Internship report, Attention growing networks

Léo Burgund

April 29, 2025

1 Nomenclature

1.1 Dimensions

- b Mini-batch size
- d_e Embedding dimension
- d_s Sequence length
- d_k Query/Keys dimension
- d_v Value dimension
- h Number of heads

1.2 Matrix operations in an attention block

We will first place ourselves in the case where $b = 1$, we study only one instance.

In the case of multi head attention, for each head $i = 1, ..., h$, we have:

- Input $X \in \mathbb{R}^{d_s \times d_e}$
- $W_{Q_i} \in \mathbb{R}^{d_e \times \frac{d_k}{h}}, Q_i := XW_{Q_i} \in \mathbb{R}^{d_s \times \frac{d_k}{h}}$
- $W_{K_i} \in \mathbb{R}^{d_e \times \frac{d_k}{h}}, K_i := XW_{K_i} \in \mathbb{R}^{d_s \times \frac{d_k}{h}}$
- $S_i := \frac{Q_i K_i^\top}{\sqrt{\frac{d_k}{h}}} \in \mathbb{R}^{d_s \times d_s}$
- $A_i := \text{softmax}_{\text{row}}(S)$
- $W_{V_i} \in \mathbb{R}^{d_e \times \frac{d_v}{h}}, V_i := XW_{V_i} \in \mathbb{R}^{d_s \times \frac{d_v}{h}}$
- $H_i := A_i V_i \in \mathbb{R}^{d_s \times \frac{d_v}{h}}$, $H = [H_1, ..., H_h] \in \mathbb{R}^{d_s \times d_v}$
- $W_O \in \mathbb{R}^{d_v \times d_e}$
- Output $Y := HW_O + X \in \mathbb{R}^{d_s \times d_e}$

***Remark* 1.1.** The number of parameters to learn

$$\left(\underbrace{2\left(d_e \frac{d_k}{h}\right)}_{W_{Q_i}, W_{K_i}} + \underbrace{d_e \frac{d_v}{h}}_{W_{V_i}} \right) h + \underbrace{d_v d_e}_{W_O}$$

is the same for any $h \in \mathbb{N}_+^*$.

***Remark* 1.2.** We can easily consider the bias by augmenting the matrices:

$$X' = [X \mid \mathbf{1}] \in \mathbb{R}^{d_s \times (d_e+1)}$$

$$H' = [H \mid \mathbf{1}] \in \mathbb{R}^{d_s \times (d_v+1)}$$

And adding a row of parameters to $W_{Q_i}, W_{K_i}, W_{V_i}, W_O$. For example:

$$W'_{Q_i} = \begin{pmatrix} W_{Q_i} \\ (b^Q)^\top \end{pmatrix} \in \mathbb{R}^{(d_e+1)\times \frac{d_k}{h}}.$$

2 Problem

We study the case where $h = 1$.

We are interested in growing the d_k dimension. We consider the first order approximation, using the functional gradient,

$$\mathcal{L}(f + \partial f(d\theta, d\mathcal{A})) = \mathcal{L}(f) + \left\langle \nabla_f \mathcal{L}(f), \partial f(\partial\theta, \partial\mathcal{A}) \right\rangle + o(\|\partial f(\partial\theta, \partial\mathcal{A})\|).$$

To avoid the softmax's non linearity, we will consider the gradient with respect to the matrix S, just before the softmax.

We then have

$$\mathcal{L}(S + \partial S) = \mathcal{L}(S) + \langle \nabla_S \mathcal{L}(S), \partial S \rangle + o(\|\partial S\|)$$

with

$$\partial S = X\big(W_Q + \partial W_Q\big)(W_K + \partial W_K)^\top X^\top - X W_Q W_K^\top X^\top.$$

We have the following optimization problem:

$$\arg\min_{\partial S} \langle \nabla_S \mathcal{L}(S), \partial S \rangle, \text{such that } \|\partial S\| \leq \gamma$$

$$\arg\min_{\partial W_Q, \partial W_K} \left\| B - X\big(W_Q + \partial W_Q\big)(W_K + \partial W_K)^\top X^\top \right\|_F^2$$

with

$$B := \nabla_S \mathcal{L}(S) + X W_Q W_K^\top X^\top$$

$$= \nabla_S \mathcal{L}(S) + \sqrt{\frac{d_k}{h}} S$$

Which is a low rank regression limited by d_k (if $d_k < d_e$). B is known.

We can approximate $X\underbrace{\big(W_Q + \partial W_Q\big)}_{d_e \times d_k}\underbrace{(W_K + \partial W_K)^\top}_{d_k \times d_e} X^\top$ with a truncated SVD, taking the first d_k singular values.

If we want to grow the inner dimension of the attention matrix by p neurons, we can instead approximate by taking the first $d_{k'} := d_k + p$ singular values.

Hence, instead of approximating a matrix $\underbrace{\big(W_Q + \partial W_Q\big)}_{d_e \times d_k}\underbrace{(W_K + \partial W_K)^\top}_{d_k \times d_e}$, we approximate

$$\underbrace{Z}_{d_e \times d_e} = \underbrace{\mathring{W}_Q}_{d_e \times (d_{k'})} \underbrace{\mathring{W}_K^\top}_{(d_{k'}) \times d_e} = \left[W_Q + \partial W_Q \mid \underbrace{\widetilde{W}_Q}_{d_e \times p} \right] \left[W_K + \partial W_K \mid \underbrace{\widetilde{W}_K}_{d_e \times p} \right]^\top$$

with $\text{rank}(Z) \leq d_{k'}$ (we make the hypothesis that $d_{k'} < d_e$).

We then have the optimization problem

$$\arg\min_{Z} \left\| B - XZX^\top \right\|_F^2 \text{ subject to } \operatorname{rank}(Z) \leq d_{k'}.$$

Which is a low rank regression problem, limited by $d_{k'}$.

Let f such that

$$f(Z) = \left\| B - XZX^\top \right\|_F^2,$$

f is convex.

We have

$$\nabla_Z f = -2X^\top (B - XZX^\top)X,$$

so

$$\nabla_Z f = 0 \iff X^\top X Z^\star X^\top X = X^\top B X. \tag{2.1}$$

In the case where $d_e \leq d_s$ and $\operatorname{rank}(X) = d_e$, then $X^\top X$ is non-singular, and we have the solution

$$Z^\star = (X^\top X)^{-1} X^\top B X (X^\top X)^{-1}.$$

In the general case,

$$\boxed{Z^\star = X^+ B (X^+)^\top = X^+ \left(\nabla_S \mathcal{L}(S) + \sqrt{\frac{d_k}{h}} S \right) (X^+)^\top,}$$

with X^+ the pseudoinverse (Moore-Penrose).

Proof. Suppose $Z^\star = X^+ B (X^+)^\top$. Then,

$$\begin{aligned} X^\top X Z^\star X^\top X &= X^\top X X^+ B (X^+)^\top X^\top X \\ &= X^\top X X^+ B (X^\top X X^+)^\top. \end{aligned}$$

We have

$$\begin{aligned} X^\top X X^+ &= X^\top (X X^+)^\top \text{ by definition of the pseudoinverse} \\ &= X^\top (X^+)^\top X^\top \\ &= (X X^+ X)^\top \\ &= X^\top \qquad \text{by definition .} \end{aligned}$$

Then

$$X^\top X Z^\star X^\top X = X^\top B X$$

we have verified equation (2.1). □

2.1 Factorization

We now have $Z^\star$, which is equal to $\mathring{W}_Q \mathring{W}_K^\top$, and want to factorize it to find $\mathring{W}_Q$ and $\mathring{W}_K$.

If we had $d_{k'} \geq d_e$, we could use the trivial factorization $\mathring{W}_Q = Z^\star, \mathring{W}_K = I_{d_e}$.

However, as most of the time $d_{k'} < d_e$, we have to approximate the factorization.

According to the Eckart–Young–Mirsky theorem, the best approximations $\breve{W}_Q$ and $\breve{W}_K$ to get $\breve{W}_Q \breve{W}_K^\top \approx Z^\star$ with $\operatorname{rank}\left(\breve{W}_Q \breve{W}_K^\top\right) = d_{k'}$ is obtained with a truncated SVD.

Indeed, we have

$$Z^\star = U \Sigma V^\top, \ \Sigma = \operatorname{diag}\left(\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_{d_e}\right).$$

We keep the $d_{k'}$ largest singular values

$$U_{k'} = \left[u_1, ..., u_{d_{k'}}\right],\ V_{k'} = \left[v_1, ..., v_{d_{k'}}\right],\ \Sigma_{k'} = \text{diag}\left(\sigma_1, ..., \sigma_{d_{k'}}\right).$$

We get

$$Z^\star_{k'} = U_{k'}\Sigma_{k'}V_{k'}^\top,\ \ \text{rank}(Z^\star_{k'}) = d_{k'}$$

$$\breve{W}_Q = U_{k'}\Sigma_{k'}^{\frac{1}{2}},\ \breve{W}_K = V_{k'}\Sigma_{k'}^{\frac{1}{2}}.$$

***Remark* 2.1.**

$$\min_{\breve{W}_Q, \breve{W}_K} \left\| B - X\breve{W}_Q\breve{W}_K^\top X^\top \right\|_F^2 = \sum_{i>d_{k'}} \sigma_i^2,\ \ \text{subject to rank}\left(\breve{W}_Q\breve{W}_K^\top\right) \leq d_{k'}$$

***Remark* 2.2.** For implementation:

Keep the matrices apart, for example for the weight matrix of Q :

$$\breve{W}_Q = W'_Q + \partial W'_Q + W_Q^{\text{new}}$$

with (remind that $d_{k'} = d_k + p$)

$$\underset{d_e \times (d_k+p)}{W'_Q} = \left[w_1\ \ ...\ \ w_k \middle| \mathbf{0}_1\ \ ...\ \ \mathbf{0}_p\right]$$

$$\underset{d_e \times (d_k+p)}{\partial W'_Q} = \left[\partial w_1\ \ ...\ \ \partial w_k \middle| \mathbf{0}_1\ \ ...\ \ \mathbf{0}_p\right]$$

$$\underset{d_e \times (d_k+p)}{W_Q^{\text{new}}} = \left[\mathbf{0}_1\ \ ...\ \ \mathbf{0}_k \middle| w_1^{\text{new}}\ \ ...\ \ w_p^{\text{new}}\right]$$

with any vector $w \in \mathbb{R}^{d_e}$, and $\mathbf{0} \in \mathbb{R}^{d_e}$ the 0 vector.

If we wanted to account for the bias, it's the same but include a new last row for each matrix, each vector has one more element.

2.2 Summary

$$Z = X^+\left(\nabla_S \mathcal{L}(S) + XW_QW_K^\top X^\top\right)\left(X^+\right)^\top$$

$$= X^+\left(\nabla_S \mathcal{L}(S) + \sqrt{\frac{d_k}{h}}S\right)\left(X^+\right)^\top$$

and

$$U_{k'}\Sigma_{k'}V_{k'}^\top = \text{SVD}_{\text{trunc } k'}(Z)$$

$$\breve{W}_Q = U_{k'}\Sigma_{k'}^{\frac{1}{2}},\ \breve{W}_K = V_{k'}\Sigma_{k'}^{\frac{1}{2}}.$$

2.3 Notes on Computing

2.3.1 Mini-batch

Note: The "Mini-batch size" can refer either to the machine batch size taken in by the GPU which can optimize computations, or the statistical batch size used to estimate a statistic (this is important as a machine batch may not be of size large enough to get a good estimation of a statistic). In this section, the mini-batch will refer to the statistical batch.

Let b be the mini-batch size, and $i \in \{1, ..., b\}$.

As Z depends on $\nabla_S \mathcal{L}(S)$, the "quality" of the new weight matrices is dependant on b.

To account for the batch, we identified two possibilities:

(i) For each instance, calculate Z_i, get the empirical mean $\bar{Z}_b$ then do $\text{SVD}\left(\bar{Z}_b\right)$ to find $\breve{W}_Q, \breve{W}_K$.

$$\bar{Z}_b = \mathbb{E}_X[Z_i]$$

$$U_{k'}\Sigma_{k'}V_{k'}^\top = \text{SVD}_{\text{trunc } k'}\left(\bar{Z}_b\right)$$

$$\breve{W}_Q = U_{k'}\Sigma_{k'}^{\frac{1}{2}},\ \breve{W}_K = V_{k'}\Sigma_{k'}^{\frac{1}{2}}.$$

We do one SVD per mini-batch.

(ii) For each instance, calculate Z_i, do $\text{SVD}(Z_i)$ to get $\breve{W}_{Q,i}, \breve{W}_{K,i}$, then get the empirical means $\overline{\breve{W}}_Q, \overline{\breve{W}}_K$.

$$U_{k',i}\Sigma_{k',i}V_{k',i}^\top = \text{SVD}_{\text{trunc } k'}(Z_i)$$

$$\breve{W}_{Q,i} = U_{k',i}\Sigma_{k',i}^{\frac{1}{2}},\ \breve{W}_{K,i} = V_{k',i}\Sigma_{k',i}^{\frac{1}{2}}$$

$$\breve{W}_Q = \overline{\breve{W}}_Q = \mathbb{E}_X\left[\breve{W}_{Q,i}\right],\ \breve{W}_K = \overline{\breve{W}}_K = \mathbb{E}_X\left[\breve{W}_{K,i}\right].$$

Here, we do one SVD for each instance.

Note: This is not counting the SVD we will have to do to find X^+.

We choose the first method as it requires only one SVD, which may require less computational ressources, and may truncate (through the SVD) less valuable "information" away, as the SVD is applied after the mean.

2.3.2 Computing Z_i

Note: This is probably useless since we already have computed $S = XW_QW_K^\top X^\top$ during the forward pass.

We denote different ways to compute Z_i.

$$Z = X^+\left(\nabla_S\mathcal{L}(S) + XW_QW_K^\top X^\top\right)(X^+)^\top \tag{2.2}$$

$$Z = X^+\nabla_S\mathcal{L}(S)(X^+)^\top + X^+XW_QW_K^\top X^+X \tag{2.3}$$

Z_i can either be computed by using (2.2), or (2.3), which can be further decomposed.

- If $\text{rank}(X) = d_e$:

$$X^+ = \left(X^\top X\right)^{-1}X^\top \Rightarrow X^+X = I_{d_e}$$

$$Z = X^+\nabla_S\mathcal{L}(S)(X^+)^\top + W_QW_K^\top$$

- If $\text{rank}(X) = d_s$:

$$X^+ = X^\top\left(XX^\top\right)^{-1} \Rightarrow X^+X = X^\top\left(X^\top X\right)^{-1}X$$

$$Z = X^+\nabla_S\mathcal{L}(S)(X^+)^\top + X^\top\left(XX^\top\right)^{-1}XW_QW_K^\top X^\top\left(XX^\top\right)^{-1}X$$

- In the general case , with $r = \text{rank}(X)$:

$$Z = X^+\nabla_S\mathcal{L}(S)(X^+)^\top + V_X\begin{pmatrix}\underbrace{I_r}_{r\times r} & 0\\ 0 & \underbrace{0}_{(e-r)\times(e-r)}\end{pmatrix}V_X^\top W_QW_K^\top V_X\begin{pmatrix}\underbrace{I_r}_{r\times r} & 0\\ 0 & \underbrace{0}_{(e-r)\times(e-r)}\end{pmatrix}V_X^\top.$$

Proof. We compute the SVD for X,

$$\underbrace{X}_{d_s \times d_e} = \underbrace{U_X}_{d_s \times d_s} \underbrace{\Sigma_X}_{d_s \times d_e} \underbrace{V_X^\top}_{d_e \times d_e}, \; X^+ = V_X \underbrace{\Sigma_X^+}_{d_e \times d_s} U_X^\top$$

$$X^+ X = V_X \Sigma_X^+ U_X^\top U_X \Sigma_X V_X^\top = V_X \Sigma_X^+ \Sigma_X V_X^\top$$

with

$$\Sigma_X^+ \Sigma_X = \begin{pmatrix} \underbrace{\Sigma_r^{-1}}_{r \times r} & 0 \\ 0 & \underbrace{0}_{(e-r)\times(s-r)} \end{pmatrix} \begin{pmatrix} \underbrace{\Sigma_r}_{r \times r} & 0 \\ 0 & \underbrace{0}_{(s-r)\times(e-r)} \end{pmatrix} = \begin{pmatrix} \underbrace{I_r}_{r \times r} & 0 \\ 0 & \underbrace{0}_{(e-r)\times(e-r)} \end{pmatrix}$$

□