume a value of 1 only at the single point $(x_1, x_2) = (3, 2)$ of the input space $\mathbb{R}^2$. Note that the inputs can be any real numbers. The neuron(s) of the network should use the step-activation function $u : \mathbb{R} \to \{0, 1\}$ given by $u(v) = 1$ if $v \geq 0$, and $u(v) = 0$, otherwise. The network should consist of **5 neurons at most**. Spoilers and their partial credits:

  (a) **(3 pts):** Design a network with input x_1 and output y such that $y = 1$ if $x_1 \geq 3$, and $y = 0$, otherwise.

  (b) **(4 pts):** Design a network with input x_1 and output y such that $y = 1$ if $x_1 \leq 3$, and $y = 0$, otherwise.

  (c) **(6 pts):** Design a network with input x_1 and output y such that $y = 1$ if $x_1 = 3$, and $y = 0$, otherwise.

  (d) **(6 pts):** Solve the original problem without any restrictions on the number of neurons.

  (e) **(6 pts):** Solve the original problem.

  You are not required to do (a)-(d) provided you can provide a correct solution to (e).