Problem Set 2

Please submit a typed PDF addressing all problems below. This problem set contains 3 questions and is worth 25 points. **All responses MUST be in *your* own words.** Justification must be provided for **all** written answers. Statements made without any supporting explanation/justification will receive **no credit**. For mathematical derivations and plots, you may insert pictures of handwritten work if you find this easier. The required weekly readings and lecture slides should be helpful in completing the assignment. You can find these on our course website.

1. **Parameters vs. Hyper-Parameters [6 points]:**

   (a) Define and explain the fundamental difference between parameters and hyper-parameters. How is this difference related to training and validation dataset splits?

   Model parameters are the internal variable, such as the weights and biases, of the model that some optimization algorithm can adjust to minimize the training loss. They are learned from the training data.

   Hyper-parameters are external variables of the model, such as learning rate, that control how the model should train or what kind of model is trained. They are set by the practitioner and are not updated by some optimization algorithm.

   (b) For each of the rows in the below table, indicate which items are parameters (P) and which are hyper-parameters (HP). Justify your answers.

Item	P or HP?
A weight matrix **w**	P
The learning rate	HP
A bias term **b**	P
The minibatch size	HP
The non-linear activation function	HP
The optimizer	HP

(c) Provide examples of two hyper-parameters not present on the table in part (b). Justify your answers.

Weight decay λ → This scalar value is chosen by the user to penalize large weights. It influenced training but is never updated by back propagation.

Number of hidden layers/units per layer → This decides the model's capacity before training starts. Layer sizes stay fixed throughout optimization.

2. **Overfitting versus Underfitting [4 points]:**

   (a) You're training a deep learning classification model and observe the following:

   i. The validation accuracy increases quickly during the first 5 epochs, then begins to decrease.

   ii. The training loss decreases to zero during the first 10 epochs.

   Is the model overfitting or underfitting? Suggest a possible modification to the hyper-parameters that would improve model generalization.

   The model is underfitting since the validation loss could continue the trend of decreasing along with the training loss. We could increase the capacity of the model with more layers/units, reduce regularization, or raise the learning rate and train for more epochs to let the model fit the data.

   (b) You're training a deep learning classification model and observe the following:

   i. The validation accuracy increases slowly during the entire training run of 100 epochs.

   ii. The training loss decreases continually over the entire training run, but never approaches zero.

   Is the model overfitting or underfitting? Suggest a possible modification to the hyper-parameters that could improve model generalization.

   The model is overfitting as the gap between training loss and validation loss is increasing overtime. We could introduce some kind of regularization, such as increasing dropout rate, adding weight-decay, or using early stopping to reduce the model capacity.

3. **Gradient Backpropagation on Computational Graphs [15 points]**: For parts (a) and (b), a computational graph (with input values) is given. Your task for each part is threefold:

   i: Feed the provided input values forward through the computational graph to obtain the predicted value, $\hat{Y}$.

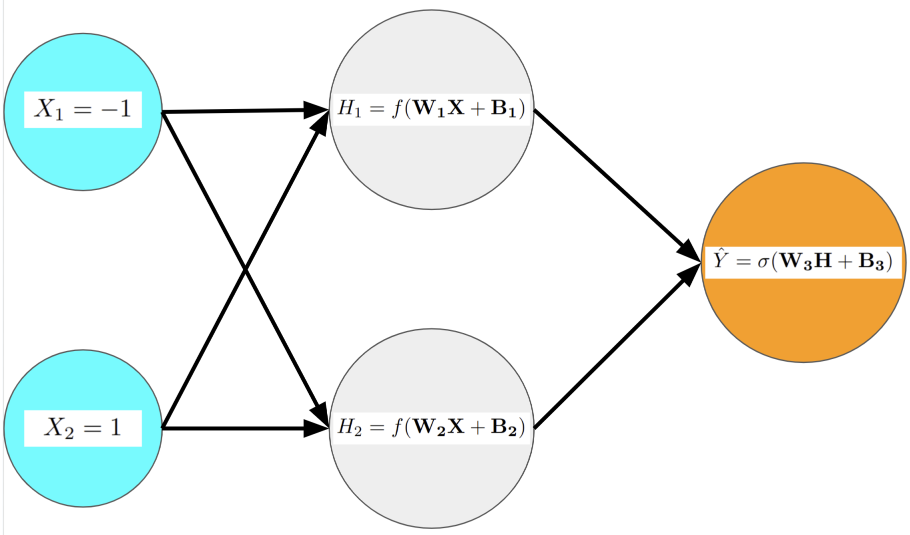
   ii: Compute the binary cross-entropy loss between $\hat{Y}$ and the ground truth value Y.

   iii: Backpropagate the loss computed in part (ii) to obtain gradients for all parameters (W_i, B_i), hidden nodes (H_i), and input nodes (X_i).

You are only required to perform one forward/backward pass on each graph. Recall that the gradients for parameters/hidden nodes/etc. are partial derivatives of the loss function. In other words, your task is to find the quantities $\frac{dL}{dH}$, $\frac{dL}{dX}$, $\frac{dL}{dW}$, and $\frac{dL}{dB}$ for all relevant H, X, W, B. L denotes the loss. You will need to employ the chain rule for derivatives in order to accomplish this. For all mathematical derivations, you must **show your work**. Note that this is **NOT a programming assignment**; submissions based on automatic differentiation tools (such as PyTorch) will receive **no credit**. The details for each computational graph begin on the next page.

(a) **Standard Neural Network**

Consider the computational graph below:



$X_1 = -1$
$X_2 = 1$
$H_1 = f(\mathbf{W_1 X} + \mathbf{B_1})$
$H_2 = f(\mathbf{W_2 X} + \mathbf{B_2})$
$\hat{Y} = \sigma(\mathbf{W_3 H} + \mathbf{B_3})$

With the following parameter values:

$$\begin{aligned}\mathbf{W_1} &= [1, -1]\\ \mathbf{B_1} &= 0\\ \mathbf{W_2} &= [-1, 0.5]\\ \mathbf{B_2} &= 0.5\\ \mathbf{W_3} &= [1, 1]\\ \mathbf{B_3} &= -1\end{aligned}$$

σ denotes the sigmoid activation function. Use the "Leaky ReLU" non-linear activation function for $f(\cdot)$. Leaky ReLU is widely used in deep learning, and is defined as follows:

$$f(x) = max(x, 0.1x) \tag{1}$$

For the provided sample, the ground truth label is $Y = 0$. Follow steps (i-iii) to populate the below table with the appropriate values (**Show your work!**):

i.

$$X_1 = -1$$

$$X_2 = 1$$

$$\begin{aligned} H_1 &= f(W_1X + B_1) \\ &= f\left(\begin{bmatrix}1 & -1\end{bmatrix}\begin{bmatrix}-1\\1\end{bmatrix} + 0\right) \\ &= f(-1 - 1 + 0) \\ &= \max(-2, 0.1 \cdot (-2)) \\ &= -0.2 \end{aligned}$$

$$\begin{aligned} H_2 &= f(W_2X + B_2) \\ &= f\left(\begin{bmatrix}-1 & 0.5\end{bmatrix}\begin{bmatrix}-1\\1\end{bmatrix} + 0.5\right) \\ &= f(1 + 0.5 + 0.5) \\ &= \max(2, 0.1 \cdot 2) \\ &= 2 \end{aligned}$$

$$\begin{aligned} \hat{Y} &= \sigma(W_3H + B_3) \\ &= \sigma\left(\begin{bmatrix}1 & 1\end{bmatrix}\begin{bmatrix}0\\2\end{bmatrix} - 1\right) \\ &= \sigma(-0.2 + 2 - 1) \\ &= \frac{1}{1 + e^{-0.8}} \approx 0.690 \end{aligned}$$

ii.

$$\begin{aligned} L_{CE}(\hat{Y}, Y) &= -Y\log(\hat{Y}) - (1 - Y)\log(1 - \hat{Y}) \\ &= -0\log\left(\frac{1}{1 + e^{-0.8}}\right) - (1 - 0)\log\left(1 - \frac{1}{1 + e^{-0.8}}\right) \\ &\approx 1.171 \end{aligned}$$

iii.

Let $z_3 = W_3 H + B_3$ and $\hat{Y} = \sigma(z_3)$,

$$\frac{\partial L_{CE}}{\partial \hat{Y}} = \frac{\hat{Y} - Y}{\hat{Y}(1-\hat{Y})} = \frac{\sigma(z_3) - Y}{\sigma(z_3)(1-\sigma(z_3))}$$

$$\frac{\partial L_{CE}}{\partial z_3} = \frac{\partial L_{CE}}{\partial \hat{Y}} \frac{\partial \hat{Y}}{\partial z_3} = \left(\frac{\sigma(z_3) - Y}{\sigma(z_3)(1-\sigma(z_3))} \right) (\sigma(z_3)(1-\sigma(z_3))) = \sigma(z_3) - Y = 0.690 - 0 = 0.690$$

$$\frac{\partial L_{CE}}{\partial W_3} = \frac{\partial L_{CE}}{\partial z_3} \frac{\partial z_3}{\partial W_3} = 0.690H = 0.690 \begin{bmatrix} -0.2 & 2 \end{bmatrix} = \begin{bmatrix} -0.138 & 1.380 \end{bmatrix}$$

$$\frac{\partial L_{CE}}{\partial B_3} = \frac{\partial L_{CE}}{\partial z_3} \frac{\partial z_3}{\partial B_3} = 0.690(1) = 0.690$$

$$\frac{\partial L_{CE}}{\partial H_1} = \frac{\partial L_{CE}}{\partial z_3} \frac{\partial z_3}{\partial H_1} = 0.690 w_{3,1} = 0.690 \cdot 1 = 0.690$$

$$\frac{\partial L_{CE}}{\partial H_2} = \frac{\partial L_{CE}}{\partial z_3} \frac{\partial z_3}{\partial H_2} = 0.690 w_{3,2} = 0.690 \cdot 1 = 0.690$$

Let $z_2 = W_2 X + B_2$ and $H_2 = f(z_2)$,

$$\frac{\partial L_{CE}}{\partial z_2} = \frac{\partial L_{CE}}{\partial H_2} \frac{\partial H_2}{\partial z_2} = 0.690(1) = 0.690$$

$$\frac{\partial L_{CE}}{\partial W_2} = \frac{\partial L_{CE}}{\partial z_2} \frac{\partial z_2}{\partial W_2} = 0.690X = 0.690 \begin{bmatrix} -1 & 1 \end{bmatrix} = \begin{bmatrix} -0.690 & 0.690 \end{bmatrix}$$

$$\frac{\partial L_{CE}}{\partial B_2} = \frac{\partial L_{CE}}{\partial z_2} \frac{\partial z_2}{\partial B_2} = 0.690(1) = 0.690$$

Let $z_1 = W_1 X + B_1$ and $H_1 = f(z_1)$,

$$\frac{\partial L_{CE}}{\partial z_1} = \frac{\partial L_{CE}}{\partial H_1} \frac{\partial H_1}{\partial z_1} = 0.690(0.1) = 0.069$$

$$\frac{\partial L_{CE}}{\partial W_1} = \frac{\partial L_{CE}}{\partial z_1} \frac{\partial z_1}{\partial W_1} = 0.069X = 0.069 \begin{bmatrix} -1 & 1 \end{bmatrix} = \begin{bmatrix} -0.069 & 0.069 \end{bmatrix}$$

$$\frac{\partial L_{CE}}{\partial B_1} = \frac{\partial L_{CE}}{\partial z_1} \frac{\partial z_1}{\partial B_1} = 0.069(1) = 0.069$$

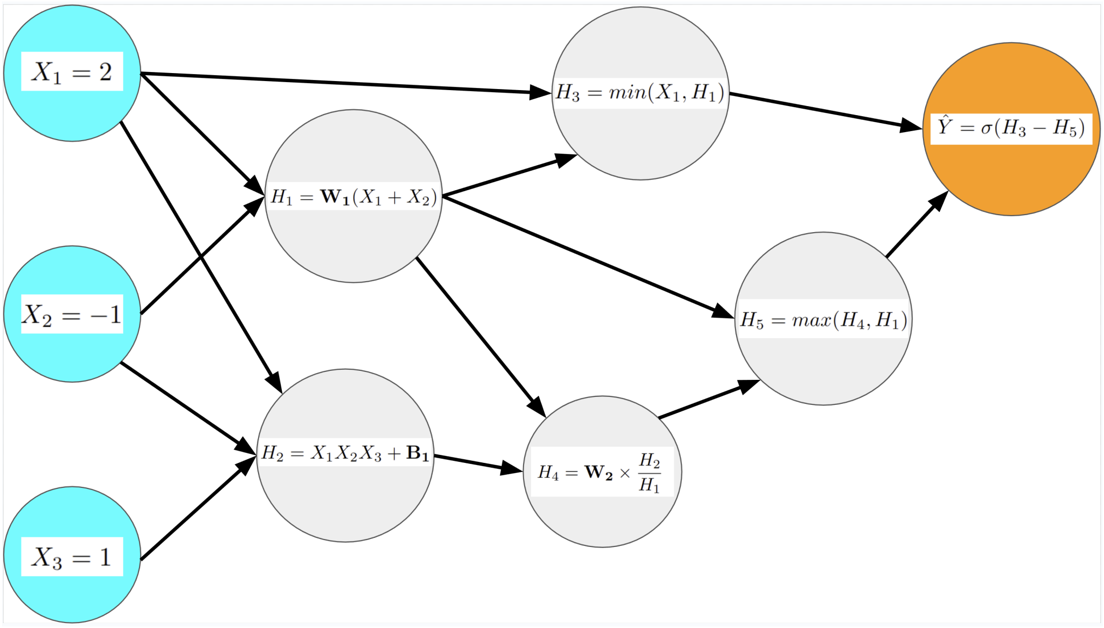
$$\frac{\partial L_{CE}}{\partial X_1} = \frac{\partial L_{CE}}{\partial z_1} \frac{\partial z_1}{\partial X_1} + \frac{\partial L_{CE}}{\partial z_2} \frac{\partial z_2}{\partial X_1} = 0.069 w_{1,1} + 0.690 w_{2,1} = 0.069 \cdot 1 + 0.690 \cdot (-1) = -0.621$$

$$\frac{\partial L_{CE}}{\partial X_2} = \frac{\partial L_{CE}}{\partial z_1} \frac{\partial z_1}{\partial X_2} + \frac{\partial L_{CE}}{\partial z_2} \frac{\partial z_2}{\partial X_2} = 0.069 w_{1,2} + 0.690 w_{2,2} = 0.069 \cdot (-1) + 0.690 \cdot 0.5 = 0.276$$

$\hat{Y}$	L	$\triangledown\mathbf{W_3}$	$\triangledown\mathbf{B_3}$	$\triangledown\mathbf{W_2}$	$\triangledown\mathbf{B_2}$	$\triangledown H_2$	$\triangledown\mathbf{W_1}$	$\triangledown\mathbf{B_1}$	$\triangledown H_1$	$\triangledown\mathbf{X_1}$	$\triangledown\mathbf{X_2}$
0.731	1.313	[-0.138,1.38]	0.690	[-0.690,0.690]	0.690	0.690	[-0.069,0.069]	0.069	0.690	-0.621	0.276

(b) **A More "Interesting" Graph**

Consider the computational graph below:


$X_1 = 2$
$X_2 = -1$
$X_3 = 1$
$H_1 = \mathbf{W_1}(X_1 + X_2)$
$H_2 = X_1 X_2 X_3 + \mathbf{B_1}$
$H_3 = min(X_1, H_1)$
$H_4 = \mathbf{W_2} \times \frac{H_2}{H_1}$
$H_5 = max(H_4, H_1)$
$\hat{Y} = \sigma(H_3 - H_5)$

With the following parameter values:

$$\mathbf{W_1} = 1.5$$
$$\mathbf{B_1} = 1$$
$$\mathbf{W_2} = -3$$

σ denotes the sigmoid activation function. For the provided sample, the ground truth label is $Y = 1$. Follow steps (i-iii) to populate the below table with the appropriate values (**Show your work!**):

i.

$$X_1 = 2, X_2 = -1, X_3 = 1$$

$$\begin{aligned}
H_1 &= W_1(X_1 + X_2) = 1.5(2 - 1) = 1.5 \\
H_2 &= X_1 X_2 X_3 + B_1 = (2)(-1)(1) + 1 = -2 + 1 = -1 \\
H_3 &= \min(X_1, H_1) = \min(2, 1.5) = 1.5 \\
H_4 &= W_2 \cdot \frac{H_2}{H_1} = -3 \cdot \frac{-1}{1.5} = -3 \cdot \frac{-2}{3} = 2 \\
H_5 &= \max(H_4, H_1) = \max(2, 1.5) = 2 \\
\hat{Y} &= \sigma(H_3 - H_5) = \sigma(1.5 - 2) = \frac{1}{1 + e^{0.5}} \approx 0.378
\end{aligned}$$

ii.

$$\begin{aligned}L_{CE}(\hat{Y},Y) &= -Y\log(\hat{Y}) - (1-Y)\log(1-\hat{Y}) \\ &= -\log\left(\frac{1}{1+e^{0.5}}\right) - (1-1)\log\left(1-\frac{1}{1+e^{0.5}}\right) \\ &\approx 0.974\end{aligned}$$

iii.

Let $z_5 = H_3 - H_5$ and $\hat{Y} = \sigma(z_5)$,

$$\begin{aligned}
\frac{\partial L_{CE}}{\partial \hat{Y}} &= \frac{\hat{Y}-Y}{\hat{Y}(1-\hat{Y})} = \frac{\sigma(z_5)-Y}{\sigma(z_5)(1-\sigma(z_5))} \\
\frac{\partial L_{CE}}{\partial z_5} &= \frac{\partial L_{CE}}{\partial \hat{Y}}\frac{\partial \hat{Y}}{\partial z_5} = \left(\frac{\sigma(z_5)-Y}{\sigma(z_5)(1-\sigma(z_5))}\right)(\sigma(z_5)(1-\sigma(z_5))) = \sigma(z_5) - Y = 0.378 - 1 = -0.622 \\
\frac{\partial L_{CE}}{\partial H_5} &= \frac{\partial L_{CE}}{\partial z_5}\frac{\partial z_5}{\partial H_5} = -0.622(-1) = 0.622 \\
\frac{\partial L_{CE}}{\partial H_4} &= \frac{\partial L_{CE}}{\partial H_5}\frac{\partial H_5}{\partial H_4} = 0.622(1) = 0.622 \\
\frac{\partial L_{CE}}{\partial H_3} &= \frac{\partial L_{CE}}{\partial z_5}\frac{\partial z_5}{\partial H_3} = -0.622(1) = -0.622 \\
\frac{\partial L_{CE}}{\partial H_2} &= \frac{\partial L_{CE}}{\partial H_4}\frac{\partial H_4}{\partial H_2} = 0.622 \cdot \frac{W_2}{H_2} = 0.622 \cdot \frac{-3}{1.5} = -1.245 \\
\frac{\partial L_{CE}}{\partial H_1} &= \frac{\partial L_{CE}}{\partial H_3}\frac{\partial H_3}{\partial H_1} + \frac{\partial L_{CE}}{\partial H_4}\frac{\partial H_4}{\partial H_1} + \frac{\partial L_{CE}}{\partial H_5}\frac{\partial H_5}{\partial H_1} = -0.622(1) + 0.622\left(-\frac{W_2 \cdot H_2}{H_1^2}\right) + 0.622(0) \\
&= -0.622 + 0.622\left(-\frac{-3\cdot(-1)}{1.5^2}\right) + 0 = -1.452 \\
\frac{\partial L_{CE}}{\partial W_2} &= \frac{\partial L_{CE}}{\partial H_4}\frac{\partial H_4}{\partial W_2} = 0.622 \cdot \frac{H_2}{H_1} = 0.622 \cdot \frac{-1}{1.5} = -0.415 \\
\frac{\partial L_{CE}}{\partial W_1} &= \frac{\partial L_{CE}}{\partial H_1}\frac{\partial H_1}{\partial W_1} = -1.452(X_1 + X_2) = -1.452(2-1) = -1.452 \\
\frac{\partial L_{CE}}{\partial B_1} &= \frac{\partial L_{CE}}{\partial H_2}\frac{\partial H_2}{\partial B_1} = -1.245(1) = -1.245 \\
\frac{\partial L_{CE}}{\partial X_3} &= \frac{\partial L_{CE}}{\partial H_2}\frac{\partial H_2}{\partial X_3} = -1.245(X_1X_2) = -1.245 \cdot 2 \cdot (-1) = 2.490 \\
\frac{\partial L_{CE}}{\partial X_2} &= \frac{\partial L_{CE}}{\partial H_1}\frac{\partial H_1}{\partial X_2} + \frac{\partial L_{CE}}{\partial H_2}\frac{\partial H_2}{\partial X_2} = -1.452W_1 - 1.245(X_1X_3) = (-1.452)(1.5) - (1.245)(2)(1) = -4.668 \\
\frac{\partial L_{CE}}{\partial X_1} &= \frac{\partial L_{CE}}{\partial H_1}\frac{\partial H_1}{\partial X_1} + \frac{\partial L_{CE}}{\partial H_2}\frac{\partial H_2}{\partial X_1} + \frac{\partial L_{CE}}{\partial H_3}\frac{\partial H_3}{\partial X_1} = -1.452W_1 - 1.245(X_2X_3) - 0.622(0) \\
&= (-1.452)(1.5) - (1.245)(-1)(1) = -0.934
\end{aligned}$$

$\hat{Y}$	L	$\triangledown H_5$	$\triangledown H_4$	$\triangledown H_3$	$\triangledown H_2$	$\triangledown H_1$	$\triangledown \mathbf{W_2}$	$\triangledown \mathbf{W_1}$	$\triangledown \mathbf{B_1}$	$\triangledown X_3$	$\triangledown X_2$	$\triangledown X_1$
0.378	0.974	0.622	0.622	-0.622	-1.245	-1.452	-0.415	-1.452	-1.245	2.490	-4.668	-0.934

You may find the following information useful:

The cross entropy loss function: Like mean-squared error (MSE), the cross entropy function can be used to calculate the loss between the predicted output of a neural network and the target. For the binary classification task, it is defined by:

$$L_{CE}(\hat{Y}, Y) = -Y log(\hat{Y}) - (1 - Y) log(1 - \hat{Y})$$

where $\hat{Y}$ is the prediction from the model, and Y is the target label (0 or 1). The first derivative is given by:

$$\frac{d}{d\hat{Y}} L_{CE} = \frac{\hat{Y} - Y}{\hat{Y}(1 - \hat{Y})}$$

The sigmoid function: The logistic sigmoid function is often used as a non-linearity in neural networks. It is defined by:

$$\sigma(x) = \frac{e^x}{1 + e^x} = \frac{1}{1 + e^{-x}}$$

Its first derivative is given by:

$$\frac{d}{dx}\sigma(x) = \frac{1}{1 + e^{-x}} \left(1 - \frac{1}{1 + e^{-x}}\right) = \sigma(x)(1 - \sigma(x))$$

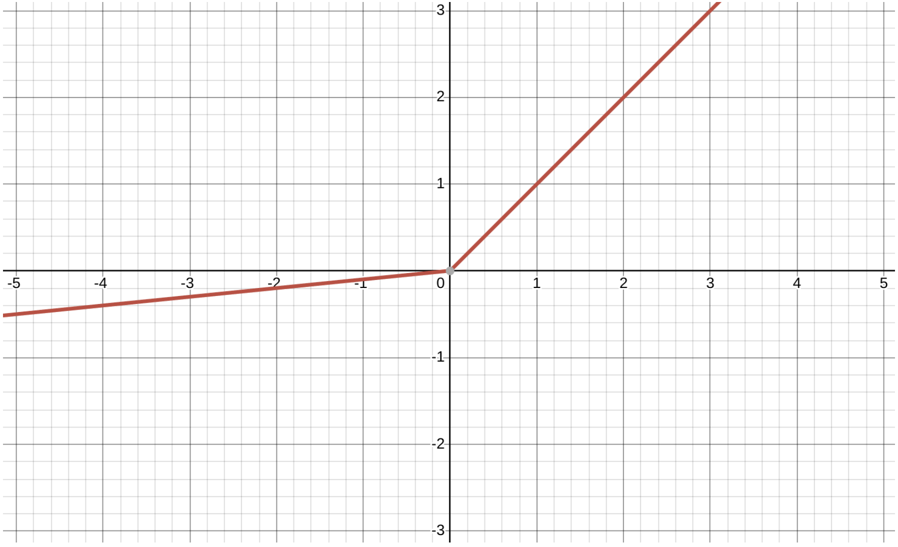
This function is closely related to the hyperbolic tangent function, which is another popular non-linear activation used in deep learning. *tanh* is defined by:

$$tanh(x) = \frac{e^{2x} - 1}{e^{2x} + 1} = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

The two functions are related by the identity:

$$tanh(x) = 2\sigma(2x) - 1 \qquad (2)$$

The Leaky ReLU activation function: Leaky ReLU, or (leaky) rectified linear unit, non-linearly modifies the input value depending on its sign. Equation (1) is shorthand for a piece-wise linear function which looks like this:



Examining the graph reveals that Leaky ReLU's derivative is given by:

$$\begin{aligned} f'(x) &= 1,\ x \geq 0 \\ f'(x) &= 0.1,\ \text{otherwise.} \end{aligned} \tag{3}$$

Note that while the slope for negative inputs is commonly 0.1, other variants exist. ReLU (not leaky) uses 0 for this slope, and sometimes other values are used as well.

4. **Extra Credit [2.5 points]:**

   (a) Verify the relationship between *tanh* and σ given by equation (2).

$$\tanh(x) = \frac{e^{2x}-1}{e^{2x}+1} = \frac{e^{x}-e^{-x}}{e^{x}+e^{-x}} = 2\sigma(2x) - 1$$

$$\begin{aligned} 2\sigma(2x) - 1 &= 2\left(\frac{1}{1+e^{-2x}}\right) - 1 \\ &= \frac{2}{1+e^{-2x}} - \frac{1+e^{-2x}}{1+e^{-2x}} \\ &= \frac{1-e^{-2x}}{1+e^{-2x}} \\ &= \frac{e^{2x}-1}{e^{2x}+1} \\ &= \tanh(x) \end{aligned}$$

(b) For both computational graphs, replace σ with $tanh$ in the final output "neuron". Compute a new loss using the mean squared error function, which is defined below:

$$MSE(Y,\hat{Y}) = \frac{1}{2}(\hat{Y} - Y)^2 \tag{4}$$

For the computational graph in Question 3, part (a), use a ground truth value of $Y = -1$ instead of $Y = 0$. Why is this change needed when we are using $tanh$ activation?

For 3a,

$$\hat{Y} = \tanh(W_3 H + B_3) = \tanh(0.8) \approx 0.664$$

$$MSE(Y,\hat{Y}) = \frac{1}{2}(\hat{Y} - Y)^2 = \frac{1}{2}(\tanh(0.8) - (-1))^2 \approx 1.385$$

The change from ground truth of Y = 0 to Y = -1 is needed since that is the range produced by tanh, and the tanh would never converge on a value of 0.

For 3b,

$$\hat{Y} = \tanh(H_3 - H_5) = \tanh(-0.5) \approx -0.462$$

$$MSE(Y,\hat{Y}) = \frac{1}{2}(\hat{Y} - Y)^2 = \frac{1}{2}(\tanh(-0.5) - 1)^2 \approx 1.069$$

(c) Show the steps of back-propagating the new loss through the final output "neuron". You do **NOT** need to back-prop the loss through the entire graphs, just show how the equations change when using the new activation function.

For 3a,

Let $z = W_3 H + B$ and $\hat{Y} = \tanh(z)$

$$\frac{\partial MSE}{\partial \hat{Y}} = \hat{Y} - Y$$

$$\frac{\partial MSE}{\partial z} = \frac{\partial MSE}{\partial \hat{Y}}\frac{\partial \hat{Y}}{\partial z} = (\hat{Y} - Y)(sech^2(z)) = (\tanh(0.80) - (-1))(sech^2(0.8)) = 0.930$$

For 3b,

Let $z = H_3 - H_5$ and $\hat{Y} = \tanh(z)$

$$\frac{\partial MSE}{\partial \hat{Y}} = \hat{Y} - Y$$

$$\frac{\partial MSE}{\partial z} = \frac{\partial MSE}{\partial \hat{Y}}\frac{\partial \hat{Y}}{\partial z} = (\hat{Y} - Y)(sech^2(z)) = (\tanh(-0.5) - 1)(sech^2(-0.5)) = -1.150$$

Collaboration versus Academic Misconduct: Collaboration with other students (or AI) is permitted, but the work you submit must be your own. Copying/plagiarizing work from another student (or AI) is not permitted and is considered academic misconduct. For more information about University of Colorado Boulder's Honor Code and academic misconduct, please visit the course syllabus.