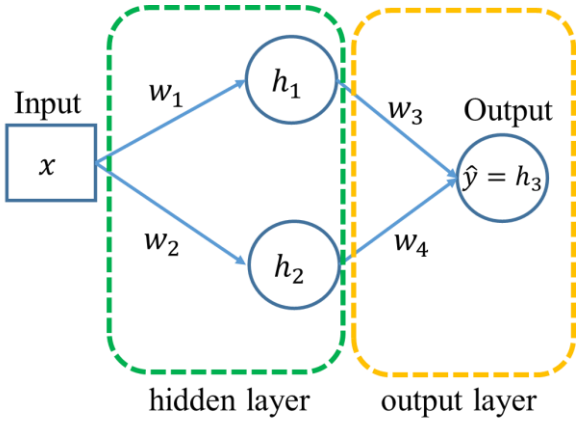
**Homework Assignment 4**

**Part-1: Basic Concepts**

## 1. Backpropagation In A Neural Network




$$h_1 = f_1(w_1 x + b_1)$$

$$h_2 = f_2(w_2 x + b_2)$$

$$\hat{y} = f_3(w_3 h_1 + w_4 h_2 + b_3)$$

$$f_n' = \frac{\partial f_n(v)}{\partial v},\ \text{n} = 1,2,3$$

$x, w_1, w_2, w_3, w_4, b_1, b_2, b_3, h_1, h_2, h_3$ are scalars

Compute the derivatives of the loss $L$ with respect to parameters and input, assuming $\frac{\partial L}{\partial h_3}$ is known.

$$\frac{\partial L}{\partial w_1} = \frac{\partial L}{\partial h_3}\frac{\partial h_3}{\partial h_1}\frac{\partial h_1}{\partial w_1} = \frac{\partial L}{\partial h_3} f_3' w_3 f_1' x$$

$\frac{\partial L}{\partial w_2}$

$\frac{\partial L}{\partial w_3}$

$\frac{\partial L}{\partial w_4}$

$\frac{\partial L}{\partial b_1}$

$\frac{\partial L}{\partial b_2}$

$\frac{\partial L}{\partial b_3}$

$\frac{\partial L}{\partial x}$

**2. Computational Graph**

$h = 2x + 1$

$z = x^2$

$y = \frac{1}{1 + e^{-h}}$

(1) Draw the computational graph based on the above three equations

(2) What is $\frac{\partial y}{\partial z}$ from the graph?

**3. Output Normalization for a Neural Network**

Usually, we need to apply normalization/standardization to the inputs for classification and regression tasks, so that the input will be in the range of 0 to 1, or -1 to +1. For example, if the input is an image, then every pixel value is divided by 255, so that the pixel values of the normalized image are in the range of 0 to 1. Input normalization facilitates the convergence of training algorithms.

We may also need to apply normalization to the output. Assume the input is an image of a person, the output vector has two components, $\hat{y}_{(1)}$ and $\hat{y}_{(2)}$: $\hat{y}_{(1)}$ is the monthly income (in the range of 0 to 10,000), and $\hat{y}_{(2)}$ is the age (in the range of 0 to 100). The MSE loss for a single data sample is

$$L = (\hat{y}_{(1)} - y_{(1)})^2 + (\hat{y}_{(2)} - y_{(2)})^2$$

where $y_{(1)}$ and $y_{(2)}$ are ground truth values of an input data sample.

Question: is output (i.e., the output target $y_{(1)}$, $y_{(2)}$) normalization necessary for this task? Why?

If it is necessary, what normalization can be applied?

**4. Activation Functions for Regression**

Neural networks can be used for regression. To model nonlinear input-output relationship, a neural network needs nonlinear activation functions in the hidden layers. Usually, the output layer does not need nonlinear activation functions. However, sometimes, there are requirements for outputs. For example, if the output is the sale price of a house, then the output should be nonnegative.

Assume $\boldsymbol{z}$ is the scalar output of a network, and the network does not have nonlinear activation function in the output layer. Now, there is some requirement for output, and you decide to add a nonlinear activation function.

You design nonlinear activation functions for three different requirements:

(1) the final output $y$ should be nonnegative ($y \geq 0$), then what is the activation function $y = f(z)$ ?

(2) the final output $y$ should be nonpositive ($y \leq 0$), then what is the activation function $y = f(z)$ ?

(3) the final output $y$ should be $a \leq y \leq b$, then what is the activation function $y = f(z)$ ?

You may use a combination of the basic activation functions that you can find in the lecture notes or the documents of Keras and Pytorch.

**5. Normalization inside a Neural Network**

It is often necessary to normalize the input (and output if the task is regression) for training a neural network. To facilitate the convergence of training a deep neural network, it is necessary to normalize the output of each layer of a neural network.

Read the paper https://arxiv.org/abs/1803.08494 and answer the questions:

(1) Batch Normalization will be highly unstable if batch_size is very small. Why?

(2) Why is Layer Normalization independent of batch_size?

**6. ReLU and piecewise linear**

Show that an MLP with ReLU acitivations is a piecewise linear function of the input

## Part-2 programming

Programming tasks: H4P2T1 and H4P2T2

(1) use Keras to get 80 points

(2) use Pytorch to get extra 10 points

I highly suggest that you use both packages, so that you can put this line in your resume for a job:

"familiar with deep learning packages (Keras and Pytorch)"

Note: the dateset for classification is class-imbalanced and you need to use weighted accuracy.

You will lose at least 20 points if you do not use weighted accuracy in H4P2T1.

**Grading**

The number of points for each question/task

| | Undergraduate Student | Graduate Student |
|---|---|---|
| 1. Backpropagation | 5 | 5 |
| 2. Computational Graph | 5 | 2.5 |
| 3. Output Normalization | 5 | 2.5 |
| 4. Activations for Regression | 5 | 5 |
| 5. Normalization inside Network | N.A. | 5 |
| 6. ReLU and piecewise linear | N.A. | extra 5 points |
| H4P2T1 (classification - Keras) | 40 | 40 |
| H4P2T2 (regression - Keras) | 40 | 40 |
| H4P2T1 (classification - Pytorch) | extra 5 points | extra 5 points |
| H4P2T2 (regression - Pytorch) | extra 5 points | extra 5 points |