



Université Constantine 2 Abdelhamid Mehri
Faculté des Nouvelles Technologies de l'Information et de la Communication
Département de l'Informatique Fondamentale et ses Applications

Module

Machine Learning and Computational Intelligence

MLCI

Unité d'enseignement: UEF3

Crédit: 3

Coefficient: 3

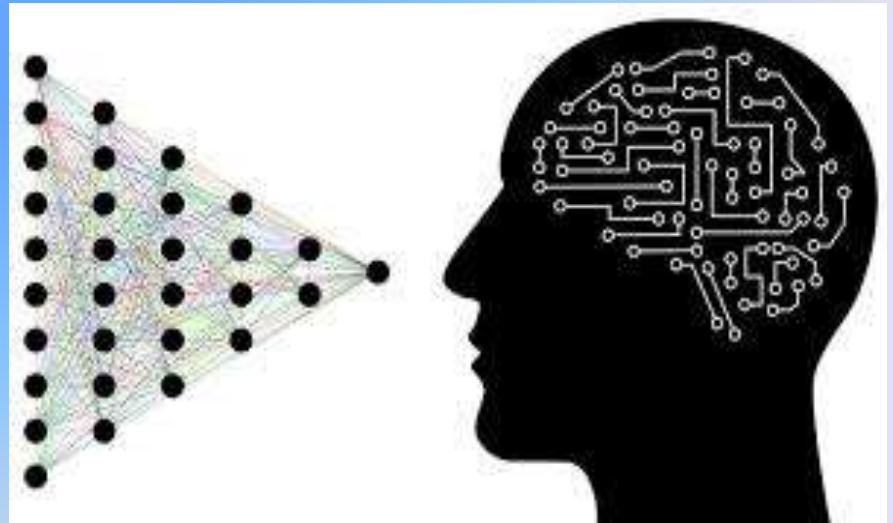
Cours: 1H30/semaine

TP: 1H30/semaine

Dr. Fergani

Baha.fergani@univ-constantine2.dz

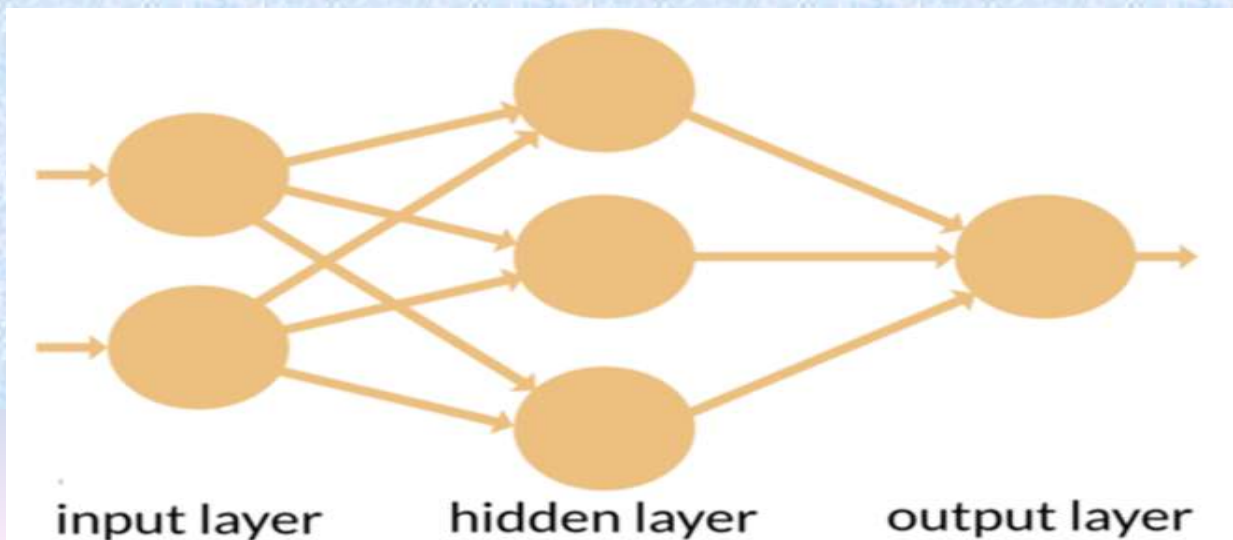
Apprentissage profond (Deep Learning)



Qu'est-ce qu'un Feed-Forward ?

Dans un réseau de neurones Feed-Forward:

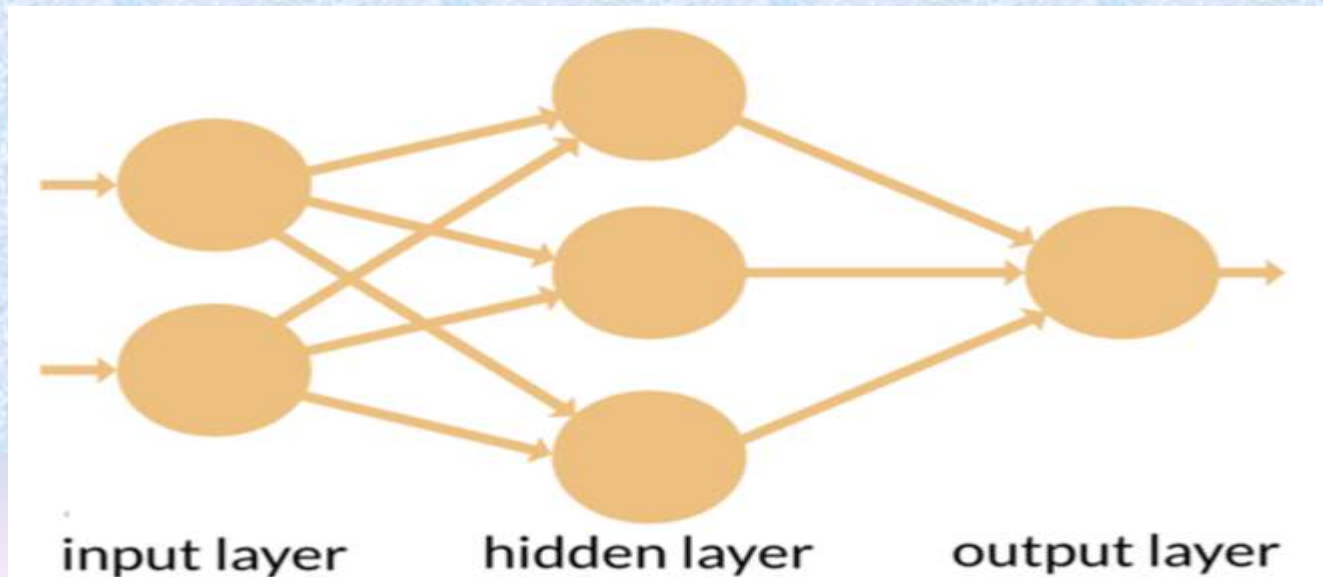
- L'information ne se déplace que dans **une direction**, de la couche d'entrée, à travers les couches cachées, à la couche de sortie.



Qu'est-ce qu'un Feed-Forward ?

Dans un réseau de neurones Feed-Forward:

- Pour cette raison, l'information ne touche jamais un nœud deux fois.
- Le calcul de la sortie d'une couche: $y_j^t = f(\sum_i W_{ji} x_i^t)$



Qu'est-ce qu'un Feed-Forward ?

Limites liées aux dépendances entre les données:



- Il est facile pour un être humain de reconnaître cette séquence malgré la rature.
- Car nous sommes capables d'analyser le contexte du reste de la phrase.

Qu'est-ce qu'un Feed-Forward ?

Limites liées aux dépendances entre les données

Voici un autre exemple : saurez-vous **deviner** le mot manquant dans cette séquence ?

"En me rasant ce matin, je me suis _____"

Là aussi, c'est très facile pour nous ! Il nous suffit de nous souvenir du début de la phrase pour en deviner la fin.

Mais pour un réseau de neurones, **comment procéder à la reconnaissance ?**

Qu'est-ce qu'un Feed-Forward ?

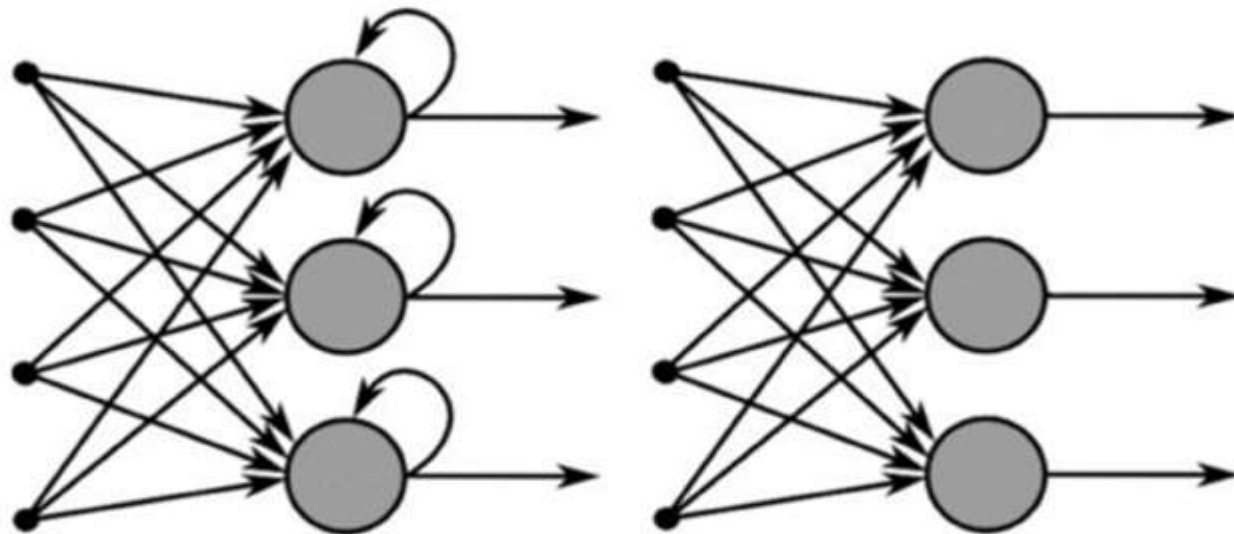
Mais pour un réseau de neurones, **comment procéder à la reconnaissance ?**



**Un réseau de neurones récurrent
RNN**

Qu'est-ce qu'un RNN ?

- Un réseau de neurones dont le graphe comporte au moins un **cycle**.
- Cela leur permet de **stocker** des informations lors du traitement de nouvelles entrées.



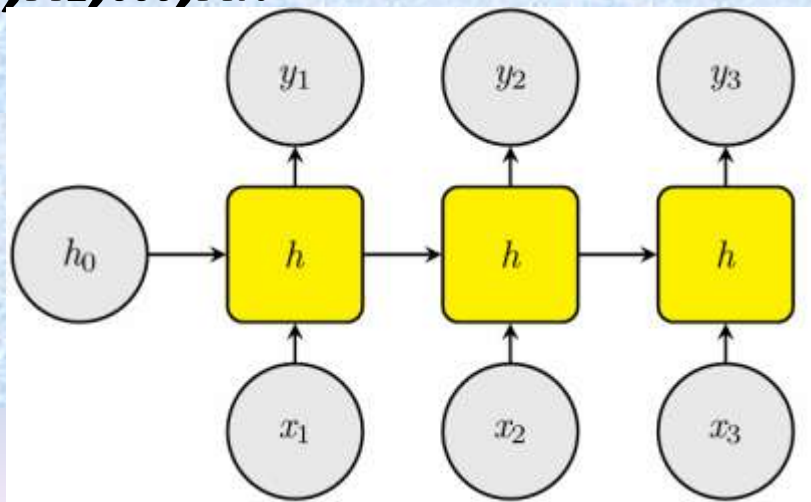
Recurrent Neural Network

Feed-Forward Neural Network

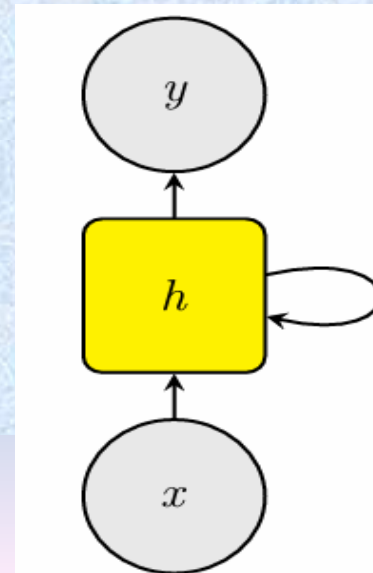
Qu'est-ce qu'un RNN ?

Idée :

- On a une série d'entrées $x_1, x_2 \dots x_n$.
- Une série de sorties $y_1, y_2, \dots y_n$.
- y_i dépend non seulement de x_i , mais aussi de x_0, x_1, \dots, x_i .



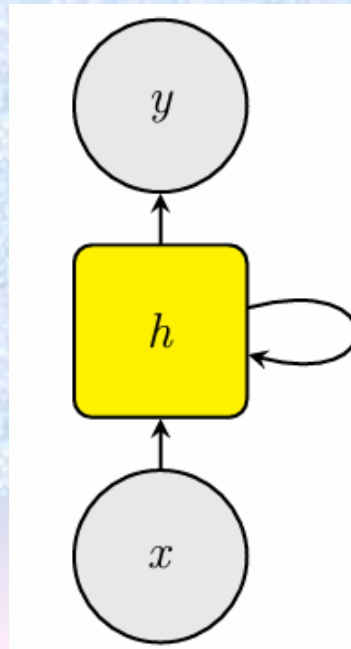
vue dépliée



Qu'est-ce qu'un RNN ?

Idée :

- On utilise un réseau où la valeur de la couche cachée h est réinjectée comme entrée à la couche cachée de l'étape suivante.



Qu'est-ce qu'un RNN ?

Les réseaux de neurones récurrents reposent sur deux principes :

- le premier principe est l'astuce de la **fenêtre glissante**.
- le second principe est l'utilisation de **connexions récurrentes** qui permettent **d'analyser la partie passée du signal**.

Qu'est-ce qu'un RNN ?

Un réseau neuronal récurrent est:

- Un réseau de neurones profond.
- Entraîné sur des données **séquentielles** ou des données de **séries temporelles**.
- Pour créer un modèle de Machine Learning capable de tirer des prédictions séquentielles ou de tirer des conclusions sur la base d'entrées séquentielles.

Qu'est-ce qu'un RNN ?

Un réseau neuronal récurrent est:

- Un réseau de neurones profond.
- Entraîné sur des données **séquentielles** ou des données de **séries temporelles**.
- Pour créer un modèle de Machine Learning capable de tirer des prédictions séquentielles ou de tirer des conclusions sur la base d'entrées séquentielles.

Comment fonctionnent les RNN?

- Les RNN utilisent leur « **mémoire** » :
Les informations des entrées antérieures ont une influence sur l'entrée et la sortie en cours.
- La sortie des RNN dépend des éléments antérieurs de la séquence.

Variantes d'architectures RNN

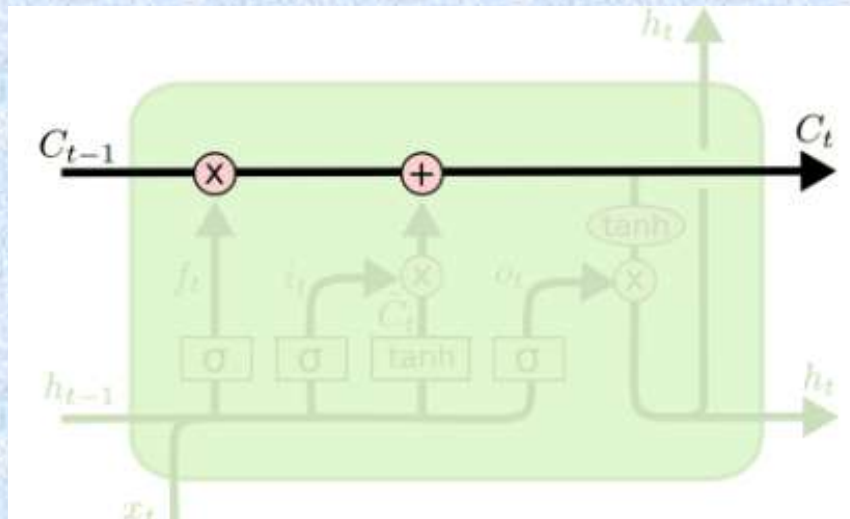
Les variantes d'architecture RNN populaires incluent

- Réseaux de neurones récurrents bidirectionnels (Bidirectional Recurrent Neural Networks: **BRRN**)
- Long Short-Term Memory (**LSTM**).
- Unités récurrentes fermées (Gated Recurrent Units:**GRU**).

Long Short-Term Memory (LSTM)

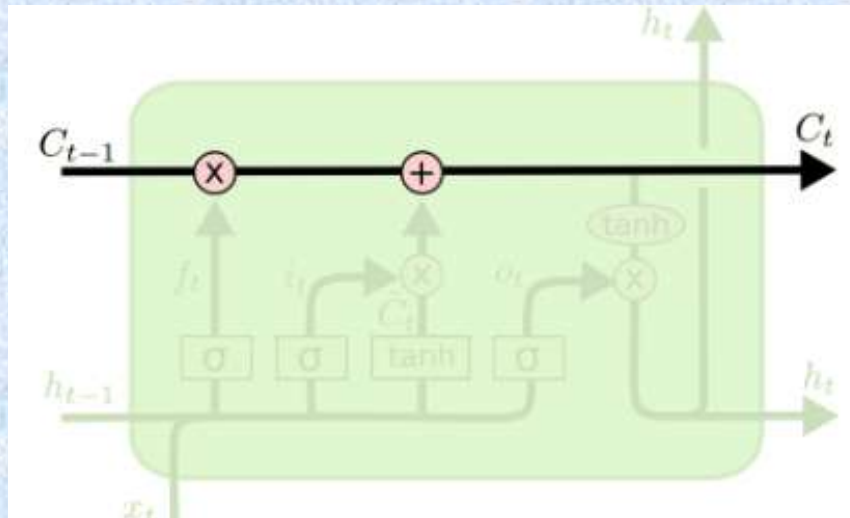
Long short-Term Memory (LSTM)

- **LSTM** est une architecture RNN populaire.
- La clé des LSTM est **l'état de la cellule**, la ligne horizontale qui traverse le haut du diagramme.



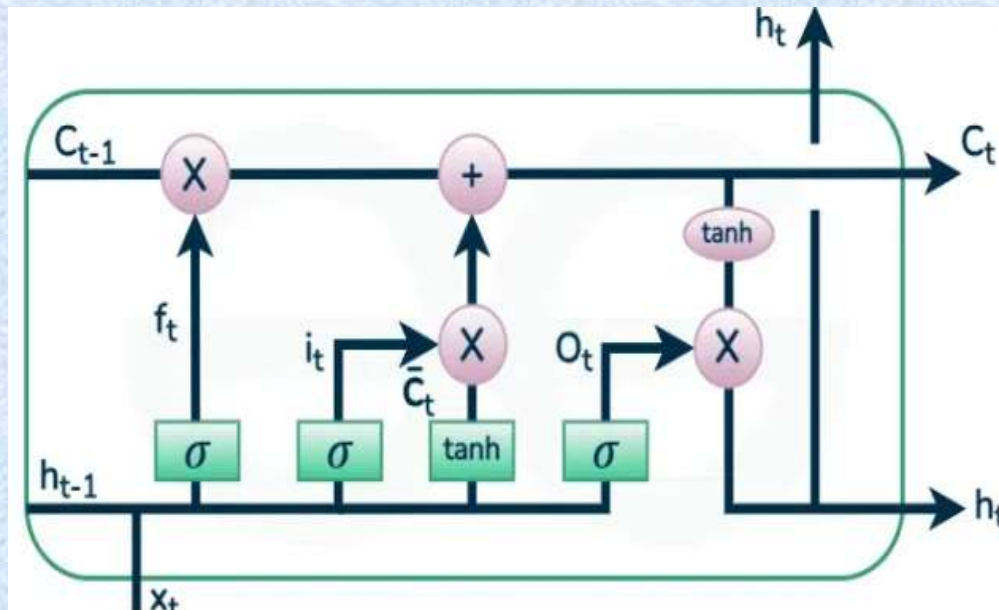
Long short-Term Memory (LSTM)

- Le LSTM a la capacité de **supprimer ou d'ajouter** des informations à l'état de la cellule, soigneusement régulées par des structures appelées portes.



Long short-Term Memory (LSTM)

- Les LSTM possèdent **trois** portes:
 1. Une porte d'entrée.
 2. Une porte de sortie.
 3. Une porte d'oubli.



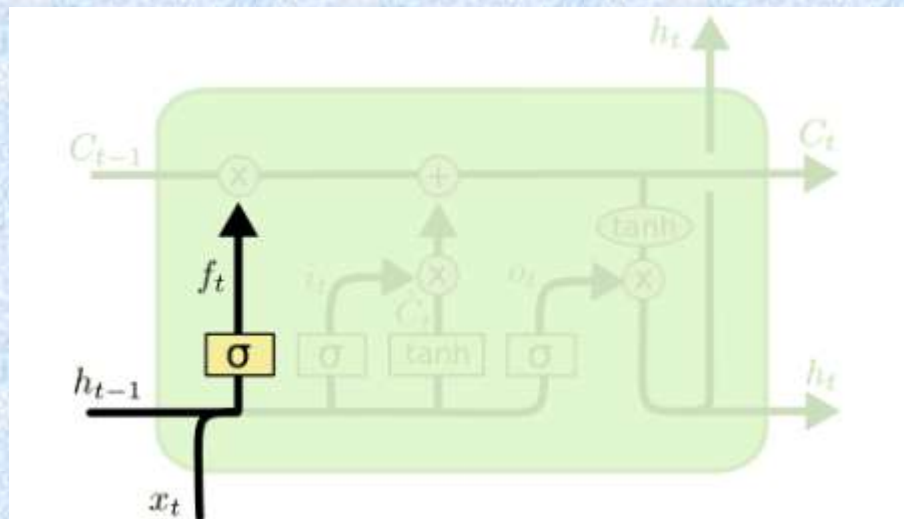
Long short-Term Memory (LSTM)

- La porte **d'entrée** contrôle les informations ajoutées à la cellule de mémoire.
- La porte **d'oubli** contrôle les informations supprimées de la cellule de mémoire.
- La porte de **sortie** contrôle les informations générées par la cellule de mémoire.

Long short-Term Mmemory (LSTM)

La porte d'oubli:

- Les informations qui ne sont plus utiles dans l'état de la cellule sont supprimées avec la porte d'oubli.
- La première étape consiste à décider quelles informations nous allons jeter de l'état cellulaire.

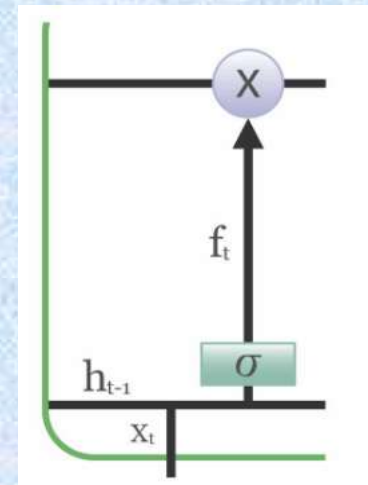


Long short-Term Memory (LSTM)

La porte **d'oubli**:

- Deux entrées x_t (entrée à t) et h_{t-1} (sortie de cellule précédente) sont introduites dans la porte et multipliées par des matrices de poids suivies de l'ajout de biais.

$$(W_f \cdot [h_{t-1}, x_t] + b_f)$$



W_f représente la matrice de poids associée à la porte d'oubli.

$[h_{t-1}, x_t]$ désigne la concaténation de l'entrée actuelle et de l'état caché précédent.

b_f est le biais avec la porte d'oubli.

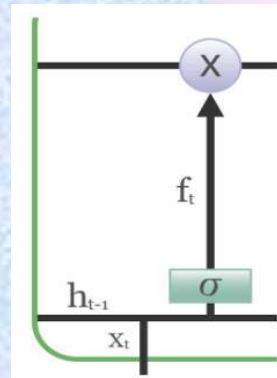
Long short-Term Memory (LSTM)

La porte d'oubli:

- La résultante est passée par une fonction d'activation qui donne une sortie binaire.
- Si, pour un état de cellule particulier, la sortie est 0, l'élément d'information est **oublié**.
- Pour la sortie 1, l'information est conservée pour une utilisation ultérieure.
- L'équation de la porte d'oubli est la suivante :

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$$

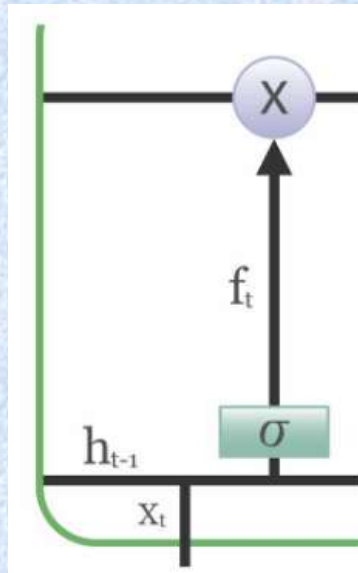
σ est la fonction d'activation sigmoïde.



Long short-Term Mmemory (LSTM)

Par exemple:

Si les **pronoms de genre**, tels que « **elle** », ont été répétés plusieurs fois dans les phrases précédentes, vous pouvez les exclure de l'état de la cellule.

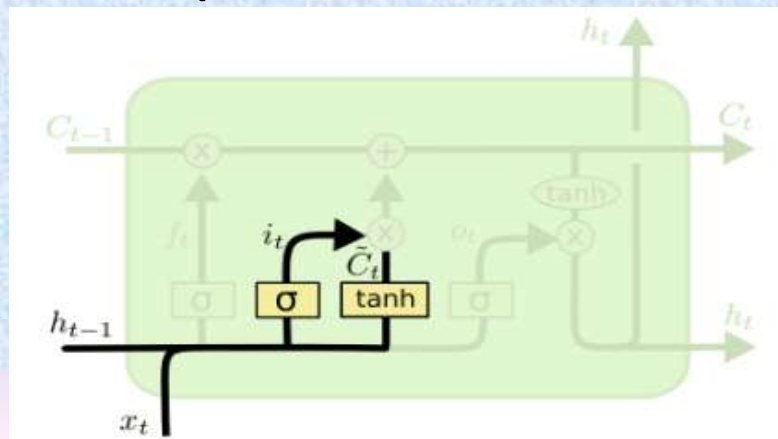


Long short-Term Memory (LSTM)

La porte d'entrée:

- L'étape suivante consiste à décider quelles nouvelles informations nous allons stocker dans l'état de la cellule.
$$\mathbf{i}_t = \sigma(W_{Je} \cdot [h_{t-1}, x_t] + b_{Je})$$

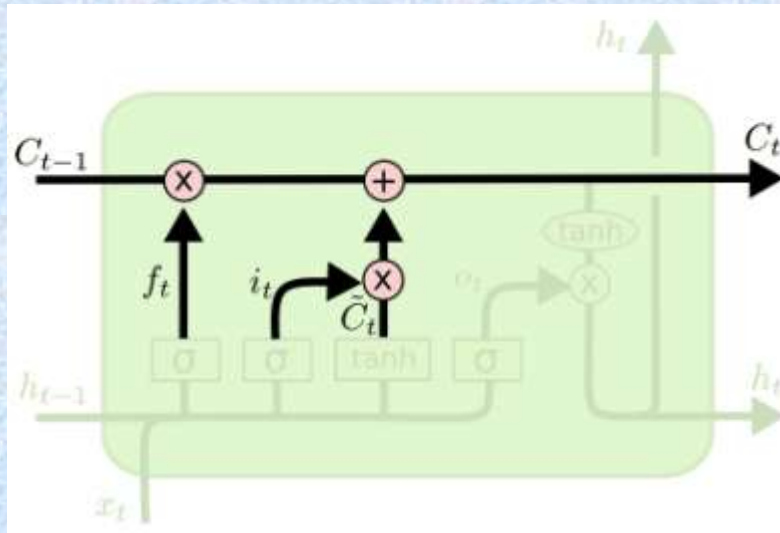
- Elle comporte deux parties. Tout d'abord, une couche sigmoïde appelée « couche de porte d'entrée » décide des valeurs que nous allons mettre à jour.



Long short-Term Memory (LSTM)

- Ensuite, une couche de \tanh crée un vecteur de nouvelles valeurs candidates, qui pourraient être ajoutés à l'État.

$$\hat{C}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c)$$

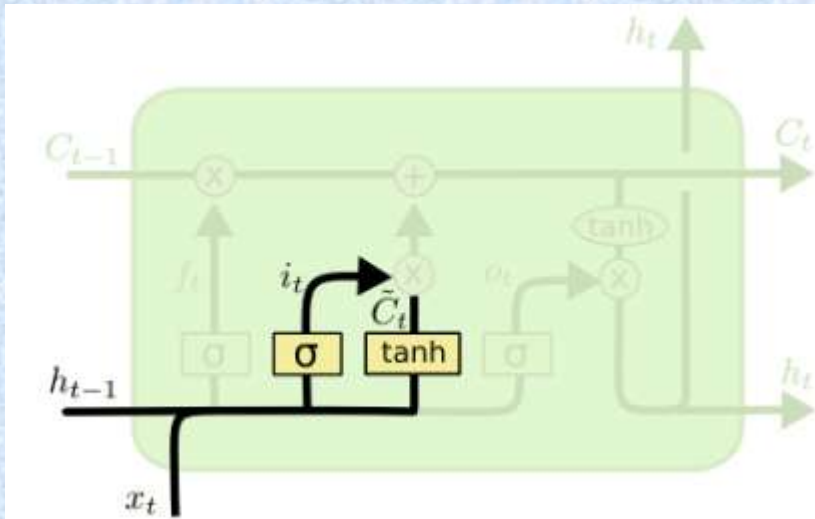


Long short-Term Memory (LSTM)

- Combiner ces deux éléments pour créer une mise à jour de l'état.

$$C_t = f_t \odot C_{t-1} + \mathbf{i}_t \odot \hat{C}_t$$

\odot désigne la multiplication élément par élément



$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$$

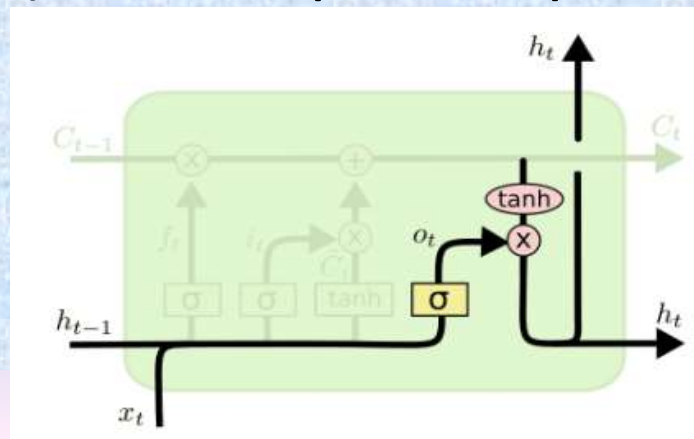
Long short-Term Memory (LSTM)

Porte de sortie:

- Tout d'abord, nous exécutons une couche **sigmoïde** qui décide des parties de l'état cellulaire que nous allons produire.
- Ensuite, nous passons l'état de la cellule à travers \tanh (pour pousser les valeurs à être comprises entre -1 et 1) et multipliez-le par la sortie de la porte sigmoïde.

$$o_t = \sigma(W_o [h_{t-1}, x_t] + b_o)$$

$$h_t = o_t * \tanh(C_t)$$



Applications du LSTM

Applications du LSTM

Certaines des applications célèbres de LSTM comprennent :

- **Modélisation du langage** : Les LSTM ont été utilisés pour des tâches de traitement du langage naturel telles que:
 - La modélisation du langage.
 - La traduction automatique.
 - La synthèse de texte.

Applications du LSTM

Certaines des applications célèbres de LSTM comprennent :

- **Reconnaissance vocale** : Les LSTM ont été utilisés pour des tâches de reconnaissance vocale telles que la transcription de la parole en texte et la reconnaissance de commandes vocales. Ils peuvent être entraînés à reconnaître des modèles dans le discours et à les faire correspondre au texte correspondant.

Applications du LSTM

Certaines des applications célèbres de LSTM comprennent :

- **Systèmes de recommandation** : Les LSTM ont été utilisés pour des tâches de recommandation telles que la recommandation de films, de musique et de livres. Ils peuvent apprendre des modèles de comportement des utilisateurs et les utiliser pour faire des recommandations personnalisées.

Applications du LSTM

Certaines des applications célèbres de LSTM comprennent :

- **Analyse vidéo** : Les LSTM ont été utilisés pour des tâches d'analyse vidéo telles que la détection d'objets, la reconnaissance d'activité et la classification d'actions.

Ils peuvent être utilisés en combinaison avec d'autres architectures de réseaux neuronaux, tels que les CNN, pour analyser les données vidéo et extraire des informations utiles.

LSTM / RNN

Caractéristique	LSTM (mémoire à long terme et à court terme)	RNN (réseau neuronal récurrent)
Mémoire	Dispose d'une unité de mémoire spéciale qui lui permet d'apprendre les dépendances à long terme dans les données séquentielles	N'a pas d'unité de mémoire
Formation	Plus difficile à entraîner que RNN en raison de la complexité des portes et de l'unité de mémoire	Plus facile à entraîner que le LSTM

LSTM / RNN

Caractéristique	LSTM (mémoire à long terme et à court terme)	RNN (réseau neuronal récurrent)
Apprentissage de la dépendance à long terme	Oui	Limité
Capacité à apprendre des données séquentielles	Oui	Oui
Applications	Traduction automatique, reconnaissance vocale, synthèse de textes, traitement du langage naturel, prévision de séries chronologiques	Traitement du langage naturel, traduction automatique, reconnaissance vocale, traitement d'images, traitement vidéo

Réseaux de neurones récurrents bidirectionnels (Bidirectional Recurrent Neural Networks: BRRN)

RNN bidirectionnels

Réseaux de neurones récurrents bidirectionnels

- **RNN unidirectionnels** ne peuvent s'appuyer que sur des entrées antérieures pour faire des prédictions sur l'état actuel.
- **Les RNN bidirectionnels**, ou BRNN, extraient des données postérieures pour en améliorer la précision.
- Pour revenir à l'exemple « se sentir grippé », un modèle basé sur un BRNN peut mieux prédire que le deuxième mot de cette phrase est « sentir » s'il sait que le dernier mot de la séquence est « grippé ».

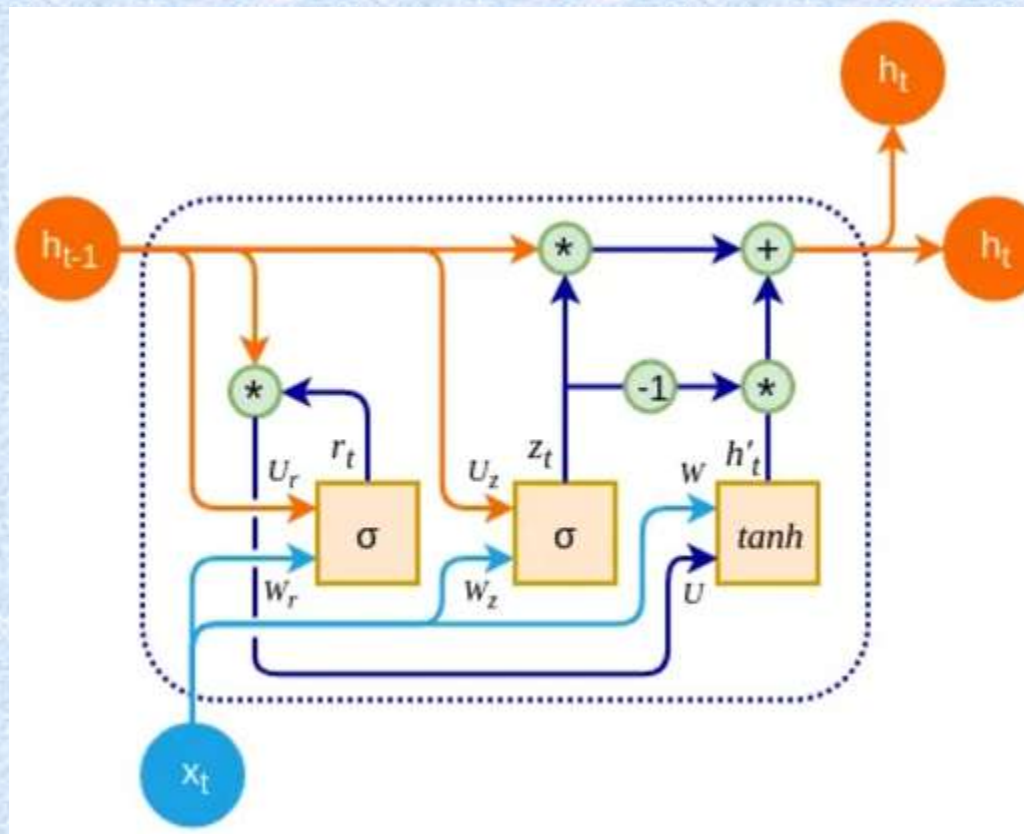
Unités récurrentes fermées (Gated Recurrent Units) (GRU)

Unités récurrentes fermées: GRU

- Une **GRU** est similaire à une **LSTM**, car elle permet également de résoudre le problème de mémoire à court terme des modèles RNN.
- Au lieu de trois portes, il en a **deux** :
 - Une porte de réinitialisation: elle contrôle quelles informations doivent être conservées.
 - Une porte de mise à jour: elle contrôle la quantité d'informations.

Le vecteur d'états complet est sorti à chaque étape temporelle.

Unités récurrentes fermées: GRU



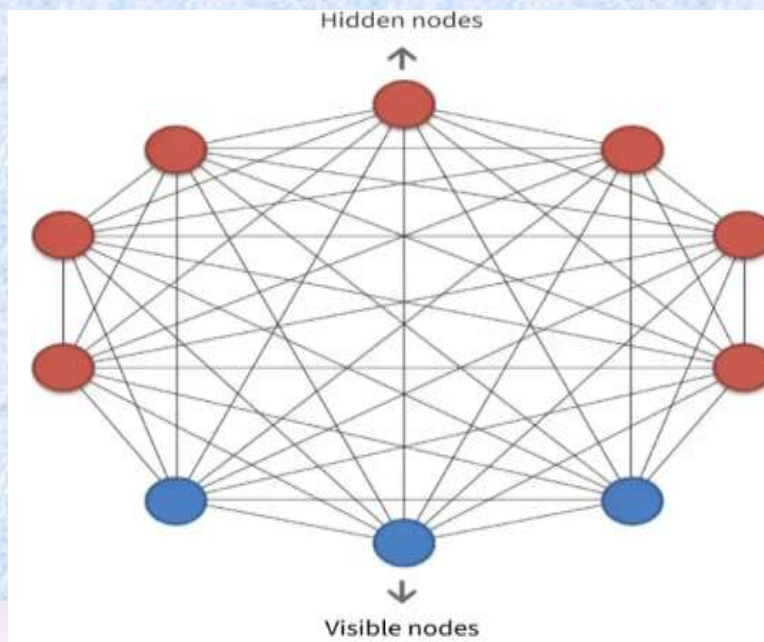
Machine de Boltzmann

(Boltzmann Machine ou BM)

Machine de Boltzmann

Une machine Boltzmann est:

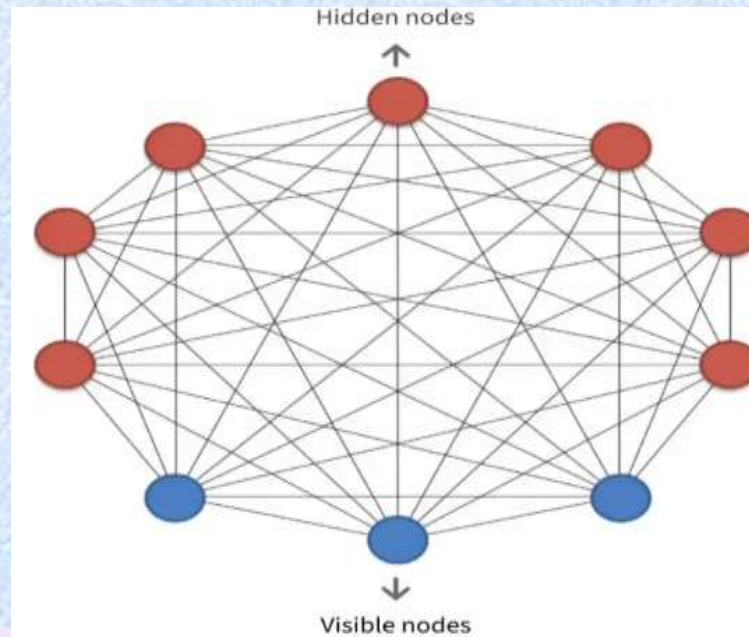
- Un **modèle d'apprentissage profond** non supervisé.
- Dans lequel **chaque nœud** est connecté à tous les autres nœuds.



Machine de Boltzmann

Une machine Boltzmann est:

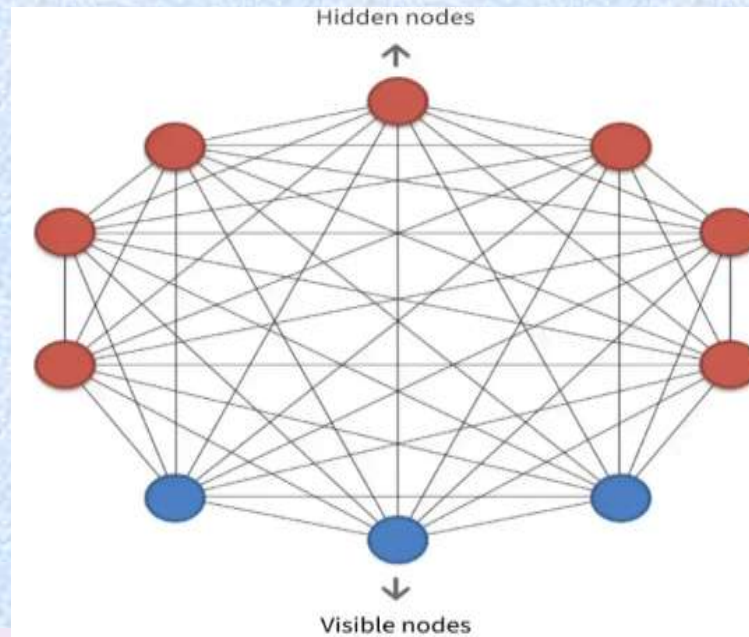
- Un modèle non orienté: il n'a pas de direction dans les connexions.



Machine de Boltzmann

Une machine Boltzmann a **deux types** de nœuds

- **Nœuds visibles :**
il s'agit de nœuds qui peuvent être mesurés et qui sont mesurés.

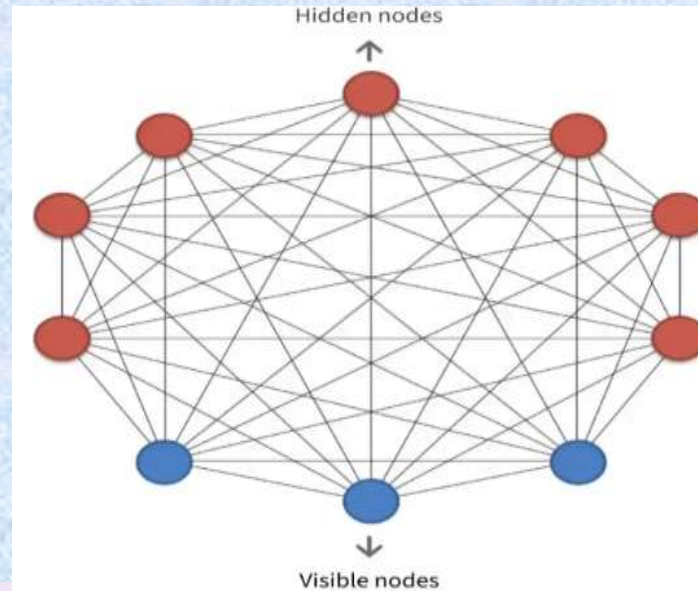


Machine de Boltzmann

Une machine Boltzmann a **deux types** de nœuds

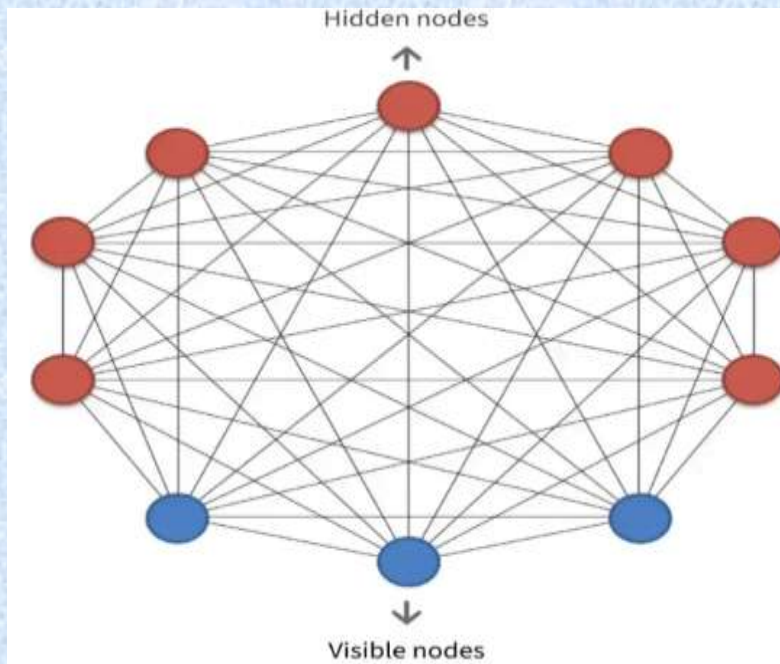
- **Nœuds cachés :**

Il s'agit de nœuds qui ne peuvent pas être mesurés ou qui ne le sont pas.



Machine de Boltzmann

- Chaque **nœud** est connecté à tous les autres nœuds, qu'il s'agisse de nœuds d'entrée ou de nœuds cachés.

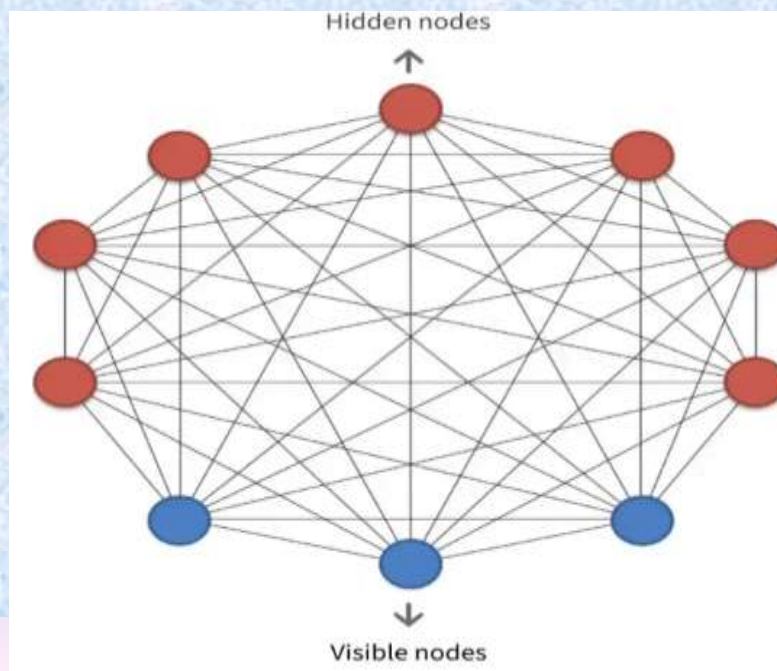
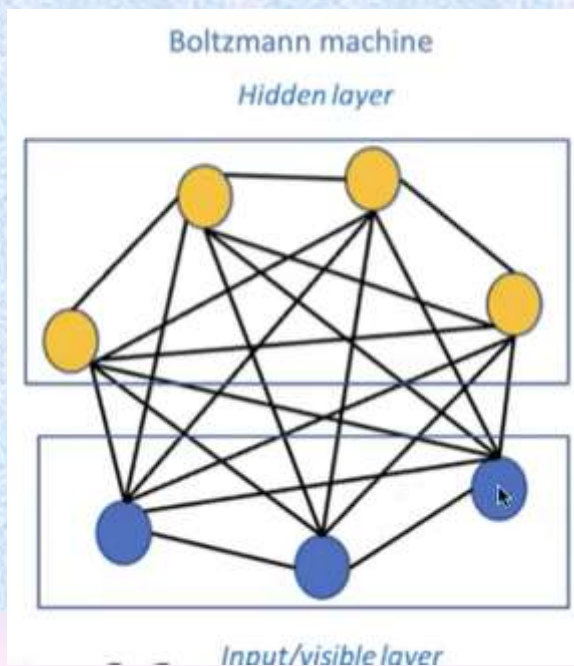


Machine de Boltzmann

- Les nœuds visibles prennent en charge l'entrée.
- Les mêmes nœuds qui prennent l'entrée renverront l'entrée reconstruite en tant que sortie.
- Ceci est réalisé grâce à des **poids bidirectionnels** qui se propagent à l'envers et rendent la sortie sur les nœuds visibles.

Machine de Boltzmann

- Cela leur permet de partager des informations entre eux et de générer eux-mêmes des données ultérieures.



Quelles sont les applications des BM?

Quelles sont les applications des BM?

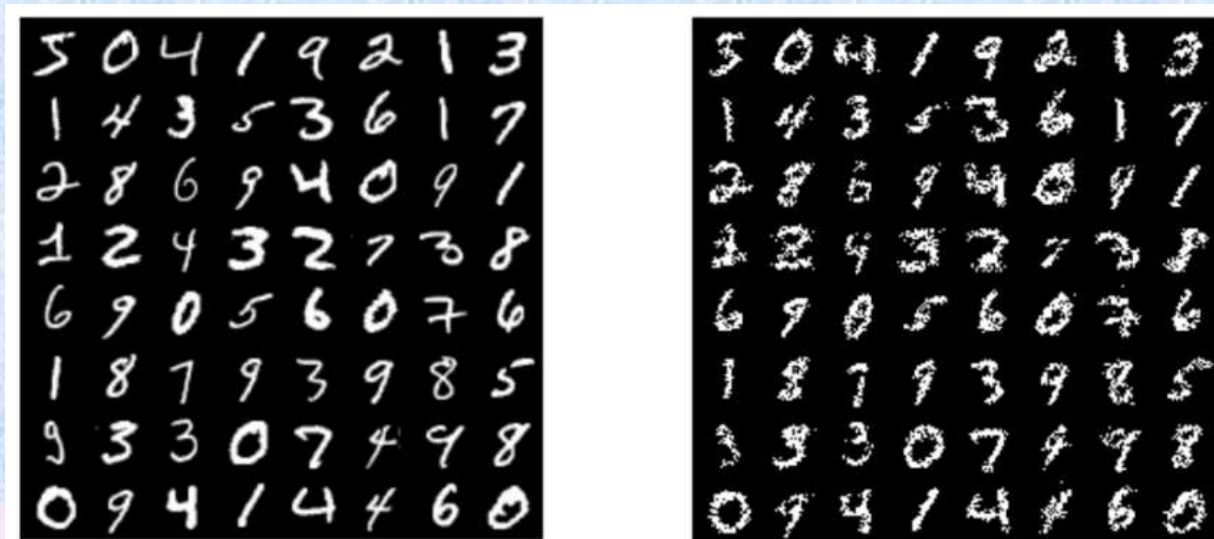
Réduction de la dimensionnalité :

- Les machines Boltzmann peuvent être utilisées pour des tâches d'apprentissage non supervisées, telles que l'apprentissage de représentations compactes ou l'extraction de caractéristiques à partir de données de grande dimension.

Quelles sont les applications des BM?

Modélisation générative :

- **Les** machines Boltzmann peuvent être utilisées pour modéliser et générer de nouveaux échantillons qui ressemblent aux données d'entraînement.



Hyper-paramètres pour les machines Boltzmann

Initialisation du poids :

- L'initialisation des poids est une étape importante dans le processus d'entraînement.
- Une initialisation correcte des poids peut faire gagner beaucoup de temps car elle peut optimiser le temps nécessaire à l'apprentissage de ces poids, ce qui est toute l'idée de l'entraînement d'un réseau. Au fur et à mesure que les poids s'améliorent, le réseau peut envoyer de meilleurs signaux pour rendre ses résultats de classification plus précis.

Hyper-paramètres pour les machines Boltzmann

Unités visibles et cachées :

- Le nombre d'entrées est la caractéristique qui est explicitement donnée au réseau.
- Le nombre de fonctionnalités cachées doit être choisi de manière optimale pour que le réseau saisisse une majorité de fonctionnalités.
- Chacune de ces couches a sa propre fonction de transformation pour traiter les entrées et les passer à la couche suivante.

Quels sont les types de machines Boltzmann ?

Types of Boltzmann machines



Restricted
Boltzmann Machines
(RBMs)



Deep Belief
Networks (DBNs)



Deep Boltzmann
Machines (DBMs)

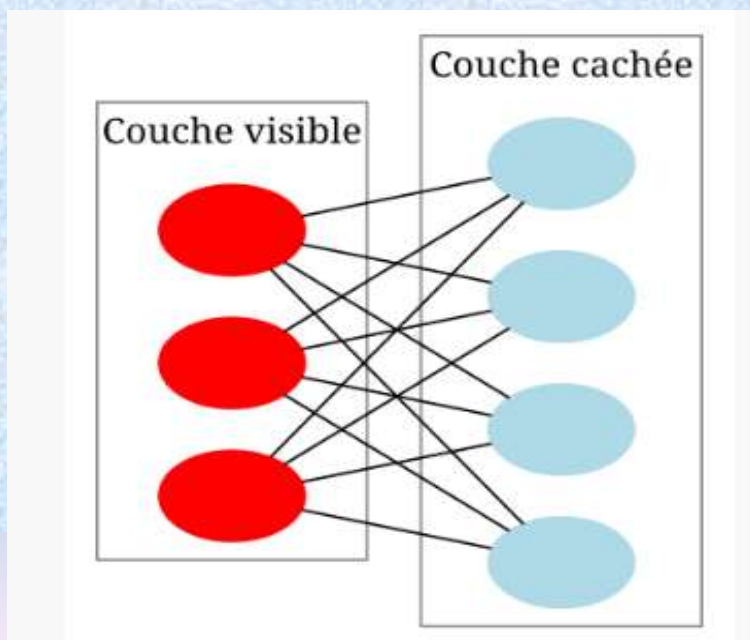
Il existe trois types de machines Boltzmann. Il s'agit de :
Machines **Boltzmann restreintes** (RBM)
Réseaux de **Croyances Profondes** (DBNs)
Machines **Boltzmann profondes** (DBM)

Machine de Boltzmann restreinte (RBM)

Machine de Boltzmann restreinte

La machine de Boltzmann restreinte est:

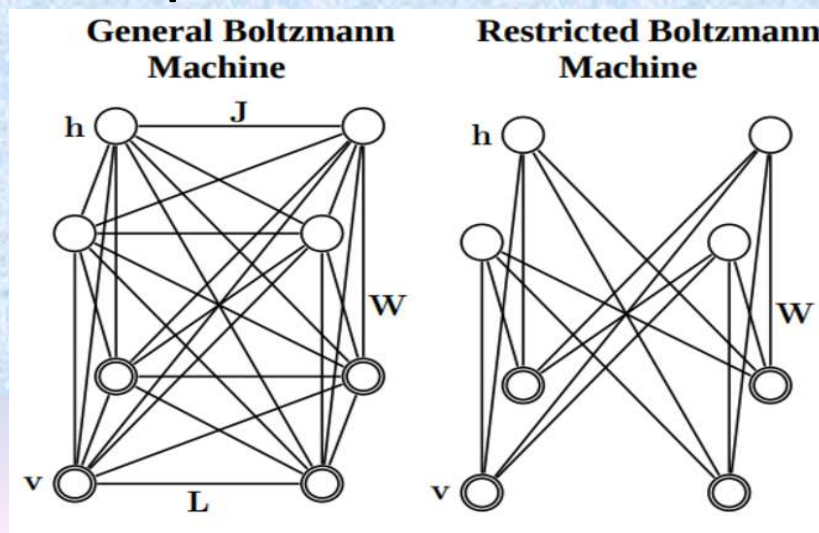
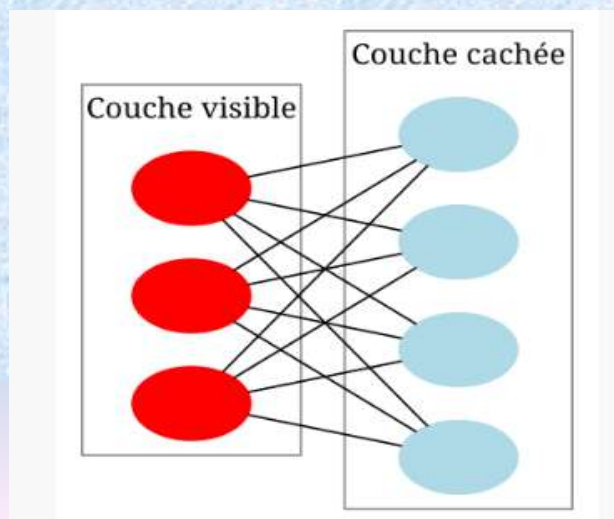
- Constituée de deux couches de neurones : **une couche visible** et une couche cachée.



Machine de Boltzmann restreinte

Les unités visibles

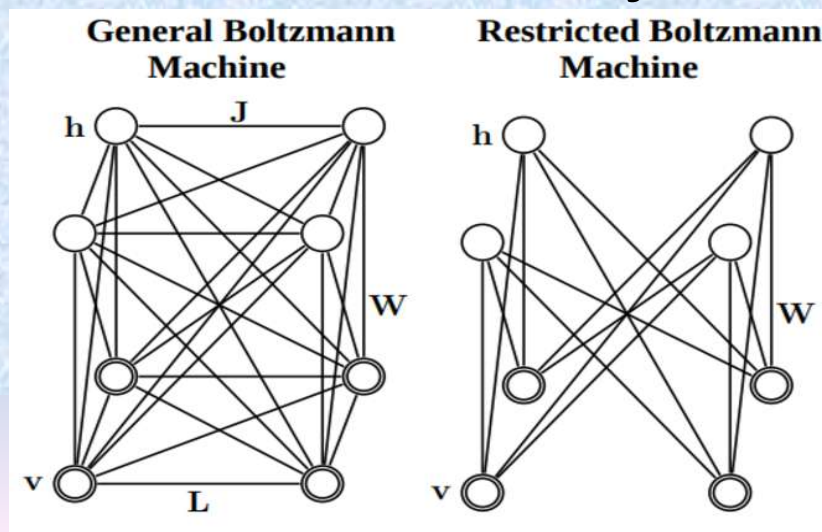
- Constituent la première couche.
- Connectés aux neurones cachés, sans connexion au sein de chaque couche.



Machine de Boltzmann restreinte

Les unités visibles

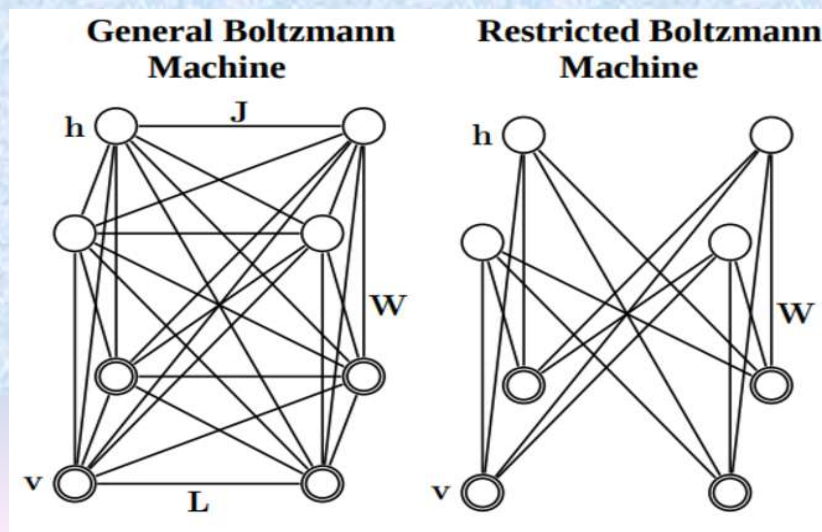
- Constituent la première couche.
- Chaque nœud visible prend une caractéristique de bas niveau d'un élément du jeu de données à apprendre.



Machine de Boltzmann restreinte

Les unités visibles

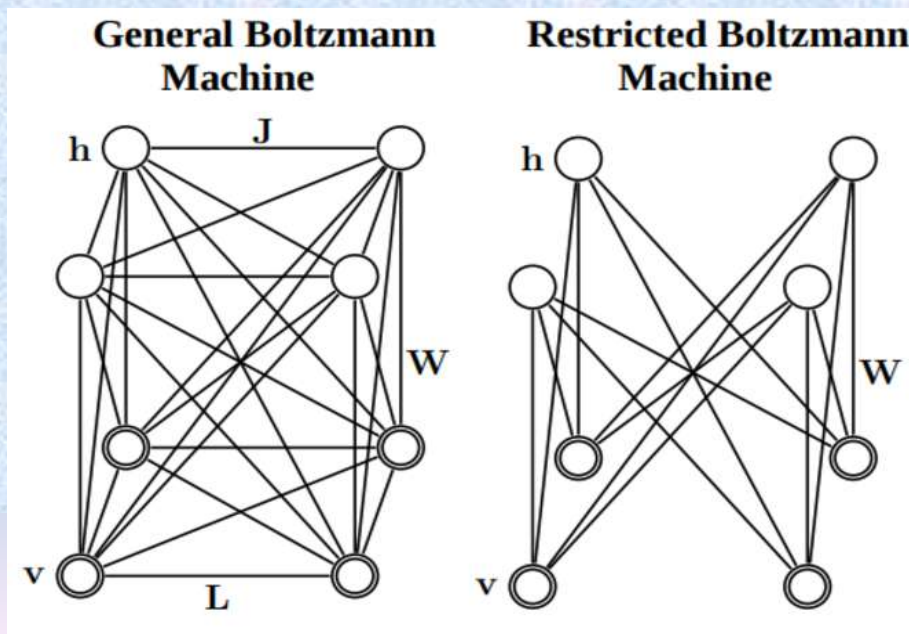
- Elles correspondent aux composantes d'une observation (par exemple, une unité visible pour chaque pixel d'une image d'entrée numérique).



Machine de Boltzmann restreinte

Les **unités cachées** modélisent:

- Les dépendances entre les composants des observations (par exemple, les dépendances entre les pixels des images).



Comment fonctionnent les RBM ?

Étape 1: Configuration initiale avec deux couches

- Les RBMs commencent avec deux couches distinctes : la couche visible, qui est le point d'entrée pour les données d'entrée.
- La couche cachée est responsable de la détection des caractéristiques.
- Ces couches sont interconnectées.

Comment fonctionnent les RBM?

Étape 2 : Traitement des données à travers la couche visible

- Les données d'entrée sont alimentées dans la couche visible.
- Chaque neurone de cette couche représente une caractéristique des données d'entrée, comme un pixel dans une image ou un mot dans les données textuelles.

Comment fonctionnent les RBM?

Étape 2 : Traitement des données à travers la couche visible

Par exemple:

- A partir d'un ensemble de données d'images en niveaux de gris, chaque nœud visible reçoit une valeur de pixel pour chaque pixel d'une image.
- Les images MNIST ont 784 pixels, donc les réseaux neuronaux qui les traitent doivent avoir 784 nœuds d'entrée sur la couche visible.

Comment fonctionnent les RBM?

Étape 3: Détection des fonctionnalités dans la couche cachée

- Les données de la couche visible sont ensuite transmises à la couche cachée.
- La RBM commence à reconnaître des motifs et les caractéristiques dans les données d'entrée, qui ne sont pas explicitement étiquetées.

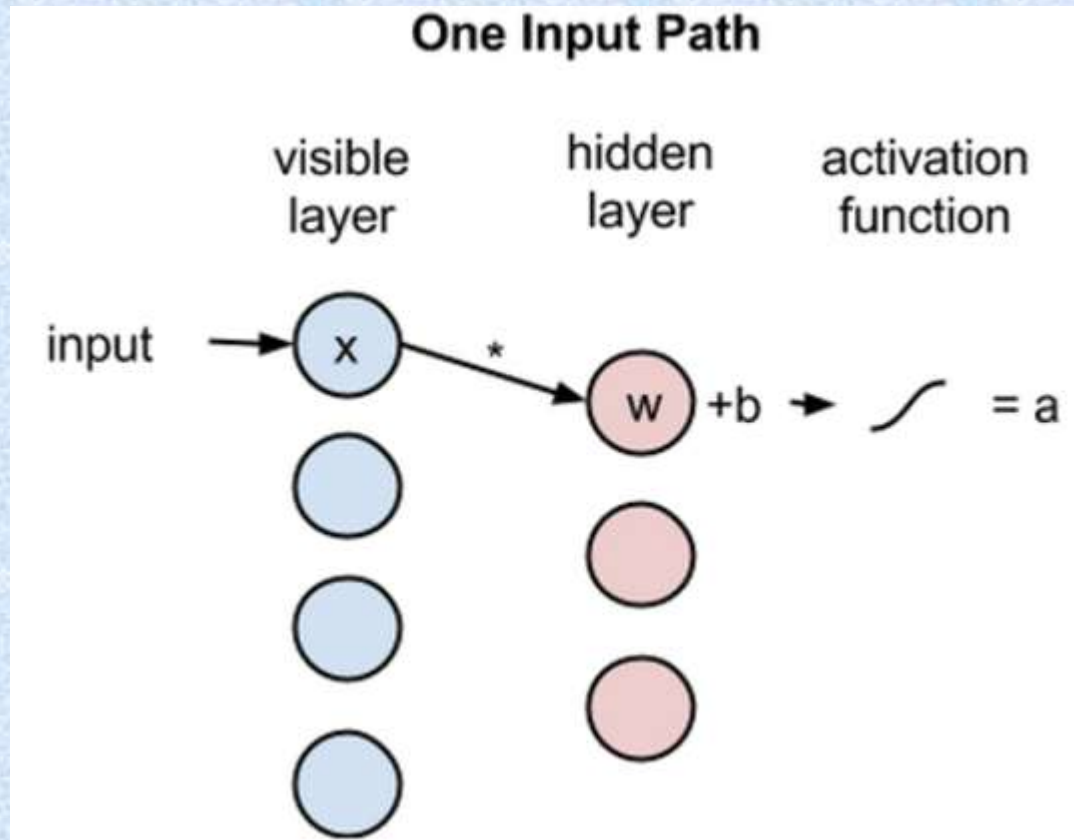
Comment fonctionnent les RBM?

Étape 3: Détection des fonctionnalités dans la couche cachée

- Suivons maintenant la valeur d'un seul pixel (**exemple de l'étape 2**), x , à travers le réseau à deux couches.
- Au nœud 1 de la couche cachée, x est multiplié par un *poids* et ajouté au *biais*.
- Le résultat de ces deux opérations est introduit dans une *fonction d'activation*, qui produit la sortie du nœud.

Comment fonctionnent les RBM?

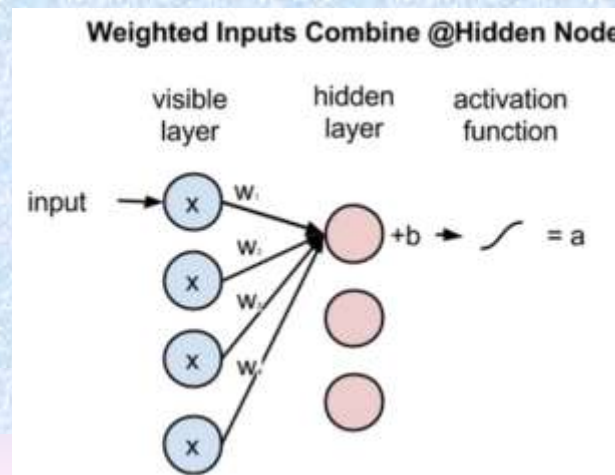
Étape 3: Détection des fonctionnalités dans la couche cachée



Comment fonctionnent les RBM?

Étape 3:

- Plusieurs entrées se combinent sur un nœud caché.
- Chaque x est multiplié par un poids distinct.
- Les produits sont additionnés, ajoutés à un biais.
- A nouveau le résultat est transmis par une fonction d'activation pour produire **la sortie du nœud**.



Comment fonctionnent les RBM?

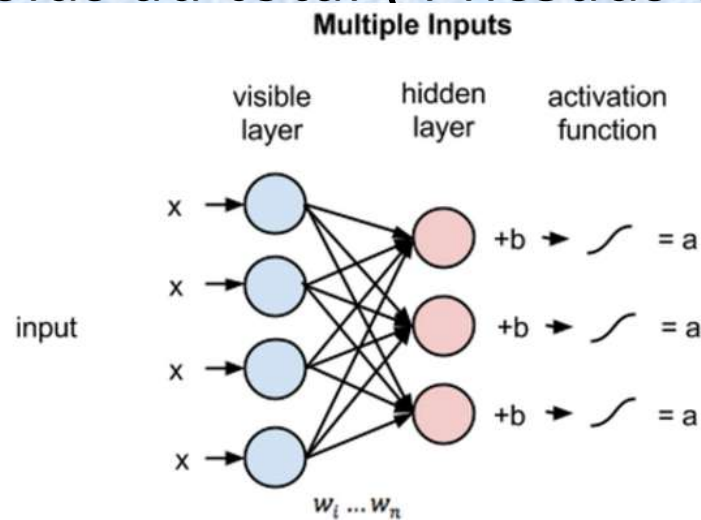
Étape 3:

- Étant donné que les entrées de tous les nœuds visibles sont transmises à tous les nœuds cachés, un **RBM** peut être défini comme un ***graphe bipartite symétrique***.
- *Symétrique* signifie que chaque nœud visible est connecté à chaque nœud caché. *Bipartite* signifie qu'il a deux parties, ou couches.

Comment fonctionnent les RBM?

Étape 3:

- À chaque nœud caché.
- Chaque entrée x est multipliée par son poids respectif w .
- C'est-à-dire qu'une seule entrée x aurait trois poids ici, ce qui fait 12 poids au total (4 nœuds d'entrée x 3 nœuds cachés).

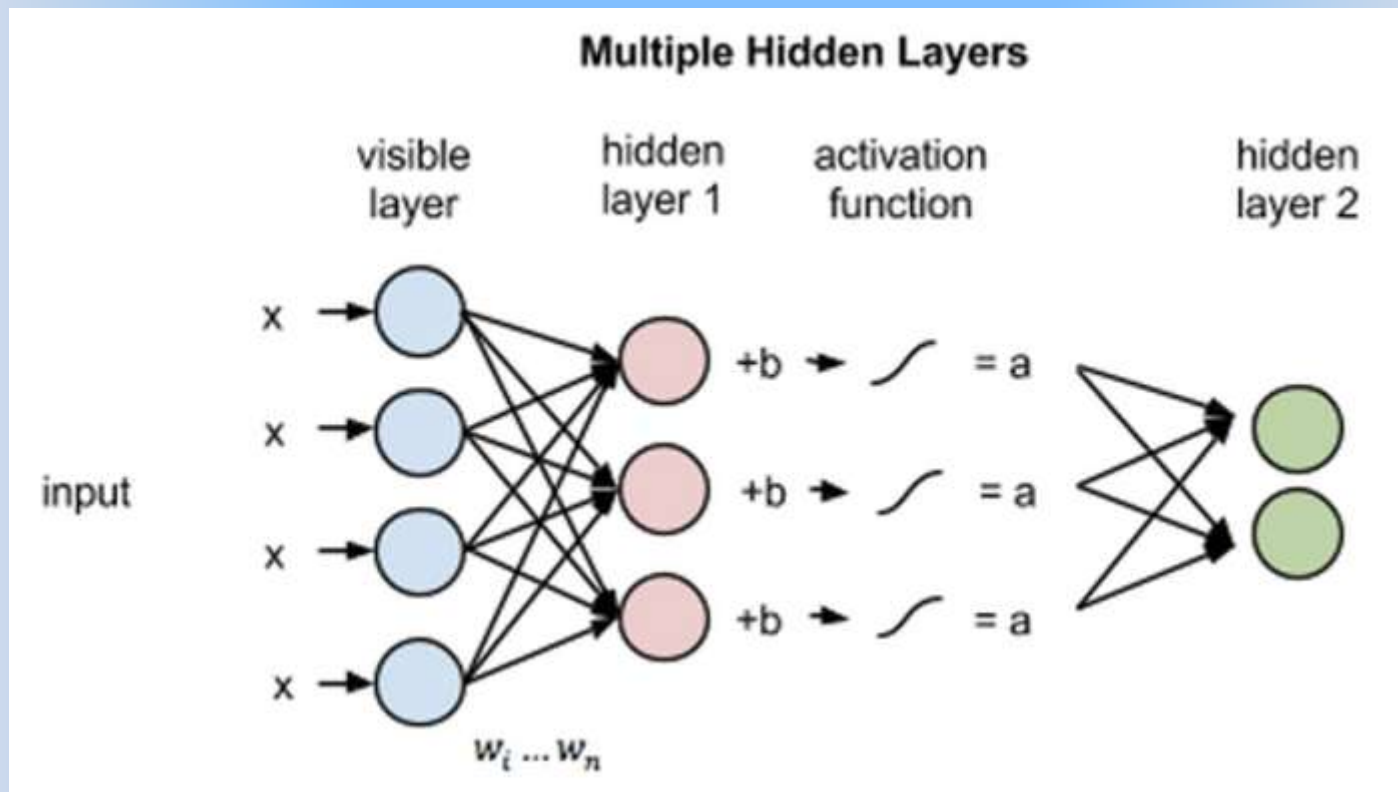


Comment fonctionnent les RBM?

Étape 3:

- Si ces deux couches faisaient partie d'un réseau neuronal plus profond.
- Les sorties de la couche cachée n° 1 seraient transmises en tant qu'entrées à la couche cachée n° 2.
- Et de là à travers autant de couches cachées que vous le souhaitez jusqu'à ce qu'elles atteignent une couche de classification finale.

Comment fonctionnent les RBM?



Comment fonctionnent les RBM?

Étape 4: Apprentissage de la distribution des données

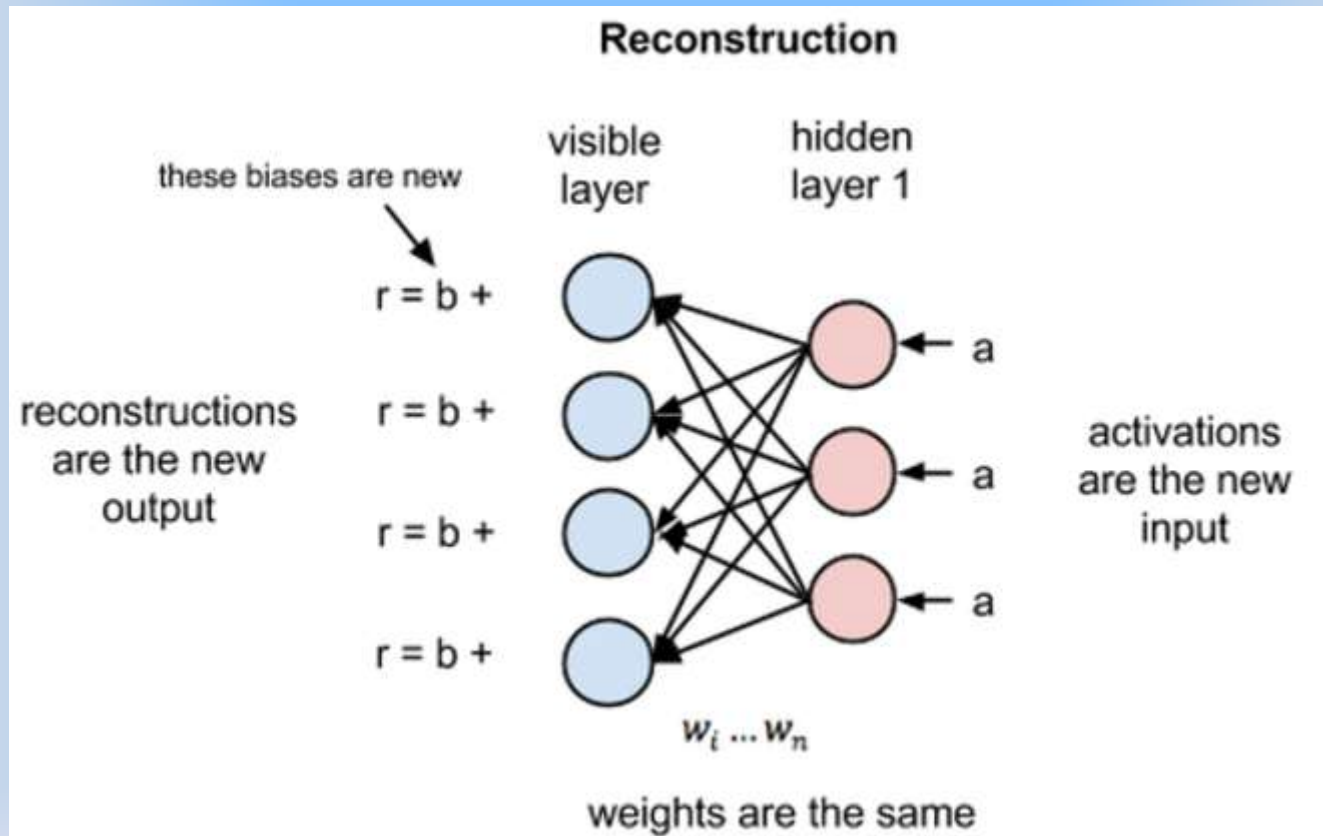
- Pendant l'entraînement, la RBM ajuste les poids des connexions entre les couches visibles et cachées.
- Ce processus aide le modèle à apprendre la distribution de probabilité des données d'entrée.

Comment fonctionnent les RBM?

Étape 5 : Reconstruction des données d'entrée

- Dans cette phase de reconstruction, les activations de la couche cachée n° 1 deviennent l'entrée d'un passage à rebours.
- Ils sont multipliés par les mêmes poids, un par arête inter-nodale, tout comme x a été ajusté au poids lors de la passe avant.
- La somme de ces produits est ajoutée à un biais de couche visible à chaque nœud visible.
- Le résultat de ces opérations est une reconstruction ; c'est-à-dire une approximation de l'entrée d'origine. 82

Comment fonctionnent les RBM?



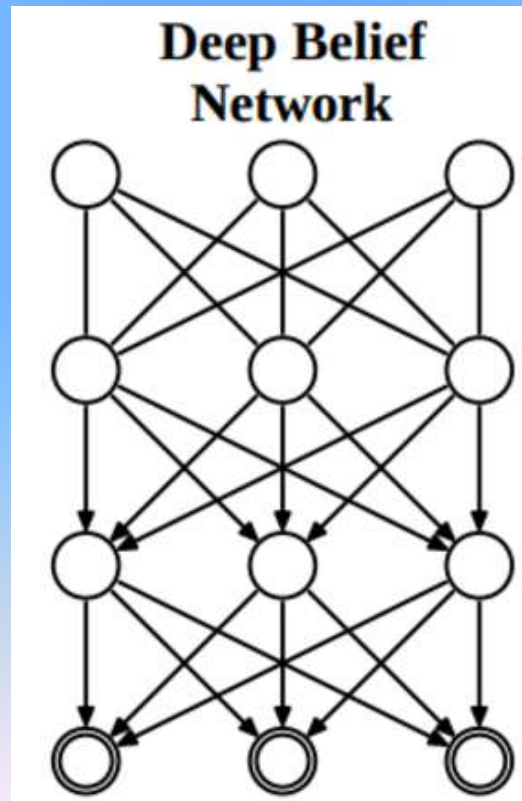
Comment fonctionnent les RBM?

- Étant donné que les poids du RBM sont initialisés de manière aléatoire, la différence entre les reconstructions et l'entrée d'origine est souvent importante.
- Vous pouvez considérer l'erreur de reconstruction comme la différence entre les valeurs de et les valeurs d'entrée, et cette erreur est ensuite rétro-propagée par rapport aux poids du RBM, encore et encore, dans un processus d'apprentissage itératif jusqu'à ce qu'un minimum d'erreur soit atteint.

Applications des RBM en IA

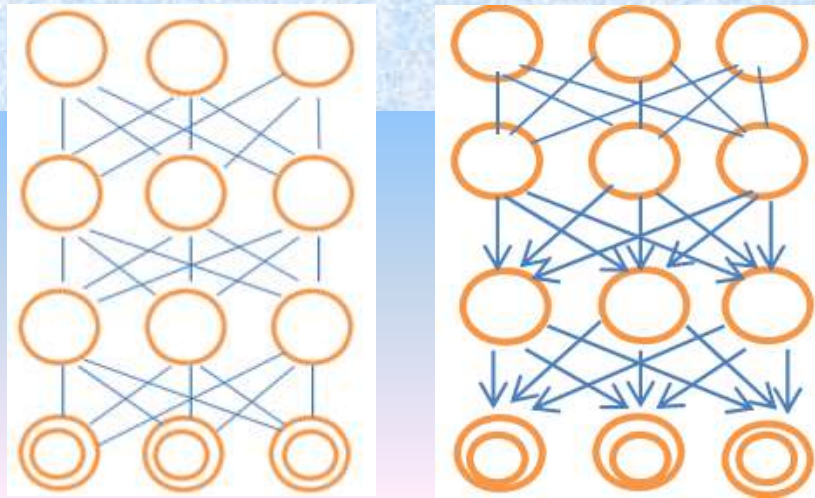
- Les RBMs sont largement utilisés pour la reconnaissance et la classification d'images, en capitalisant sur leur capacité à détecter des motifs complexes dans les données visuelles.
- Dans le traitement du langage naturel, Les RBMs aident à comprendre et à générer des modèles de langage, contribuant ainsi de manière significative à la traduction automatique, à la génération de texte et à l'analyse des sentiments.

Réseaux de croyances profondes (Deep Belief Network) (DBN)



Réseaux de croyances profondes: DBN

- Dans un **DBN**:
 - Plusieurs machines de Boltzmann restreintes sont empilées, de sorte que les sorties du premier RBM sont les entrées du RBM suivant.
- Les connexions à l'intérieur des couches individuelles ne sont pas dirigées, tandis que les connexions entre les couches sont dirigées. Cependant, il y a une exception ici. La connexion entre les deux couches supérieures n'est pas dirigée.



Réseaux de croyances profondes: DBN

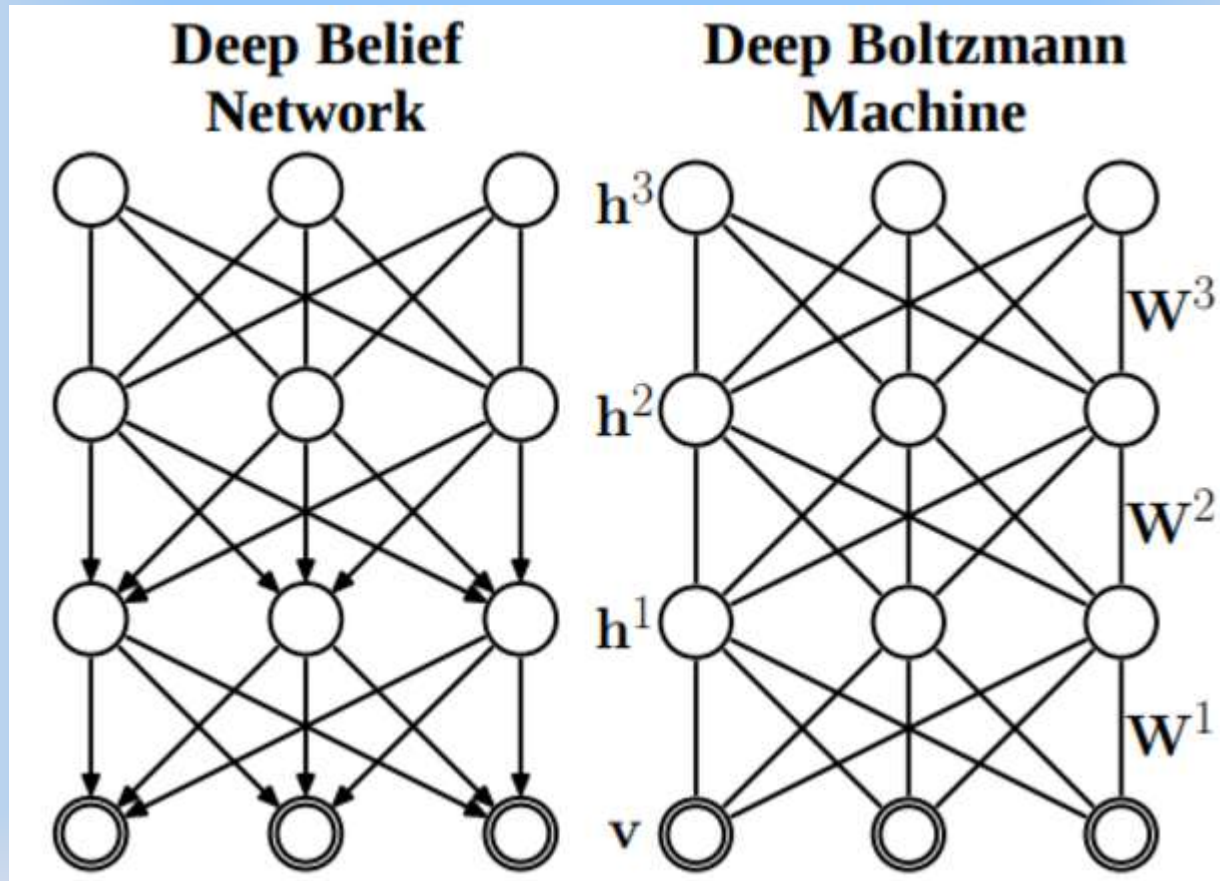
- Un **DBN** est un modèle génératif multicouche.
- Il est entraîné à extraire les caractéristiques essentielles des données d'entrée en maximisant la probabilité de ses données d'entraînement.
- Les DBN utilisent des (RBM) pour entraîner efficacement chaque couche d'un réseau profond.
- Les DBN ont été démontrés avec succès dans diverses applications, telles que la reconnaissance de chiffres manuscrits et la modélisation du mouvement humain.

Réseaux de croyances profondes (Deep Boltzman Machine) (DBM)

Machines Boltzmann profondes (DBM)

- Les **DBM** sont très similaires aux réseaux de croyances profondes (DBN).
- Les connexions à l'intérieur des couches, ainsi que les connexions entre les couches, sont toutes non dirigées.

Machines Boltzmann profondes (DBM)



Merci pour votre attention