Introducere Matematică şi Exemplu Numeric pentru LoRA (Low-Rank Adaptation)

1. Introducere

LoRA (*Low-Rank Adaptation*) este o metodă de *fine-tuning* eficient al modelelor mari, care își propune să reducă numărul de parametri antrenabili prin introducerea unei aproximări de rang redus a modificărilor aduse greutăților unui strat neuronal. În loc să actualizeze direct matricea de greutăți $W \in \mathbb{R}^{d\times d}$, LoRA învață două matrici mici A și B astfel încât:

$$W_{\text{nou}} = W + \Delta W, \quad \Delta W = BA$$

unde $\text{rank}(A) = \text{rank}(B) = r \ll d$.

Această aproximare permite ca gradientul să se propage doar prin A și B, reducând semnificativ costul de antrenare și memoria necesară.

2. Reamintire: concepte matematice de bază

Spații vectoriale și transformări liniare

Un strat liniar dintr-o rețea neuronală efectuează o transformare de forma:

$$y = Wx$$

unde:

- $x \in \mathbb{R}^d$ este vectorul de intrare,
- $W \in \mathbb{R}^{d\times d}$ este matricea de greutăți,
- $y \in \mathbb{R}^d$ este ieșirea.

Noțiunea de rang

Rangul unei matrice W reprezintă numărul maxim de coloane (sau linii) independente. O matrice W de dimensiune $d \times d$ și rang $r < d$ poate fi scrisă ca produsul a două matrici mai mici:

$$W \approx BA, \quad \text{cu } A \in \mathbb{R}^{r\times d},\ B \in \mathbb{R}^{d\times r}$$

Aceasta este o *aproximare low-rank*. În LoRA, acest principiu este folosit pentru a modela doar modificarea ΔW.

3. Formula generală a LoRA

Presupunem un strat liniar cu ieșirea originală:

$$y = Wx$$

În LoRA, în loc să actualizăm întreaga matrice W, definim o corecție de rang redus:

$$\Delta W = BA$$

și noua ieșire devine:

$$y = Wx + \frac{\alpha}{r} B(Ax)$$

unde:

- A proiectează vectorul x într-un spațiu de dimensiune mică (r),
- B îl mapează înapoi în spațiul original (d),
- factorul $\frac{\alpha}{r}$ controlează magnitudinea corecției.

Parametri antrenabili

$$\text{Număr parametri LoRA} = d \times r + r \times d = 2dr$$

comparativ cu d^2 pentru un strat complet antrenabil. Pentru valori tipice ($d = 4096$, $r = 16$), se obține o reducere de peste $100\times$ a numărului de parametri.

4. Exemplu numeric ilustrativ

Considerăm valorile:

$$d = 4, \quad r = 2, \quad \alpha = 4 \Rightarrow \text{scaling} = \frac{\alpha}{r} = 2$$

Vectorul de intrare:

$$x = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix}, \quad x \in \mathbb{R}^{4\times 1}$$

Matricea de bază (nemodificată):

$$W = I_{4\times 4}$$

Rezultatul original (fără LoRA):

$$\text{orig\_out} = Wx = x = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix}$$

Matricele LoRA:

$$A = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix}, \quad B = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Pași de calcul

1. Calculăm proiecția redusă:

$$z = Ax = \begin{bmatrix} 1 \cdot 1 + 0 \cdot 2 + 1 \cdot 3 + 0 \cdot 4 \\ 0 \cdot 1 + 1 \cdot 2 + 0 \cdot 3 + 1 \cdot 4 \end{bmatrix} = \begin{bmatrix} 4 \\ 6 \end{bmatrix}$$

2. Reconstruim corecția în spațiul original:

$$\Delta x = Bz = \begin{bmatrix} 4 \\ 6 \\ 4 \\ 6 \end{bmatrix}$$

3. Aplicăm scaling-ul:

$$\text{lora\_out} = \frac{\alpha}{r} \cdot \Delta x = 2 \cdot \begin{bmatrix} 4 \\ 6 \\ 4 \\ 6 \end{bmatrix} = \begin{bmatrix} 8 \\ 12 \\ 8 \\ 12 \end{bmatrix}$$

4. Obținem ieșirea finală:

$$y = x + \text{lora\_out} = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix} + \begin{bmatrix} 8 \\ 12 \\ 8 \\ 12 \end{bmatrix} = \begin{bmatrix} 9 \\ 14 \\ 11 \\ 16 \end{bmatrix}$$

5. Interpretare

- W rămâne fixă (nu se antrenează).
- A și B definesc o ajustare de rang redus.
- Scaling-ul α/r controlează cât de puternic influențează corecția ieșirea.
- Numărul total de parametri antrenabili scade dramatic.

6. Aspecte practice

- De regulă, A este inițializată aleator, iar B cu zero, astfel încât $\Delta W = 0$ la start.
- Doar A și B primesc gradient, W este înghețată (requires_grad = False).
- LoRA este utilizată în modele mari (LLM, Vision Transformers, etc.) pentru *fine-tuning* eficient.

7. Concluzie

LoRA este o metodă elegantă care aplică teoria decompoziției low-rank la rețele neuronale. Ea păstrează capacitatea de exprimare a modelului, dar cu mult mai puțini parametri, ceea ce permite antrenarea rapidă, adaptarea eficientă și stocarea economică a multiplelor versiuni fine-tunate ale aceluiași model de bază.