# Homework 2

For the following problems, please use LaTex to type your solution. Please submit your solution in PDF. Handwritten solution will not be accepted. You may use the template to write down your solution: https://www.cs.ucsb.edu/~leili/course/dl23w/hw_template.tex.

## Problem 1: Forward propagation (20')

Let's consider a simple two layer neural network.

It has input size 2, one hidden layer size 3, and output size 2.

The input $x = \begin{bmatrix} 6 \\ 10 \end{bmatrix}$,

two sets of weights $W_1 = \begin{bmatrix} -0.3 & 0.2 \\ 0.6 & 0.6 \\ 0.4 & 0.8 \end{bmatrix}$, $W_2 = \begin{bmatrix} 1.8 & 0.9 & -0.5 \\ 0.5 & -1.1 & 0.2 \end{bmatrix}$,

and biases $b_1 = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}$, $b_2 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$.

1. (4') Calculate $z_1 = W_1 x + b_1$.
2. (4') Suppose we use the Sigmoid function $S(x)$ for activation for the hidden layer, write out the Sigmoid function formula, and calculate $a_1 = S(z_1)$.
3. (4') Calculate $z_2 = W_2 a_1 + b_2$.
4. (4') Use Softmax function $\sigma(x)$ to calculate $\hat{y} = \sigma(z_2)$.
5. (4') Explain the difference between Sigmoid and Softmax, and their role in neural networks.

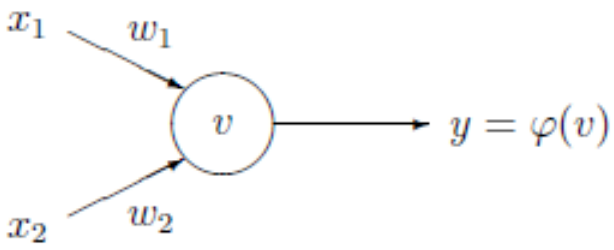
# Homework 2

## Problem 2: Forward Propagation (30')

Logical operators (i.e. NOT, AND, OR, XOR, etc) are the building blocks of any computational device. Logical functions return only two possible values, true or false, based on the truth or false values of their arguments. For example, the operator AND returns true only when all its arguments are true, otherwise (if any of the arguments is false) it returns false. If we denote truth by 1 and false by 0, then logical function AND can be represented by the following table:

| | | | | |
|---|---|---|---|---|
| $x_1$ : | 0 | 1 | 0 | 1 |
| $x_2$ : | 0 | 0 | 1 | 1 |
| $x_1$ AND $x_2$ : | 0 | 0 | 0 | 1 |

This function can be implemented by a single-unit with two inputs:




if the weights are w1 = 1 and w2 = 1 and the activation function is:

$$\varphi(v) = \begin{cases} 1 & \text{if } v \geq 2 \\ 0 & \text{otherwise} \end{cases}$$

Note that the threshold level is 2 (v $\geq$ 2).

1. (10') Test how the neural AND function works.
2. (10') Suggest how to change either the weights or the threshold level of this single-unit in order to implement the logical OR function (true when at least one of the arguments is true)
3. (10') The XOR function (exclusive or) returns true only when one of the arguments is true and another is false. Otherwise, it returns always false. Do you think it is possible to implement this function using a single unit? Explain your reason.

# Homework 2

## Problem 3: Backpropagation (30')

Let's assume we have a two layer neural network which is applied to a binary classification task. The architecture is defined below:

$$z_1 = W_1 x^{(i)} + b_1$$
$$a_1 = \mathrm{LR}(z_1)$$
$$z_2 = W_2 a_1 + b_2$$
$$\hat{y}^{(i)} = \sigma(z_2)$$
$$\mathcal{L}^{(i)} = -y^{(i)} * \log(\hat{y}^{(i)}) - (1 - y^{(i)}) * \log(1 - \hat{y}^{(i)})$$
$$J = \frac{1}{m} \sum_{i=1}^{m} \mathcal{L}^{(i)}$$

where $\mathcal{L}^{(i)}$ is called cross-entropy loss, LR is the leaky ReLU function, and $\sigma$ is the sigmoid function:

$$\mathrm{LR}(x) = \max(0.01x, x), \quad \sigma(x) = \frac{1}{1 + e^{-x}} \tag{1}$$

Now we have a dataset $\mathcal{D} = \{(x^{(i)}, y^{(i)})\}_{i=1}^{N}$ with N = 5000, where for each sample $x^{(i)}$ represents a single input and $y^{(i)}$ is the corresponding ground-truth label. The shape of $x^{(i)}$ is $64 \times 1$ and $y^{(i)}$ is a scalar. We use 128 nodes in the hidden layer, say, $z_1$ 's shape is $128 \times 1$ . Please list step by step procedure and calculate the final answers.

1. (5') What are the shapes of $W_1, b_1, W_2, b_2$ ? If we were vectorizing across multiple examples, what would be the shapes of $X$ and $Y$?
2. (4') What is $\partial J$ / $\partial \hat{y}^{(i)}$?
3. (4') What is $\partial \hat{y}^{(i)}$ / $\partial z_2$?
4. (4') What is $\partial z_2$ / $\partial a_1$?
5. (4') What is $\partial a_1$ / $\partial z_1$?
6. (4') What is $\partial z_1$ / $\partial W_1$?
7. (5') What is $\partial J$ / $\partial W_1$? You may reuse work from previous parts. Be careful with the shapes.

# Problem 4: Gradient Descent (Bonus, 20')

Let's assume we have a loss function $f(x, \theta)$ with learnable parameter $\theta$. Now we have a set of observations $\{x_i\}_{i=1}^n$ from data distribution p(x). Then the loss function is given by:

$$\mathcal{L} = \frac{1}{n} \sum_{i=1}^{n} f(x_i, \theta) \tag{2}$$

The learning rate is $\eta_t$ at step t. Please answer the following questions:

1. (5') If we use full gradient descent algorithm to learn the model, what is the full gradient step for step t?
2. (5') If we use stochastic gradient descent algorithm to learn the model, what is the stochastic gradient step for step t? Is it an unbiased estimator for full gradient descent?
3. (5') If we use mini-batch gradient descent with minibatch size b, what is the gradient descent step? Is it an unbiased estimator for full gradient descent?
4. (5') If we have dimension d for $\theta$, considering just one step, what are the respective time complexity for full gradient descent, stochastic gradient descent and mini-batch gradient descent?