ANALYSE DÉTAILLÉE DE MIXTURE-OF-RECURSIONS (MOE)

Mokira
Ingénieur Machine Learning
(+229) 019 798 5109
dr.mokira@gmail.com

August 24, 2025

ABSTRACT

Pourquoi MoR est une petite révolution ? Imaginez un modèle de langage qui combine l'efficacité mémoire des architectures à partage de paramètres (comme ALBERT) avec l'intelligence computationnelle du calcul adaptatif (comme Mixture-of-Depths) : c'est exactement ce que propose Mixture-of-Recursions (MoR). Au lieu d'utiliser des couches distinctes, MoR réutilise un même bloc de calcul de manière récursive, tandis qu'un routeur léger décide dynamiquement pour chaque mot s'il doit « sortir » rapidement ou "réfléchir" plus longtemps en passant par des recursions supplémentaires. Résultat ? Une réduction simultanée de la taille du modèle (-50% de paramètres), du temps de calcul (meilleur débit d'inférence) et de la mémoire cache, sans sacrifier la performance — ouvrant la voie à des LLMs à la fois agiles, économiques et puissants. Une avancée architecturale majeure qui mérite d'être explorée dans les moindres détails !

1 Introduction

Aujourd'hui, les intelligences artificielles qui comprennent et génèrent du langage, comme celles qui animent les assistants virtuels, reposent sur des architectures dites « Transformers ». Si elles sont impressionnantes, leur formidable puissance a un coût : une gourmandise excessive en calcul et en mémoire. Pour fonctionner, ces modèles doivent en effet traiter chaque mot d'un texte avec la même intensité. Par exemple, pour comprendre un mot complexe comme « philosophique », un modèle de langue doit lui accorder autant de temps et de ressources qu'à un mot simple comme « le » ou « et ». Cette approche uniforme est coûteuse et inefficace.

Les LLMs sont puissants, mais gourmands en mémoire et en calcul. Et si on pouvait créer un modèle à la fois compact et intelligent, capable de concentrer ses efforts sur les mots qui en valent vraiment la peine ?

C'est la réponse à cette question qui a donné naissance à la **Mixture-of-Recursions (MoR)**. une nouvelle architecture qui permet à un modèle de langage d'allouer intelligemment son « effort de calcul » de manière adaptive, token par token. Imaginez une usine de traitement où les produits simples sont expédiés rapidement après une étape, tandis que les produits complexes passent par plusieurs stations de contrôle pour un travail

approfondi. MoR opère de la même façon : en réutilisant un même groupe de couches de neurones de manière recursive, et en utilisant un mécanisme de décision légère pour déterminer quels mots méritent plus de « réflexion ». Cette méthode unifie pour la première fois les gains en efficacité mémoire (moins de paramètres) et en efficacité computationnelle (moins de calculs superflus), sans compromettre les performances.

Dans ce didacticiel, nous commencerons par un rappel des concepts fondamentaux nécessaires à la compréhension. Ensuite, nous définirons précisément le problème qui se pose. Puis, nous détaillerons le fonctionnement de la solution proposée par MoR pour résoudre le problème, Nous illustrerons cette solution par des exemples concrets et une implémentation simplifiée en Python. Nous discuterons ensuite des apports majeurs et des limites de cette solution, et proposerons des pistes d'amélioration futures, et enfin conclure sur la portée de ce travail.

2 Task description and data construction

We are provided with five datasets from Kaggle: Sales train, Sale test, items, item categories and shops. In the Sales train dataset, it provides the information about the sales' number of an item in a shop within a day. In the Sales test dataset, it provides the shop id and item id which are the items and shops we need to predict. In the other three datasets, we can get the information about item's name and its category, and the shops' name.

Task modeling. We approach this task as a regression problem. For every item and shop pair, we need to predict its next month sales(a number).

Construct train and test data. In the Sales train dataset, it only provides the sale within one day, but we need to predict the sale of next month. So we sum the day's sale into month's sale group by item, shop, date(within a month). In the Sales train dataset, it only contains two columns(item id and shop id). Because we need to provide the sales of next month, we add a date column for it, which stand for the date information of next month.

2.1 Headings: second level

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetuer.

$$\xi_{ij}(t) = P(x_t = i, x_{t+1} = j|y, v, w; \theta) = \frac{\alpha_i(t) a_{ij}^{w_t} \beta_j(t+1) b_j^{v_{t+1}}(y_{t+1})}{\sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i(t) a_{ij}^{w_t} \beta_j(t+1) b_j^{v_{t+1}}(y_{t+1})} \tag{1}$$

2.1.1 Headings: third level

Suspendisse vel felis. Ut lorem lorem, interdum eu, tincidunt sit amet, laoreet vitae, arcu. Aenean faucibus pede eu ante. Praesent enim elit, rutrum at, molestie non, nonummy vel, nisl. Ut lectus eros, malesuada sit

amet, fermentum eu, sodales cursus, magna. Donec eu purus. Quisque vehicula, urna sed ultricies auctor, pede lorem egestas dui, et convallis elit erat sed nulla. Donec luctus. Curabitur et nunc. Aliquam dolor odio, commodo pretium, ultricies non, pharetra in, velit. Integer arcu est, nonummy in, fermentum faucibus, egestas vel, odio.

Paragraph Sed commodo posuere pede. Mauris ut est. Ut quis purus. Sed ac odio. Sed vehicula hendrerit sem. Duis non odio. Morbi ut dui. Sed accumsan risus eget odio. In hac habitasse platea dictumst. Pellentesque non elit. Fusce sed justo eu urna porta tincidunt. Mauris felis odio, sollicitudin sed, volutpat a, ornare ac, erat. Morbi quis dolor. Donec pellentesque, erat ac sagittis semper, nunc dui lobortis purus, quis congue purus metus ultricies tellus. Proin et quam. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Praesent sapien turpis, fermentum vel, eleifend faucibus, vehicula eu, lacus.

3 Examples of citations, figures, tables, references

Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Donec odio elit, dictum in, hendrerit sit amet, egestas sed, leo. Praesent feugiat sapien aliquet odio. Integer vitae justo. Aliquam vestibulum fringilla lorem. Sed neque lectus, consectetuer at, consectetuer sed, eleifend ac, lectus. Nulla facilisi. Pellentesque eget lectus. Proin eu metus. Sed porttitor. In hac habitasse platea dictumst. Suspendisse eu lectus. Ut mi mi, lacinia sit amet, placerat et, mollis vitae, dui. Sed ante tellus, tristique ut, iaculis eu, malesuada ac, dui. Mauris nibh leo, facilisis non, adipiscing quis, ultrices a, dui. [1, 2] and see [3].

The documentation for `natbib` may be found at

`http://mirrors.ctan.org/macros/latex/contrib/natbib/natnotes.pdf`

Of note is the command `\citet`, which produces citations appropriate for use in inline text. For example,

```
\citet{hasselmo} investigated\dots
```

produces

Hasselmo, et al. (1995) investigated. . .

`https://www.ctan.org/pkg/booktabs`

3.1 Figures

Suspendisse vitae elit. Aliquam arcu neque, ornare in, ullamcorper quis, commodo eu, libero. Fusce sagittis erat at erat tristique mollis. Maecenas sapien libero, molestie et, lobortis in, sodales eget, dui. Morbi ultrices rutrum lorem. Nam elementum ullamcorper leo. Morbi dui. Aliquam sagittis. Nunc placerat. Pellentesque tristique sodales est. Maecenas imperdiet lacinia velit. Cras non urna. Morbi eros pede, suscipit ac, varius vel, egestas non, eros. Praesent malesuada, diam id pretium elementum, eros sem dictum tortor, vel consectetuer odio sem sed wisi. See Figure 1. Here is how you add footnotes. [1] Sed feugiat. Cum sociis natoque penatibus

[1] Sample of the first footnote.

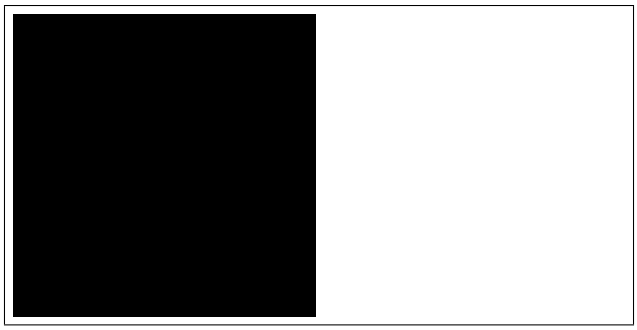

Figure 1: Sample figure caption.

	item_name	item_id	item_category_id
0	! ВО ВЛАСТИ НАВАЖДЕНИЯ (ПЛАСТ.) D	0	40
1	!ABBYY FineReader 12 Professional Edition Full...	1	76
2	***В ЛУЧАХ СЛАВЫ (UNV) D	2	40
3	***ГОЛУБАЯ ВОЛНА (Univ) D	3	40
4	***КОРОБКА (СТЕКЛО) D	4	40

et magnis dis parturient montes, nascetur ridiculus mus. Ut pellentesque augue sed urna. Vestibulum diam eros, fringilla et, consectetuer eu, nonummy id, sapien. Nullam at lectus. In sagittis ultrices mauris. Curabitur malesuada erat sit amet massa. Fusce blandit. Aliquam erat volutpat. Aliquam euismod. Aenean vel lectus. Nunc imperdiet justo nec dolor.

3.2 Tables

Etiam euismod. Fusce facilisis lacinia dui. Suspendisse potenti. In mi erat, cursus id, nonummy sed, ullamcorper eget, sapien. Praesent pretium, magna in eleifend egestas, pede pede pretium lorem, quis consectetuer tortor sapien facilisis magna. Mauris quis magna varius nulla scelerisque imperdiet. Aliquam non quam. Aliquam porttitor quam a lacus. Praesent vel arcu ut tortor cursus volutpat. In vitae pede quis diam bibendum placerat. Fusce elementum convallis neque. Sed dolor orci, scelerisque ac, dapibus nec, ultricies ut, mi. Duis nec dui quis leo sagittis commodo. See awesome Table 1.

3.3 Lists

- Lorem ipsum dolor sit amet

Table 1: Sample table title

Part		
Name	Description	Size (μm)
Dendrite	Input terminal	$\sim$100
Axon	Output terminal	$\sim$10
Soma	Cell body	up to 10^6

- consectetur adipiscing elit.
- Aliquam dignissim blandit est, in dictum tortor gravida eget. In ac rutrum magna.

References

[1] George Kour and Raid Saabne. Real-time segmentation of on-line handwritten arabic script. In *Frontiers in Handwriting Recognition (ICFHR), 2014 14th International Conference on*, pages 417–422. IEEE, 2014.

[2] George Kour and Raid Saabne. Fast classification of handwritten on-line arabic characters. In *Soft Computing and Pattern Recognition (SoCPaR), 2014 6th International Conference of*, pages 312–318. IEEE, 2014.

[3] Guy Hadash, Einat Kermany, Boaz Carmeli, Ofer Lavi, George Kour, and Alon Jacovi. Estimate and replace: A novel approach to integrating deep neural networks with existing applications. *arXiv preprint arXiv:1804.09028*, 2018.