resit2024

52041DEL6Y Deep Learning 1 23/24 (Period 1.2) · 12 exercises · 86.0 points

MCQ Softmax #31171082

2 pts · Last updated 19 Feb, 2024, 13:12 · Saved in Deep Learning questions

2 PTS · MULTIPLE RESPONSE

Let $f\left(x\right)$ be the softmax operation usually used in classification problems. According to the definition of the softmax, which of these changes still gives you the same loss? $a \in \mathbb{R}$.

- ☐ $e^{-a}\ \ f\left(x+a\right)$ — 0.0

  Feedback

  wrong. f(x+a) does not change the loss, but multiplying it with exp(-a) does. the gradient is the same though.

- ☐ $f\left(a \cdot x\right)$ — 0.0

  Feedback

  wrong.

- ☐ $f\left(x\right)+a$ — 0.0

  Feedback

  wrong. the gradient is the same, but the loss is different.

- ☐ $\frac{1}{a}\,f(x)^{a}$ — 0.0

  Feedback

  wrong.

1x1 conv #31171980

9 pts · Last updated 1 Mar, 2024, 16:08 · Saved in Deep Learning questions

TEXT BLOCK

1x1 convolutions are essential in deep learning. Let's explore some of their features

2 PTS · OPEN · 1/5 PAGE

Jeremy wants to apply two convolutions with kernel sizes 3 and 5 after each other. What do you recommend him do to save compute?

note that there's no nonlin... 2 pts

note that there's no nonlinearity. just multiply the two convs first, then it's just a single linear operation.

3 PTS · MULTIPLE RESPONSE

Let's define a "1x1 *k*-MLP" a module that applies *k* linear layers each followed by a non-linearity. What is true?

- ☐ This 1x1 *k*-MLP can be applied to learn deep features from the raw images and should be quite efficient at it. -2.0
- ☑ **the 1x1 *k*-MLP can be used to downsample deep features by applying strides >1** 1.5
- ☐ if it has hidden dimension (incoming and outgoing) of *d,* overall it either has learnable parameters of d^k or $d^2k + k$ -2.0
- ☑ **Imagine applying this 1x1 *k*-MLP (let's call it *m)* on a set of features *f* multiple times: m(m(f)). This would not raise issues with pytorch's backprop engine.** 2.0

2 PTS · MULTIPLE RESPONSE

Hypernetworks are neural networks that output or modify a neural network's weights based on e.g. some domain signal *x*. Imagine a hypernetwork *h* has been trained to output the weights of the last two layers of network pretrained to generate images of satellite images. Consider a 1x1 conv *c()*, that we apply after the hypernetwork *c(h(x))*.

☐	Provided a learning signal, the 1x1 conv can only linearly affect the output of the generative model	-2.0
☑	**Given data that only consists of domains *x*, and corresponding satellite images *I*, you also need to know what architecture the generative model should have to train c()**	2.0
☐	One advantage of learning *c()* only for the last layer is that we can interpret the values of *c()* in terms of how they modify the final generated image's RGB values.	-2.0

2 PTS · MULTIPLE RESPONSE

Imagine the following architecture: You first apply a large 32x32 convolution (+ nonlinearity) on an image, then you repeat reshaping the output to be of size (dim, N^2) and applying a 1x1 convolution (+ nonlinearity) and reshaping it back to (N,N,dim) to apply a different 1x1 convolution (+ nonlinearity).

☑	**This model achieves a global receptive field by the reshaping operation**	2.0

Feedback
it's true.

☐	This model is faster or equal compared to a CNN at an equal FLOP count	-2.0

Feedback
no because reshaping is slow yet doesn't cost flops

☐	If we change the first 1x1 conv (which is applied on the (dim, N^2) features) to be a grouped 1x1 convolution with N^2 groups, this operation is the same as an attention-operation with k=q and v=1	-2.0

Understanding the Graph Laplacian #31171236

12 pts · Last updated 22 Mar, 2024, 13:16 · Saved in Deep Learning questions

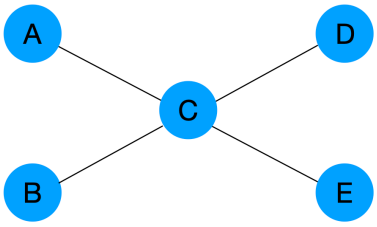
TEXT BLOCK

In this question, we will explore the concept of the graph Laplacian and its implications in different

1 PT · MULTIPLE CHOICE

Basic Structure in Graphs

Consider the graph provided below, which consists of four nodes labeled A, B, C, and D.



which adjacency matrix is correct (ordering is A,B,C,D,E)?

- ○ $\begin{bmatrix} 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \end{bmatrix}$
- ◉ $\begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} \\ \mathbf{1} & \mathbf{1} & \mathbf{0} & \mathbf{1} & \mathbf{1} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} \end{bmatrix}$
- ○ $\begin{bmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 \end{bmatrix}$
- ○ $\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \end{bmatrix}$

1 PT · MULTIPLE CHOICE

Compute M^2. What is entry $M^2_{A,D}$ + $M^2_{C,D}$?

- (●) 1
- ( ) 2
- ( ) 0
- ( ) 3

2 PTS · OPEN · 1/5 PAGE

What do the entries of M^k mean?

M^k_{ij} means how many pat... 2 pts

M^k_{ij} means how many paths with k-hops are connecting node i and j

1 PT · OPEN · 1/10 PAGE

Now consider a graph with self-connections (going from one node to itself). Name an example real-life application where these would be useful?

For a valid answer: twitter... 1 pt

For a valid answer: twitter, network connection, financial transactions, paper references etc.

TEXT BLOCK

In the next sections, we'll explore the graph Laplacian in three different graph configurations: a full

2 PTS · OPEN · 1/5 PAGE

Fully Connected Graph with Uniform Features

Now, assume the graph is fully connected and the features at each node are identical. What values does the graph Laplacian yield for this configuration, and why?

For a valid answer: -1 ever...	2 pts
For a valid answer: -1 everywhere and n-1 in the diagonal!	
Partially correct answer!	1 pt
Partially correct answer!	

2 PTS · OPEN · 1/5 PAGE

Linearly Connected Nodes with Linearly Increasing Features

Consider a new graph where the nodes are connected in a 1-D line (A → B → C → D) and the features at each node increase linearly. The features are given by [10,1], [20,2], [30,3], [40,4]. Discuss the result of the Laplacian in this scenario.

laplacian should be zero be...	2 pts
laplacian should be zero because all features move linearly.	
Partially correct answer!	1 pt
Partially correct answer!	

3 PTS · OPEN · 3/10 PAGE

Circular Graph Structure

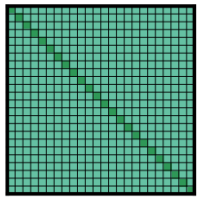
Finally, modify the graph from part (c) by connecting the last node (D) back to the first node (A), forming a circle. Explain what happens to the Laplacian due to this change and why.

laplacian will be nonzero f...	3 pts
laplacian will be nonzero for A and D, because of the sudden "jump" in features	
Partially correct answer!	1 pt
Partially correct answer!	

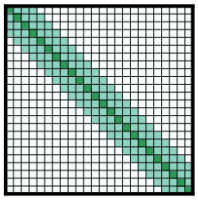
Attention and transformers #31172408

15 pts · Last updated 5 Mar, 2024, 14:05 · Saved in Deep Learning questions

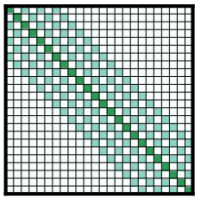
TEXT BLOCK



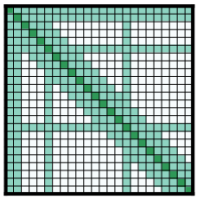
(a) Full n^2 attention



(b) Sliding window attention



(c) Dilated sliding window



(d) Global+sliding window

Figure: Attention Mechanism. Comparing the full self-attention pattern and the configuration of att

The original Transformer model has a self-attention component with $O\left(n^2\right)$ time and memory con

2 PTS · OPEN · 1/5 PAGE

Compare Full n^2 attention with causal attention used in the GPT-style model. Draw a sketch, as in the above figure, illustrating the causal attention matrix for an input with length $n = 2$.

2 100% (2 pts)

[[X,0],[X,X]], generally a triangular matrix

First 0% (0 pts)

Incorrect answer!

2 PTS · MULTIPLE CHOICE

Sliding window attention: Given a fixed window size w, each token attends to $\left(\frac{1}{2}\right) w$ tokens on each side. In a transformer with l layers, what is the receptive field size at the top layer? (assuming w is fixed for all layers).

- (●) *lw*
- ( ) *2lw*
- ( ) *0.5lw*
- ( ) w

2 PTS · MULTIPLE RESPONSE

Dilated sliding window attention: Assuming a fixed d, referred to dilation size, and w for all layers, what is the receptive field size at the top layer?

☐	$(d + w)\, l$	-2.0
☐	$(d + 0.5w)l$	-2.0
☑	$\boldsymbol{dwl}$	2.0
☐	$(lw)^d$	-2.0

2 PTS · MULTIPLE RESPONSE

What is true?

- ☑ **(b) does not reduce the number of learnable parameters**
- ☐ (d) corresponds to the vision transformer (ViT)'s attention pattern
- ☐ the attention values of (b) yield a circulant matrix and can therefore be computed efficiently
- ☐ the values of the entries at *(i,j)* for the attention matrix from (a) and (b) differ by a factor of $\frac{N}{\sqrt{(}w)}$, where *N* is the number of inputs

TEXT BLOCK

Consider auto-regressive generation using a transformer, like GPT-2. To speed up computation, we

3 PTS · OPEN · 3/10 PAGE

How does that save compute? And why are we not saving the query values?

The reason for that is that... 2 pts

The reason for that is that is not needed to recalculate these matrices for doing a forward pass for all the tokens except the current!

We do not need to recalcula... 1 pt

We do not need to recalculate key and values matrices!

2 PTS · OPEN · 1/5 PAGE

How much compute does this save when generating a sentence with *s* words? Consider a full-forward pass to be of compute cost *C*

Valid answer, roughly 2/3 o...	2 pts
Valid answer, roughly 2/3 of computation!	
Partially correct answer!	1 pt
Partially correct answer!	

2 PTS · OPEN · 1/5 PAGE

Consider the case of you working at OpenAI and hosting a language model that answers the queries of multiple users on a server. What problems would you encounter with this approach if the batch-size is larger than 1?

Valid answer/memory issues	2 pts
Valid answer/memory issues	
Partially correct answer!	1 pt
Partially correct answer!	

Initialization #31313119

10 pts · Last updated 28 Feb, 2024, 14:44 · Saved in Deep Learning questions

TEXT BLOCK

Consider the computation between adjacent pre-activations $\mathbf{f}$ (input layer) and $\mathbf{f}'$ (output layer) tha

$$\mathbf{h} = a(\mathbf{f})$$
$$\mathbf{f}' = \boldsymbol{\beta} + \mathbf{W}\,\mathbf{h}$$

with $a()$ the activation function and $\mathbf{W}$, $\boldsymbol{\beta}$ represents the weights and biases.

Assume the pre-activations f_j in the input layer $\mathbf{f}$ have variance σ_f^2. Consider initializing the biases

2 PTS · OPEN · 3/10 PAGE

Derive the mean expression of the pre-activations f_i' in the output layer.

Assume that the distributions between the hidden units and the network weights are independent. The dimensionality of the hidden layer can be denoted as D_h.

$$$E[f_{i}^{'}] = E[\beta_i... 2 pts

$$E[f_i^{'}] = E[\beta_i + \sum_{j=1}^{D_h} W_{ij}h_j]$$

$$= E[\beta_i] + E[\sum_{j=1}^{D_h} W_{ij}h_j]$$

$$= E[\beta_i] + \sum_{j=1}^{D_h} E[W_{ij}]E[h_j]$$
$$= 0$$

Partially correct! 1 pt

Partially correct!

2 PTS · OPEN · 1/2 PAGE

Derive the variance $\sigma^2_{f'}$ for the pre-activations f'_i in the output layer.

Assume that the distributions between the hidden units and the network weights are independent. The dimensionality of the hidden layer can be denoted as D_h.

Hint; make use of the variance identity:

$$\boldsymbol{\sigma}_z^2 = Var[z] = E[(z - E[z])^2] = E[z^2] - E[z]^2$$

.

$$\sigma_{f^{\prime}}^{2} ... 2 pts

$$\sigma^2_{f'} = E[\sigma^2_{f'_i}] - E[\sigma_{f'_i}]^2$$
$$= E[(\beta_i + \sum_{j=1}^{D_h}(W_{ij}h_j)^2] - 0.$$

Then, the expected value of the sum is the sum of expected values (also the bias is initialized with zero), thus:

$$\sigma^2_{f'} = 0 + \sum_{j=1}^{D_h} E[(W_{ij}h_j)^2] - 0$$
$$= \sum_{j=1}^{D_h} E[W^2_{ij}]E[h^2_j]$$
$$= \sum_{j=1}^{D_h} \sigma^2_W E[h^2_j] = \sigma^2_W \sum_{j=1}^{D_h} E[h^2_j]$$

Some steps are correct! 1 pt

Some steps are correct!

2 PTS · OPEN · 3/10 PAGE

Assuming that the input distribution of pre-activations f_i is symmetric about zero, half of these pre-activations will be clipped by the ReLU function, thus, $\mathbb{E}[h^2_j]$ will be half the variance of f_j. Find an expression that connects the variances from f and f'.

$$\sigma_{f^{\prime}}^{2} ... 2 pts

$$\sigma^2_{f'} = \sigma^2_W \sum_{j=1}^{D_h} E[h^2_j] = \sigma^2_W \sum_{j=1}^{D_h} \frac{\sigma^2_{f'}}{2} = \sigma^2_W D_h \frac{\sigma^2_{f'}}{2}$$

2 PTS · OPEN · 3/10 PAGE

Given the previous expression how you would initialize the weights $\mathbf{W}$ for forward and backward pass?

$$\sigma_{h}^{2} = \frac{2}... 2 pts

$$\sigma_h^2 = \frac{2}{D_h}$$

Partially correct! 1 pt

Partially correct!

2 PTS · OPEN · 1/5 PAGE

What would you expect to happen if we initialize all the weights and the biases of a neural network to zero?

Not good idea due to the sy... 2 pts

Not good idea due to the symmetry problem!

Partially correct! 1 pt

Partially correct!

Optimization basics #31172518

9 pts · Last updated 14 Feb, 2024, 13:06 · Saved in Deep Learning questions

2 PTS · OPEN · 1/5 PAGE

Name two advantages of stochastic gradient descent over vanilla gradient descent.

The stochasticity of SGD al...	1 pt
The stochasticity of SGD allows it to escape shallow local minima more easily.	
SGD is computationally mor...	1 pt
SGD is computationally more efficient.	
SGD can be used to train ve...	1 pt
SGD can be used to train very large training samples.	

3 PTS · OPEN · 1/5 PAGE

Describe (i) what a **pathological curve** (i.e. ravine) is, (ii) why they may be problematic during optimization using vanilla (stochastic) gradient descent, and (iii) a way to address this problem.

In a pathological curve, th...	1 pt
In a pathological curve, the gradient is large in one direction and small in another.	
They are problematic for va...	1 pt
They are problematic for vanilla (stochastic) gradient descent because it will likely osccilate along the ridges of the ravine as the gradient is larger in that direction, which results in slower learning.	
We can circumvent this prob...	1 pt
We can circumvent this problem by incorporating momentum in our optimization algorithm.	

2 PTS · OPEN · 1/5 PAGE

Despite the popularity of the Adam optimizer, there are scenarios in which it is preferable to use SGD over Adam. Give one possible reason why it may be preferable to use SGD over Adam in some scenarios.

Adam is capable of finding ...	1 pt
Adam is capable of finding very steep optima, whereas SGD tends to find flatter minima.	
Flat minima generalize bett...	1 pt
Flat minima generalize better to unseen data.	

2 PTS · OPEN · 1/5 PAGE

Second-order optimization method such as Newton's method take the curvature around a point into account in order to update the weights in more promising directions. Why are second-order methods nevertheless rarely used in practice?

Second-order methods are co... 2 pts

Second-order methods are computationally expensive due to the inversion of the Hessian.

Theory: Init & Bn #31172695

10 pts · Last updated 4 Mar, 2024, 16:51 · Saved in Deep Learning questions

TEXT BLOCK

You are training a deep, dense neural network on some flat input X with feature dimensionality of .

2 PTS · OPEN · 1/5 PAGE

At *i*-th layer, your activation output is 32x128, where 32 is the batch size and 128 is the hidden features' dimensionality. You would like to add a *BatchNormalization* layer here. How many parameters in your *BatchNormalization* layer will be trainable, and how many will be not?

128+128=256 trainable, and ...	2 pts
128+128=256 trainable, and 256 non-trainable, adding up to 512.	
Partially correct!	0.5 pts
Partially correct!	

2 PTS · OPEN · 1/5 PAGE

Assume you managed to implement the *BatchNormalization* layer, and you also extended the idea to all other layers, so now each layer block is a combination of $Linear \rightarrow Bn \rightarrow ReLU$ respectively. What change do you expect in your training and why?

regularisation due to noise...	1 pt
regularisation due to noise in batches	
speeds up training	1 pt
speeds up training	

2 PTS · OPEN · 1/5 PAGE

We typically subtract the mean and average from our input data. What happens if we instead apply a BatchNorm layer as the very first layer?

should work fine	2 pts
should work fine	

2 PTS · OPEN · 1/5 PAGE

Now consider applying a LayerNorm as the first layer in a transformer (e.g. a ViT). What would you expect to happen?

Valid answer!	2 pts
Valid answer!	
Partially valid answer!	1 pt
Partially valid answer!	

2 PTS · MULTIPLE RESPONSE

Different norm layers: which statement(s) are true?

- ☐ LayerNorm and BatchNorm are linear layers, ie can be expressed as an affine transformation of the incoming features.
- ☑ InstanceNorm and LayerNorm do not provide the same gradient after re-scaling the activations with $\frac{1}{\sqrt{(d)}}$, where d is the dimensionality
- ☑ Larger batch sizes improve the mean and variance estimates of LayerNorm

  Feedback

  they actually do: since we still have a running mean of feature-wise activations.

- ☑ It is better to initialise BatchNorm layers' running estimate values with random values than with mean=0 and variance=1

CNNs #31223052

6 pts · Last updated 1 Mar, 2024, 15:57 · Saved in Deep Learning questions

2 PTS · OPEN · 1/2 PAGE

Consider applying five 3x3 convolutions with padding=1, stride=1, one after another (with some nonlinearity in between). What is the resulting feature size for an incoming feature of size 224x224x*D*? Assume that the hidden-layer dimensionality *D* stays constant.

The output dimension will b... 2 pts

The output dimension will be the same as the input $224 \times 224 \times D$

2 PTS · MULTIPLE CHOICE

What is the size of the receptive field of a 11x11 convolution (stride=4, padding=2) followed by maxpooling of size 3x3 (padding=2)?

- ○ 14
- ○ 12
- ◉ **19**
- ○ 21

2 PTS · MULTIPLE RESPONSE

ConvNets

- ☐ After training, it is always possible to turn a ResNet (with only common hidden dimensionality size) into a network without any "skip-connections" without any loss in accuracy -2.0
- ☑ **Depthwise convolutions allow for higher throughput and latency by splitting a convolution into the channel and spatial domains** 2.0
- ☐ One difference between faster R-CNN and R-CNN lies in the former model's faster computation of region proposals straight from the RGB image -2.0
- ☐ Starting from a detection model (e.g. faster R-CNN), semantic segmentation for *K* classes (e.g. Mask R-CNN) is usually obtained by learning a fully convolutional head which outputs *K+1* dimensional maps for each box -2.0

LoRA #31262531

7 pts · Last updated 19 Feb, 2024, 13:25 · Saved in Deep Learning questions

TEXT BLOCK

Consider a fully connected layer without bias, consisting of a single matrix multiply $W \in \mathbb{R}^{a,b}$, with

2 PTS · MULTIPLE RESPONSE

What is the number of parameters that need to be fine-tuned with LoRA (p) compared to the number of parameters that are require gradients when normally fine-tuning *W (P)?*

☑	$\boldsymbol{\frac{p}{P} = \frac{r\left(1+\frac{b}{a}\right)}{b}}$	1.0
☐	$\frac{p}{P} = \frac{r\left(1+\frac{a}{b}\right)}{b}$	-1.0
☐	$\frac{p}{P} = r^2$	-1.0
☑	$\boldsymbol{\frac{p}{P} = \frac{r\left(a^2-b^2\right)}{ab\left(a-b\right)}}$	1.0

4 PTS · OPEN · 2/5 PAGE

Imagine applying this adaptation at every fully connected layer of a pretrained transformer model. Describe the advantage of adapting a pretrained model to conduct a new task this way, compared to linear probing and full-finetuning. Name 1 relative advantage and 1 relative disadvantage for each.

(+) compared to linear prob...	1 pt
(+) compared to linear probing: it can adapt early and late layers	
(+) compared to full-finetu...	1 pt
(+) compared to full-finetuning is parameter-efficient	
(-) can overfit more than l...	1 pt
(-) can overfit more than linear-probing	
(-) less powerful than full...	1 pt
(-) less powerful than full-finetuning	

1 PT · OPEN · 1/5 PAGE

Consider another case of fine-tuning a pretrained model by inserting learnable 1x1 convolutions *c()*, such that $y'=c(\sigma(Wx))$, where σ is a non-linearity. Why can a LoRA-adapted model be faster at inference time than this adaptation method? You can assume applying the non-linearity does not cost any time/compute.

because the LoRA adaptation... 1 pt

because the LoRA adaptation W = W' + AB = W_new can simply be absorbed into a new W_new matrix, leading to no additional inference cost.

MQA #31266766

6 pts · Last updated 14 Feb, 2024, 13:06 · Saved in Deep Learning questions

6 PTS · OPEN · 1/2 PAGE

In a typical transformer block, we have H heads and for each head, we learn a key, query and value matrix. More recently, researchers started using a "Multi-query attention" (MQA) formulation, where only a single key and value matrix are used, no matter how many heads. While this performed not as well as the previous method, a novel approach called "Grouped Multi-query" attention (GQA) is now standard and used in models like Llama-2. The idea is similar to grouped convolutions, whereby not all dimensions are in communication with each other.

Say GQA-G refers to grouped-query with G groups. GQA-1, with a single group and therefore single key and value head, is equivalent to MQA, while GQA-H, with groups equal to number of heads, is equivalent to MHA. Give an expression for the number of multiplication and addition operations for a single transformer block with H heads and G groups and dimensionality D for GQA-H.

(note: consider only orders... 6 pts

- (note: consider only orders of magnitude: ie if it's "(D-1)" or "D" or "2D" does not matter here.
- (assume dimension after projections is D/H)
- (ignore masking, non-linearities, normalisation)
- compute Q projections :
  - mul: DDT
  - add: (D-1)DT +DDT --> DDT
  - [the matmul is 2DDT, the bias is (D-1)DT]
- compute K projections:
  - mul: DDT(G/H)
  - add: DDT(G/H)
- same for V projections:
  - mul: DDT(G/H)
  - add: DDT(G/H)
- compute dot product Q K^T for each head
  - [outer product: 2D/H*H (half add, half mul)]
  - [for each item in T*T: TT*2D]
  - mul: TTD
  - add: TTD
- softmaxes (without max-substract):
  - exp for every item: TT
  - sum of T items per row: (add): T(T-1)
  - dividing each TT item by a the sum (mul): TT
  - for each head H:
  - overall : 3TTH (add: HT(T-1), mul: TTH)
- Attention-based weighted avg of value projections (HxTxT and TxD/H):
  - for each head:
    - for each token
      - weight each token by their attention value: T*D/H (mul)
      - then: compute the average of those: T*D/H (add)
    - ie 2TTD/H

- ie 2TTD (half add, half mul)
- mul: TTD
- add: TTD (to be precise: H*((D/H)-1)*TT)

Right order of magnitude fo... 2 pts

Right order of magnitude for attention

Right order of magnitude fo... 2 pts

Right order of magnitude for the projections and proportionality of G/H

Right order of magnitude fo... 2 pts

Right order of magnitude for the reduce sum (attention multiplied with values)

blank space #31266344

0 pts · Last updated 14 Feb, 2024, 13:06

OPEN · 4/5 PAGE