



HAROKOPIO UNIVERSITY
DEPARTMENT OF INFORMATION & TELEMATICS

3rd Assignment Report: Machine Learning and Applications

Manousos Linardakis, it22064

Part 1: Using GPT-2:

1. Running the speech code (and changing it a bit to show the prompt in the notebook with print, and only run for 3 prompts), we can see the results for the following 3 prompts:

The first is the first sentence from the Wizard of Oz book:

Dorothy lived in the middle of the great Kansas prairies, with Uncle Henry, who was a farmer, and Aunt Em, who was the farmer's wife.

GPT-2's answer is:

Dorothy lived in the middle of the great Kansas prairies, with Uncle Henry, who was a farmer, and Aunt Em, who was the farmer's wife. After he had made some changes, the Indians began to grow and multiply. The great land was given to us as "the country that gave birth to the man's body," and the Indians had to be removed from the lands to be placed in other lands. At last the Indians were able to take possession of the land and the land was taken, and we, as a whole, have a right to a presentation of this same soil.

We do not want to have our presentations to be a part of it, because, as we know, we have been in that territory many and many years, and the Indians have a lot more land, and they must give us a presentation before we can do that again. We have had no right, in any part of this country ever to have a part whatever in our presentations. No right to make this land our presentation? we have always had a right to the presentation, for it is what we have been through

we notice that for this prompt, GPT produced a text that (besides the fact that it doesn't follow the Wizard of Oz story) doesn't have much "coherence". For example, the person of the text changes from "he" (in the first sentences) to "we" in the rest of the text, changing the form of the narrative. Also, he uses the word "presentation" several times without making any particular sense. However, the text is quite realistic and looks like the way a human would write (except for the face change). More generally, GPT-2 connected the original prompt to the theme of "land ownership" and specifically to the relocation of Indians, which is historically correct! The text continues to refer to this topic, and in the next paragraph. Therefore, the specific text is a fairly successful text production.

The second prompt tested is Shakespeare's famous line:

To be or not to be: that is the question.

The model gave the following result:

To be or not to be: that is the question.

But to me, this was not about this. This was about the fact -- it was something that was part of that whole debate. And now here's the thing,

if you're a feminist or a humanist and you've been doing this for a long time enough, you'll realize that, I think, the question that you have to ask when you say "feminism" or "humanism" is whether or not that's what you really think of as feminism -- and to me, feminism as a concept, as a concept as it relates to things that, I think, are more or less the same concept. And I have to say, this is one of those things that's a pretty simple and easy answer and that's the sort of question that makes me think in that way. But at the same time, that's just something that I've been doing and that's the sort of thing that I think is going to make feminists, and I will say this, if it's

Here (except that again he does not continue Shakespeare's words), he talks about the subject of "feminism" or "humanism". This time the quality got worse, as sometimes it repeats the same phrases (eg "and to me, feminism as a concept, as a concept") or writes sentences that do not make much sense and have little to do with rest of the text – i.e. they could have been removed (eg "This was about the fact -- it was something that was part of that whole debate."). Therefore, the final result does not look much like human "writing" and cannot be considered as successful text production.

Finally the third prompt tested is the first sentence in Star Wars movies:

In a galaxy far, far away,

The result is:

In a galaxy far, far away, in a galaxy far, away, in a galaxy far, the world is a place where you can go to the right place at the right time and place of the wrong time. And in all of that time, if I'm on the wrong side, what can I do to change it?

That's where we're going. You're going from a time where you're sitting in a room with a television and you're talking and you're thinking about things and you're looking for the right thing that has to do with the season and the people you want to find. The season is already done. We're at a point where we're able to get things done. So there's no question about that.

But let's look at this season. What kind of season is there in the universe where there is this idea that there's an entire universe that's in a perfect place in which you can have a family without having your spouse be on the wrong side of something?

Here again it seems as if no human wrote it as it repeats phrases again (as can be seen at the beginning of the text, which repeats "in a galaxy far away" in various places). Also, some sentences do not make sense (eg "to the right place in the right time and place of the wrong time") and are syntactically incorrect ("in the right time" should become "at the right time"). More generally, this text lacks much coherence and has syntactical errors making it an example of unsuccessful text production.

here we notice that it has repeated the word “vast” several times and continues to do so until max_tokens.

So the greedy search repeats the same words many times!

Search Beam Results: 1st prompt:

Dorothy lived in the middle of the great Kansas prairies, with Uncle Henry, who was a farmer, and Aunt Em, who was the farmer's wife.

In the spring of 1848, the family moved to a small town on the western side of the Missouri River, near the Missouri River.

In the spring of 1848, the family moved to a small town on the western side of the Missouri River, near the Missouri River.

In the spring of 1848, the family moved to a small town on the western side of the Missouri River, near the Missouri River.

In the spring of 1848, the family moved to a small town on the western side of the Missouri River, near the Missouri River.

In the spring of 1848, the family moved to a small town on the western side of the Missouri River, near the Missouri River.

In the spring of 1848, the family moved to a small town on the western side of the Missouri River, near the Missouri River.

In the spring of 1848, the family moved to a small town on the western side

Here again we see that he repeats phrases!

Let's see the results of the 2nd prompt:

In a galaxy far, far away, there is a galaxy far, far, far away.

In a galaxy far, far away, there is a galaxy far, far, far away.

In a galaxy far, far away, there is a galaxy far, far, far away.

In a galaxy far, far away, there is a galaxy far, far, far away.

In a galaxy far, far away, there is a galaxy far, far, far away.

In a galaxy far, far away, there is a galaxy far, far, far away.

this time there is again repetition of the same phrase, but not just one word (as was done before in the grid search).

Therefore the beam search is a little better than the grid search as (in the grid search) in the 2nd prompt we had a repetition of only one word (which doesn't look like it was written by a human). Now (with beam search) sentences are repeated which again doesn't offer anything new (to the text) but at least it's not just one word repeated over and over, thus making the text a bit more realistic.

Use Search Beam + no_repeat_ngram_size:

By using no_repeat_ngram_size, we tell the model not to repeat the same sequences in the generated output. The results are clearly better than the simple Beam Search, as it "solves" the problem observed before with the repetition of sentences. For the 1st prompt the result is:

Dorothy lived in the midst of the great Kansas prairies, with Uncle Henry, who was a farmer, and Aunt Em, who was the farmer's *wife*.

In the spring of 1848, the family moved to a small town on the outskirts of K

For the 2nd prompt:

In a galaxy far, far away, there is no such thing as a safe place to be.

It's a place where you can be anything you want. You can go anywhere anywhere. There's no place like home. It's the only place you'll ever be able to go, and that's why I'm here. I want you to know that I love you. And I know you're going to love me, too.

the texts produced above look more like they were written by a human, but they are not very large in size, due to the terms being "banned" by no_repeat_ngram_size.

Using top-k:

The prompts, results, and commentary for this strategy can be found at [first question](#) (as this method was used for the 1st answer). The conclusion of this strategy is that the model most often does not produce coherent and "realistic" texts. However, it gives better results than beam & greedy search as the generated texts are large in size and do not repeat the same phrases/words often.

Using top-p:

1st prompt:

Dorothy lived in the middle of the great Kansas prairies, with Uncle Henry, who was a farmer, and Aunt Em, who was the farmer's wife. They married in 1479, to whom they have continued to live ever since. The two children were: George a son of Samuel, who died of heart disease in 1477; and Alice a daughter of Samuel, who died of disease in 1484. In consequence, they live well, have a common life, have a good home, and live in peace.

The eldest daughter, a daughter of Edward, died in 1492; the son is buried in the old Church of England in Leicester, which she lived in until the mid-1630s; his widow, Esther, died in 1650.

In 1744, Charles Dandridge became a member of the Continental Parliament. The Continental Parliament was elected in 1745. It was formed with the support of a majority in Congress? the president was John Adams, who was appointed by George Washington, a member of the Continental Parliament in 1776, and John Jay, a member of Congress until 1801. When Jay

we notice that the text produced uses a variety of words making the text look like it was written by a human. Also, the text stays close to one of the "themes" of the prompt, which is the family (since in the prompt we describe how Dorothy lives with her uncle and aunt). Of course, if we read the text we see that it sometimes does not make sense (eg it says that the two children George and Alice died of a heart problem and then says that "as a result" they live well).

2nd prompt:

In a galaxy far, far away, one of the brightest galaxies that ever exists, lies hidden beneath the sky.

In a galaxy far, far away, one of the brightest galaxies that ever exists, lies hidden beneath the sky. The universe looks really, really dark. When NASA's Keck Observatory and the SETI program found the galaxy, it became a mystery and was left out of the search. The Keck observatory, which includes Kota and Chico, in California, also turned up a giant, glowing gas giant called a dwarf galaxy known as M1. But a team of scientists from UCLA, New Mexico and California State University has found something even stranger: They can observe it from inside a star.

The Keck Telescope, now under construction at the European Southern Observatory in Chile, is the largest telescope in the world to study the Keck Cluster. The telescope has a broad field of view of at least three million light-years, in contrast to the thousands or even millions that are visible over

in this generated text we again see that top-p is closer to the subject of the original prompt, specifically talking about space. In addition, the text this time is more coherent and "correct" than in the previous prompt and therefore resembles human handwriting more.

Therefore, top-p is a little better than top-k as it stays on the same "topic" with the prompt we have given it while at the same time making quite realistic texts.

Using top-k & top-p:

1st prompt:

Dorothy lived in the middle of the great Kansas prairies, with Uncle Henry, who was a farmer, and Aunt Em, who was the farmer's wife. On the same day, she and her father took a boat to the town, where they were found by a man from the same company. The young woman and the young man who accompanied them were killed in the collision with a herd of cattle, and they were buried at the cemetery. They were buried in a large grave at Woburn, a little north of Wichita, and the remains of Mrs. Ortega are said to be there. They were buried in a large mound on a hillside near the town. There was an old brick church built in 1837. The church of

Mrs. Ortega was found in the vicinity of the burial place of her mother. Another church in which the family is said to have lived is the Methodist Church, at Kuehlman, Kansas City. The oldest surviving church is located in the churchyard of the church where they lived. It is said that they were baptized at the stake and that they were buried near the spot where they died

we notice that the generated text is more realistic than before (where we only used top-p). Again, it stays close to the original topic of the prompt, but without there being a point that doesn't make much sense or goes too far off topic.

2nd prompt:

In a galaxy far, far away, and from a galaxy far, far away, is a strange, strange world.

The galaxy is a vast, unspoiled, multiverse, filled with many galaxies, and many planets, many universes, many universes, many universes, and many universes. Every universe is inhabited by several intelligent beings, and even many of them have their own unique characteristics, but the main star in the universe is the largest galaxy in the entire known universe and is the most powerful. And this is not all.

This planet is the "Gigantic Moon" and all the stars that inhabit it are of this world. Every star is one massive, massive star. Every star is made up of billions of galaxies. Each galaxy is filled with trillions of stars that are all the same size. Each galaxy is home to millions of galaxies. Every galaxy is a huge star in which there are many galaxies and billions of galaxies and millions of planets, billions of planets, and hundreds of planets. So

Likewise here we notice that he again stays on the topic of the prompt and talks about planets. So the combination of top-k & top-p gave the best results so far (where better means closer to human handwriting). In Part 2 I use this combination to test the requested model.

Part 2: Model fitting

Observation : I trained the model for a total of 5 seasons as it says in the pronunciation code. It's worth noting that I changed the original code a bit so that if the model is already saved (in the save directory) it continues to train it (by loading it from the directory first). So, first I ran the colab for 1 season, then I saved the model and at another time I loaded it from the catalog and ran it another 2 + 2 seasons. I saved the final model in the model_save directory.

1. The prompt tested is Shakespeare's phrase:

To be or not to be: that is the question.

The model gave the following result:

To be or not to be: that is the question. What good is a question of precedence if you start down the south passage and speak of the Rebellion? I know the procedure of the law. I feel confident we can overcome it
I must speak with the Jei Council immediately, Your Honor. The situation has become more complicated.

Ani, come on.

Da queen's a bein grossly nice, mesa tink. Pitty hot.

the Republic is not what it once was. The Senate is full of greed, squabbling delegates who are only looking out for themselves and their home systems. There is no interest in the common good no civility, only politics its disgusting. I must be frank, Your Majesty, there is little chance the Senate will act on the invasion.

Chancellor Valorum seems to think there is hope.

If I may say so, Your Majesty, the Chancellor has little real power he is mired down by baseless accusations of corruption. A manufactured scandal surrounds him

We notice that using this prompt, the model continued the text based on the star wars subtitles. Also, the word "question" in the prompt caused the next sentence of the generated text to contain the word "question". So we notice that the model tries to produce text that is as close as possible to the prompt (based on dialogue patterns from star wars movies, as it learns them).

2. Prompts used:

"the dark side", "the force"

Using the prompt "the dark side" 2 times we notice that the output – the texts produced are the same! Specifically, both times the specific prompt was used, the following text was produced:

the dark side of the Force is a pathway to many abilities some consider to be unnatural.

What happened to him?

He became so powerful... the only thing he was afraid of was losing his power, which eventually, of course, he did. Unfortunately, he taught his apprentice everything he knew, then his apprentice killed him in his sleep. (smiles) Plagueis never saw it coming. It's ironic he could save

others from death, but not himself. Is it possible to learn this power? Not from a Jedi.

(holo) Palpatine thinks General Grievous is on Utapau. We have had no reports of this from our agents.

(holo) How could the Chancellor have come by this information and we do you know anything about it? We have had contact with Baron Papanoida and he said no one was there.

The text following "the dark side" is exactly the same as the one in the star_wars.txt subtitles.

Using the prompt "the force" 2 times we got different outputs. The first text produced is:

the force is with us, my Master.

Welcome home, Lord Tyranus. You have done well.

I bring you good news, my Lord. The war has begun.

Excellent. (smiling) Everything is going as planned. Where is your apprentice?

On his way back to Naboo. He is escorting Senator Amidala home.

(continuing) I must admit without the clones, it would not have been a victory.

Victory? Victory, you say?

This "paragraph" is also the same from the subtitles!

The next text produced is:

the force grows dark, Anakin, and we are all affected by it. Be wary of your feelings.

Anakin, this afternoon the Senate is going to call on me to take direct control of the Jedi Council.

The Jedi will no longer report to the Senate?

They will report to me... personally. The Senate is too unfocused to conduct a war.

This will bring a quick end to things.

I agree, but the Jedi Council may not see it that way.

There are times when we must all endure adjustments to the constitution in the name of security.

With all due respect, sir, the Council is in no mood for more constitutional amendments.

Thank you, my friend, but in this case I have no choice... this war must be won.

which is also the same text as the subtitles.

In conclusion, the model produces text that is very "faithful" to the subtitles of the star wars movies. Also, as can be seen from the "the dark side" prompt, there may be a case that generates the same text (for the same prompt).

3. Running the code, it outputs:

```
The GPT-2 model has 148 different named parameters.

==== Embedding Layer ====

transformer.wte.weight                (50259, 768)
transformer.wpe.weight                (1024, 768)

==== First Transformer ====

transformer.h.0.ln_1.weight           (768,)
transformer.h.0.ln_1.bias             (768,)
transformer.h.0.attn.c_attn.weight    (768, 2304)
transformer.h.0.attn.c_attn.bias      (2304,)
transformer.h.0.attn.c_proj.weight    (768, 768)
transformer.h.0.attn.c_proj.bias      (768,)
transformer.h.0.ln_2.weight           (768,)
transformer.h.0.ln_2.bias             (768,)
transformer.h.0.mlp.c_fc.weight        (768, 3072)
transformer.h.0.mlp.c_fc.bias          (3072,)
transformer.h.0.mlp.c_proj.weight      (3072, 768)
transformer.h.0.mlp.c_proj.bias        (768,)

==== Output Layer ====

transformer.ln_f.weight               (768,)
transformer.ln_f.bias                 (768,)
```

First, we see that GPT-2 has 148 different “named” parameters. Looking at these parameters, we can conclude the following about their number:

For the Embedding Layer (transforms the text into numeric vectors so the model can process them):

- `transformer.wte.weight` = weight matrix for token embeddings. It shows that there are 50,259 distinct embeddings and each (embedding) is represented by a vector of **768** dimensions.
- `transformer.wpe.weight` = weight matrix for positional embeddings. It shows that there are 1024 positions, each represented by a 768-dimensional vector.

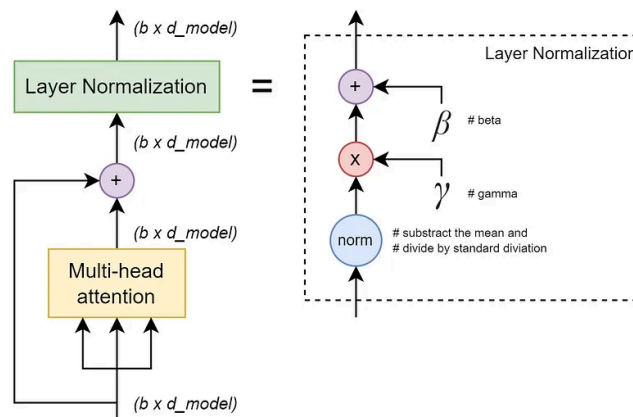
`dmodel` is a parameter that refers to the number of dimensions of the model's embeddings. Therefore, `dmodel` = 768.

For the First Transformer:

- `ln_1`: (generally `ln` = layer normalization) there are 768 (=dmodel) parameters for both weight and bias.

Explanation:

The layer-normalization is described by the following scheme:

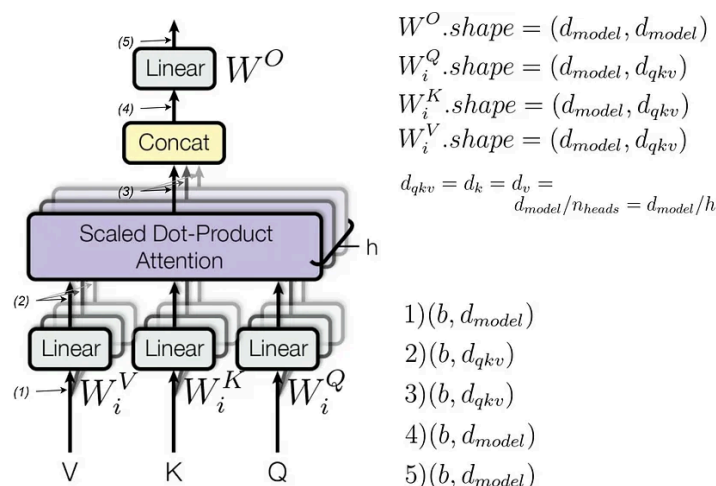


In the GPT-2 model, layer-normalization involves computing the mean and standard deviation of the input dimensions ($d_{\text{model}}=768$). These results are then used to normalize the inputs before applying the scaling (c) and shifting (b) parameters. These parameters allow the model to adjust the normalized outputs, providing flexibility in the representation. Therefore, the bias (b) and weight (c) vectors have the same dimension as the input embeddings (768) to ensure consistency in the normalization process.

- `attn.c_attn`: Its parameters attention mechanism. The weight matrix has dimensions (768, 2304) and the bias has size 2304.
- `attn.c_proj`: The "output" of the attention mechanism. The weight matrix acquired dimensions (768, 768) and the bias has a size of 768.

Explanation:

The output has these parameters – dimensions as multiple head attention (MHA) schematically looks like this:



The outlet has the dimensions of W_{THE} which is $(d_{\text{model}}, d_{\text{model}})=(768, 768)$.

Also, the MHA parameters – described in `attn.c_attn` – have dimensions $(d_{\text{model}}, 3 \cdot d_{\text{model}}) = (768, 3 \cdot 768) = (768, 2304)$. The multiplication by 3 is done because we see that there are 3 "inputs" at the bottom (of the figure).

- \ln_2 : parameters of the next normalization layer, like \ln_1 , with 768 weight and bias parameters (similar explanation to [ln_1](#)).
- mlp.c_fc : The feed-forward parameters. The weight matrix has size (768, 3072) and the bias has 3072, where $3072 = 4 \times d_{\text{model}}$.
- mlp.c_proj : The feed-forward output. The weight matrix has size (3072, 768) and the bias has 768.

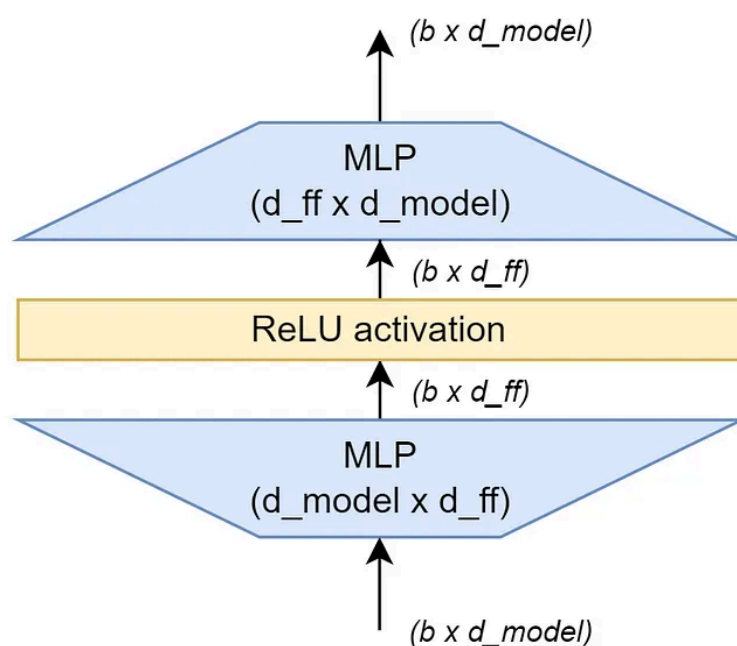
Feed-forward explanation:

The formula used in the paper ("Attention is all you need") for feed-forward is:

$$\text{FFN}(x) = \max(0, xW_1 + b_1)W_2 + b_2 \quad (2)$$

While the linear transformations are the same across different positions, they use different parameters from layer to layer. Another way of describing this is as two convolutions with kernel size 1. The dimensionality of input and output is $d_{\text{model}} = 512$, and the inner-layer has dimensionality $d_{\text{ff}} = 2048$.

Schematically this is written:



where $d_{\text{ff}} = 4 \times d_{\text{model}}$. We can see that at first the input comes in – as described in c_fc – with dimensions $(d_{\text{model}}, d_{\text{ff}}) = (768, 3072)$ and then goes through ReLU. The output is then – as marked in c_proj – with dimensions $(d_{\text{ff}}, d_{\text{model}}) = (3072, 768)$.

For the output layer: shows the parameters of the final normalization layer (\ln_f), which are 768 parameters for weight and bias (as in the previous normalization layers – the explanation of how these parameters were obtained is similar to that of [ln_1](#)).

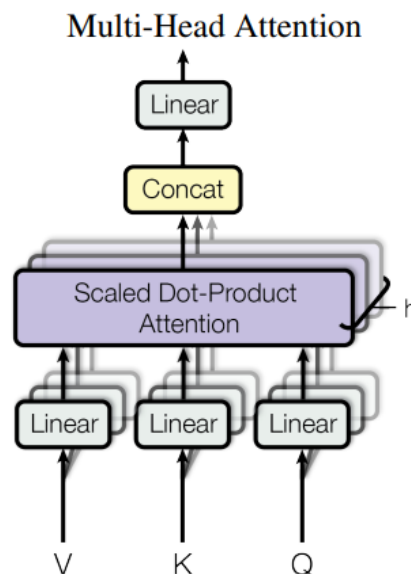
The figures used to answer this question can be found in the following sources [\[1\]](#), [\[2\]](#), [\[3\]](#).

Bonus question

By carefully studying the MultiHeadAttention class, we conclude that there is no error in the code.

To justify this proposition, we first need to study the MultiHead Attention (MHA) of the paper "[Attention is all you need](#)".

According to the paper, multi-head attention is described by the following scheme:



A common shape of V, K, Q is (seq_length, d_model), where seq_length represents the length of each sequence and d_model the dimension of the feature space. In the paper we also have d_k for which $d_k = d_{\text{model}} / \text{num_heads}$. Here is the corresponding excerpt:

In this work we employ $h = 8$ parallel attention layers, or heads. For each of these we use $d_k = d_v = d_{\text{model}}/h = 64$. Due to the reduced dimension of each head, the total computational cost is similar to that of single-head attention with full dimensionality.

For the implementation of the MHA, they found it useful to project h times (where h = number of heads) with different linear mode the Q, K, V in dimensions $d_k, d_k, d_v (=d_k)$ respectively. Corresponding reference from the paper:

3.2.2 Multi-Head Attention

Instead of performing a single attention function with d_{model} -dimensional keys, values and queries, we found it beneficial to linearly project the queries, keys and values h times with different, learned linear projections to d_k, d_k and d_v dimensions, respectively. On each of these projected versions of queries, keys and values we then perform the attention function in parallel, yielding d_v -dimensional

This can also be seen from the lower part of the MHA figure (before the scaled dot-product attention), where Q, K, V enter (and are consequently projected) in h different linear planes.

with the paper) are:

$$\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$

The
Q
W
Pi
(

Where the projections are parameter matrices $W_i^Q \in \mathbb{R}^{d_{\text{model}} \times d_k}$, $W_i^K \in \mathbb{R}^{d_{\text{model}} \times d_k}$, $W_i^V \in \mathbb{R}^{d_{\text{model}} \times d_v}$ and $W^O \in \mathbb{R}^{hd_v \times d_{\text{model}}}$.

It is noteworthy that the attention mechanism applies in parallel to each of these projected versions of Q, K, V, as mentioned here:

3.2.2 Multi-Head Attention

Instead of performing a single attention function with d_{model} -dimensional keys, values and queries, we found it beneficial to linearly project the queries, keys and values h times with different, learned linear projections to d_k , d_k and d_v dimensions, respectively. On each of these projected versions of queries, keys and values we then perform the attention function in parallel, yielding d_v -dimensional output values. These are concatenated and once again projected, resulting in the final values, as depicted in Figure 2.

After all heads are calculated, we concat them and then multiply them by the final linear W^O which has shape $(h \times d_v, d_{\text{model}}) = (d_{\text{model}}, d_{\text{model}})$. Followed by

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \dots, \text{head}_h)W^O$$

$$\begin{aligned} \text{MultiHead}(Q, K, V) &= \text{Concat}(\text{head}_1, \dots, \text{head}_h)W^O \\ \text{where head}_i &= \text{Attention}(QW_i^Q, KW_i^K, VW_i^V) \end{aligned}$$

Where the projections are parameter matrices $W_i^Q \in \mathbb{R}^{d_{\text{model}} \times d_k}$, $W_i^K \in \mathbb{R}^{d_{\text{model}} \times d_k}$, $W_i^V \in \mathbb{R}^{d_{\text{model}} \times d_v}$ and $W^O \in \mathbb{R}^{hd_v \times d_{\text{model}}}$.

So let's compare this process with his code [datacamp](#). First, the entire code can be described by the following figure:

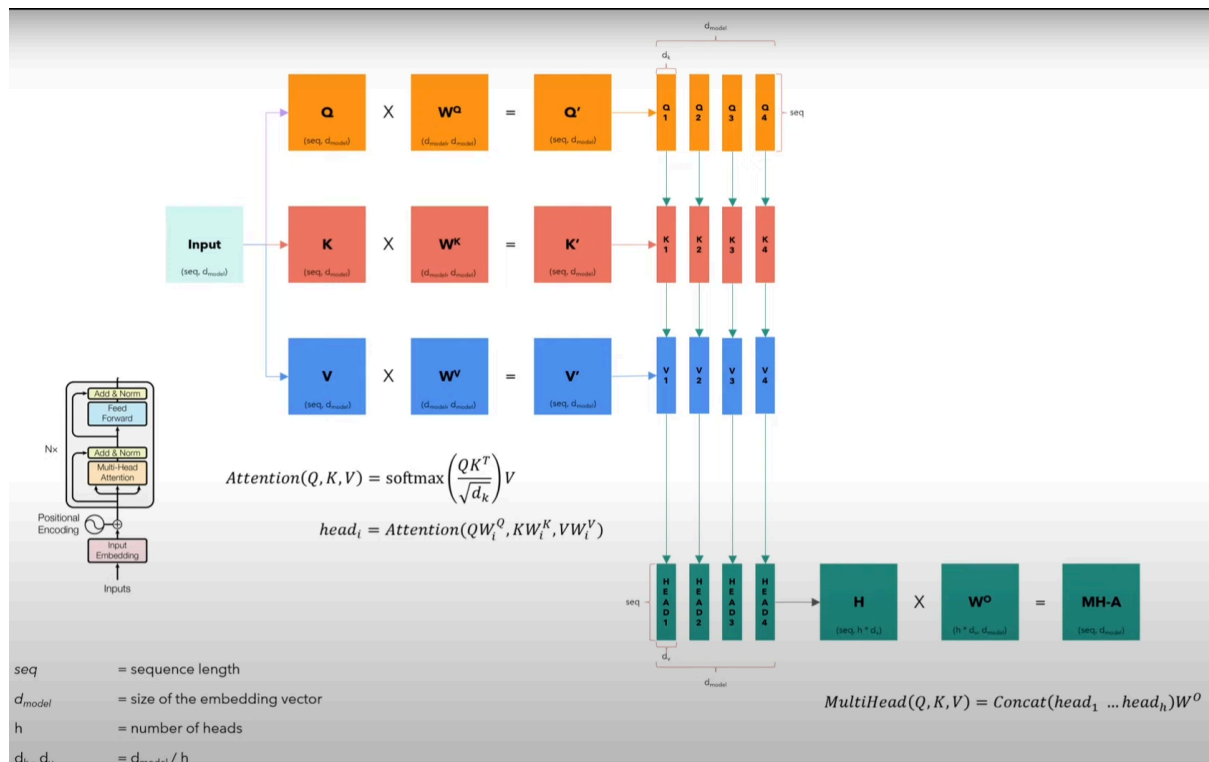


Figure [1] ([image source – youtube](#))

We see that we have inputs Q, K, V with shapes (seq_length, d_model) – as we pointed out when we analyzed the paper. Then we multiply Q, K, V by the linear matrices W_Q, W_K, W_V which have dimensions (d_model, d_model) . This results in the production of matrices Q', K', V' of dimensions (seq_length, d_model) , since $(seq_length, d_model) \times (d_model, d_model) \Rightarrow (seq_length, d_model)$. We split Q', K', V' (with `split_heads` in the code) into different linear sets with dimensions (seq_length, d_k) as many times as the number of heads (as we mentioned in the analysis of the paper). Thus, we create the attention inputs

–

QW_Q, KW_K, VW_V of the paper (which as we had previously calculated have shape (seq_length, d_k) after `split_heads`). Then using as inputs in `Attention` the QW_Q, KW_K, VW_V , we calculate the attention on each head alongside as it is done on paper. The resulting attention results (the i heads that are with **Green** color in the figure) we concat them and then multiply by W_O to find the MHA.

Therefore, the scheme (and consequently the code) correctly follows the "general" procedure of the MHA of the paper as we analyzed it.

More specifically, in the constructor (of the `datacamp` code):

```
class MultiHeadAttention(nn.Module):
    def __init__(self, d_model, num_heads):
        great(MultiHeadAttention, self).__init__()
        # Initialize dimensions
        self.d_model = d_model # Model's dimensions
        self.num_heads = num_heads # Number of attention heads
        self.d_k = d_model // num_heads # Dimension of each head's key,
```


query, and value

```
# Linear layers for transforming inputs
self.W_q = nn.Linear(d_model, d_model) # Query transformation
self.W_k = nn.Linear(d_model, d_model) # Key transformation
self.W_v = nn.Linear(d_model, d_model) # Value transformation
self.W_o = nn.Linear(d_model, d_model) # Output transformation
```

we notice that the d_{model} and d_k dimensions are calculated, where $d_k = d_{\text{model}} / \text{num_heads}$, as done in the paper (the `//` operation ensures that the result will always be an integer, which is what we want for the d_k dimension calculation). Also, as we saw from the figure, we initialize W_q , W_k , W_v and W_o with dimensions $(d_{\text{model}}, d_{\text{model}})$.

Let's now look at the code forward:

```
def forward(self, Q, K, V, mask=None):
    # Apply linear transformations and split heads
    Q = self.split_heads(self.W_q(Q))
    K = self.split_heads(self.W_k(K))
    V = self.split_heads(self.W_v(V))
    # Perform scaled dot-product attention
    attn_output = self.scaled_dot_product_attention(Q, K, V, mask)
    # Combine heads and apply output transformation
    output = self.W_o(self.combine_heads(attn_output))
    return output
```

from here we see that it initially "multiplies" Q , K , V with the linear matrices W_q , W_k and W_v respectively (as in the figure). Thus, Q' , K' , V' of the figure arise [1]. It then uses `split_heads` to produce the attention inputs QW_Q , KW_K , VW_V .

Here is the `split_heads` code:

```
def split_heads(self, x):
    # Reshape the input to have num_heads for multi-head attention
    batch_size, seq_length, d_model = x.size()
    return x.view(batch_size, seq_length, self.num_heads, self
    .d_k).transpose(1,2)
```

`split_heads` returns a tensor with shape $(\text{batch_size}, \text{num_heads}, \text{seq_length}, d_k)$, something which ultimately correctly represents the QW_Q , KW_K , VW_V that we want to put then to Attention, as each input has shape $(\text{seq_length}, d_k)$ as we calculated previously, and is also "split" into `num_heads`. That is, the `split_heads` finally creates the proper QW_Q , KW_K , VW_V which are then entered as input to the attention for the attention production of each head.

Then the head attentions are combined (concat) into `combine_heads`:

```
def combine_heads(self, x):
    # Combine the multiple heads back to original shape
```

```
batch_size,_, seq_length, d_k = x.size()
return x.transpose(1,2).contiguous().view(batch_size, seq_length, self.d_model)
```

and they are then rotated to d_model dimension so that we pass them through the final Linear W_o.

As for the scale_dot_product attention of the code, it is correctly implemented and follows the guidelines of the paper. Also, the attention calculations are done in parallel (as mentioned in the paper) as the input shape of "Q, K, V" in the scale_dot_product function is of the form [batch_size, num_heads, seq_length, d_k] i.e. we have initialized the num_heads which ensures the "parallelism" of calculations.

So the code is properly implemented. Here are other sources that have a very similar implementation:

- <https://github.com/hkproj/pytorch-transformer/blob/main/model.py#L83>
- <https://nlp.seas.harvard.edu/annotated-transformer/>
- https://d2l.ai/chapter_attention-mechanisms-and-transformers/multihead-attention.html

A "bug" noticed "outside" of MHA is in the Transformer class forward masking is used to produce the output in the encoder and again later when the encoder output is used in the decoder. This is not consistent with the "standard" Transformer model as the encoder (in the standard model) can monitor all input positions, to understand the context. For this, I removed the encoder input/output masking from the code. The new forward looks like this:

```
def forward(self, src, tgt):
    src_mask, tgt_mask = self.generate_mask(src, tgt)
    src_embedded =
self.dropout(self.positional_encoding(self.encoder_embedding(src)))
    tgt_embedded =
self.dropout(self.positional_encoding(self.decoder_embedding(tgt)))

    enc_output = src_embedded
    for enc_layer in self.encoder_layers:
        # Does not apply mask
        enc_output = enc_layer(enc_output, None)

    dec_output = tgt_embedded
    for dec_layer in self.decoder_layers:
        dec_output = dec_layer(dec_output, enc_output, None, tgt_mask)

    output = self.fc(dec_output)
    return output
```

Typical Transformer Architecture:

