2023/09/06

Exercise 3

Consider a sigmoid neuron with 1D input x, weight w, bias b and output $y = \sigma(wx+b)$. The target is the variable z. Consider the cost function $J(w,b) = \frac{1}{2}(y-z)^2$.

- Find $\nabla J(w,b)$ and show that $\|\nabla J\| < \frac{1}{4}\sqrt{1+x^2}(1+|z|)$.
- Write the gradient descent iteration for the sequence (w_n, b_n).

· Gradient $\nabla J(w,b)$

define $u = wx + b$ $\quad y = \sigma(u)$ $\quad J = \frac{1}{2}(y-z)^2$

Using chain rule:
$$\begin{cases} \frac{\partial J}{\partial w} = \frac{\partial J}{\partial y} \cdot \frac{\partial y}{\partial u} \cdot \frac{\partial u}{\partial w} \\ \frac{\partial J}{\partial b} = \frac{\partial J}{\partial y} \cdot \frac{\partial y}{\partial u} \cdot \frac{\partial u}{\partial b} \end{cases}$$

↳ compute each derivative

$$\begin{cases} \frac{\partial J}{\partial y} = y - z \\ \frac{\partial y}{\partial u} = \sigma(u)(1-\sigma(u)) = y(1-u) \\ \frac{\partial u}{\partial w} = x, \quad \frac{\partial u}{\partial b} = 1 \end{cases}$$
(derivative of sigmoid)

→ chain rule
$$\begin{cases} \frac{\partial J}{\partial w} = (y-z) \cdot y(1-y)x \\ \frac{\partial J}{\partial b} = (y-z) \cdot y(1-y) \end{cases}$$

· Bound on $\|J\| = \sqrt{\left(\frac{\partial J}{\partial w}\right)^2 + \left(\frac{\partial J}{\partial b}\right)^2}$

with $0 < y < 1 \rightarrow 0 < y(1-y) \leq \frac{1}{4}$ (maximum when $y = 0.5 \rightarrow y(1-y) = 0.25$)

denote $A = (y-z)\, y(1-y)$

$$\left(\frac{\partial J}{\partial w}\right)^2 + \left(\frac{\partial J}{\partial b}\right)^2 = A^2(x^2+1)$$

using the bound $|A| = |y-z| \cdot y(1-y) \leq (|y| + |z|) \cdot \frac{1}{4} < \frac{1+|z|}{4}$

hence $\|\nabla J\| < \sqrt{(x^2+1)} \cdot \left(\frac{1+|z|}{4}\right) = \frac{1}{4}\sqrt{1+x^2}\,(1+|z|)$

· Gradient descent iteration

Take $\eta > 0$

$$w^{i+1} = w^i - \eta \frac{\partial J}{\partial w} = w^i - \eta (y-z) y (1-y) x$$

$$b^{i+1} = b^i - \eta \frac{\partial J}{\partial w} = b^i - \eta (y-z) y (1-y)$$

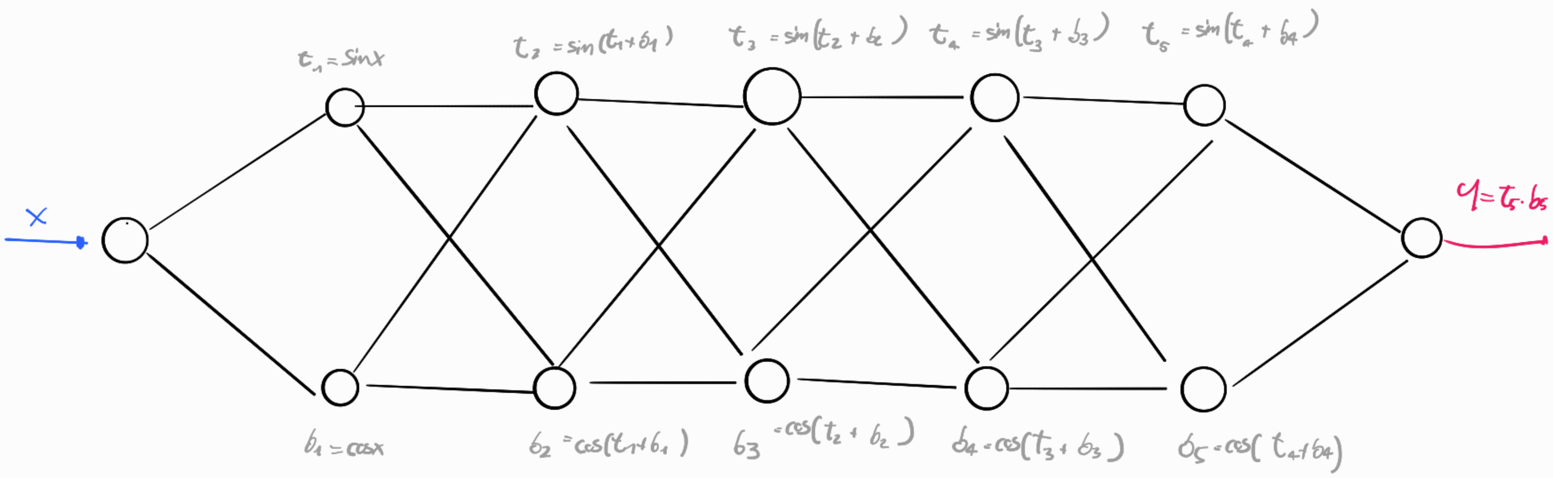
(with $y = \sigma(w^i x + b^i)$)

2023/07/06

Exercise 3
Consider the following computational graph:

Figure 1: Computational Graph

The upper node in each layer computes $\sin(x+y)$ and the lower computes $\cos(x+y)$ with respect to its 2 inputs. For the first hidden layer, there is only a single input x, and therefore the values $\sin(x)$ and $\cos(x)$ are computed. The final node computes the product of the two inputs. The single input is denoted by x (value in radiants). Compute the numerical value of the partial derivative of the output with respect to x for $x = 1$ using the backpropagation algorithm. Explain clearly each step you have performed.



$x = 1$

Forward pass

h_1: $t_1 = \sin(1) = 0,84$; $b_1 = \cos(1) = 0.54$

h_2: $t_1 + b_1 = 1,38$; $t_2 = \sin(1,38) = 0,98$; $b_2 = \cos(1.38) = 0,18$

h_3: $t_2 + b_2 = 1.16$; $t_3 = \sin(1.16) = 0,91$; $b_3 = \cos(1.16) = 0,40$

h_4: $t_3 + b_3 = 1,31$; $t_4 = \sin(1.31) = 0,97$; $b_4 = \cos(1.31) = 0,25$

h_5: $t_4 + b_4 = 1,22$; $t_5 = \sin(1.22) = 0,94$; $b_5 = \cos(1.22) = 0,34$

output $y = t_5 \cdot b_5 = 0,32$

Backward pass $\left(\frac{d \sin x}{dx} = \cos x \,,\ \frac{d \cos x}{dx} = -\sin x \right)$ $\quad \dot{t} = \frac{\partial y}{\partial t}$

obs: T and b have same contribute on derivative (symmetry!)

h_5: $\dot{t}_5 = b_5 = 0,34$; $\dot{b}_5 = t_5 = 0,94$

h_4: $\dot{t}_4 = \dot{t}_5 \cdot \cos(t_4 + b_4) + \dot{b}_5 (-\sin(t_4+b_4)) = \dots = -0,77$; $\dot{b}_4 = -0,77$

$$h_3 \begin{cases} \dot{t_3} = \dot{t_4} \cos(t_3 + b_3) + \dot{b_4} \left(-\sin(t_3 + b_3)\right) = \dots \quad 0{,}54 \\ \dot{b_3} = 0{,}54 \end{cases}$$

$$h_2 \begin{cases} \dot{t_2} = \dot{t_3} \cos(t_2 + b_2) \cdot \dot{b_3} \left(-\sin(t_2 + b_2)\right) = \dots \quad -0{,}28 \\ \dot{b_2} = -0{,}28 \end{cases}$$

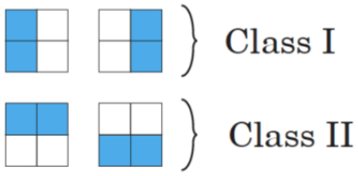
$$h_1 \begin{cases} \dot{t_1} = \dot{t_2} \cos(t_1 + b_1) \cdot \dot{b_2} \left(-\sin(t_1 + b_1)\right) = 0{,}22 \\ \dot{b_1} = 0{,}22 \end{cases}$$

$$\text{input} \quad \dot{x} = \dot{t_1} \cos(1) + \dot{b_1} \left(-\sin(1)\right) = -0{,}07$$

2023/06/15

Exercise 3

Consider the two classes of patterns that are shown in the following figure where Class I represents vertical lines and Class II represents horizontal lines.



1. Are these categories linearly separable ?
2. Design a multilayer network to distinguish these categories.

• Encode input as $[x_1, x_2, x_3, x_4]$

$$\begin{array}{cccc} x_1 & x_2 & x_3 & x_4 \end{array}$$

$[1 \; 0 \; 1 \; 0] \rightarrow$ class 1

$[0 \; 1 \; 0 \; 1] \rightarrow$ class 1

$[1 \; 1 \; 0 \; 0] \rightarrow$ class 2

$[0 \; 0 \; 1 \; 1] \rightarrow$ class 2

↳ Linear decision boundary in the form $W^T x + b > 0$ (e.g. $>0 \rightarrow$ class 1)

$[1 \; 0 \; 1 \; 0] \rightarrow w_1 + w_3 > 0$ (1)

$[0 \; 1 \; 0 \; 1] \rightarrow w_2 + w_4 > 0$ (2)

$[1 \; 1 \; 0 \; 0] \rightarrow w_1 + w_2 < 0$ (3)

$[0 \; 0 \; 1 \; 1] \rightarrow w_3 + w_4 < 0$ (4)

→ It raises inconsistencies :

From (1) and (3) $\rightarrow \begin{cases} w_1 + w_3 > 0 \\ -w_1 > w_2 \end{cases} \xrightarrow{\text{sum}} W_3 > W_2$ ⚠

From (2) and (4) $\rightarrow \begin{cases} w_2 + w_4 > 0 \\ -w_4 > w_3 \end{cases} \xrightarrow{\text{sum}} W_2 > W_3$

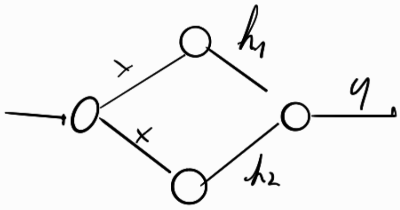
• Multi-layer perceptron

vertical detector : $h_1 = \sigma(x_1 + x_3 - x_2 - x_4)$

horizontal detector : $h_2 = \sigma(x_1 + x_2 - x_3 - x_4)$

output : $y = \sigma(w_1 h_1 + w_2 h_2 + b)$

loss : cross-entropy



2023/02/07

Exercise 3

Suppose that the output $\hat{y}_k$ of a given unit in a neural network is given by the softmax function *i.e.*:

$$\hat{y}_k = \frac{\exp(a_k)}{\sum_j \exp(a_j)}. \tag{2}$$

- Show that the output of the softmax function does not change if you shift, in all components, the activations a_j by some constant c.
- Explain why the shift $c = -\max_j(a_j)$ can be useful.

· Invariance to shift

suppose shift all activation a_j by constant c : $a_j' = a_j + c$

$$\hat{y}_k' = \frac{e^{a_k+c}}{\sum_j e^{a_j+c}} = \frac{e^{a_k} e^{c}}{\sum_j e^{a_j} e^{c}} = \frac{e^{a_k} \cancel{e^{c}}}{\cancel{e^{c}} \sum_j e^{a_j}} = \frac{e^{a_k}}{\sum_j e^{a_j}} = \hat{y}_k$$

· Choosing $a_j' = a_j - \max_j (a_j)$

obs: $a_j' \leq 0$ it helps numerical stability!

- when some a_j are large e^{a_j} can overflow
- if maximum is $\emptyset$ we ensure all components to be ≤ 1 and avoid overflo!

2023/01/20

Exercise 3

Show that a multi-layer neural network with linear activation function $s(x) = x$ is equivalent to a single layer linear network. Assume that in each layer the inputs follow a Normal distribution with mean zero and small variance, *i.e.* $\sigma \ll 1$. For which of the activation functions $s(x) = 1/(1 + \exp(-x))$, $s(x) = \tanh(x)$, $s(x) = \mathrm{relu}(x)$ and $s(x) = \mathrm{selu}(x)$ is a deep network equivalent to a linear network for the given distribution ? The selu function is given by:

$$\mathrm{selu}(x) = \begin{cases} \lambda x \text{ if } x > 0 \\ \alpha(\exp(x) - 1) \text{ otherwise} \end{cases}, \tag{4}$$

where $\lambda \approx 1.0507$ and $\alpha \approx 1.75814$. (*Hint:* consider the case $\sigma \to 0$ using a Taylor expansion around 0.)

- Deep Linear = single linear

MLP (with L layers) has output: $y = W^L \cdot W^{L-1} \cdot \ldots W^2 \cdot W^1 \cdot x$

we can just handle $W^L \cdot W^{L-1} \ldots W^2 \cdot W^1 = W_{eff}$ $\longrightarrow y = W_{eff}\, x$

(single neuron!)

- for which activation functions is a deep approximately linear when $x \sim N(0, \sigma^2)$ with $\sigma^2 \ll 1$?

1) Sigmoid $s(x) = \dfrac{1}{1+e^{-x}}$

Taylor expansion in $\emptyset$: $s(x) \approx \frac{1}{2} + \frac{x}{4} - \frac{x^3}{48} + \ldots$

for small σ $s(x) \approx \frac{1}{2} + \frac{x}{4}$ Not linear: constant bias breaks linearity over multiple layers ✗

2) $\tanh(x)$

Taylor expansion $s(x) = x - \frac{x^3}{3} + \frac{2x^5}{15} - \ldots$

for small σ $s(x) = x$ Linear ✓

3) $\mathrm{Relu}(x) = \max(0, x)$

$x < 0$ → output 0

$x > 0$ → output x

↳ normal distribution around $\emptyset$ means that half input are mapped into $\emptyset$

Non linearity!

4) $\text{Selu}(x) = \begin{cases} \lambda x & x > 0 \\ \lambda\alpha(e^{x}-1) & x \leq 0 \end{cases}$

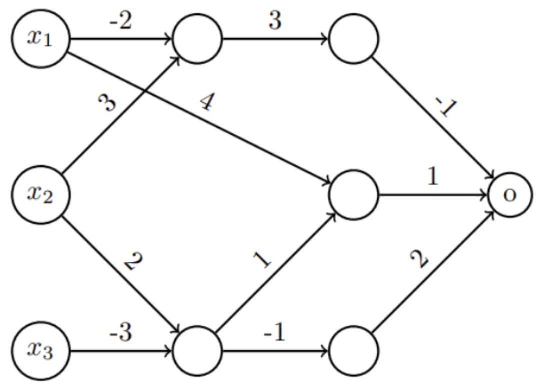
Taylor expansion $\begin{cases} \lambda x & x > 0 \\ \lambda\alpha\left(x + \frac{x^2}{2} + \dots\right) \approx \lambda\alpha x & x \leq 0 \end{cases}$

↳ this is piecewise linear (but not globally) slope differs between $x > 0$ and $x < 0$

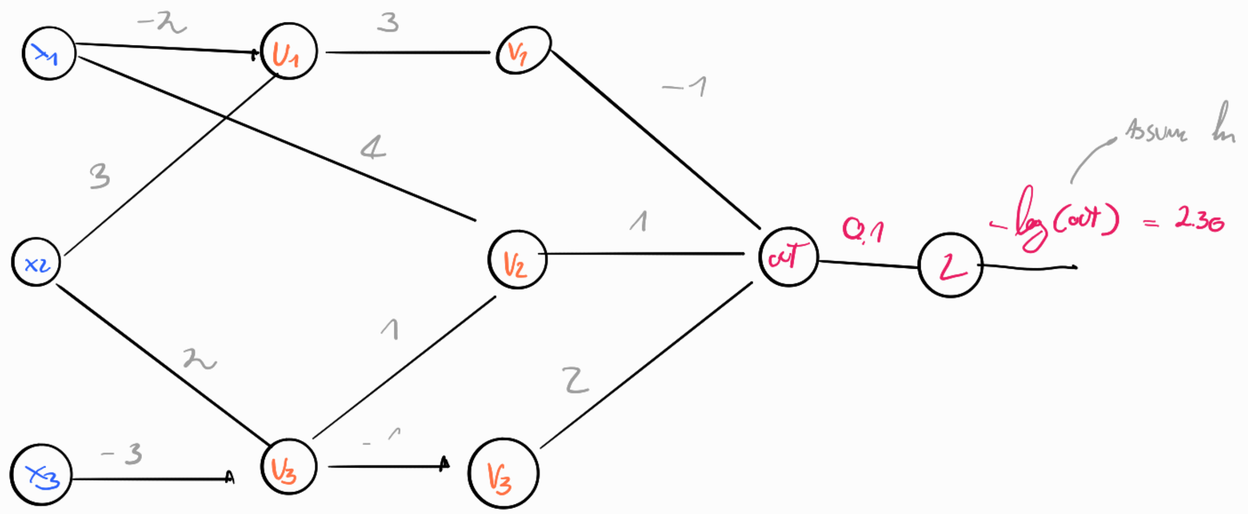
2022/09/01

Exercise 3

Consider the following network where on each edge (i, j) the value of $\frac{\partial y(j)}{\partial y(i)}$ is given; $y(k)$ denotes the activation of node k.



The output o is equal to 0.1 and the loss function is $L = -log(o)$. Compute the value of $\frac{\partial L}{\partial x_i}$ for each input x_i using the backpropagation method.



$$\frac{\partial L}{\partial out} = -\frac{1}{out} = -10$$

$$\frac{\partial L}{\partial V_1} = \frac{\partial L}{\partial out} \cdot \frac{\partial out}{\partial V_1} = -10 \cdot -1 = 10$$

$$\frac{\partial L}{\partial V_2} = \frac{\partial L}{\partial out} \cdot \frac{\partial out}{\partial V_2} = -10 \cdot 1 = -10$$

$$\frac{\partial L}{\partial V_3} = \frac{\partial L}{\partial out} \cdot \frac{\partial out}{\partial V_3} = -10 \cdot 2 = -20$$

$$\frac{\partial L}{\partial U_1} = \frac{\partial L}{\partial V_1} \cdot \frac{\partial V_1}{\partial U_1} = 10 \cdot 3 = 30$$

$$\frac{\partial L}{\partial U_3} = \frac{\partial L}{\partial V_2} \cdot \frac{\partial V_2}{\partial U_3} + \frac{\partial L}{\partial V_3} \cdot \frac{\partial V_3}{\partial U_3} = (-10 \cdot 1) + (-20 \cdot -1) = 10$$

$$\frac{\partial L}{\partial x_1} = \frac{\partial L}{\partial u_1} \cdot \frac{\partial u_1}{\partial x_1} + \frac{\partial L}{\partial V_2} \cdot \frac{\partial V_1}{\partial u_1} = (30 \cdot -2) + (-10 \cdot 4) = -60 - 40 = -100$$

$$\frac{\partial L}{\partial x_2} = \frac{\partial L}{\partial u_1} \cdot \frac{\partial u_1}{\partial x_2} + \frac{\partial L}{\partial u_3} \cdot \frac{\partial u_3}{\partial x_2} = (30 \cdot 3) + (10 \cdot 2) = 90 + 20 = 110$$

$$\frac{\partial L}{\partial x_3} = \frac{\partial L}{\partial u_3} \cdot \frac{\partial u_3}{\partial x_3} = 10 \cdot -3 = -30$$

2022/07/08

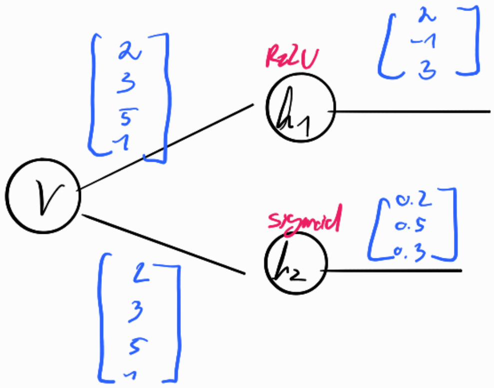
Exercise 3

Consider a neural network in which a vectored node v feeds into two distinct vectored nodes h_1 and h_2 computing different functions. The functions computed at the nodes are $h_1 = ReLU(W_1 v)$ and $h_2 = sigmoid(W_2 v)$. We do not know anything about the values of the variables in other parts of the network, but we know that $h_1 = [2, -1, 3]^T$ and $h_2 = [0.2, 0.5, 0.3]^T$, that are connected to the node $v = [2, 3, 5, 1]^T$. Furthermore the loss gradients are $\frac{\partial L}{\partial h_1} = [-2, 1, 4]^T$ and $\frac{\partial L}{\partial h_2} = [1, 3, 2]^T$, respectively. Show that the backpropagated loss gradient $\frac{\partial L}{\partial v}$ can be computed in terms of W_1 and W_2 as follows:

$$\frac{\partial L}{\partial v} = W_1^T \begin{bmatrix} -2 \\ 0 \\ 4 \end{bmatrix} + W_2^T \begin{bmatrix} 0.16 \\ 0.75 \\ -0.42 \end{bmatrix} \tag{5}$$

What are the sizes of W_1, W_2 and $\frac{\partial L}{\partial v}$?

Remember that $ReLU(x) = \max(0, x)$ and $sigmoid(x) = \frac{\exp(x)}{(\exp(x)+1)}$.



$$\frac{\partial L}{\partial h_1} = \begin{bmatrix} -2 \\ 1 \\ 4 \end{bmatrix}$$

$$\frac{\partial L}{\partial h_2} = \begin{bmatrix} 1 \\ 3 \\ 2 \end{bmatrix}$$

- With backpropagation rule

$$\frac{\partial L}{\partial v} = \frac{\partial L}{\partial h_1} \cdot \frac{\partial h_1}{\partial v} + \frac{\partial L}{\partial h_2} \cdot \frac{\partial h_2}{\partial v}$$

$$= W_1^T \left(\frac{\partial L}{\partial h_1} \odot \frac{\partial h_1}{\partial z_1} \right) + W_2^T \left(\frac{\partial L}{\partial h_2} \odot \frac{\partial h_2}{\partial z_2} \right)$$

with $z_1 = W_1 v$, $h_1 = ReLU(z_1) \Longrightarrow \frac{\partial h_1}{\partial z_1} = ReLU'(z_1)$

$z_2 = W_2 v$, $h_2 = sigmoid(z_2) \rightarrow \frac{\partial h_2}{\partial z_2} = \sigma(z_2)(1-\sigma(z_2))$ $\sigma(z_2)$

$$\Longrightarrow \frac{\partial h_1}{\partial z_1} \qquad ReLU' = \begin{cases} 1 & z_i > 0 \\ \emptyset & z_i \le 0 \end{cases}$$

from $h_1 = \begin{bmatrix} 2 \\ -1 \\ 3 \end{bmatrix} \longrightarrow z_1 = W_1 v = \begin{bmatrix} 2 \\ -1 \\ 3 \end{bmatrix} \longrightarrow ReLU'(z_1) = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}$

$$\rightarrow \frac{\partial L}{\partial h_1} \odot ReLU'(z_1) = \begin{bmatrix} -2 \\ 1 \\ 4 \end{bmatrix} \odot \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} -2 \\ 0 \\ 4 \end{bmatrix}$$

$$\Longrightarrow \frac{\partial h_2}{\partial z_2} \qquad \sigma'(z) = \sigma'(z)\left(1-\sigma(z)\right)$$

From $h_2 = \begin{bmatrix} 0.2 \\ 0.5 \\ 0.3 \end{bmatrix} \longrightarrow z_2 = W_2 V \begin{bmatrix} 0.2 \\ 0.5 \\ 0.3 \end{bmatrix} \longrightarrow \sigma'(z_2) = \begin{bmatrix} 0.2 \cdot 0.8 \\ 0.5 \cdot 0.5 \\ 0.3 \cdot 0.7 \end{bmatrix} = \begin{bmatrix} 0.16 \\ 0.25 \\ 0.21 \end{bmatrix}$

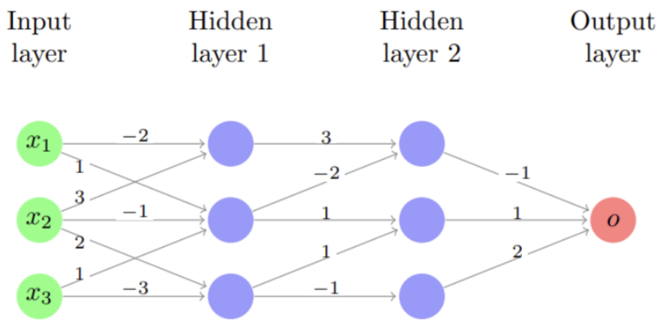
$$\Longrightarrow \frac{\partial L}{\partial h_2} \odot \sigma'(z_2) = \begin{bmatrix} 1 \\ 3 \\ 2 \end{bmatrix} \odot \begin{bmatrix} 0.16 \\ 0.25 \\ 0.21 \end{bmatrix} = \begin{bmatrix} 0{,}16 \\ 0.25 \\ 0{,}42 \end{bmatrix}$$

Sizes: $V \in \mathbb{R}^4 \qquad h_1, h_2 \in \mathbb{R}^3 \qquad W_1, W_2 \in \mathbb{R}^{3\times 4} \qquad \frac{\partial L}{\partial V} \in \mathbb{R}^4$

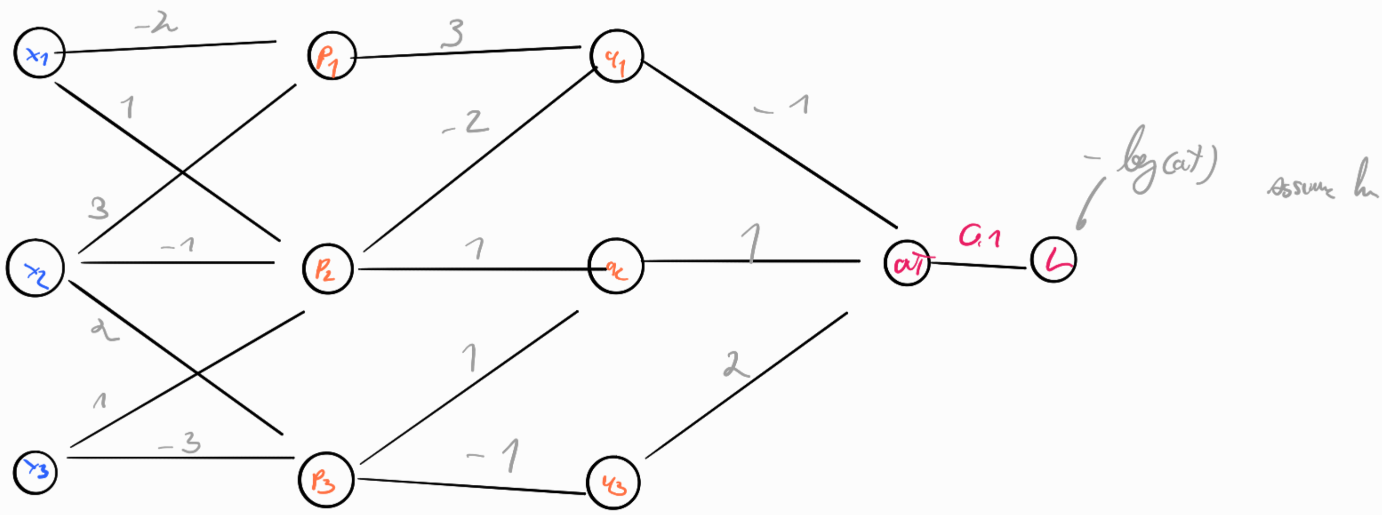
2022/06/16

Exercise 3

Consider the following network where on each edge (i, j) the value of $\frac{\partial y(j)}{\partial y(i)}$ is given; $y(k)$ denotes the activation of node k.



The output o is equal to 0.1 and the loss function is $L = -log(o)$. Compute the value of $\frac{\partial L}{\partial x_i}$ for each input x_i using the backpropagation method.



Backpropagation

$$\frac{\partial L}{\partial out} = -\frac{1}{out} = \frac{1}{0.1} = -10$$

$$\frac{\partial L}{\partial q_1} = \frac{\partial L}{\partial out} \cdot \frac{\partial out}{\partial q_1} = -10 \cdot -1 = +10$$

$$\frac{\partial L}{\partial q_2} = \frac{\partial L}{\partial out} \cdot \frac{\partial out}{\partial q_2} = -10 \cdot 1 = -10$$

$$\frac{\partial L}{\partial q_3} = \frac{\partial L}{\partial out} \cdot \frac{\partial out}{\partial q_3} = -10 \cdot 2 = -20$$

$$\frac{\partial L}{\partial p_1} = \frac{\partial L}{q_1} \cdot \frac{\partial q_1}{\partial p_1} = 10 \cdot 3 = 30$$

$$\frac{\partial L}{\partial p_2} = \frac{\partial L}{q_1} \cdot \frac{\partial q_1}{\partial p_2} + \frac{\partial L}{\partial q_2} \cdot \frac{\partial q_2}{\partial p_2} = (10 \cdot -2) + (-10 \cdot 1) = -30$$