# Modèle :

Dans notre cas d'usage le type de modèle utilisé est un RNN. La première couche est un embeddings qui utilise la matrice pré entrainé GloVe (https://nlp.stanford.edu/projects/glove/) qui se trouve être très performante. Il existe d'autres matrices pré entrainées comme FastText (Facebook), BERT (Google), etc. La taille de la matrice d'embeddings est de 200 dimensions. Cette couche d'embeddings est connectée à une couche LSTM avec comme sortie une dimension de taille 50. Une couche Flatten pour réduire la dimension de sortie du LSTM. Et pour finir une couche Dense de 2 neurones où chaque neurone donne une probabilité sur la classe avec comme fonction d’activation Sigmoid. On a donc en sortie un vecteur à 2 dimensions, un pour chaque classe. On récupère l'index de la colonne avec la fonction Numpy argmax pour connaitre à quel sentiment appartient le texte.

# Critique :

Les résultats sont assez satisfaisants notamment pour une tâche de classification de texte. On note dans la matrice de confusion que les classes sont équilibrés dû à l'homogénéité du dataset (50% pour chaque classe). Un des axes d'améliorations serait de revoir la taille maximale de la séquence. Dans notre cas, chaque séquence fait 150 tokens. Il faudrait réduire la taille pour éviter d’avoir trop de zéros dans les séquences pour limiter l'ajout d'informations inutiles et éviter de biaiser le modèle. Augmenter la taille de la matrice d'embeddings pourrait aussi être bénéfique. En passant de 200 à 300 dimensions on pourrait améliorer la performance du modèle. Choisir d’utiliser des n-grams au lieu de simple tokens aurait été efficace. Le modèle a stagné vers ~84% d'accuracy, il faudrait tester avec un plus gros batch\_size (actuellement 32) pour voir si la descente de gradient n'est pas bloquée dans un minimum local (malgré la couche de Dropout de 0.5). On aurait pu utiliser d’autres modèles de RNN comme GRU, etc. Un modèle de type CNN est aussi envisageable pour les classifications textuelles, un modèle avec des couches Conv1D serait à tester.