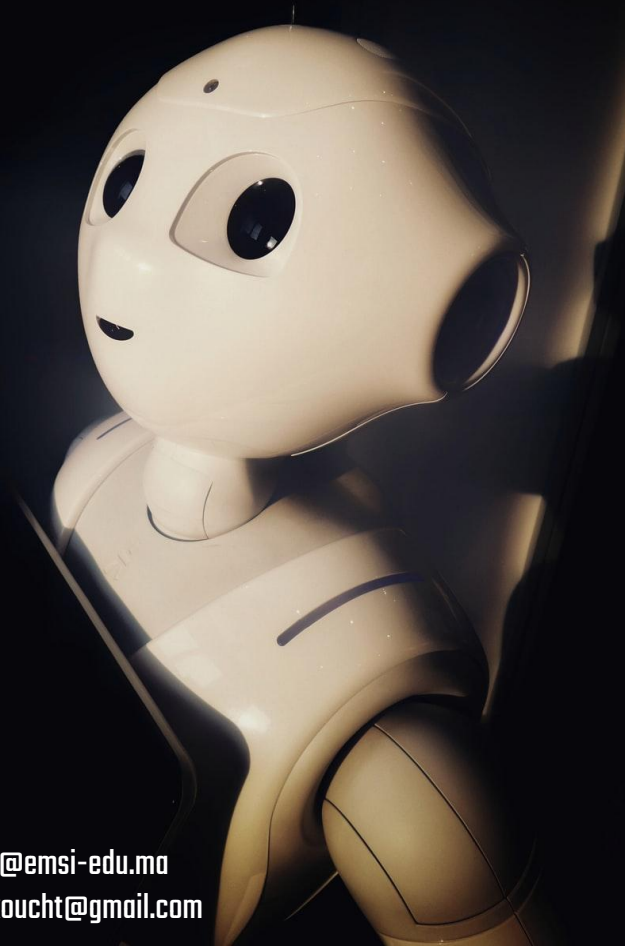


DEEP LEARNING: INTRODUCTION



Aissam Outchakoucht

a.outchakoucht@emsi-edu.ma
aissam.outchakoucht@gmail.com

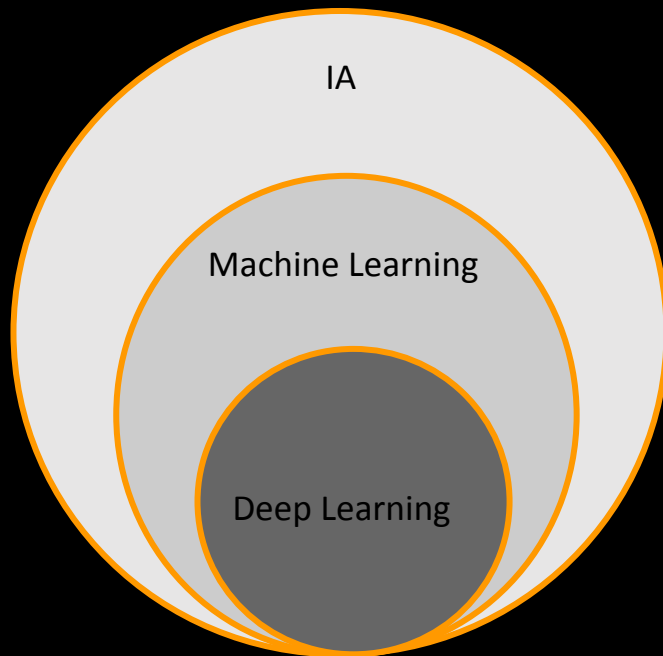


RECAP

IA est une technique qui permet à une machine d'imiter le comportement humain.

ML est une technique pour atteindre l'IA à travers des algorithmes entraînés avec des données.

DL est un type de ML inspiré par la structure du cerveau humain.

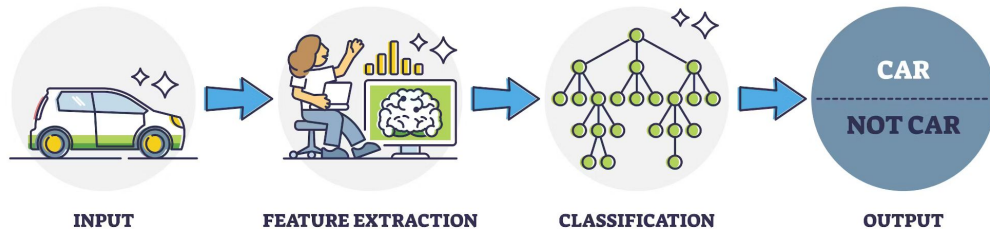


LIMITES DES ALGORITHMES DE ML

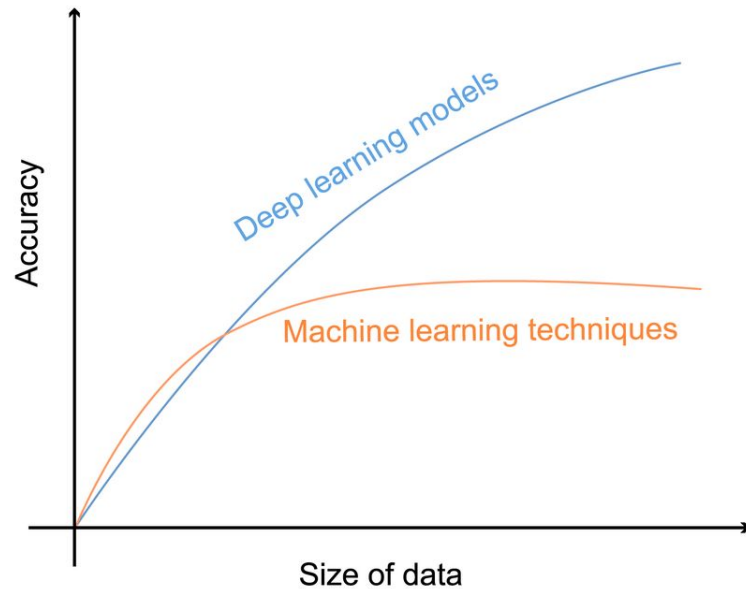
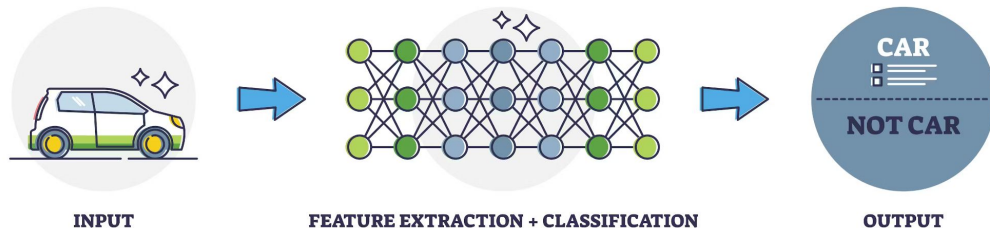
- **Ingénierie des caractéristiques** : De nombreux algorithmes dépendent de caractéristiques bien conçues.
- **Évolutivité** : Des algorithmes comme SVM et KNN ont du mal avec de grands ensembles de données.
- **Relations complexes** : Les relations non linéaires sont difficiles pour des modèles comme la Régression Linéaire/Logistique.
- **Haute dimensionnalité** : Le surapprentissage devient un problème lorsque la dimensionnalité augmente.
- **Sparsité des données** : De nombreux algorithmes sous-performent avec des données comme les textes ou les images.

DL, PK?

MACHINE LEARNING

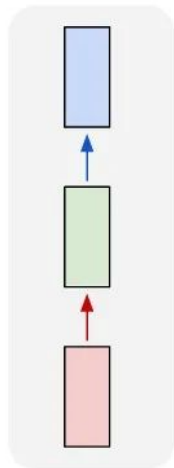


DEEP LEARNING

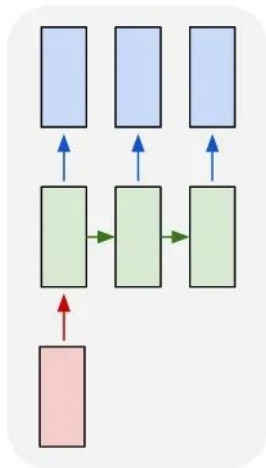


DL ?

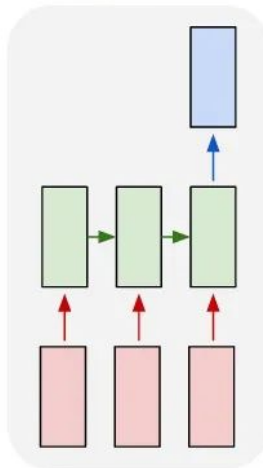
one to one



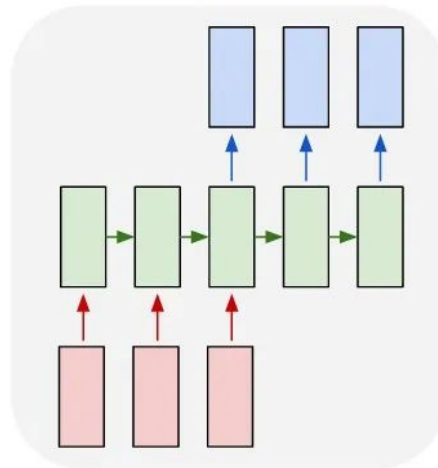
one to many



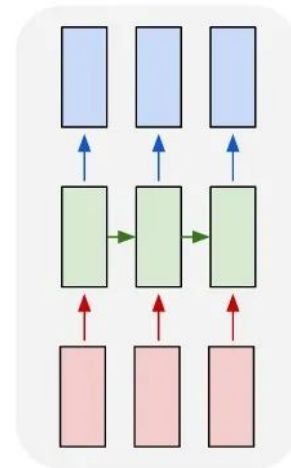
many to one



many to many



many to many



DL ?

one to one

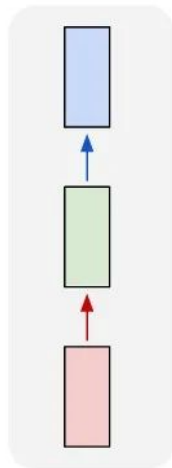


Image
Label

one to many

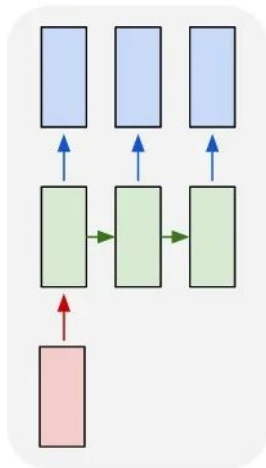
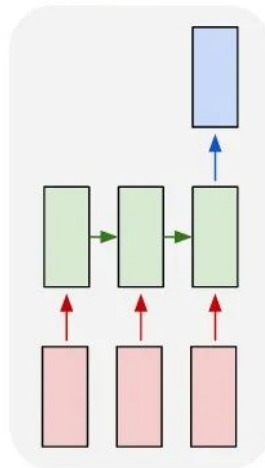


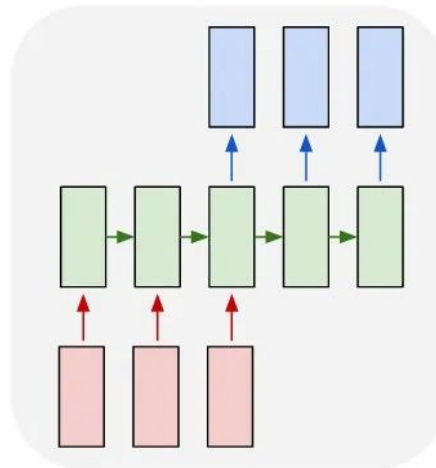
Image
Description

many to one



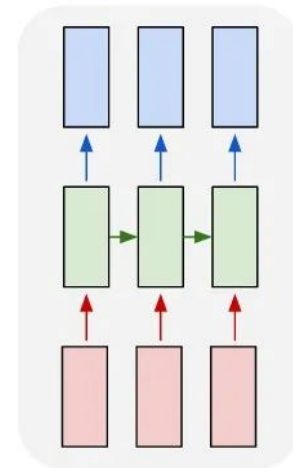
Sentiment
Analysis

many to many



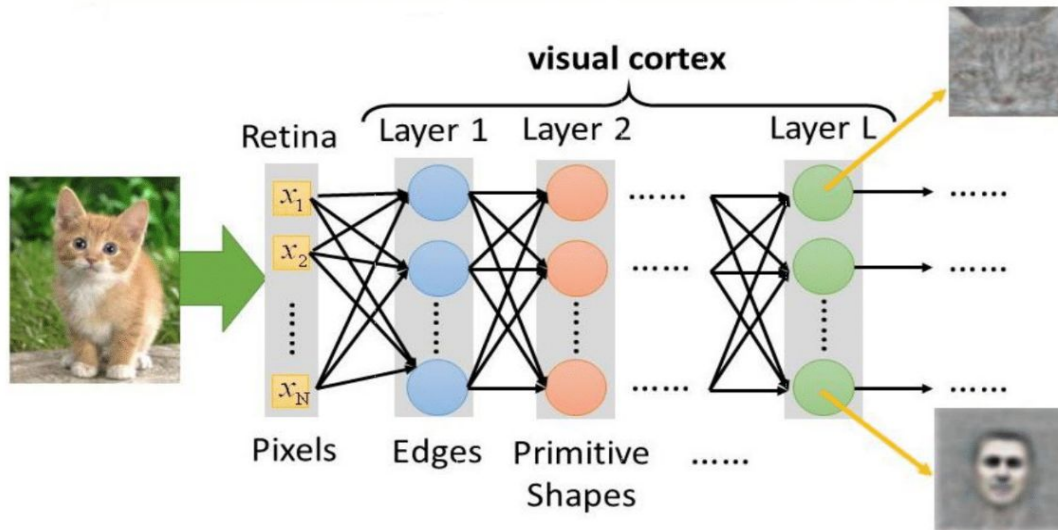
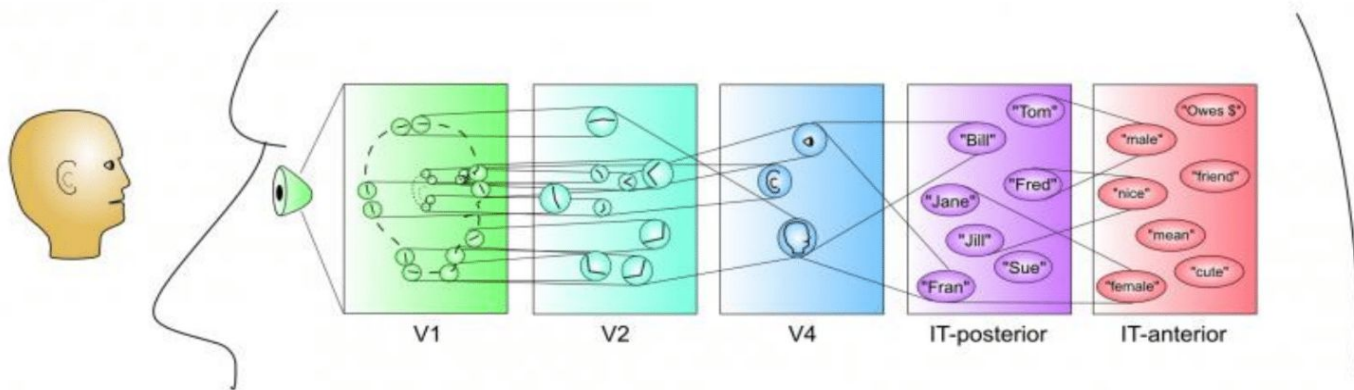
Sentence
Translation

many to many

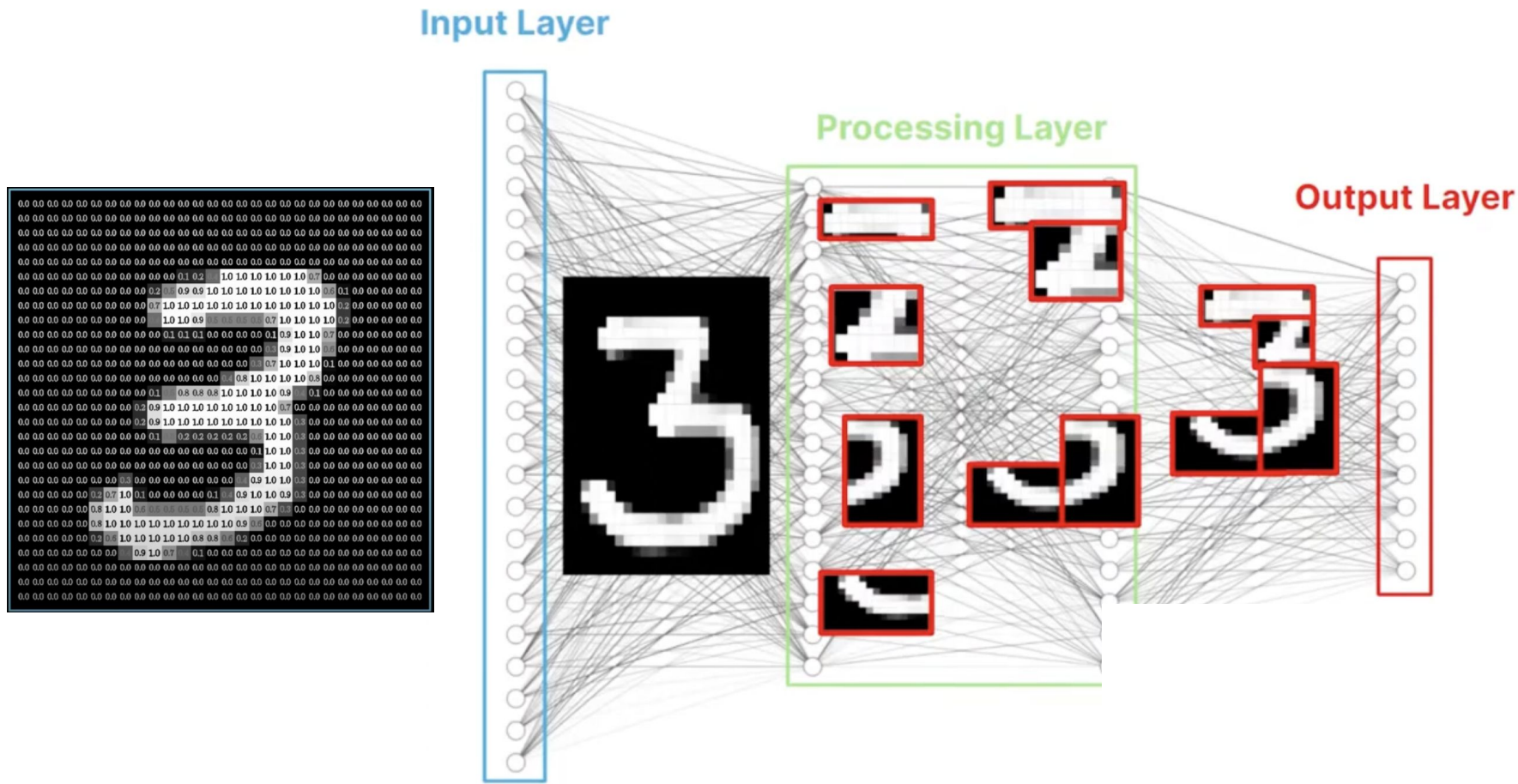


Next word
prediction

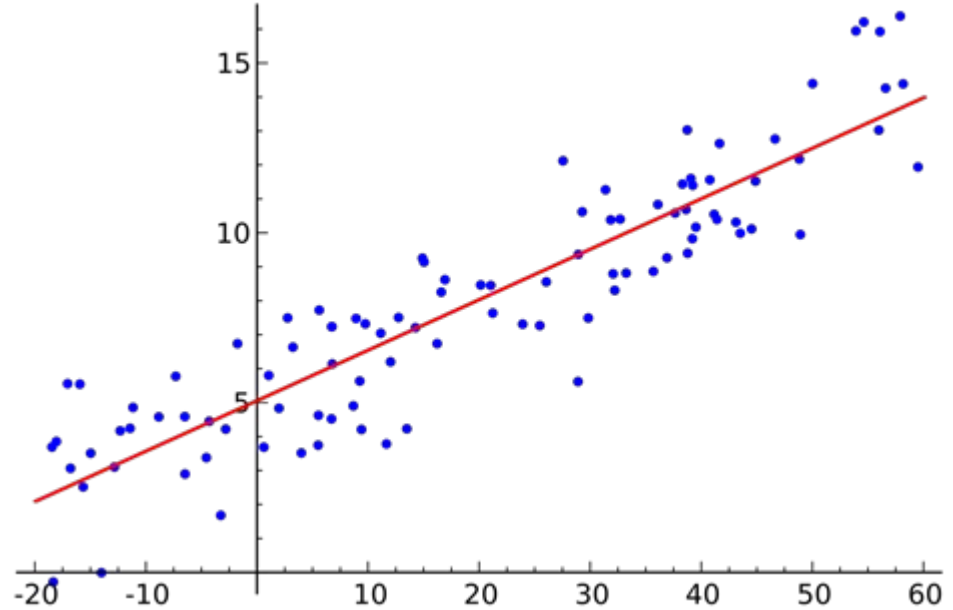
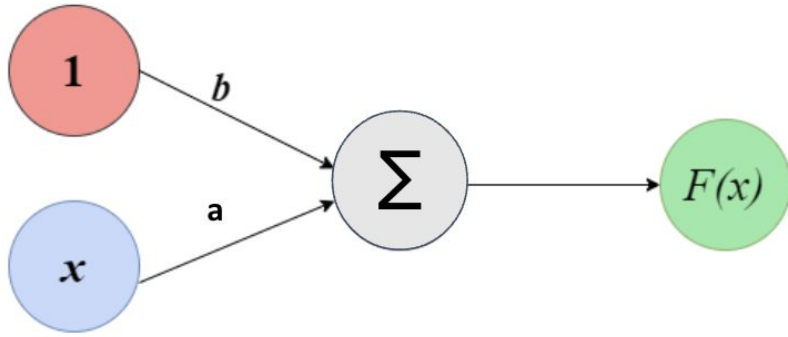
DL, COMMENT?



DL, COMMENT?

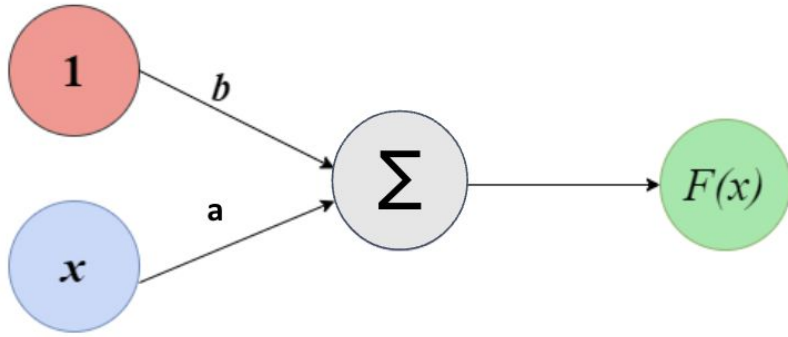


DL, COMMENT?

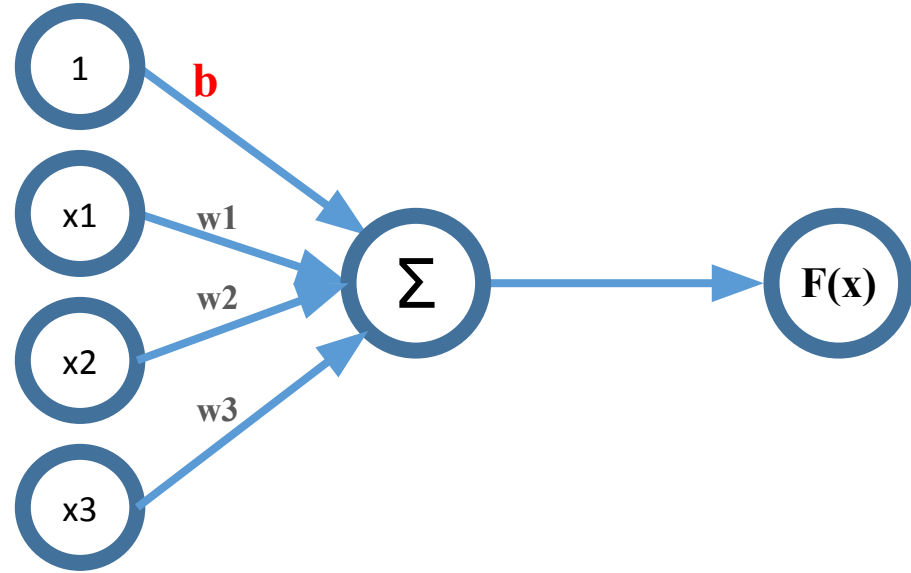


$$F(x) = a.x + b$$

DL, COMMENT?

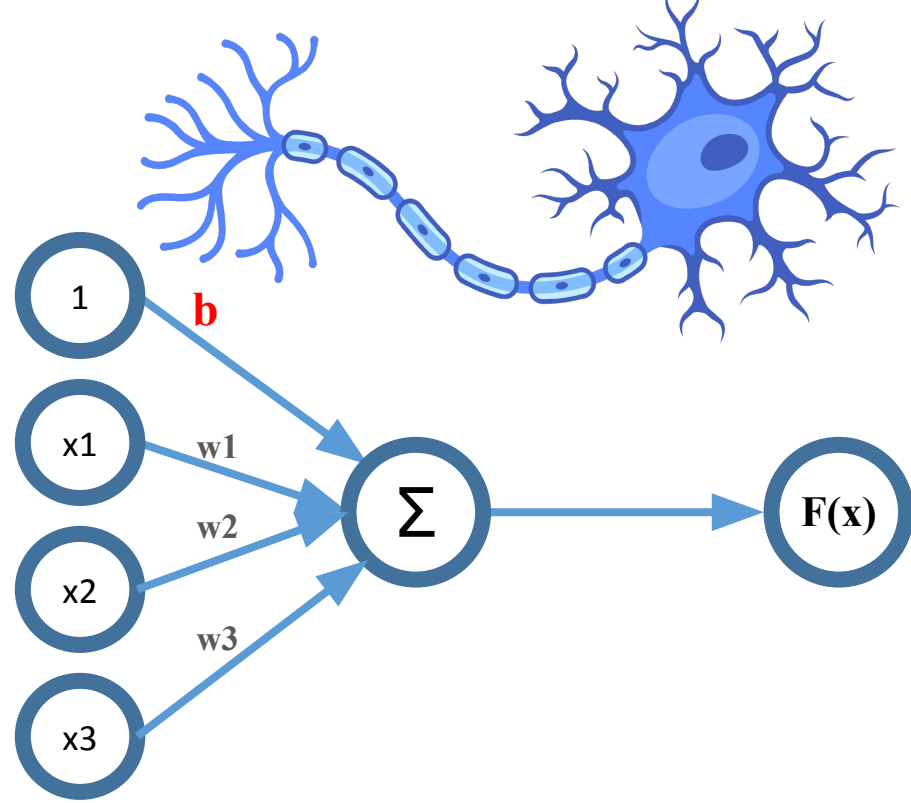
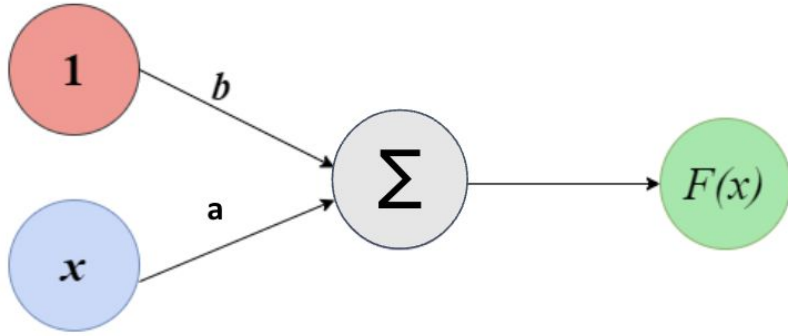


$$F(x) = a.x + b$$



$$F(x) = w1 . x1 + w2 . x2 + w3 . x3 + 1 . b$$

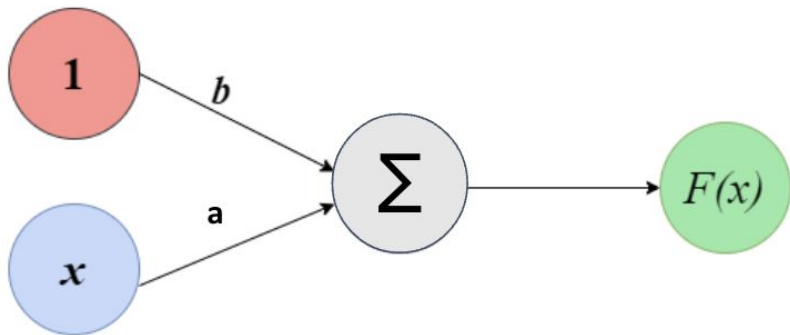
DL, COMMENT?



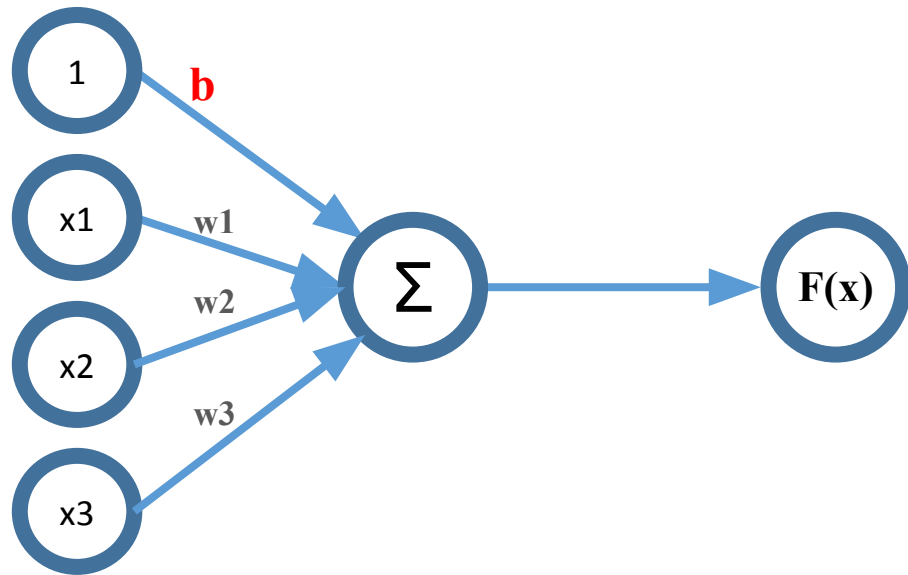
$$F(x) = w1 \cdot x1 + w2 \cdot x2 + w3 \cdot x3 + 1 \cdot b$$

$$F(x) = a \cdot x + b$$

DL, COMMENT?



$$F(x) = a \cdot x + b$$



$$F(x) = w1 \cdot x1 + w2 \cdot x2 + w3 \cdot x3 + 1 \cdot b$$

$$F(x) = W^T \cdot X + b$$

DL, COMMENT?

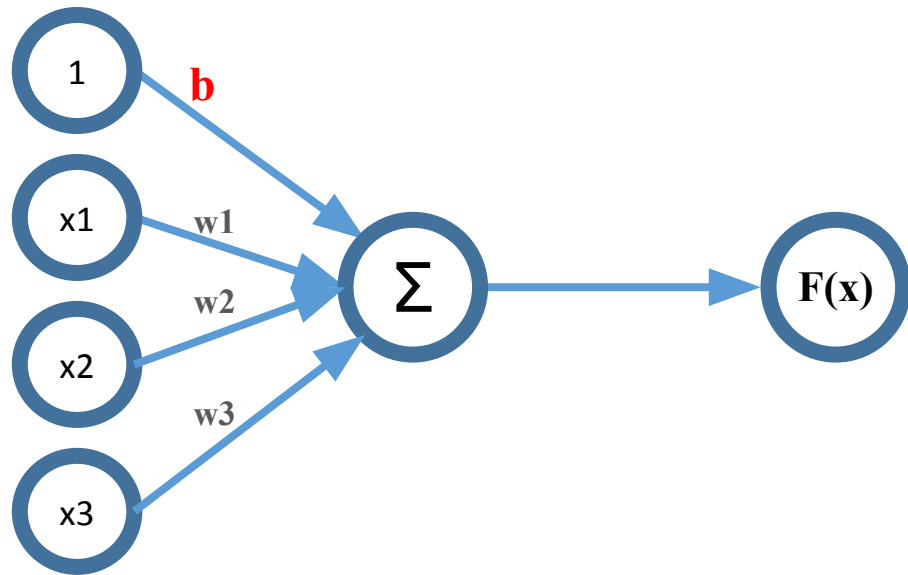
Dot product:

$$\mathbf{a} = \begin{bmatrix} 2 \\ 3 \\ 4 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix}$$

$$\mathbf{a}^T \mathbf{b} = \begin{bmatrix} 2 & 3 & 4 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix}$$

$$= a_1 \cdot b_1 + a_2 \cdot b_2 + a_3 \cdot b_3$$

$$= 2 + 0 + (-4) = -2$$



$$F(x) = w1 \cdot x1 + w2 \cdot x2 + w3 \cdot x3 + 1 \cdot b$$

$$F(x) = \mathbf{W}^T \cdot \mathbf{X} + \mathbf{b}$$

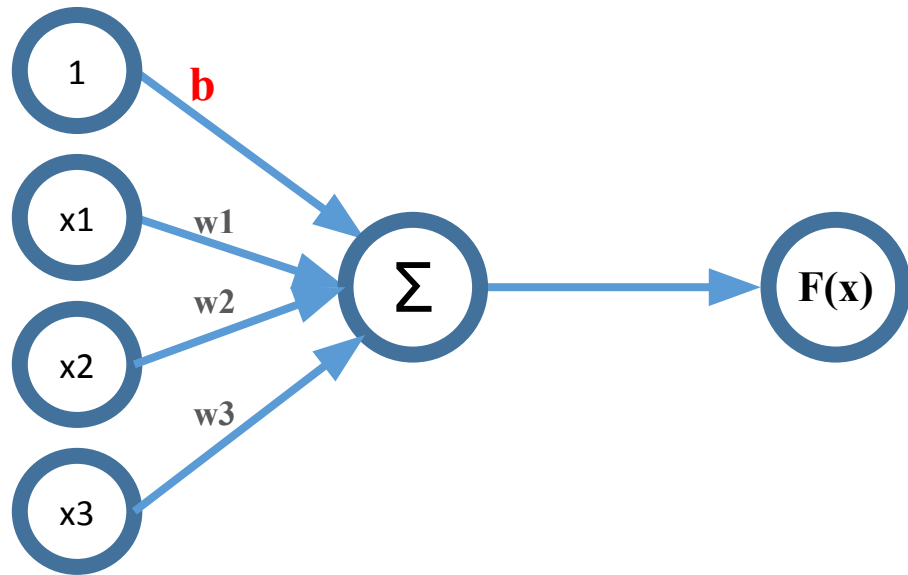
DL, COMMENT?

Dot product:

$$\mathbf{a} = \begin{bmatrix} 2 \\ 3 \\ 4 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix}$$

$$\mathbf{a}^T \mathbf{b} = [2 \quad 3 \quad 4] \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix}$$

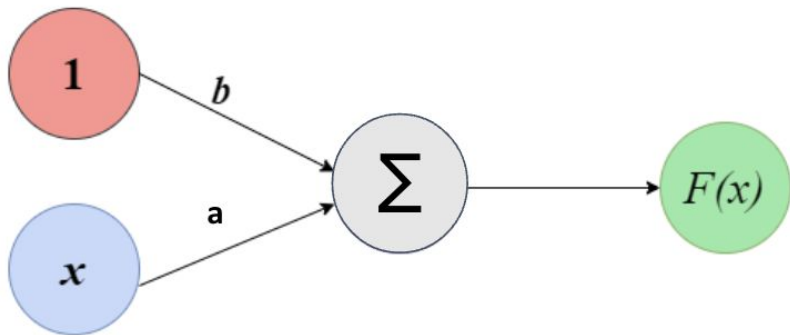
$$= \begin{bmatrix} 2 & 3 & 4 \\ 1 & 0 & -1 \\ 0 & 5 & 2 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix} = \begin{bmatrix} -2 \\ 2 \\ -2 \end{bmatrix}$$



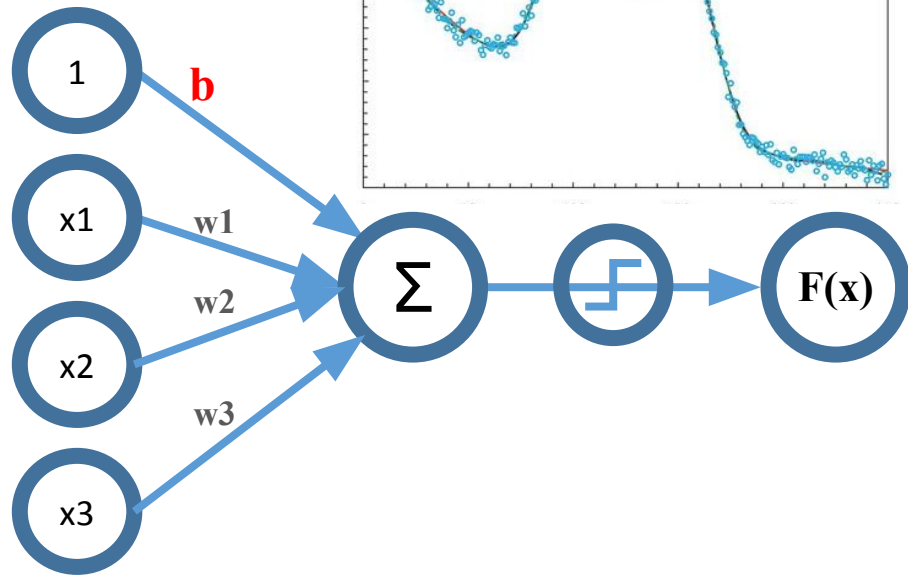
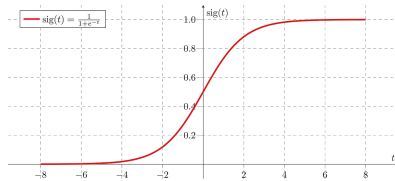
$$F(x) = w1 \cdot x1 + w2 \cdot x2 + w3 \cdot x3 + 1 \cdot b$$

$$F(x) = \mathbf{W}^T \cdot \mathbf{X} + \mathbf{b}$$

DL, COMMENT?



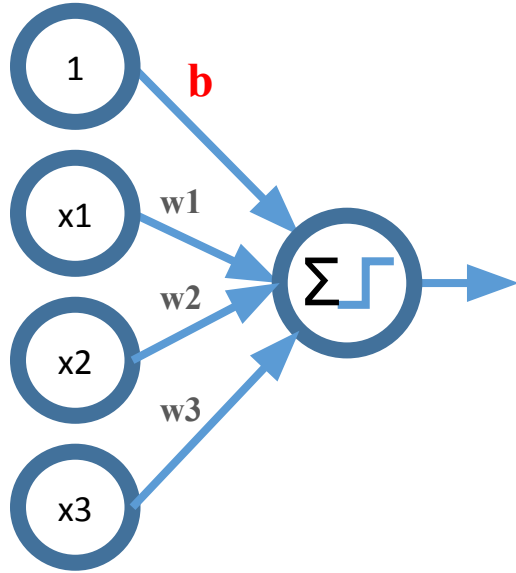
$$F(x) = a.x + b$$



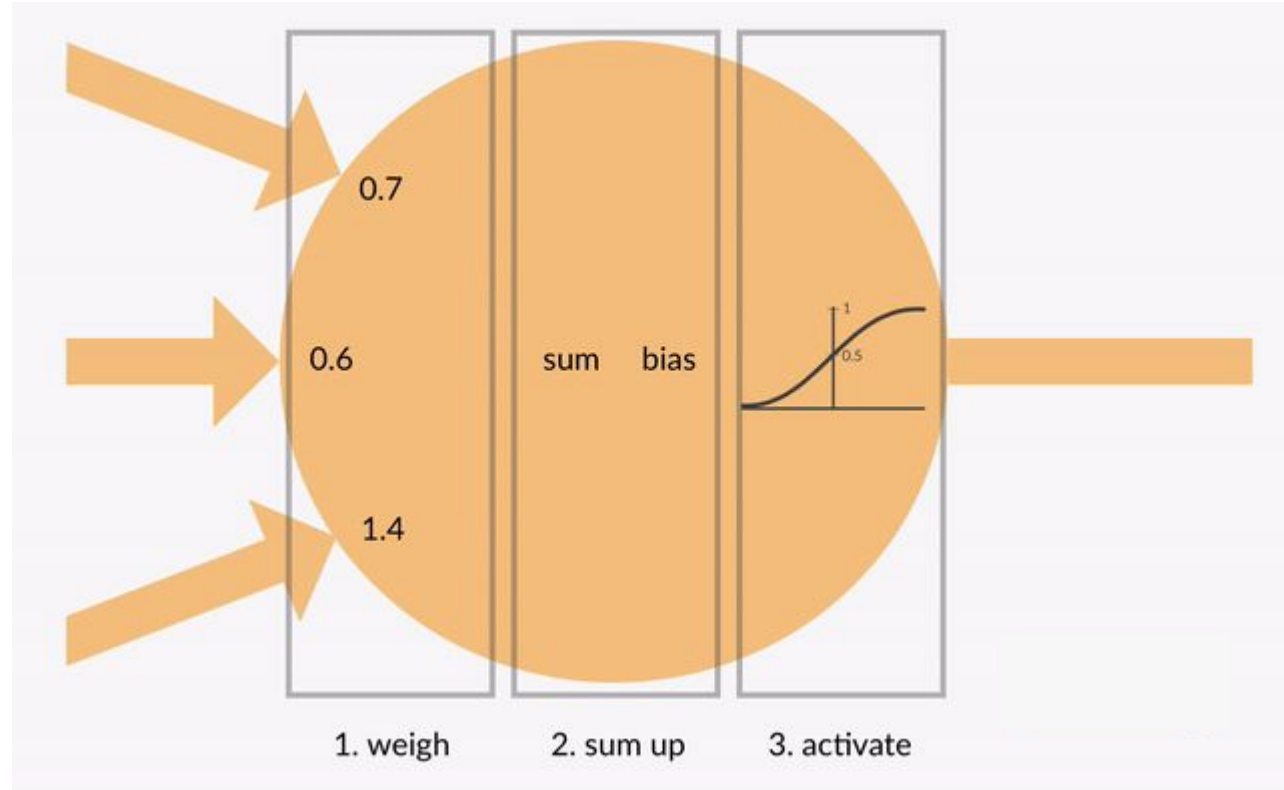
$$F(x) = w1 . x1 + w2 . x2 + w3 . x3 + 1 . b$$

$$F(x) = \sigma(\mathbf{w}^T . \mathbf{x} + b)$$

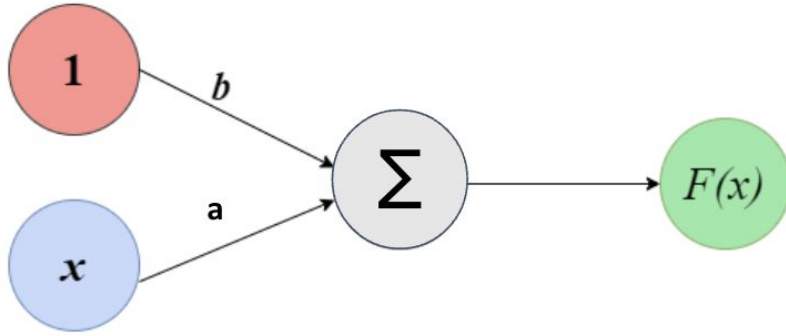
DL, COMMENT?



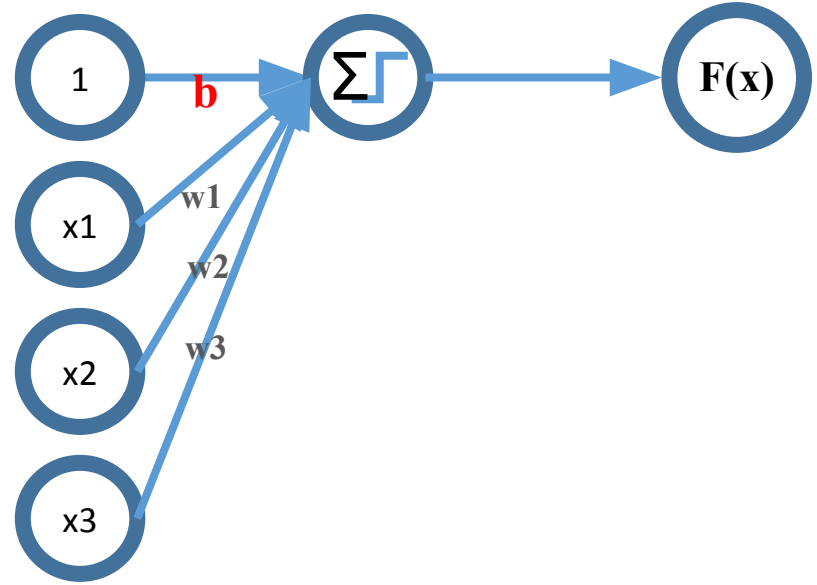
$$F(x) = \sigma(\textcolor{blue}{W} . X + \textcolor{red}{b})$$



DL, COMMENT?

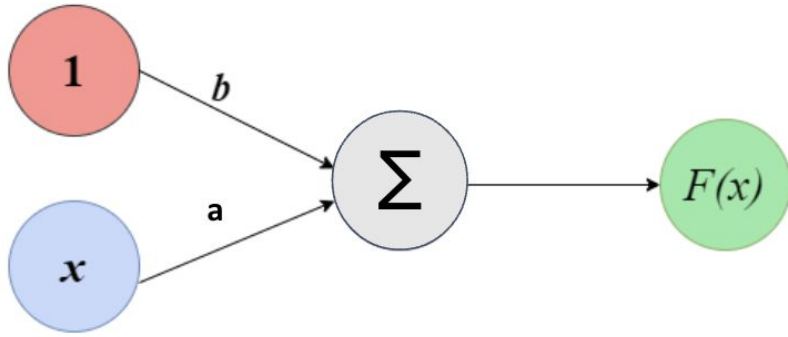


$$F(x) = a.x + b$$

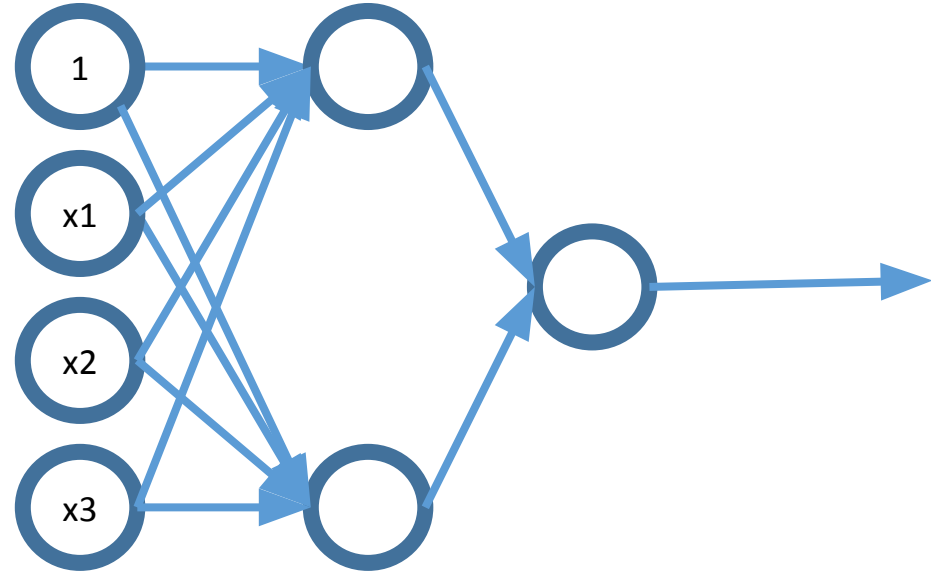


$$F(x) = \sigma(W.X + b)$$

DL, COMMENT?

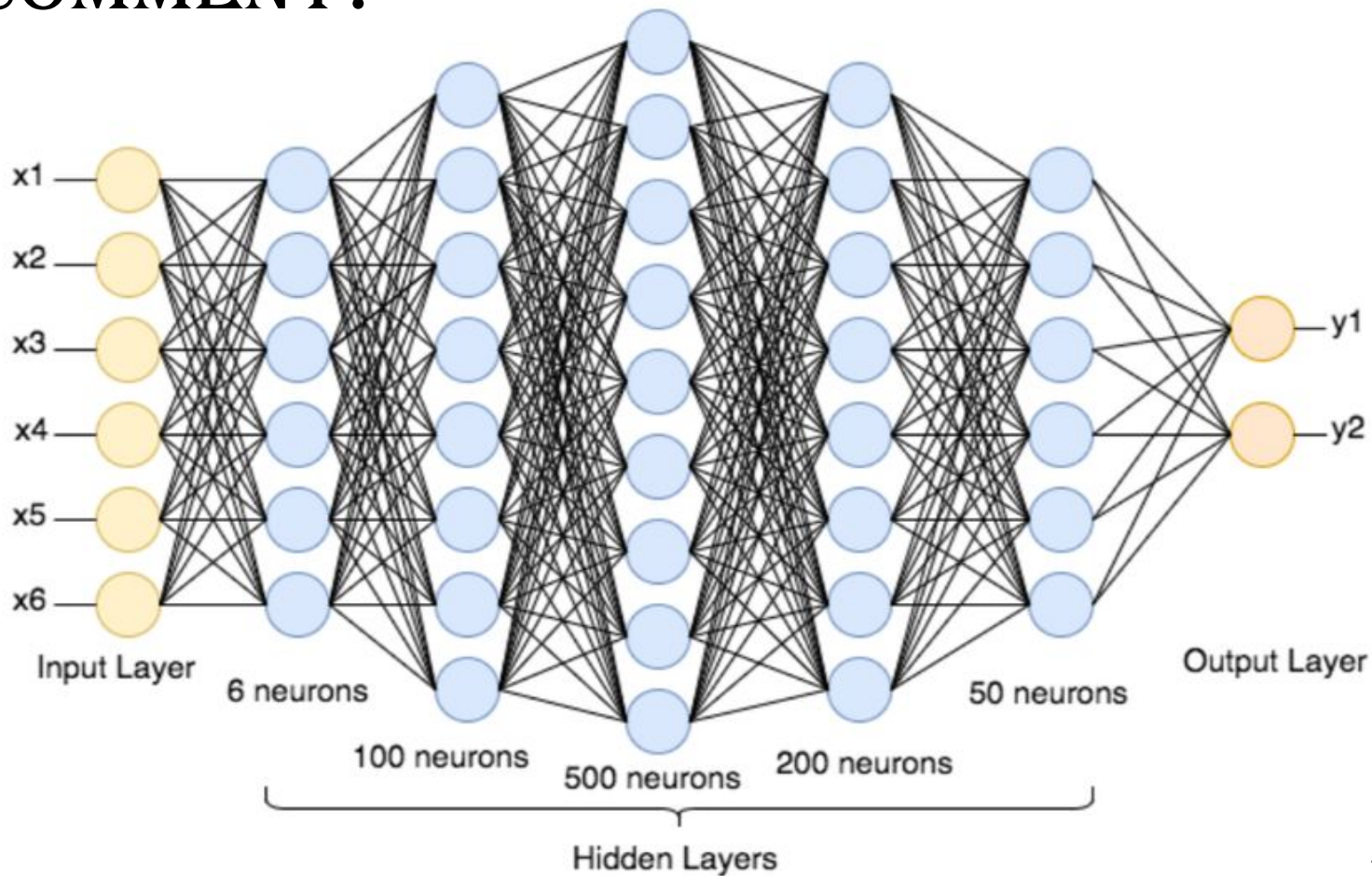


$$F(x) = a.x + b$$



$$\sigma(W.X + b)$$

DL, COMMENT?

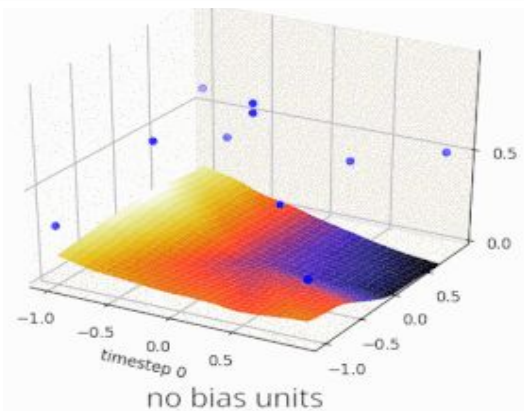
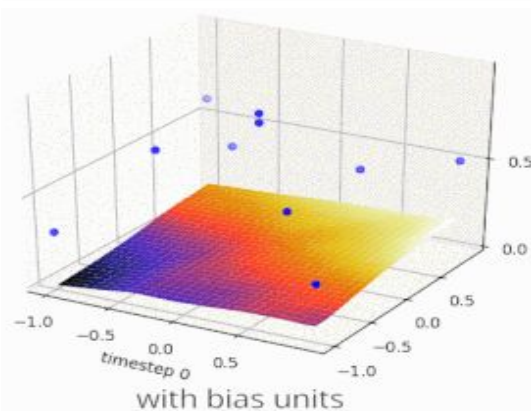
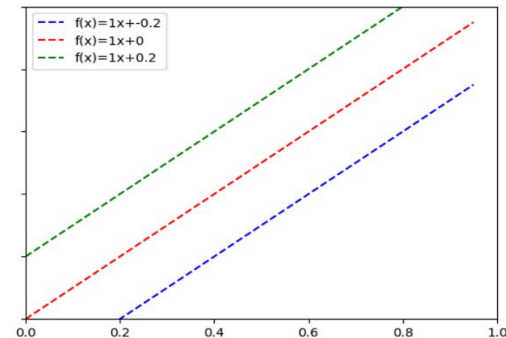
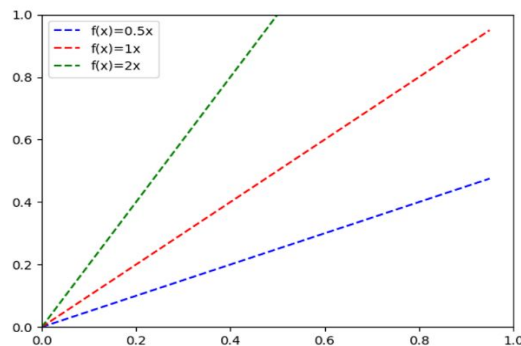


BIAIS

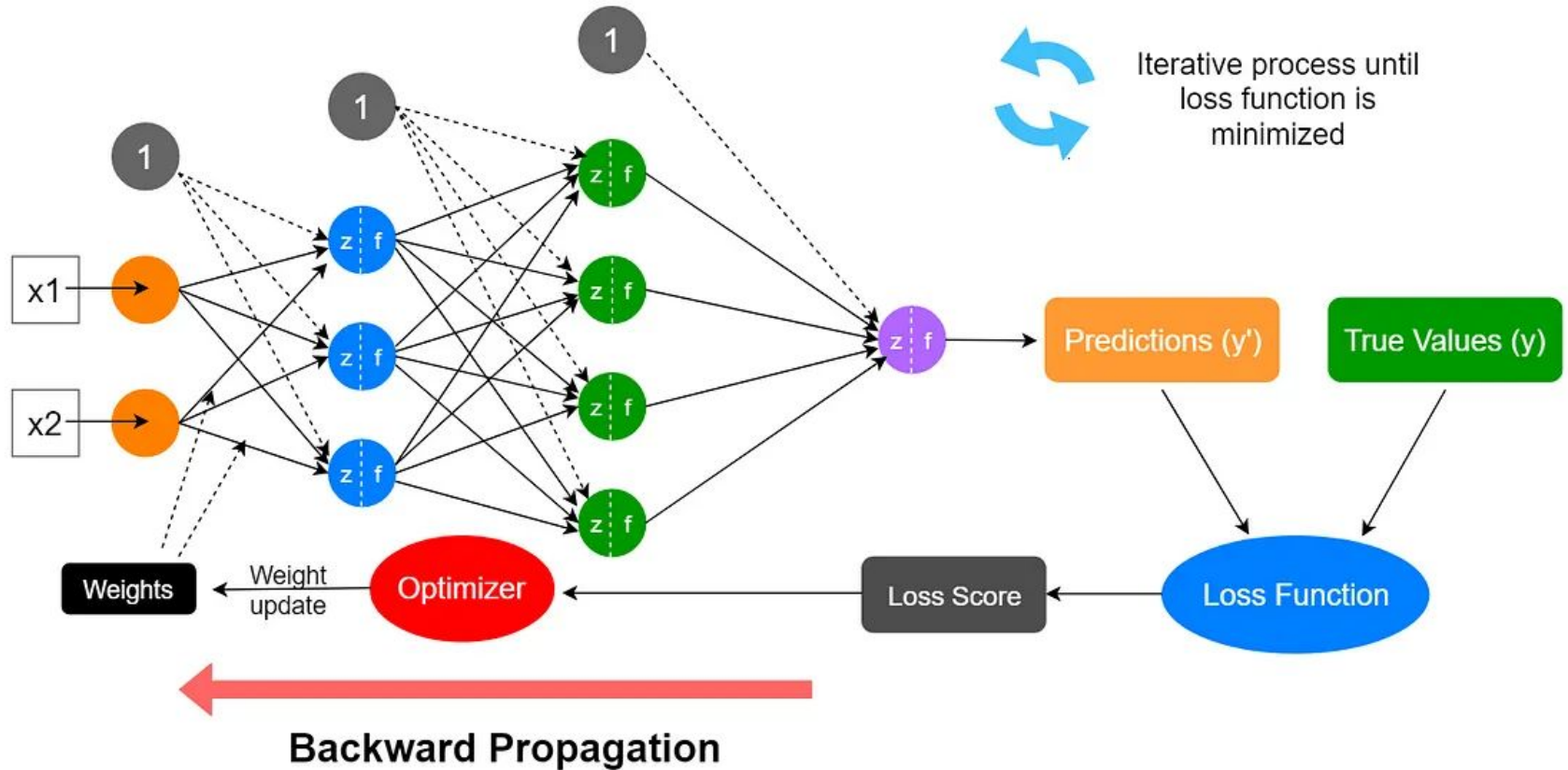
La modification des poids ne sert qu'à manipuler la forme/courbure/ souplesse de votre fonction.

L'introduction de biais permet de déplacer la courbe de la fonction verticalement tout en laissant la forme/courbure inchangée.

De plus, le biais vous permet d'utiliser un seul réseau neuronal pour représenter des cas similaires.



Forward Propagation



DL, OBJECTIF

Hypothèse :

$$f(x) = a \cdot x + b$$

Paramètres :

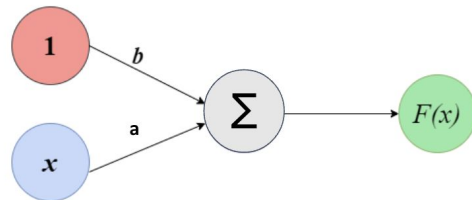
$$a, b$$

Fonction Coût :

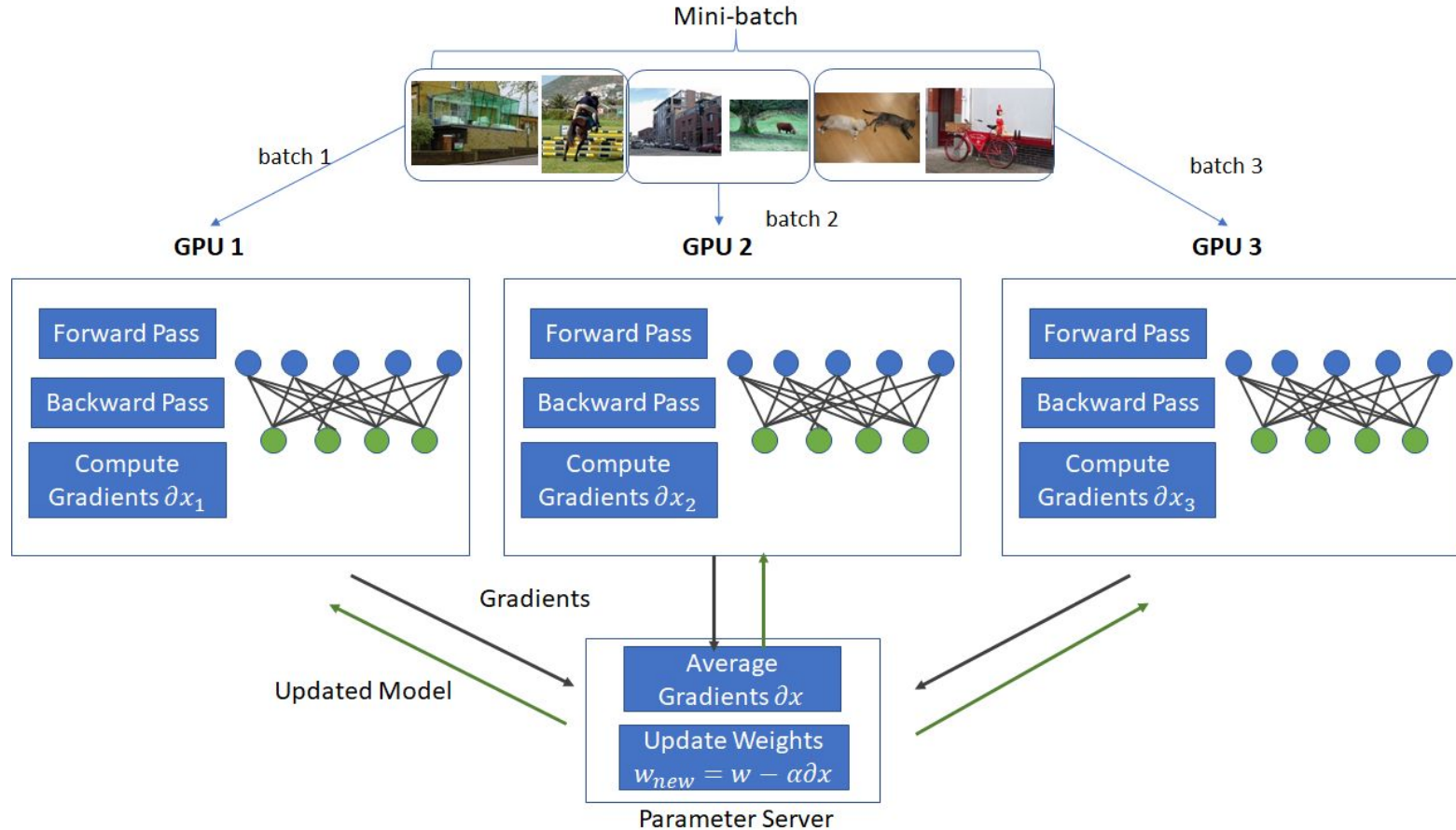
$$J(a, b) = \frac{1}{2m} \sum_{i=1}^m \left(f(x^{(i)}) - y^{(i)} \right)^2$$

Objectif :

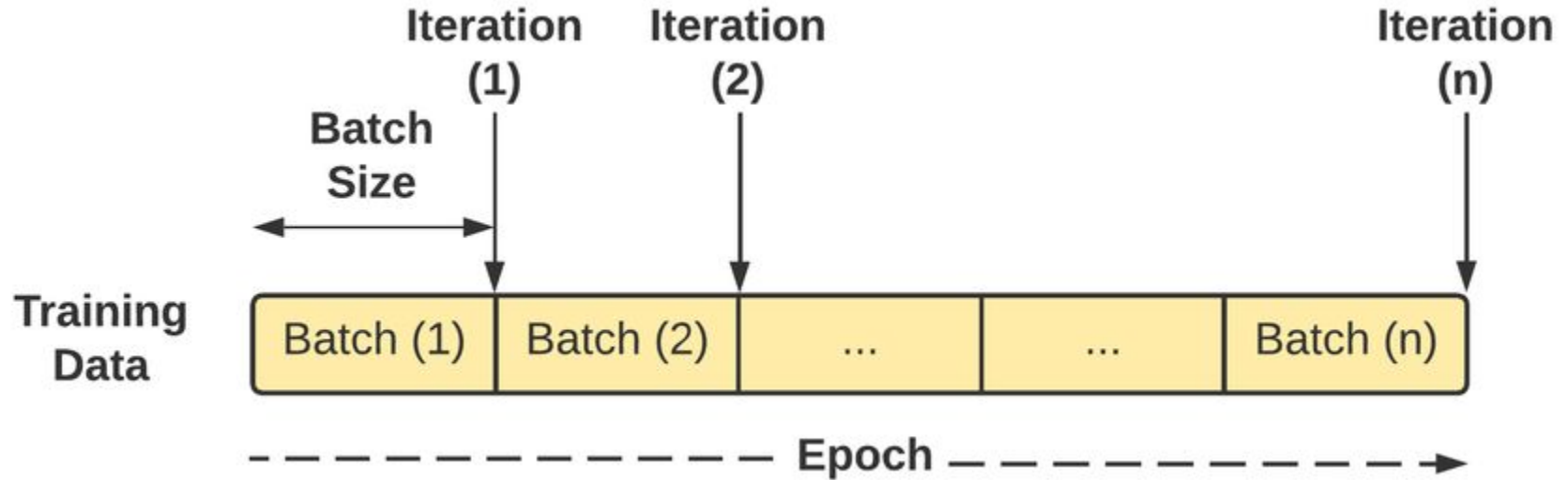
$$\min_{a, b} J(a, b)$$



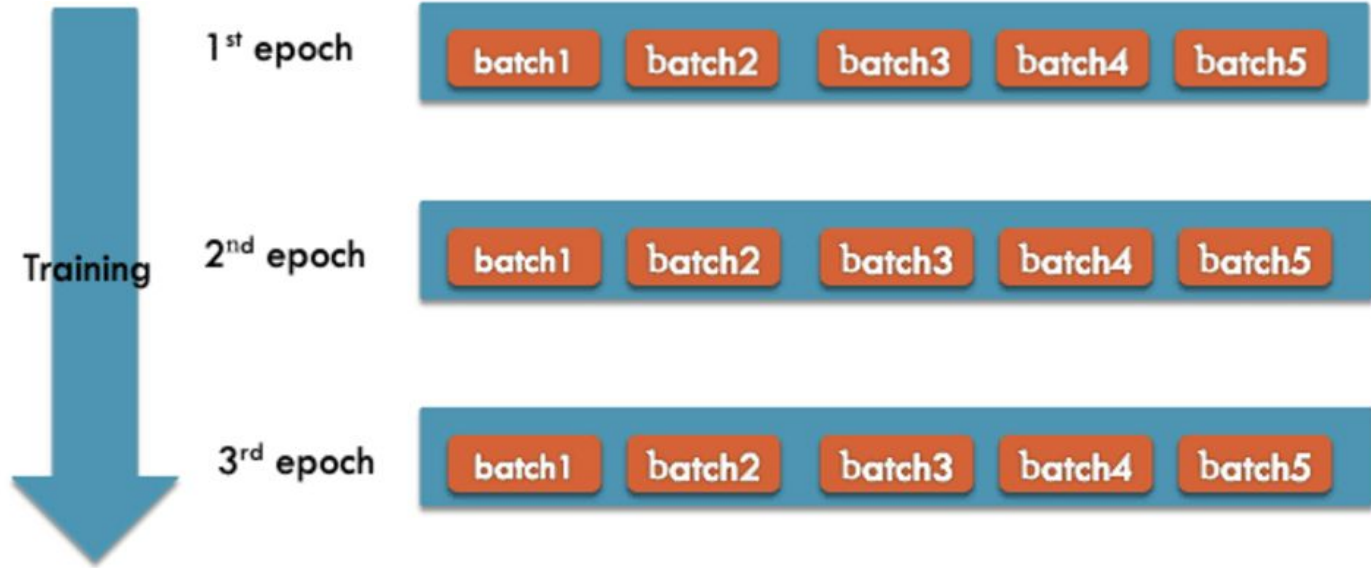
PARALLÉLISATION



ITERATION VS EPOCH



ITERATION VS EPOCH



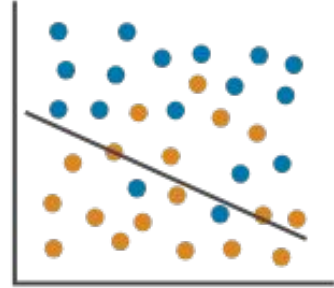
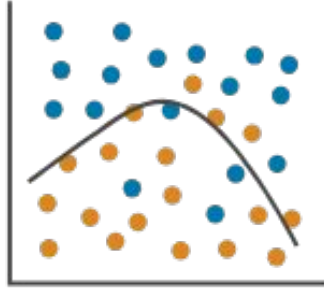
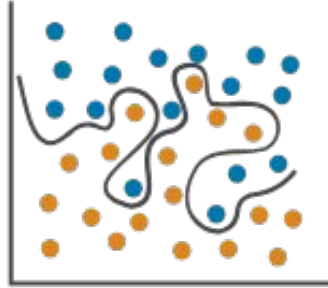
OVERFITTING

Overfitting

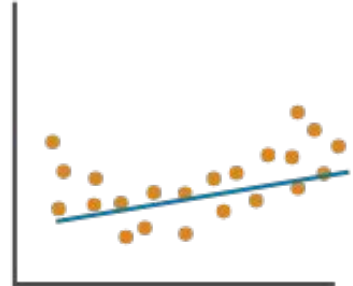
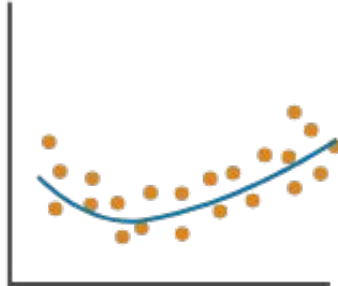
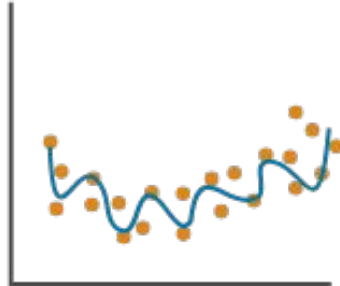
Right Fit

Underfitting

Classification



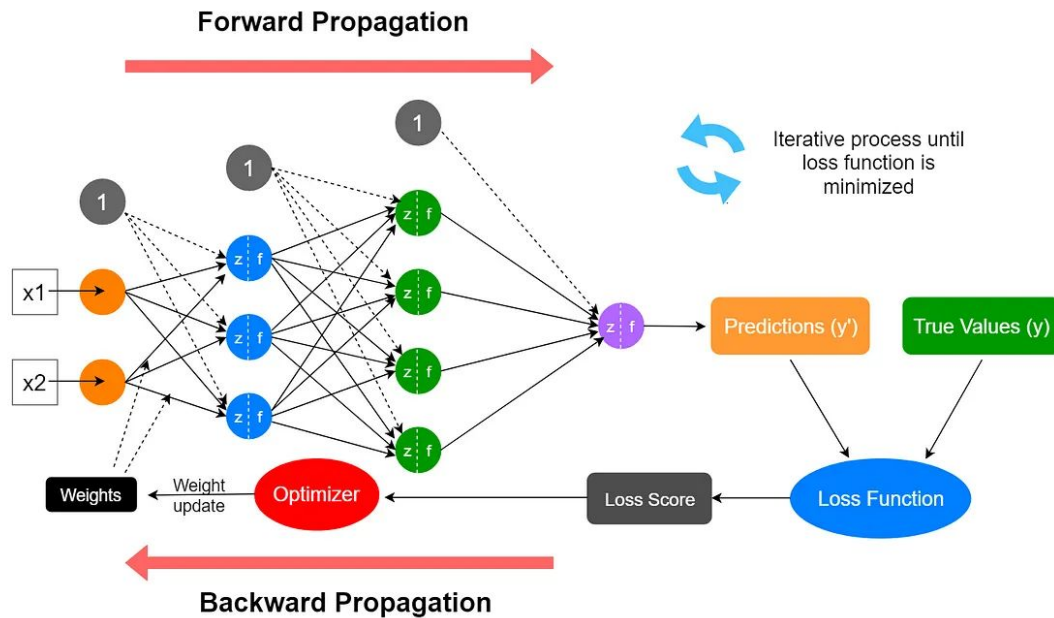
Regression



OVERFITTING, SOLUTIONS

Solution	Description
Régularisation (L1/L2)	Ajoute une pénalité au terme de coût pour limiter les valeurs des poids.
Dropout	Désactive aléatoirement certaines connexions entre les neurones pendant l'entraînement.
Augmentation des données	Génère des variations des données d'entraînement (rotation, recadrage, etc.) pour augmenter leur diversité.
Réduction de la complexité du modèle	Diminue la taille ou la profondeur du modèle pour réduire sa capacité à mémoriser les données.
Utilisation d'un ensemble de validation	Surveille les performances sur des données de validation pour détecter le surapprentissage.
Arrêt anticipé (Early Stopping)	Arrête l'entraînement dès que les performances sur l'ensemble de validation cessent de s'améliorer.
Collecte de plus de données	Ajoute davantage de données pour réduire le risque de mémorisation excessive.
Normalisation/Standardisation des données	Met les données d'entrée sur une échelle uniforme pour stabiliser l'apprentissage.

OVERFITTING, REGULARISATION



$$Loss = Error(y, \hat{y}) + \lambda \sum_{i=1}^N |w_i|$$