Deep Learning - Homework 2

99222 - Frederico Silva, 99326 - Sebastião Carvalho

January 6, 2024

Contents

1 Question 1

1.1 Question 1.1

Using a single attention head, the output Z is given by:

$$Z = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{1}$$

The complexity of computing this Z is the complexity of computing all the matrix multiplications and the softmax.

Starting with the dimensions of each matrix, we have: $Q \in \mathbb{R}^{L \times D}, K \in \mathbb{R}^{L \times D}, V \in \mathbb{R}^{L \times D}$

The complexity of computing a matrix product AB where $A \in \mathbb{R}^{m \times n}$ and $B \in \mathbb{R}^{n \times p}$ is $O(mnp)$. We can prove this since to compute the element c_{ij} of the matrix $C = AB$ we need to compute the dot product of the i-th row of A with the j-th column of B, and this dot product has complexity $O(n)$, because it is the sum of n products of real numbers. Since we need to compute mn elements of C, the complexity of computing C is $O(mnp)$.

Let's consider $P = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)$. With this P we can compute $Z = PV$.

1

In our case, we first need to compute QK^T and since $Q \in \mathbb{R}^{L \times D}$ and $K^T \in \mathbb{R}^{D \times L}$, the complexity of this operation is $O(L^2 D)$.

Then, we need to divide each element of QK^T by $\sqrt{d_k}$, which has complexity $O(L^2)$. Finally, we need to compute the softmax of each row of the matrix $QK^T/\sqrt{d_k}$, which has complexity $O(L^2)$. The complexity of computing P is $O(L^2 D + L^2 + L^2) = O(L^2 D)$. Since $P = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)$, $P \in \mathbb{R}^{L \times L}$.

Lastly we need to compute the matrix product PV. Since $P \in \mathbb{R}^{L \times L}$ and $V \in \mathbb{R}^{L \times D}$, the complexity of this operation is $O(L^2 D)$.

With this the final complexity of computing Z is $O(L^2 D)$, since the complexity of computing P is $O(L^2 D)$ and the complexity of computing PV is $O(L^2 D)$.

Let's consider the numebr of hidden units (D) is fixed, so the complexity of computing Z is $O(L^2)$.

This may cause a problem for long sequences of text, since the complexity of computing Z is $O(L^2)$, where L is the length of the sequence. This means that the complexity of computing Z is quadratic in the length of the sequence, which is not good for long sequence inputs.

1.2 Question 1.2

For this exercise we will use the McLaurin series expansion of the exponential function to approximate the softmax and reduce the computational computational complexity. The McLaurin series expansion of the exponential function is given by:

$$exp(t) = \sum_{n=0}^{\infty} \frac{t^n}{n!}$$

First, considering $exp(t) \approx 1 + t + \frac{t^2}{2}$, we want to create a feature map $\phi : \mathbb{R}^D \to \mathbb{R}^M$ such that, for arbitrary $q \in \mathbb{R}^D$ and $k \in \mathbb{R}^D$ we have $exp(q^T k) \approx \phi(q)^T \phi(k)$.

With this, we want to find a mapping ϕ such that: $\phi(q)^T \phi(k) = 1 + q^T k + \frac{(q^T k)^2}{2}$

The first two terms of the series are trivial, since we can define $\phi(q) = [1, q_1, \ldots, q_n]$. For the third term, we need to decompose the square of the dot product of q and k into a sum of products of the elements of q and k.

For vectors x and z with the same dimension n, we have that $x^T z = \sum_{i=1}^{D} x_i z_i$. Now for the square of the doct product we have:

$(x^T z)^2 = (x_1 z_1 + \cdots + x_n)(x_1 z_1 + \cdots + x_n z_n) =$
$(x_1 z_1)^2 + x_1 z_1 x_2 z_2 + \cdots + x_1 z_1 x_n z_n$
$x2_1 z_2 x_1 z_1 + (x_2 z_2)^2 + \cdots + x_2 z_2 x_n z_n$
$\ldots$
$x_n z_n x_1 z_1 + x_n z_n x_2 z_2 + \cdots + (x_n z_n)^2 =$
$\sum_{i=1}^{n}(x_i)^2(z_i)^2 + \sum_{i=1}^{n}\sum_{j=i+1}^{n} x_i z_i x_j z_j$

This way we can define $\phi(x) = [1, x_1, \ldots, x_n, \sqrt{\frac{1}{2}}x_1 x_1, \sqrt{\frac{1}{2}}x_1 x_2, \ldots, \sqrt{\frac{1}{2}}x_1 x_n, \sqrt{\frac{1}{2}}x_2 x_1, \ldots, \sqrt{\frac{1}{2}}x_2 x_n, \ldots, \sqrt{\frac{1}{2}}x_n x_1, \ldots$

With this $exp(q^T k) \approx \phi(q)^T \phi(k)$, and we can use this to approximate the softmax function.

In terms of dimensionality, if the vector x has dimension D, the vector $\phi(x)$ will have dimension $M = 1 + D + D^2$.

1.3 Question 1.3

1.4 Question 1.4

2 Question 2

2.1 Question 2.1

After running the code, the best configuration was for the learning rate of 0.01. The following plots were generated:
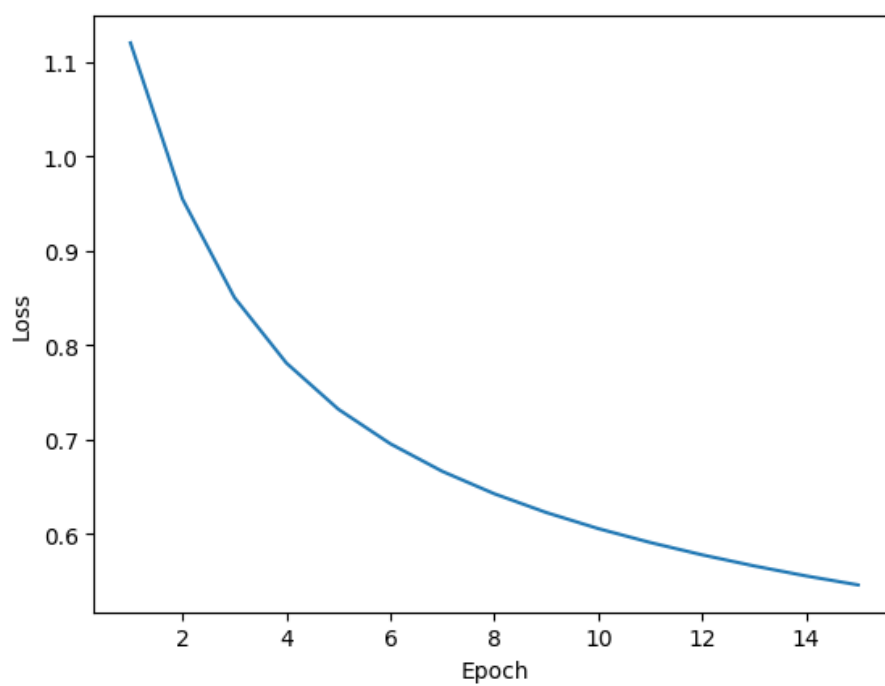

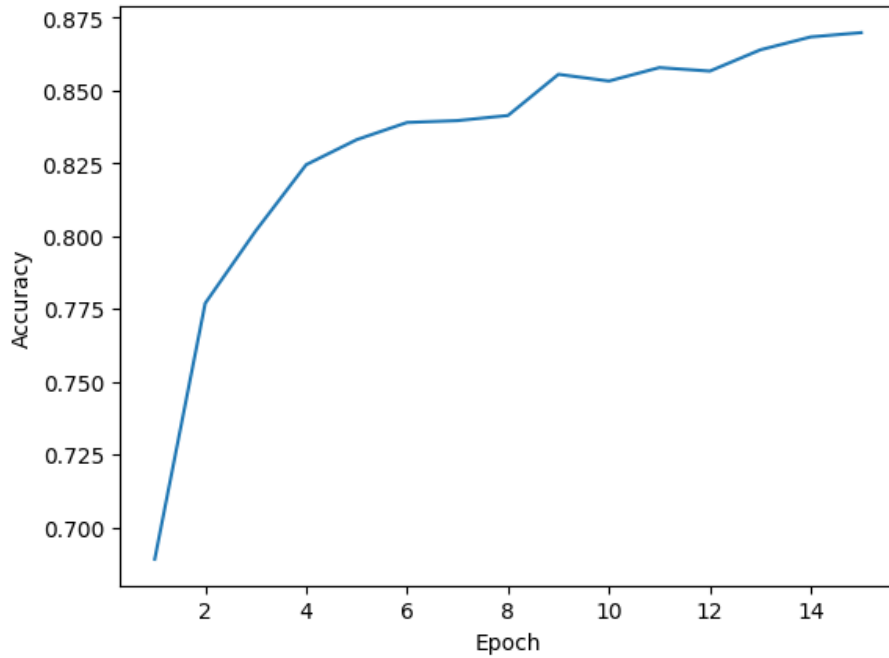Figure 1: Training loss for $\eta = 0.01$

Figure 2: Validation accuracy for $\eta = 0.01$

The final test accuracy was 0.8280.

2.2 Question 2.2

The performance of this network was slightly worse than the previous one, having achieved a final test accuracy of 0.8147.

2.3 Question 2.3

Both network present the same number of parameters, 224892. The difference in performance between the two networks, resides in the use of max pooling layers. Max pooling can help the network focusing on the most important features, making the network more robust to small changes in the input. Furthermore, max pooling can also help with overfitting. In our case, the use of max pooling layers helped the network to achieve a better test accuracy results.

3 Question 3

3.1 Question 3.1

3.2 Question 3.2

3.3 Question 3.3

3.4 Question 3.4

4 Credits

5 Sources

- What is a max pooling layer in CNN?