auxiliary-loss-free load balancing strategy for mixture-of-experts

- https://arxiv.org/pdf/2408.15664

概要

- MoEのauxiliary lossを使わずにload balancingを実現する方法を提案
- MoEでは各expertがそれぞれ使われるように学習する必要がある
- load imbalanceが起こると以下につながるから
  - routing collapse (数個のexpertsだけが選択され続けてほかのexpertsが全然学習されない状況)
  - expertsが複数のデバイスに分散されてしまって計算を遅くしてしまう
- このload imbalanceを解決するためにauxiliary lossがある
- auxiliary lossは以下のトレードオフ
  - load balance
  - model performance
- auxiliary lossのハイパラ係数を大きくするとload balanceは達成されるがlanguage modelが満足に学習されなくなる
- auxiliary lossのハイパラ係数を小さくするとlanguage modelの学習は進む一方で数個のexpertsのみが選択され続けてしまう(load imbalanceが起こる)

MoEとは

- https://zenn.dev/deepkawamura/articles/eea77a9a16d037
- N個のexpertを持つMoE layerは以下のようにかける

$$\mathbf{h}_t = \mathbf{u}_t + \sum_{i=1}^{N} g_{i,t}\,\mathrm{FFN}_i\left(\mathbf{u}_t\right),$$

$$g_{i,t} = \begin{cases} s_{i,t}, & s_{i,t} \in \mathrm{Topk}\left(\{s_{j,t} \mid 1 \le j \le N\}, K\right), \\ 0, & \text{otherwise}, \end{cases}$$

$$s_{i,t} = G\left(\mathbf{u}_t{}^T \mathbf{e}_i\right),$$

- Topkは第一引数の集合から値の大きい順に第二引数で指定された数だけ選択するものである
- gの定義からN個のexpertのスコアから大きいものをk個してそれらの値をそのまま重みとして使用して、それら以外の重みは0に更新している
- このとき重みが0になったFFNは順伝搬も逆伝搬も計算しない

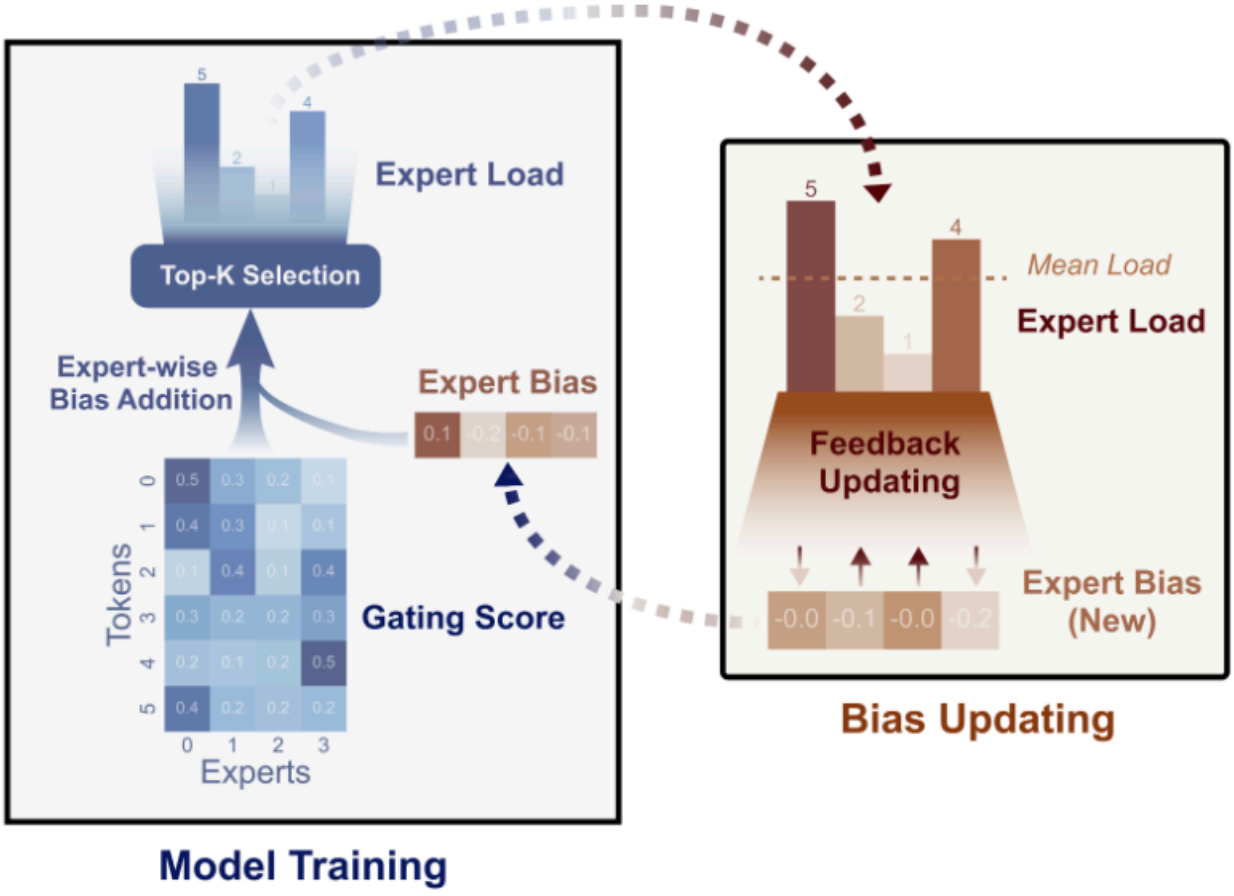
auxiliary lossとは

- load balanceのためのloss

$$g_{i,t} = \begin{cases} s_{i,t}, & s_{i,t} + b_i \in \mathrm{Topk}\left(\{s_{j,t} + b_j \mid 1 \le j \le N\}, K\right), \\ 0, & \text{otherwise}. \end{cases}$$

- Nはexpertの個数
- KはtopKで選択されるexpertの個数
- $s_{i,t}$はrouting score of Expert i for Token t
- 1はindicatorでf_iはfraction of tokens routed to Expert i
- P_iはaverage gating scores of Expert i
- αはハイパラ

auxiliary loss free load balancing strategy(提案手法)



- auxiliary lossを使わずにload balancingを実現する
- 各expertごとにハイパラとしてbias $\{b_i\}_{i=1}^{N}$を用意
- このbiasをスコアに足し合わせて以下のようにtopkで選択する

$$g_{i,t} = \begin{cases} s_{i,t}, & s_{i,t} + b_i \in \text{Topk}\left(\{s_{j,t} + b_j \mid 1 \le j \le N\}, K\right), \\ 0, & \text{otherwise.} \end{cases}$$

- このときbiasを足し合わせた値でtopkで選択して、gating scoreはもとのスコアでbiasを足し合わせたものではないことに注意
- これよりbiasの値によってload balancingを調整できる一方で、gating scoreにはbiasは関与しないのでLLMの誤差逆伝搬には一切関係ない
- 学習過程でbiasは動的に変化させる
  - 多く使われている(重みが大きくなっている)expertに対応するbiasの値を小さくする
  - あまり使われていないexpertに対応するbiasの値を大きくする

- これによりload balanceとmodel performanceの両方を達成できauxiliary lossのtrade offという問題を解決