Homework 1: Backpropagation

CSCI-GA 2572 Deep Learning

1 Theory (50pt)

1.1 Two-Layer Neural Nets

You are given the following neural net architecture:

$$\text{Linear}_1 \rightarrow f \rightarrow \text{Linear}_2 \rightarrow g$$

where $\text{Linear}_i(x) = \boldsymbol{W}^{(i)}\boldsymbol{x} + \boldsymbol{b}^{(i)}$ is the i-th affine transformation, and f, g are element-wise nonlinear activation functions. When an input $\boldsymbol{x} \in \mathbb{R}^n$ is fed to the network, $\tilde{\boldsymbol{y}} \in \mathbb{R}^K$ is obtained as the output.

1.2 Regression Task

We would like to perform regression task. We choose $f(\cdot) = 3(\cdot)^+ = 3\,\text{ReLU}(\cdot)$ and g to be the identity function. To train this network, we want to minimize the energy loss L and this is computed via the squared Euclidean distance cost C, such that $L(\boldsymbol{w}, \boldsymbol{x}, \boldsymbol{y}) = F(\boldsymbol{x}, \boldsymbol{y}) = C(\boldsymbol{y}, \tilde{\boldsymbol{y}}) = \|\tilde{\boldsymbol{y}} - \boldsymbol{y}\|^2$, where $\boldsymbol{y}$ is the output target.

(a) (1pt) Name and mathematically describe the 5 programming steps you would take to train this model with PyTorch using SGD on a single batch of data.

ANSWER

(i) Forward step

(ii) Define the loss function and compute the loss

(iii) clear the cache which had saved the gradients

(iv) Backward propogation to calculate gradients

(v) Training step- update the parameters

(b) (b) (4pt) For a single data point (x, y), write down all inputs and outputs for forward pass of each layer. You can only use variable $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{W}^{(1)}, \boldsymbol{b}^{(1)}, \boldsymbol{W}^{(2)}, \boldsymbol{b}^{(2)}$ in your answer. (note that Linear $i(\boldsymbol{x}) = \boldsymbol{W}^{(i)}\boldsymbol{x} + \boldsymbol{b}^{(i)}$).

ANSWER

	Input	Output
Linear Layer 1	$\mathbf{x}$	$\boldsymbol{s}_1 = \boldsymbol{W}^{(1)}\boldsymbol{x} + \boldsymbol{b}^{(1)}$
Activation function f	$\boldsymbol{s}_1 = \boldsymbol{W}^{(1)}\boldsymbol{x} + \boldsymbol{b}^{(1)}$	$\boldsymbol{a}_1 = f(\boldsymbol{s}_1) = 3ReLu(\boldsymbol{W}^{(1)}\boldsymbol{x} + \boldsymbol{b}^{(1)}) = 3(\boldsymbol{W}^{(1)}\boldsymbol{x} + \boldsymbol{b}^{(1)})^+$
Linear Layer 2	$\boldsymbol{a}_1 = 3(\boldsymbol{W}^{(1)}\boldsymbol{x} + \boldsymbol{b}^{(1)})^+$	$\boldsymbol{s}_2 = \boldsymbol{W}^{(2)}\boldsymbol{a}_1 + \boldsymbol{b}^{(2)} = 3\boldsymbol{W}^{(2)}(\boldsymbol{W}^{(1)}\boldsymbol{x} + \boldsymbol{b}^{(1)})^+ + \boldsymbol{b}^{(2)}$
Activation function g	$3\boldsymbol{W}^{(2)}(\boldsymbol{W}^{(1)}\boldsymbol{x} + \boldsymbol{b}^{(1)})^+ + \boldsymbol{b}^{(2)}$	$3\boldsymbol{W}^{(2)}(\boldsymbol{W}^{(1)}\boldsymbol{x} + \boldsymbol{b}^{(1)})^+ + \boldsymbol{b}^{(2)}$
Loss Function	$3\boldsymbol{W}^{(2)}(\boldsymbol{W}^{(1)}\boldsymbol{x} + \boldsymbol{b}^{(1)})^+ + \boldsymbol{b}^{(2)}, \boldsymbol{y}$	$\|\|3\boldsymbol{W}^{(2)}(\boldsymbol{W^{(1)}x + b^{(1)}})^+ + \boldsymbol{b}^{(2)} - \boldsymbol{y}\|\|^2$

Dimensions:
$x \in \mathbb{R}^n$ and $y \in \mathbb{R}^K$
Let s_1 and $a_1 \in \mathbb{R}^b$ then $W^{(1)} \in \mathbb{R}^{b \times n}$ and $b^{(1)} \in \mathbb{R}^b$
$W^{(2)} \in \mathbb{R}^{K \times b}$ and $b^{(2)} \in \mathbb{R}^b$ and $\tilde{y} \in \mathbb{R}^{K \times K}$

(c) (6pt) Write down the gradients calculated from the backward pass. You can only use the following variables: $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{W}^{(1)}, \boldsymbol{b}^{(1)}, \boldsymbol{W}^{(2)}, \boldsymbol{b}^{(2)}, \frac{\partial C}{\partial \tilde{\boldsymbol{y}}}, \frac{\partial \boldsymbol{a_1}}{\partial \boldsymbol{s_1}}, \frac{\partial \tilde{\boldsymbol{y}}}{\partial \boldsymbol{s_2}}$ in your answer, where $\boldsymbol{s}_1, \boldsymbol{a}_1, \boldsymbol{s}_2, \tilde{\boldsymbol{y}}$ are the outputs of Linear 1, f, Linear,g.

ANSWER

$\frac{\partial s_1}{\partial W^{(1)}} = x$ and $\frac{\partial s_1}{\partial b^{(1)}} = 1$

$\frac{\partial s_2}{\partial W^{(2)}} = a_1$ and $\frac{\partial s_2}{\partial b^{(2)}} = 1$ and $\frac{\partial s_2}{\partial a_1} = W^{(2)}$

$\frac{\partial C}{\partial s_2} = \frac{\partial C}{\partial \tilde{y}} \frac{\partial \tilde{y}}{\partial s_2}$

$\frac{\partial C}{\partial s_1} = \frac{\partial C}{\partial \tilde{y}} \frac{\partial \tilde{y}}{\partial s_2} \frac{\partial s_2}{\partial a_1} \frac{\partial a_1}{\partial s_1} = \frac{\partial C}{\partial \tilde{y}} \frac{\partial \tilde{y}}{\partial s_2} W^{(2)} \frac{\partial a_1}{\partial s_1}$

$\frac{\partial \tilde{y}}{\partial s_2} = 1$ since g is identity function

$\frac{\partial C}{\partial \tilde{y}} = 2(\tilde{y} - y)$

Using the above :

$\frac{\partial C}{\partial b_1} = \frac{\partial C}{\partial s_1} \frac{\partial s_1}{\partial b_1} = \frac{\partial C}{\partial \tilde{y}} \frac{\partial \tilde{y}}{\partial s_2} W^{(2)} \frac{\partial a_1}{\partial s_1}$

$\frac{\partial C}{\partial W^{(1)}} = \frac{\partial C}{\partial s_1} x = x \frac{\partial C}{\partial \tilde{y}} \frac{\partial \tilde{y}}{\partial s_2} W^{(2)} \frac{\partial a_1}{\partial s_1}$

$\frac{\partial C}{\partial b_2} = \frac{\partial C}{\partial s_2} \frac{\partial s_2}{\partial b_2}$

$\frac{\partial C}{\partial W^{(2)}} = \frac{\partial C}{\partial s_2} \frac{\partial s_2}{\partial W^{(2)}} = \frac{\partial C}{\partial \tilde{y}} \frac{\partial \tilde{y}}{\partial s_2} a_1 = 3(W^{(1)}x + b^{(1)})^+ \frac{\partial C}{\partial \tilde{y}} \frac{\partial \tilde{y}}{\partial s_2}$

(d) (2pt) Show us the elements of $\frac{\partial a_1}{\partial s_1}$, $\frac{\partial \tilde{y}}{\partial s_2}$ and $\frac{\partial C}{\partial \tilde{y}}$ (be careful about the dimensionality)?

$(\frac{\partial C}{\partial \tilde{y}})$ is a row vector

$(\frac{\partial C}{\partial \tilde{y}})_i = 2(\tilde{y} - y)_i$

$(\frac{\partial C}{\partial \tilde{y}}) = \begin{bmatrix} 2(\tilde{y}_1 - y_1) & .. & .. & 2(\tilde{y}_n - y_n) \end{bmatrix} \in \mathbb{R}^{1 \times n}$

$(\frac{\partial \tilde{y}}{\partial s_2})$ is a identity matrix since the second activation function g is identity function.

$(\frac{\partial \tilde{y}}{\partial s_2}) = \begin{bmatrix} 1 & 0 & 0 & .. & .. \\ 0 & 1 & .. & .. & .. \\ .. & .. & .. & .. & .. \\ .. & .. & .. & .. & 1 \end{bmatrix} \in \mathbb{R}^{K \times K}$

$\frac{\partial a_1}{\partial s_1}$ is a matrix where diagonal elements are 3 and all other elements are 0

1.3 Classification Task

We would like to perform multi-class classification task, so we set $f = \tanh$ and $g = \sigma$, the logistic sigmoid function $\sigma(x) \doteq (1 + \exp(-x))^{-1}$.

(a) (4pt + 6pt + 2pt) If you want to train this network, what do you need to change in the equations of (b), (c) and (d), assuming we are using the same squared Euclidean distance loss function.

	Input	Output
Linear Layer 1	$\boldsymbol{x}$	$\boldsymbol{s}_1 = \boldsymbol{W}^{(1)}\boldsymbol{x} + \boldsymbol{b}^{(1)}$
Activation function f	$\boldsymbol{s}_1 = \boldsymbol{W}^{(1)}\boldsymbol{x} + \boldsymbol{b}^{(1)}$	$\boldsymbol{a}_1 = f(\boldsymbol{s}_1) = tanh(\boldsymbol{W}^{(1)}\boldsymbol{x} + \boldsymbol{b}^{(1)})$
Linear Layer 2	$\boldsymbol{a}_1 = tanh(\boldsymbol{W}^{(1)}x + \boldsymbol{b}^{(1)})$	$\boldsymbol{s}_2 = \boldsymbol{W}^{(2)}\boldsymbol{a}_1 + \boldsymbol{b}^{(2)} = \boldsymbol{W}^{(2)}tanh(\boldsymbol{W}^{(1)}\boldsymbol{x} + \boldsymbol{b}^{(1)}) + \boldsymbol{b}^{(2)}$
Activation function g	$\boldsymbol{W}^{(2)}tanh(\boldsymbol{W}^{(1)}\boldsymbol{x} + \boldsymbol{b}^{(1)}) + \boldsymbol{b}^{(2)}$	$\sigma(\boldsymbol{W}^{(2)}tanh(\boldsymbol{W}^{(1)}\boldsymbol{x} + \boldsymbol{b}^{(1)}) + \boldsymbol{b}^{(2)})$
Loss Function	$\sigma(\boldsymbol{W}^{(2)}tanh(\boldsymbol{W}^{(1)}\boldsymbol{x} + \boldsymbol{b}^{(1)}) + \boldsymbol{b}^{(2)}), \boldsymbol{y}$	$\|\|\sigma(\boldsymbol{W}^{(2)}tanh(\boldsymbol{W}^{(1)}\boldsymbol{x} + \boldsymbol{b}^{(1)}) + \boldsymbol{b}^{(2)}) - \boldsymbol{y}\|\|^2$

b) Gradients are:

$\frac{\partial C}{\partial b_1} = \frac{\partial C}{\partial s_1}\frac{\partial s_1}{\partial b_1} = \frac{\partial C}{\partial \tilde{y}}\frac{\partial \tilde{y}}{\partial s_2}W^{(2)}\frac{\partial a_1}{\partial s_1}$

$\frac{\partial C}{\partial W^{(1)}} = \frac{\partial C}{\partial s_1}x = x\frac{\partial C}{\partial \tilde{y}}\frac{\partial \tilde{y}}{\partial s_2}W^{(2)}\frac{\partial a_1}{\partial s_1}$

$\frac{\partial C}{\partial b_2} = \frac{\partial C}{\partial s_2}\frac{\partial s_2}{\partial b_2} = \frac{\partial C}{\partial \tilde{y}}\frac{\partial \tilde{y}}{\partial s_2}$

Change in the following gradient:

$\frac{\partial C}{\partial W^{(2)}} = \frac{\partial C}{\partial s_2}\frac{\partial s_2}{\partial W^{(2)}} = \frac{\partial C}{\partial \tilde{y}}\frac{\partial \tilde{y}}{\partial s_2}a_1 = tanh(W^{(1)} + b^{(1)})\frac{\partial C}{\partial \tilde{y}}\frac{\partial \tilde{y}}{\partial s_2}$

$\frac{\partial \tilde{y}}{\partial s_2}$ also changes since g is now the sigmoid function

$\frac{\partial a_1}{\partial s_1}$ also changes since f is now the tanh function.
Derivative of $tanh(x) = (1 - (tanh(x))^2)$

c) Elements are:

$(\frac{\partial C}{\partial \tilde{y}})$ is a row vector

$(\frac{\partial C}{\partial \tilde{y}})_i = 2(\tilde{y} - y)_i$

$(\frac{\partial C}{\partial \tilde{y}}) = \begin{bmatrix} 2(\tilde{y}_1 - y_1) & .. & .. & 2(\tilde{y}_n - y_n) \end{bmatrix} \in \mathbb{R}^{1\times n}$

Diagonal elements of $\frac{\partial \tilde{y}}{\partial s_2}$ are:

$(\frac{\partial \tilde{y}}{\partial s_2})_{ii} = \sigma((s_2)_i)(1 - \sigma((s_2)_i))$ and other elements are 0

$$(\frac{\partial \tilde{y}}{\partial s_2}) = \begin{bmatrix} \sigma((s_2)_1)(1-\sigma((s_2)_1) & 0 & 0 & .. & .. \\ 0 & \sigma((s_2)_2)(1-\sigma((s_2)_2) & .. & .. & .. \\ .. & .. & .. & .. & .. \\ .. & .. & .. & .. & \sigma((s_2)_K)(1-\sigma((s_2)_K) \end{bmatrix}$$

$\frac{\partial \tilde{y}}{\partial s_2} \in \mathbb{R}^{K\times K}$

Diagonal elements of $\frac{\partial a_1}{\partial s_1} \in \mathbb{R}^{b\times b}$ are:

$(\frac{\partial a_1}{\partial s_1})_{ii} = (1 - tanh(((s_1)^2)_i)$ and other elements are 0

$$(\frac{\partial a_1}{\partial s_1}) = \begin{bmatrix} 1-(tanh(s_1)_1)^2 & 0 & 0 & .. & .. \\ 0 & 1-(tanh(s_1)_2)^2 & .. & .. & .. \\ .. & .. & .. & .. & .. \\ .. & .. & .. & .. & 1-(tanh(s_1)_b)^2 \end{bmatrix}$$

(b) (4pt+6pt+2pt) Now you think you can do a better job by using a *Bi− nary Cross Entropy* (BCE) loss function $D_{\text{BCE}}(\boldsymbol{y}, \tilde{\boldsymbol{y}}) = \frac{1}{K}\sum_{i=1}^{K} - [y_i \log(\tilde{y}_i) + (1 - y_i)\log(1 - \tilde{y}_i)]$. What do you need to change in the equations of (b), (c) and (d) ?

a) Input output of different layer

	Input	Output
Linear Layer 1	$\boldsymbol{x}$	$\boldsymbol{s}_1 = \boldsymbol{W}^{(1)}\boldsymbol{x} + \boldsymbol{b}^{(1)}$
Activation function f	$\boldsymbol{s}_1 = \boldsymbol{W}^{(1)}\boldsymbol{x} + \boldsymbol{b}^{(1)}$	$\boldsymbol{a}_1 = f(\boldsymbol{s}_1) = tanh(\boldsymbol{W}^{(1)}\boldsymbol{x} + \boldsymbol{b}^{(1)})$
Linear Layer 2	$\boldsymbol{a}_1 = tanh(\boldsymbol{W}^{(1)}x + \boldsymbol{b}^{(1)})$	$\boldsymbol{s}_2 = \boldsymbol{W}^{(2)}\boldsymbol{a}_1 + \boldsymbol{b}^{(2)} = \boldsymbol{W}^{(2)}tanh(\boldsymbol{W}^{(1)}\boldsymbol{x} + \boldsymbol{b}^{(1)}) + \boldsymbol{b}^{(2)}$
Activation function g	$\boldsymbol{W}^{(2)}tanh(\boldsymbol{W}^{(1)}\boldsymbol{x} + \boldsymbol{b}^{(1)}) + \boldsymbol{b}^{(2)}$	$\sigma(\boldsymbol{W}^{(2)}tanh(\boldsymbol{W}^{(1)}\boldsymbol{x} + \boldsymbol{b}^{(1)}) + \boldsymbol{b}^{(2)})$
Loss Function	$\sigma(\boldsymbol{W}^{(2)}tanh(\boldsymbol{W}^{(1)}\boldsymbol{x} + \boldsymbol{b}^{(1)}) + \boldsymbol{b}^{(2)}), \boldsymbol{y}$	$\frac{-1}{K}[y^T \log\sigma(\boldsymbol{W}^{(2)}tanh(\boldsymbol{W}^{(1)}\boldsymbol{x} + \boldsymbol{b}^{(1)}) + \boldsymbol{b}^{(2)}) + (1-\boldsymbol{y})^T \log\sigma(\boldsymbol{W}^{(2)}tanh(\boldsymbol{W}^{(1)}\boldsymbol{x} + \boldsymbol{b}^{(1)}) + \boldsymbol{b}^{(2)}]$

b) Gradients are:

$\frac{\partial C}{\partial b_1} = \frac{\partial C}{\partial s_1}\frac{\partial s_1}{\partial b_1} = \frac{\partial C}{\partial \tilde{y}}\frac{\partial \tilde{y}}{\partial s_2}W^{(2)}\frac{\partial a_1}{\partial s_1}$

$\frac{\partial C}{\partial W^{(1)}} = \frac{\partial C}{\partial s_1}x = x\frac{\partial C}{\partial \tilde{y}}\frac{\partial \tilde{y}}{\partial s_2}W^{(2)}\frac{\partial a_1}{\partial s_1}$

$\frac{\partial C}{\partial b_2} = \frac{\partial C}{\partial s_2}\frac{\partial s_2}{\partial b_2} = \frac{\partial C}{\partial \tilde{y}}\frac{\partial \tilde{y}}{\partial s_2}$

Change in the following gradient:
$\frac{\partial C}{\partial W^{(2)}} = \frac{\partial C}{\partial s_2}\frac{\partial s_2}{\partial W^{(2)}} = \frac{\partial C}{\partial \tilde{y}}\frac{\partial \tilde{y}}{\partial s_2}a_1 = tanh(W^{(1)} + b^{(1)})\frac{\partial C}{\partial \tilde{y}}\frac{\partial \tilde{y}}{\partial s_2}$

$\frac{\partial C}{\partial \tilde{y}}$ also changes as the BCE loss function is used.

c)Show elements:

$\frac{\partial C}{\partial \tilde{y}}$ is a row vector $\in \mathbb{R}^{1\times K}$

$(\frac{\partial C}{\partial \tilde{y}})_i = \frac{1}{K}(\frac{y_i - \tilde{y}_i}{\tilde{y}_i(\tilde{y}_i - 1)})$

$\frac{\partial a_1}{\partial s_1}$ is a matrix where the non diagonal elements are 0 and $\in \mathbb{R}^{b\times b}$

Diagonal elements are:
$(\frac{\partial a_1}{\partial s_1})_{ii} = (1 - tanh(s_1)^2{}_i)$

$$(\frac{\partial a_1}{\partial s_1}) = \begin{bmatrix} 1-(tanh(s_1)_1)^2 & 0 & 0 & .. & .. \\ 0 & 1-(tanh(s_1)_2)^2 & .. & .. & .. \\ .. & .. & .. & .. & .. \\ .. & .. & .. & .. & 1-(tanh(s_1)_b)^2 \end{bmatrix}$$

$\frac{\partial \tilde{y}}{\partial s_2}$ is a matrix where the non diagonal elements are 0 and $\in \mathbb{R}^{K\times K}$

Diagonal elements are:
$(\frac{\partial \tilde{y}}{\partial s_2})_{ii} = \sigma((s_2)_i)(1 - \sigma((s_2)_i)$

(c) (1pt) Things are getting better. You realize that not all intermediate hidden activations need to be binary (or soft version of binary). You decide to use $f(\cdot) = (\cdot)^+$but keep g as tanh. Explain why this choice of f can be beneficial for training a (deeper) network.

ANSWER
ReLu is a much simpler function as compared to sigmoid and would take less time to compute in the forward and backward step.
ReLu also has no vanishing gradient as its derivative is 1 for values >0 and 0 otherwise. The gradient can't vanish and can be transferred across the fat network.

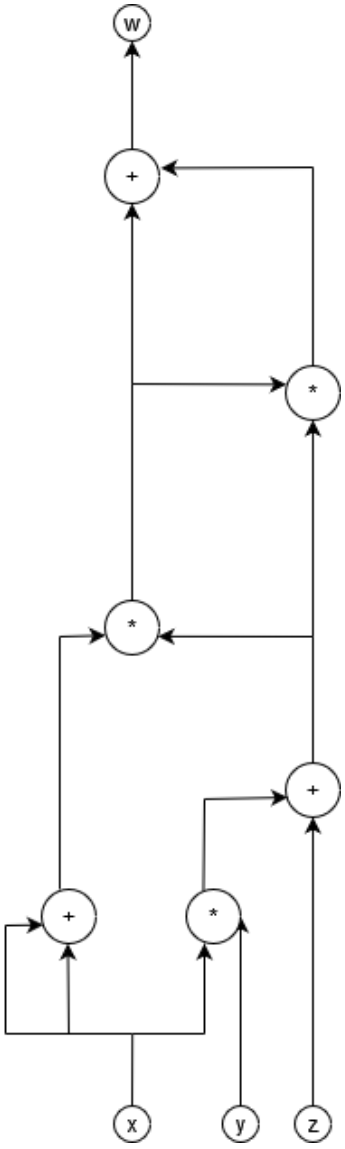
1.4 Conceptual Questions

(a) (1pt) Why is softmax actually softargmax?

ANSWER
Softmax gives the smooth appoximation to the argmax function.

(b) (3pt) Draw the computational graph defined by this function, with inputs $x, y, z \in \mathbb{R}$ and output $w \in \mathbb{R}$. You make use symbols x, y, z, o, and operators $*, +$ in your solution. Be sure to use the correct shape for symbols and operators as shown in class.

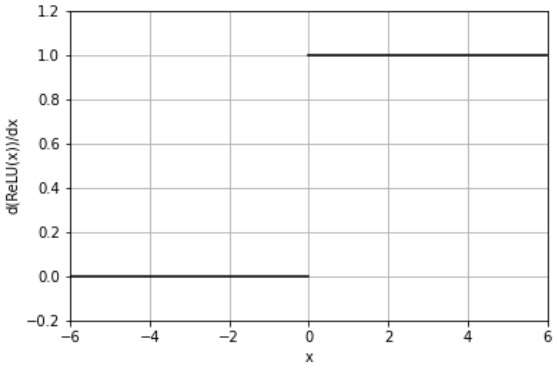
$$a = xy + z$$
$$b = a(x + x)$$
$$w = ab + b$$


w
+
*
*
+
+
*
x
y
z

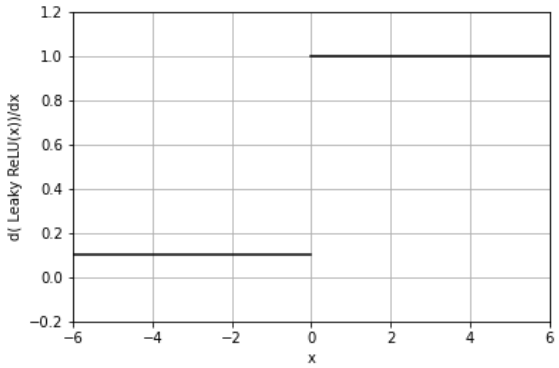
(c) (2pt) Draw the graph of the derivative for the following functions?

- ReLU()
- LeakyReLU (negative_slope=0.1)
- Softplus (beta=1)
- GELU()

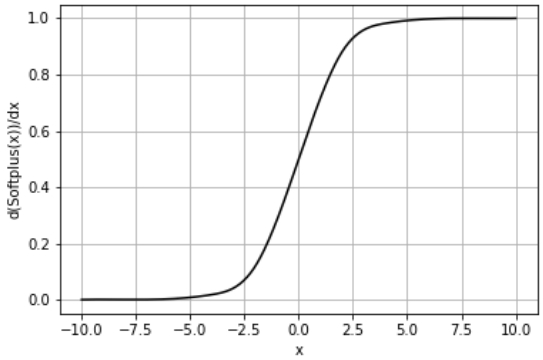
Derivative of ReLU()



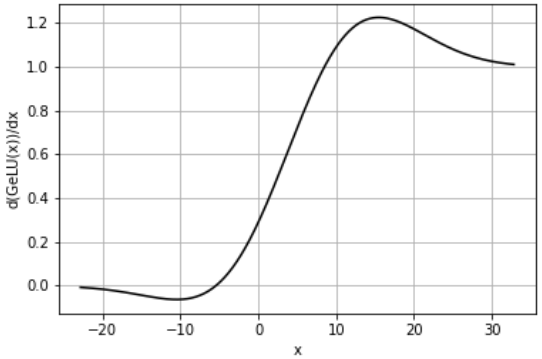
Derivative of Leaky ReLU()



Derivative of Softplus()



Derivative of GeLU()



(d) (3pt) Given function $f(x) = \boldsymbol{W}_1\boldsymbol{x}$ with $\boldsymbol{W}_1 \in \mathbb{R}^{b\times a}$ and $g(\boldsymbol{x}) = \boldsymbol{W}_2\boldsymbol{x}$ with $W_2 \in \mathbb{R}^{b\times a}$

(a) What is the Jacobian matrix of f and g

Jacobian matrix $f = \boldsymbol{W}_1 \in \mathbb{R}^{b\times a}$
Jacobian matrix $g = \boldsymbol{W}_2 \in \mathbb{R}^{b\times a}$

(b) What is the Jacobian matrix of $h(x) = f(x) + g(x)$

$\boldsymbol{W}_1 + \boldsymbol{W}_2 \in \mathbb{R}^{b\times a}$

(c) What is the Jacobian matrix of $h(\boldsymbol{x}) = f(\boldsymbol{x}) + g(\boldsymbol{x})$ if $\boldsymbol{W}_1 = \boldsymbol{W}_2$

$2\boldsymbol{W}_1$ or $2\boldsymbol{W}_2 \in \mathbb{R}^{b\times a}$

(e) (3pt) Given function $f(\boldsymbol{x}) = \boldsymbol{W}_1\boldsymbol{x}$ with $\boldsymbol{W}_1 \in \mathbb{R}^{b\times a}$ and $g(\boldsymbol{x}) = \boldsymbol{W}_2\boldsymbol{x}$ with $\boldsymbol{W}_2 \in \mathbb{R}^{c\times b}$

(a) What is the Jacobian matrix of f and g

Jacobian of matrix $f = \boldsymbol{W}_1 \in \mathbb{R}^{b\times a}$
Jacobian of matrix $g = \boldsymbol{W}_2 \in \mathbb{R}^{c\times b}$

(b) What is the Jacobian matrix of $h(\boldsymbol{x}) = g(f(\boldsymbol{x})) = (g \circ f)(\boldsymbol{x})$

$g(f(\boldsymbol{x})) = g(\boldsymbol{W}_1\boldsymbol{x}) = \boldsymbol{W}_2(\boldsymbol{W}_1\boldsymbol{x}) \in \mathbb{R}^{c\times 1}$
Jacobian is $\boldsymbol{W}_2\boldsymbol{W}_1 \in \mathbb{R}^{c\times a}$

(c) What is the Jacobian matrix of $h(\boldsymbol{x})$ if $\boldsymbol{W}_1 = \boldsymbol{W}_2$ (so $a = b = c$)

$J(h(\boldsymbol{x}))_{ij} = (\boldsymbol{W}_1)_{ij} \times (\boldsymbol{W}_1)_{ij}$