University of Waterloo
CS 480
r2knowle: 2023-09-20

Exercise # 1

Q1) To begin we will consider the infinite sequence A where for any $n \in \mathbb{N}$ we have:

$$a_n = \begin{cases} \left(\begin{bmatrix} 0.1 \\ 0.02 \\ 0 \end{bmatrix}, 1 \right) & \text{if } n \equiv 0 \bmod 3 \\ \left(\begin{bmatrix} 0 \\ 0.1 \\ 0 \end{bmatrix}, -1 \right) & \text{if } n \equiv 1 \bmod 3 \\ \left(\begin{bmatrix} -0.1 \\ -0.02 \\ 0.1 \end{bmatrix}, 1 \right) & \text{if } n \equiv 2 \bmod 3 \end{cases}$$

In other words our infinite sequence will look something like:

$$A = \left(\begin{bmatrix} 0.1 \\ 0.02 \\ 0 \end{bmatrix}, 1 \right), \left(\begin{bmatrix} 0 \\ 0.1 \\ 0 \end{bmatrix}, -1 \right), \left(\begin{bmatrix} -0.1 \\ -0.02 \\ 0.1 \end{bmatrix}, 1 \right), \left(\begin{bmatrix} 0.1 \\ 0.02 \\ 0 \end{bmatrix}, 1 \right), ...$$

Lets start by proving the first property of the question, namely that this sequence is linearly separable. From the lectures we know this sequence is linearly separable if and only if there exists a weight function w such that $\forall n$ and some constant $s > 0$ (where x_n and y_n represent the data and label for term a_n):

$$(y_n * x_n)^\top w \geq s$$

Without loss of generality lets pick s to be a very small number such as 0.0001. We will also pick our weight function to be:

$$w = \begin{bmatrix} 0.1 \\ -0.08 \\ 0.1 \end{bmatrix}$$

Now since there are only 3 distinct terms, proving that each a_0, a_1, a_2 satisfy the equation will be sufficient to prove that the whole sequence satisfies the equation. Therefore we get:

$$\begin{aligned} (y_0 * x_0)^\top w &= \left(1 * \begin{bmatrix} 0.1 \\ 0.02 \\ 0 \end{bmatrix} \right)^\top \begin{bmatrix} 0.1 \\ -0.08 \\ 0.1 \end{bmatrix} \\ &= \begin{bmatrix} 0.1 \\ 0.02 \\ 0 \end{bmatrix}^\top \begin{bmatrix} 0.1 \\ -0.08 \\ 0.1 \end{bmatrix} \\ &= (0.1 \times 0.1) + (0.02 \times -0.08) + (0 \times 0.1) \\ &= 0.01 - 0.0016 + 0 \\ &= 0.0084 > 0.0001 \end{aligned}$$

$$
\begin{aligned}
(y_1 * x_1)^\top w &= \left(-1 * \begin{bmatrix} 0 \\ 0.1 \\ 0 \end{bmatrix}\right)^\top \begin{bmatrix} 0.1 \\ -0.08 \\ 0.1 \end{bmatrix} \\
&= \begin{bmatrix} 0 \\ -0.1 \\ 0 \end{bmatrix}^\top \begin{bmatrix} 0.1 \\ -0.08 \\ 0.1 \end{bmatrix} \\
&= (0 \times 0.1) + (-0.01 \times -0.08) + (0 \times 0.1) \\
&= 0 + 0.0008 + 0 \\
&= 0.0008 > 0.0001
\end{aligned}
$$

$$
\begin{aligned}
(y_2 * x_2)^\top w &= \left(1 * \begin{bmatrix} -0.1 \\ -0.02 \\ 0.1 \end{bmatrix}\right)^\top \begin{bmatrix} 0.1 \\ -0.08 \\ 0.1 \end{bmatrix} \\
&= \begin{bmatrix} -0.1 \\ -0.02 \\ 0.1 \end{bmatrix}^\top \begin{bmatrix} 0.1 \\ -0.08 \\ 0.1 \end{bmatrix} \\
&= (-0.1 \times 0.1) + (-0.02 \times -0.08) + (0.1 \times 0.1) \\
&= -0.01 + 0.0016 + 0.01 \\
&= 0.0016 > 0.0001
\end{aligned}
$$

We have now proven that these points are indeed linearly separable. We will also prove that the l2 norm of each of these terms of the sequence is less then 1. We will do this by calculating the l2 nrom for the only 3 distinct terms:

$$
\begin{aligned}
||x_0||_2 &= \sqrt{(0.01 + 0.04)} \\
&= 0.22360679775 < 1
\end{aligned}
$$

$$
\begin{aligned}
||x_1||_2 &= \sqrt{(0.01)} \\
&= 0.1 < 1
\end{aligned}
$$

$$
\begin{aligned}
||x_2||_2 &= \sqrt{(0.01 + 0.04 + 0.01)} \\
&= 0.244948974278 < 1
\end{aligned}
$$

We now need to prove the last property, note that the sequence repeats every 3 terms and that the weight will always update regardless of if a point is classified. Importantly this means that the weight at any n divisible by 3 will be equal to:

$$
w = \left\lfloor \frac{n}{3} \right\rfloor \begin{bmatrix} 0 \\ -0.1 \\ 0.1 \end{bmatrix}
$$

Now to prove that this version of perceptron makes an infinite number of mistakes across the infinite sequence we will use proof by contradiction. Lets start by assuming that this version of perceptron doesnt make an infinite number of mistakes, this would imply that for any $n \equiv 0 \bmod 3$ that we could correctly classify a_n in other words this implies:

$$(y_n * x_n)^\top w \geq 0$$

Expanding this we get:

$$\begin{aligned}
(y_i * x_i)^\top w &= \left(1 * \begin{bmatrix} 0.1 \\ 0.02 \\ 0 \end{bmatrix}\right)^\top \begin{bmatrix} 0 \\ -\lfloor \frac{n}{3} \rfloor \times 0.1 \\ \lfloor \frac{n}{3} \rfloor \times 0.1 \end{bmatrix} \\
&= \begin{bmatrix} 0.1 \\ 0.02 \\ 0 \end{bmatrix}^\top \begin{bmatrix} 0 \\ -\lfloor \frac{n}{3} \rfloor \times 0.1 \\ \lfloor \frac{n}{3} \rfloor \times 0.1 \end{bmatrix} \\
&= (0.1 \times 0) + (-0.02 \times \left\lfloor \frac{n}{3} \right\rfloor) + (0 \times \left\lfloor \frac{n}{3} \right\rfloor) \\
&= -0.02 \times \left\lfloor \frac{n}{3} \right\rfloor
\end{aligned}$$

Since we assumed that this is classifies any a_n correctly for any $n \equiv 0 \bmod 3$ this would imply that:

$$-0.02 \times \left\lfloor \frac{n}{3} \right\rfloor > 0$$

However this can only occur when n is negative. Which since the start of our sequence is at a_0 and n only increases means that we have a contradiction. Thus this version of perceptron will make an infinite number of mistakes.

Q2) To begin consider the following 2 points to maximize the marginal half space:

$$\begin{aligned}
\text{p1} &= (1, 1), 1 \\
\text{p2} &= (2, 0), -1
\end{aligned}$$

If we run perception on the first point, it will incorrectly classify it however after wards it will always correctly classify p1 and p2. w will be (1,1) and b will be 1. Note that the marginal half space is as large as possible as p1 is barley classified and $w^T p2 + b$ is maximized, as theres no other combination of w and b that classify p1 and provide a larger value. Note the margin between the two points is > 1. To minimize marginal half space consider the $\epsilon = 0.25$ and the points:

$$\begin{aligned}
\text{p1} &= (1, 1), 1 \\
\text{p2} &= (10, 9.99), -1
\end{aligned}$$

If we run perception on the first point, it will incorrectly classify it however after wards it will always correctly classify p1 and p2. w will be (1,1) and b will be 1. Note that the margin for both values will be far below ϵ, thus converging to an arbitrary small half space. Note the margin between the two points is > 1.

Q3) We are given that SDG updates w using the learning rate η and the loss function $Q_i(w)$:

$$w = w - \eta \Delta Q_i(w)$$

if we set η to be -1, we then get:

$$w = w + \Delta Q_i(w)$$

Setting $Q_i = w x_i y_i$ and taking the derivative gives us:

$$w = w + x_i y_i$$

which is the same as perceptron.