THE DEEP LEARNING DUMP

ERNEST YEUNG ERNESTYALUMNI@GMAIL.COM

From the beginning of 2016, I decided to cease all explicit crowdfunding for any of my materials on physics, math. I failed to raise *any* funds from previous crowdfunding efforts. I decided that if I was going to live in *abundance*, I must lose a scarcity attitude. I am committed to keeping all of my material **open-sourced**. I give all my stuff *for free*.

In the beginning of 2017, I received a very generous donation from a reader from Norway who found these notes useful, through *PayPal*. If you find these notes useful, feel free to donate directly and easily through PayPal. PayPal does charge a fee, so I also have a Venmo, ernestyalumni, and CashApp (via email `mailto:ernestyalumni@gmail.com`).

gmail : ernestyalumni
linkedin : ernestyalumni
twitter : ernestyalumni

Contents

Abstract. Everything Deep Learning, Deep Neural Networks

Part 1. Deep Neural Networks

1. Gaussian Processes

Yang (2021) for Tensor Programs I[1]

Part 2. Transformer Networks

Including *Attention*

2. Transformers

2.1. Input. See Turner (2023) [2].

Let input data s.t. sequence of N $\mathbf{x}_n^{(0)}$ of dim. D, $n = 0, 1, \dots N-1$, $\mathbf{x}_n^{(0)} \in F^D$, where F is some field (i.e. some data type such as float, double, etc.).

Let matrix $X^{(0)} \in F^{D \times N}$ or $\text{Mat}_F(D, N)$, a sequence of N arrays of dim. D collected into a matrix.

Let $M \in 0, 1, \dots$ i.e. $M \in \mathbb{Z}^+$.

The goal is to map $X^{(0)}$ to $X^{(M)} \in \text{Mat}_F(D, N)$ i.e. $X^{(M)}$ of size $D \times M$ s.t. since $x_n = X_{;n}^{(M)}$ is a vector of features representing the sequence at location of n in the sequence.

2.2. Attention $A^{(m)}$. Consider output vector at location n, $\mathbf{y}_n^{(m)}$, where

$$\mathbf{y}_n^{(m)} = \mathbf{x}_{n'}^{(m-1)} A_{n'n}^{(m)}, \quad n' = 0, 1, \dots N-1 \tag{1}$$

(Eq. (1) of Turner (2023) [2]), where $A_{n'n}^{(m)} = A^{(m)}$ is called the attention matrix, $A^{(m)} \in \text{Mat}_F(N, N)$ and normalizes over its columns:

$$\sum_{n'=1}^{N} A_{n'n}^{(m)} = 1 \tag{2}$$

2.3. Projection of Q, K, V, queries, keys, and values. Recall an input $\mathbf{x}_n = X_{;n}^{(M)} \in F^D$. Recall the linear transform resulting so-called queries or query vectors:

$$\mathbf{q}_{h;n}^{(m)} = U_{q;h}^{(m)} \mathbf{x}_n^{(m-1)} \in F^K, \quad U_{q;h}^{(m)} \in \text{Mat}_F(K, D)$$

where $h = 0, 1, \dots H-1$ with H heads in Turner's notation (Turner (2023)[2]). Compare this with NVIDIA's notation, [3], $i = 0, 1, \dots \texttt{nHeads} - 1$, so that $H \equiv \texttt{nHeads}$.

Generalize K in dimensiosn $k \times D$ of $U_{q;h}^{(m)}$ to $\texttt{qSize} \equiv D_q$, i.e.

$$\mathbf{q} \equiv \mathbf{q}_{h;n}^{(m)} = U_{q;h}^{(m)} \mathbf{x}_n^{(m-1)} \in F^{D_q}, \quad U_{q;h}^{(m)} \in \text{Mat}_F(D_q, D)$$

See 7.2.45. `cudnnSetAttnDescriptor`

Date: 19 July 2023.
Key words and phrases. Deep Learning, Deep Neural Networks.

References

[1] Greg Yang. "Tensor Programs I: Wide Feedforward or Recurrent Neural Networks of Any Architecture are Gaussian Processes." arXiv:1910.12478v3 8 May 2021
[2] Richard E. Turner. "An Introduction to Transformers". arXiv:2304.10557v3 cs.LG 4 Jul 2023
[3] NVIDIA. "7.2.45 `cudnnSetAttnDescriptor`". cuDNN API Documentation. 7.2.45. `cudnnSetAttnDescriptor`