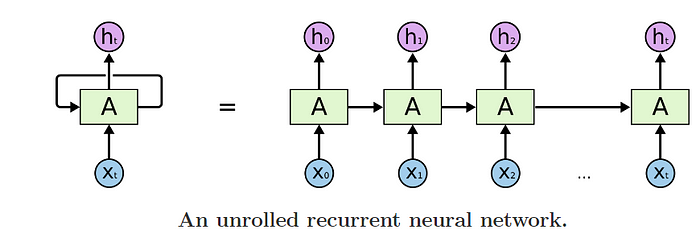
Pour mieux comprendre le LSTM, comprenons un peu les RNN et leurs inconvénients.

**Les réseaux neuronaux récurrents (RNN)** sont conçus pour traiter des données séquentielles en conservant les informations des entrées précédentes dans un état caché. Ils sont couramment utilisés pour des tâches telles que la prédiction de séries chronologiques, le traitement du langage naturel et la reconnaissance vocale.

Dans un RNN, la sortie à chaque pas de temps dépend non seulement de l'entrée actuelle, mais aussi de l'état caché du pas de temps précédent. Cette structure permet au modèle de conserver une forme de mémoire sur les séquences, ce qui le rend adapté aux tâches où les informations passées sont importantes pour prédire les sorties futures.



**Inconvénients des RNN :**

1. **Problème de gradient disparaissant et explosant :**
2. **Difficultés liées aux dépendances à long terme**
3. **Traitement séquentiel (pas de traitement parallèle)**
4. **Mémoire à court terme**

Pour pallier ce manque, LSTM a été créé en 1995.

**Qu'est-ce que LSTM ?**

*LSTM signifie****Long Short-Term Memory (LSTM)****.****Une mémoire à long court terme****est un type de réseau neuronal récurrent (RNN) conçu pour surmonter les limites des RNN standard, notamment pour la rétention d'informations sur de longues séquences. Contrairement aux RNN traditionnels, qui peinent à mémoriser des informations sur de longues périodes en raison du***problème du gradient nul***, les LSTM disposent d'un système de mémoire intégré qui leur permet de stocker, de mettre à jour et de récupérer les informations pertinentes plus efficacement.*

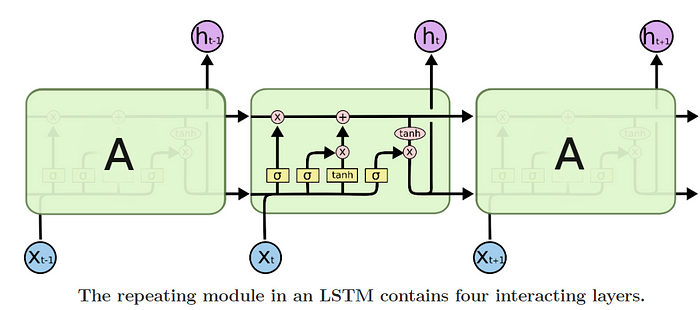


image lstm : blog de colah

**Composants clés :**

* **Cellule mémoire (Ct) :** cœur du LSTM, responsable de la rétention des informations au fil du temps. Elle permet au modèle de mémoriser des détails importants sur de longues séquences.
* **Porte d'oubli :** détermine les informations de la cellule mémoire à oublier ou à supprimer en fonction de l'entrée actuelle et de l'état masqué précédent. Elle aide le modèle à oublier les informations non pertinentes ou obsolètes.
* **Porte d'entrée :** détermine les nouvelles informations à stocker dans la cellule mémoire. Cette porte met à jour la cellule mémoire avec les nouvelles données pertinentes.
* **Porte de sortie :** contrôle les informations de la cellule mémoire à utiliser pour produire la sortie actuelle. Elle détermine la part de la mémoire qui influencera la prédiction suivante.

Décomposons le fonctionnement de chaque porte dans un LSTM simple :

**1. Oubliez la porte :**

* **Objectif :** Décider quelles informations éliminer de l’état de la cellule.
* **Fonctionnement :** L'état caché précédent et l'entrée actuelle sont analysés et transmis à une fonction sigmoïde. Celle-ci génère des valeurs comprises entre 0 et 1, où 0 signifie « tout oublier » et 1 « tout conserver ».

**Exemple :** Si le sujet actuel change dans un modèle de langage, nous pourrions vouloir oublier des informations sur le sujet précédent (comme le genre).

**Équation:**

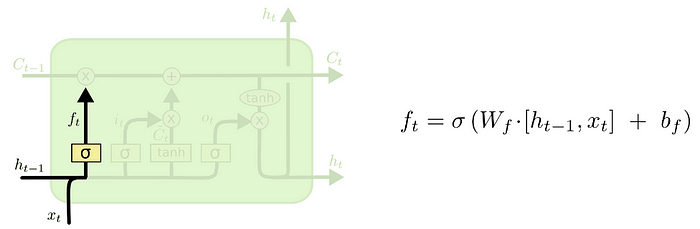


image : colah blog

où σ — est la fonction sigmoïde qui convertit les valeurs entre 0 et 1,

Wf — pondérations associées à l'état caché et à l'état actuel

**h** t-1 — Sortie de l'horodatage précédent également appelé état caché transmis en entrée.

Xt — entrée d'horodatage actuel

bf — valeur de biais

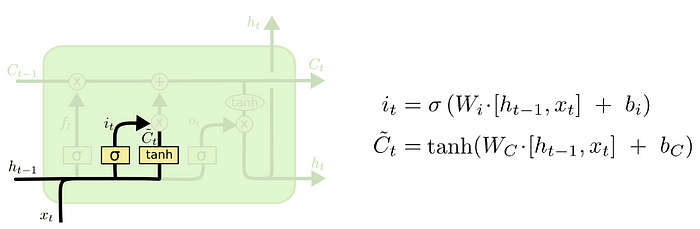
**2. Porte d'entrée :**

* **Objectif :** Décider quelles nouvelles informations stocker dans l’état de la cellule.
* **Comment ça marche :** Deux étapes se déroulent ici :

1. Une couche sigmoïde décide quelles parties des nouvelles informations mettre à jour.
2. Une couche tanh crée de nouvelles valeurs candidates à ajouter à l'état de la cellule.

**Exemple :** Dans un modèle de langage, si nous rencontrons un nouveau sujet, la porte d’entrée ajouterait le sexe du nouveau sujet à l’état de la cellule.

**Équation:**

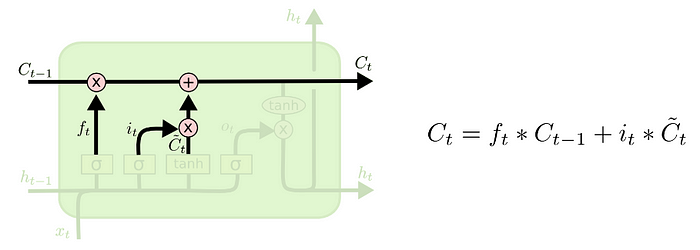


Où Wi et Wc sont des poids et bi et bc sont des valeurs biaisées.

D'autres, comme la porte oubliée.

**3. Mise à jour de l'état de la cellule :**

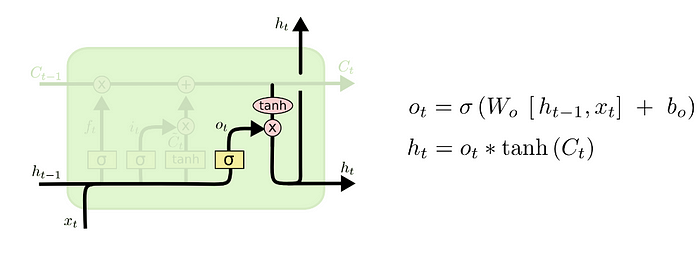
* **Objectif :** Mettre à jour l'état de la cellule en combinant les portes d'oubli et d'entrée.
* **Comment cela fonctionne :** l'ancien état de la cellule est multiplié par la sortie de la porte d'oubli (pour oublier les informations non pertinentes), et le résultat est ajouté aux nouvelles valeurs candidates (pour stocker les nouvelles informations pertinentes).



**4. Porte de sortie :**

* **Objectif :** Détermine quelles informations seront générées à partir du pas de temps actuel.
* **Comment cela fonctionne :** une couche sigmoïde détermine quelles parties de l'état de la cellule seront générées, et l'état de la cellule est transmis via une fonction tanh pour mettre à l'échelle les valeurs entre −1 et 1. La sortie finale est une version filtrée de l'état de la cellule.

**Équation:**



**Avantages de l'utilisation de LSTM :**

1. **Gère les dépendances à long terme :** les LSTM excellent dans la capture de modèles à longue portée dans des données séquentielles.
2. **Atténue le problème de gradient de disparition :** les LSTM résolvent le problème de gradient de disparition courant dans les RNN traditionnels.
3. **Mémoire sélective :** les LSTM conservent ou suppriment sélectivement les informations à l'aide de portes d'oubli, d'entrée et de sortie.
4. **Efficace pour les données séquentielles :** idéal pour des tâches telles que la prévision de séries chronologiques, la reconnaissance vocale, etc.
5. **Polyvalence :** les LSTM sont utilisés pour diverses tâches basées sur des séquences telles que la classification, la régression, la génération de texte.

**Inconvénients du LSTM :**

1. **Coût de calcul élevé :** les LSTM nécessitent beaucoup de ressources et sont plus lents à former en raison de leur structure complexe.
2. **Consommation de mémoire :** ils consomment plus de mémoire, en particulier lors de la gestion de séquences longues ou de grands ensembles de données.
3. **Difficulté de parallélisation :** les LSTM traitent les données de manière séquentielle, ce qui rend la parallélisation difficile et ralentit la formation.
4. **Sur-ajustement avec de petites données :** les LSTM ont tendance à sur-ajuster les petits ensembles de données sans régularisation appropriée.
5. **Complexité de l'architecture :** les LSTM sont plus complexes et plus difficiles à régler que les modèles récurrents plus simples.

**Conclusion:**

Les LSTM ont fait un excellent travail pour gérer les longues séquences, mais leur exécution est très longue (les données étant transmises séquentiellement). Pour résoudre ce problème, **des transformateurs** ont été introduits. Veuillez trouver le lien ci-dessous.