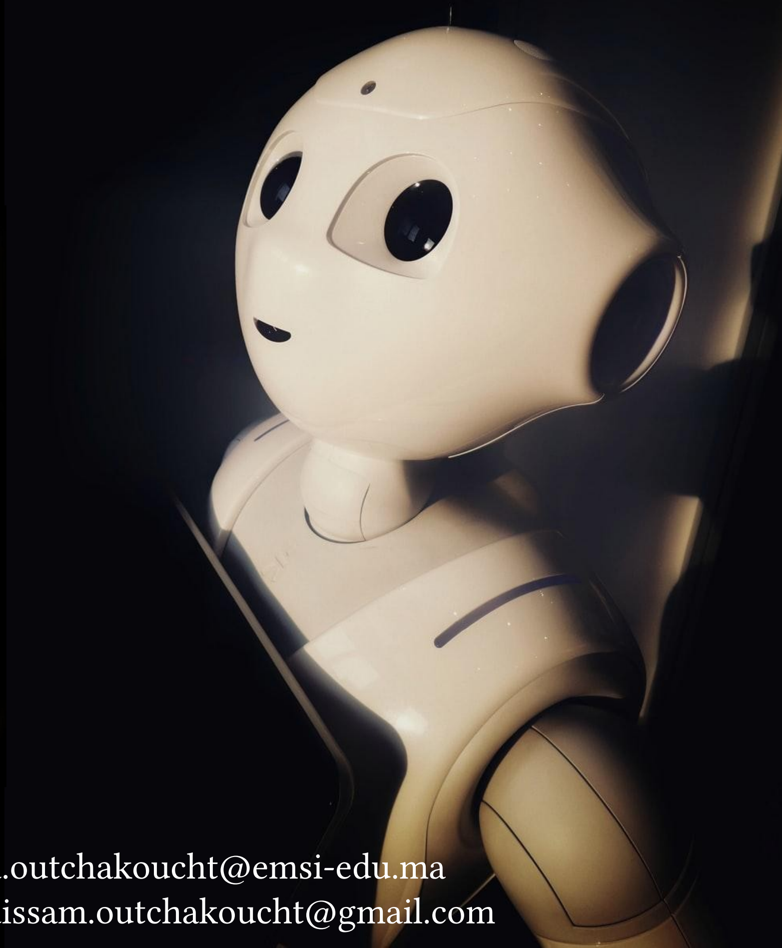
ECOLE MAROCAINE DES SCIENCES DE L'INGENIEUR
Membre de HONORIS UNITED UNIVERSITIES

Deep Learning: INtroduction



Aissam Outchakoucht

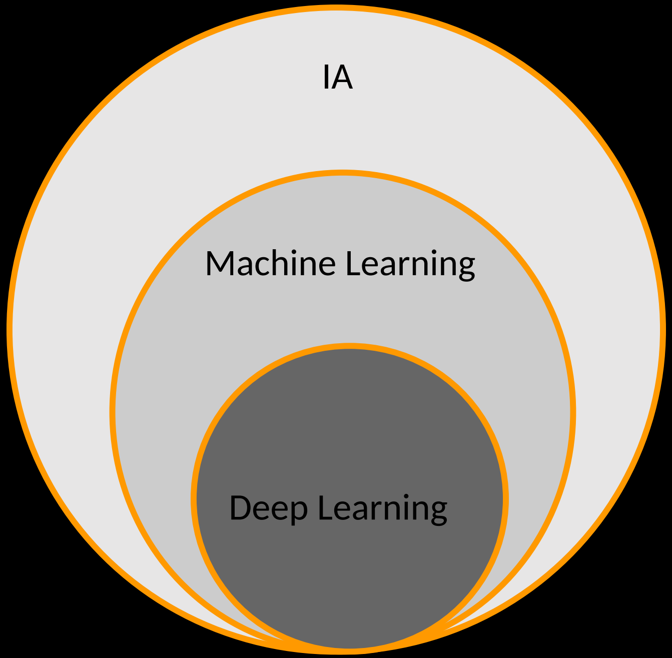
a.outchakoucht@emsi-edu.ma
aissam.outchakoucht@gmail.com

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.

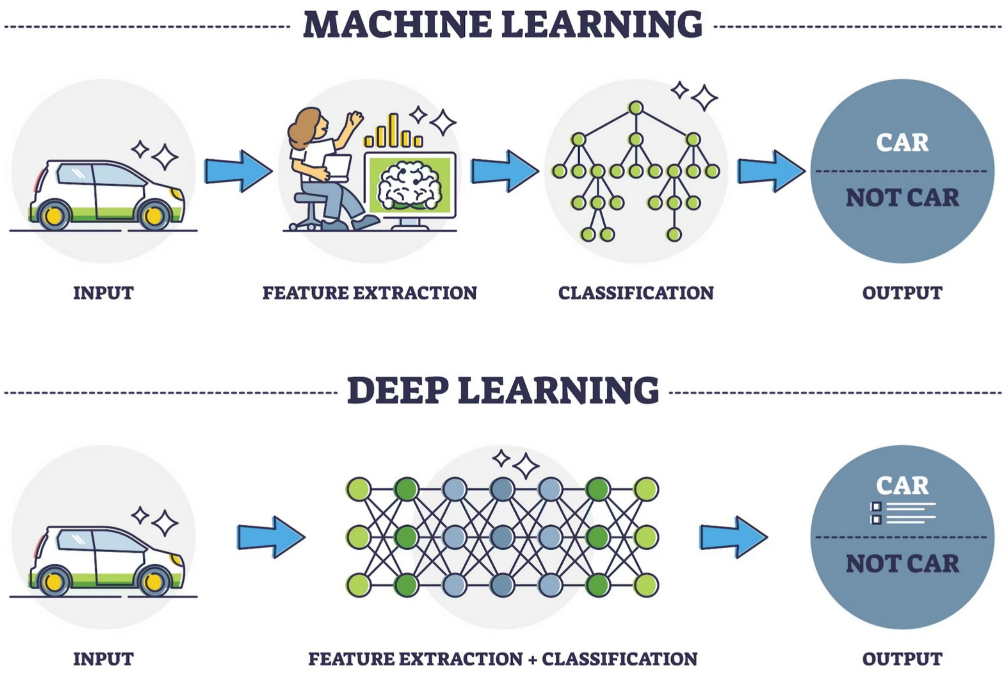
DL est un type de ML inspiré par la structure du cerveau humain.

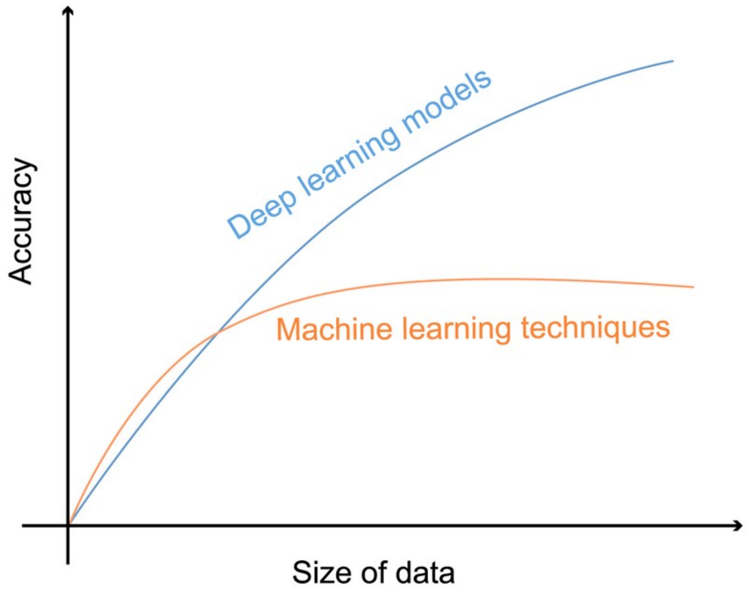


Limites des algorithmes de ML

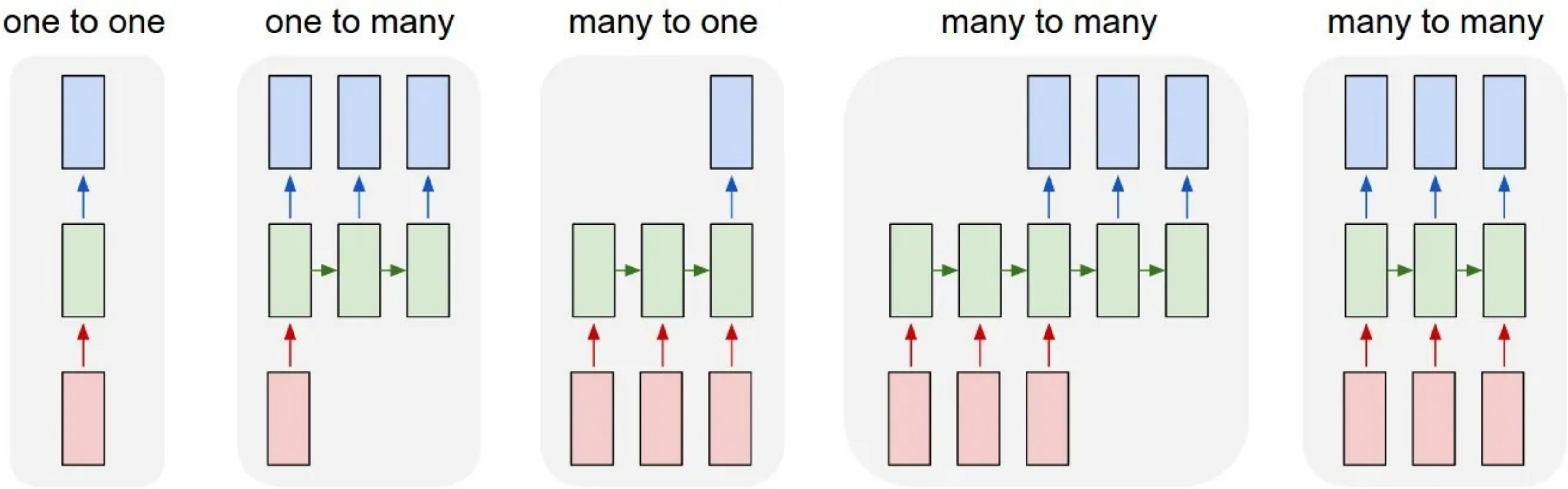
- **Ingénierie des caractéristiques** : De nombreux algorithmes dépendent de caractéristiques bien conçues.
- **Évolutivité** : Des algorithmes comme SVM et KNN ont du mal avec de grands ensembles de données.
- **Relations complexes** : Les relations non linéaires sont difficiles pour des modèles comme 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
CAR
NOT CAR
INPUT
FEATURE EXTRACTION
CLASSIFICATION
OUTPUT
DEEP LEARNING
CAR
NOT CAR
INPUT
FEATURE EXTRACTION + CLASSIFICATION
OUTPUT


Accuracy
Deep learning models
Machine learning techniques
Size of data

DL ?



DL ?

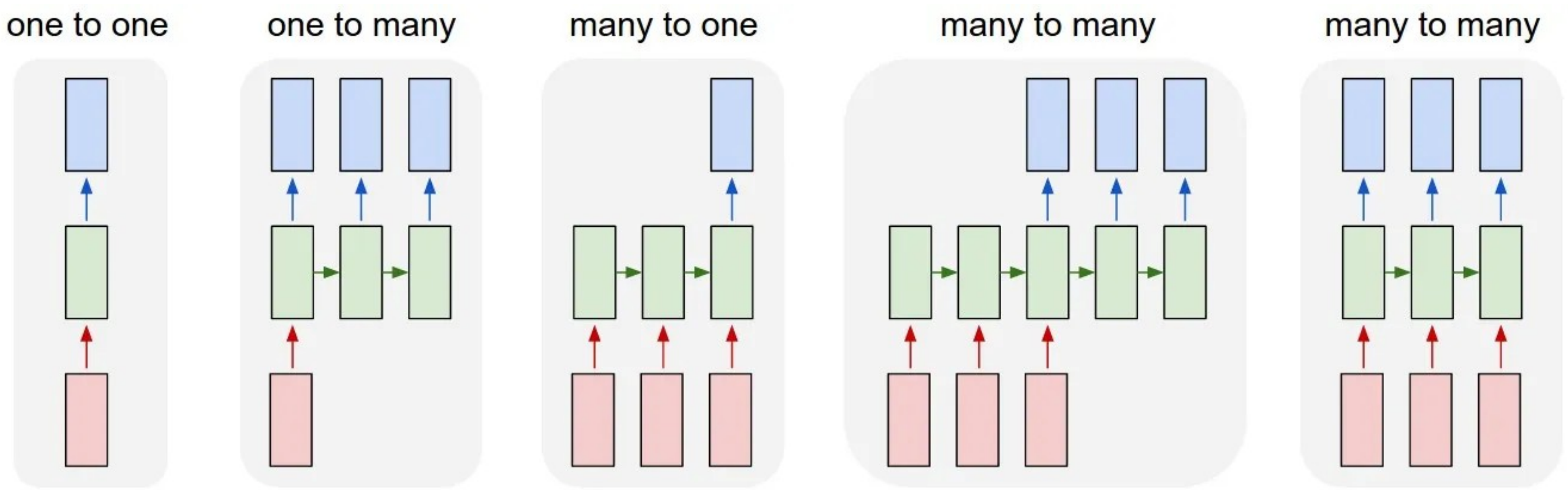


Image Label

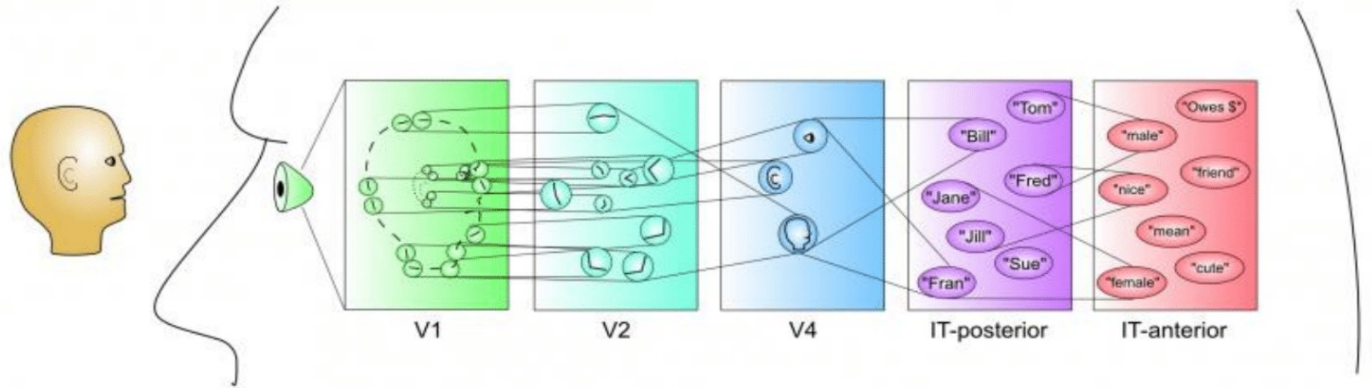
Image Description

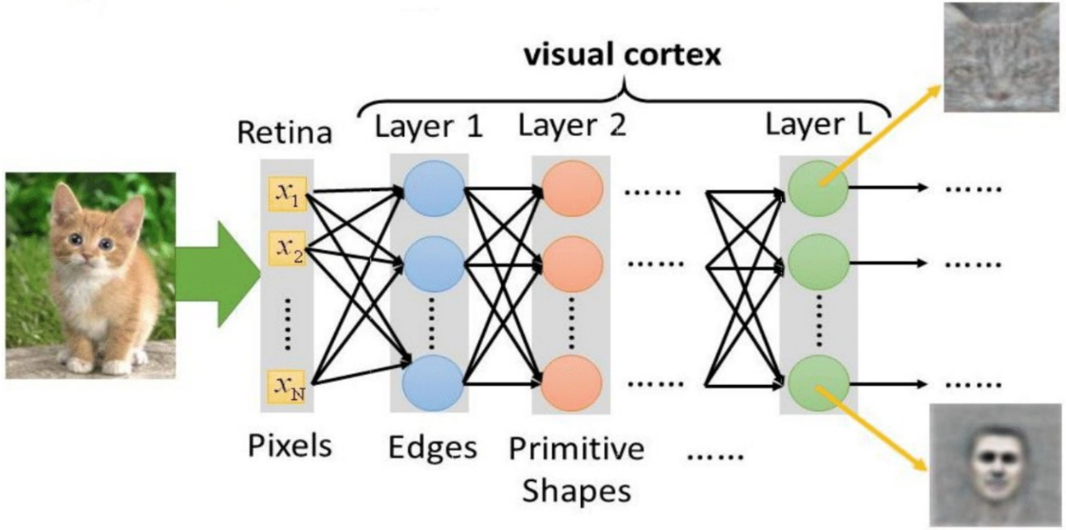
Sentiment Analysis

Sentence Translation

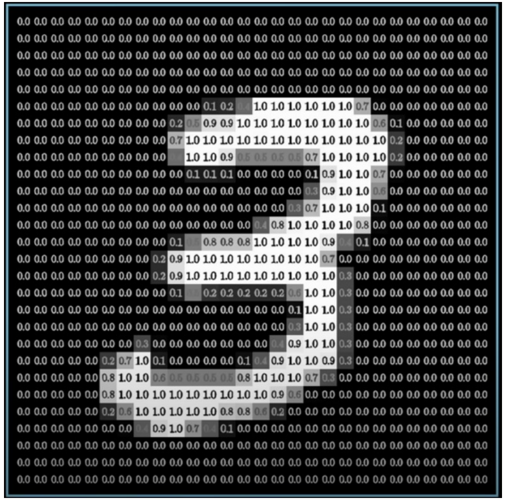
Next word prediction

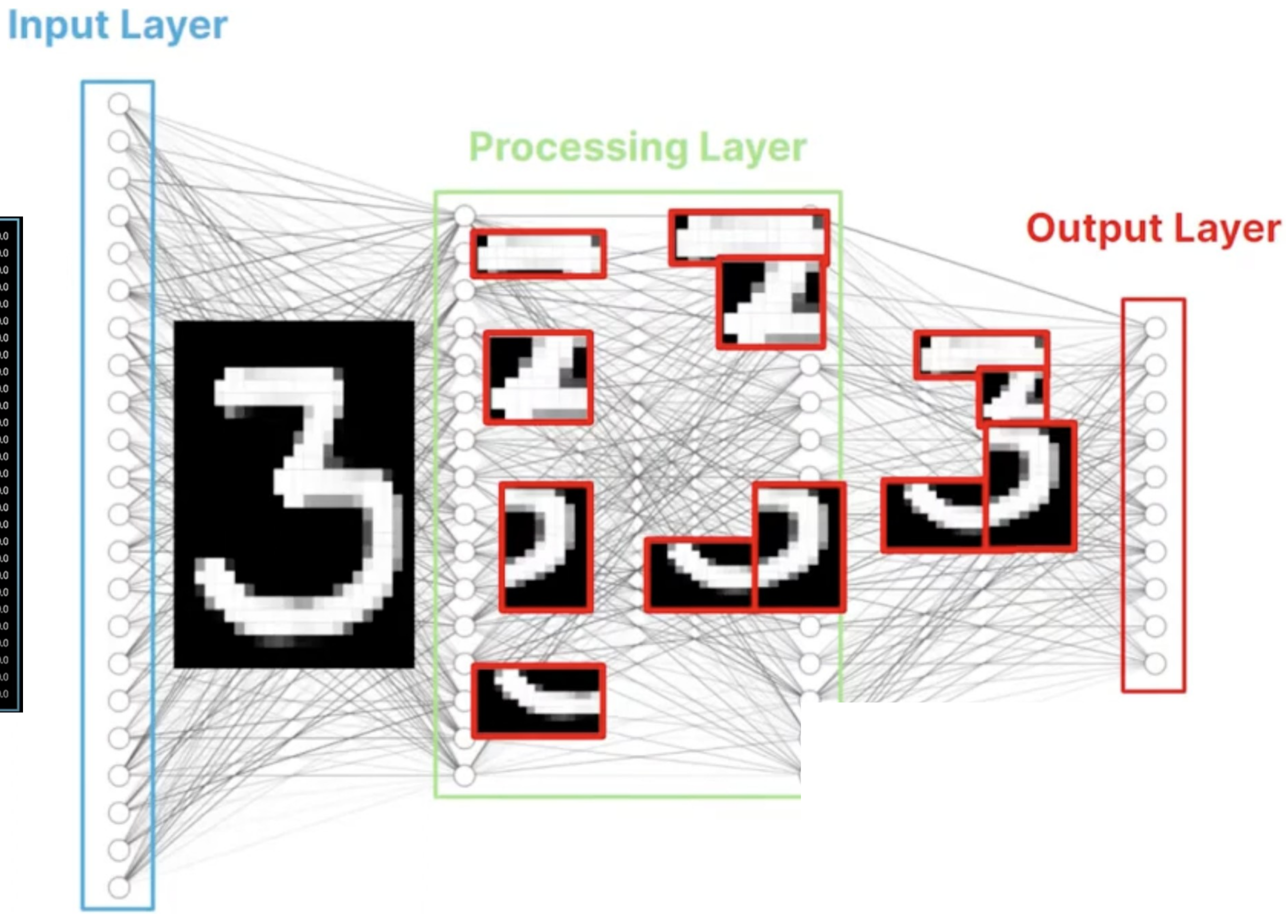
DL, Comment?


"Tom"
"Bill"
"Fred"
"Jane"
"Jill"
"Sue"
"Fran"
"Owes $"
"male"
"friend"
"nice"
"mean"
"cute"
"female"
V1
V2
V4
IT-posterior
IT-anterior

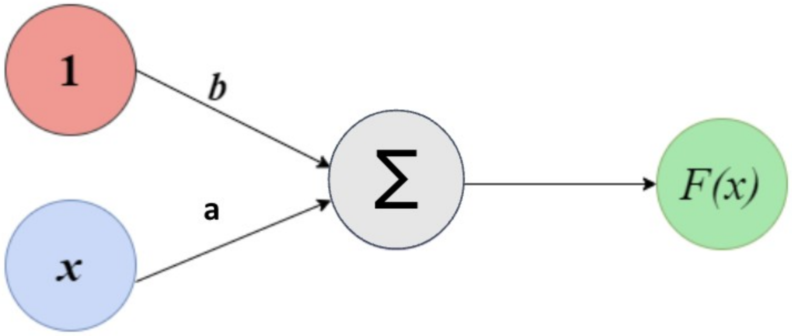

visual cortex
Retina
Layer 1
Layer 2
Layer L
x_1
x_2
x_N
Pixels
Edges
Primitive Shapes
......

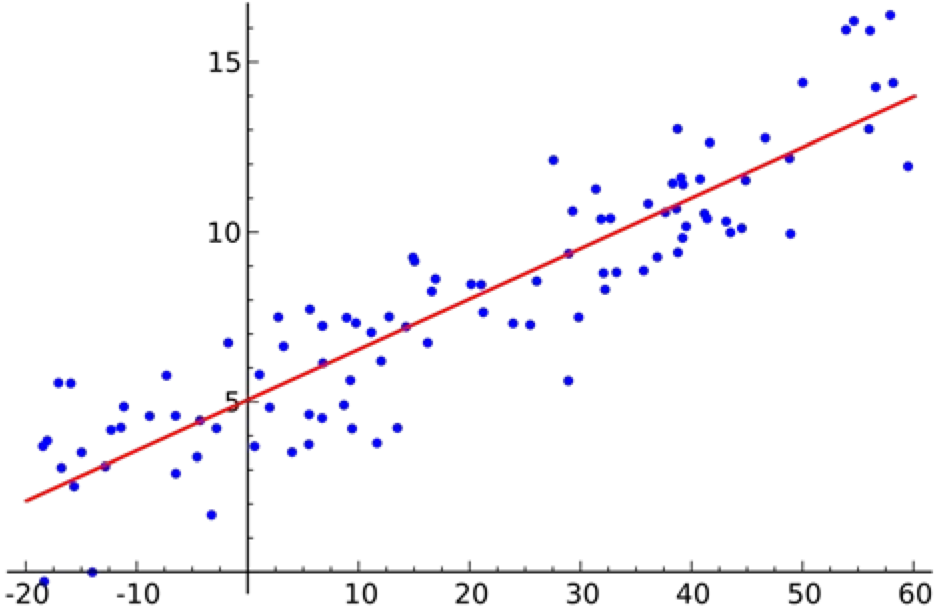
DL, Comment?





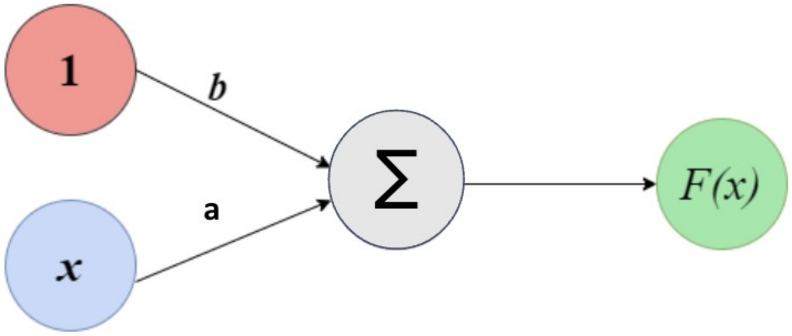
DL, Comment?



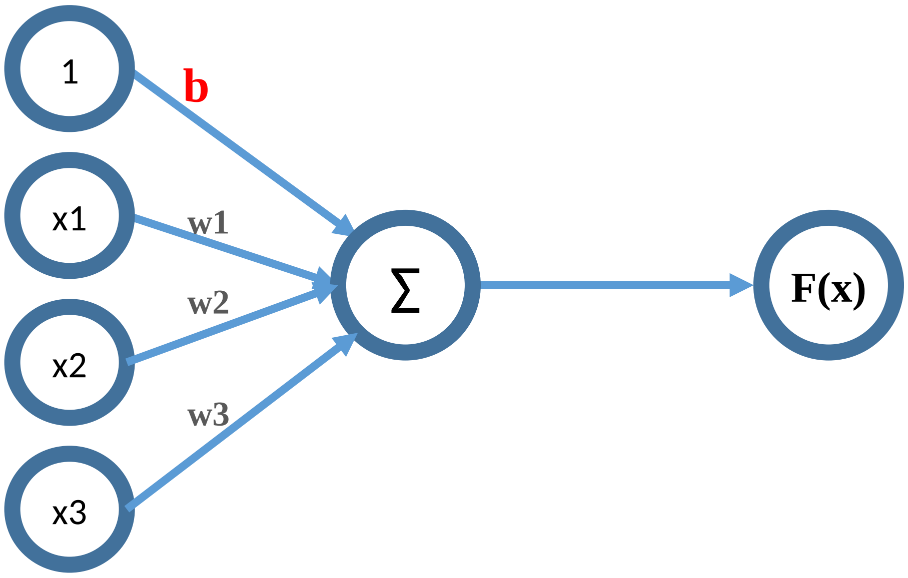


$F(x) = a.x + b$

DL, Comment?

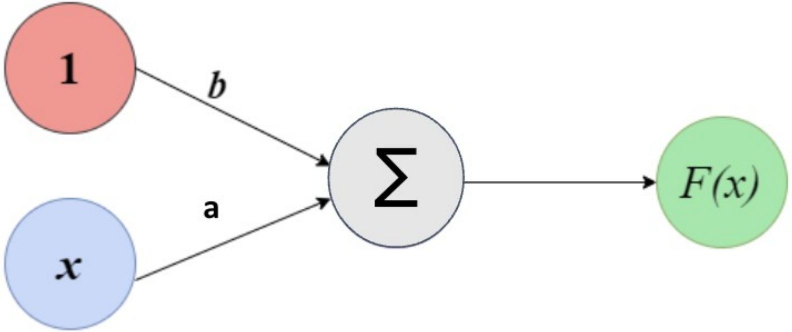


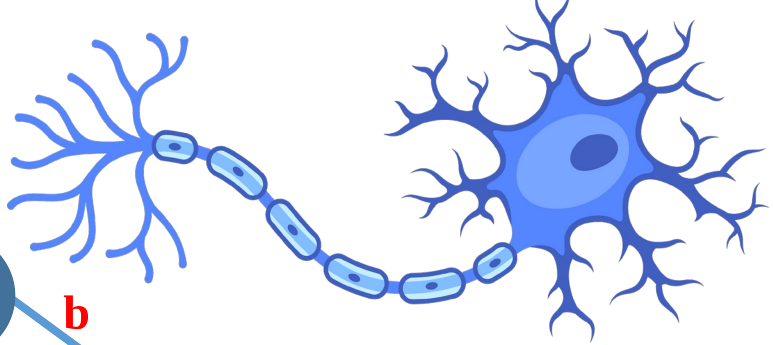
$F(x) = \color{blue}{a}.x + \color{red}{b}$

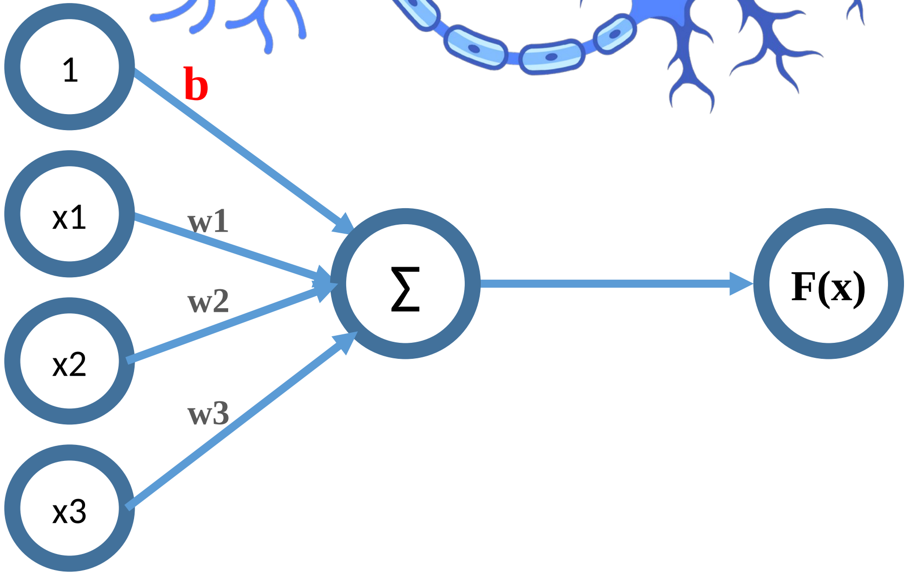


$F(x) = w1 \,.\, x1 + w2 \,.\, x2 + w3 \,.\, x3 + 1 \,.\, b$

DL, Comment?



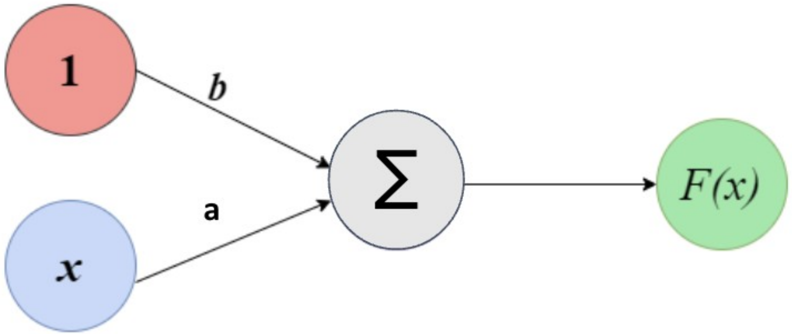


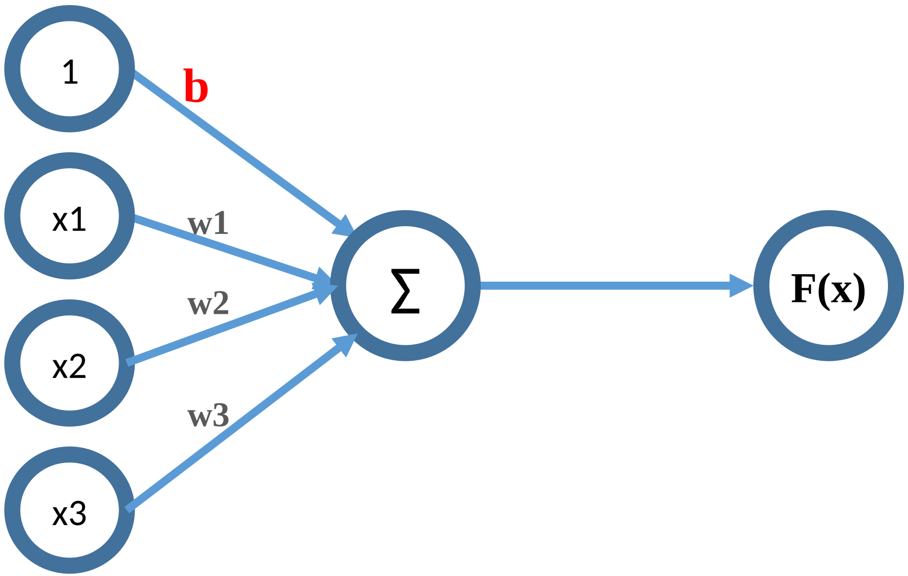


$$F(x) = w1 \,.\, x1 + w2 \,.\, x2 + w3 \,.\, x3 + 1 \,.\, b$$

$$F(x) = {\color{blue}a}.x + {\color{red}b}$$

DL, Comment?





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

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

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

DL, Comment?

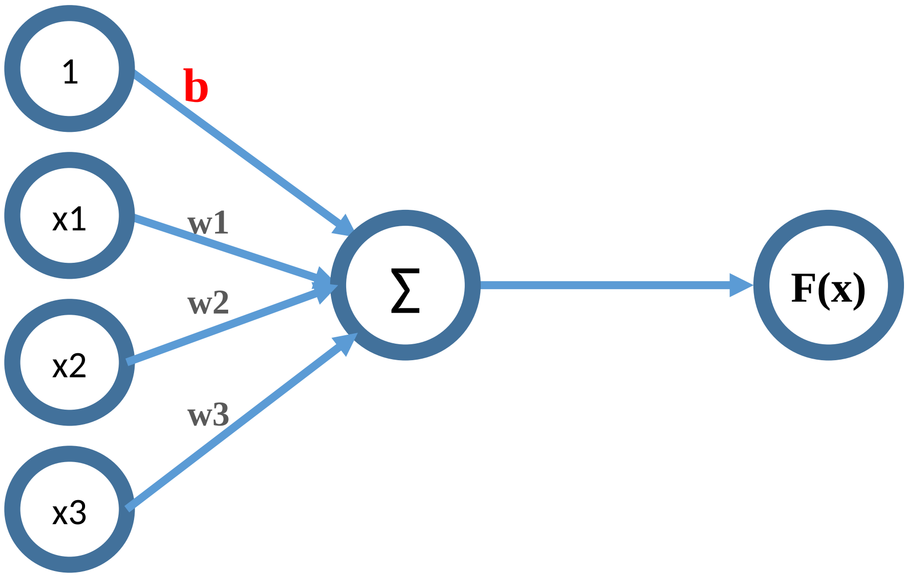
Dot product:

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

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

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

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



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

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

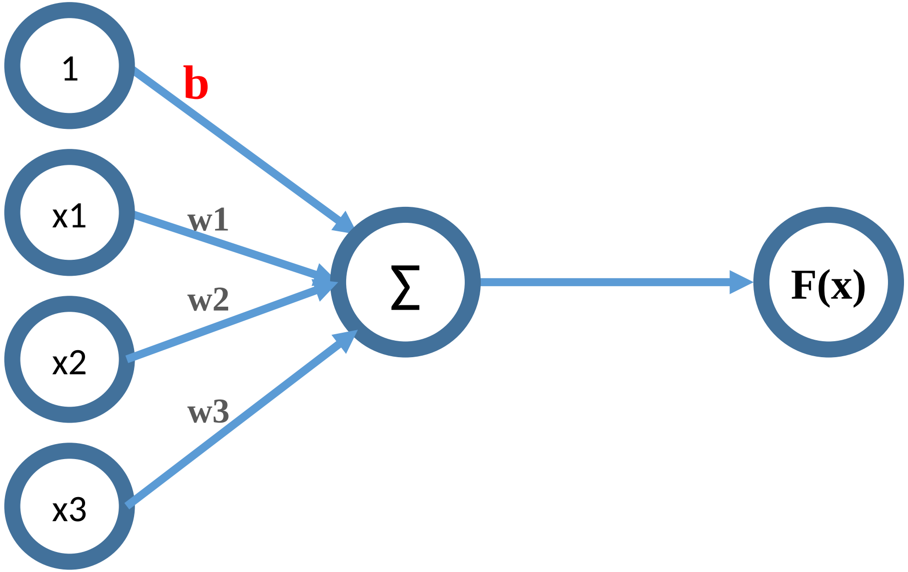
DL, Comment?

Dot product:

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

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

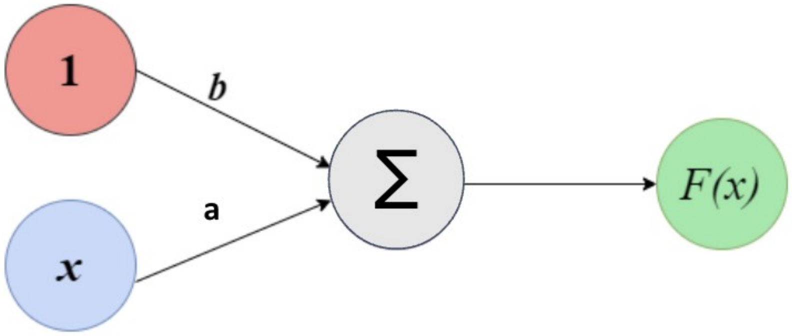
$$= \begin{bmatrix} 2 & 3 & 4 \\ 1 & 0 & -1 \\ 0 & 5 & 2 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix} = \begin{bmatrix} -2 \\ 2 \\ -2 \end{bmatrix}$$

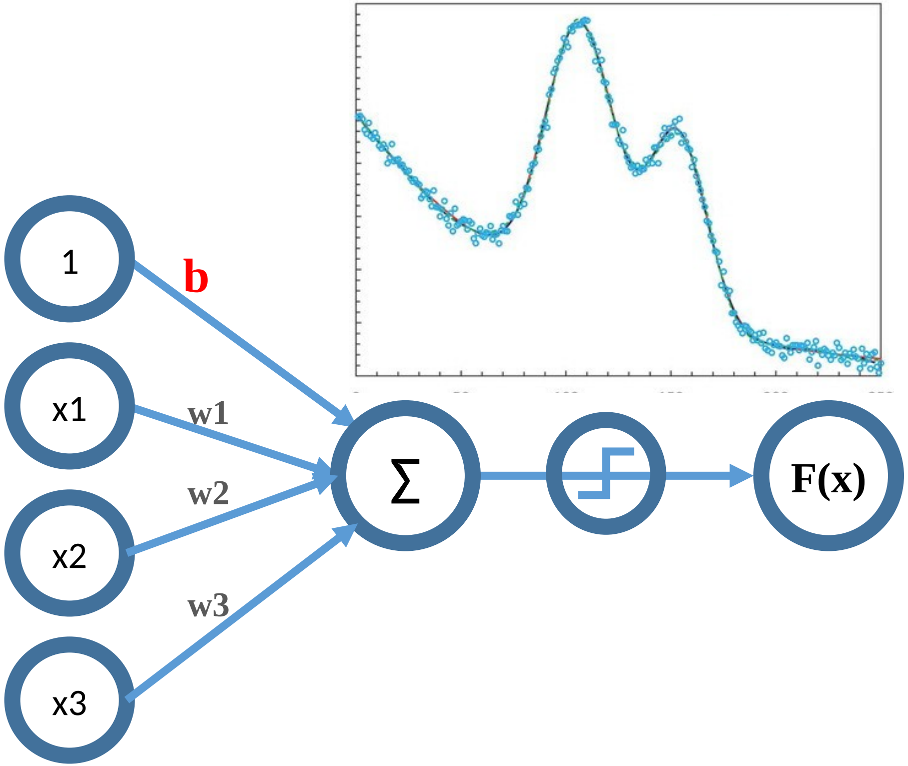


$$F(x) = w1 \,.\, x1 + w2 \,.\, x2 + w3 \,.\, x3 + 1 \,.\, b$$

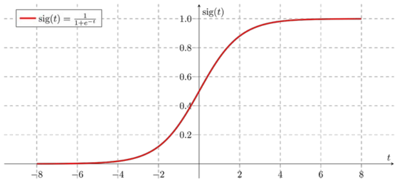
$$F(x) = W^T.X + b$$

DL, Comment?





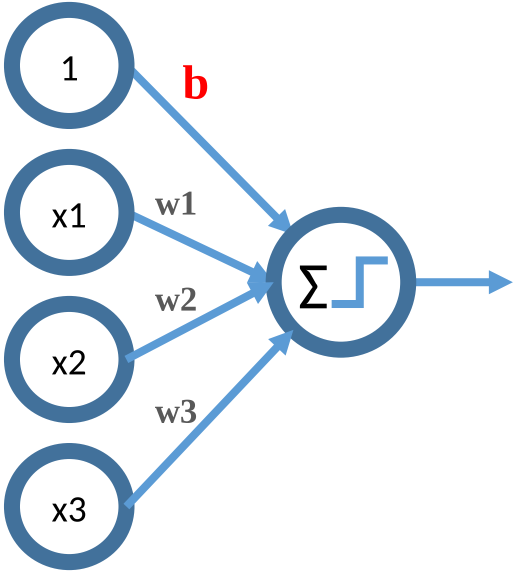
$F(x) = a.x + b$



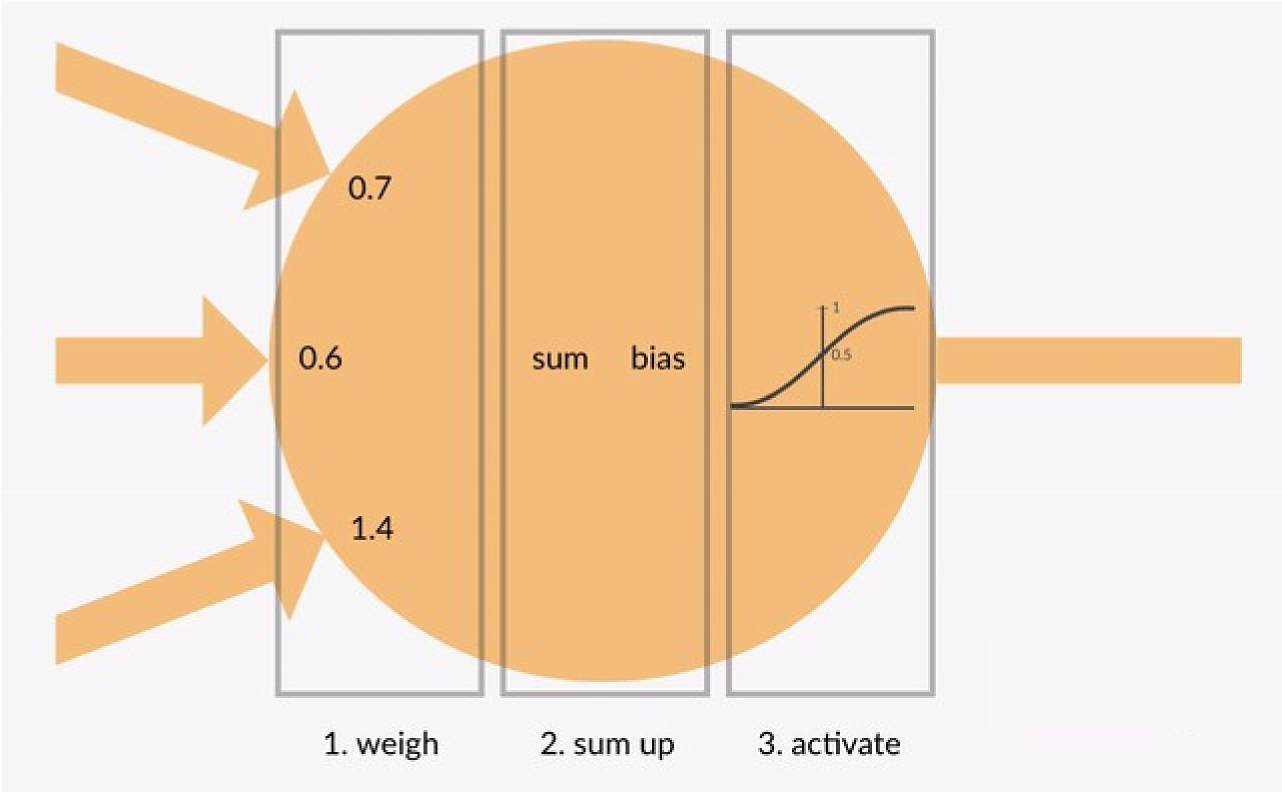
$F(x) = w1 \,.\, x1 + w2 \,.\, x2 + w3 \,.\, x3 + 1 \,.\, b$

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

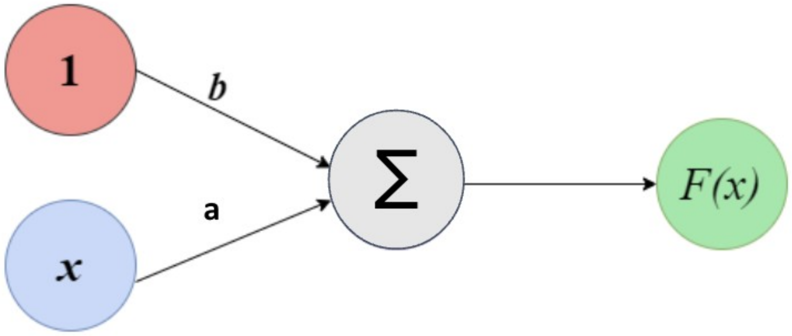
DL, Comment?

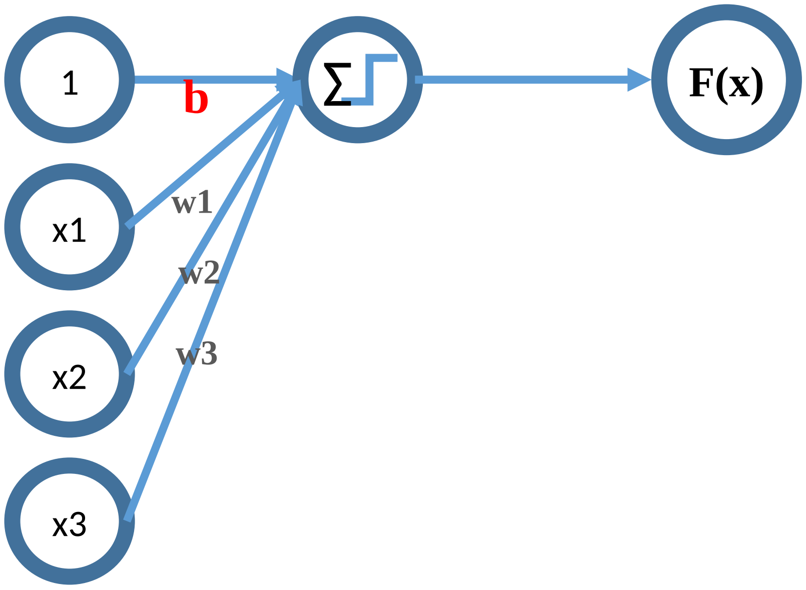


$$F(x) = \sigma(\; W.X + b \;)$$



DL, Comment?

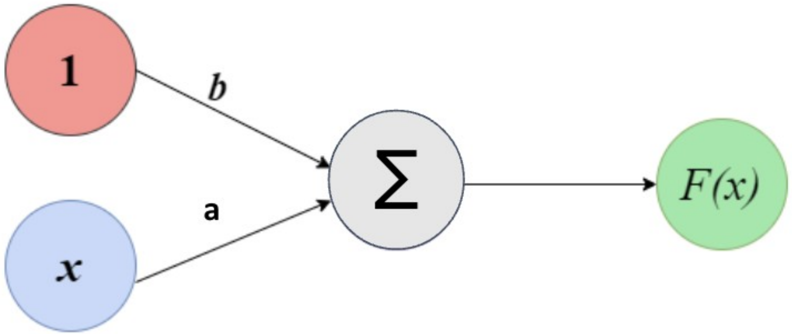


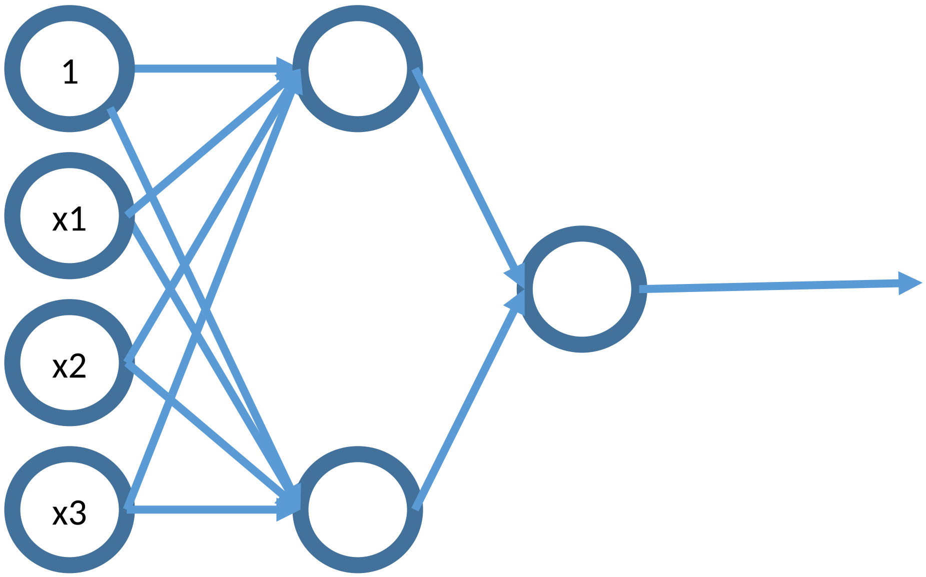


F(x) = a.x + b

F(x) = σ(W .X + b)

DL, Comment?

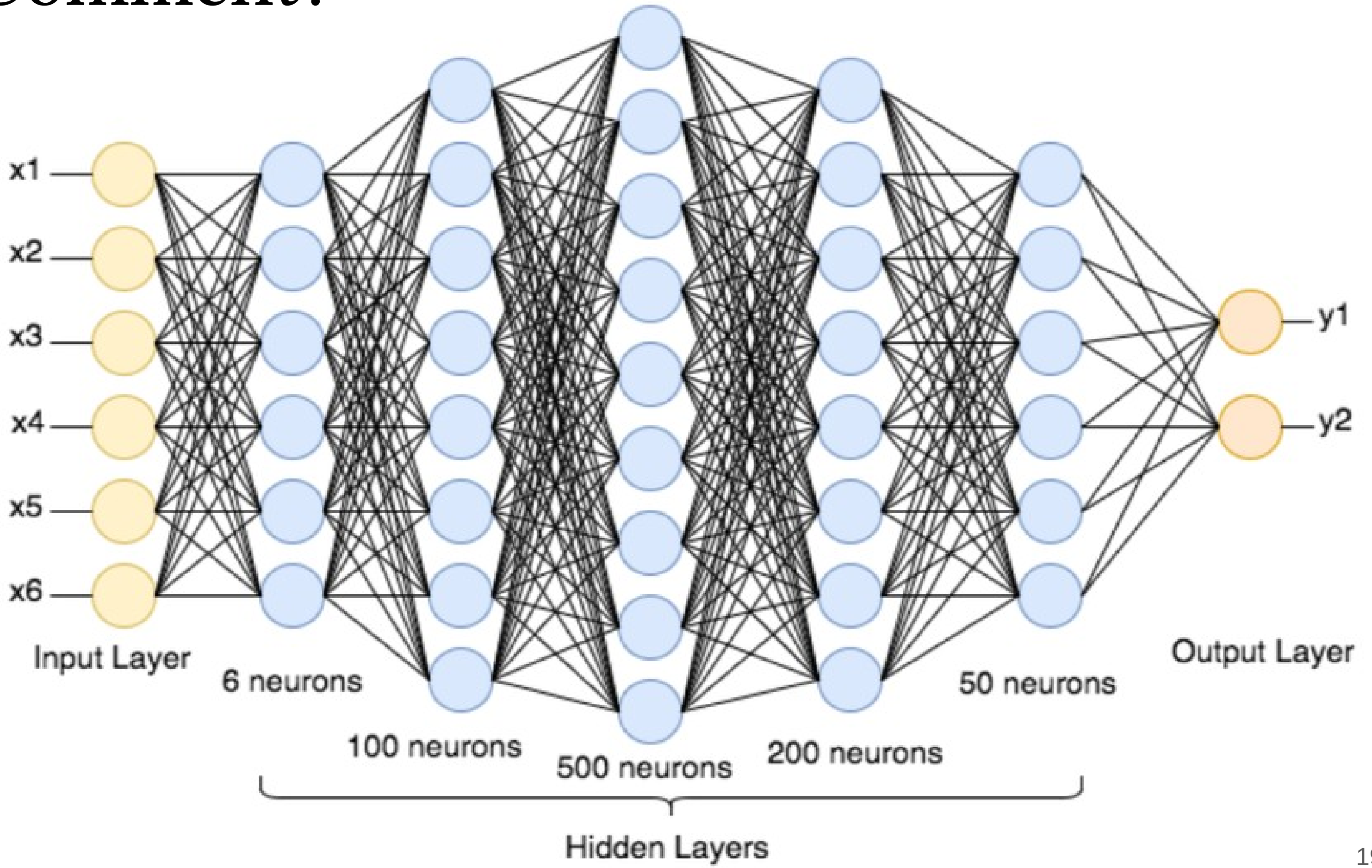




F(x) = a.x + b

σ(W .X + b)

DL, Comment?

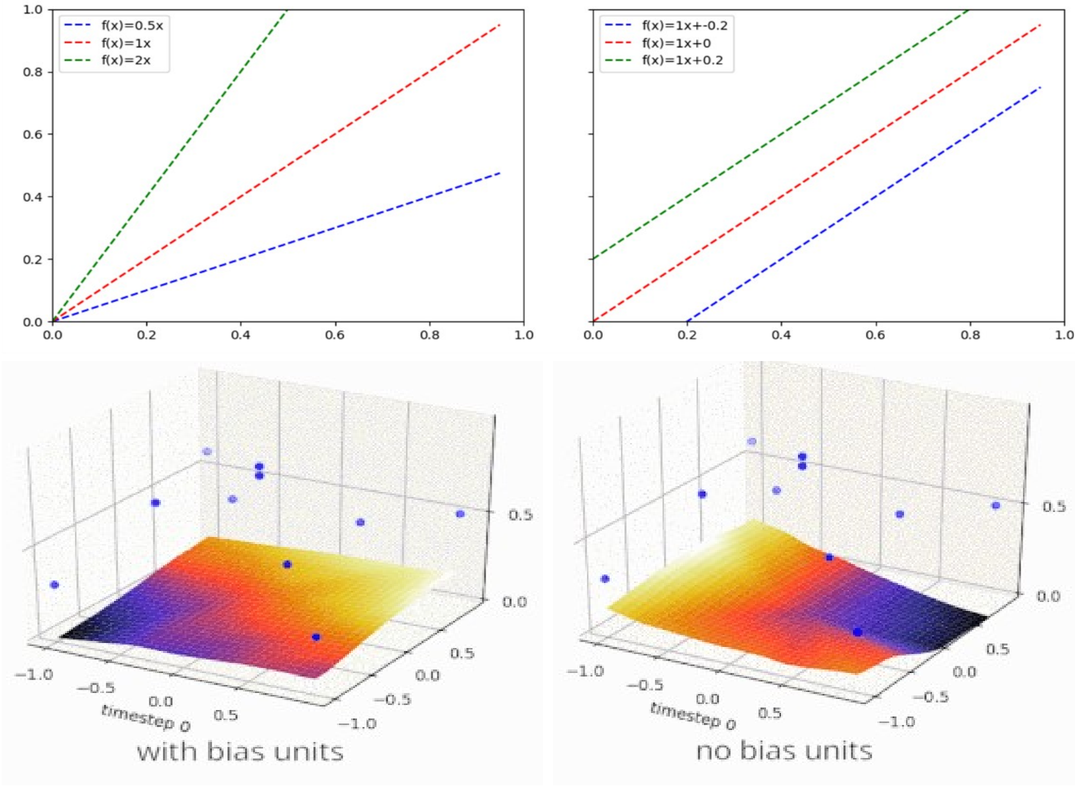

x1
x2
x3
x4
x5
x6
y1
y2
Input Layer
6 neurons
100 neurons
500 neurons
200 neurons
50 neurons
Output Layer
Hidden Layers

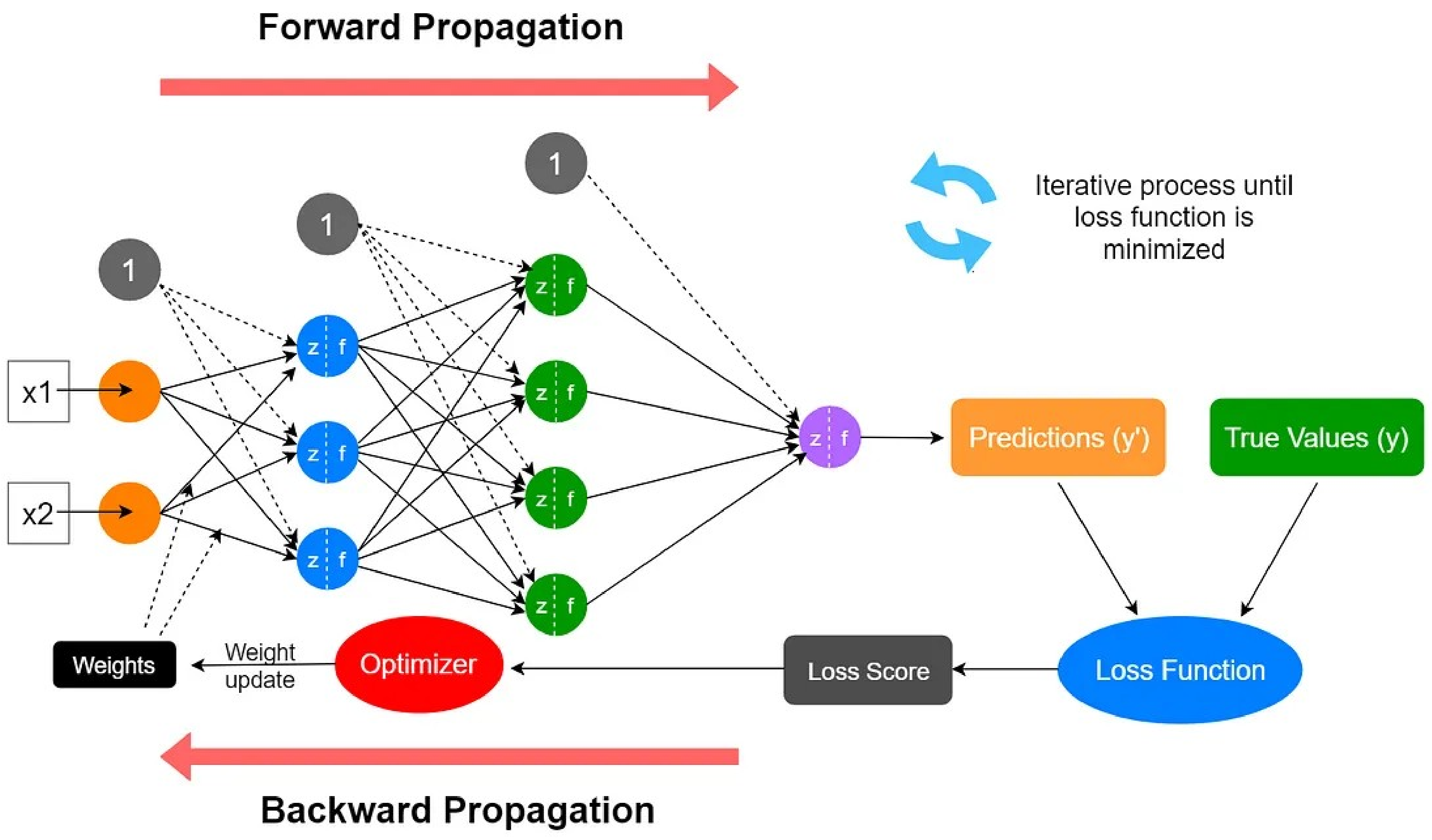
Biais

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

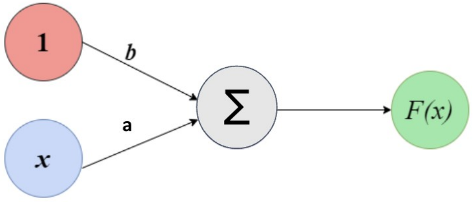
L'introduction de biais permet de déplacer la courbe de la fonction verticalement tout en laissant la forme/courbure inchangée.

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




Forward Propagation
Iterative process until
loss function is
minimized
1
1
1
x1
x2
z f
Predictions (y')
True Values (y)
Weights
Weight
update
Optimizer
Loss Score
Loss Function
Backward Propagation

DL, Objectif



Hypothèse :

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

Paramètres :

$$a, b$$

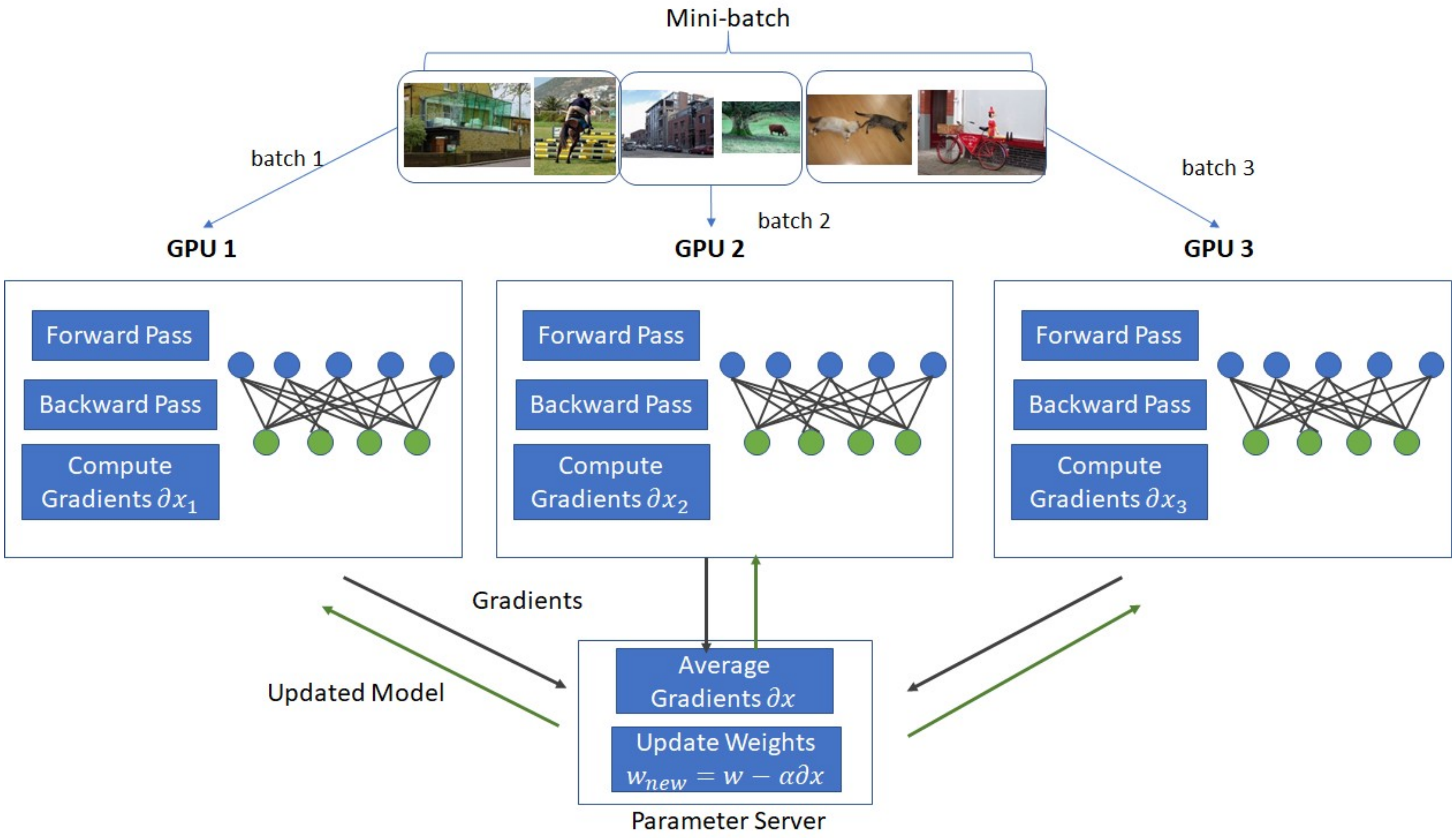
Fonction Coût :

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

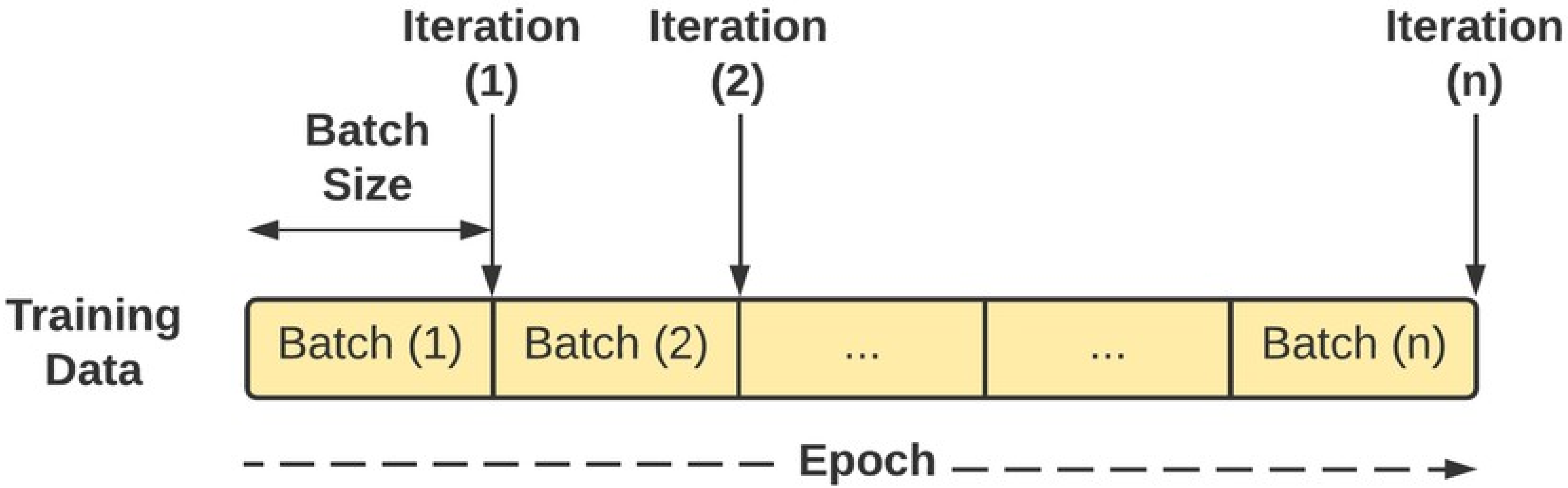
Objectif :

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

Parallélisation

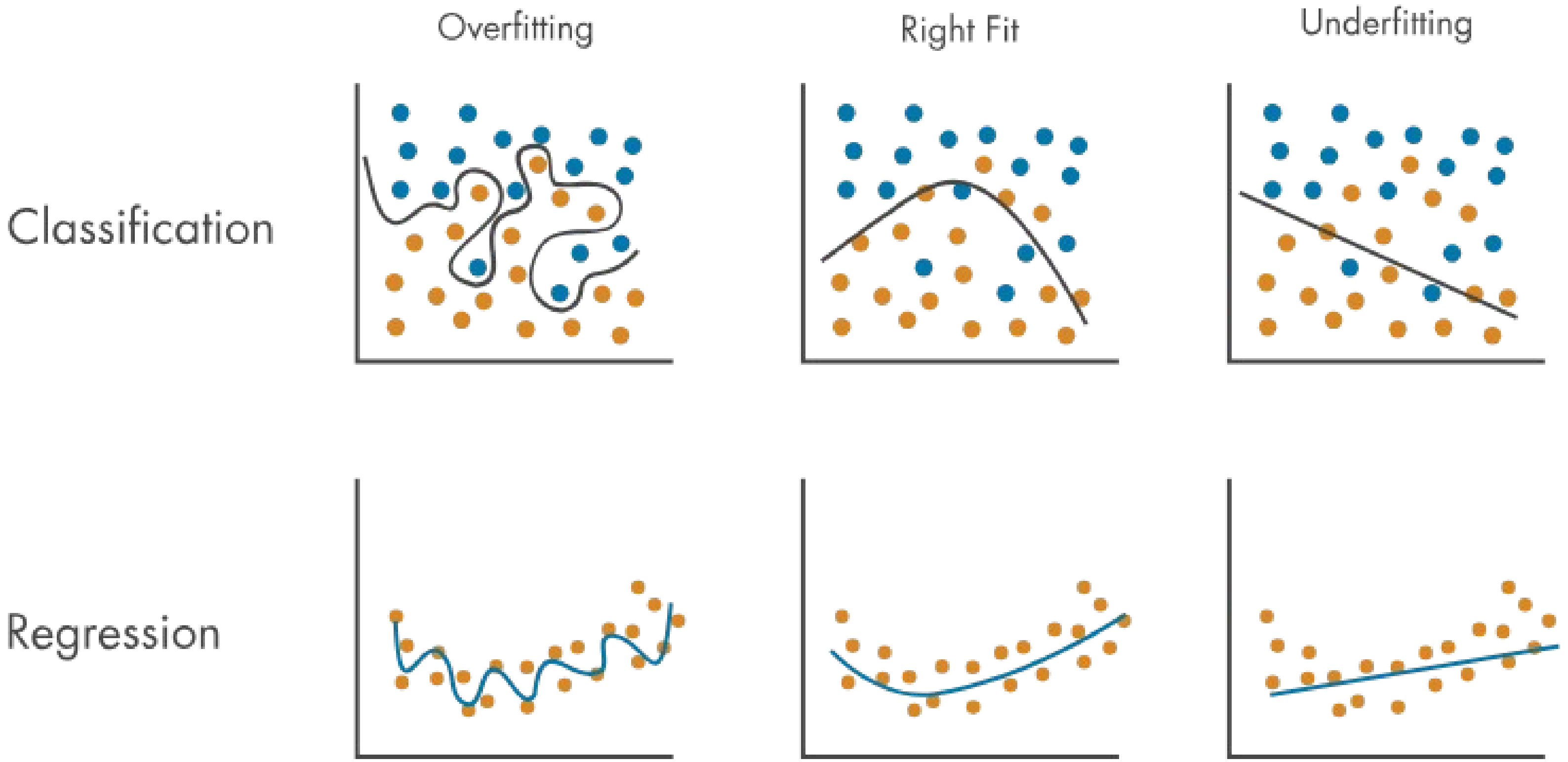


Iteration vs Epoch



Iteration vs Epoch

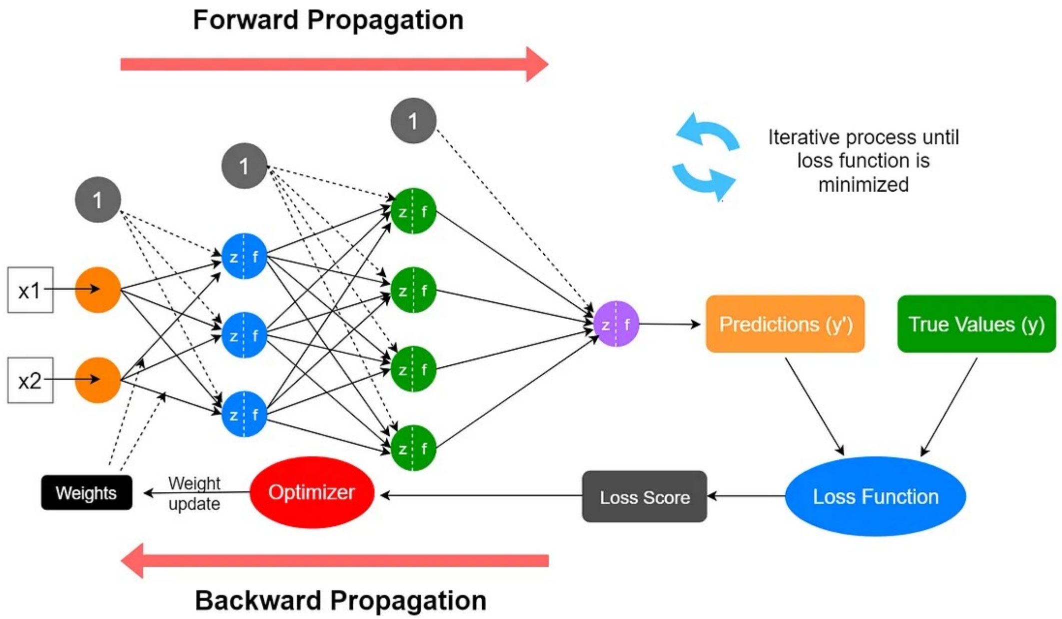
Overfitting



Overfitting, 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



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