

# RNNs, LSTMs, GRUs

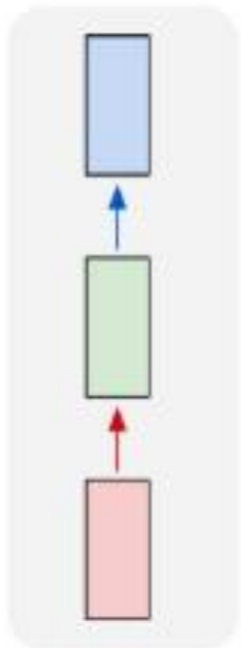
April 2023

# Séquences

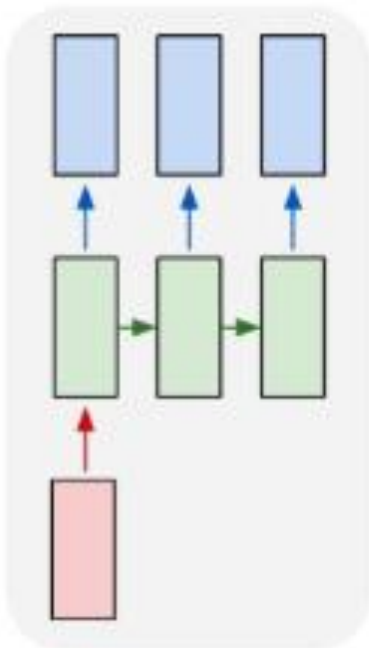
# Séquences

- Dans l'ensemble, les types de séquences qui existent dépendent de l'application ou du domaine spécifique. Les séquences peuvent se produire dans de nombreux formats différents et peuvent être analysées ou modélisées à l'aide de diverses techniques. Cependant, toutes les séquences partagent la caractéristique commune de contenir des données ordonnées qui se produisent au fil du temps.
- **Séries chronologiques** : Les séries chronologiques sont des séquences de points de données recueillies à intervalles réguliers au fil du temps. Les exemples incluent les prix des actions, les données météorologiques et les lectures de capteurs.
- **Langage naturel** : Les séquences en langage naturel sont des séquences de mots ou de caractères qui forment des phrases ou des documents. Les exemples incluent la parole, le texte et les publications sur les réseaux sociaux.
- **Musique** : Les séquences musicales sont des séquences de notes de musique ou d'événements qui forment des mélodies, des harmonies et des rythmes. Les exemples incluent les fichiers MIDI et les enregistrements audio.
- **Données génomiques** : Les séquences génomiques sont des séquences de nucléotides d'ADN ou d'ARN qui codent pour l'information génétique. Les exemples incluent les séquences d'ADN, les séquences d'ARN et les séquences de protéines.
- **Image et vidéo** : Les séquences d'images et de vidéos sont des séquences d'images ou de pixels qui forment un contenu visuel. Les exemples incluent des vidéos, des films et des images capturés par des caméras ou des capteurs.
- **Séquences d'événements** : Les séquences d'événements sont des séquences d'événements discrets qui se produisent au fil du temps. Les exemples incluent les interactions des utilisateurs avec un site Web ou une application et les données de journal des systèmes informatiques.
- **Parole**: Les séquences de parole sont des séquences de phonèmes ou de sons de la parole qui forment des mots et des phrases parlés. Les exemples incluent les enregistrements audio de la parole humaine et les modèles de synthèse vocale.
- **Mouvement**: Les séquences de mouvement sont des séquences de mouvements et de poses qui décrivent le mouvement d'objets ou de personnes. Les exemples incluent les données de capture de mouvement et les enregistrements vidéo des mouvements humains.

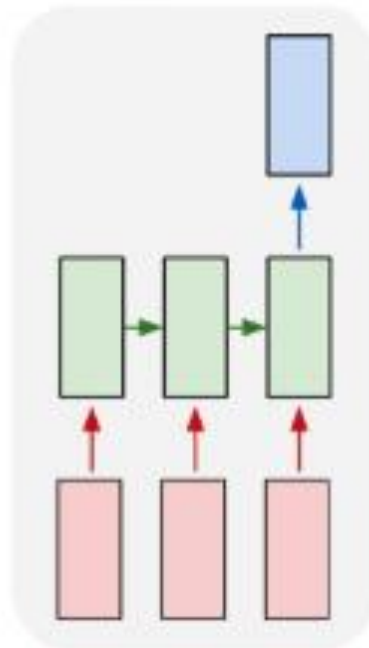
1-one to one



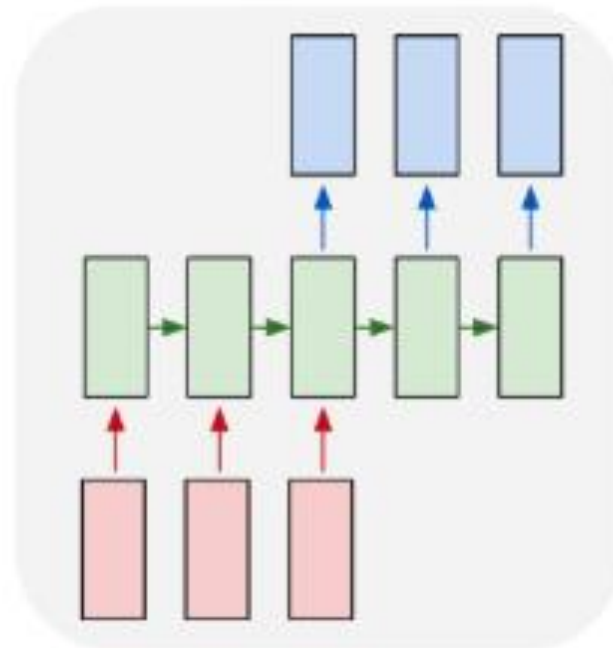
2-one to many



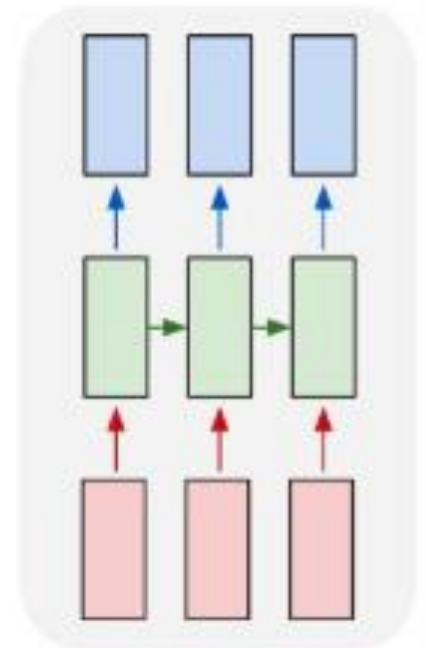
3-many to one



4-many to many

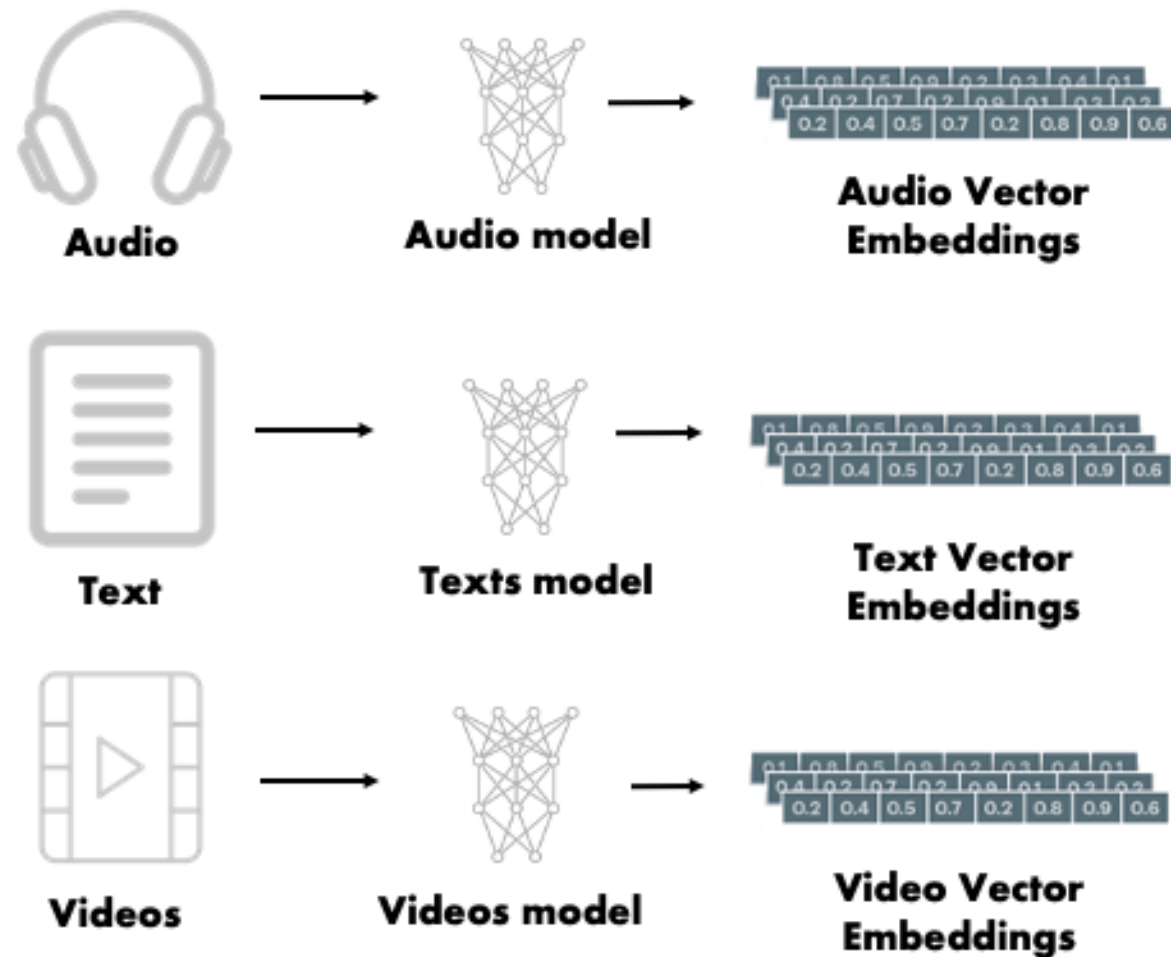


5-many to many



# Comment le contenu devient NN Input

# Comment le contenu devient NN Input



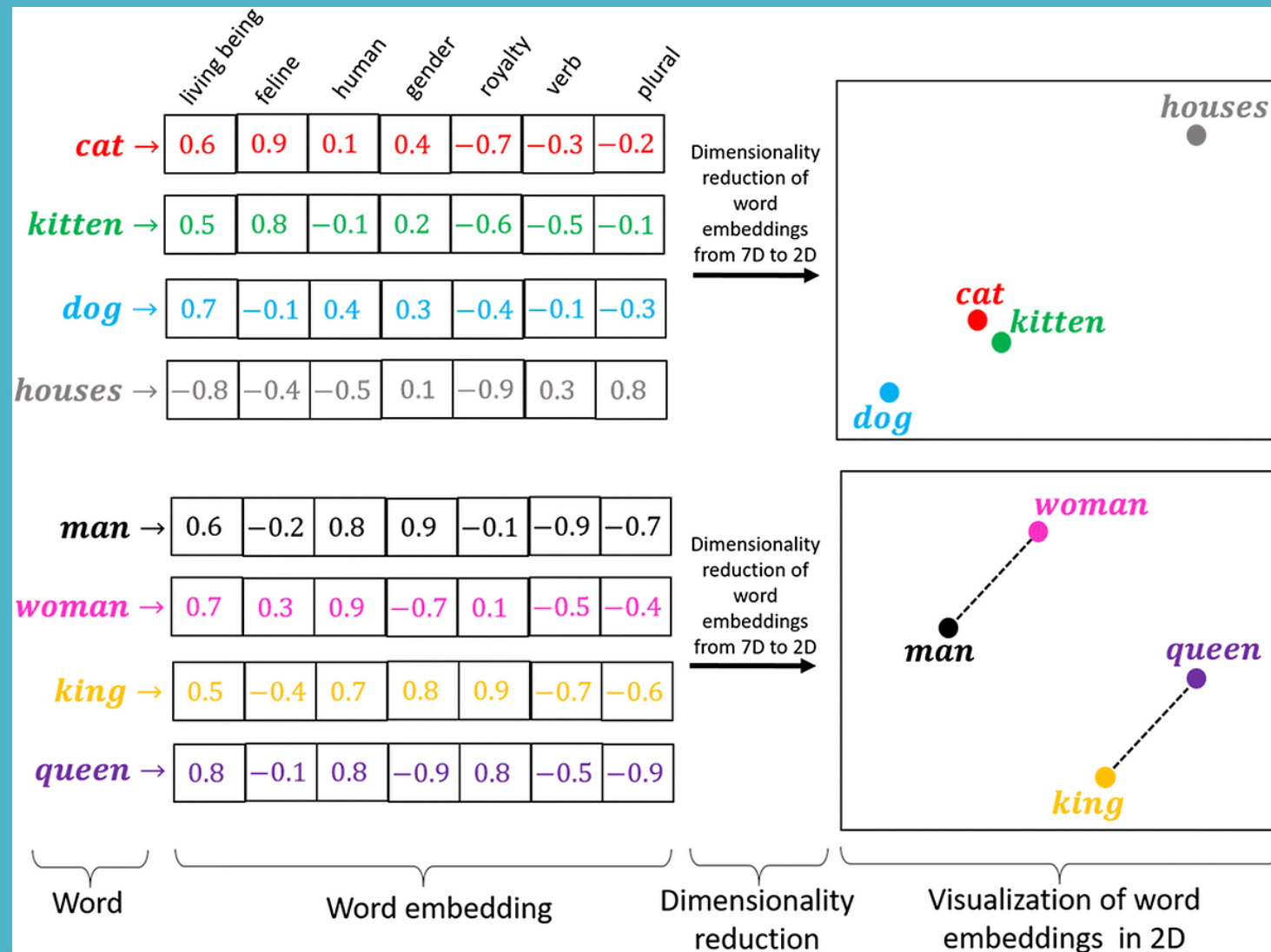
# Encodage One-Hot

- L'encodage à chaud unique est une méthode de représentation de données catégorielles sous forme de vecteur binaire.
- Chaque catégorie d'entrée se voit attribuer un vecteur binaire unique, avec un 1 dans la position correspondant à la catégorie et des 0 dans toutes les autres positions.
- Par exemple, dans un modèle de langage, chaque mot du vocabulaire peut être représenté par un vecteur codé à chaud.
- L'encodage à chaud unique est simple et efficace, mais peut être de grande dimension et clairsemé, ce qui peut être coûteux en calcul.

|     | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z |
|-----|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| L → | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| U → | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| K → | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| A → | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| S → | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

# Word embeddings

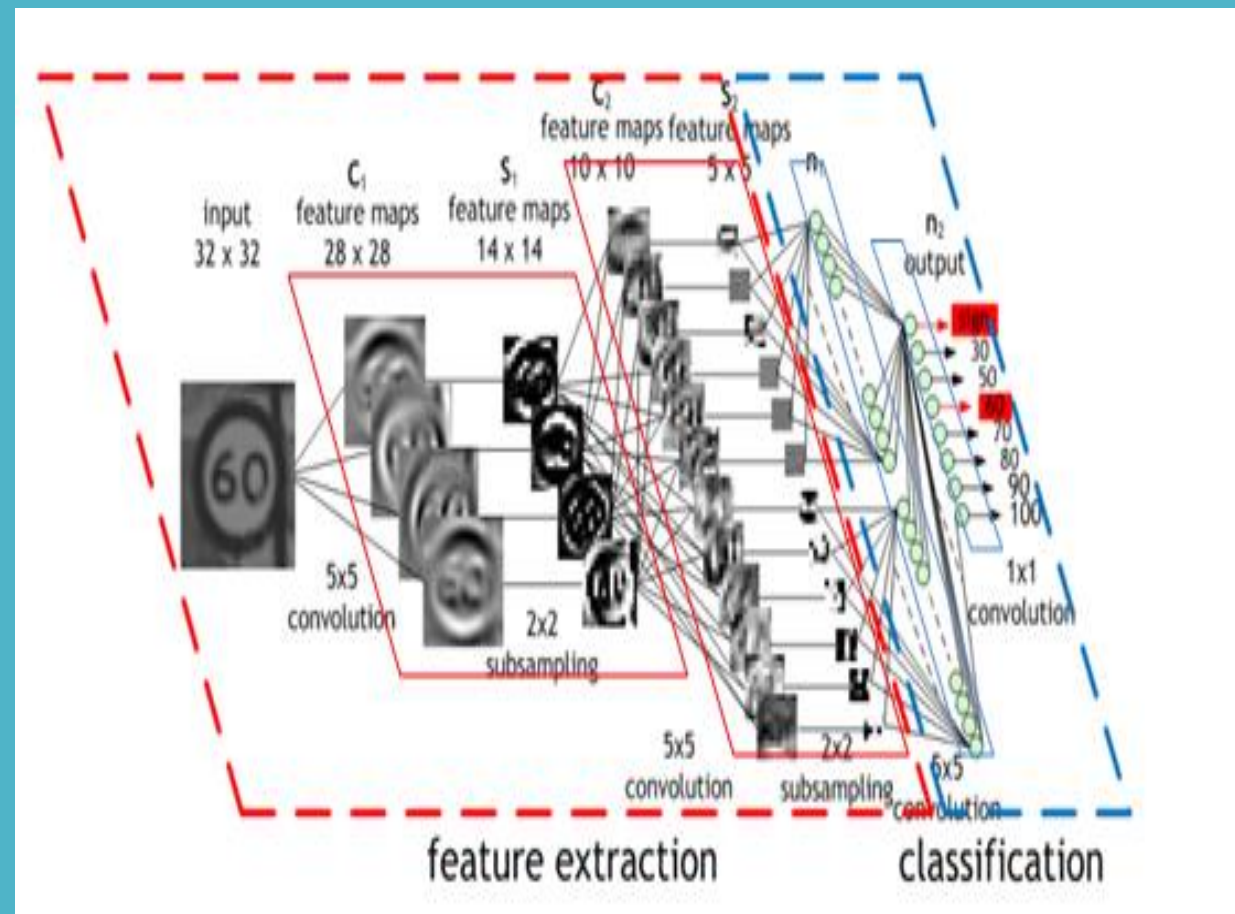
- Les intégrations de mots sont des représentations vectorielles denses de mots qui capturent les relations sémantiques et syntaxiques entre eux.
- Les intégrations de mots sont généralement apprises à partir d'un grand corpus de texte à l'aide de techniques d'apprentissage non supervisées telles que Word2Vec ou GloVe.
- Dans un modèle de langage, les mots peuvent être représentés par leurs incorporations de mots correspondantes.
- Les intégrations de mots sont de dimensions inférieures et plus denses que les vecteurs codés à une seule pression, ce qui les rend plus efficaces sur le plan informatique.





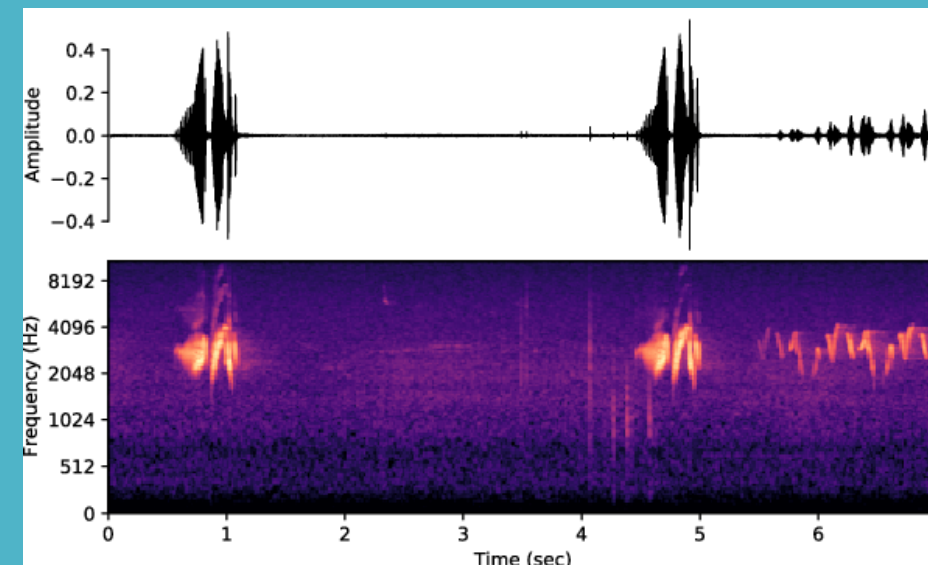
# Image embeddings

- Les intégrations d'images sont des représentations vectorielles denses d'images qui capturent des caractéristiques visuelles et des relations entre elles.
- Les intégrations d'images peuvent être générées à l'aide de réseaux neuronaux convolutifs (CNN) ou d'autres techniques de traitement d'image.
- Dans un modèle de sous-titrage vidéo, les images ou les régions d'intérêt de la vidéo peuvent être représentées par leurs intégrations d'images correspondantes.



# Audio embeddings

- Les intégrations audio sont des représentations vectorielles denses de signaux audio qui capturent les caractéristiques acoustiques et les relations entre eux.
- Les intégrations audio peuvent être générées à l'aide de réseaux neuronaux convolutifs (CNN) ou d'autres techniques de traitement audio.
- Dans un modèle de reconnaissance vocale, les segments audio peuvent être représentés par leurs intégrations audio correspondantes.







# Recurrent Neural Networks (RNN)

# Recurrent Neural Networks (RNN)

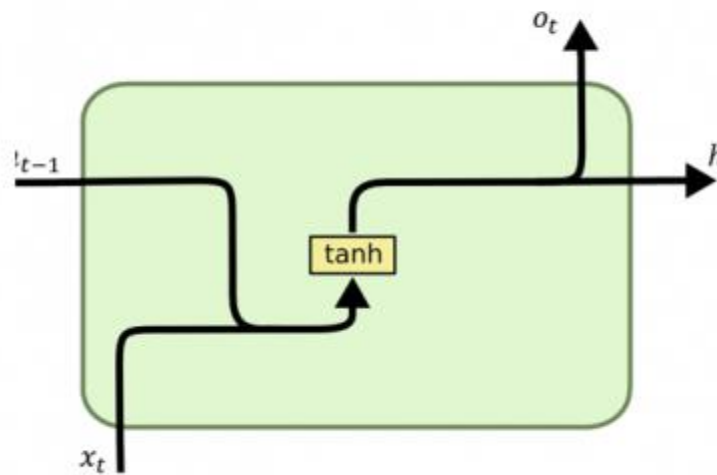
- Les réseaux neuronaux récurrents (RNN) sont un type d'architecture de réseau neuronal conçu pour traiter des données séquentielles, telles que des séries chronologiques ou le langage naturel.
- Contrairement aux réseaux neuronaux feedforward, qui reçoivent des entrées et produisent des sorties dans un flux unidirectionnel, les RNN ont des boucles qui permettent à l'information de persister et d'être traitée au fil du temps.
- Cette structure de boucle permet aux RNN de capturer les dépendances entre les éléments d'une séquence, ce qui les rend particulièrement utiles pour des tâches telles que la reconnaissance vocale, la prédiction de texte et l'analyse des sentiments.
- Cependant, l'une des limites des RNN est qu'ils peuvent avoir du mal à maintenir une mémoire à long terme et peuvent souffrir de gradients de disparition ou d'explosion lorsqu'ils sont entraînés sur des séquences plus longues.



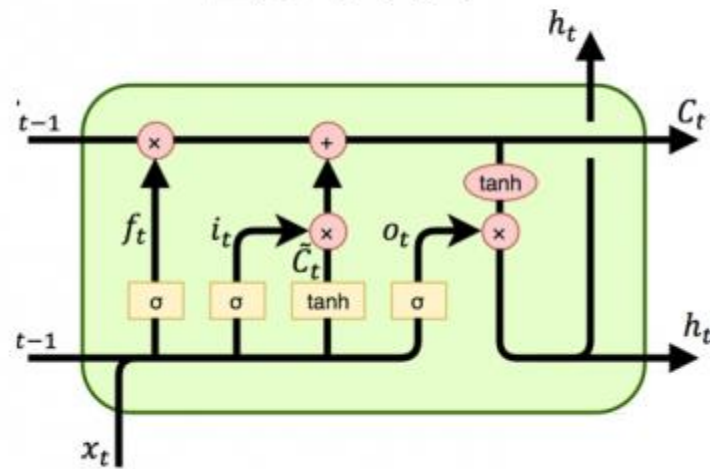
# Applications des RNN

- Traitement du langage naturel (NLP): Les RNN peuvent être utilisés pour des tâches telles que la modélisation du langage, la traduction automatique et la classification de texte.
- Reconnaissance vocale: Les RNN peuvent être utilisés pour des tâches telles que la reconnaissance vocale et l'identification des locuteurs.
- Analyse de séries chronologiques : Les RNN peuvent être utilisés pour des tâches telles que la prévision des cours des actions, les prévisions météorologiques et la prévision de la demande d'énergie.
- Analyse d'images et de vidéos: Les RNN peuvent être utilisés pour des tâches telles que la reconnaissance d'actions, le sous-titrage vidéo et le sous-titrage d'images.
- Génération de musique: Les RNN peuvent être utilisés pour des tâches telles que la génération et la composition de musique.

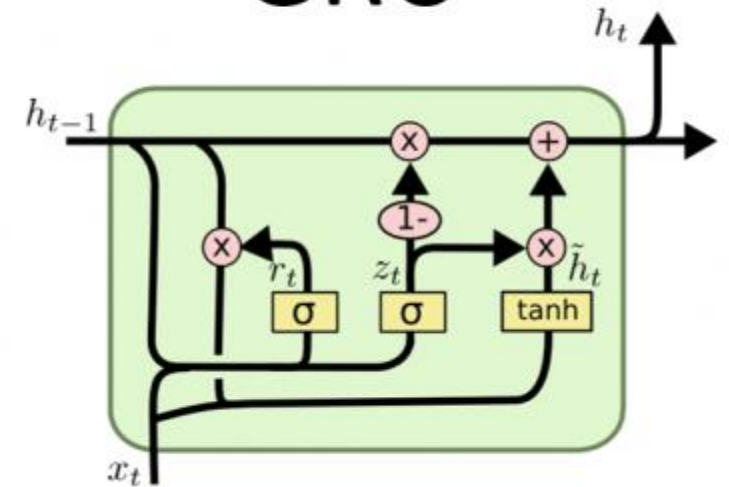
# RNN

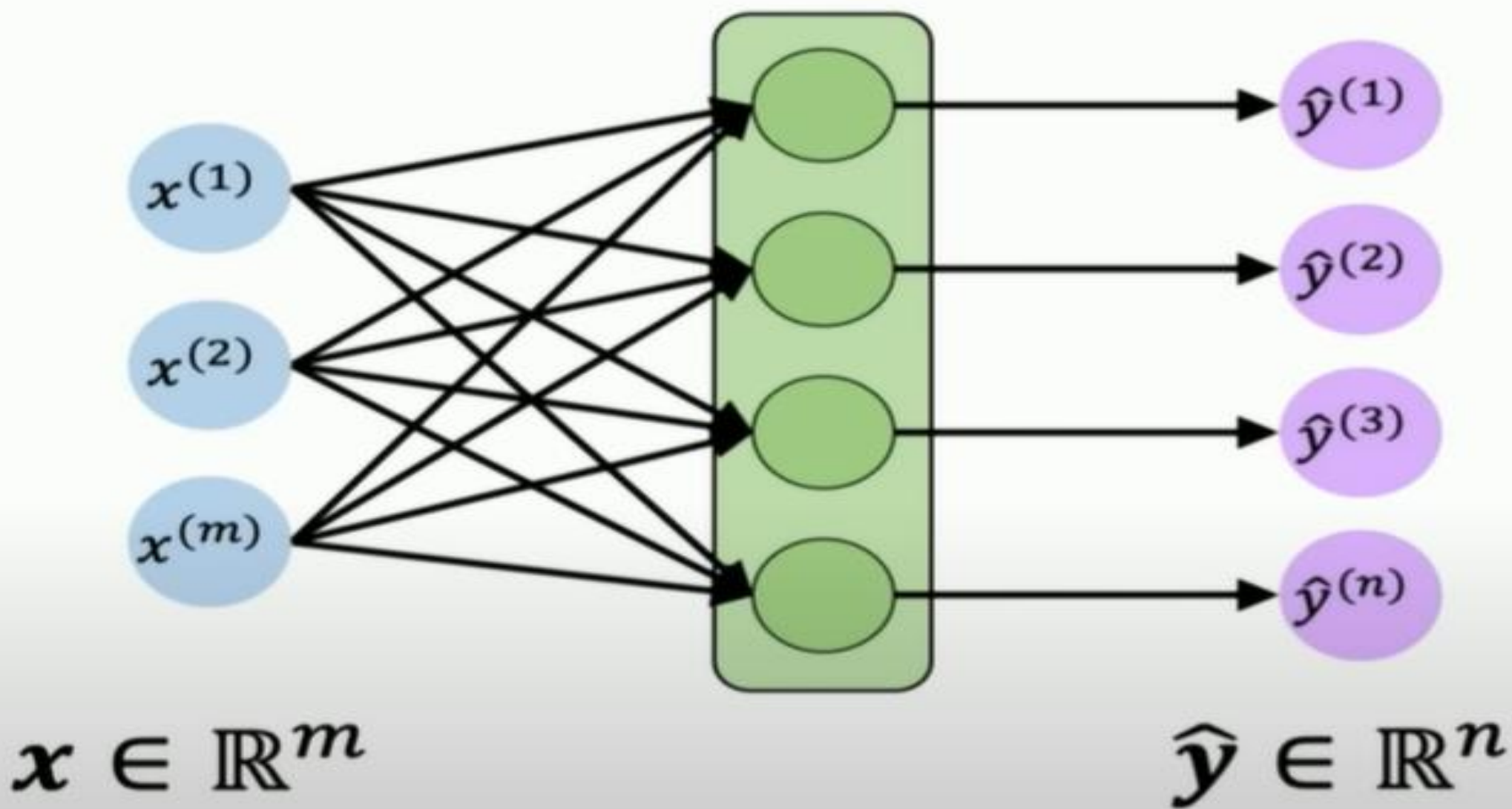


# LSTM

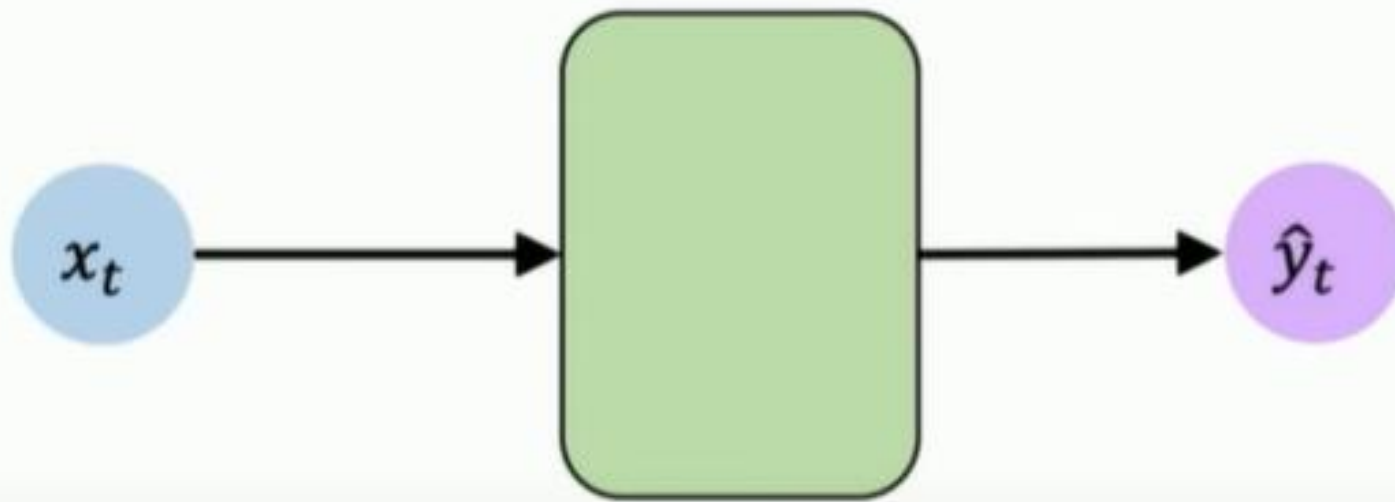


# GRU



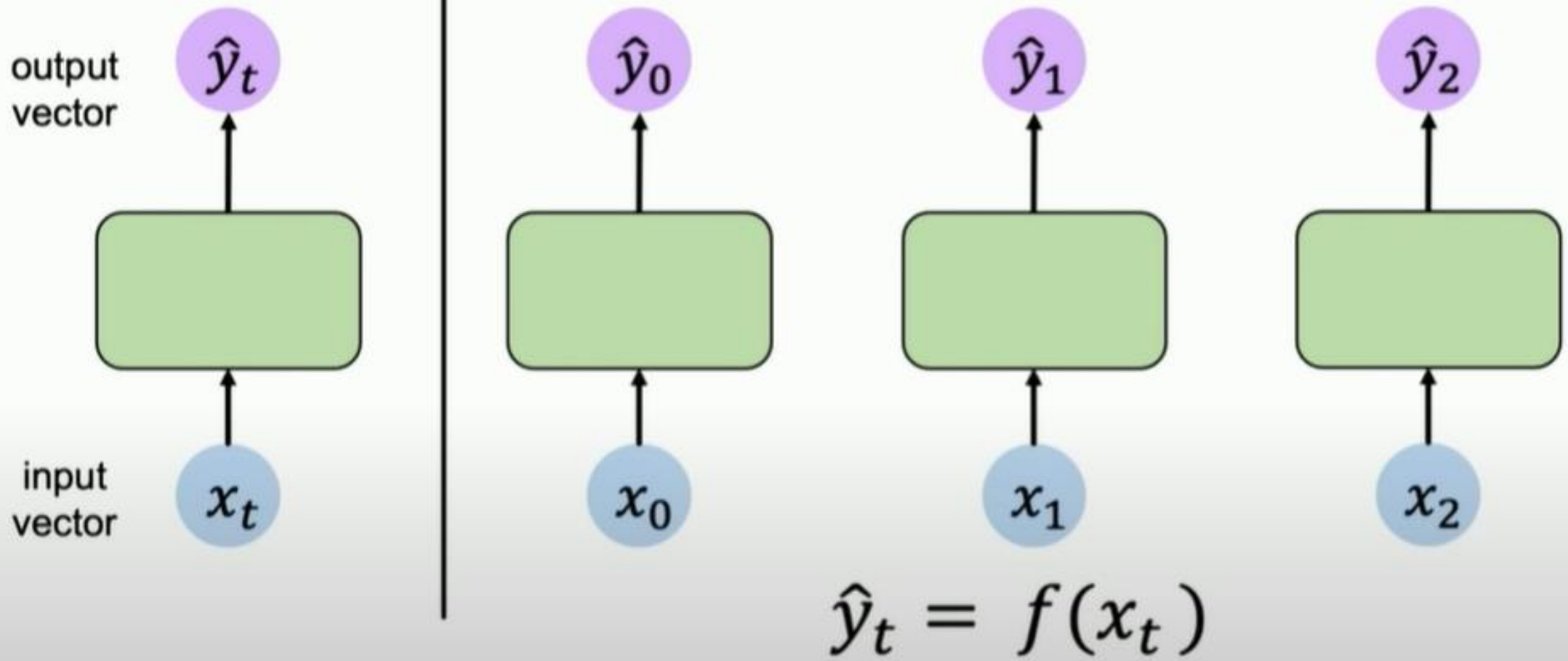


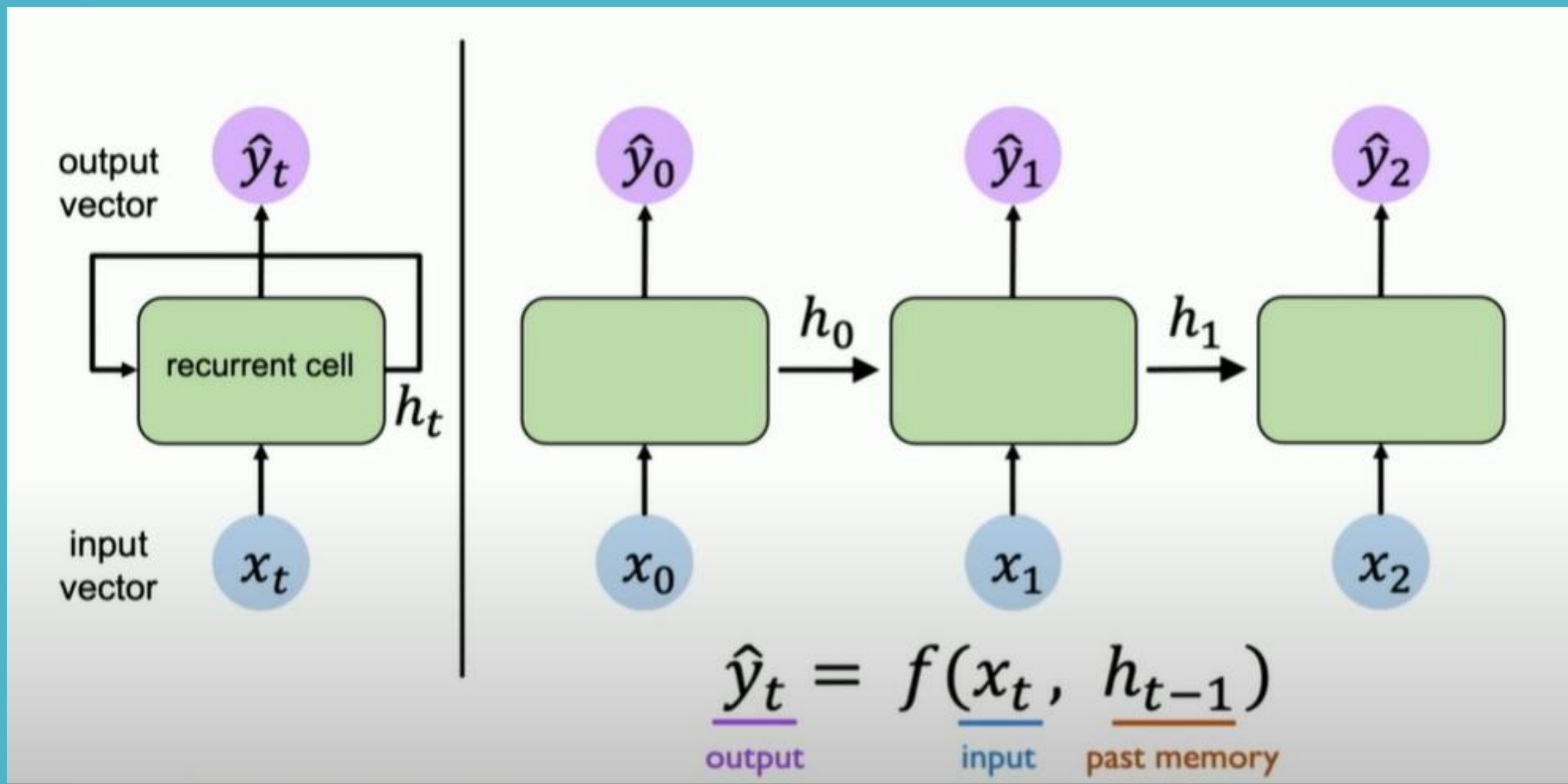


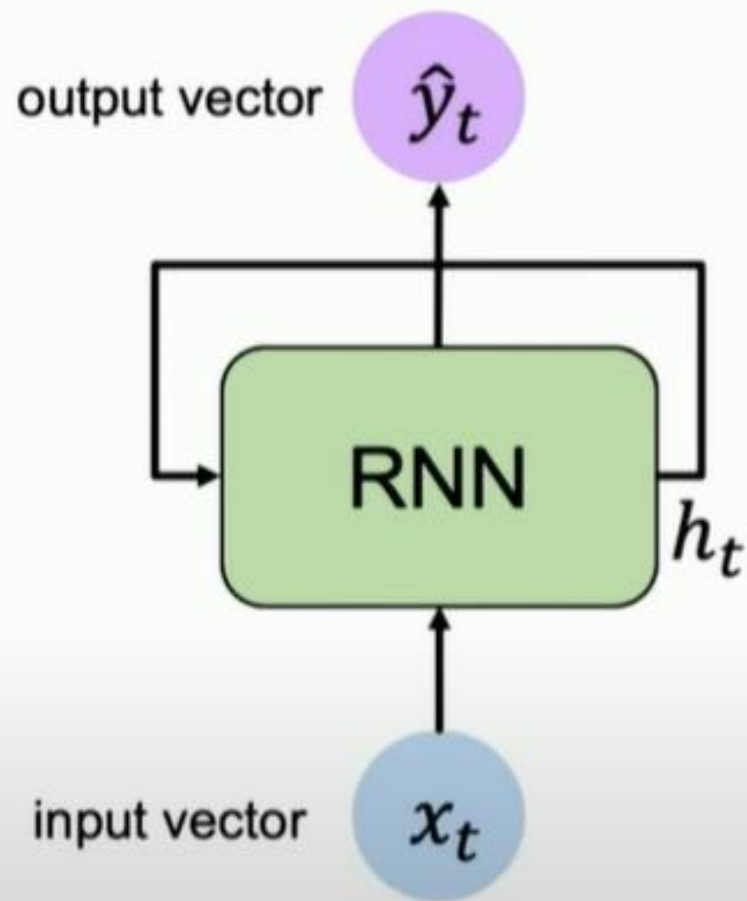


$$x_t \in \mathbb{R}^m$$

$$\hat{y}_t \in \mathbb{R}^n$$





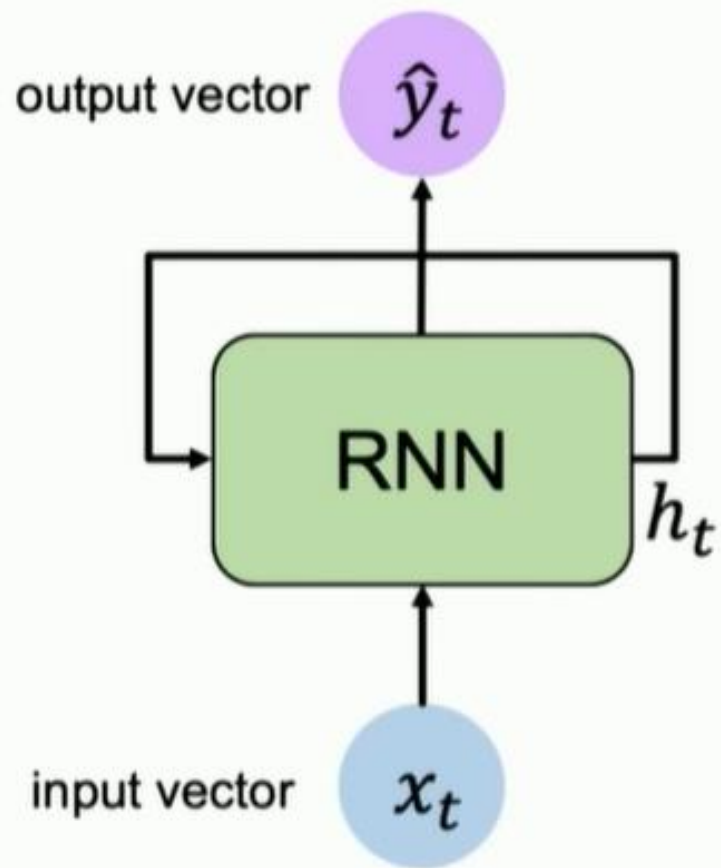


Apply a **recurrence relation** at every time step to process a sequence:

$$\boxed{h_t} = \boxed{f_W}(\boxed{x_t}, \boxed{h_{t-1}})$$

cell state      function with weights  $W$       input      old state

RNNs have a **state**,  $h_t$ , that is updated **at each time step** as a sequence is processed

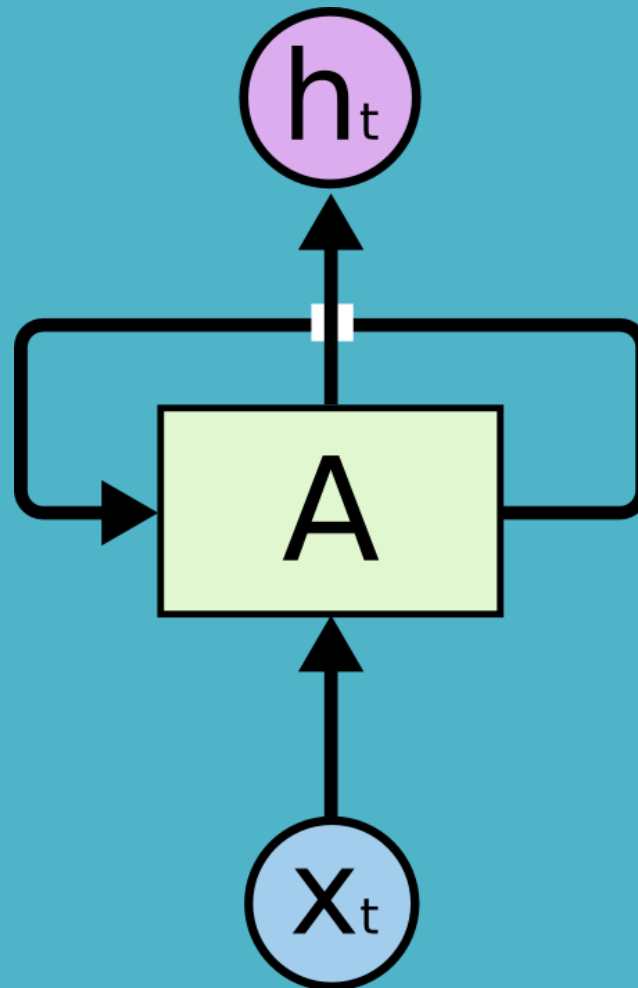


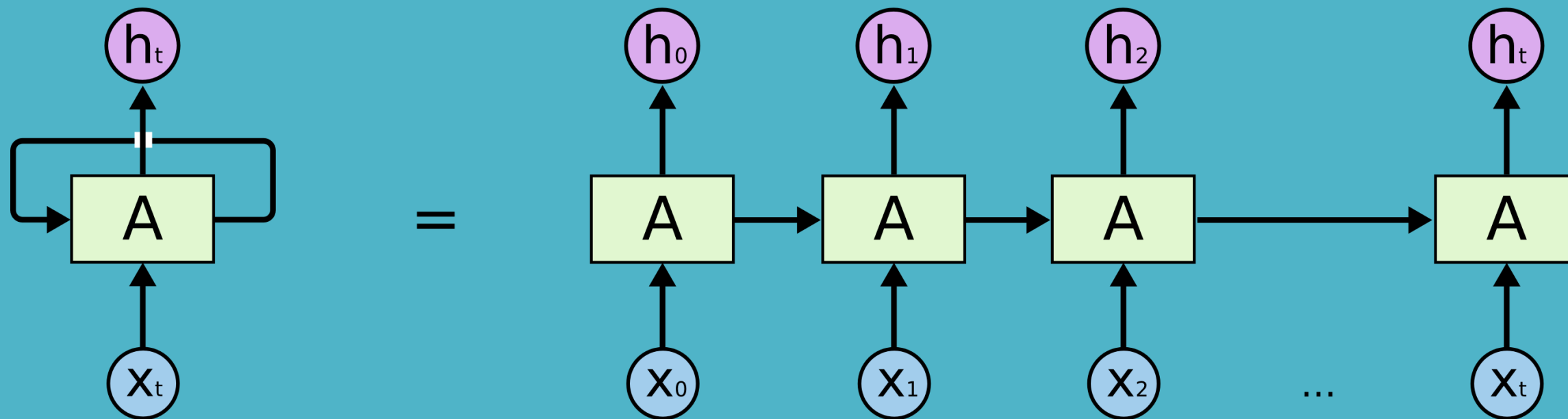
Update Hidden State

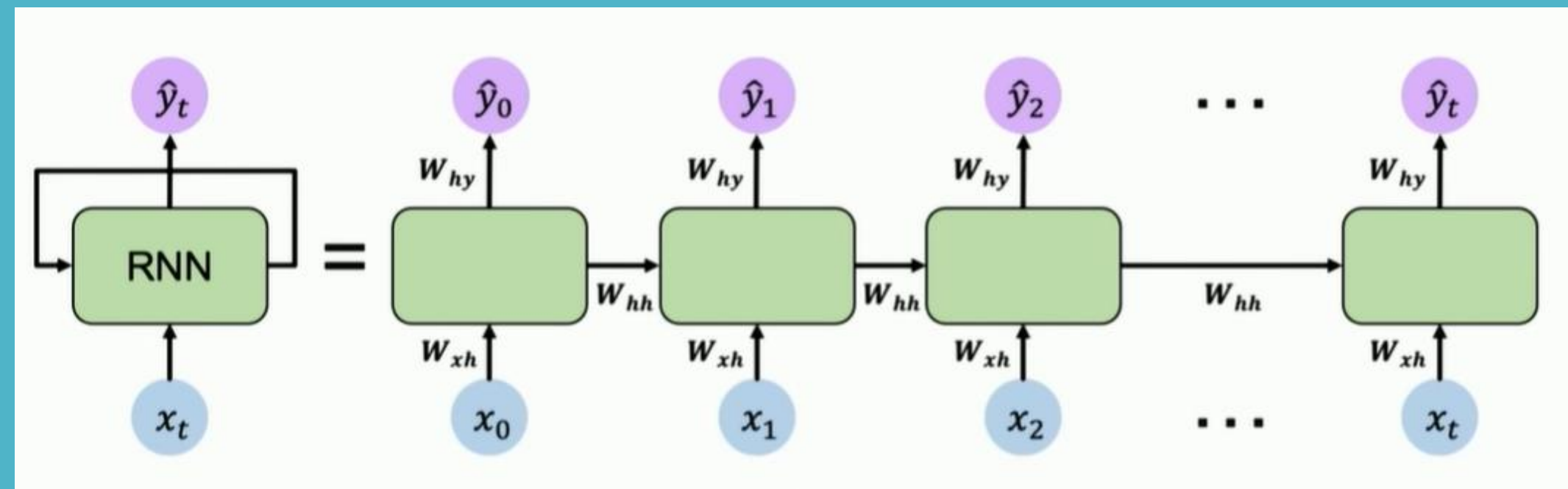
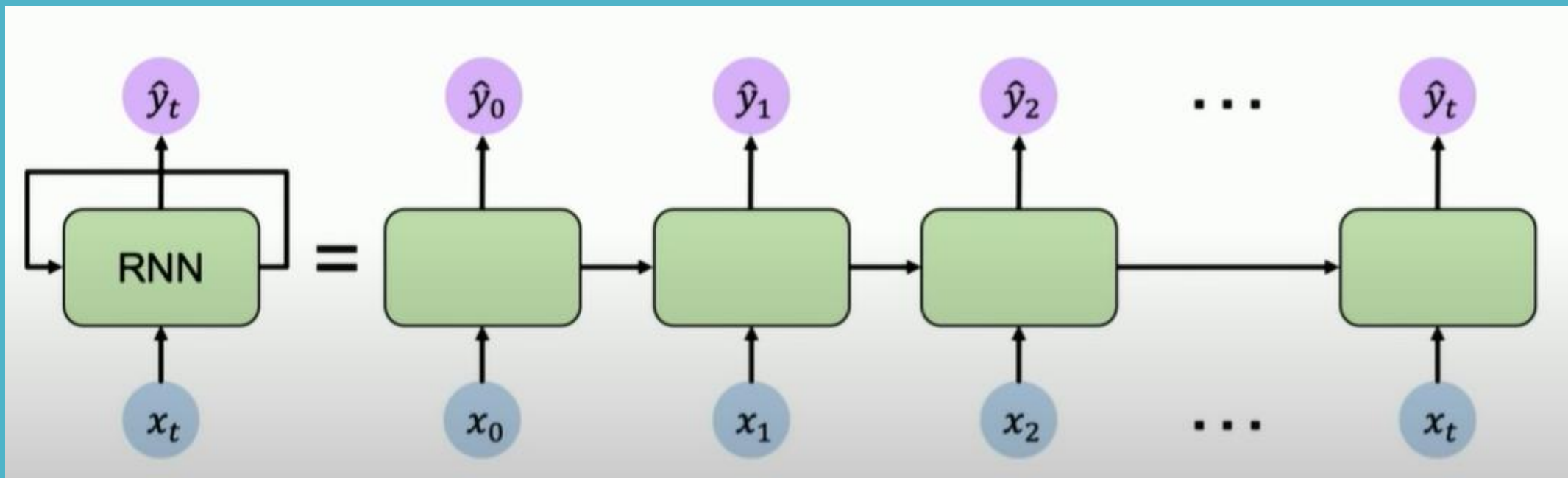
$$h_t = \tanh(W_{hh}^T h_{t-1} + W_{xh}^T x_t)$$

Input Vector

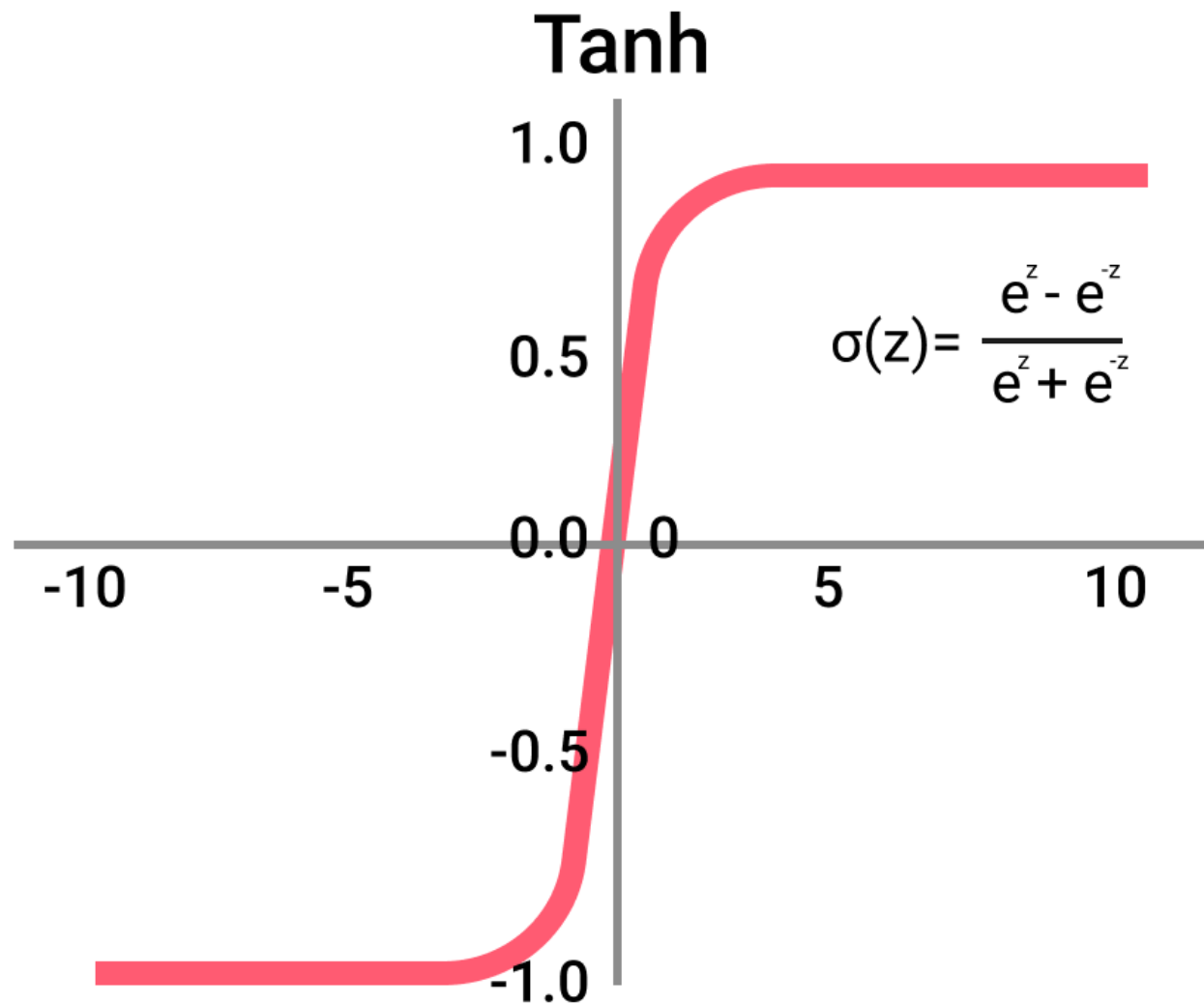
$x_t$



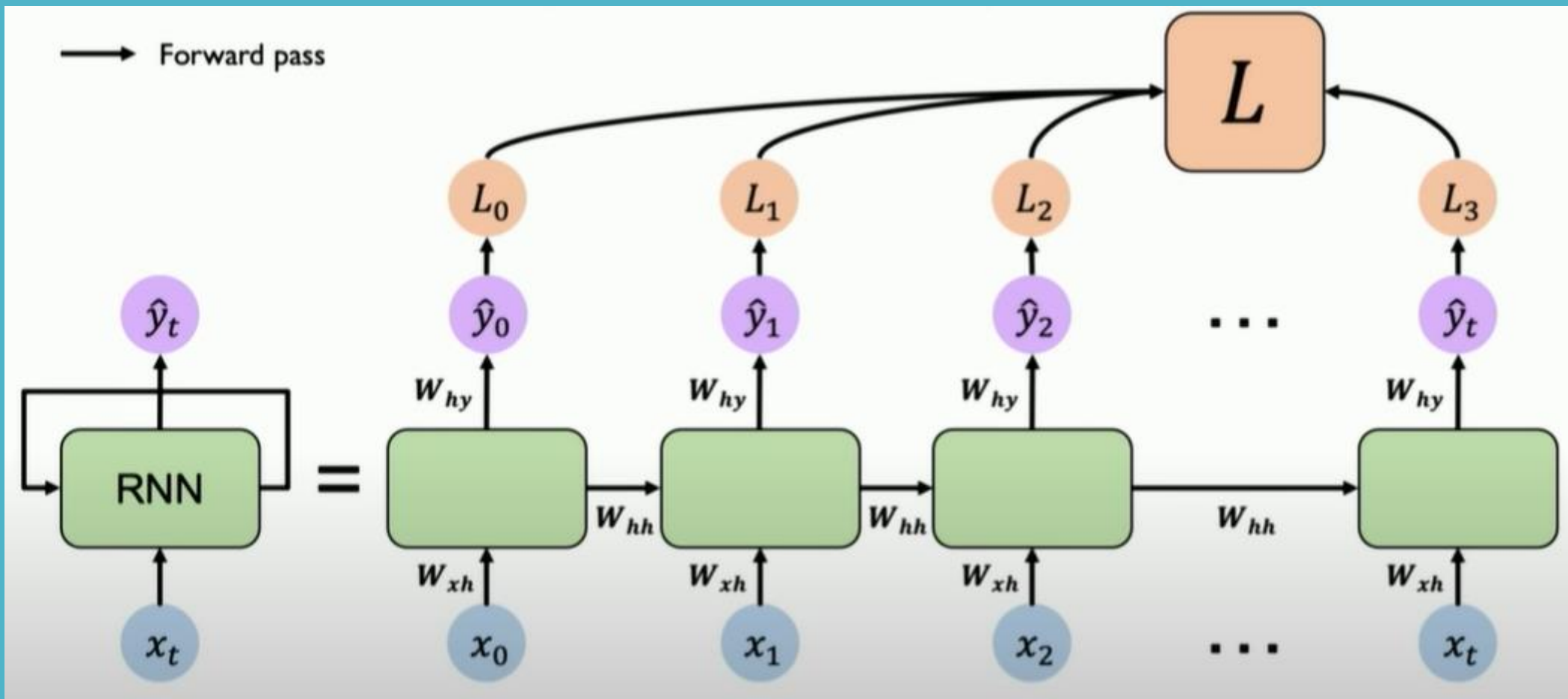








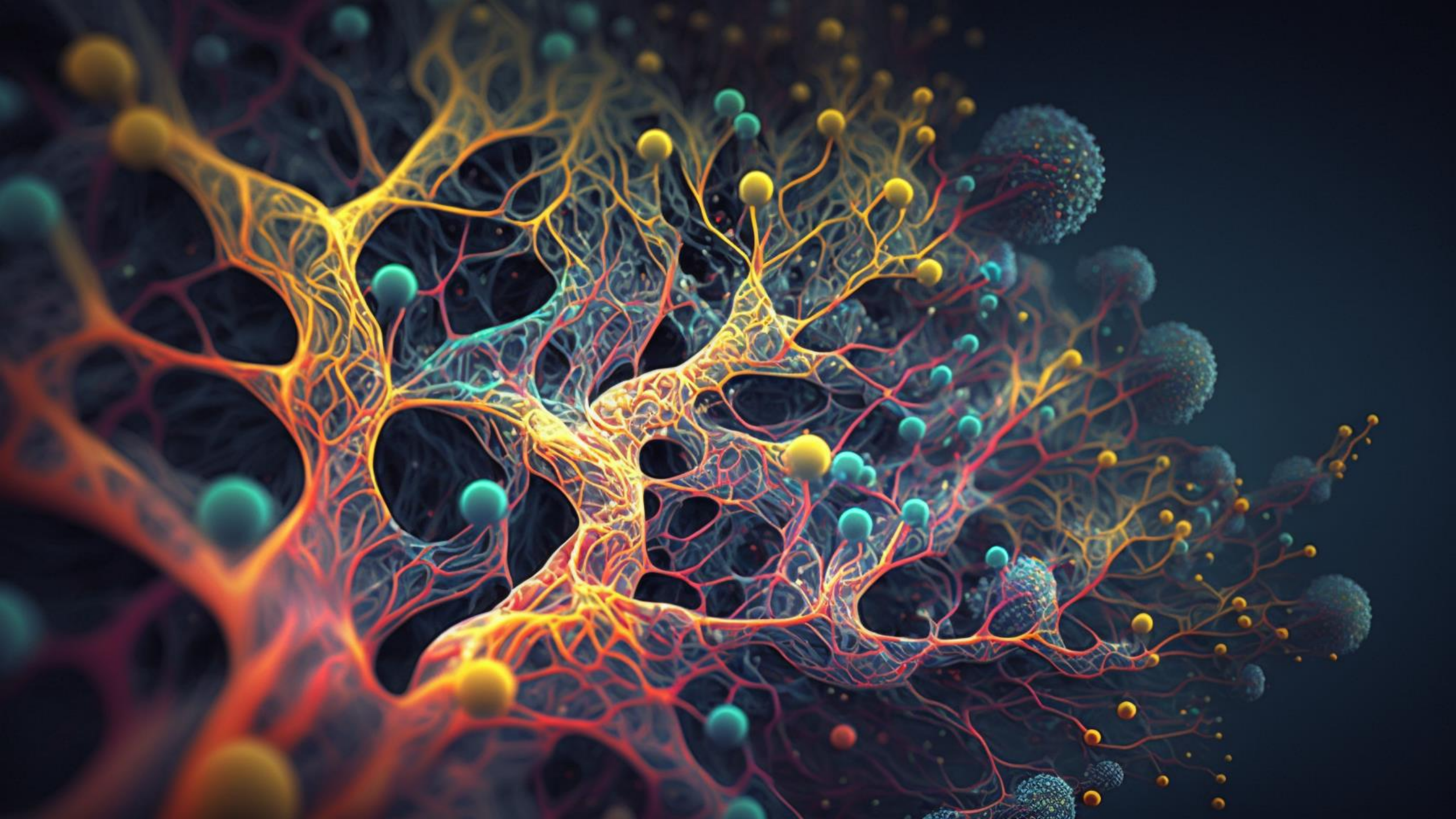
# Loss function calculation



# Vanishing Gradient Problem

- Le problème du gradient de fuite se produit dans les réseaux de neurones récurrents (RNN) lorsque les gradients propagés à rebours dans le temps deviennent très petits ou disparaissent complètement.
- Le problème se pose en raison de la façon dont les gradients sont calculés et propagés dans les RNN, ce qui peut faire disparaître le gradient à zéro lorsqu'il se propage à rebours dans le temps.
- Le problème du gradient de fuite est particulièrement prononcé dans les RNN profonds, où les gradients doivent passer par plusieurs couches de fonctions d'activation non linéaires.
- Les fonctions d'activation telles que la fonction sigmoïde peuvent exacerber le problème du gradient de fuite, car elles peuvent saturer lorsque l'entrée est grande ou petite.
- Le problème du gradient de disparition peut entraîner une mauvaise performance et une convergence lente pendant l'entraînement.
- Pour résoudre le problème, plusieurs techniques ont été proposées, telles que l'utilisation de fonctions d'activation telles que l'unité linéaire rectifiée (ReLU) qui ne sature pas, l'initialisation des poids à l'aide de méthodes telles que l'initialisation Glorot ou l'initialisation He, et l'utilisation d'architectures telles que les réseaux à mémoire courte durée (LSTM) ou les unités récurrentes fermées (GRU) spécialement conçues pour éviter le problème du gradient de fuite.





# Long Short-Term Memory (LSTM) networks

# Long Short-Term Memory (LSTM) networks

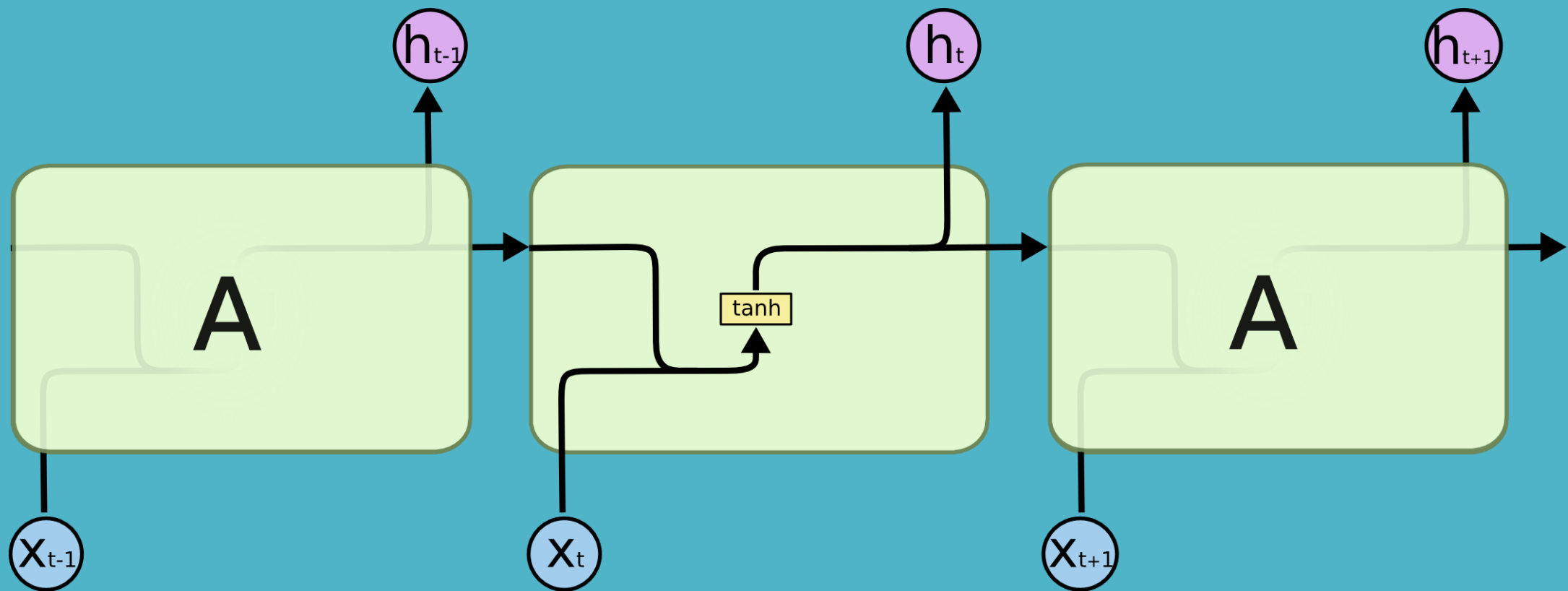
- Les réseaux LSTM ont été introduits pour la première fois dans un article de 1997 par Hochreiter et Schmidhuber comme un moyen de résoudre le problème des gradients de fuite dans les réseaux neuronaux récurrents.
- L'innovation clé des LSTM est l'utilisation de cellules de mémoire, qui permettent de stocker et de récupérer des informations sur de longues périodes.
- Chaque cellule de mémoire d'un LSTM est associée à une « porte d'oubli » et à une « porte d'entrée » qui contrôlent respectivement l'effacement ou la mise à jour du contenu de la cellule.
- La porte d'oubli est une fonction sigmoïde qui décide quelles informations conserver dans la cellule mémoire et lesquelles rejeter, en fonction de l'entrée actuelle et de l'état précédent.
- La porte d'entrée détermine la quantité de nouvelles informations à ajouter à la cellule mémoire, en fonction de l'entrée et de l'état précédent.
- Les LSTM ont également une « porte de sortie » qui contrôle la quantité d'informations de la cellule de mémoire transmise à l'étape suivante de la séquence.
- En oubliant ou en conservant sélectivement des informations au fil du temps, les LSTM peuvent éviter le problème des gradients de disparition et maintenir des dépendances à long terme dans les données séquentielles.
- Les LSTM ont été appliqués avec succès à un large éventail de tâches, notamment la reconnaissance vocale, la traduction automatique et l'analyse des sentiments.



# Applications des LSTM

- NLP : les LSTM peuvent être utilisés pour des tâches telles que l'analyse des sentiments, la génération de texte et la reconnaissance vocale.
- Reconnaissance vocale: Les LSTM peuvent être utilisés pour des tâches telles que la reconnaissance vocale et l'identification du locuteur.
- Analyse d'images et de vidéos: Les LSTM peuvent être utilisés pour des tâches telles que le sous-titrage vidéo et le sous-titrage d'images.
- Robotique: Les LSTM peuvent être utilisés pour des tâches telles que le contrôle et la manipulation de robots.
- Soins de santé: Les LSTM peuvent être utilisés pour des tâches telles que la prédiction des résultats pour les patients et la détection de maladies.

# RNN

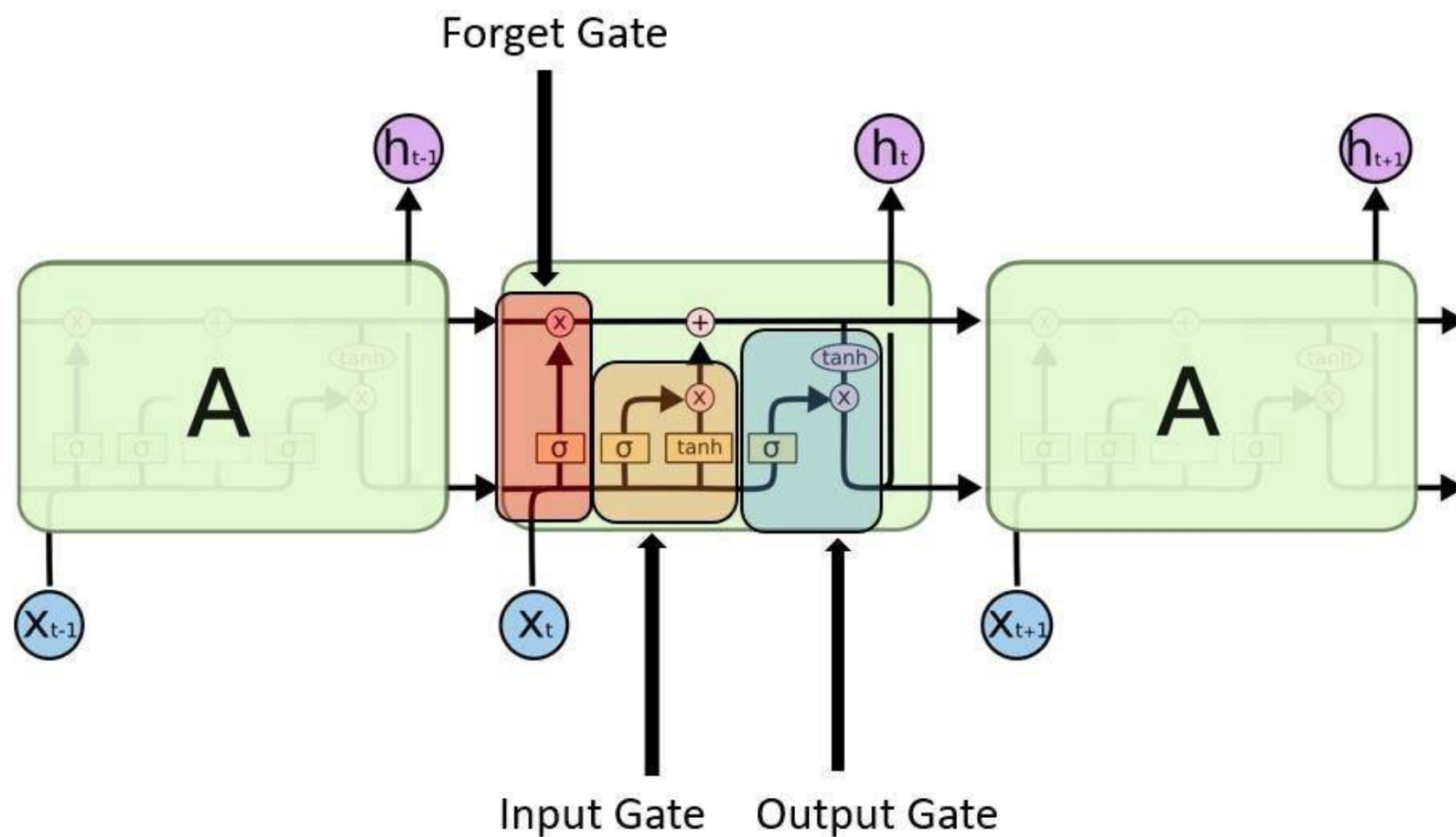


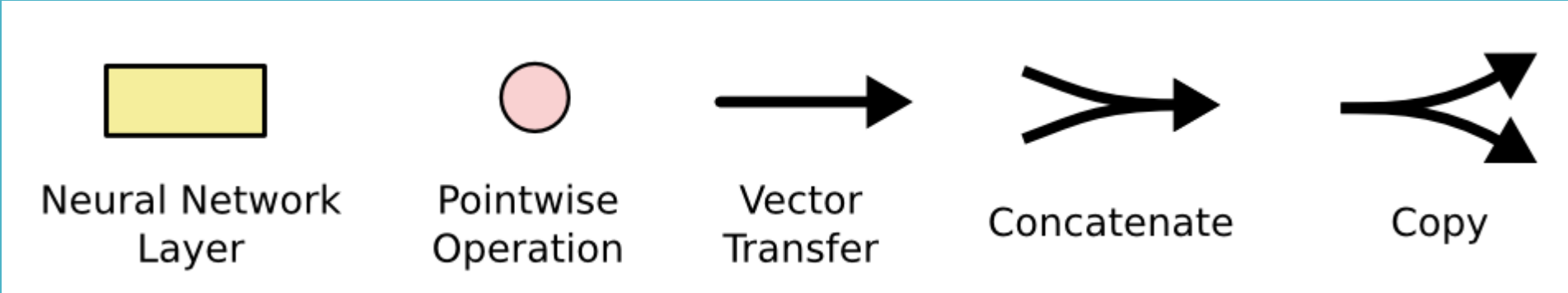
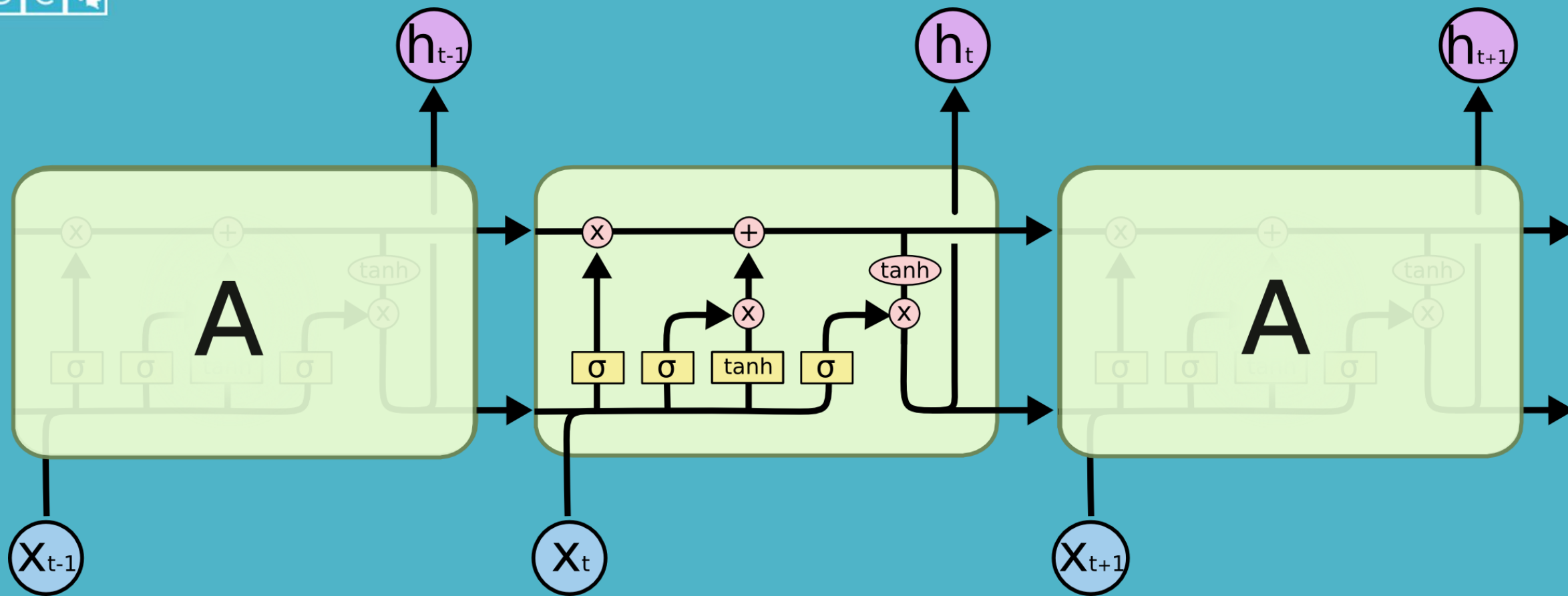
<https://colah.github.io/posts/2015-08-Understanding-LSTMs/>

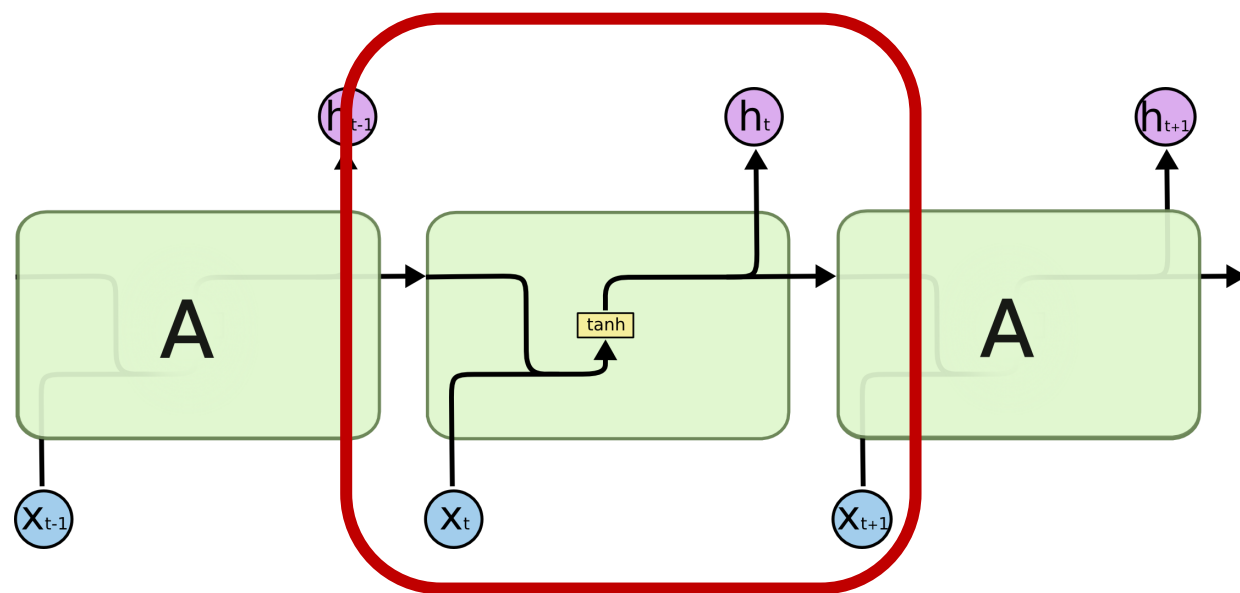
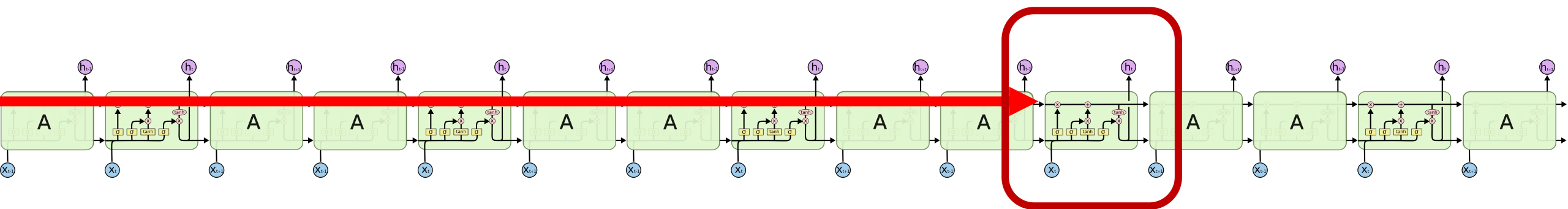


# Comment les LSTM résolvent le problème du gradient de fuite

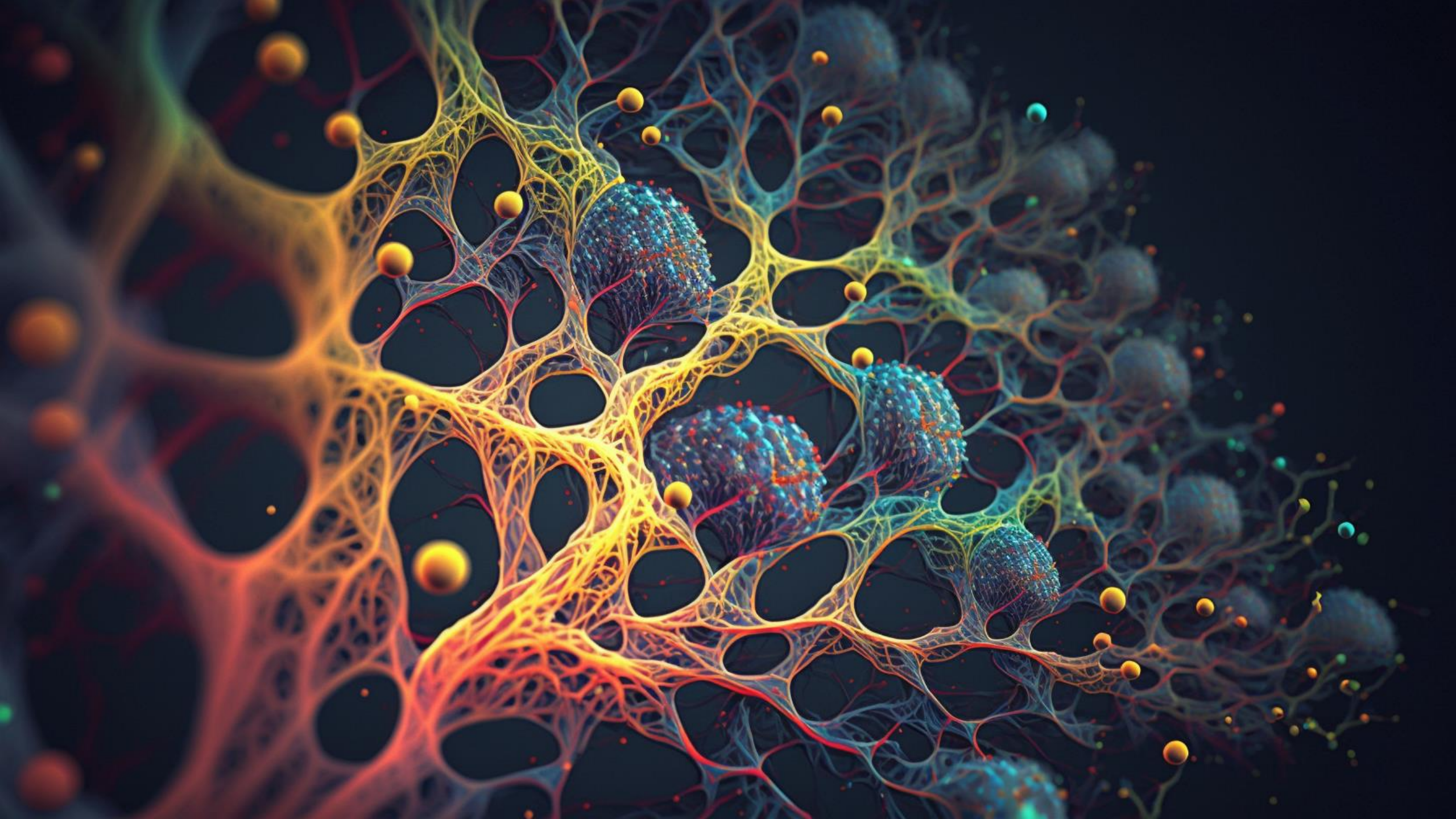
- Cellules de mémoire: Les LSTM utilisent des cellules de mémoire pour stocker des informations sur de longues périodes. Chaque cellule de mémoire est associée à une « porte d'oubli » et à une « porte d'entrée » qui contrôlent l'effacement ou la mise à jour du contenu de la cellule, respectivement. En oubliant ou en conservant sélectivement des informations au fil du temps, les LSTM peuvent maintenir des dépendances à long terme dans les données séquentielles.
- Mécanisme de contrôle : Les LSTM utilisent un mécanisme de contrôle qui permet à l'information de circuler de manière sélective dans le réseau. Le mécanisme de contrôle implique l'utilisation de la porte d'oubli, de la porte d'entrée et de la porte de sortie. La porte d'oubli contrôle les informations à conserver dans la cellule de mémoire et celles à supprimer, en fonction de l'entrée actuelle et de l'état précédent. La porte d'entrée détermine la quantité de nouvelles informations à ajouter à la cellule mémoire, en fonction de l'entrée et de l'état précédent. La porte de sortie contrôle la quantité d'informations de la cellule de mémoire transmise à l'étape suivante de la séquence.
- Non-linéarités : les LSTM utilisent des fonctions d'activation non linéaires qui peuvent aider à éviter le problème du gradient de fuite. Par exemple, la fonction tangente hyperbolique et la fonction d'unité linéaire rectifiée (ReLU) sont couramment utilisées dans les LSTM.
- Initialisation: Les LSTM peuvent être initialisés à l'aide de techniques telles que l'initialisation Glorot ou l'initialisation He, ce qui peut aider à empêcher les gradients de disparaître ou d'exploser pendant l'entraînement.
- Rétropropagation dans le temps (BPTT): Les LSTM peuvent utiliser des techniques telles que la rétropropagation tronquée dans le temps (TBPTT) ou l'écrêtage de gradient pour résoudre le problème de gradient de fuite pendant l'entraînement.
- Dans l'ensemble, la combinaison des cellules de mémoire, du mécanisme de contrôle, des non-linéarités, de l'initialisation et de la rétropropagation dans le temps rend les LSTM efficaces pour résoudre le problème du gradient de fuite dans les RNN. Cela leur permet de maintenir des dépendances à long terme dans les données séquentielles et de bien effectuer des tâches telles que la reconnaissance vocale, la traduction automatique et l'analyse des sentiments.









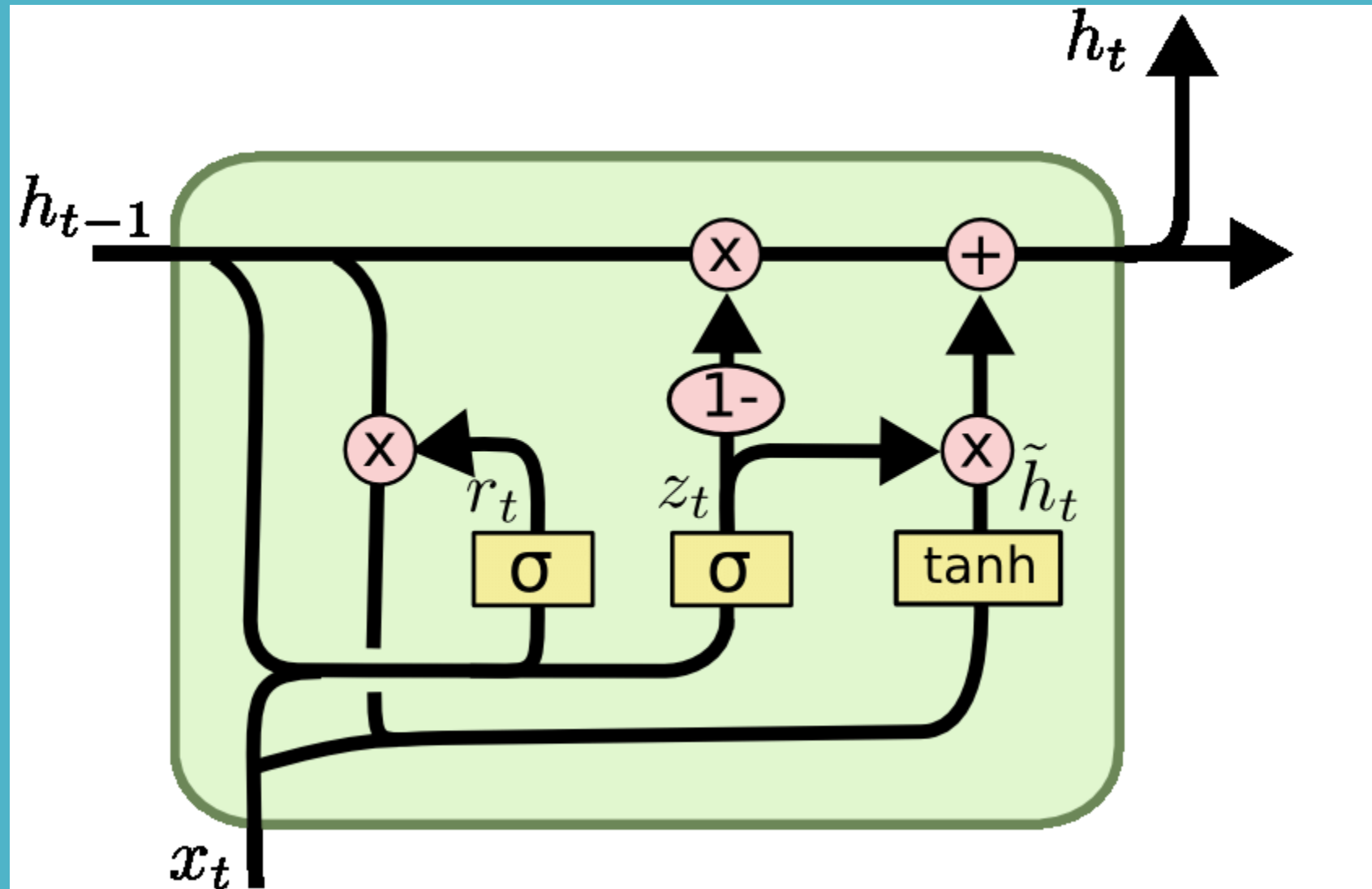


# Gated Recurrent Unit (GRU)

# Gated Recurrent Unit (GRU)

- Les GRU sont un type d'architecture de réseau neuronal récurrent qui a été introduit par Cho et al. en 2014 comme une alternative plus simple aux LSTM.
- Comme les LSTM, les GRU utilisent des portes pour contrôler le flux d'informations à travers le réseau. Cependant, au lieu d'avoir des portes d'entrée et d'oubli séparées, les GRU ont une seule « porte de mise à jour » qui combine les rôles des portes d'entrée et d'oubli.
- La porte de mise à jour dans un GRU détermine la quantité de l'état masqué précédent à conserver et la quantité de la nouvelle entrée à incorporer, en fonction de l'entrée actuelle et de l'état précédent.
- Les GRU ont également une « porte de réinitialisation » qui contrôle la quantité de l'état précédent à oublier, en fonction de l'entrée actuelle.
- En combinant les portes d'entrée et d'oubli en une seule porte de mise à jour, les GRU ont moins de paramètres et sont souvent plus rapides à former que les LSTM. Cependant, ils peuvent ne pas fonctionner aussi bien que les LSTM sur les tâches qui nécessitent une gestion de la mémoire plus complexe.
- Les GRU se sont révélés efficaces pour un large éventail de tâches, notamment la reconnaissance vocale, le sous-titrage d'images et l'analyse des sentiments.
- En général, le choix entre l'utilisation d'un LSTM ou d'un GRU dépendra de la tâche spécifique et de la quantité de données d'entraînement disponibles. Alors que les LSTM sont plus puissants et polyvalents, les GRU peuvent être un bon choix lorsque les ressources de calcul sont limitées ou lorsqu'un modèle plus simple est préféré.







# Comment les GRU simplifient-ils les LSTM ?

- Moins de portes : les GRU n'ont que deux portes (mise à jour et réinitialisation), tandis que les LSTM ont trois portes (entrée, sortie et oubli). Cela signifie que les GRU ont moins de paramètres à former et nécessitent moins de mémoire à stocker.
- Moins d'états : les GRU n'ont que deux états cachés, tandis que les LSTM ont trois états cachés (état de cellule, état de sortie et état d'entrée). Cela signifie que les GRU nécessitent moins de mémoire pour stocker les états cachés, ce qui peut les rendre plus efficaces à former.
- Règle de mise à jour simplifiée : la règle de mise à jour dans les GRU est plus simple que celle des LSTM, ce qui facilite le calcul et l'optimisation pendant la formation. Dans un GRU, la porte de mise à jour contrôle la quantité de l'état masqué précédent qui est reportée à l'étape de temps suivante, tandis que la porte de réinitialisation détermine la quantité de la nouvelle entrée utilisée pour calculer le nouvel état masqué.
- Formation plus rapide : En raison de leur architecture plus simple, les GRU peuvent être plus rapides à former que les LSTM. Cela peut être particulièrement important lorsqu'il s'agit de grands ensembles de données ou lors de la formation de modèles sur des ressources de calcul limitées.
- Performances comparables: Malgré leur architecture plus simple, les GRU peuvent fonctionner aussi bien que les LSTM sur de nombreuses tâches, telles que le traitement du langage naturel, la reconnaissance vocale et le sous-titrage d'images. Dans certains cas, il a même été démontré que les GRU surpassaient les LSTM.
- Dans l'ensemble, les GRU offrent une alternative plus simple aux LSTM qui peuvent être plus efficaces à former et nécessitent moins de ressources pour les stocker. Bien qu'ils ne soient pas aussi puissants que les LSTM dans tous les cas, ils constituent un bon choix pour de nombreuses applications où les ressources de calcul sont limitées ou un modèle plus simple est préféré.

# Applications des GRU

- NLP : Les GRU peuvent être utilisés pour des tâches telles que l'analyse des sentiments, la génération de texte et la traduction automatique.
- Reconnaissance vocale: Les GRU peuvent être utilisés pour des tâches telles que la reconnaissance vocale et l'identification du locuteur.
- Analyse d'images et de vidéos: Les GRU peuvent être utilisés pour des tâches telles que le sous-titrage d'images et l'analyse vidéo.
- Systèmes de recommandation: Les GRU peuvent être utilisés pour des tâches telles que la prédiction du comportement des utilisateurs et la formulation de recommandations.
- Véhicules autonomes: Les GRU peuvent être utilisés pour des tâches telles que la prédiction de trajectoire et l'évitement d'obstacles.





# RNN, LSTM, GRU Summary (1)

- Les réseaux neuronaux récurrents (RNN), les réseaux de mémoire à long terme (LSTM) et les unités récurrentes fermées (GRU) sont tous des types d'architectures de réseaux neuronaux conçues pour traiter des données séquentielles, telles que des séries chronologiques ou le langage naturel. Voici une caractérisation générale de chacune de ces architectures :
- RNN: Les RNN sont des réseaux neuronaux qui ont des boucles qui permettent à l'information de persister et d'être traitée au fil du temps. Cette structure de boucle permet aux RNN de capturer les dépendances entre les éléments d'une séquence, ce qui les rend particulièrement utiles pour des tâches telles que la reconnaissance vocale, la prédiction de texte et l'analyse des sentiments. Cependant, les RNN peuvent avoir du mal à maintenir leur mémoire à long terme et peuvent souffrir de gradients de disparition ou d'explosion lorsqu'ils sont entraînés sur des séquences plus longues.
- LSTM : Les LSTM sont un type de RNN conçu pour résoudre le problème du gradient de fuite en utilisant des cellules de mémoire et des portes pour oublier ou conserver sélectivement des informations au fil du temps. Les LSTM ont une « porte d'entrée » et une « porte d'oubli » qui contrôlent s'il faut mettre à jour ou effacer la cellule mémoire, respectivement. Les LSTM ont également une « porte de sortie » qui contrôle la quantité d'informations de la cellule de mémoire transmise à l'étape suivante de la séquence. En oubliant ou en conservant sélectivement des informations au fil du temps, les LSTM peuvent maintenir des dépendances à long terme et éviter le problème du gradient de disparition.
- GRU : Les GRU sont une variante plus simple des LSTM qui utilisent une « porte de mise à jour » et une « porte de réinitialisation » pour contrôler le flux d'informations à travers le réseau. La porte de mise à jour détermine la quantité de l'état masqué précédent à conserver et la quantité de la nouvelle entrée à incorporer, en fonction de l'entrée actuelle et de l'état précédent. La porte de réinitialisation détermine la quantité de l'état précédent à oublier, en fonction de l'entrée actuelle. Les GRU sont souvent plus rapides à former et nécessitent moins de paramètres que les LSTM, tout en étant efficaces pour de nombreuses tâches.
- Dans l'ensemble, les RNN, les LSTM et les GRU sont de puissantes architectures de réseaux neuronaux qui peuvent être utilisées dans un large éventail d'applications impliquant des données séquentielles. Le choix de l'architecture dépendra de la tâche spécifique et de la quantité de données d'apprentissage disponibles. Alors que les LSTM sont plus puissants et polyvalents, les GRU peuvent être un bon choix lorsque les ressources de calcul sont limitées ou lorsqu'un modèle plus simple est préféré.

# RNN, LSTM, GRU Summary (2)

- Les RNN, LSTM et GRU peuvent tous gérer des données séquentielles, qui peuvent être représentées sous la forme d'une série chronologique ou d'une séquence de symboles.
- Les RNN sont particulièrement utiles pour les tâches qui impliquent des mappages un-à-un ou un-à-plusieurs, où une seule entrée est mappée à une seule sortie ou à plusieurs sorties. Les exemples incluent la reconnaissance vocale, la traduction automatique et l'analyse des sentiments.
- Les LSTM sont bien adaptés aux tâches qui impliquent des dépendances à long terme, où les informations du début de la séquence doivent être mémorisées au fil du temps. Les exemples incluent la génération de texte, la reconnaissance vocale et le sous-titrage d'images.
- Les GRU sont également efficaces pour les tâches qui impliquent des dépendances à long terme, mais sont plus simples et plus rapides à former que les LSTM. Les exemples incluent la classification de texte, l'analyse des sentiments et la traduction automatique.
- Les RNN, LSTM et GRU peuvent tous gérer des séquences de longueur variable, où la longueur de la séquence peut varier d'une entrée à l'autre.
- Les RNN, LSTM et GRU peuvent tous gérer des séquences multivariées, où chaque pas de temps peut avoir plusieurs dimensions d'entrée ou de sortie. Les exemples incluent des séquences d'images ou de vidéos, où chaque pixel ou image peut avoir plusieurs canaux de couleur ou caractéristiques.
- Les RNN, LSTM et GRU peuvent également gérer des séquences échantillonnées de manière irrégulière, où l'intervalle de temps entre chaque entrée peut varier. Les exemples incluent les données de séries chronologiques médicales, où l'intervalle entre les mesures peut être irrégulier.
- Dans l'ensemble, les RNN, LSTM et GRU peuvent gérer un large éventail de types de séquences et peuvent être adaptés à une variété d'applications dans des domaines tels que le traitement du langage naturel, la reconnaissance vocale et l'analyse d'images et de vidéos..

# References

- **A friendly introduction to Recurrent Neural Networks**
  - <https://www.youtube.com/watch?v=UNmqTiOnRfg>