

ECE/CS 559 - Fall 2019 - Midterm #1.

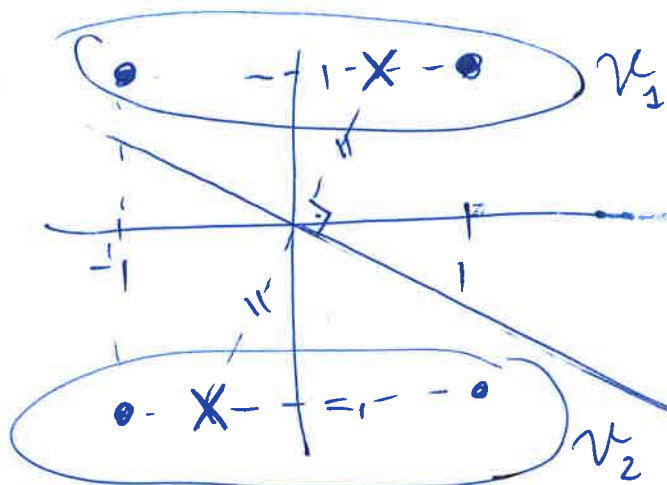
Full Name:

James Bond

ID Number:

007

Q1 (10 pts). Consider applying the k -means algorithm to the set of vectors $C = \left\{ \begin{bmatrix} -1 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ -1 \end{bmatrix}, \begin{bmatrix} -1 \\ -1 \end{bmatrix} \right\}$ with initial centers $\begin{bmatrix} 0.5 \\ 1 \end{bmatrix}, \begin{bmatrix} -0.5 \\ -1 \end{bmatrix}$. What are the resulting centers after the algorithm converges? Recall that, given S is the input (training set), and c_i are the centers, the k -means algorithm relies on the update $c_i \leftarrow \frac{1}{|V_i|} \sum_{x \in V_i} x$, where $V_i = \{x \in S : \|x - c_i\| \leq \|x - c_j\|, \forall j\}$, and $|V_i|$ is the number of elements in V_i .



Hence, at iteration 1, the centers will be updated to $\begin{pmatrix} 0 \\ 1 \end{pmatrix}$ and $\begin{pmatrix} 0 \\ -1 \end{pmatrix}$.

These centers will remain the same at subsequent iterations as V_1 & V_2 does not change.

Q2 (30 pts). Let u be the step activation function with $u(x) = 1$ if $x \geq 0$, and $u(x) = 0$, otherwise. Consider the perceptron $y = u(w_0 + w_1x_1 + w_2x_2)$, where w_1 and w_2 are the weights for inputs x_1 and x_2 , respectively, w_0 is the perceptron bias, and y is the perceptron output. Let $C_0 = \left\{ \begin{bmatrix} 0 \\ 2 \end{bmatrix}, \begin{bmatrix} 2 \\ 0 \end{bmatrix} \right\}$, and $C_1 = \left\{ \begin{bmatrix} 1 \\ 1 \end{bmatrix} \right\}$. The desired output for class C_0 is 0, and the desired output for class C_1 is 1. Correspondingly, let $d(x) = 0$ if $x \in C_0$, and otherwise, let $d(x) = 1$ if $x \in C_1$.

(a) (10 pts) If possible, find w_0, w_1, w_2 that can separate C_0 and C_1 (i.e., provide the desired output for all 4 possible input vectors). Otherwise, prove that no choice of weights can separate the two classes.

We need

$$w_0 + 2w_2 < 0 \quad (i)$$

$$w_0 + 2w_1 < 0 \quad (ii)$$

$$w_0 + w_1 + w_2 \geq 0 \quad (iii)$$

$$\frac{-(i) - (ii)}{2} + (iii) \text{ leads to } 0 > 0$$

a contradiction.

So no choice of weights can separate the classes.

- (b) (10 pts) Recall that the perceptron training algorithm relies on the update $\mathbf{w} \leftarrow \mathbf{w} + \eta(d(\mathbf{x}) - y)[1 \ \mathbf{x}]$, where $\mathbf{w} = [w_0 \ w_1 \ w_2]$ is the weight vector. Let $\eta = 1$ and the initial weight vector be given by $\mathbf{w} = [-1 \ 0 \ 0]$. Calculate the updated weights after one epoch of training.

With extended notation (biases)

$$C_0 = \left\{ \begin{pmatrix} 1 \\ 0 \\ 2 \end{pmatrix}, \begin{pmatrix} 1 \\ 2 \\ 0 \end{pmatrix} \right\}, \quad C_1 = \left\{ \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \right\}$$

Epoch 1.

Feed $\begin{pmatrix} 1 \\ 0 \\ 2 \end{pmatrix}$. $y = u\left(\begin{pmatrix} -1 \\ 0 \\ 0 \end{pmatrix}^T \begin{pmatrix} 1 \\ 0 \\ 2 \end{pmatrix}\right) = 0 = \text{desired output}$
So, no update.

Feed $\begin{pmatrix} 1 \\ 2 \\ 0 \end{pmatrix}$, $y = u\left(\begin{pmatrix} -1 \\ 0 \\ 0 \end{pmatrix}^T \begin{pmatrix} 1 \\ 2 \\ 0 \end{pmatrix}\right) = 0 = \text{desired output}$
So, again, no update.

Feed $\begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$, $y = u\left(\begin{pmatrix} -1 \\ 0 \\ 0 \end{pmatrix}^T \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}\right) = 0 \neq \text{desired output}$

So update $\mathbf{w} \leftarrow \begin{pmatrix} -1 \\ 0 \\ 0 \end{pmatrix} + 1 \cdot (1 - 0) \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix}$.

\therefore Find weights after one epoch of training
 $\begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix}$.

- (c) (10 pts) Will the weights provided by the algorithm (as setup in (b)) eventually converge after a sufficiently larger number of epochs? Justify your answer.

No. ~~Ques~~ If converges $\Rightarrow \exists w$ to separate the two classes. However, the classes are not linearly separable.

Q3 (40 pts): Consider a single-neuron network with input-output relationship $y = f(b + \mathbf{w}^T \mathbf{x})$, where y is the network output, f is some activation function, b is the bias term, $\mathbf{w} = \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}$ are the synaptic weights, and $\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$ is the network input.

- (a) (10 pts): Let $E = (d - y)^2$, where d is a constant (a generic desired output). Write down the delta-learning rule (the gradient-descent update equations) for b, w_1, w_2 given learning parameter $\eta = \frac{1}{2}$.

Simple chain rule yields:

$$\begin{pmatrix} b \\ w_1 \\ w_2 \end{pmatrix} \leftarrow \begin{pmatrix} b \\ w_1 \\ w_2 \end{pmatrix} + (d - y) f'(b + \mathbf{w}^T \mathbf{x}) \begin{pmatrix} 1 \\ x_1 \\ x_2 \end{pmatrix}.$$

(b) (15 pts): Consider the same delta-learning setup as in (a) with the activation function

$$f(v) = \begin{cases} v, & v \geq 0, \\ 0, & v < 0. \end{cases}$$

This is also known as the rectified linear unit (ReLU). Consider the training vectors $\mathbf{x}_1 = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$, $\mathbf{x}_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$, with desired outputs $d_1 = -1$, $d_2 = 2$, respectively. Find the updated bias and the updated weights after 2019 epochs of online learning given initial conditions $b = 0$, $\mathbf{w} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$.

Epoch 1

Show $\begin{pmatrix} 1 \\ -1 \end{pmatrix}$. Local field $v = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}^T \begin{pmatrix} 1 \\ -1 \end{pmatrix} = -1$

According to the update rule in (a), we obtain $y = f(v) = 0$

$$\begin{pmatrix} b \\ w_1 \\ w_2 \end{pmatrix} \leftarrow \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} + (-1 - 0) f'(-1) \begin{pmatrix} 1 \\ -1 \end{pmatrix}$$

Since $f'(-1) = 0$, we have no update.

Show $\begin{pmatrix} 1 \\ 1 \end{pmatrix}$. Local field $v = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}^T \begin{pmatrix} 1 \\ 1 \end{pmatrix} = 1$ $y = f(1) = 1$

$$\begin{pmatrix} b \\ w_1 \\ w_2 \end{pmatrix} \leftarrow \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} + (2 - 1) f'(1) \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 2 \end{pmatrix}$$

Epoch 2.

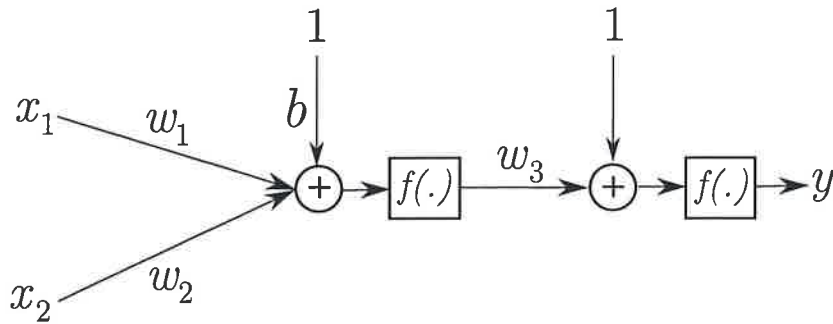
Show $\begin{pmatrix} 1 \\ -1 \end{pmatrix}$. $v = \begin{pmatrix} 1 \\ 0 \\ 2 \end{pmatrix}^T \begin{pmatrix} 1 \\ -1 \end{pmatrix} = -1 \Rightarrow$ no update.

Show $\begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}$. $v = \begin{pmatrix} 1 \\ 0 \\ 2 \end{pmatrix}^T \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} = 3$, $y = f(v) = 3$

$$\begin{pmatrix} b \\ w_1 \\ w_2 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 2 \end{pmatrix} + (2 - 3) f'(3) \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}$$

At the end of epoch 2, we end up with the same weights we began with. So after even numbered epochs, the weights are $\begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$, and after odd-numbered epochs, the weights are $\begin{pmatrix} 1 \\ 0 \\ 2 \end{pmatrix}$. In particular, after 2019 epochs, we get $\begin{pmatrix} 1 \\ 0 \\ 2 \end{pmatrix}$.

(c) (15 pts) Consider now a multi-layer network as shown below.



Consider again a general f , as in (a). Let $E = (d - y)^4$. Find the gradient-descent update equations for b, w_1, w_2, w_3 given $\eta = \frac{1}{4}$.

$y = f(1 + w_3 f(b + w_1 x_1 + w_2 x_2))$. So, by chain rule.

$$w_3 \leftarrow w_3 + (d - y)^3 f'(1 + w_3 f(b + w_1 x_1 + w_2 x_2)) \times f(b + w_1 x_1 + w_2 x_2).$$

$$\begin{bmatrix} b \\ w_1 \\ w_2 \end{bmatrix} \leftarrow \begin{bmatrix} b \\ w_1 \\ w_2 \end{bmatrix} + (d - y)^3 f'(1 + w_3 f(b + w_1 x_1 + w_2 x_2)) \times w_3 f'(b + w_1 x_1 + w_2 x_2) \begin{bmatrix} 1 \\ x_1 \\ x_2 \end{bmatrix}.$$

Q4 (20 pts). Let $f(x, y) = x^2 + xy + y^2$. Consider Newton's method given by the update rule $\begin{bmatrix} x \\ y \end{bmatrix} \leftarrow \begin{bmatrix} x \\ y \end{bmatrix} - H^{-1}g$, where the gradient and the Hessian are given by $g = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}$ and $H = \begin{bmatrix} \frac{\partial^2 f}{\partial x \partial x} & \frac{\partial^2 f}{\partial x \partial y} \\ \frac{\partial^2 f}{\partial y \partial x} & \frac{\partial^2 f}{\partial y \partial y} \end{bmatrix}$, respectively.

(a) (10 pts) Calculate g and H .

$$g = \begin{bmatrix} 2x + y \\ 2y + x \end{bmatrix} \quad H = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}.$$

(b) (10 pts) Consider the initial point $x = 2019, y = 2020$. Find the next point after one update with Newton's method.

$$H^{-1} = \frac{1}{3} \begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix}. \text{ So, } H^{-1}g = \begin{bmatrix} x \\ y \end{bmatrix}$$

and any initial point $\begin{bmatrix} x \\ y \end{bmatrix}$ will end up at

$\begin{bmatrix} x \\ y \end{bmatrix} - H^{-1}g = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$ after one update with Newton's method. This is also the global minimum.