Modèle Seq2seq

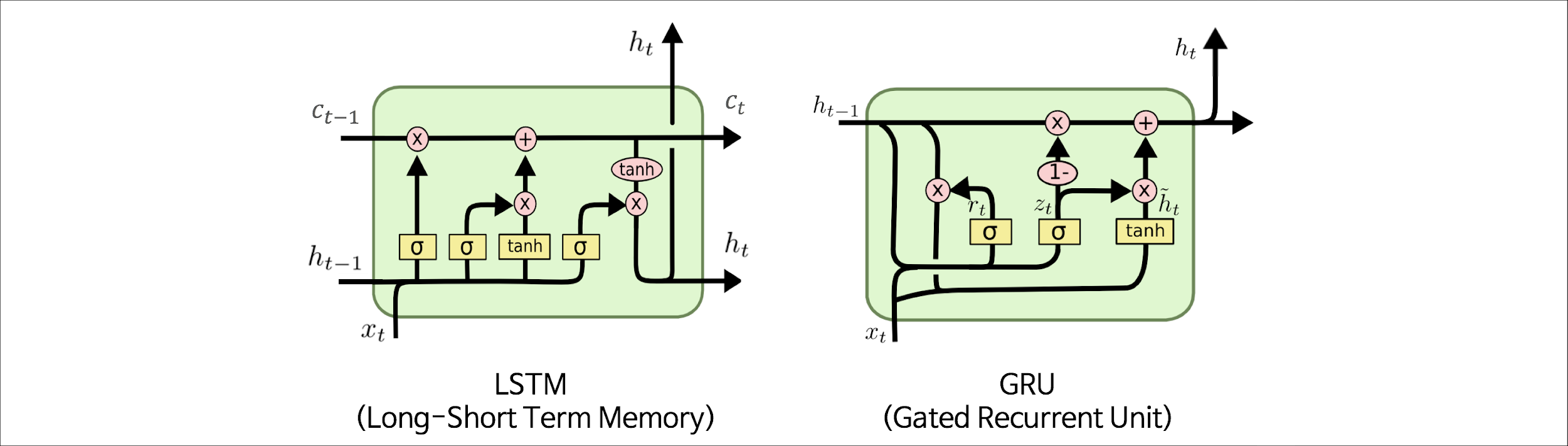
Le **modèle Seq2seq** est un réseau neuronal qui va convertir une séquence de données en entrée vers une nouvelle séquence de données en sortie. Il est utilisé dans de nombreux domaines d’application tels que du traitement automatique du langage naturel (NLP) ou encore pour l’analyse de séries temporelles.

Il sera composé de deux couches de réseaux de neurones récurrents (RNN) : l’encoder et le decoder. La séquence d’entrée et celle de sortie n’ont pas forcément la même longueur.

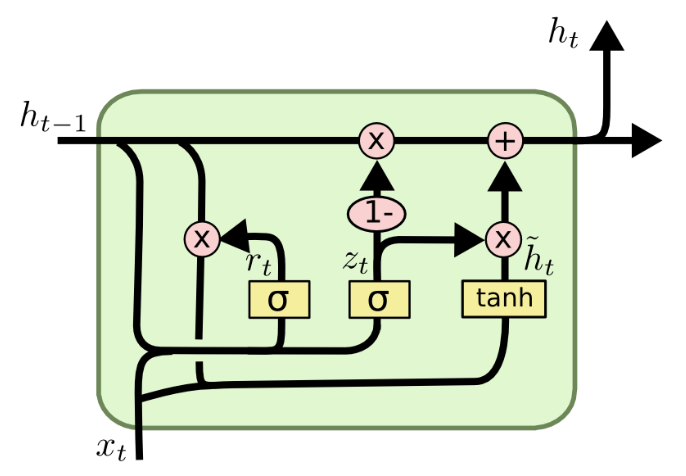
**RNN**

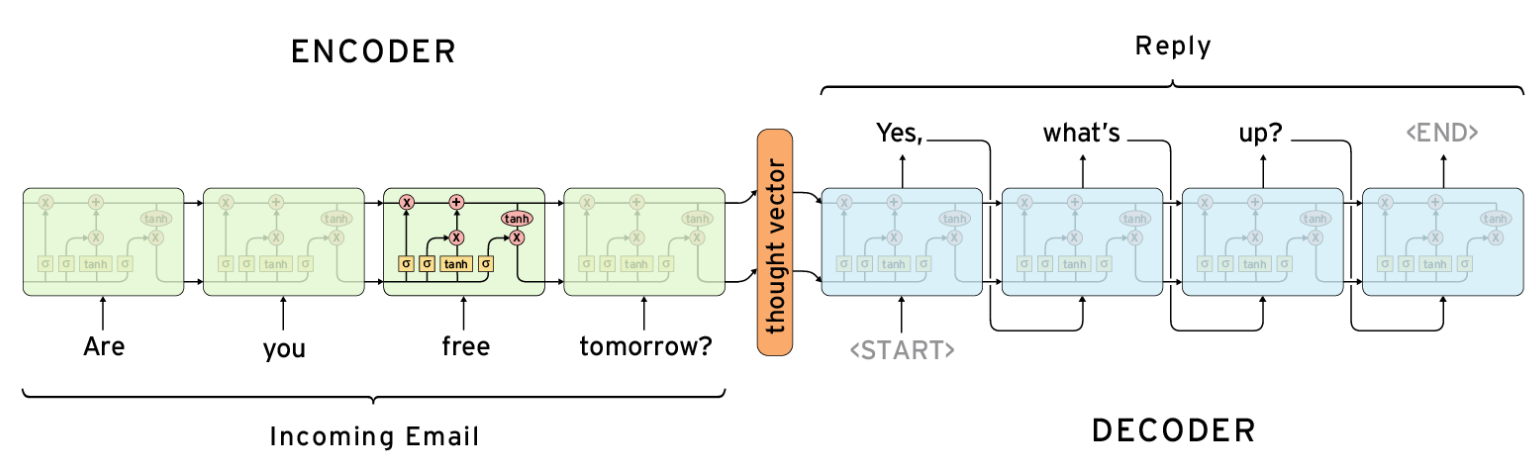
Le réseau de neurones récurrents (RNN) est un réseau d’unités reliées par des arcs possédant des poids. Ce réseau possède des connexions récurrentes puisque l’information est conservée en mémoire en prenant en compte les états précédents. C’est pour cette raison que les RNN sont particulièrement adaptés à des données faisant intervenir un contexte comme des suites de données. Ce réseau ne peut cependant mémoriser qu’une cinquantaine d’itérations et donc rester sur un passé proche, c’est le problème de disparition de gradient. Les réseaux Long short-term memory (LSTM) et les réseaux Gated Recurrent Unit (GRU) vont alors pouvoir répondre à ce problème.

Long short-term memory (LSTM)

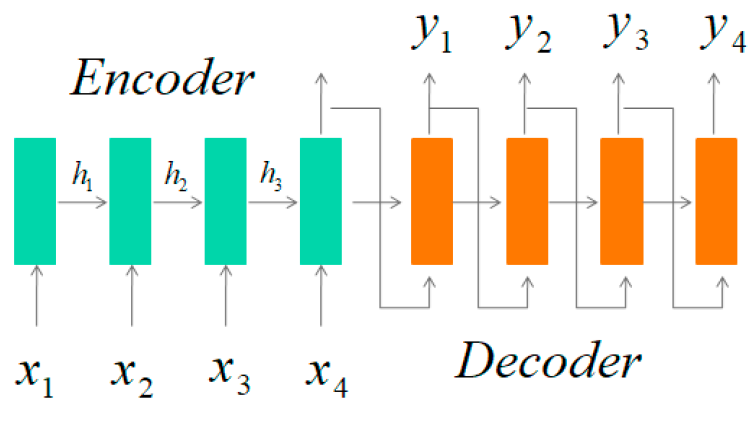


Gated Recurrent Unit (GRU)





<https://towardsdatascience.com/nlp-sequence-to-sequence-networks-part-2-seq2seq-model-encoderdecoder-model-6c22e29fd7e1>

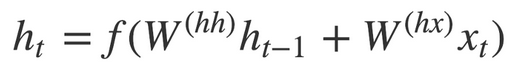


**Encoder**

L’encoder va récupérer le *contexte* de la séquence en entrée sous la forme d’un *vecteur d’état caché*, ainsi que la valeur d’entrée pour générer un nouveau vecteur d’état caché et l’envoyer au decoder pour produire la séquence en sortie.

En général il sera composé de réseaux de type RNN comme LSTM (Long Short-Term Memory) ou GRU (Gated Recurrent Units).

Chaque état caché sera calculé avec la formule :



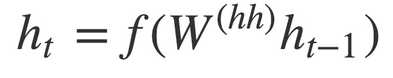
avec ht l’état caché, W le poids et xt le vecteur d’entrée

L’état caché final en sortie de l’encoder est donc un vecteur qui sera en même temps état caché initial du decoder. Ce vecteur va encapsuler les informations de l’ensemble des éléments en entrée pour permettre au decoder de faire ses prédictions.

**Decoder**

La couche decoder prend en entrée le vecteur généré à la sortie de l’encoder. Elle sera également créée avec les architectures RNN (LSTM, GRU).

Chaque unité récurrente prendra un état caché précédent en entrée et produira un autre état caché en sortie, selon la formule :



avec ht-1 l’état caché précédent

Le vecteur de sortie sera alors calculé avec la formule suivante :

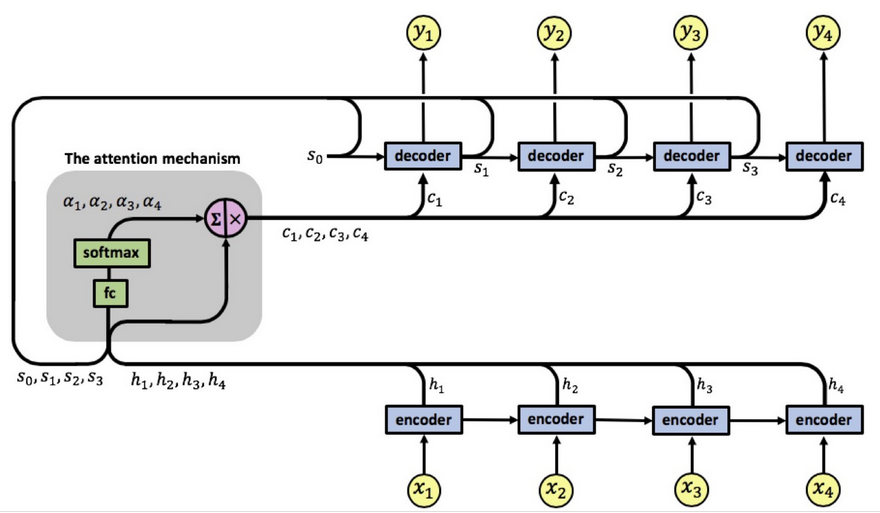


avec Ws leur poids respectif

La séquence en sortie va alors énormément dépendre du contexte défini par l’état caché à la sortie de l’encoder. Il est donc possible de perdre le contexte initial si la séquence devient trop longue. La solution du mécanisme de l’*Attention* a été introduite par les papiers de [Bahdanau et al., 2014](https://arxiv.org/abs/1409.0473) ainsi que [Luong et al., 2015](https://arxiv.org/abs/1508.04025), permettant de conserver le contexte du début jusqu'à la fin.

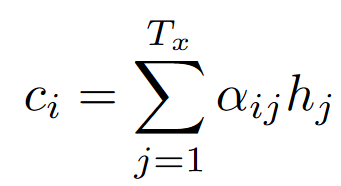
**Attention**

Le mécanisme d’attention permet au modèle de se concentrer sur une partie de la séquence d’entrée à chaque étape de la génération de la séquence de sortie. Au lieu d’alimenter le decoder avec un vecteur contexte de longueur fixe, un vecteur de contexte dynamique sera utilisé, différent sur chaque étape. L’idée est de prendre en compte uniquement les unités les plus pertinentes en entrée selon l’unité à générer en sortie.

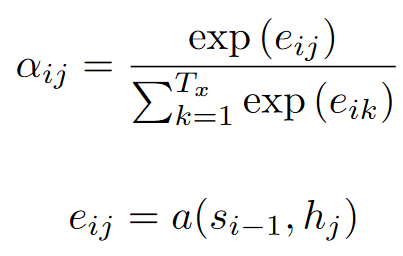


<https://medium.com/datadriveninvestor/attention-in-rnns-321fbcd64f05>

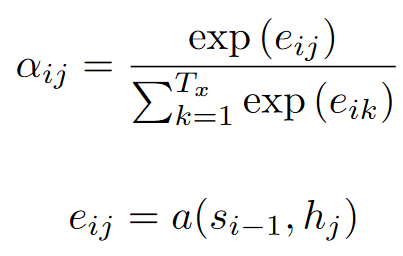
Le vecteur de contexte se calcule :



Chaque poids est calculé par :



avec



papiers sur RNN encoder-decoder :

· Sutskever, I., Vinyals, O., and Le, Q. (2014). Sequence to sequence learning with neural networks.InAdvances in Neural Information Processing Systems (NIPS 2014)<https://arxiv.org/pdf/1409.3215.pdf>

· Cho, K., van Merrienboer, B., Gulcehre, C., Bougares, F., Schwenk, H., and Bengio, Y. (2014a). Learning phrase representations using RNN encoder-decoder for statistical machine translation.InProceedings of the Empiricial Methods in Natural Language Processing (EMNLP 2014). to appear.<https://arxiv.org/pdf/1406.1078.pdf>

papier sur le mécanisme de l’Attention :

D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate.arXiv preprint arXiv:1409.0473, 2014 <https://arxiv.org/abs/1409.0473>