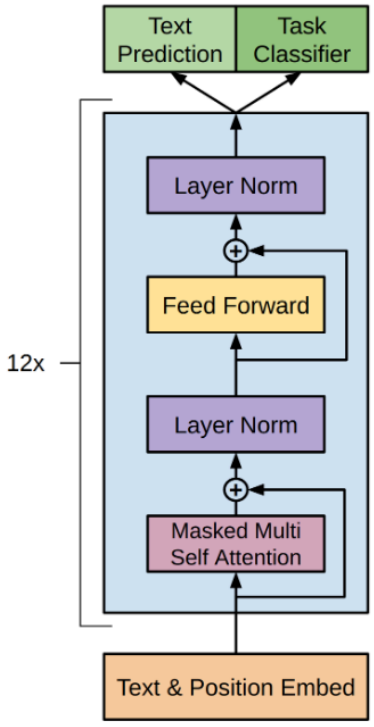
Mamba à partir de zéro (édition RUST)

Audric HARRIS

Dans ce projet, nous allons créer un modèle Mamba de A à Z. Je partagerai tous les sites web et vidéos que j'ai trouvés pour approfondir mes connaissances, sous forme de résumés d'articles.

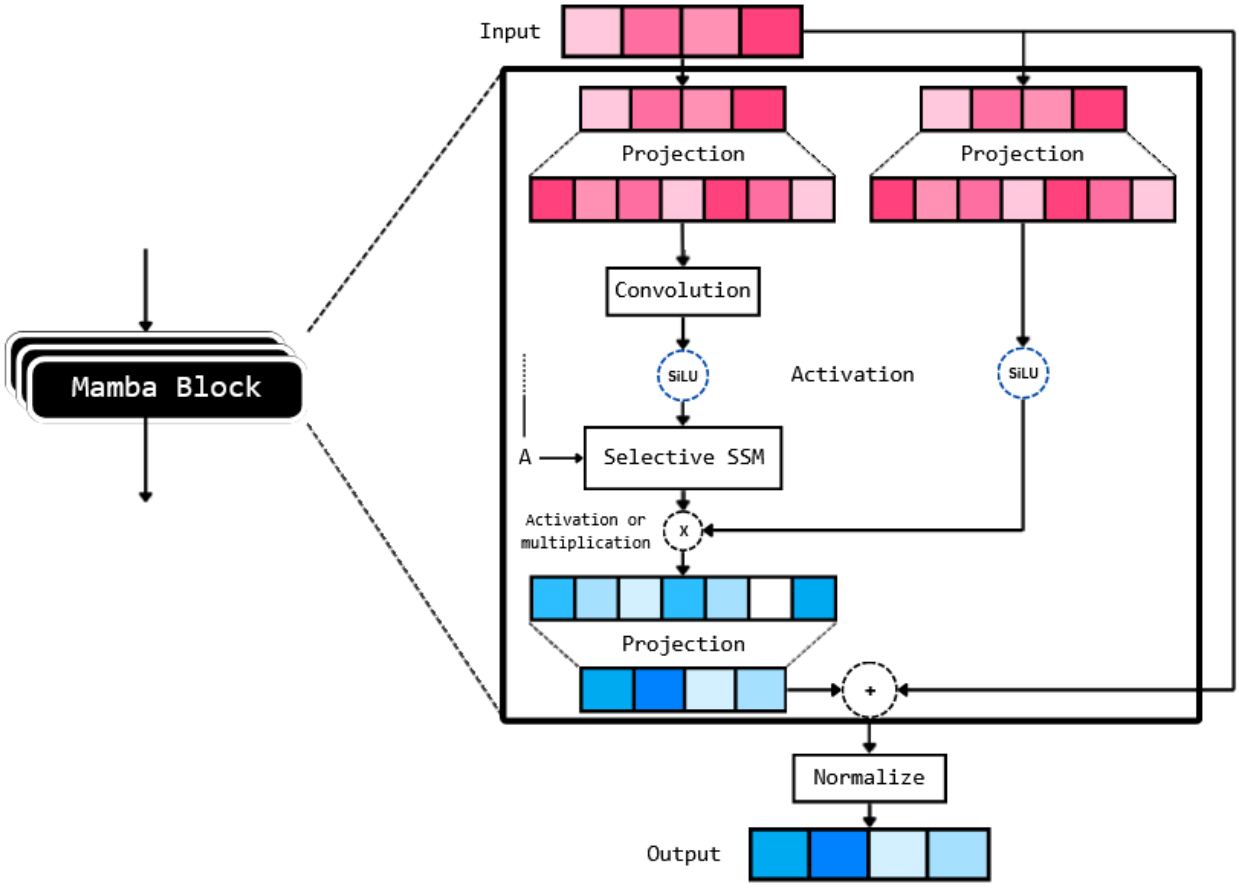
Répondons à la question : qu'est-ce que Mamba ? Comme vous le savez peut-être, ces trois dernières années, l'IA a progressé à une vitesse impressionnante. Toutes les grandes entreprises d'IA s'appuient sur la célèbre architecture GPT, popularisée par le modèle ChatGPT d'OpenAI. Mais une question clé réside dans le coût de calcul de ces modèles ; il est suffisamment élevé pour nécessiter d'immenses centres de données d'IA, consommant plus d'électricité que de nombreuses grandes villes.

https://cdn.openai.com/research-covers/language-unsupervised/language_understanding_paper.pdf



L'architecture GPT tire la majeure partie de son coût de calcul du mécanisme d'auto-attention (en particulier, l'auto-attention multi-tête masquée), qui évolue de manière quadratique avec la longueur de la séquence et peut rendre difficile l'exécution de grandes instances sur du matériel grand public moyen sans optimisations telles que la quantification ou la distillation.

Une équipe de chercheurs a proposé une architecture alternative de traitement du langage naturel bien plus performante : Mamba (https://arxiv.org/pdf/2312.00752v2). Mamba s'appuie sur un composant central appelé « modèle d'espace d'état sélectif » (Selective SSM). Ce modèle rend les paramètres clés du modèle dépendants des entrées, ce qui lui permet de propager ou d'oublier dynamiquement des informations le long de la séquence en fonction de l'entrée actuelle, permettant ainsi un raisonnement basé sur le contenu sans recalculer les interactions par paires.



Comparé à l'architecture GPT (basée sur Transformers), Mamba traite les séquences de manière récurrente, jeton par jeton, à la manière des réseaux neuronaux récurrents, mais avec un mécanisme sélectif qui filtre efficacement les informations pertinentes. Transformers calcule les interactions entre les jetons précédents grâce à l'attention, ce qui conduit à une complexité temporelle quadratique de la longueur des séquences. À l'inverse, l'analyse récurrente de Mamba atteint une complexité temporelle linéaire, ce qui permet une évolutivité vers des séquences beaucoup plus longues (par exemple, jusqu'à 1 million de jetons) sans l'explosion mémoire liée à l'attention. Bien que tous deux utilisent implicitement l'information positionnelle de par leur nature séquentielle, Mamba ne nécessite pas d'intégrations positionnelles explicites comme Transformers, s'appuyant plutôt sur l'évolution de son état pour capturer l'ordre.

Mon objectif est d'implémenter un modèle Mamba complet pour exploiter pleinement son efficacité. D'après l'article, il offre des performances comparables à celles de Transformers de taille similaire pour les tâches de langage et surpasse parfois des modèles deux fois plus grands, tout en étant jusqu'à cinq fois plus rapide en inférence. Mon objectif n'est pas de surpasser GPT ; j'ai simplement besoin d'un modèle de langage naturel performant pour un projet impliquant un PNJ, afin de permettre des interactions IA en temps réel, même sur des ordinateurs bas de gamme.

J'ai réalisé un prototype de ce projet en août-septembre, mais je vais tout reprendre de zéro en utilisant le framework Burn plutôt que PyTorch. Les modèles devraient être disponibles sur kaggle.com , et je vais continuer à tester de nouveaux concepts et idées pour améliorer les performances de l'IA en m'appuyant sur les articles que j'ai lus et mon expérience personnelle. Je ne maîtrise pas encore aussi bien l'apprentissage automatique que je le souhaiterais. Mes calculs ne sont peut-être pas toujours très précis, mais je m'efforcerai de présenter les informations que je partage le plus fidèlement possible et d'éviter toute affirmation dont je ne suis pas absolument sûr.