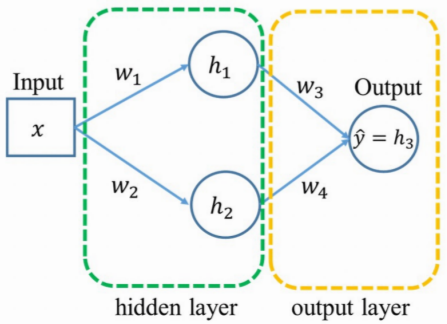
Diep Vu

Homework 4

1. Backpropagation in a Neural Network



$h_1 = f_1(w_1 x + b_1)$

$h_2 = f_2(w_2 x + b_2)$

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

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

$$\frac{\partial L}{\partial w_2} = \frac{\partial L}{\partial h_3}\frac{\partial h_3}{\partial h_2}\frac{\partial h_2}{\partial w_2} = \frac{\partial L}{\partial h_3} \cdot f_3' w_4 f_2' x$$

$$\frac{\partial L}{\partial w_3} = \frac{\partial L}{\partial h_3}\frac{\partial h_3}{\partial w_3} = \frac{\partial L}{\partial \hat{y}} f_3' h_1$$

$$\frac{\partial L}{\partial w_4} = \frac{\partial L}{\partial h_3}\frac{\partial h_3}{\partial w_3} = \frac{\partial L}{\partial \hat{y}} f_3' h_2$$

$$\frac{\partial L}{\partial b_1} = \frac{\partial L}{\partial h_3}\cdot\frac{\partial h_3}{\partial h_1}\frac{\partial h_1}{\partial b_1} = \frac{\partial L}{\partial h_3} f_3' w_3 f_1'$$

$$\frac{\partial L}{\partial b_2} = \frac{\partial L}{\partial h_3}\cdot\frac{\partial h_3}{\partial h_2}\frac{\partial h_2}{\partial b_2} = \frac{\partial L}{\partial h_3} \cdot f_3' w_4 f_2'$$

$$\frac{\partial L}{\partial b_3} = \frac{\partial L}{\partial h_3}\cdot\frac{\partial h_3}{\partial b_3} = \frac{\partial L}{\partial h_3} \cdot f_3'$$

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

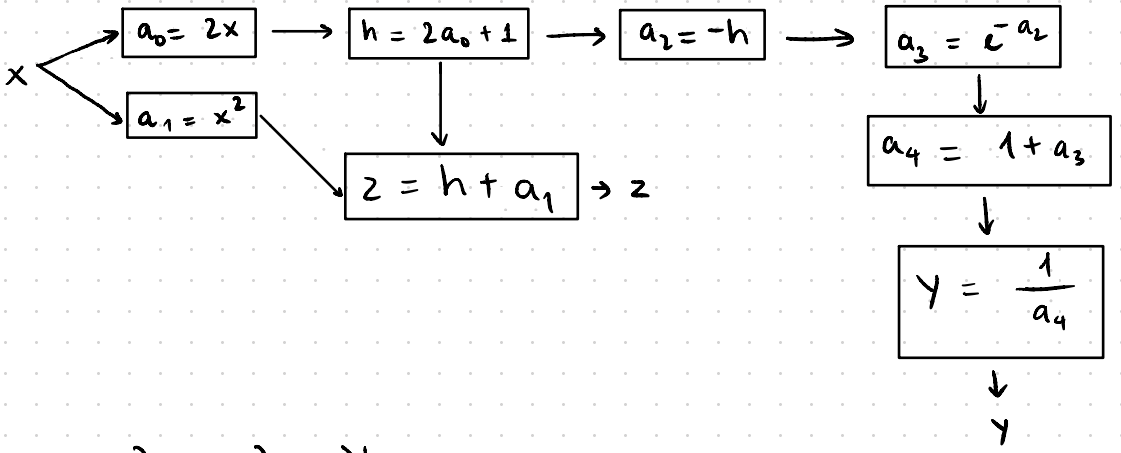
$$= \frac{\partial L}{\partial h_3}\cdot\frac{\partial h_3}{\partial h_2}\cdot\frac{\partial h_2}{\partial x} = \frac{\partial L}{\partial h_3} \cdot f_3' w_4 f_2' w_2$$

2. Computational Graph

$h = 2x + 1$ $\qquad y = \frac{1}{1+e^{-h}}$

$z = x^2 + h$

a. Computation graph based on those equations



b. $\frac{\partial y}{\partial z} = \frac{\partial y}{\partial h} \cdot \frac{\partial h}{\partial z}$

$\frac{\partial y}{\partial h} = \frac{\partial y}{\partial a_4} \cdot \frac{\partial a_4}{\partial a_3} \cdot \frac{\partial a_3}{\partial a_2} \cdot \frac{\partial a_2}{\partial h} = \frac{-1}{a_4^2} \cdot 1 \cdot e^{a_2} \cdot (-1)$ $\qquad a_3 = 1 + a_2 = 1 + e^{-h}$

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

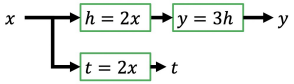
$\frac{\partial z}{\partial h} = 1 \rightarrow \frac{\partial y}{\partial z} = \frac{\partial y}{\partial h} \cdot \frac{\partial h}{\partial z} = \frac{-e^{-h}}{(1+e^{-h})^2}$ Mathematically

However, according to the slides

A computation process

$t = 2x$
$h = 2x$
$y = 3h$

The computation graph (3 nodes)



In 'pure' Math:
$y = 3t$
Thus:
$\frac{\partial y}{\partial t} = 3$

From the computation graph,
y is not a function of t, thus
$\frac{\partial y}{\partial t}$ is not defined: it does not exist.
$\frac{\partial y}{\partial t}$ can bet set to None or 0.

y is not a function of z, thus

$\frac{\partial y}{\partial z}$ is not defined: it does not exist

$\frac{\partial y}{\partial z}$ can be set to None or 0

3. Target (output) Normalization for a Neural Network

Yes, output normalization is necessary for this task since the range of y_1 and y_2 are very different (0 to 10,000) vs (0 to 100)

When calculating the loss L, the values of y will dominate in making the decision

The error in y_2 will be neglected → Poor result in y_2 → Training is y_1-biased

Since we know the constraints for $\hat{y}_1$ and $\hat{y}_2$, we can divide every $\hat{y}_1$ value by 10000 and $\hat{y}_2$ by 100

↳ Range of values will be 0 to 1

4. Activation Functions for Regression (Extra Credit)

a. $y \geq 0$, $y = f(z) = \begin{cases} 0 & \text{for } z < 0 \\ z & \text{for } z \geq 0 \end{cases}$ (ReLU)

b. $y \leq 0$, $y = f(z) = \tanh(z) = \frac{2}{1 + e^{-2x}} - 1$ (tanH)

c. $a \leq y \leq b$, $y = f(z) = z$ (Linear)

5. Normalization Inside a Neural Network

a. In batch normalization, these 2 statistics are taken into consideration: the mean and standard deviation corresponding to the current mini batch. If the size is too small, those values will not be representative enough of the actual distribution

b. Layer normalization is independent of batch size because the statistics for layer normalization are computed across all channels and spatial dime

6. Skip Connections in a Neural Network

Skip connections introduce identity mapping that bypasses a convulational block. Skip / residual connections are useful to build a deep network because they allow information and gradients to flow more easily through the network → Improving the network's accuracy.

7. Randomness of a Neural Network

The cause of the randomness might be the initialization of the weights. The weights of the networks are initialized randomly → Each model has different weights → Perform differently on the test set

It depends on the purpose of the paper.

If the paper is about showing the method is effective on a certain dataset, I would choose to report the best model.

If the paper is about randomness of a neural network, I would report all 3 and analyze each.