The sigmoid neuron:

A neuron with sigmoid activation σ

$\sigma(\cancel{w}\, W^T X)$ is usually interpreted as the probability that target $z=1$

Given a supervised dataset $X, \vec{y}$ $y_i \in \{0,1\}$ $\forall i$ sampled from $p_{data}(\vec{x}, y)$, we can try to learn target y via predicting

$y=1$ if $\sigma(W^T x) > .5$

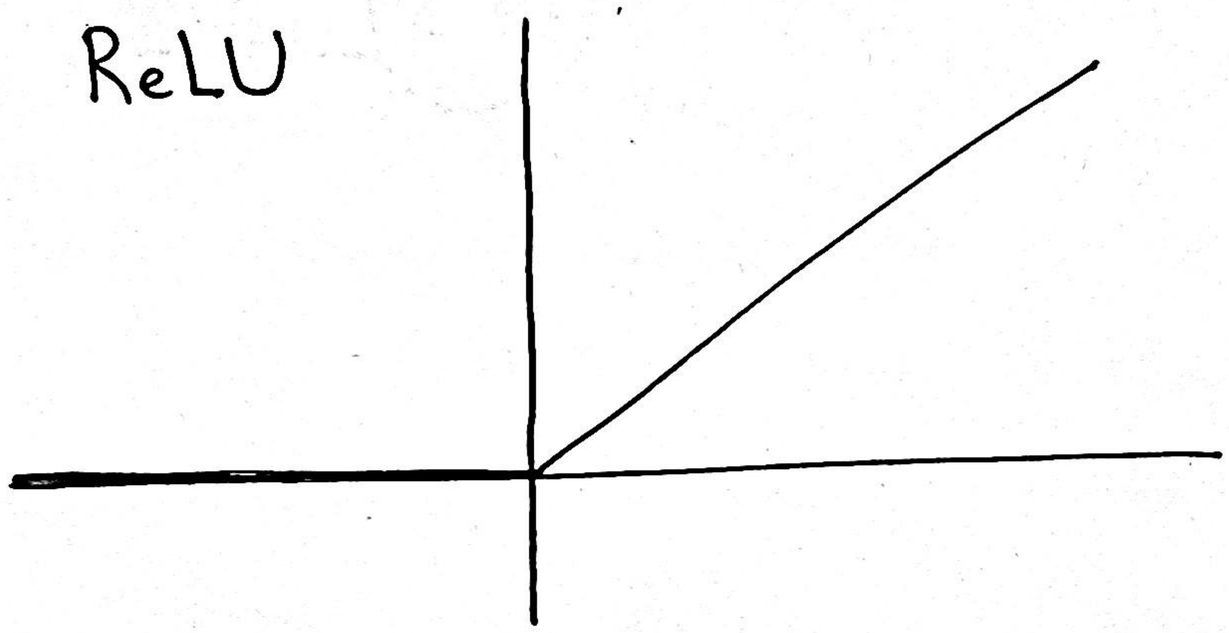
$y=0$ if $\sigma(W^T x) < .5$

etc... maximum likelyhood ... (recreate logistic regression)

Linear Neurons: Neurons with activation $\varphi(x) = x$

ReLU (rectified linear unit)

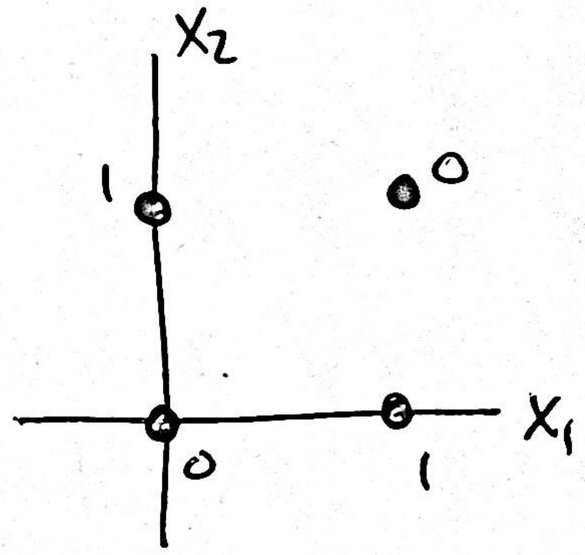
Activation function $ReLU(x) = x\,H(x) = \begin{cases} 0 & x<0 \\ x & x \geq 0 \end{cases}$



A Network of Neurons:

E.g. $f^*(\vec{x}) = x_1 \text{ xor } x_2$ $\quad \vec{x} \in \{0,1\}^2$

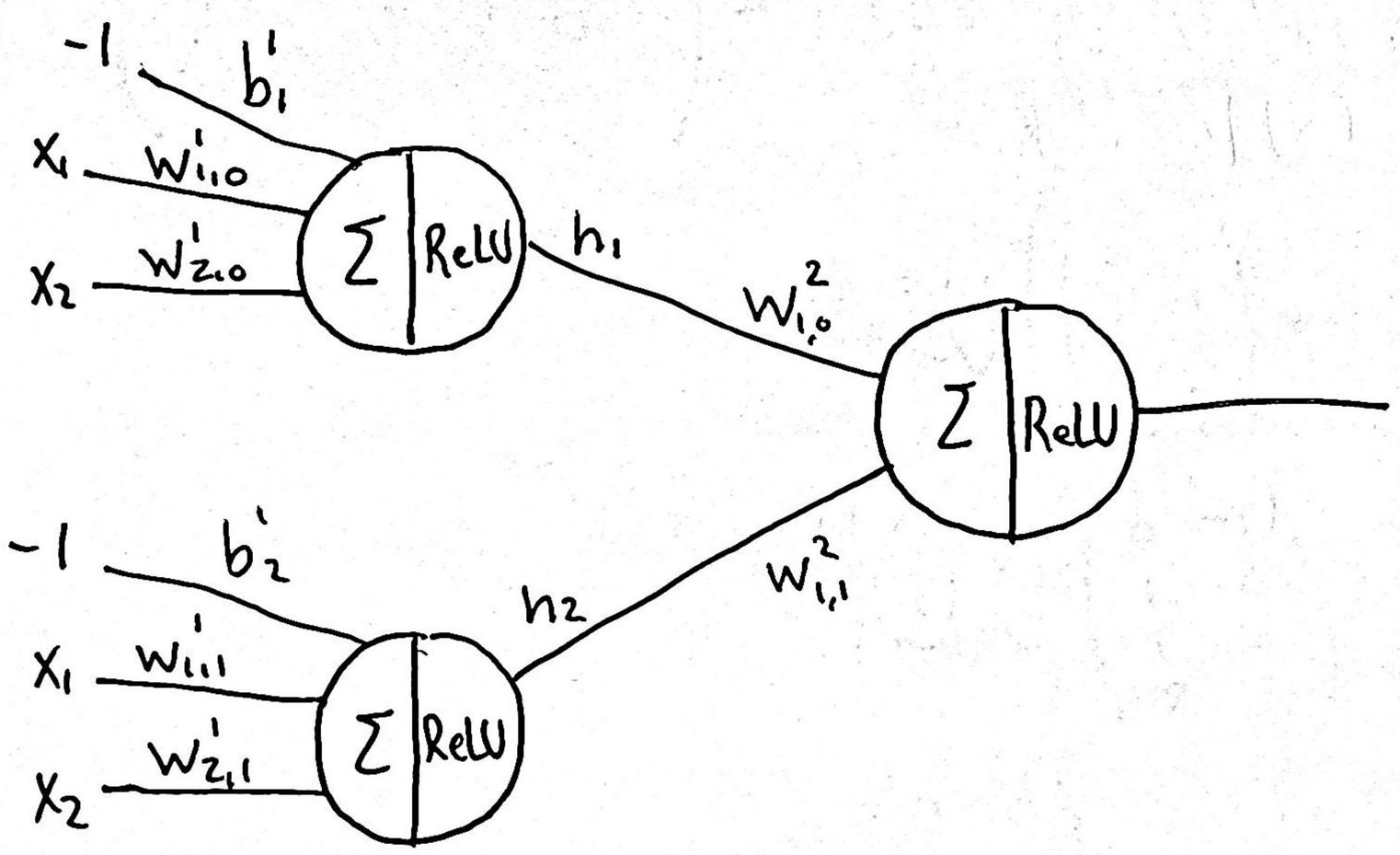
dataset $X = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{bmatrix}$ $\quad y = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$



Consider a model $f(\vec{x};\vec{\theta})$ and cost function

$$J(\theta) = \frac{1}{4} \sum_{\vec{x} \in X} \left(f^*(\vec{x}) - f(\vec{x};\theta) \right)$$

Idea



Idea: In $x_1 x_2$-plane the points are not separable by a line but maybe by transforming $x_1, x_2 \longrightarrow h_1, h_2$ the points can be separated by a line in the $h_1 h_2$ plane.

$$\begin{pmatrix} h_1 \\ h_2 \end{pmatrix} = \text{ReLU}\left(W^{(1)}\vec{x} + \vec{b}^{(1)}\right)$$

Try $W^{(1)} = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$ $\quad \vec{b}^{(1)} = \begin{pmatrix} 0 \\ -1 \end{pmatrix}$

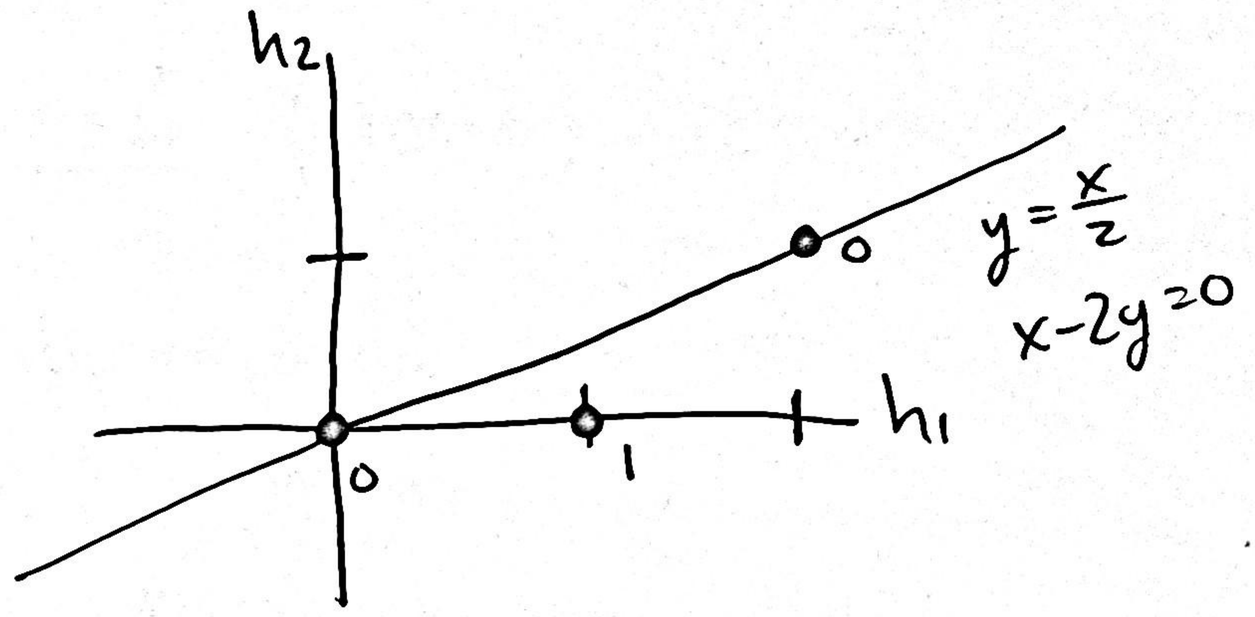
Apply to whole dataset

$$\begin{pmatrix} h_1 \\ h_2 \end{pmatrix} = \text{ReLU}\left(XW^{(1)} + T\vec{b}^{(1)T}\right) = \text{ReLU}\left(\begin{pmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{pmatrix}\begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} + \begin{pmatrix} 0 & -1 \\ 0 & -1 \\ 0 & -1 \\ 0 & -1 \end{pmatrix}\right)$$

$$= \text{ReLU}\left(\begin{pmatrix} 0 & 0 \\ 1 & 1 \\ 1 & 1 \\ 2 & 2 \end{pmatrix} + \begin{pmatrix} 0 & -1 \\ 0 & -1 \\ 0 & -1 \\ 0 & -1 \end{pmatrix}\right)$$

$$= \text{ReLU}\begin{pmatrix} 0 & -1 \\ 1 & 0 \\ 1 & 0 \\ 2 & 1 \end{pmatrix}$$

$$= \begin{pmatrix} 0 & 0 \\ 1 & 0 \\ 1 & 0 \\ 2 & 1 \end{pmatrix}$$

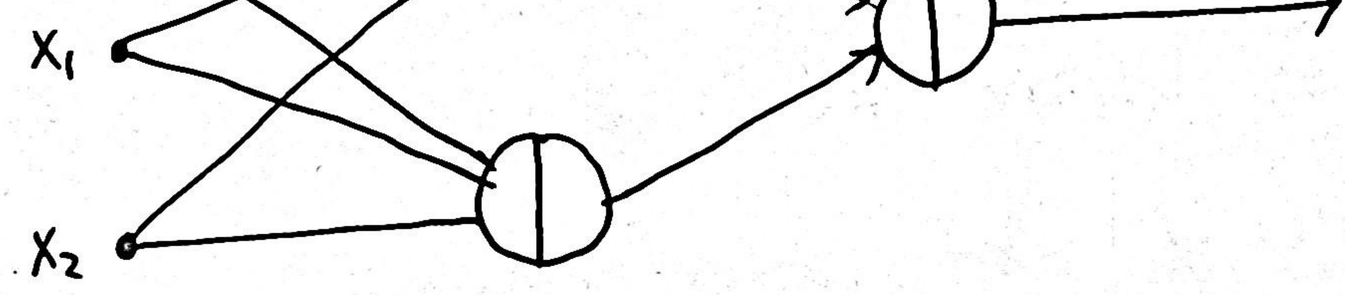


Final output is $W^{(2)}\begin{pmatrix} n_1 \\ n_2 \end{pmatrix} + b^{(2)}$

Let $W^{(2)} = \begin{pmatrix} 1 \\ -2 \end{pmatrix}$ $b^{(2)} = 0$

$$\text{ReLU}\left(\begin{pmatrix} 0 & 0 \\ 1 & 0 \\ 1 & 0 \\ 2 & 1 \end{pmatrix}\begin{pmatrix} 1 \\ -2 \end{pmatrix}\right) = \text{ReLU}\begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \end{pmatrix}$$

Better way to draw this:



Def: A <u>partially ordered set</u> (poset) is a pair $(P, \leq)$ where P is a set and $\leq$ is a relation on P s.t.

① $x \leq x \quad \forall x \in P$

② if $x \leq y$ and $y \leq x$ then $x = y \quad \forall x,y \in P$

③ if $x \leq y$ and $y \leq z$ then $x \leq z \quad \forall x,y,z \in P$

<u>Notation</u> write $x < y$ if $x \leq y$ and $x \neq y$

Call $(P, \leq)$ a <u>total order</u> if $\forall\, x,y \in P$

$x \leq y$ or $y \leq x$.

A <u>cover relation</u> is a pair (x,y) s.t.

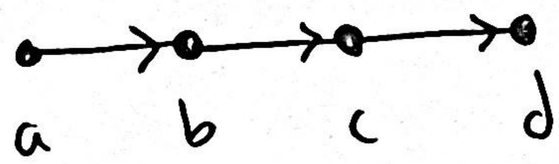
$x \leq y$ and $\nexists\, z$ with $x < z < y$.

The <u>Hasse diagram</u> of a finite poset $(P, \leq)$ consists of a vertex for each $x \in P$ and a directed edge from x to y iff $x \lessdot y$ drawn from left to right

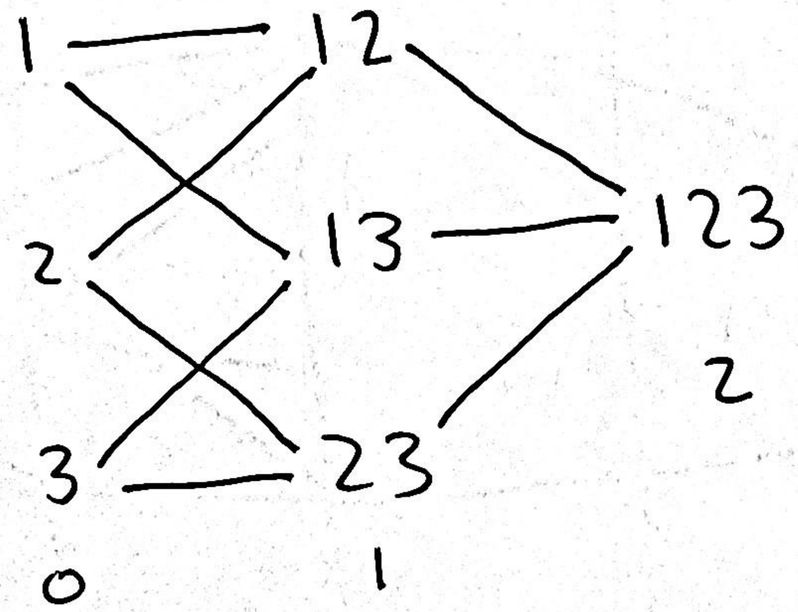
(so if $x \leq y$, draw x on the left of y)

Write this as $\mathcal{H}(P)$

E.g. $P = \{a, b, c, d\}$ $a \lessdot b \lessdot c \lessdot d$

$\mathcal{H}(P) =$ 

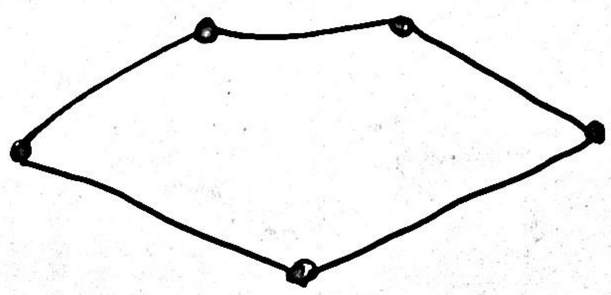
E.g. $P = \{S \subseteq \{1,2,3\} \mid S \neq \emptyset\}$ ordered by containment



Def: A poset is graded if $\exists\ f : P \to \mathbb{N}$ (rank function)

s.t.
① $x < y \implies f(x) < f(y)$ $\forall\ x, y \in P$
② $x \lessdot y \implies f(x) + 1 = f(y)$ $\forall\ x, y \in P$

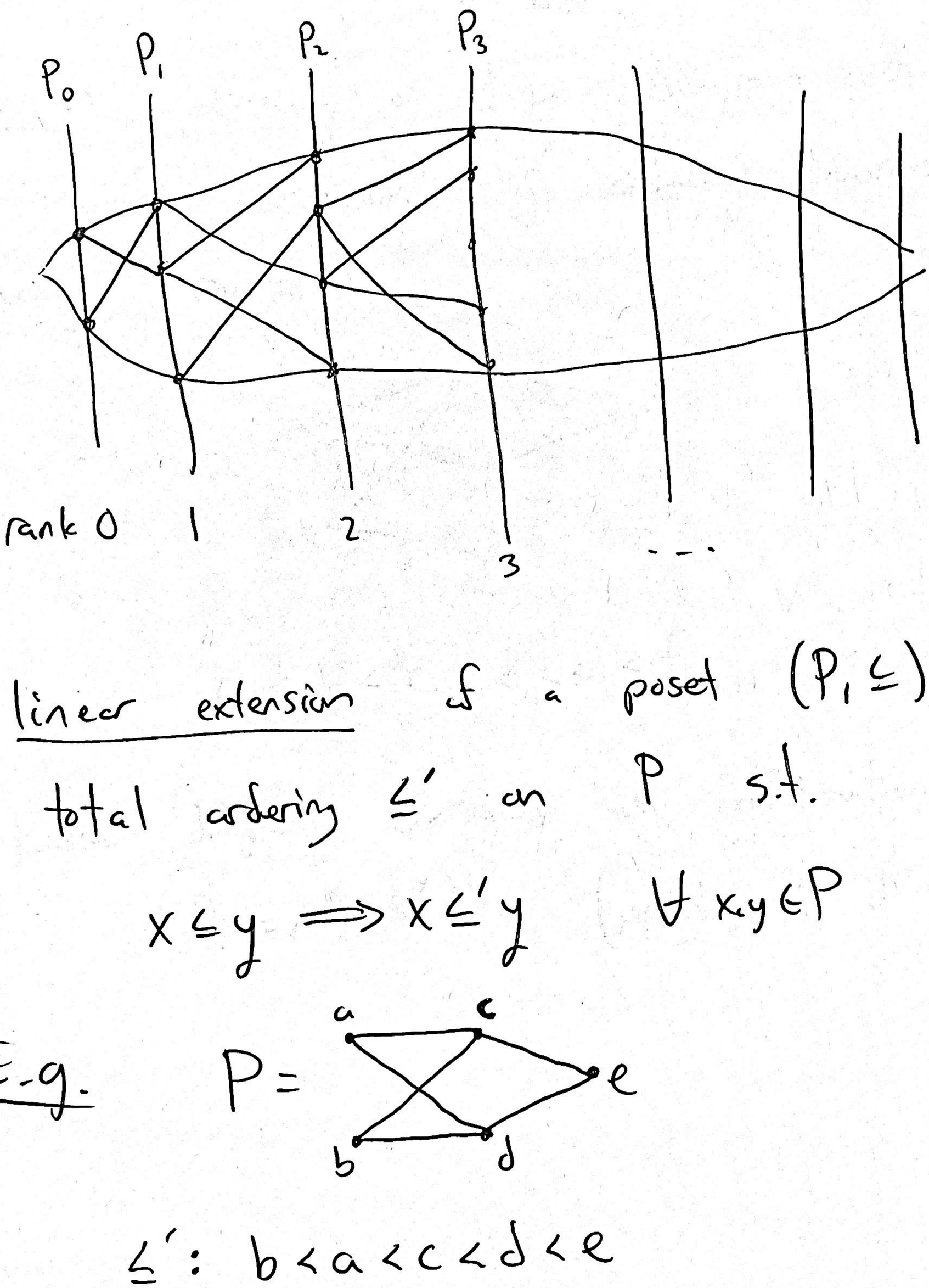
E.g. $\{S \subseteq \{1,2,3\} \mid S \neq \emptyset\}$ is graded

E.g.  not graded

i.e. in a graded poset there is a notion of "depth"

The rank decomposition of a graded poset $(P, \leq, \rho)$ is

$$P = \biguplus_{i \in \mathbb{N}} P_i \qquad P_i = \{x \in P \mid \rho(x) = i\}$$



A <u>linear extension</u> of a poset $(P, \leq)$ is a total ordering $\leq'$ on P s.t.

$$x \leq y \Rightarrow x \leq' y \qquad \forall\, x, y \in P$$

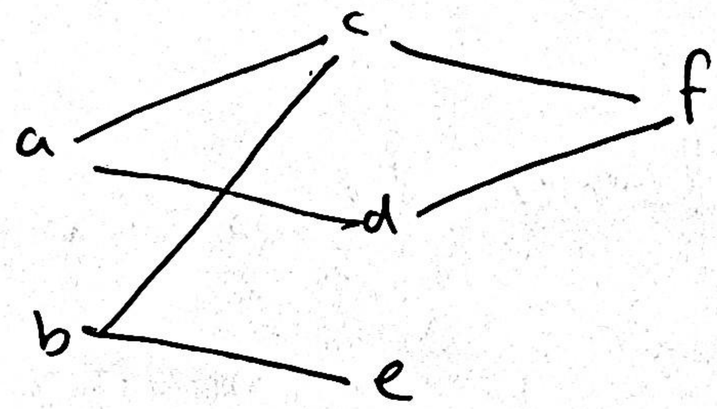
<u>E.g.</u> $P =$

$\leq'$: $b < a < c < d < e$

Lemma A linear extension of a graded poset is determined by a choice of total ordering of each of its ranks

$\therefore$ There are $\prod_{i \in \mathbb{N}} |P_i|!$

Convention Drawing a Hasse diagram determines a linear extension by ordering downwards along ranks



$a < b < c < d < e < f$

$\therefore$ Hasse diagram contains all information $(P_1 \leq, P, \leq')$ (up to rank of P_0)

Def: A **$\mathbb{N}$-graded vector space** is a vector space V together with a decomposition

$$V = \bigoplus_{i \in \mathbb{N}} V_i$$

(where V_i is a subspace of V)

E.g. $\mathbb{R}[x]$ is an $\mathbb{N}$-graded vector space where $(\mathbb{R}[x])_i = \{cx^i \mid c \in \mathbb{R}\}$

Def: Given a graded poset $(P, \leq, \rho, \leq')$ with linear extension $\leq'$, the associated graded vector space is

$$V_P^{\leq'} = \bigoplus_{i \in \mathbb{N}} V_P^{(i)}$$

$$V_P^{(i)} = \langle x \in P \mid \rho(x) = i \rangle$$

(real vector space with basis P_i)

basis P_i is an **ordered** basis with ordering specified by $\leq'$

Def: Given vector spaces V, V', an **affine map** $f: V \to V'$ is a map s.t.

$\forall \{x_1, \dots, x_n\} \subseteq V$ and $\forall \lambda_1, \dots, \lambda_n \in \mathbb{R}$ with $\sum_i \lambda_i = 1$

we have $f(\sum_i \lambda_i x_i) = \sum_i \lambda_i f(x_i)$

Lemma $f: V \to V'$ is affine iff $\exists$ a linear map $L: V \to V'$ and $\vec{b}_0 \in V'$ s.t.

$$f(v) = L(v) + \vec{b}_0 \qquad \forall\, v \in V$$

Proof: Exercise

Note: After picking bases for V, V' an affine map looks like

$$f(v) = Mv + \vec{b}$$

$M \in \mathbb{R}^{\dim V' \times \dim V}$

$\vec{b} \in \mathbb{R}^{\dim V'}$

Def Let $\mathcal{H} = (P, \leq, \rho, \leq')$ be a finite graded poset with linear extension $\leq'$, s.t. $\min\{\rho(x) \mid x \in P\} = 0$.

A <u>feedforward Neural Network</u> $\mathcal{N}$ with architecture $\mathcal{H}$ consists of

(1) a collection $F = (F^{(0)}, \ldots, F^{(D-1)})$ of affine maps where $D = \max\{\rho(x) \mid x \in P\}$

s.t. $F^{(i)} : V_P^{(i)} \to V_P^{(i+1)}$ for $0 \leq i \leq D-1$

and for $x \in V_P^{(i)}$, $y \in V_P^{(i+1)}$

$$\text{proj}_y F^{(i)}(x) = 0 \quad \text{unless} \quad x \lessdot y$$

(2) a collection of maps

$$G = (G^{(1)}, \ldots, G^{(D)}) \quad \text{where}$$

$$G_x^{(i)} : V_P^{(i)} \to V_P^{(i)}$$

Remark It is almost always the case that

$G_{\vec{x}}^{(i)}$ is diagonal so $G_{\vec{x}}^{(i)}(v_1, \dots, v_k) = (g_1^{(i)}(v_1), \dots g_k^{(i)}(v_k))$

for some functions $g_j^{(i)} : \mathbb{R} \to \mathbb{R}$

Also usually $g_1^{(i)} = g_2^{(i)} = \dots = g_k^{(i)}$

Terminology ① D is called the depth of $\mathcal{N}$

② $V_p^{(i)}$ is the i^{th} layer of $\mathcal{N}$

③ $V_p^{(1)}, \dots, V_p^{(D-1)}$ are called hidden layers

④ $V_p^{(D)}$ is the output layer

⑤ $V_p^{(0)}$ is the input layer

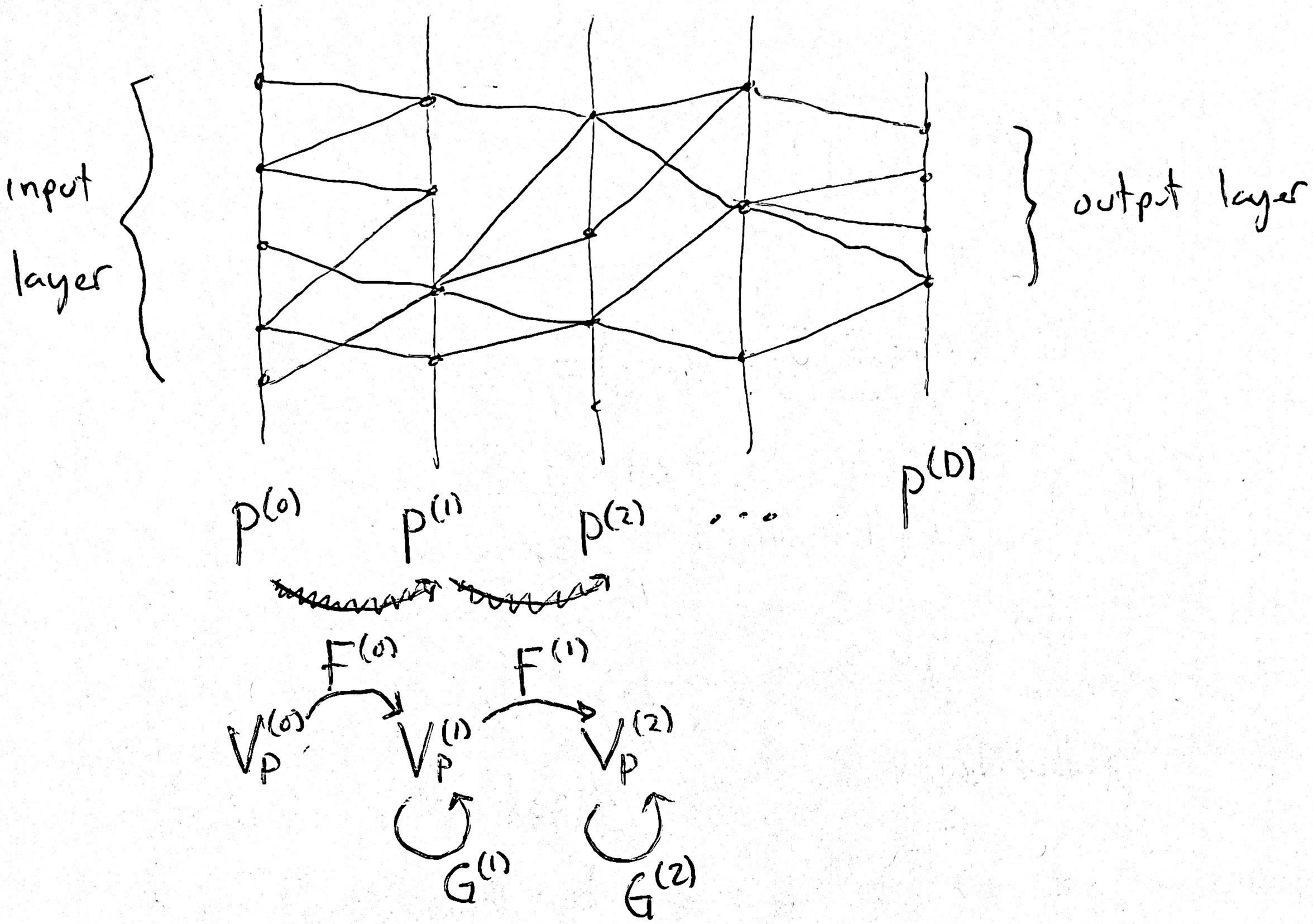
⑥ $G^{(i)}$ is the activation at the i^{th} layer

⑦ $g_j^{(i)}$ is the activation at the j^{th} neuron of the i^{th} layer

⑧ The matrix $M^{(i)}$ is the matrix of weights at layer i

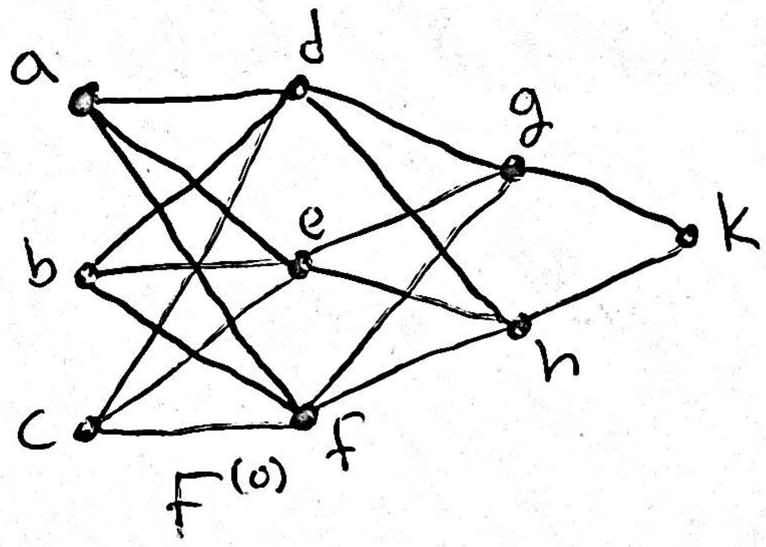
$\vec{b}^{(i)}$ is the bias at layer i

$\left(F_{\vec{x}}^{(i)} = M^{(i)}\vec{x} + \vec{b}^{(i)} \right)$



<u>E.g.</u> Consider the poset $P(n_0, \dots, n_D)$ which has n_i elements at rank i for $0 \le i \le D$ and $x \lessdot y$ iff $\rho(y) = \rho(x) + 1$

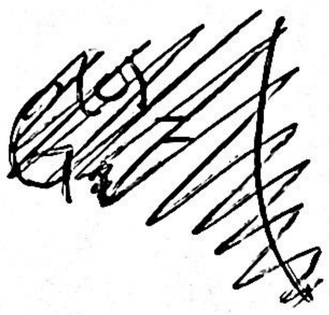
$P(3,3,2,1)$:



$$F^{(0)}_{\vec{x}} = M^{(0)}\vec{x} + \vec{b}^{(0)} \qquad M^{(0)} \in \mathbb{R}^{3\times 3} \qquad \vec{b}^{(0)} \in \mathbb{R}^3$$

$$F^{(1)}_{\vec{x}} = M^{(1)}\vec{x} + \vec{b}^{(1)} \qquad M^{(1)} \in \mathbb{R}^{2\times 3} \qquad \vec{b}^{(1)} \in \mathbb{R}^2$$

$$F^{(2)}_{\vec{x}} = M^{(2)}\vec{x} + \vec{b}^{(2)} \qquad M^{(2)} \in \mathbb{R}^{1\times 2} \qquad \vec{b}^{(2)} \in \mathbb{R}$$



$$G^{(1)}_{\vec{x}} = \left(g^{(1)}_{\mathbb{R}}(x_1), g^{(1)}_{\mathbb{R}}(x_2), g^{(1)}_{\mathbb{R}}(x_3)\right)$$

$$G^{(2)}_{\vec{x}} = \left(g^{(2)}_{\mathbb{R}}(x_1), g^{(2)}_{\mathbb{R}}(x_2)\right)$$

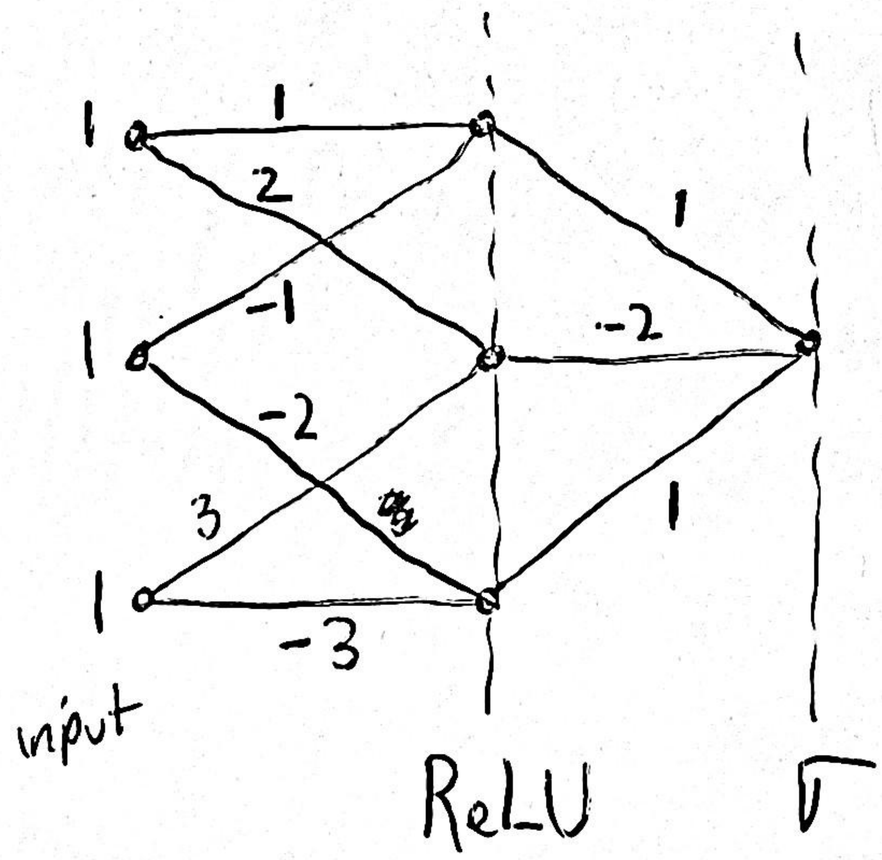
$$G^{(3)}_{\vec{x}} = \left(g^{(3)}(x)\right)$$

Def: Let $\mathcal{N} = (\mathcal{H}, F, G)$ be a neural network with depth D and architecture $\mathcal{H} = (P_i \subseteq_i P_i')$. The feedforward of $\vec{x} \in \mathbb{R}^{\dim V_P^{(0)}}$ is

$$\mathcal{N}\vec{x} = G_D F_{D-1} \cdots G_2 F_1 G_1 F_0 \vec{x} \in \mathbb{R}^{\dim V_P^{(D)}}$$

E.g.

$\mathcal{N} =$



$$\vec{b}^{(0)} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \qquad \vec{b}^{(1)} = 0$$

$$\mathcal{N}\begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} = G^{(2)}F^{(1)}G^{(1)}F^{(0)}\begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$$

$$F^{(0)}\begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & -1 & 0 \\ 2 & 0 & 3 \\ 0 & -2 & -3 \end{pmatrix}\begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} + \underbrace{\begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}}_{\vec{b}^{(0)}}$$