1. Assume we have a dataset {(x(1), y(1)), (x(2), y(2)), …, (x(n), y(n))} of points in the 2D plane and we want to estimate the following **linear regression** model: ^y = w1*x + w0.
   Compute w0, w1 in terms of x(1), y(1), …, x(n), y(n). Numerical application (1, 1), (2, -1), (3, 2).

Assume we have a dataset $\{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \ldots, (x^{(n)}, y^{(n)})\}$ of points in the 2D plane, and we want to estimate the following linear regression model:

$$\hat{y} = w_1 x + w_0.$$

Compute w_0, w_1 in terms of $x^{(1)}, y^{(1)}, x^{(2)}, y^{(2)}, \ldots, x^{(n)}, y^{(n)}$. Numerical application: $(1, 1), (2, -1), (3, 2)$.

Linear Regression

1) Assume we have a dataset $\{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \dots, (x^{(m)}, y^{(m)})\}$ of points in the 2D plane and we want to estimate the following linear regression model:

$$\hat{y} = w_1 x + w_0$$

Compute w_0, w_1 in terms of $x^{(1)}, y^{(1)}, x^{(2)}, y^{(2)}, \dots, x^{(m)}, y^{(m)}$. Numerical application: $(1,1), (2,-1), (3,2)$

$m = 3$

$$X = \begin{bmatrix} 1 & 1 \\ 1 & 2 \\ 1 & 3 \end{bmatrix}, \quad y = \begin{bmatrix} 1 \\ -1 \\ 2 \end{bmatrix}$$

$$\boxed{w^* = (X^T \cdot X)^{-1} \cdot X^T \cdot y}$$

$$X^T \cdot X = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 3 \end{bmatrix} \cdot \begin{bmatrix} 1 & 1 \\ 1 & 2 \\ 1 & 3 \end{bmatrix} = \begin{bmatrix} 3 & 6 \\ 6 & 14 \end{bmatrix}$$

$(X^T \cdot X)^{-1}$ — A

$$\boxed{A^{-1} = \frac{1}{\det(A)} \cdot A^*}$$

$$A^* = \begin{bmatrix} |a_{22}| \ (\det) & - \\ - & |a_{11}| \ (\det) \end{bmatrix}$$

$$\det(A) = \begin{vmatrix} 3 & 6 \\ 6 & 14 \end{vmatrix} = 42 - 36 = 6$$

$$A^T = \begin{bmatrix} 3 & 6 \\ 6 & 14 \end{bmatrix} \rightarrow A^* = \begin{bmatrix} 14 & -6 \\ -6 & 3 \end{bmatrix}$$

! La matricile 2×2 A^* - se inversează elementele de pe diagonala principală, iar cele de pe diagonala secundară le negăm

$$A^{-1} = \frac{1}{6} \cdot \begin{bmatrix} 14 & -6 \\ -6 & 3 \end{bmatrix}$$

$$w = \frac{1}{6} \cdot \begin{bmatrix} 14 & -6 \\ -6 & 3 \end{bmatrix} \cdot \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 3 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ -1 \\ 2 \end{bmatrix} = \frac{1}{6} \cdot \begin{bmatrix} -2 \\ 3 \end{bmatrix} = \begin{bmatrix} -\frac{1}{3} \\ \frac{1}{2} \end{bmatrix} \begin{matrix} w_0 \\ w_1 \end{matrix}$$

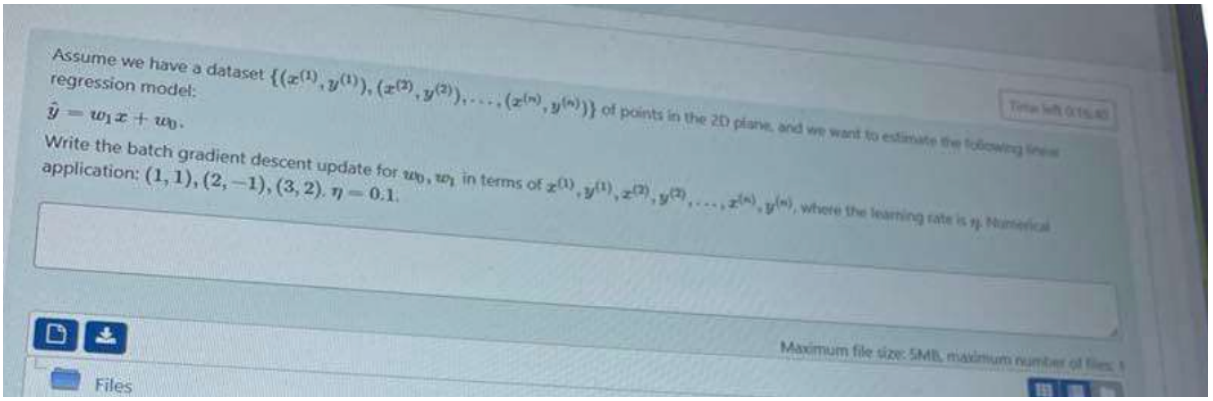
+ eroarea

$err = \sum (y - \hat{y})^2$ ← eroarea

$\hat{y} = X \cdot w = \begin{bmatrix} 1 & 1 \\ 1 & 2 \\ 1 & 3 \end{bmatrix} \cdot \begin{bmatrix} -\frac{1}{3} \\ \frac{1}{2} \end{bmatrix} = \begin{bmatrix} \frac{1}{6} \\ \frac{2}{3} \\ \frac{7}{6} \end{bmatrix}$

$err = \left(1 - \frac{1}{6}\right)^2 + \left(-1 - \frac{2}{3}\right)^2 + \left(2 - \frac{7}{6}\right)^2 = \frac{25}{6} = 4,1(6)$

2. Assume we have a dataset {(x(1), y(1)), (x(2), y(2)), …, (x(n), y(n))} of points in the 2D plane and we want to estimate the following linear regression model: ^y = w1*x + w0.
   Write the batch **gradient descent** update for w0, w1 in terms of x(1), y(1), …, x(n), y(n) where the following rate is n. Numerical application: (1, 1), (2, -1), (3, 2), n=0.1.



Gradient descendent

2) Assume we have a dataset $\{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \dots, (x^{(m)}, y^{(m)})\}$ of points in the 2D plane and we want to estimate the following linear regression model:

$$\hat{y} = w_1 x + w_0$$

Write the batch gradient descent update for w_0, w_1 in terms of $x^{(1)}, y^{(1)}, x^{(2)}, y^{(2)}, \dots, x^{(m)}, y^{(m)}$ where the learning rate is η. Numerical application: $(1,1), (2,-1), (3,2)$, $\eta = 0,1$.

$$\boxed{\frac{1}{m} \cdot x^T \cdot (xw - y) = \nabla_w f(w)}, \quad m = 3$$

$$x = \begin{bmatrix} 1 & 1 \\ 1 & 2 \\ 1 & 3 \end{bmatrix}, \quad y = \begin{bmatrix} 1 \\ -1 \\ 2 \end{bmatrix}$$

$$x^T = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 3 \end{bmatrix}$$

$$w \leftarrow w - \eta \nabla_w f(w)$$

$$w \leftarrow \begin{bmatrix} w_0 \\ w_1 \end{bmatrix} - \frac{0,1}{3} \cdot x^T \cdot \underbrace{\begin{pmatrix} w_0 + w_1 \\ w_0 + 2w_1 \\ w_0 + 3w_1 \end{pmatrix}}_{x \cdot w} - \begin{pmatrix} 1 \\ -1 \\ 2 \end{pmatrix}$$

$$\begin{bmatrix} w_0 \\ w_1 \end{bmatrix} \leftarrow \begin{bmatrix} w_0 \\ w_1 \end{bmatrix} - \frac{0,1}{3} \cdot x^T \cdot \begin{bmatrix} w_0 + w_1 - 1 \\ w_0 + 2w_1 + 1 \\ w_0 + 3w_1 - 2 \end{bmatrix}$$

$$\begin{bmatrix} w_0 \\ w_1 \end{bmatrix} \leftarrow \begin{bmatrix} w_0 \\ w_1 \end{bmatrix} - \frac{0,1}{3} \begin{bmatrix} 3w_0 + 6w_1 - 2 \\ 6w_0 + 14w_1 - 5 \end{bmatrix}$$

$$\begin{cases} w_0 \leftarrow w_0 - 0,1w_0 - 0,2w_1 + \frac{0,2}{3} \\ w_1 \leftarrow w_1 - 0,2w_0 - \frac{1,4}{3} w_1 + \frac{0,5}{3} \end{cases}$$

3. Assume we have a training set for two-class classification in two dimensions that contains seven sample points: points (1, 1), (2, 2), (3, 1) with label +1 and points (4, 5), (5, 7), (6, 5), (7, 7) with label -1. Which points are the support vectors for a **hard margin SVM**?

> Assume we have a training set for two-class classification in two dimensions that contains seven sample points: points $(1, 1), (2, 2), (3, 1)$ with label $+1$ and points $(4, 5), (5, 7), (6, 5), (7, 7)$ with label -1. Which points are the support vectors for a hard-margin SVM?

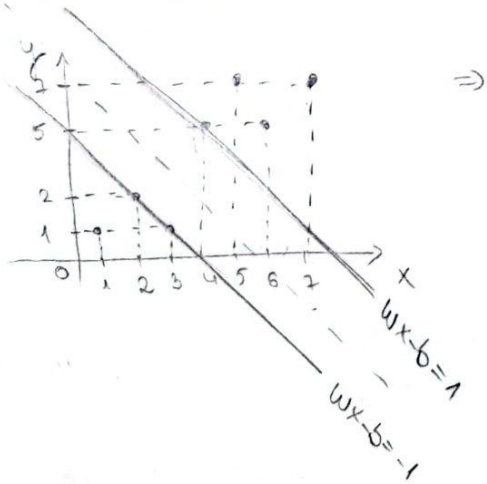
SVM

3) Assume we have a training set for two-class classification in two dimensions that contains seven sample points: points (1,1), (2,2), (3,1) with label +1 and points (4,5), (5,7), (6,5), (7,7) with label -1. Which points are the support vectors for a hard-margin SVM?

- punctele care sunt pe margine: support vectors (grafic)
  (3; 3,5) → mij punctelor (4,5), (2,2)

$$\min_{w, w_0} \frac{1}{2}\|w\|_2^2$$

$$y^{(i)}(w^T x^{(i)} + w_0) \geq 1, \quad 1 \leq i \leq m$$



⇒ punctele care pot fi support vectors sunt: (2,2), (3,1) și (4,5)
(2,2), (3,1) cu label +1; (4,5) cu label -1

⇒ (3; 3,5)

4. Consider a training classification **decision tree** given a design matrix X = [6 3; 2 7; 9 6; 4 2] and labels y = [1; 0; 1; 0]. Let x1 denote feature 1, corresponding to the first column of X and let x2 denote feature 2, corresponding to the second column of X, Which of the following splits at the root node gives the smallest entropy: x1 > 2, x1 > 4, x2 > 3, x2 > 6? Motivate your answer

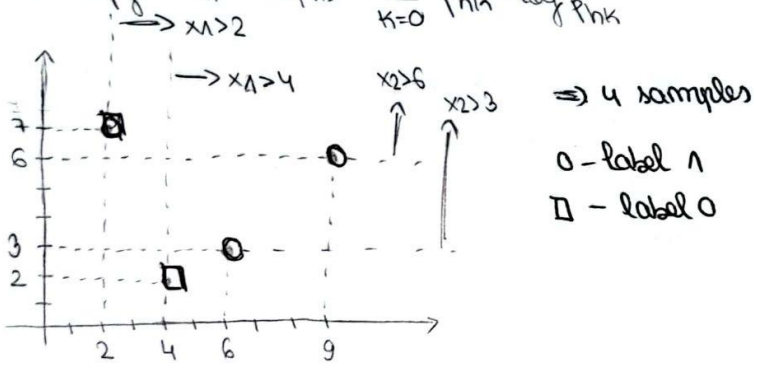
Consider training a classification decision tree given a design matrix $X = \begin{bmatrix} 6 & 3 \\ 2 & 7 \\ 9 & 6 \\ 4 & 2 \end{bmatrix}$ and labels $y = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix}$. Let x_1 denote feature 1, corresponding to the first column of X, and let x_2 denote feature 2, corresponding to the second column of X. Which of the following splits at the root node gives the smallest entropy: $x_1 > 2$, $x_1 > 4$, $x_2 > 3$, $x_2 > 6$? Motivate your answer.

Decision tree

4) Consider a training classification decision tree given a design matrix $X = \begin{bmatrix} 6 & 3 \\ 2 & 7 \\ 9 & 6 \\ 4 & 2 \end{bmatrix}$ and labels $y = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix}$. Let x_1 denote feature 1, corresponding to the first column of X and let x_2 denote feature 2, corresponding to the second column of X. Which of the following splits at the root node gives the smallest entropy: $x_1 > 2$, $x_1 > 4$, $x_2 > 3$, $x_2 > 6$? Motivate your answer.

SOLUȚIE 1

entropy: $H_h = H(\hat{p}_h) = \sum_{k=0}^{q-1} \hat{p}_{hk} \cdot \log \hat{p}_{hk}$ ← pt. fiecare clasă, q-nr. clase



a) $x_1 > 2$

pt. jumătatea stângă: $H_1 = -\hat{p}_{11} \cdot \log \hat{p}_{11} - \hat{p}_{10} \cdot \log \hat{p}_{10}$

$\hat{p}_{hk} = \frac{1}{|D_h|} \cdot \sum_{(x_i, y_i) \in D_h} 1 \cdot y(i) = k$

- câte elem. există în reg. h: 1
- câte elem. există în reg. h și fac parte din cls 1: 0

$\hat{p}_{11} = \frac{0}{1} = 0$

$\hat{p}_{10} = \frac{1}{1} = 1$

$H_1 = -0 \cdot \log 0 - 1 \cdot \log 1 = -1 \cdot 0 = 0$

pt. jumătatea <u>dreaptă</u>: $H_2 = -\hat{p_{20}} \cdot \log \hat{p_{20}} - \hat{p_{21}} \cdot \log \hat{p_{21}}$

$\hat{p_{20}} = \frac{1}{3}$, $\hat{p_{21}} = \frac{2}{3}$

$H_2 = -\frac{1}{3} \cdot \log \frac{1}{3} - \frac{2}{3} \cdot \log \frac{2}{3} = 0,16 + 0,11 = 0,27$

↓ lg ↓ lg

b) $x_1 > 4$

jum. <u>stângă</u>: $H_3 = -\hat{p_{30}} \cdot \log \hat{p_{30}} - \hat{p_{31}} \cdot \log \hat{p_{31}}$

$\hat{p_{30}} = \frac{2}{2} = 1$

$\hat{p_{31}} = 0 \Rightarrow H_3 = 0$

jum. <u>dreaptă</u>: $H_4 = -\hat{p_{40}} \cdot \log \hat{40} - \hat{p_{41}} \cdot \log \hat{p_{41}}$

$\hat{p_{40}} = 0$

$\hat{p_{41}} = 1 \Rightarrow H_4 = 0$

c) $x_2 > 3$

jum. <u>jos</u>:

$\hat{p_{50}} = \frac{1}{2}$

$\hat{p_{51}} = \frac{1}{2} \Rightarrow H_5 = -\frac{1}{2} \log \frac{1}{2} - \frac{1}{2} \cdot \log \frac{1}{2} = 0,3$

jum. <u>sus</u>:

$\hat{p_{60}} = \frac{1}{2}$

$\hat{p_{61}} = \frac{1}{2} \Rightarrow H_6 = 0,3$

d) $x_2 > 6$

jum. <u>jos</u>:

$\hat{p_{10}} = \frac{1}{3}$

$\hat{p_{11}} = \frac{2}{3} \Rightarrow H_1 = 0,27$

jum. <u>sus</u>:

$\hat{p_{20}} = 1$

$\hat{p_{21}} = 0 \Rightarrow H_2 = 0$

→ Concluzie

b) → entropia cea mai mică

⇒ $x_1 > 4$ cel mai bun split

SOLUȚIE 2

$$X=\begin{pmatrix} 6 & 3 \\ 2 & 7 \\ 9 & 6 \\ 4 & 2 \end{pmatrix} \qquad y=\begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \end{pmatrix}$$

x_1 x_2 features

Gini index:

$$G_h = \sum_{k=1}^{2} \hat{p}_{hk}(1-\hat{p}_{hk}) = \sum_{k=1}^{2} \hat{p}_{hk} - \sum_{k=1}^{2} \hat{p}_{hk}^2 = 1 - \sum_{k=1}^{2} \hat{p}_{hk}^2,$$

unde $\hat{p}_{hk}$ = procentul exemplelor din regiunea h care fac parte din clasa k

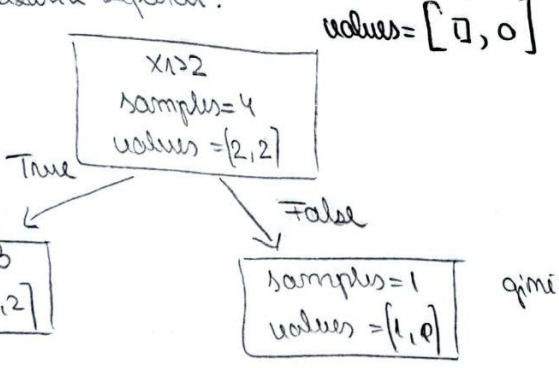
Gini index-ul la root va rămâne mereu același, deoarece toate datele ajung acolo.

samples = nr. de puncte = nr. de linii din X

$X_i > ?$
samples = 4
values = [2, 2]

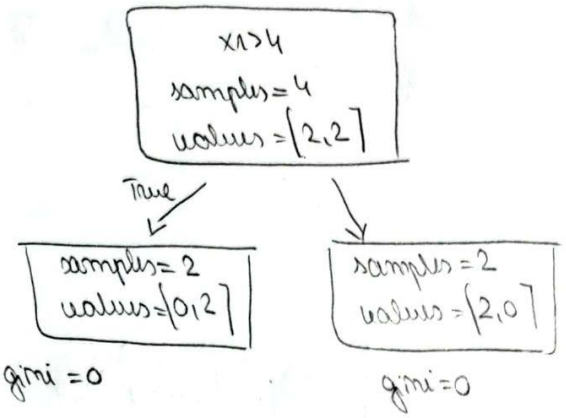
$$\Rightarrow \text{gini} = 1 - \left(\frac{1}{2}\right)^2 - \left(\frac{1}{2}\right)^2 = \frac{1}{2} = 0{,}5$$

Dacă întrebarea se referă la cel mai mic gini dintre cele ale nodurilor copiilor, studiem cazurile separat:



gini = 0 => nod pur

$$\text{gini} = 1 - \frac{1}{9} - \frac{4}{9} = \frac{4}{9}$$



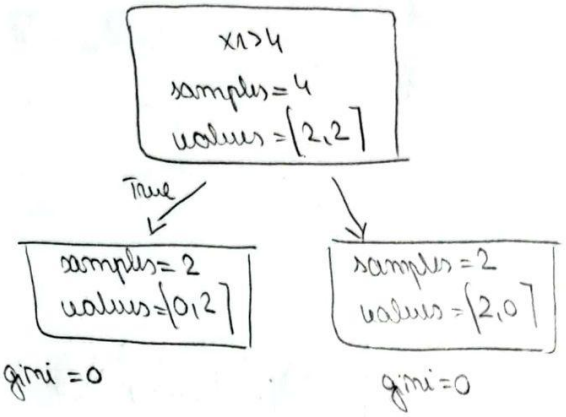
=) Nu are rost să încercăm și altele variante, deoarece x1>4 realizează clasificarea perfectă, fiecare nod frunză având doar instanțele unei singure clase.

5. Assume we have the design matrix X = [1 0; 0 1; -1 -1]. Use **PCA** to reduce the dimension from 2 to 1 to compute the first principal component.

Assume we have the design matrix

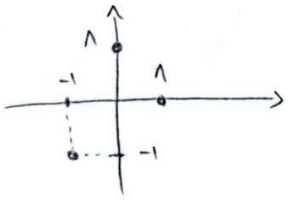
$$X = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ -1 & -1 \end{bmatrix}.$$

Use PCA to reduce the dimension from 2 to 1, i.e., to compute the first principal component.



=) Nu are rost să încercăm și altele variante, deoarece xAM realizează clasificarea perfectă, fiecare nod frunză având doar instanțele unei singure clase.

5) Assume we have the design matrix $X=\begin{bmatrix}1 & 0\\ 0 & 1\\ -1 & -1\end{bmatrix}$. Use PCA to reduce the dimension from 2 to 1, i.e., to compute the first principal component.



$\bar{x}_0=0$, $\bar{x}_1=0$

↑ x medie: suma coordonatelor pe coloana=0 |m

$\Rightarrow$ avem $\bar{x}=0$

$\bar{x}^T=[0\ \ 0]$

$$A:=\frac{1}{\sqrt{m}}\cdot X$$

$m=3 \Rightarrow A:=\frac{1}{\sqrt{3}}\begin{bmatrix}1 & 0\\ 0 & 1\\ -1 & -1\end{bmatrix}$

$$\Sigma=\frac{1}{m}\cdot\sum_{i=1}^{m}(x^{(i)}-\bar{x})(x^{(i)}-\bar{x})^T=\frac{1}{m}\sum_{i=1}^{m}x^{(i)}\cdot x^{(i)T}=\frac{1}{3}\cdot\sum_{i=1}^{3}(x^{(i)}-\bar{x})(x^{(i)}-\bar{x})^T$$

$$=\frac{1}{3}\left[\left(\begin{bmatrix}1\\0\end{bmatrix}-\begin{bmatrix}0\\0\end{bmatrix}\right)\left(\begin{bmatrix}1\\0\end{bmatrix}-\begin{bmatrix}0\\0\end{bmatrix}\right)^T+\left(\begin{bmatrix}0\\1\end{bmatrix}-\begin{bmatrix}0\\0\end{bmatrix}\right)\left(\begin{bmatrix}0\\1\end{bmatrix}-\begin{bmatrix}0\\0\end{bmatrix}\right)^T+\left(\begin{bmatrix}-1\\-1\end{bmatrix}-\begin{bmatrix}0\\0\end{bmatrix}\right)\left(\begin{bmatrix}-1\\-1\end{bmatrix}-\begin{bmatrix}0\\0\end{bmatrix}\right)^T\right]$$

$$=\frac{1}{3}\left(\begin{bmatrix}1\\0\end{bmatrix}\cdot[1\ \ 0]+\begin{bmatrix}0\\1\end{bmatrix}\cdot[0\ \ 1]+\begin{bmatrix}-1\\-1\end{bmatrix}\cdot[-1\ \ -1]\right)=\frac{1}{3}\left(\begin{bmatrix}1&0\\0&0\end{bmatrix}+\begin{bmatrix}0&0\\0&1\end{bmatrix}+\begin{bmatrix}1&1\\1&1\end{bmatrix}\right)$$

$$=\frac{1}{3}\cdot\begin{bmatrix}2&1\\ \end{bmatrix}$$

sau

când $\bar{x}=0 \Rightarrow \Sigma = \frac{1}{m} \cdot X^T \cdot X$

$A^T \cdot A = \Sigma$

$\det(\Sigma - \lambda \cdot I_2)=0$

$$\left| \begin{pmatrix} \frac{2}{3} & \frac{1}{3} \\ \frac{1}{3} & \frac{2}{3} \end{pmatrix} - \begin{pmatrix} \lambda & 0 \\ 0 & \lambda \end{pmatrix} \right| = \begin{vmatrix} \frac{2}{3}-\lambda & \frac{1}{3} \\ \frac{1}{3} & \frac{2}{3}-\lambda \end{vmatrix} = \left(\frac{2}{3}-\lambda\right)^2 - \left(\frac{1}{3}\right)^2 = \left(\frac{2}{3}-\lambda\right)^2 - \frac{1}{9} \Rightarrow$$

$$\Rightarrow \left(\frac{2}{3}-\lambda\right)^2 - \frac{1}{9}=0 \Rightarrow \frac{4}{9} - \frac{4}{3}\lambda + \lambda^2 - \frac{1}{9}=0 \;|\cdot 9 \Rightarrow 4 - 12\lambda + 9\lambda^2 - 1 = 0$$

$$\Rightarrow 9\lambda^2 - 12\lambda + 3 = 0 \Rightarrow 3\lambda^2 - 4\lambda + 1 = 0 \Rightarrow (\lambda-1)(3\lambda-1)=0$$

$$\Rightarrow \begin{cases} \lambda_1 = 1 \\ \lambda_2 = \frac{1}{3} \end{cases}$$ se sortează descrescător

Păstrăm doar λ cât dim avem nevoie = 1 (zice în enunț "from 2 to i") $\boxed{\lambda=1}$

$$\Rightarrow \text{calc. } \Sigma - \lambda_1 \cdot I = \begin{pmatrix} \frac{2}{3}-1 & \frac{1}{3} \\ \frac{1}{3} & \frac{2}{3}-1 \end{pmatrix} = \begin{pmatrix} -\frac{1}{3} & \frac{1}{3} \\ \frac{1}{3} & -\frac{1}{3} \end{pmatrix}$$

$$\begin{pmatrix} -\frac{1}{3} & \frac{1}{3} \\ \frac{1}{3} & -\frac{1}{3} \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \Rightarrow -\frac{1}{3}v_1 + \frac{1}{3}v_2 = 0 \Rightarrow v_1 = v_2 \Rightarrow \boxed{v_1=1} \text{ Ales}$$

$\tilde{w_1} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$, $w_1 = \begin{pmatrix} v_1 \\ v_2 \end{pmatrix}$ $\quad y = \frac{v_2}{v_1} x$

$|\tilde{w_1}| = \sqrt{1^2+1^2} = \sqrt{2}$

v_1 v_2

$$w_1 = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{pmatrix}$$

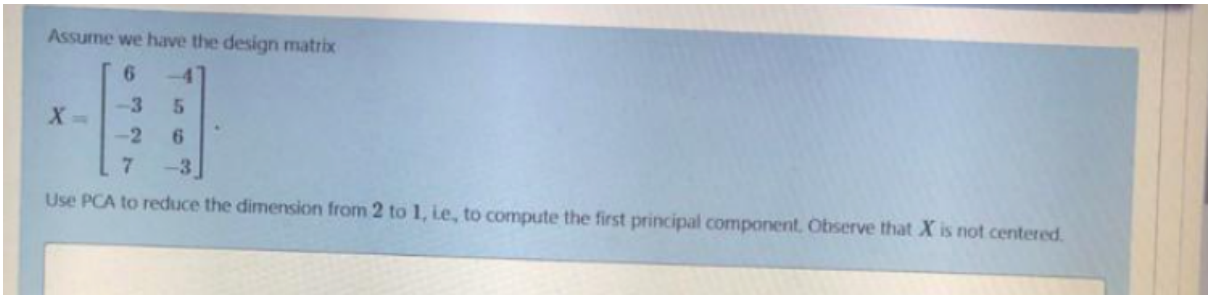
$|\tilde{w_1}|$ $\tilde{w_1}$

prima comp principală

$$\boxed{z = w^T \cdot x} \Rightarrow z^T = w^T \cdot x^T = \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & -1 \\ 0 & 1 & -1 \end{pmatrix} = \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & -\frac{2}{\sqrt{2}} \end{pmatrix}$$

$$w = \begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{pmatrix} \qquad \Rightarrow z = x' = \begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \\ -\sqrt{2} \end{pmatrix}$$

6. Assume we have the design matrix X = [6 -4; -3 5; -2 6; 7 -3]. Use **PCA** to reduce the dimension from 2 to 1 to compute the first principal component. Observe that X is not centered.



6) Assume we have the design matrix $X = \begin{bmatrix} 6 & -4 \\ -3 & 5 \\ -2 & 6 \\ 7 & -3 \end{bmatrix}$. Use PCA to reduce the dimension from 2 to 1 to compute the first principal component. Observe that X is not centered.

$\bar{x} = [8 \;\; 4] \,/\, 4 = [2 \;\; 1]$

$$\Sigma = \frac{1}{4}\left[\left(\begin{bmatrix}6\\-4\end{bmatrix}-\begin{bmatrix}2\\1\end{bmatrix}\right)\cdot\left(\begin{bmatrix}6\\-4\end{bmatrix}-\begin{bmatrix}2\\1\end{bmatrix}\right)^T + \left(\begin{bmatrix}-3\\5\end{bmatrix}-\begin{bmatrix}2\\1\end{bmatrix}\right)\cdot\left(\begin{bmatrix}-3\\5\end{bmatrix}-\begin{bmatrix}2\\1\end{bmatrix}\right)^T + \left(\begin{bmatrix}-2\\6\end{bmatrix}-\begin{bmatrix}2\\1\end{bmatrix}\right)\left(\begin{bmatrix}-2\\6\end{bmatrix}-\begin{bmatrix}2\\1\end{bmatrix}\right)^T + \left(\begin{bmatrix}7\\-3\end{bmatrix}-\begin{bmatrix}2\\1\end{bmatrix}\right)\left(\begin{bmatrix}7\\-3\end{bmatrix}-\begin{bmatrix}2\\1\end{bmatrix}\right)^T\right] =$$

$$= \frac{1}{4}\left(\begin{bmatrix}4 & 16\\16 & 64\end{bmatrix} + \begin{bmatrix}121 & -11\\-11 & 1\end{bmatrix} + \begin{bmatrix}100 & -20\\-20 & 4\end{bmatrix} + \begin{bmatrix}1 & 7\\7 & 49\end{bmatrix}\right) = \frac{1}{4}\begin{bmatrix}82 & -80\\-80 & 82\end{bmatrix}$$

pt. verificare: $\tilde{x} = x - \bar{x} = \begin{bmatrix} 4 & -5 \\ -5 & 4 \\ -4 & 5 \\ 5 & -4 \end{bmatrix}$ matricea centrată $= \begin{bmatrix} \frac{41}{2} & -20 \\ -20 & \frac{41}{2} \end{bmatrix}$

$\Sigma = \frac{1}{m} \cdot \tilde{x}^T \cdot \tilde{x}$

$\tilde{x} = [0 \;\; 0]^T$

$$\det(\Sigma - \lambda \cdot I_2) = 0 \Leftrightarrow \left| \begin{bmatrix} \frac{41}{2} & -40 \\ -40 & \frac{41}{2} \end{bmatrix} - \begin{bmatrix} \lambda & 0 \\ 0 & \lambda \end{bmatrix} \right| = 0 \Leftrightarrow \begin{vmatrix} \frac{41}{2} - \lambda & -40 \\ -40 & \frac{41}{2} - \lambda \end{vmatrix} = 0$$

$$\Leftrightarrow \left(\frac{41}{2} - \lambda\right)^2 + 40^2 = 0 \Leftrightarrow \frac{41^2}{4} + \lambda^2 - 41\lambda + 40^2 = 0 \;|\cdot 4 \Leftrightarrow 4 \cdot 41^2 + 4\lambda^2 - 41 \cdot 4\lambda + 40^2 \cdot 4 = 0$$

$$\mapsto (41-40)(41+40) = 81$$

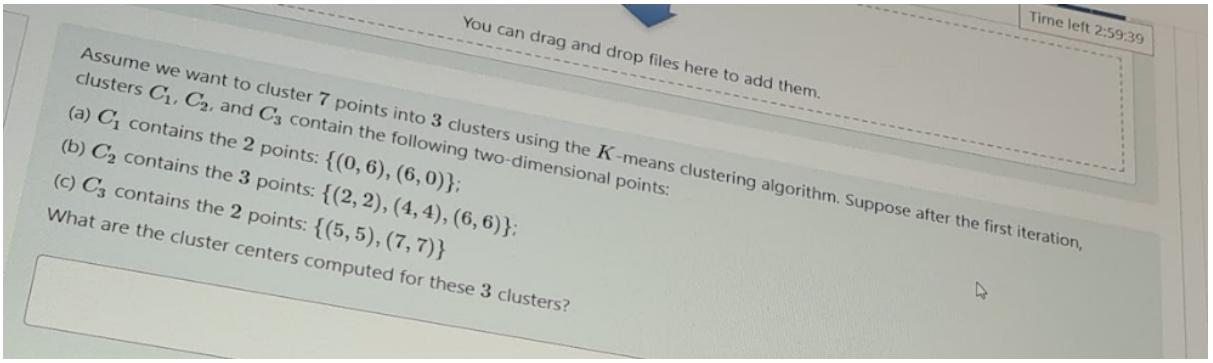
$$4\lambda^2 - 164\lambda + 4(41^2 - 40^2) = 0 \Leftrightarrow 4\lambda^2 - 164\lambda + 4 \cdot 81 = 0 \;|:4$$

$$\Leftrightarrow \lambda^2 - 41\lambda + 81 = 0 \Rightarrow \begin{cases} \lambda_1 = \frac{41 + \sqrt{1357}}{2} \\ \lambda_2 = \frac{41 - \sqrt{1357}}{2} \end{cases}$$

(sortate descrescător)

7. Assume we want to cluster 7 points into 3 clusters using the **K-means clustering algorithm**. Suppose after the first iteration, clusters C1, C2 and C3 contain the following two-dimensional points:
    a. C1 contains the 2 points: {(0, 6), (6, 0)}
    b. C2 contains the 3 points: {(2, 2), (4, 4), (6, 6)}
    c. C3 contains the 2 points: {(5, 5), (7, 7)}

    What are the cluster centers computed for these 3 clusters.



7) Assume we want to cluster 7 points into 3 clusters using the K-means clustering algorithm. Suppose after the first iteration, clusters C_1, C_2 and C_3 contain the following two-dimensional points:

a) C_1 contains the 2 points: $\{(0,6), (6,0)\}$

b) C_2 contains the 3 points: $\{(2,2), (4,4), (6,6)\}$

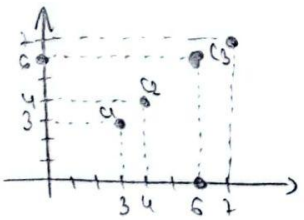
c) C_3 contains the 2 points: $\{(5,5), (7,7)\}$

What are the cluster centers computed for these 3 clusters.

C_1: $\frac{0+6}{2} = 3 \Rightarrow C_1(3,3)$

C_2: $\frac{2+4+6}{3} = \frac{12}{3} = 4 \Rightarrow C_2(4,4)$

C_3: $\frac{5+7}{2} = 6 \Rightarrow C_3(6,6)$



pt. încă o iterație

$C_3 = (7,7), (6,6), (5,5) \Rightarrow C_3: \frac{7+6+5}{3} = 6 \Rightarrow C_3(6,6)$

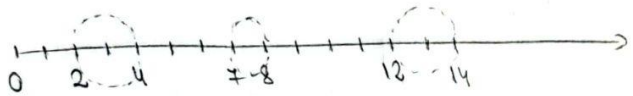
$C_2 = (4,4) \Rightarrow C_2(4,4)$

$C_1 = (2,2), (6,0), (0,6) \Rightarrow C_1: \frac{2+6}{3} = \frac{8}{3} \Rightarrow C_1(2,67; 2,67)$

8. Consider a dataset containing six one-dimensional points: {2, 4, 7, 8, 12, 14}. After three iterations of hierarchical clustering using **Euclidean distance between points**, we get the 3 clusters: C1 = {2, 4}, C2 = {7, 8} and C3 = {12, 14}. What is the distance between clusters C1 and C2 using single linkage? What is the distance between clusters C1 and C2 using complete linkage?

Consider a dataset containing six one-dimensional points: $\{2, 4, 7, 8, 12, 14\}$. After three iterations of hierarchical clustering using Euclidean distance between points, we get the 3 clusters: $C_1 = \{2, 4\}$, $C_2 = \{7, 8\}$, and $C_3 = \{12, 14\}$. What is the distance between clusters C_1 and C_2 using single linkage? What is the distance between clusters C_1 and C_2 using complete linkage?

8) Consider a dataset containing six one-dimensional points: {2,4,7,8,12,14}. After three iterations of hierarchical clustering using Euclidean distance between points, we get the 3 clusters: $C_1=\{2,4\}$, $C_2=\{7,8\}$ and $C_3=\{12,14\}$. What is the distance between clusters C_1 and C_2 using single linkage? What is the distance between clusters C_1 and C_2 using complete linkage?



$C_1 \to C_2$ single link

$d(C_1,C_2)_{single} = 7-4=3$

$d(C_1,C_2)_{complete} = 8-2=6$

$d(C_1,C_2)_{average} = \frac{3+4+5+6}{4} = 4,5$

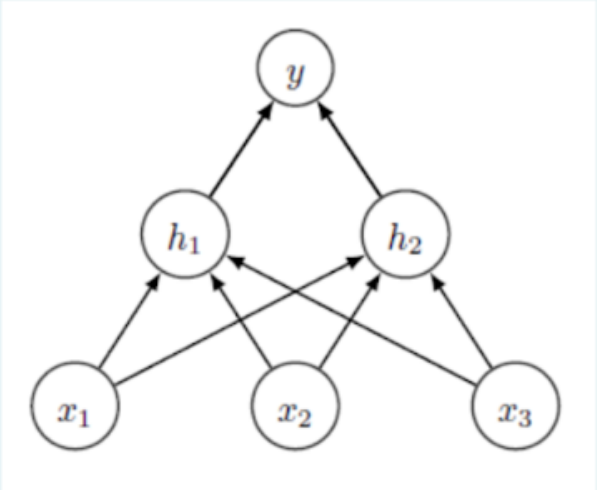
Încă o iterație fol. single

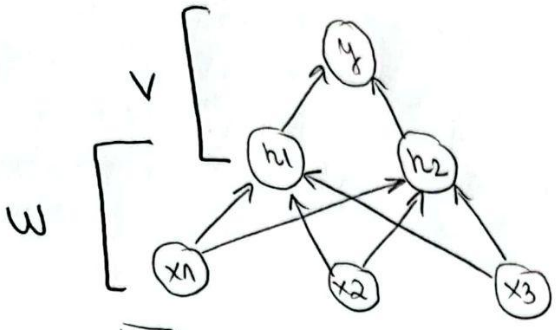
$C_1+C_2 \Rightarrow$ cluster (distanța cea mai mică)

9. The following graph shows the structure of a **simple neural network with a single hidden layer**.

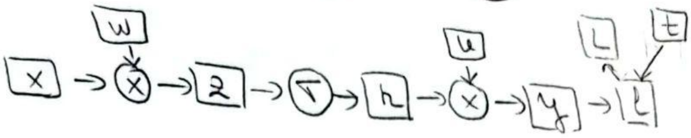
The following graph shows the structure of a simple neural network with a single hidden layer. The input layer consists of three dimensions $x = (x_1, x_2, x_3)$. The hidden layer includes two units $h = (h_1, h_2)$. The output layer includes one unit y. We ignore bias terms for simplicity. We use sigmoid $\sigma(z) = \frac{1}{1+e^{-z}}$ as activation function for the hidden layer and the identity for the output layer. Moreover, denote by $l(y, t) = \frac{1}{2}(y - t)^2$ the loss function. Here, t is the target value for the output unit y. Denote by W and V weight matrices connecting input and hidden layer, and hidden layer and output, respectively. Compute symbolically $\frac{\partial l}{\partial W}$. Numerical application:

$$W = \begin{bmatrix} 1 & 0 & 1 \\ 1 & -1 & 0 \end{bmatrix}, V = \begin{bmatrix} 0 & 1 \end{bmatrix}, x = (1, 2, 1), t = 1.$$





! nu avem regularizare ca in curs



$H = Wx$

$y = V\,\sigma(H) = V\,\sigma(Wx)$

$L(y,t) = \frac{1}{2}(y-t)^2 = \frac{1}{2}(V\cdot\sigma(Wx)-t)^2 \Rightarrow \frac{\partial L}{\partial V} = \frac{1}{2}\cdot 2\cdot(V\sigma(Wx)-t)\cdot\sigma(Wx)$

$= (V\sigma(Wx)-t)\cdot\sigma(Wx)$

Numeric: $H = Wx = \begin{pmatrix}1 & 0 & 1\\ 1 & -1 & 0\end{pmatrix}\cdot\begin{pmatrix}1\\2\\1\end{pmatrix} = \begin{pmatrix}2\\-1\end{pmatrix}$

$\sigma(H) = \begin{pmatrix}\frac{1}{1+e^{-2}}\\ \frac{1}{1+e}\end{pmatrix}$

$\Rightarrow y = (0\ \ 1)\cdot\begin{pmatrix}\frac{1}{1+e^{-2}}\\ \frac{1}{1+e}\end{pmatrix} = \frac{1}{1+e} = [0{,}269]$

$V^T = \begin{bmatrix}0\\1\end{bmatrix}$

$\frac{\partial L}{\partial W} = \left(V^T\cdot([0{,}269]-1)\odot\sigma'(z)\right)\cdot x^T$

$= \left(\begin{bmatrix}0\\1\end{bmatrix}\cdot[-0{,}731]\odot\sigma'(z)\right)\cdot x^T$

$\sigma'(z) = \frac{e^{-z}}{(1+e^{-z})^2} = \begin{bmatrix}0{,}105\\0{,}197\end{bmatrix}$

$\frac{\partial L}{\partial W} = \left(\begin{bmatrix}0\\-0{,}731\end{bmatrix}\odot\begin{pmatrix}0{,}105\\0{,}197\end{pmatrix}\right)\cdot[1\ 2\ 1] = \begin{bmatrix}0\\-0{,}144\end{bmatrix}[1\ 2\ 1] = \begin{bmatrix}0 & 0 & 0\\-0{,}144 & -0{,}288 & -0{,}144\end{bmatrix}$

10. Assume we have the following **CNN classifier**:
    INPUT, CONV-9-32, POOL-2, CONV-5-64, POOL-2, FC-3
    For each layer, calculate the number of weights, number of biases and the size of the associated feature maps. The size of INPUT is 128x128x3. The notation follows the convention:
    - CONV-K-N denoted a convolutional layer with N filters, each of them of size KxK, padding and stride parameters are always 0 and 1.
    - POOL-K indicates a KxK pooling layer with stride K and padding 0.
    - FC-N stands for a fully-connected layer with N neurons.

> Assume we have the following CNN classifier:
> INPUT, CONV-9-32, POOL-2, CONV-5-64, POOL-2, CONV-5-64, POOL-2, FC-3.
> For each layer, calculate the number of weights, number of biases, and the size of the associated feature maps. The size of INPUT is $128 \times 128 \times 3$. The notation follows the convention:
> - CONV-K-N denotes a convolutional layer with N filters, each of them of size K×K, padding and stride parameters are always 0 and 1, respectively.
> - POOL-K indicates a K×K pooling layer with stride K and padding 0.
> - FC-N stands for a fully-connected layer with N neurons.

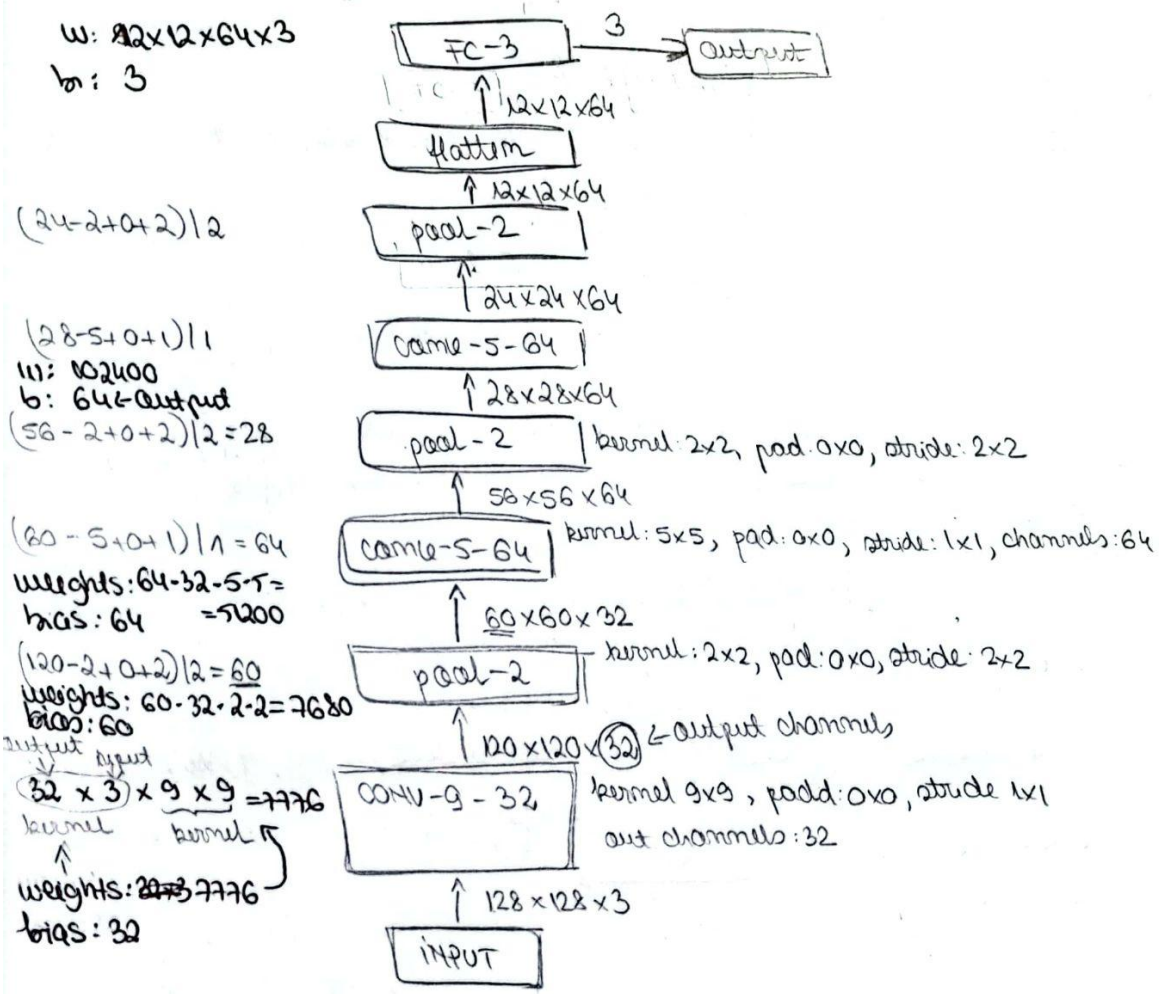
CNN classifier

10) Assume we have the following CNN classifier:

INPUT, CONV-9-32, POOL-2, CONV-5-64, POOL-2, FC-3.

For each layer, calculate the number of weights, number of bias and the size of the associated feature maps. The size of INPUT is 128×128×3. The notation follows the convention:

- CONV-K-N denotes a convolutional layer with N filters, each of them of size K×K, padding and stride parameters are always 0 and 1
- POOL-K indicates a K×K pooling layer with stride K and padding 0
- FC-N stands for a fully-connected layer with N neurons.

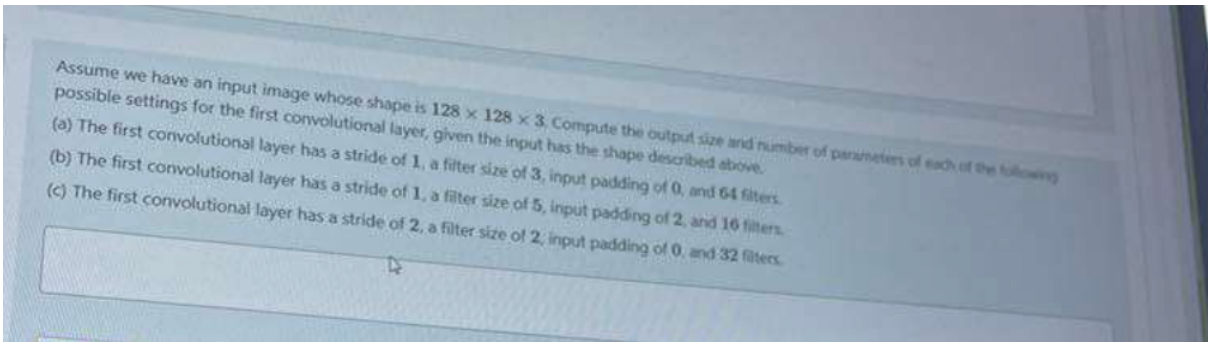


! Pt. pad — nr. liniii output — latimea output

$$\left\lceil (n_h - K_h + P_h + S_h)/S_h \right\rceil \times \left\lceil (n_w - K_w + P_w + S_w)/S_w \right\rceil$$

$$(128 - 9 + 0 + 1)/1 = 120 \times (128 - 9 + 0 + 1)/1 = 120)$$

11. Assume we have an input **image** whose shape is 128x128x3. Compute the output size and number of parameters of each of the following possible settings for the first convolutional layer, given the input has the shape described above:
    a. The first convolutional layer has a stride of 1, a filter size of 3, input padding 0 and 64 filters.
    b. The first convolutional layer has a stride of 1, a filter size of 5, input padding 2 and 16 filters.
    c. The first convolutional layer has a stride of 2, a filter size of 2, input padding 0 and 32 filters.



M) Assume we have an input image whose shape is 128 × 128 × 3. Compute the output size and number of parameters of each of the following possible settings for the first convolutional layer, given the input has the shape described above.

a) The first convolutional layer has a stride of 1, a filter size of 3, input padding of 0 and 64 filters.

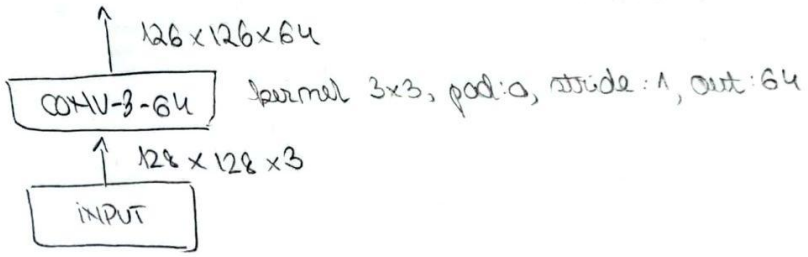
b) The first convolutional layer has a stride of 1, a filter size of 5, input padding of 2 and 16 filters.

c) The first convolutional layer has a stride of 2, a filter size of 2, input padding of 0 and 32 filters.

a)

$(128 - 3 + 0 + 1)/1$
$= 126$

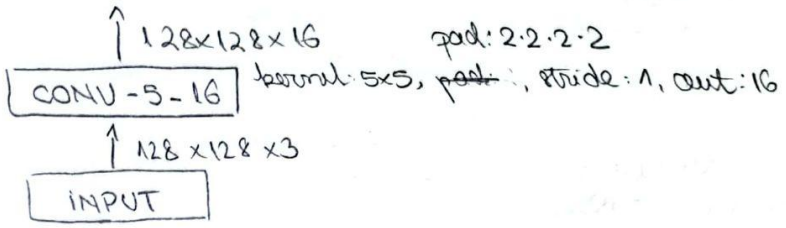
$W = (3 \times 3) \times 64 \times 3$
$b = 64$



b)

$W = (5 \times 5) \times 16 \times 3$
$= 1200$

$b = 16$

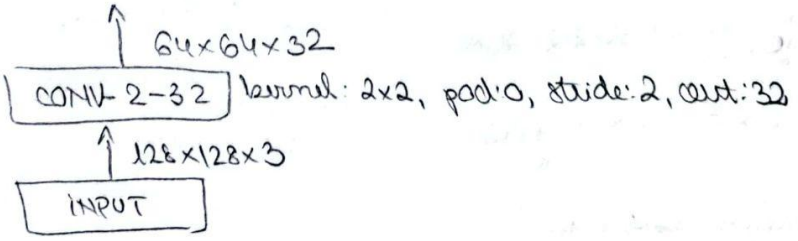


c)

$(128 - 2 + 0 + 2)/2$
$= 64$

$W = 2 \times 2 \times 32 \times 3 = 384$

$b = 32$



12. Compute symbolically the total number of parameters of a **GRU layer**, knowing that the number of examples is n, the number of input is d and the number of hidden units is h. Numerical application n=2, d=3, h=4.

Compute symbolically the total number of parameters of a GRU layer, knowing that the number of examples is n, the number of inputs is d, and the number of hidden units is h. Numerical application: $n = 2, d = 3, h = 4$.

12) Compute symbolically the total number of parameters of a GRU layer, knowing that the number of examples is n, the number of input is d and the number of hidden units is h.
Numerical application: $n=2, d=3, h=4$

$z_t = \sigma(x_t W_{xz} + H_{t-1} W_{hz} + b_z)$

$R_t = \sigma(X_t \cdot W_{xh} + H_{t-1} \cdot W_{hr} + b_r)$

$W_{xr}, W_{xz} \in \mathbb{R}^{d \times h} \Rightarrow 3 \cdot 4 = 12$
$+12 = 24$

$W_{hr}, W_{hz} \in \mathbb{R}^{h \times h} \Rightarrow 4 \cdot 4 = 16$
$+16 = 32$

$b_r, b_z \in \mathbb{R}^{1 \times h} \Rightarrow 4+4 = 8$

$\Rightarrow 24+32+8 = 64$

$\tilde{H}_t = \tanh(X_t \cdot W_{xh} + (R_t \odot H_{t-1}) \cdot W_{hh} + b_h)$

$W_{xh} \in \mathbb{R}^{d \times h} = 12$
$W_{hh} \in \mathbb{R}^{h \times h} = 16$
$b_h \in \mathbb{R}^{1 \times h} = 4$

$16+12+4 = 32$

$64+32 = 96$

$2 \cdot dh + 2 \cdot hh + 2 \cdot h + dh + hh + h = 3dh + 3hh + 3h = 3(dh + hh + h) =$
$= 3h(d+h+1) = 3 \cdot 4 (3+4+1) = \underline{96}$

state - at initial H_0: $4(h) \Rightarrow 96+4 = 100$
output layer: $hd + d = 4 \cdot 3 + 3 = 15$
$\Rightarrow 115$

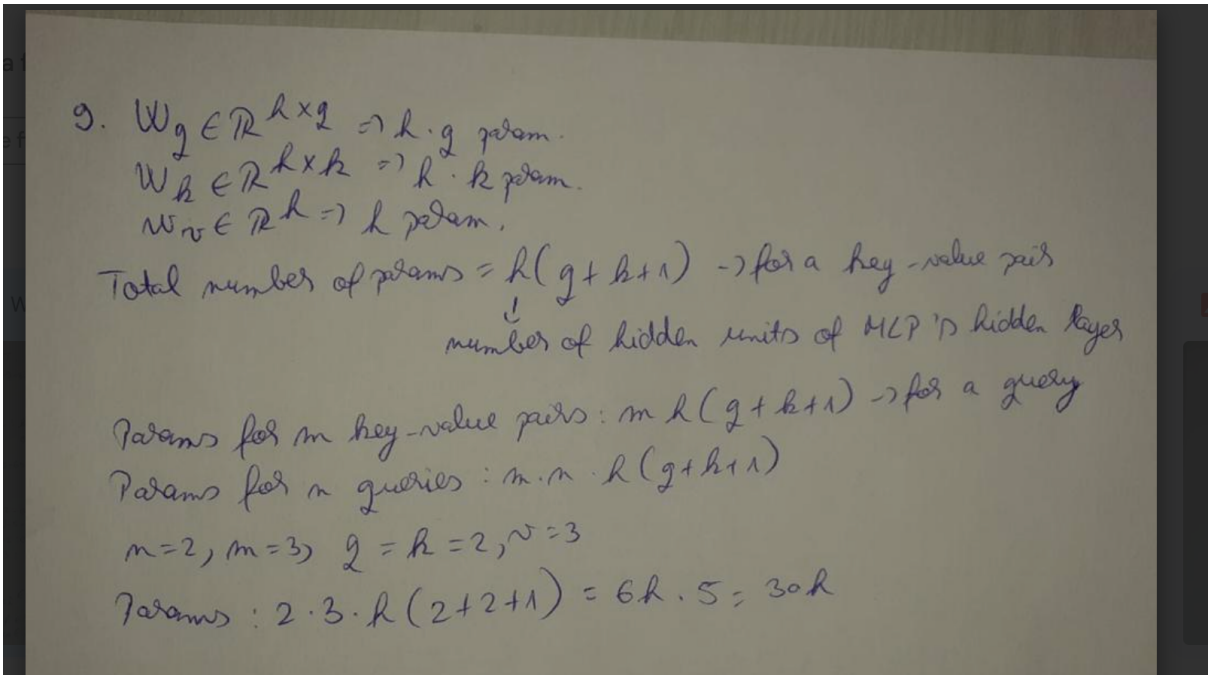
$O_t = H_t W_{hq} + b_q$

$W_{hq} \in \mathbb{R}^{h \times q}$, $q = d$

$b_q \in \mathbb{R}^{1 \times q}$

13. Compute the total number of parameters for **additive attention** for n queries and m key-value pairs, where queries are of length q, keys are of length k and values are of length v. Give the answer symbolically. Numerical application: n = 2, m = 3, q = k = 2, v = 3.

Compute the total number of parameters for additive attention for n queries and m key-value pairs, where queries are of length q, keys are of length k, and values are of length v. Give the answer symbolically. Numerical application:
$n = 2, m = 3, q = k = 2, v = 3.$



Coman Andrei-Alexandru

9.

n queries - length q

m key-value pairs

↓ length k ↓ length v

$n = 2$

$m = 3$

$q = k = 2$

$v = 3$

$$a(q, k) = w_v^T \tanh(W_q q + W_k k)$$

$$f = f(q, (k_1, v_1), \dots, (k_m, v_m)) = \sum_{i=1}^{m} \alpha(q, k_i) \cdot v_i, \quad v_i \in \mathbb{R}^v$$

$$\alpha(q, k_i) = \frac{\exp(a(q, k_i))}{\sum_{j=1}^{m} \exp(a(q, k_j))}$$

$W_q \in \mathbb{R}^{h \times q}$ -> $h \times q$ params

$W_k \in \mathbb{R}^{h \times k}$ -> $h \times k$ params

$w_v \in \mathbb{R}^h$ -> h params

=> Functia $a = hq + hk + h = h(q + k + 1)$ params

a

Functia $\alpha = \frac{\exp(a)}{\sum_{1}^{m} \exp(a)}$ => $a + m \cdot a = (m+1)(q+k+1) \cdot h$

$f = m \cdot h \cdot (m+1)(q+k+1)$ <- nr parametrii total pt q

=> 2 query => $2 \cdot f$

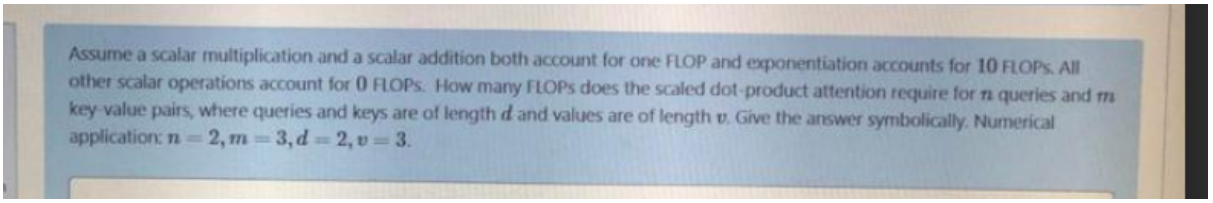
Cu valori:

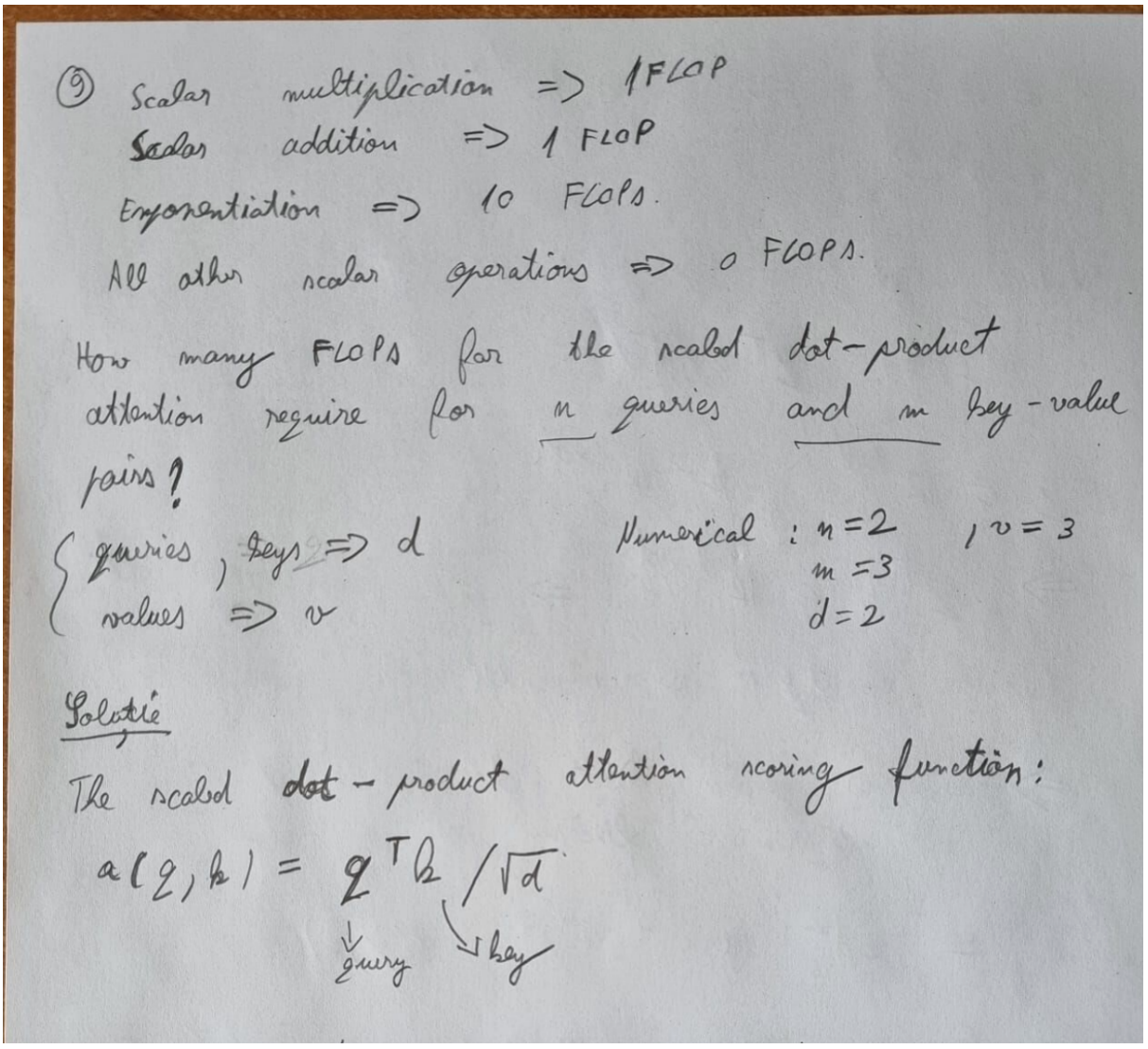
$2 \cdot 3 \cdot h(3+1)(2+2+1)$

$6 \cdot h \cdot 4 \cdot 5 = 120h$ - nr parametrii

h - nr. hidden layers

14. Assume a scalar multiplication and a scalar addition both account for one **FLOP** and exponentiation accounts for 10 FLOPS. All other scalar operations account for 0 FLOPS. How many FLOPS does the scaled dot-product attention require for n queries and m key-value pairs, where queries and keys are of length d and values are of length v. Give the answer symbolically. Numerical application: n = 2, m = 3, d = 2, v = 3.





$\Rightarrow$ $Q \in \mathbb{R}^{n \times d}$ queries, $K \in \mathbb{R}^{m \times d}$ keys,

$V \in \mathbb{R}^{m \times v}$ values

$\Rightarrow$ We have to compute FLOPS for:

$$\text{softmax}\left(\frac{Q^T K}{\sqrt{d}}\right) V \in \mathbb{R}^{n \times v} \quad (\in \mathbb{R}^{m \times v})$$

$Q \in \mathbb{R}^{n \times d} \Rightarrow Q^T \in \mathbb{R}^{d \times n}$

$K \in \mathbb{R}^{m \times d}$

Studiem pt un query și un key:

$a(q, k) = q^T k / \sqrt{d}$.

$q \in \mathbb{R}^{1 \times d}$, $k \in \mathbb{R}^{1 \times d}$

$\Rightarrow q^T k \in \mathbb{R}^{d \times d}$ $\Rightarrow$ $q^T k$ va genera d^2 FLOPS.

(fiecare element q se va înmulți cu fiecare el k)

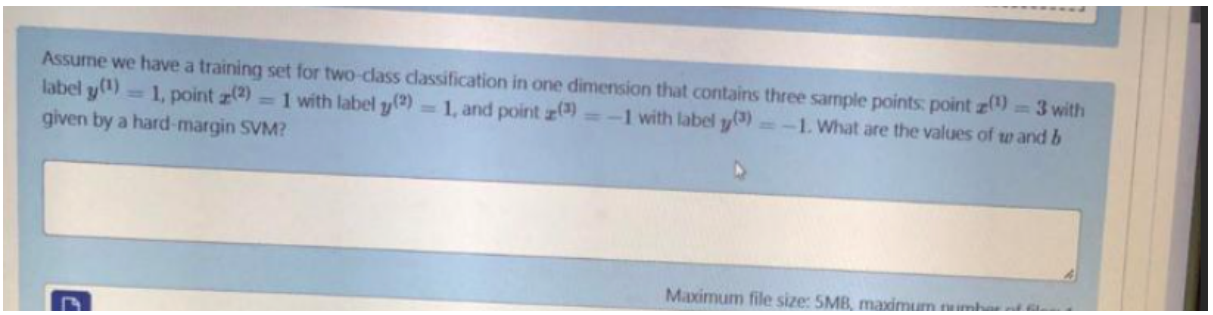
$\Rightarrow$ Se vor genera încă d^2 FLOPS prin multiplicarea fiecărui el cu $\frac{1}{\sqrt{d}}$.

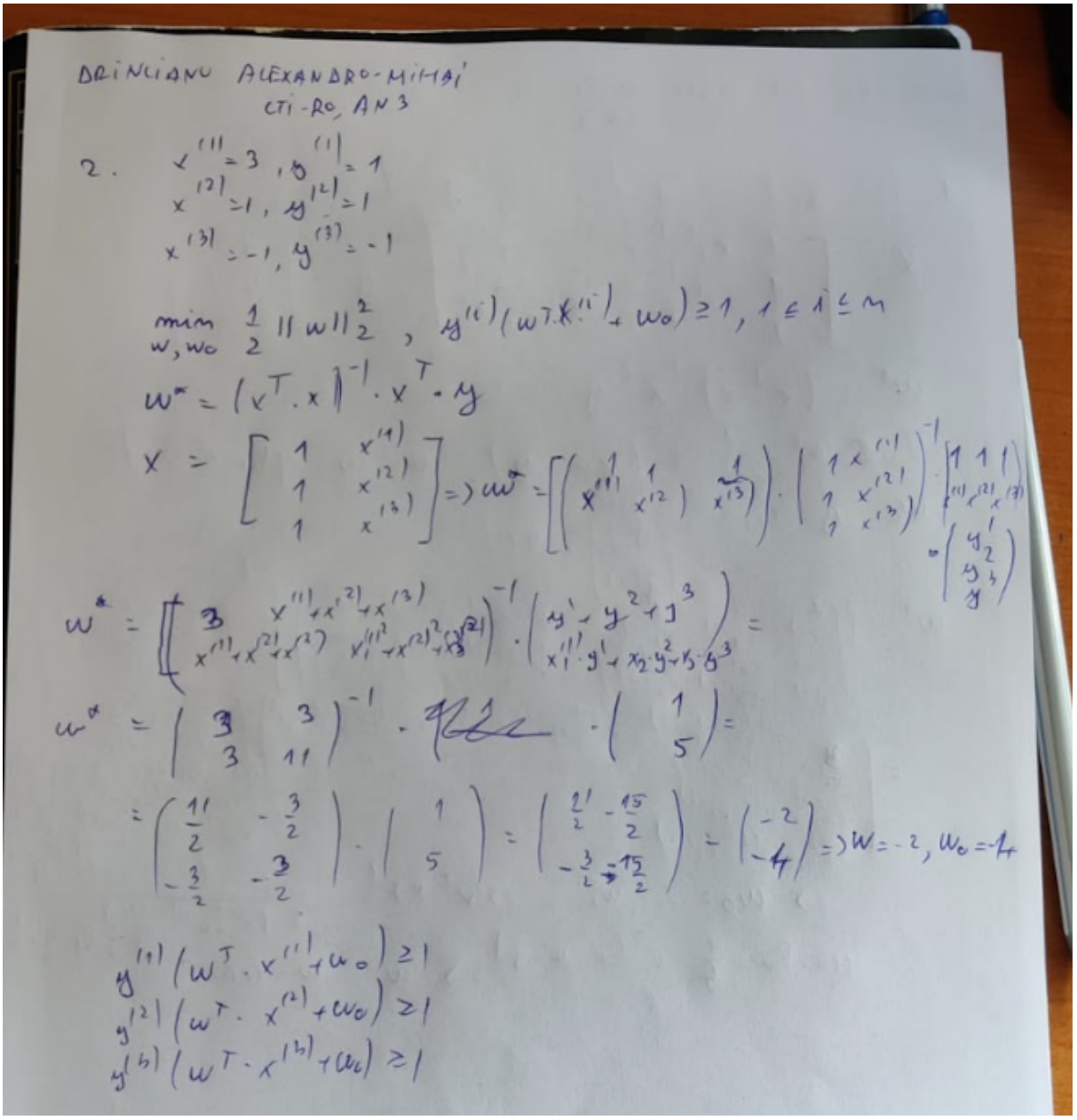
$\Rightarrow$ $2d^2$ generat pt. $a(q, k)$

$\Rightarrow$ Se vor genera $m \cdot n \cdot 2d^2$ Flop-uri prin aplicarea acestei funcții.

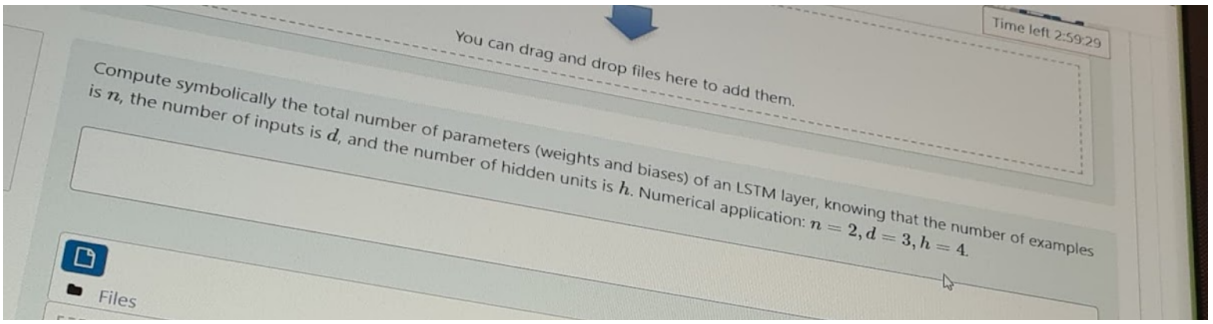
$\Rightarrow$ $m \cdot n \cdot 2d^2(d-1)$ ținând cont de înmulțirea matricelor

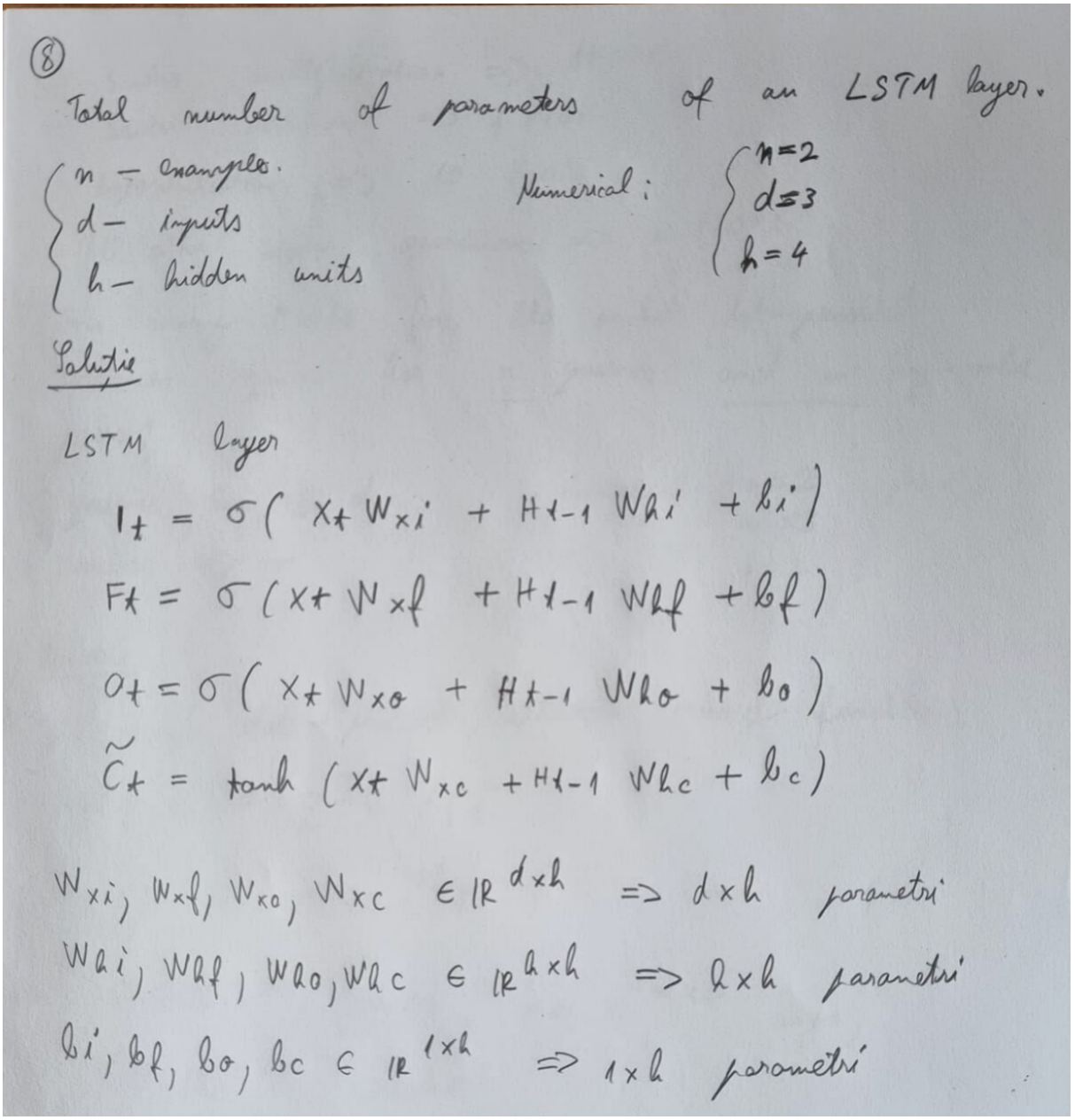
15. Assume we have a training set for **two-class classification in one dimension** that contains three sample points: point x(1) = 3 with label y(1) = 1, point x(2) = 1 with label y(2) = 1 and point x(3) = -1 with label y(3) = -1. What are the values of w and b given by a hard-margin SVM?





16. Compute symbolically the total number of parameters (weights and biases) of an **LSTM** layer, knowing that the number of examples is n, the number of inputs is d and the number of hidden units is h. Numerical application: n = 2, d = 3, h = 4.





$\Rightarrow$ Numărul total de parametri: $4(dh + h^2 + h)$

$\Rightarrow$ $\underline{4h(d+h+1)}$

Numeric: $4 \cdot 4(3+4+1) = 16 \cdot 8 = 128$ parametri

$b \in \mathbb{R}^{4 \times 1}$

$\Rightarrow 3 \cdot (3 \cdot 4) + 3 \cdot (4 \cdot 4) + 3 \cdot 4 + 4 \cdot 4 = 112$ weight parameters

$3 \cdot (1 \cdot 4) + 1 \cdot 4 = 16$ bias parameters