Exercises for the course
Deep Learning 2
Summer Semester 2025

Machine Learning Group
Faculty IV – Electrical Engineering and Computer Science
Technische Universität Berlin

Exercise Sheet **Attention**

Exercise 1: Gradient computation in attention heads (50 P)

The Transformer model [1] uses a specific form of attention, namely self-attention layers, to extract the task-relevant information from the available features. It herein uses the query, key and value projections of the layer inputs (QKV-attention). In this analytical exercise 1 of this week's sheet, we will focus on the structure of the gradient computation of the attention head module. In the programming exercise 2, you will implement the QKV-attention module used in Transformer models.

For this, recall the following equations:

$$y_j = \sum_i x_i p_{ij} \tag{1}$$

$$p_{ij} = \frac{\exp(q_{ij})}{\sum_{i'} \exp(q_{i'j})}, \tag{2}$$

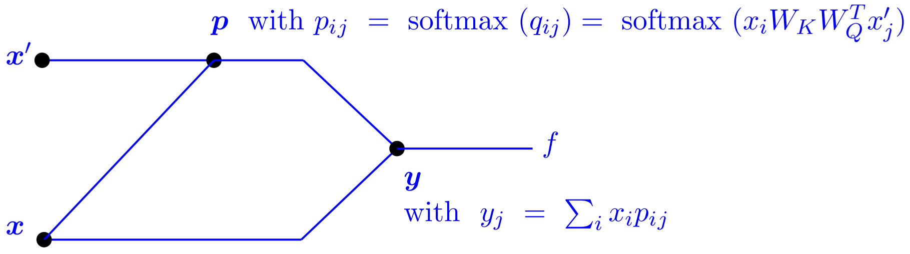
with embedded sequences $\{\boldsymbol{x}, \boldsymbol{y}\} \in \mathbb{R}^{L \times D}$ of length L and hidden dimension D, and p_{ij} the scalar attention weights computed from raw query-key dot-products q_{ij} (that depend on embedding the input $\boldsymbol{x}$ using the W_K weight matrix and $\boldsymbol{x'}$ using W_Q respectively). Inputs $\boldsymbol{x}$ and $\boldsymbol{x'}$ contain the same values but get assigned different functions in the self-attention module.

Hint: Consider the multivariate chain rule in (c) and (d). You do not need to solve for the full analytical solution, only write down the structure of the required gradients. For simplification, you may assume a hidden dimension of $D = 1$.

(a) State the formula for how q_{ij} is computed from the inputs x_i and x'_j using the key and query weight matrices.

$$q_{ij} = x_i^T W_K W_Q^T x'_j \quad \text{with} \quad W_K \in \mathbb{R}^{D \times D_K} \text{ and } W_Q \in \mathbb{R}^{D \times D_Q}.$$

(b) Draw a schematic diagram how the block inputs x_i and x'_j interact with the attention weights p_{ij} to produce the layer output y_j and finally f the output of the neural network (e.g. a class probability score). The computation graph should consider $\{f,\ \boldsymbol{p},\ \boldsymbol{y},\ \boldsymbol{x} \text{ and } \boldsymbol{x'}\}$.


$\boldsymbol{p}$ with $p_{ij} = \text{softmax}(q_{ij}) = \text{softmax}(x_i W_K W_Q^T x'_j)$
$\boldsymbol{x'}$
f
$\boldsymbol{y}$
with $y_j = \sum_i x_i p_{ij}$
$\boldsymbol{x}$

(c) Write down the gradient to compute $\partial f / \partial x'_j$ using the relevant local gradients of the involved variables.

$$\frac{\partial f}{\partial x'_j} = \sum_k \sum_i \frac{\partial f}{\partial y_k} \cdot \frac{\partial y_k}{\partial p_{ik}} \cdot \frac{\partial p_{ik}}{\partial x'_j} \tag{3}$$

(d) Write down the gradient to compute $\partial f / \partial x_i$ using the relevant local gradients of the involved variables.

$$\frac{\partial f}{\partial x_i} = \sum_j \frac{\partial f}{\partial y_j} \cdot \frac{\partial y_j}{\partial x_i} + \sum_l \sum_j \frac{\partial f}{\partial y_j} \cdot \frac{\partial y_j}{\partial p_{lj}} \cdot \frac{\partial p_{lj}}{\partial x_i} \tag{4}$$

$$\tag{5}$$

$$= \sum_j \frac{\partial f}{\partial y_j} \left(\frac{\partial y_j}{\partial x_i} + \sum_l \frac{\partial y_j}{\partial p_{lj}} \cdot \frac{\partial p_{lj}}{\partial x_i} \right) \tag{6}$$

Exercise 2: Programming (50 P)

Download the programming files on ISIS and follow the instructions.

References

[1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.