ECE/CS 559 - Spring 2020 - Midterm # 1.

Full Name: **ID Number:**

Q1. Consider a neuron with $n \geq 1$ inputs $x_1, \ldots, x_n$, and output $y = \theta(w_0 + w_1 x_1 + \cdots + w_n x_n)$, where $w_0, w_1, \ldots, w_n$ are the neuron bias and weights, and the activation function is given by $\theta(x) = 1$ if $x \in [0, 1]$, and $\theta(x) = 0$ if $x \notin [0, 1]$. Note that the activation function is different than the functions that we have encountered throughout the lectures.

(a) **(15 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 neuron with activation function θ implement the NOT gate? If your answer is "Yes," find specific w_0, w_1 such that the neuron implements the NOT gate. If your answer is "No," prove that no choice for w_0, w_1 can result in a neuron that implements the NOT gate.

Yes, geometrically, the activation function provides an output of 1 on a strip whose width an orientation you can adjust by changing the weights and the bias. Thus, any logic function of 2 variables can be implemented.

$W_0 = 1/2 \quad W_1 = -1$ works for NOT gate.

(b) **(15 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 neuron with activation function θ implement the AND gate? Justify your answer as in (a).

Yes. $W_0 = -\frac{3}{2} \quad w_1 = w_2 = 1$ works.

(b) **(15 pts)** Recall that the perceptron training algorithm relies on the update $\mathbf{w} \leftarrow \mathbf{w}+\eta(d(\mathbf{x})-y)\begin{bmatrix}1 & \mathbf{x}\end{bmatrix}$, where $\mathbf{w} = \begin{bmatrix}w_0 & w_1 & w_2\end{bmatrix}$ is the weight vector. Let $\eta = 1$ and the initial weight vector be given by $\mathbf{w} = \begin{bmatrix}-1 & 0 & 0\end{bmatrix}$. Calculate the updated weights after one epoch of training.

For the update sequence $\begin{pmatrix}1\\0\\2\end{pmatrix}, \begin{pmatrix}1\\2\\0\end{pmatrix}, \begin{pmatrix}1\\1\\1\end{pmatrix}$,

you should get no update, no update, and finally update to:

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

after one epoch of training.

(c) **(15 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. If we had convergence, we should have had weights to separate the two classes. But (a) shows this is impossible.

(c) **(15 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 neuron with activation function θ implement the XOR gate? Justify your answer as in (a).

Yes. $w_0 = -\frac{1}{2}$ $w_1 = w_2 = 1$ work.

Q2. Let u be the step activation function with $u(x) = 1$ if $x \geq 0$, and $u(x) = 0$, otherwise. Consider the perceptron $y = u(w_0 + w_1 x_1 + w_2 x_2)$, 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 $\mathcal{C}_0 = \{[0 \;\; 2], [2 \;\; 0]\}$, and $\mathcal{C}_1 = \{[1 \;\; 1]\}$. The desired output for class $\mathcal{C}_0$ is 0, and the desired output for class $\mathcal{C}_1$ is 1. Correspondingly, let $d(\mathbf{x}) = 0$ if $\mathbf{x} \in \mathcal{C}_0$, and otherwise, let $d(\mathbf{x}) = 1$ if $\mathbf{x} \in \mathcal{C}_1$.

(a) **(15 pts)** If possible, find w_0, w_1, w_2 that can separate $\mathcal{C}_0$ and $\mathcal{C}_1$ (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

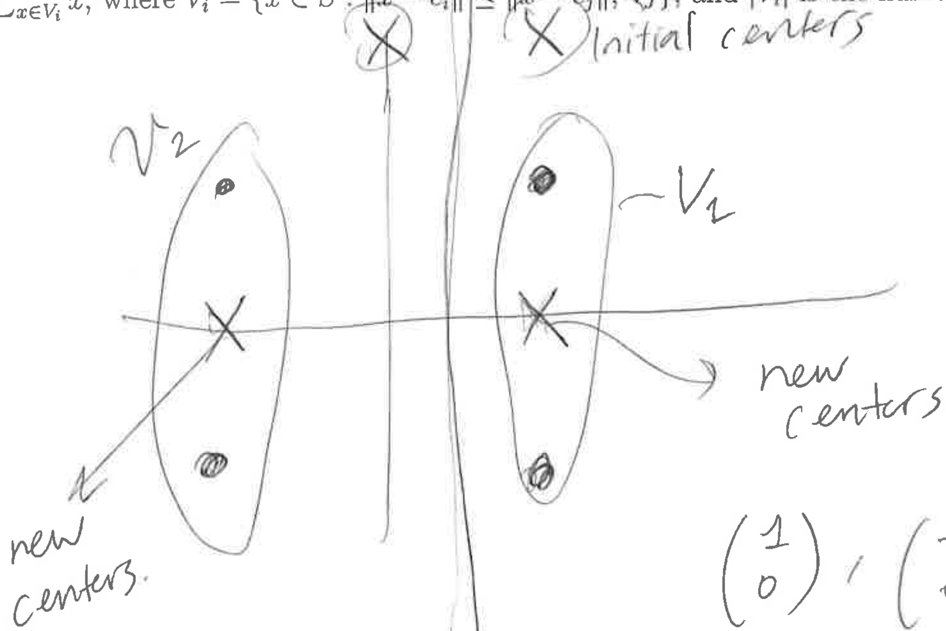
$$w_0 + 2w_2 < 0 \quad (i)$$
$$w_0 + 2w_1 < 0 \quad (ii)$$
$$w_0 + w_1 + w_2 \geq 0 \quad (iii)$$

$$\frac{-(i) - (ii)}{2} + (iii) \Leftrightarrow 0 > 0$$

a contradiction

So no choice of weights can seperate the two classes.

Q3 (10 pts). Consider applying the k-means algorithm to the set of vectors $C = \{\begin{bmatrix}-1\\1\end{bmatrix}, \begin{bmatrix}1\\1\end{bmatrix}, \begin{bmatrix}1\\-1\end{bmatrix}, \begin{bmatrix}-1\\-1\end{bmatrix}\}$ with initial centers $\begin{bmatrix}1\\2\end{bmatrix}, \begin{bmatrix}0\\2\end{bmatrix}$. 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.



$\begin{pmatrix}1\\0\end{pmatrix}, \begin{pmatrix}-1\\0\end{pmatrix}$

are the centers after one iteration.

they induce the same Voronoi cells V_1 & V_2.

in the same iterations.

Hence, the centers converge to or stay at $\begin{pmatrix}1\\0\end{pmatrix}, \begin{pmatrix}-1\\0\end{pmatrix}$.