Homework 1

Guilherme Albertini

September 28, 2022

Theory

Let $Linear_1 \to f \to Linear_2 \to g$ be a be a two-layer neural net architecture whereby $Linear_i(x) = \boldsymbol{W}^{(i)}\boldsymbol{x} + \boldsymbol{b}^{(i)}$ is the i^{th} affine transformation and f, g are element-wise nonlinear activation functions (else must use transposes and/or Hadamard products). When an input $\boldsymbol{x} \in \mathbb{R}^n$ is fed into the network, $\hat{\boldsymbol{y}} \in \mathbb{R}^K$ is obtained as output.

Problem 1: Regression Task

We would like to perform regression task. We choose $f(\cdot) = 5(\cdot)^+ = 5ReLU(\cdot)$ and g to be the identity function. To train, we choose MSE loss function, $\ell_{MSE}(\hat{\boldsymbol{y}}, \boldsymbol{y}) = ||(\hat{\boldsymbol{y}} - \boldsymbol{y})||^2$.

1. 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.

   (a) First compute a prediction from the model (the forward pass); $\tilde{y} = model(x)$

   (b) Second compute the loss through computation of the energy

   (c) Zero the gradient parameters; $optimiser.zero\_grad()$

   (d) Compute and accumulate gradient parameters; $L.backward()$

   (e) Finally step in the opposite direction of the gradient; $optimiser.step()$

2. For a single data point (x, y), write down all inputs and outputs for forward pass of each layer. You can only use variables and mechanics specified prior in your answer.

Layer	Input	Output
$Linear_1$	$\boldsymbol{x}$	$\boldsymbol{z_1} = \boldsymbol{W}^{(1)}\boldsymbol{x} + \boldsymbol{b}^{(1)}$
f	$\boldsymbol{z_1}$	$\boldsymbol{z_2} = 5(\boldsymbol{W}^{(1)}\boldsymbol{x} + \boldsymbol{b}^{(1)})^+$
$Linear_2$	$\boldsymbol{z_2}$	$\boldsymbol{z_3} = 5\boldsymbol{W}^{(2)}(\boldsymbol{W}^{(1)}\boldsymbol{x} + \boldsymbol{b}^{(1)})^+ + \boldsymbol{b}^{(2)}$
g	$\boldsymbol{z_3}$	$\boldsymbol{z_3}$
$Loss$	$\boldsymbol{z_3}, \boldsymbol{y}$	$\|\|\hat{\boldsymbol{y}} - \boldsymbol{y}\|\|^2 = \|\|5\boldsymbol{W}^{(2)}(\boldsymbol{W}^{(1)}\boldsymbol{x} + \boldsymbol{b}^{(1)})^+ + \boldsymbol{b}^{(2)} - \boldsymbol{y}\|\|^2$

3. Write down the gradients calculated from the backward pass. You can only use the following variables: $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{W^{(1)}}, \boldsymbol{b^{(1)}}, \boldsymbol{W^{(2)}}, \boldsymbol{b^{(2)}}, \frac{\partial \ell}{\partial \hat{\boldsymbol{y}}}, \frac{\partial \boldsymbol{z_2}}{\partial \boldsymbol{z_1}}, \frac{\partial \hat{\boldsymbol{y}}}{\partial \boldsymbol{z_3}}$, where $\boldsymbol{z_1}, \boldsymbol{z_2}, \boldsymbol{z_3}, \hat{\boldsymbol{y}}$ are outputs of $Linear_1, f, Linear_2, g$, respectively.

Dimensions shown are used throughout ($z_1, z_2 \in \mathbb{R}^h$):

$$
\begin{aligned}
x &\in \mathbb{R}^n \\
\hat{y} &\in \mathbb{R}^K \\
\ell &\in \mathbb{R} \\
W^{(1)} &\in \mathbb{R}^{h \times n} \\
b^{(1)} &\in \mathbb{R}^h \\
W^{(2)} &\in \mathbb{R}^{K \times h} \\
b^{(2)} &\in \mathbb{R}^K \\
\frac{\partial \ell}{\partial W^{(1)}} &\in \mathbb{R}^{n \times h} \\
\frac{\partial \ell}{\partial b^{(1)}} &\in \mathbb{R}^{1 \times h} \\
\frac{\partial \ell}{\partial W^{(2)}} &\in \mathbb{R}^{h \times K} \\
\frac{\partial \ell}{\partial b^{(2)}} &\in \mathbb{R}^{1 \times K} \\
\frac{\partial \ell}{\partial \hat{y}} &\in \mathbb{R}^{1 \times K} \\
\frac{\partial \hat{y}}{\partial z_3} &\in \mathbb{R}^{K \times K} \\
\frac{\partial z_3}{\partial z_2} &\in \mathbb{R}^{K \times h} \\
\frac{\partial z_2}{\partial z_1} &\in \mathbb{R}^{h \times h} \\
\frac{\partial z_1}{\partial b_1} &\in \mathbb{R}^{h \times h} \\
\frac{\partial z_1}{\partial W^{(1)}} &\in \mathbb{R}^{h \times (h \times n)} \\
\frac{\partial z_3}{\partial b^{(2)}} &\in \mathbb{R}^{K \times K} \\
\frac{\partial z_3}{\partial b^{(2)}} &\in \mathbb{R}^{K \times K} \\
\frac{\partial z_3}{\partial W^{(2)}} &\in \mathbb{R}^{K \times (K \times h)}
\end{aligned}
$$

$$
\begin{aligned}
\frac{\partial \ell}{\partial z_3} &= \frac{\partial \ell}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial z_3} \\
\frac{\partial \ell}{\partial z_1} &= \frac{\partial \ell}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial z_3} \frac{\partial z_3}{\partial z_2} \frac{\partial z_2}{\partial z_1} = \frac{\partial \ell}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial z_3} W^{(2)} \frac{\partial z_2}{\partial z_1} \\
\frac{\partial \boldsymbol{\ell}}{\partial \boldsymbol{b^{(2)}}} &= \frac{\partial \boldsymbol{\ell}}{\partial \hat{\boldsymbol{y}}} \frac{\partial \hat{\boldsymbol{y}}}{\partial \boldsymbol{z_3}} \frac{\partial \boldsymbol{z_3}}{\partial \boldsymbol{b^{(2)}}} \\
&= \frac{\partial \boldsymbol{\ell}}{\partial \hat{\boldsymbol{y}}} \frac{\partial \hat{\boldsymbol{y}}}{\partial \boldsymbol{z_3}} = \frac{\partial \ell}{\partial z_3} \\
\frac{\partial \ell}{\partial b^{(1)}} &= \frac{\partial \ell}{\partial z_1} \frac{\partial z_1}{\partial b^{(1)}} = \frac{\partial \ell}{\partial z_1} \\
\frac{\partial \ell}{\partial W^{(1)}} = \frac{\partial \ell}{\partial z_1} \frac{\partial z_1}{\partial W^{(1)}} &= \sum_i \frac{\partial \ell}{\partial (z_1)_i} \frac{\partial (z_1)_i}{\partial W^{(1)}} = \frac{\partial \ell}{\partial z_1} x^T = x \frac{\partial \ell}{\partial z_1} \\
&= x \frac{\partial \ell}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial z_3} W^{(2)} \frac{\partial z_2}{\partial z_1} \\
\frac{\partial \ell}{\partial W^{(2)}} = \frac{\partial \ell}{\partial z_3} \frac{\partial z_3}{\partial W^{(2)}} &= \sum_i \frac{\partial \ell}{\partial (z_3)_i} \frac{\partial (z_3)_i}{\partial W^{(2)}} = \frac{\partial \ell}{\partial z_3} {z_2}^T = z_2 \frac{\partial \ell}{\partial z_3} \\
&= 5(\boldsymbol{W}^{(1)} \boldsymbol{x} + \boldsymbol{b}^{(1)})^+ \frac{\partial \ell}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial z_3}
\end{aligned}
$$

4. Show the elements of $\frac{\partial \boldsymbol{z_2}}{\partial \boldsymbol{z_1}}$, $\frac{\partial \hat{\boldsymbol{y}}}{\partial \boldsymbol{z_3}}$, $\frac{\partial \ell}{\partial \hat{\boldsymbol{y}}}$. Be careful about dimensionality.

$$
\left(\frac{\partial \boldsymbol{z_2}}{\partial \boldsymbol{z_1}} \right)_{ii} = \begin{cases} 0, & z_{1i} < 0 \\ 5, & z_{1i} > 0 \\ \text{undefined (or assigned a value 0)}, & z_{1i} = 0 \end{cases}
$$

Note: the above is a diagonal matrix

$$
\left(\frac{\partial \hat{\boldsymbol{y}}}{\partial \boldsymbol{z_3}} \right)_{ii} = 1 \text{ (and 0 elsewhere, off of diagonal; an identity matrix)}
$$

$$
\frac{\partial \ell}{\partial \hat{\boldsymbol{y}}} = \frac{\partial (||\hat{\boldsymbol{y}} - \boldsymbol{y}||^2)}{\partial \hat{\boldsymbol{y}}} = 2(\hat{\boldsymbol{y}} - \boldsymbol{y})^{\boldsymbol{T}} \text{ (A vector)}
$$

$$
\implies 2(\hat{y} - y)_i \text{ are the elements}
$$

Layer	Input	Output
$Linear_1$	$\boldsymbol{x}$	$\boldsymbol{z_1} = \boldsymbol{W}^{(1)}\boldsymbol{x} + \boldsymbol{b}^{(1)}$
f	$\boldsymbol{z_1}$	$\boldsymbol{z_2} = \tanh(\boldsymbol{W}^{(1)}\boldsymbol{x} + \boldsymbol{b}^{(1)})$
$Linear_2$	$\boldsymbol{z_2}$	$\boldsymbol{z_3} = \boldsymbol{W}^{(2)}\tanh(\boldsymbol{W}^{(1)}\boldsymbol{x} + \boldsymbol{b}^{(1)}) + \boldsymbol{b}^{(2)}$
g	$\boldsymbol{z_3}$	$\sigma(z_3) = \frac{\mathbf{1}}{\mathbf{1+\exp(-z_3)}}$
$Loss$	$\hat{\boldsymbol{y}} = \boldsymbol{\sigma(z_3)}, \boldsymbol{y}$	$\|\|\hat{\boldsymbol{y}} - \boldsymbol{y}\|\|^2 = \|\|\boldsymbol{\sigma(z_3)} - \boldsymbol{y}\|\|^2$

Problem 2: Classification Task

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

1. If you want to train this network, what do you need to change in the equations of (1.2), (1.3) and (1.4), assuming we are using the same MSE loss function. Consult updated chart above.

$$\frac{\partial \ell}{\partial z_3} = \frac{\partial \ell}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial z_3}$$

$$\frac{\partial \ell}{\partial z_1} = \frac{\partial \ell}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial z_3} \frac{\partial z_3}{\partial z_2} \frac{\partial z_2}{\partial z_1} = \frac{\partial \ell}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial z_3} W^{(2)} \frac{\partial z_2}{\partial z_1}$$

$$\frac{\partial \boldsymbol{\ell}}{\partial \boldsymbol{b^{(2)}}} = \frac{\partial \boldsymbol{\ell}}{\partial \boldsymbol{\hat{y}}} \frac{\partial \boldsymbol{\hat{y}}}{\partial \boldsymbol{z_3}} \frac{\partial \boldsymbol{z_3}}{\partial \boldsymbol{b^{(2)}}}$$

$$= \frac{\partial \boldsymbol{\ell}}{\partial \boldsymbol{\hat{y}}} \frac{\partial \boldsymbol{\hat{y}}}{\partial \boldsymbol{z_3}} = \frac{\partial \ell}{\partial z_3}$$

$$\frac{\partial \ell}{\partial b^{(1)}} = \frac{\partial \ell}{\partial z_1} \frac{\partial z_1}{\partial b^{(1)}} = \frac{\partial \ell}{\partial z_1}$$

$$\frac{\partial \ell}{\partial W^{(1)}} = \frac{\partial \ell}{\partial z_1} \frac{\partial z_1}{\partial W^{(1)}} = \sum_i \frac{\partial \ell}{\partial (z_1)_i} \frac{\partial (z_1)_i}{\partial W^{(1)}} = \frac{\partial \ell}{\partial z_1} x^T = x \frac{\partial \ell}{\partial z_1}$$

$$= x \frac{\partial \ell}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial z_3} W^{(2)} \frac{\partial z_2}{\partial z_1}$$

$$\frac{\partial \ell}{\partial W^{(2)}} = \frac{\partial \ell}{\partial z_3} \frac{\partial z_3}{\partial W^{(2)}} = \sum_i \frac{\partial \ell}{\partial (z_3)_i} \frac{\partial (z_3)_i}{\partial W^{(2)}} = \frac{\partial \ell}{\partial z_3} {z_2}^T = z_2 \frac{\partial \ell}{\partial z_3}$$

$$= \tanh(\boldsymbol{W}^{(1)} \boldsymbol{x} + \boldsymbol{b}^{(1)}) \frac{\partial \ell}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial z_3}$$

$\left(\frac{\partial z_2}{\partial z_1}\right)_{ii} = [(\text{sech}(\boldsymbol{W}^{(1)} \boldsymbol{x} + \boldsymbol{b}^{(1)}))]^2_{1i}$ and off diagonal elements are 0

$\left(\frac{\partial \hat{y}}{\partial z_3}\right)_{ii} = \sigma((z_3)_i)(1 - \sigma(z_3)_i)$ and off diagonal elements are 0

$$\frac{\partial \ell}{\partial \hat{\boldsymbol{y}}} = \frac{\partial (||\hat{\boldsymbol{y}} - \boldsymbol{y}||^2)}{\partial \hat{\boldsymbol{y}}} = 2(\hat{\boldsymbol{y}} - \boldsymbol{y})^{\boldsymbol{T}}$$

$\implies 2(\hat{y} - y)_i$ are the elements

2. Now you think you can do a better job by using a Binary Cross Entropy (BCE) loss function $\ell_{BCE}(\hat{\boldsymbol{y}}, \boldsymbol{y}) = \frac{1}{K} \sum_{i=1}^{K} -[y_i \log(\hat{y_i}) + (1 - y_i) \log(1 - \hat{\hat{y}_i})]$. What needs to change from the previous?

—Loss—
Input: $\hat{\boldsymbol{y}}$
Output: $\ell_{BCE}(\hat{\boldsymbol{y}}, \boldsymbol{y}) = \frac{1}{K} \sum_{i=1}^{K} -[y_i \log(\hat{y_i}) + (1 - y_i) \log(1 - \hat{y_i})]$,
$\hat{y_i} = \frac{1}{1+\exp(-z_{3i})}$
$\frac{\partial \ell}{\partial \hat{y}} = \frac{1}{K}(\frac{-y}{\hat{y}} + \frac{1-y}{1-\hat{y}})^T \implies \frac{1}{K} \frac{\hat{y_i} - y_i}{\hat{y_i}(1-\hat{y_i})}$ are elements

3. 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 σ. Explain why this choice of f can be beneficial for training a (deeper) network.

> Sigmoid function is more computationally intensive to compute compared to ReLU due to exponential operation; the latter produces sparsity in matrices which encourages numerical optimization techniques taking advantage of this time and space complexity reduction.

> ReLU also avoids the vanishing gradient problem, whereby the number of parameters receive very small updates such that the nodes deviate greatly from their optimal value. As the gradient is constant for ReLU compared to the sigmoid gradient always being smaller than 1, successive operations will start to prohibit learning.

Problem 3: Conceptual Questions

1. Why is softmax actually softargmax?

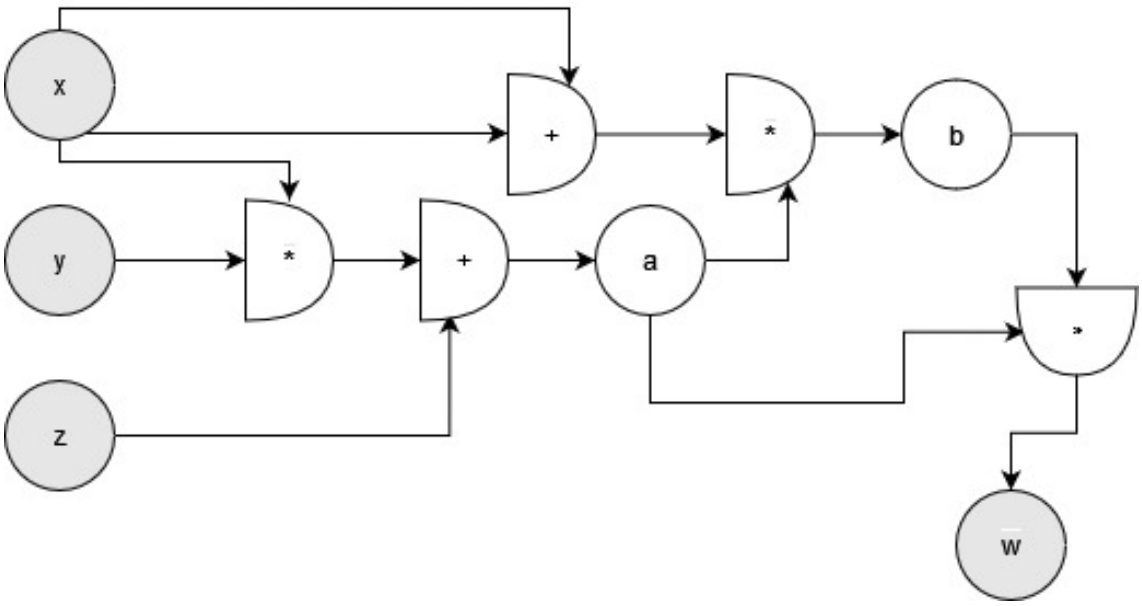
> The "softmax" is not a smooth approximation to the maximum function – it is actually an approximation to the arg max function whose value is the index that has the maximum. Accordingly, some prefer the "softargmax" terminolgy to emphasize this distinction.

2. 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, w, and operators $+, \star$ in your solution. Be sure to use the correct shape for symbols and operators as shown in class.

   (a) $a = x \star y + z$

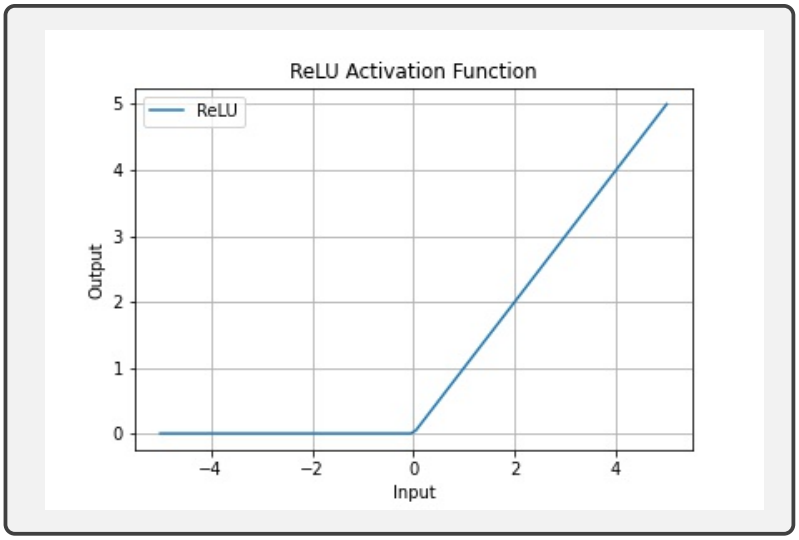
   (b) $b = (x + x) \star a$

   (c) $w = a \star b$



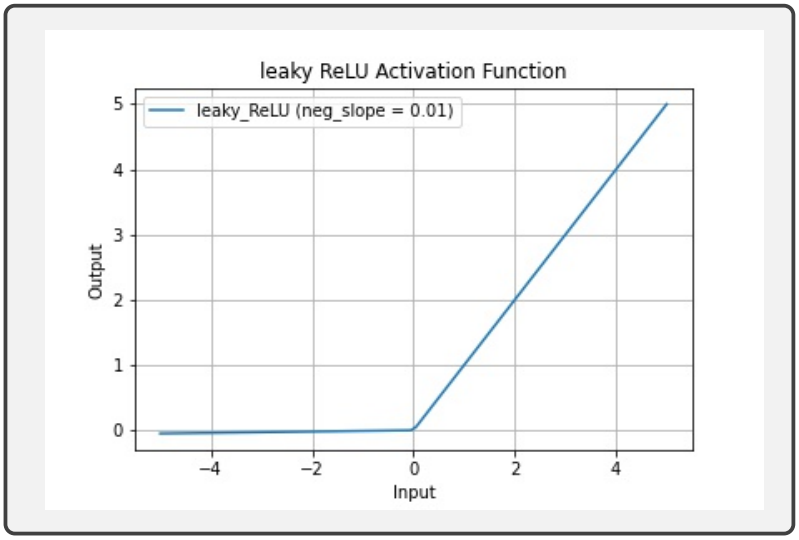
3. Draw the graph of the following:

   (a) ReLU()



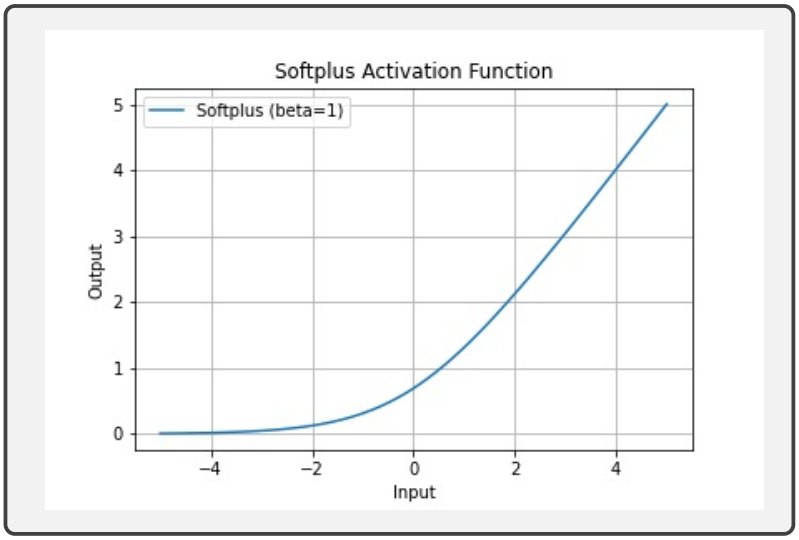

```
def ReLU(x):
  data = [max(0, value) for value in x]
  return np.array(data, dtype=float)
```

   (b) LeakyReLU(neg slope is 0.01)



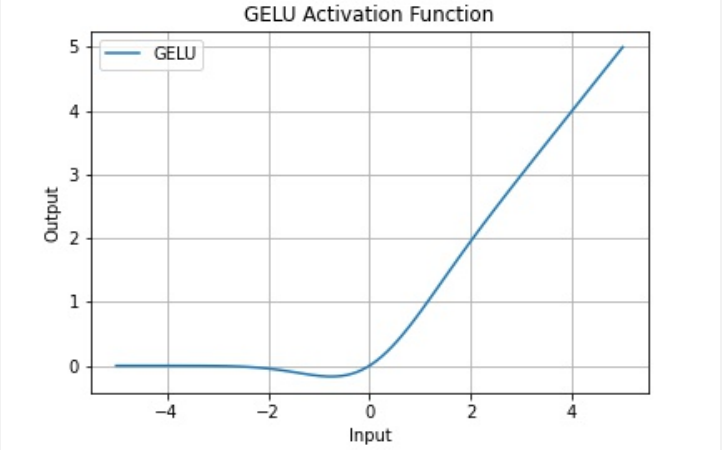

```
def leaky_ReLU(x, slope):
    data = [max(slope*value, value) for value in x]
    return np.array(data, dtype=float)
```

(c) Softplus(beta is 1)




```
def softplus(x, beta):
    # log(1+exp(B*x)) = log(1+exp(B*x)) - log(exp(B*x)) +
    # B*x = log(1+exp(-B*x)) + B*x
    data = [(math.log(1+math.exp(-abs(value*beta))) +
             max(value*beta, 0))/beta for value in x]
    return np.array(data, dtype=float)
```

(d) GELU




```
from math import erf
def GELU(x):
    data = [0.5 * value * (1.0 + erf(value / np.sqrt(2.0))) for value in x]
    return np.array(data, dtype=float)
```

4. What are 4 different types of linear transformations? What is the role of linear transformation and non linear transformation in a neural network?

> Reflection, rotation, scaling, and shear are the main types of linear transformations. In neural networks, each output unit produces the linear combination of the inputs and the connection weights. Once we allow translation, these essentially become affine transformations. To introduce nonlinearity to the model (to capture greater complexity by having a larger hypothesis space), activation functions ingest these transformations through a nonlinear function and treat that as the unit output.

5. Given a neural network F parameterized by parameters θ, denoted F_θ, dataset $D = x_1, x_2, \ldots, x_N$, and labels $Y = y_1, y_2, \ldots, y_N$, write down the mathematical definition of training a neural network with the MSE loss function $\ell_{MSE}(\hat{\boldsymbol{y}}, \boldsymbol{y}) = ||\hat{y} - y||^2$

> $$F_\theta \iff \arg\min_\theta ||F(D) - y||^2$$

Implementation

Backpropagation

You need to implement the forward pass and backward pass for Linear, ReLU, Sigmoid, MSE loss, and BCE loss in the attached mlp.py file. We provide three example test cases test1.py, test2.py, test3.py. We will test your implementation with other hidden test cases, so please create your own test cases to make sure your implementation is correct.

Gradient Descent

Given a image classifier, implement a function that performs optimization on the input (the image), to find the image that most highly represents the class. Extra credit awarded for cool visuals. See gd.py.