CS 188 Fall 2021 Introduction to Artificial Intelligence Challenge Q7 HW10

- **Due:** Tuesday 11/30 at 10:59pm.
- **Policy:** Can be solved in groups (acknowledge collaborators) but must be submitted individually.
- **Make sure to show all your work and justify your answers**.
- **Note:** This is a typical exam-level question. On the exam, you would be under time pressure, and have to complete this question on your own. We strongly encourage you to first try this on your own to help you understand where you currently stand. Then feel free to have some discussion about the question with other students and/or staff, before independently writing up your solution.
- Your submission on Gradescope should be a PDF that matches this template. Each page of the PDF should align with the corresponding page of the template (page 1 has name/collaborators, question begins on page 2.). **Do not reorder, split, combine, or add extra pages**. The intention is that you print out the template, write on the page in pen/pencil, and then scan or take pictures of the pages to make your submission. You may also fill out this template digitally (e.g. using a tablet.)

First name	NAMEERA
Last name	FAISAL AKHTAR
SID	3034244256
Collaborators	SARA IMAM

For staff use only:

Q7.	Neural Networks and Optimization	/17

Q7. [17 pts] Neural Networks and Optimization

7.1) (2pts) Which of the following models can represent exactly the XOR function for some choice of parameters? The truth table of the XOR gate $X \bigoplus Y$ is given below:

x	y	$x \oplus y$	$ax+by+c$	ax^2+by^2+c
0	0	0	c	c
0	1	1	$b+c$	$b+c$
1	0	1	$a+c$	$a+c$
1	1	0	$a+b+c$	$a+b+c$

When no specific details are mentioned (for instance number of layers, size of layers etc.), that option is correct if there exists **some** architecture that can represent the function correctly.

A) Logistic regression has a threshold value T, and all
$\sigma(x) < T \Rightarrow x = 0$
$\sigma(x) \geq T \Rightarrow x = 1$.
Also, σ is monotonically increasing. This means that in order of ascending x, we cannot go from 0 to 1 back to 0. So this doesn't work.

A. A logistic regression model that uses only x and y as features.

B. A logistic regression model that uses only x^2 and y^2 as features.

C. A neural network with no hidden layers.

D. A neural network with one hidden layer.

B) x^2, y^2 have same effect as x, y for 0/1 variable.

C) If correct, $\alpha(c) = \alpha(a+b+c) = 0$. We know $c \leq a+b+c$ and α monotonic $\Rightarrow$ all x in $[c, a+b+c]$ must have $\alpha(x) = 0$. This means $\alpha(a+c) = 0$, which is false, b/c $x=1$ and $y=0$ has $x \oplus y = 1$. Doesn't work.

D) Always works.

7.2) (2pts) Which of the following models can represent the function $\neg(x \vee y)$? The truth table for that function is given below:

x	y	$\neg(x \vee y)$	$ax+by+c$	ax^2+by^2+c
0	0	1	c	c
0	1	0	$b+c$	$b+c$
1	0	0	$a+c$	$a+c$
1	1	0	$a+b+c$	$a+b+c$

Again, if no specific details are mentioned (for instance number of layers, size of layers etc.), that option is correct if there exists **some** architecture that can represent the function correctly.

A) Logistic regression has a threshold value T, and all
$\sigma(x) < T \Rightarrow x = 0$
$\sigma(x) \geq T \Rightarrow x = 1$.
Also, σ is monotonically increasing. For positive c and negative a, b $c \geq b+c, a+c \geq a+b+c$ is true. So this works.

A. A logistic regression model that uses only x and y as features.

B. A logistic regression model that uses only x^2 and y^2 as features.

C. A neural network with no hidden layers.

D. A neural network with one hidden layer.

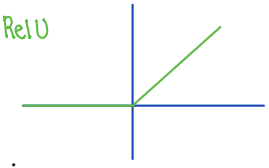
C) CASE 1: $a \geq b$
If correct, $\alpha(b+c) = \alpha(a+b+c) = 0$. We know $b+c \leq a+b+c$ and α monotonic $\Rightarrow$ all x in $[b+c, a+b+c]$ must have $\alpha(x) = 0$. This means $\alpha(a+c) = 0$, which is true. Works.
CASE 2: $a < b$. Works by same logic as above.

B) x^2, y^2 have same effect as x, y for 0/1 variable.

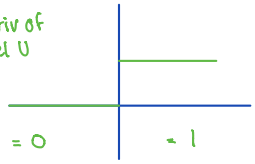
D) Always works.

Let's consider a neural network with no hidden layers (one-layer network) that predicts the value $f(x) \in \mathbb{R}$ from an input $x \in \mathbb{R}$, i.e. $f(x) = g(Wx + b)$ with $g(y) = \max(y, 0)$ denoting the ReLU activation function. The initial value for the weight and the bias is $W = -1$ and $b = 0$.

ReLU



Deriv of ReLU



7.3) (1pts) Let the derivative of the ReLU function be given as:

$$\frac{\partial ReLU(y)}{\partial y} = \begin{cases} s, & y > u \\ t, & y < u \end{cases}.$$

What are the correct values for s, t and u? Choose one answer from each column.

A. $s = 0$	A. $t = 0$	A. $u = -1$
B. $s = 1$	B. $t = 1$	B. $u = 0$
C. $s = y$	C. $t = y$	C. $u = 1$

7.4) (3pts) Compute the following partial derivatives using $W = -1$ and $b = 0$ as the parameter values.

$$\left.\frac{\partial f}{\partial W}\right|_{x=-1} = \frac{\delta}{\delta W} g(Wx+b)\Big|_{x=-1} = g'(Wx+b)\cdot x\Big|_{x=-1} = g'(-x)\cdot x\Big|_{x=-1} = g'(1)\cdot -1 = 1\cdot -1 = \boxed{= -1}$$

$$\left.\frac{\partial f}{\partial W}\right|_{x=1} = \frac{\delta}{\delta W} g(Wx+b)\Big|_{x=1} = g'(Wx+b)\cdot x\Big|_{x=1} = g'(-x)\cdot x\Big|_{x=1} = g'(-1)\cdot 1 = 0\cdot 1 = \boxed{= 0}$$

7.5) (3pts) For positive inputs $x > 0$, what are the upper and lower bounds of the values that the gradient of f with respect to W can take? Assume again that $W = -1$ and $b = 0$.

$f = g(Wx+b)$
$= g(-x+0)$
$= g(-x)$
$= \max(-x, 0)$
For $x > 0$,
$\max(-x, 0) = 0$
$\therefore \frac{\partial f}{\partial W} = 0$ ALWAYS. $\Rightarrow$ $\frac{\partial f}{\partial W} \leq 0$ and $\frac{\partial f}{\partial W} \geq 0$

$\frac{\partial f}{\partial W} \leq \boxed{0}$

$\frac{\partial f}{\partial W} \geq \boxed{0}$

7.6) (2pts) Assume that the objective function is squared loss and that we are using gradient descent to obtain the optimal values for the parameters. The initial values are $W = -1$ and $b = 0$. If your dataset contains only samples with $x > 0$, then which of the following activation functions will result in the loss function decreasing with the number of gradient descent iterations? More than one option could be correct.

A. $g(y) = ReLU(y)$ (*rectifier*)

B. $g(y) = tanh(y)$ (*hyperbolic tangent*)

C. $g(y) = \sigma(y)$ (*sigmoid*)

As shown above, for $W = -1$ and $b = 0$, the derivative of the ReLU is always 0. B/c of this, the parameters will never update and the loss will stay the same over time.

However, for hyperbolic tangent and sigmoid, the derivative never becomes exactly zero so the loss is always decreasing in each iteration of the update.

Assume we are training a deep neural network and that we achieve a low training loss. However, the loss on the held-out dataset is rather large.

7.7) (2pts) Which of the following solutions could decrease the loss on the held-out dataset?

A. Increasing the number of layers.

B. Decreasing the number of layers.

Low training loss, high test loss → overfitting on training!
Decrease overfitting by removing some layers, so we don't capture "noise" in the data.

7.8) (2pts) Which of the following solutions could decrease the loss on the held-out dataset?

A. Increasing the size of each hidden layer.

B. Decreasing the size of each hidden layer.

Low training loss, high test loss → overfitting on training!
Decrease overfitting by removing parameters from each layer, which decreases the size of each hidden layer. This ensures we don't capture "noise" in the data.