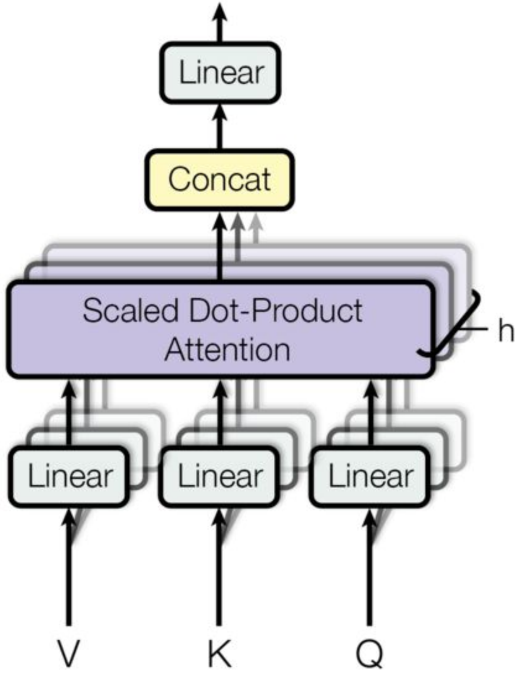
(Better) understanding the scaled dot-product multi-headed self-attention

Understanding multi-headed self-attention

- Might seem intimidating, but the individual parts are rather simple
  - The easiest way to understand **multi-headed** self-attention is to understand a *single-headed* one

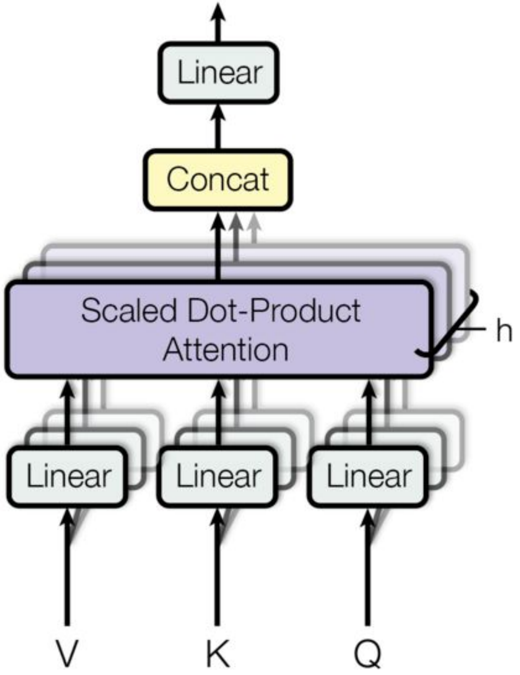


Understanding multi-headed self-attention

- Might seem intimidating, but the individual parts are rather simple
  - The easiest way to understand **multi-headed** self-attention is to understand a *single-headed* one

In matrix notation (for all pairs of token positions):

$$\mathrm{Attention}(Q, K, V) = \mathrm{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

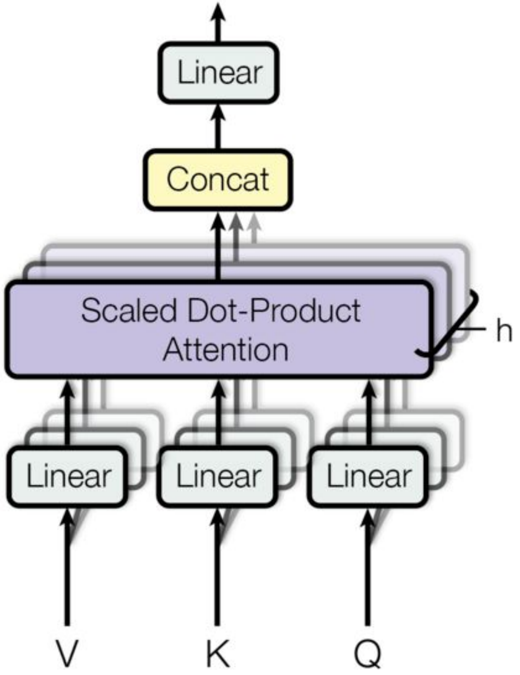


Understanding multi-headed self-attention

- Might seem intimidating, but the individual parts are rather simple
  - The easiest way to understand **multi-headed** self-attention is to understand a *single-headed* one

In matrix notation (for all pairs of token positions):

$$\mathrm{Attention}(Q, K, V) = \mathrm{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$



For a single pair of token positions

$$a_{ij} = \mathrm{softmax}(\frac{\mathbf{q}_i \mathbf{k}_j^\top}{\sqrt{d_k}}) = \frac{\exp(\mathbf{q}_i \mathbf{k}_j^\top)}{\sqrt{d_k} \sum_{r \in S_i} \exp(\mathbf{q}_i \mathbf{k}_r^\top)}$$

A *personal* view on Self-Attention (SA)

A powerful blending machine on steroids

A *personal* view on Self-Attention (SA)

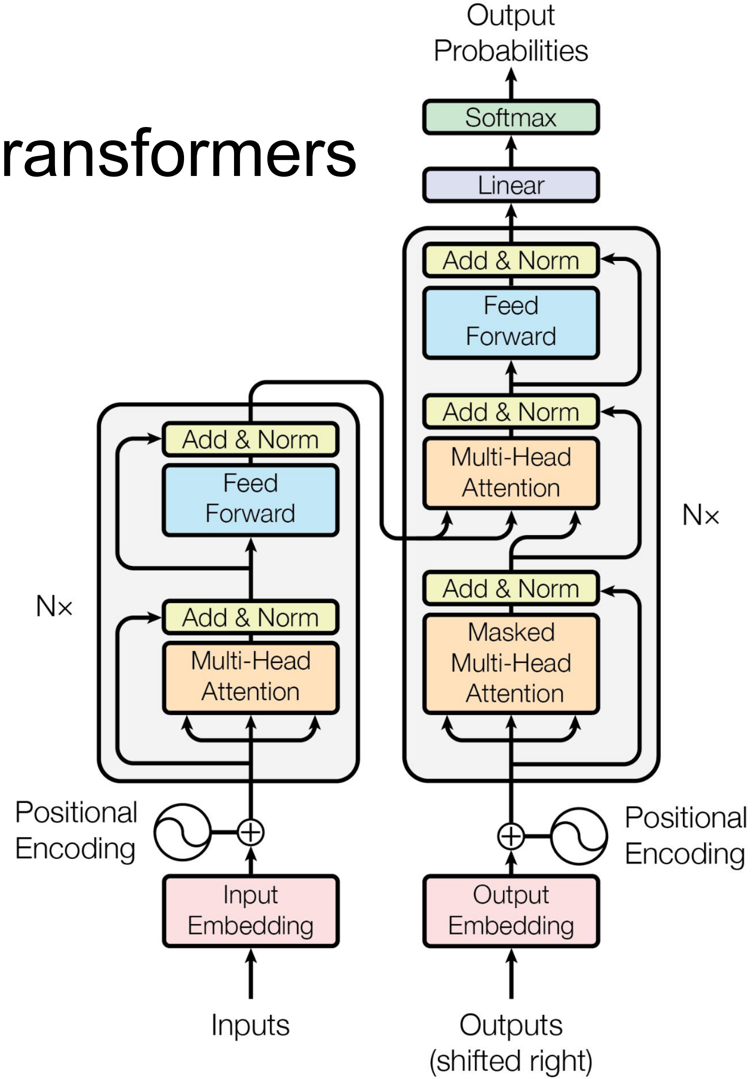
A powerful blending machine on steroids (gradients)

What are all the query / key / value abstractions, and why are they needed?

Why is multi-headed attention needed?

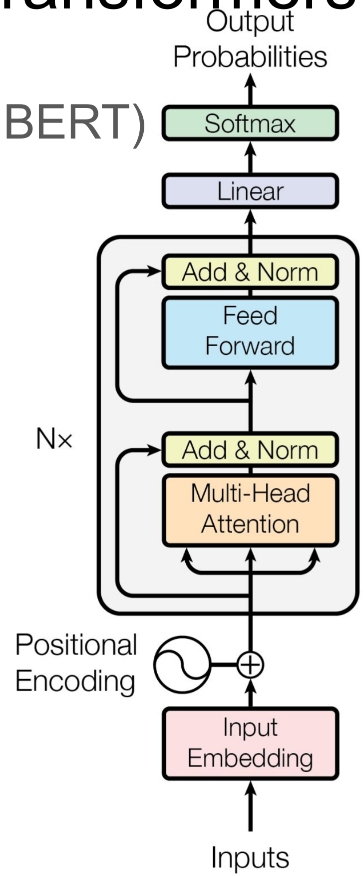
(Are they needed eventually?)

Tips for easier understanding SA in transformers



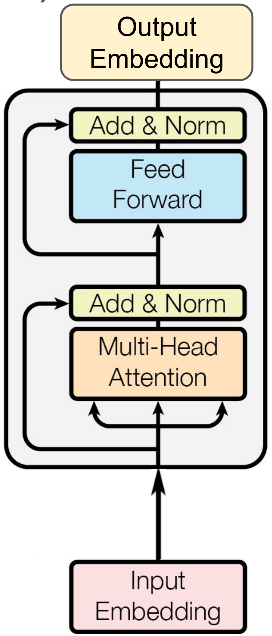
Tips for easier understanding SA in transformers

- Try forgetting about the decoding (just think of BERT)



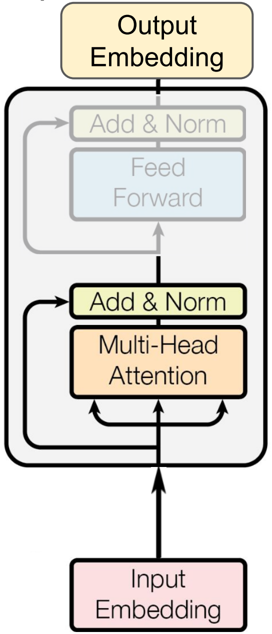
Tips for easier understanding SA in transformers

- Try forgetting about the decoding (just think of BERT)
- Try forgetting that transformer blocks stacked up
    - Essentially each block receives and omits embeddings



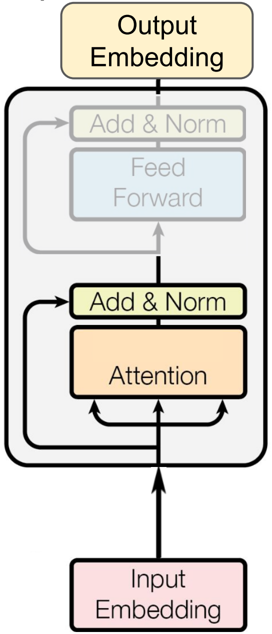
Tips for easier understanding SA in transformers

- Try forgetting about the decoding (just think of BERT)
- Try forgetting that transformer blocks stacked up
  - Essentially each block receives and omits embeddings
- Try forgetting (temporarily) the FFNN component



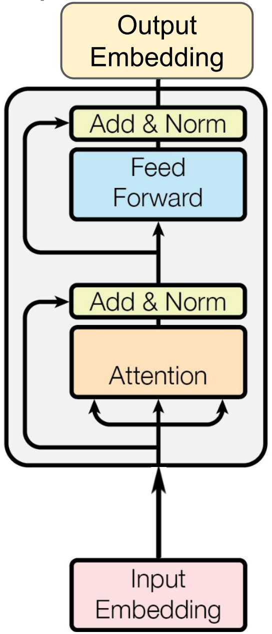
Tips for easier understanding SA in transformers

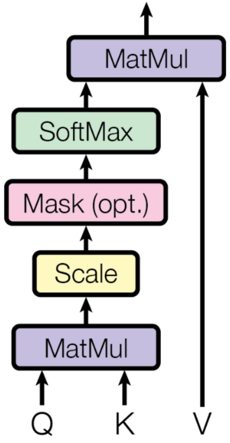
- Try forgetting about the decoding (just think of BERT)
- Try forgetting that transformer blocks stacked up
  - Essentially each block receives and omits embeddings
- Try forgetting (temporarily) the FFNN component
  - At the same time we can forget about multi-headedness



Tips for easier understanding SA in transformers

- Try forgetting about the decoding (just think of BERT)
- Try forgetting that transformer blocks stacked up
  - Essentially each block receives and omits embeddings
- Try forgetting (temporarily) the FFNN component
  - At the same time we can forget about multi-headedness

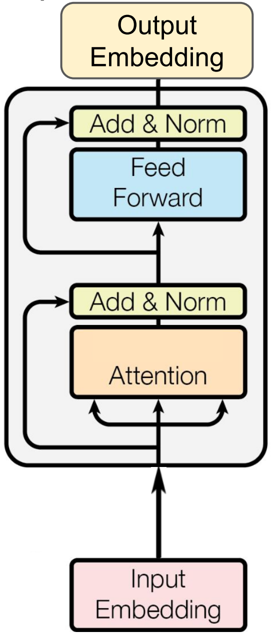


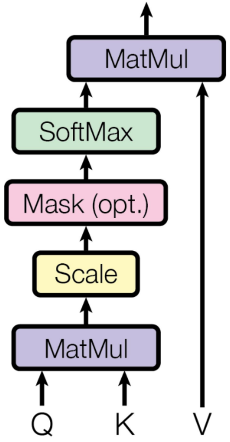


$$\mathrm{Attention}(Q, K, V) = \mathrm{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$

Tips for easier understanding SA in transformers

- Try forgetting about the decoding (just think of BERT)
- Try forgetting that transformer blocks stacked up
  - Essentially each block receives and omits embeddings
- Try forgetting (temporarily) the FFNN component
  - At the same time we can forget about multi-headedness



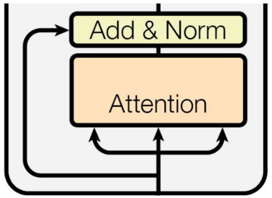


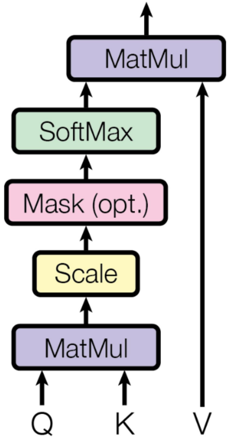
- If X contains the input embeddings then we have:
  - $Q = X W_Q$
  - $K = X W_K$
  - $V = X W_V$

$$\mathrm{Attention}(Q, K, V) = \mathrm{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

What is the role of the value transformation?

- At each layer, it allows for new semantic aspects to enter the representations
  - It adds on top of the already calculated one via the residual connection

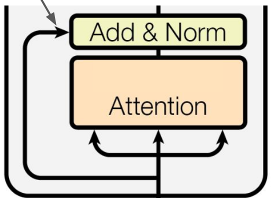


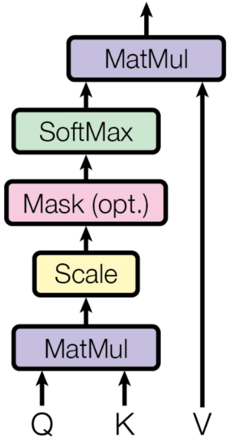


$$\mathrm{Attention}(Q,K,V) = \mathrm{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$

What is the role of the value transformation?

- At each layer, it allows for new semantic aspects to enter the representations
  - It adds on top of the already calculated one via the residual connection

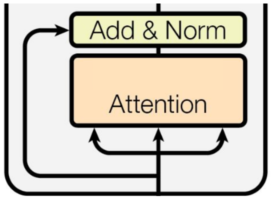


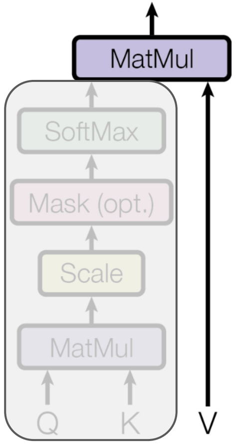


$$\text{Attention}(Q,K,V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

What is the role of the value transformation?

- At each layer, it allows for new semantic aspects to enter the representations
    - It adds on top of the already calculated one via the residual connection
- If there was no SA, this would cause the tokens to be handled independently, i.e. embedding $\mathbf{x}_i$ would simply become $\mathbf{x}_i+\mathbf{x}_i W_V$

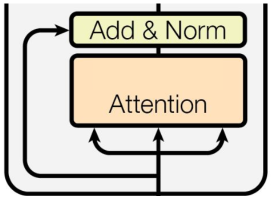


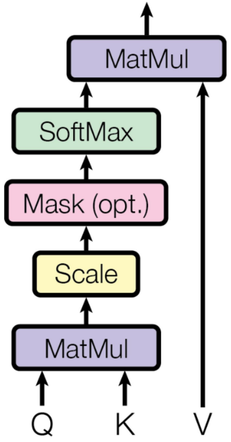


$$\mathrm{Attention}(Q, K, V) = \mathrm{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

What is the role of the value transformation?

- At each layer, it allows for new semantic aspects to enter the representations
  - It adds on top of the already calculated one via the residual connection
- If there was no SA, this would cause the tokens to be handled independently, i.e. embedding $\mathbf{x}_\mathbf{i}$ would simply become $\mathbf{x}_\mathbf{i}+\mathbf{x}_\mathbf{i}W_V$
- Representations are made context-aware by SA, i.e. instead of $\mathbf{x}_\mathbf{i}+\mathbf{x}_\mathbf{i}V$, we have $\mathbf{x}_\mathbf{i} + \Sigma_j\, a_{ij}(\mathbf{x}_\mathbf{j}W_V)$



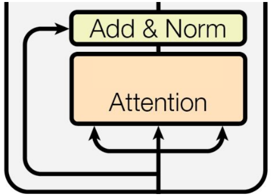


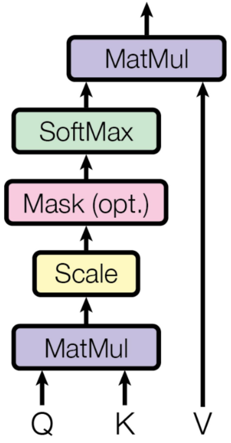
$$\mathrm{Attention}(Q,K,V) = \mathrm{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$

What is the role of the value transformation?

- At each layer, it allows for new semantic aspects to enter the representations
  - It adds on top of the already calculated one via the residual connection
- If there was no SA, this would cause the tokens to be handled independently, i.e. embedding $\mathbf{x}_\mathbf{i}$ would simply become $\mathbf{x}_\mathbf{i}+\mathbf{x}_\mathbf{i}W_V$
- Representations are made context-aware by SA, i.e. instead of $\mathbf{x}_\mathbf{i}+\mathbf{x}_\mathbf{i}V$, we have $\mathbf{x}_\mathbf{i} + \Sigma_j\, a_{ij}(\mathbf{x}_\mathbf{j}W_V)$, with

$$a_{ij} = \mathrm{softmax}(\frac{\mathbf{q}_i{\mathbf{k}_j}^\top}{\sqrt{d_k}}) = \frac{\exp(\mathbf{q}_i{\mathbf{k}_j}^\top)}{\sqrt{d_k}\sum_{r \in S_i}\exp(\mathbf{q}_i{\mathbf{k}_r}^\top)}$$

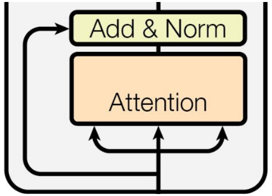


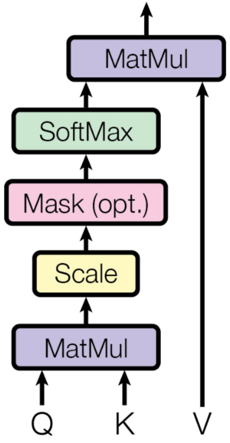


$$\mathrm{Attention}(Q, K, V) = \mathrm{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$

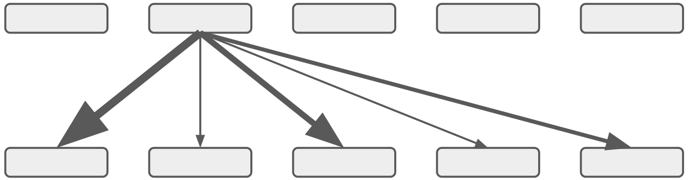
What is the role of the value transformation?

- At each layer, it allows for new semantic aspects to enter the representations
  - It adds on top of the already calculated one via the residual connection
- If there was no SA, this would cause the tokens to be handled independently, i.e. embedding $\mathbf{x}_i$ would simply become $\mathbf{x}_i+\mathbf{x}_i W_V$
- Representations are made context-aware by SA, i.e. instead of $\mathbf{x}_i+\mathbf{x}_i V$, we have $\mathbf{x}_i + \Sigma_j\, a_{ij}(\mathbf{x}_j W_V)$, with





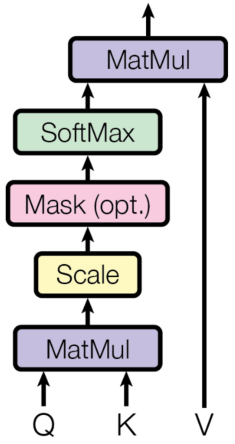
$$a_{ij} = \text{softmax}\left(\frac{\mathbf{q}_i \mathbf{k}_j^{\top}}{\sqrt{d_k}}\right) = \frac{\exp(\mathbf{q}_i \mathbf{k}_j^{\top})}{\sqrt{d_k}\sum_{r\in S_i}\exp(\mathbf{q}_i \mathbf{k}_r^{\top})}$$



$$\text{Attention}(Q,K,V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

What are the key/query matrices for?

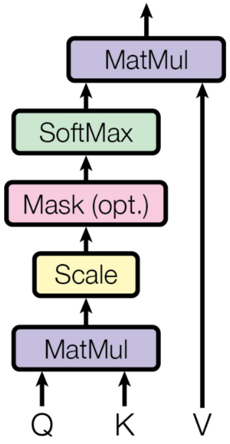
- They are responsible for the calculation of tha a_{ij} attention scores



$$\mathrm{Attention}(Q, K, V) = \mathrm{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$

What are the key/query matrices for?

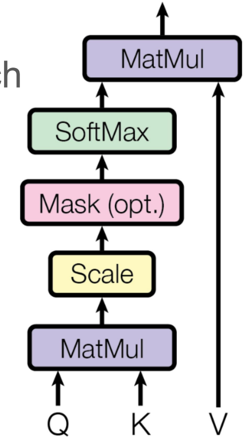
- They are responsible for the calculation of tha a_{ij} attention scores by defining a soft dictionary (similar to argmax vs. softmax)



$$\mathrm{Attention}(Q, K, V) = \mathrm{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$

What are the key/query matrices for?

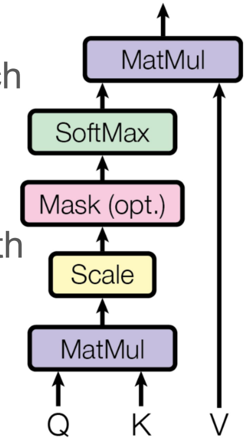
- They are responsible for the calculation of tha a_{ij} attention scores by defining a soft dictionary (similar to argmax vs. softmax)
    - Standard dictionaries assign a specific value to a given key (or none), whereas a soft dictionary assigns a(n adaptively) weighted sum of values to keys
    - Each input embedding is transformed into its query representation with W_Q and each input embedding is transformed into its key representation with W_K



$$\mathrm{Attention}(Q, K, V) = \mathrm{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$

What are the key/query matrices for?

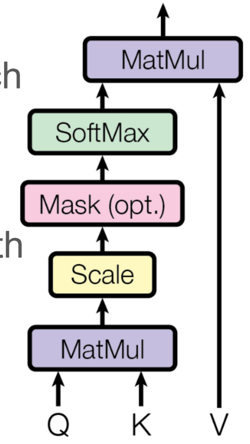
- They are responsible for the calculation of tha a_{ij} attention scores by defining a soft dictionary (similar to argmax vs. softmax)
  - Standard dictionaries assign a specific value to a given key (or none), whereas a soft dictionary assigns a(n adaptively) weighted sum of values to keys
  - Each input embedding is transformed into its query representation with W_Q and each input embedding is transformed into its key representation with W_K
- QK^T is essentially $(X\, W_Q)(W_K\, X^T)$
  - If both W_Q and W_K were the identity, we would then be left with XX^T, i.e. a matrix with the similarities between each pairs of input embeddings
    - It would artificially inflate the similarity of the embeddings with themselves



$$\mathrm{Attention}(Q,K,V) = \mathrm{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

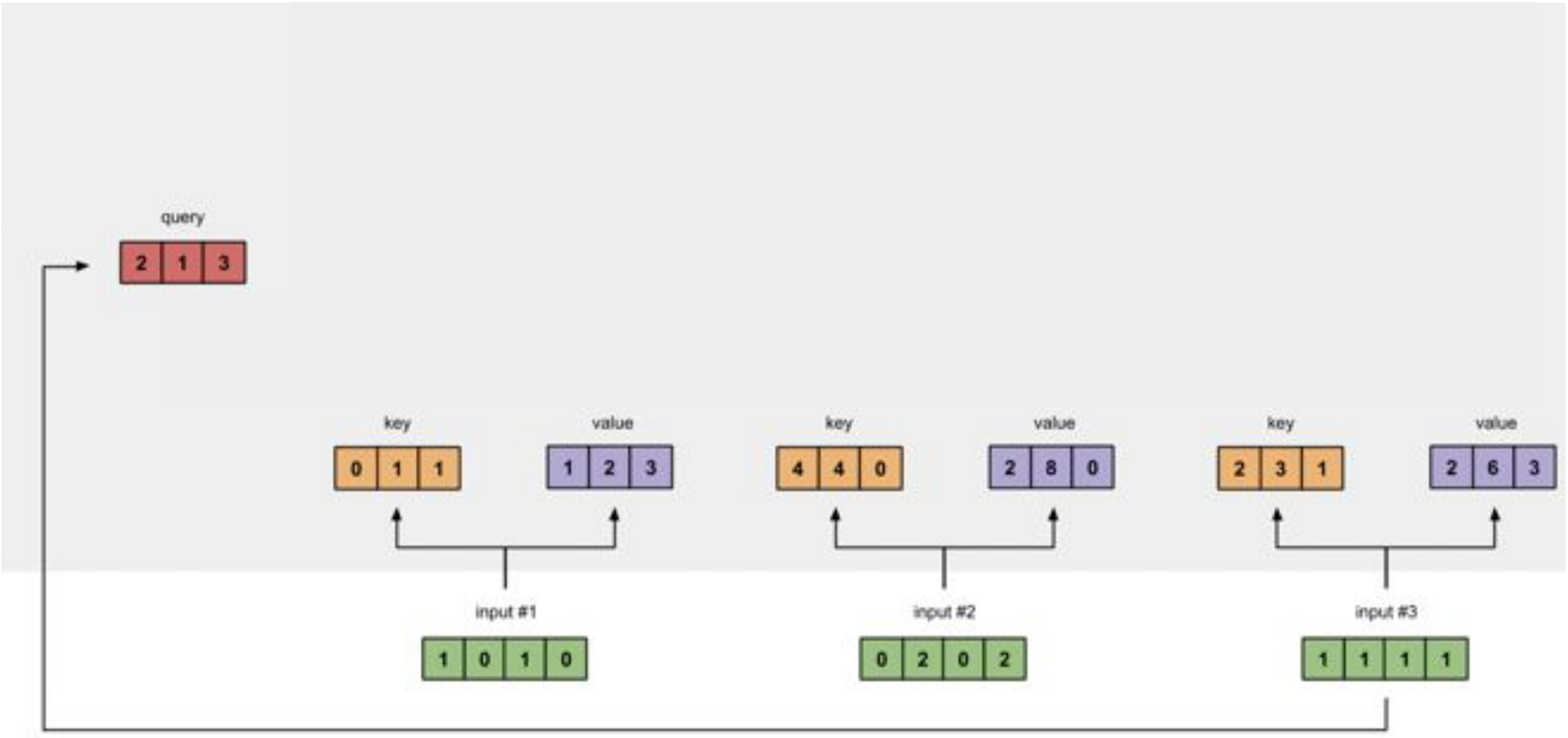
What are the key/query matrices for?

- They are responsible for the calculation of tha a_{ij} attention scores by defining a soft dictionary (similar to argmax vs. softmax)
  - Standard dictionaries assign a specific value to a given key (or none), whereas a soft dictionary assigns a(n adaptively) weighted sum of values to keys
  - Each input embedding is transformed into its query representation with W_Q and each input embedding is transformed into its key representation with W_K
- QK^T is essentially $(X W_Q)(W_K X^T)$
  - If both W_Q and W_K were the identity, we would then be left with XX^T, i.e. a matrix with the similarities between each pairs of input embeddings
    - It would artificially inflate the similarity of the embeddings with themselves
  - A single transformation ($W_{QK} \approx W_Q W_K$) could do the job as well as $(XW_Q)(W_K X^T) = X(W_Q W_K)X^T$
    - Disentangling is, however more effective when W_* has more rows than columns

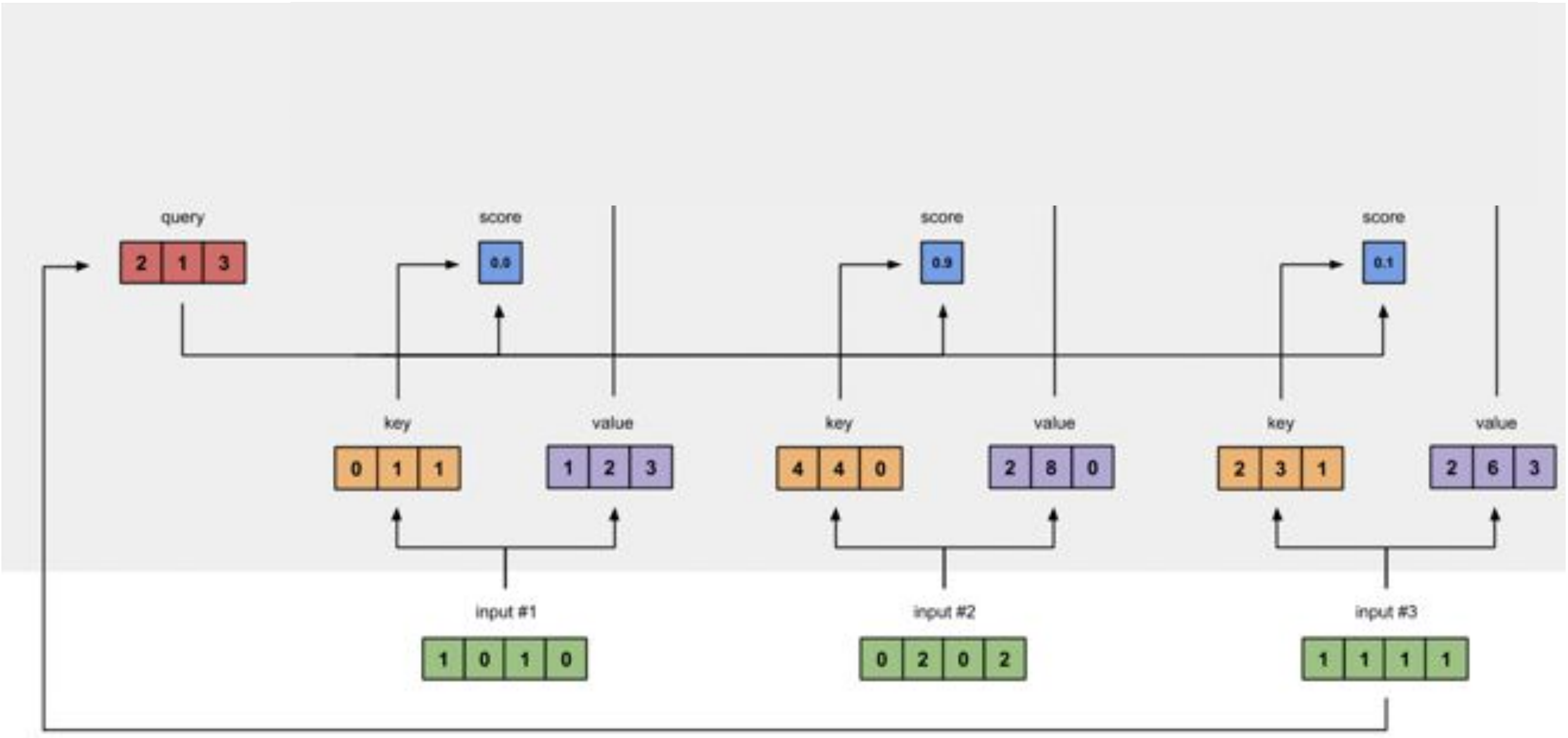


$$\mathrm{Attention}(Q,K,V) = \mathrm{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$

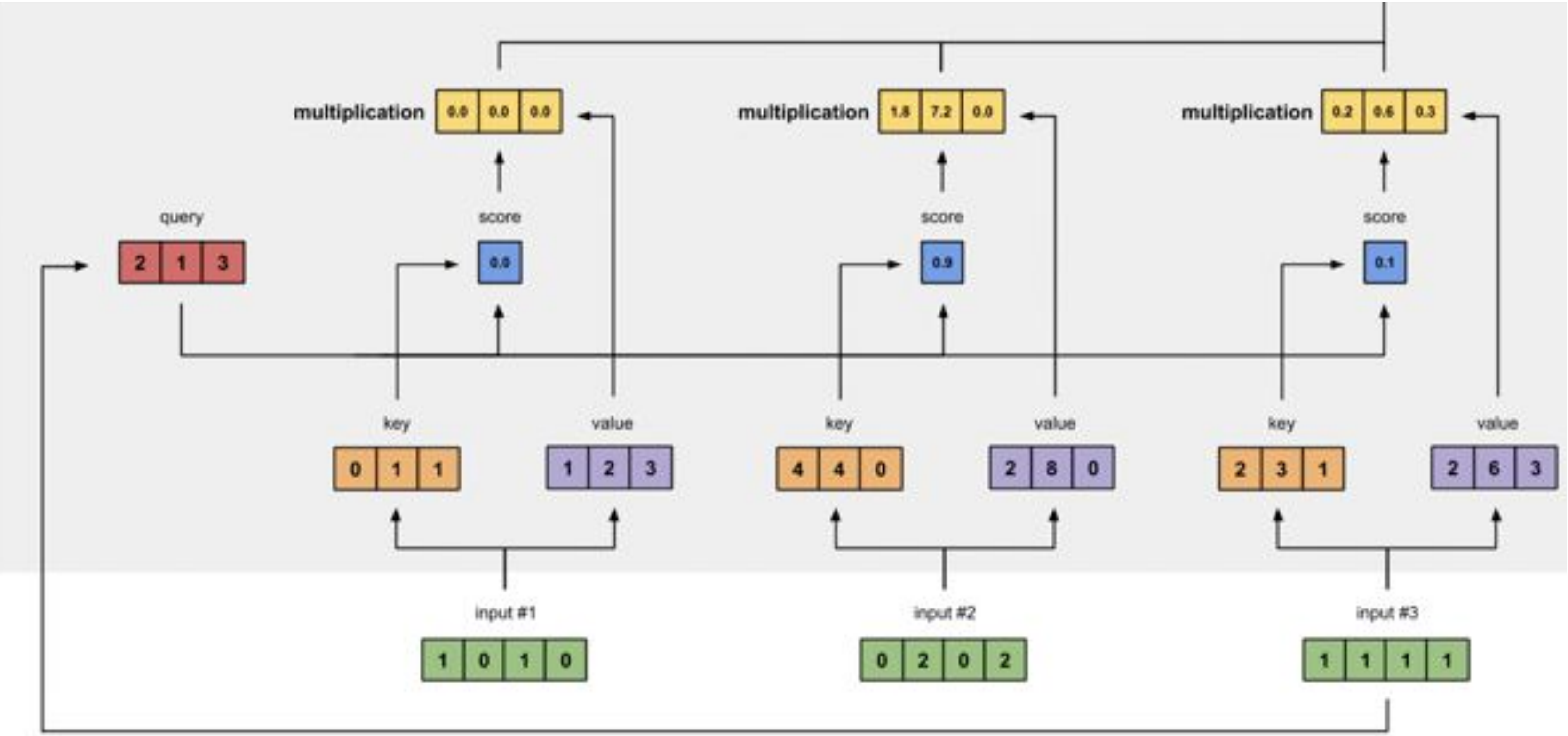
Putting all together



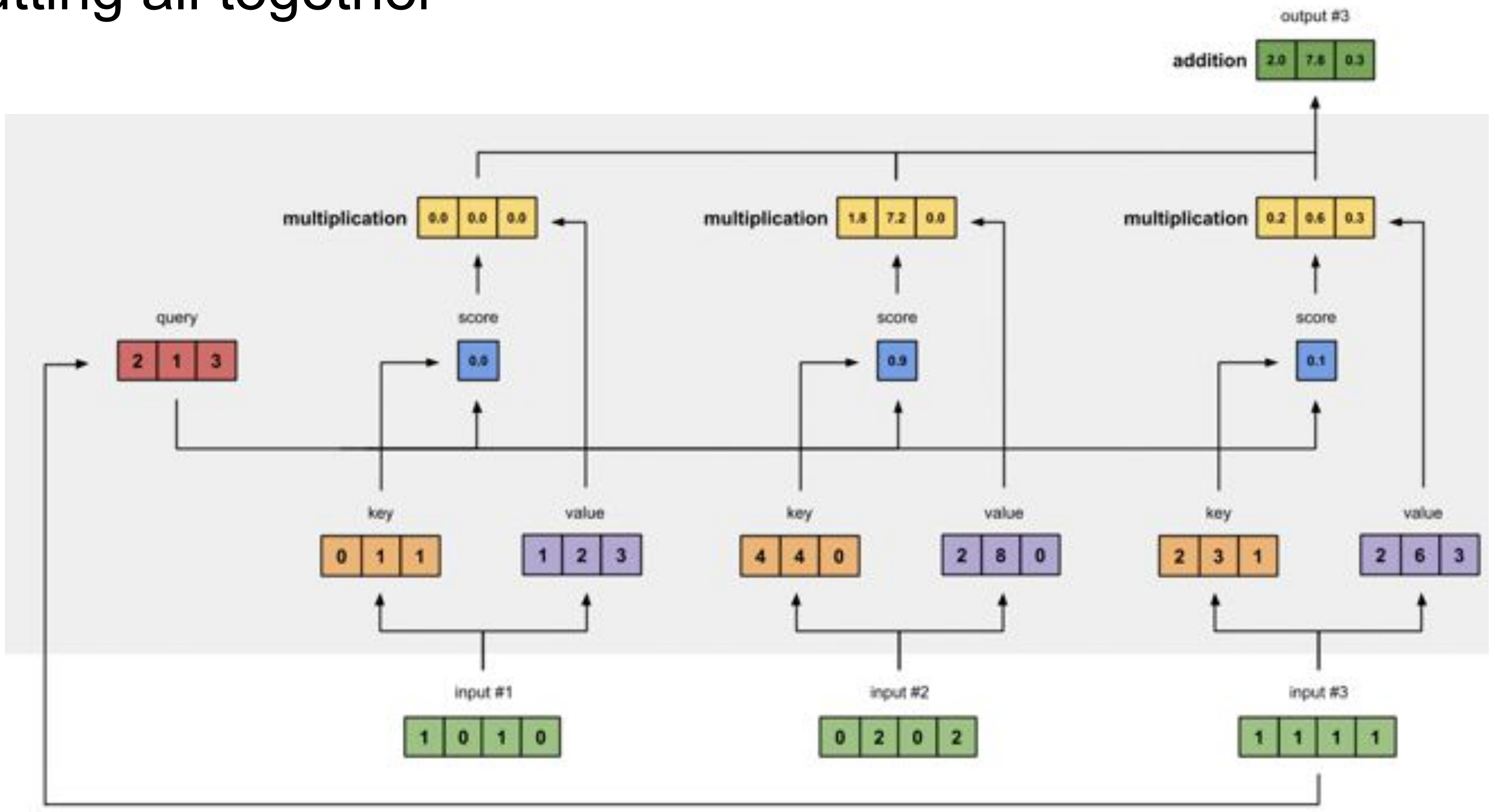
Putting all together



Putting all together

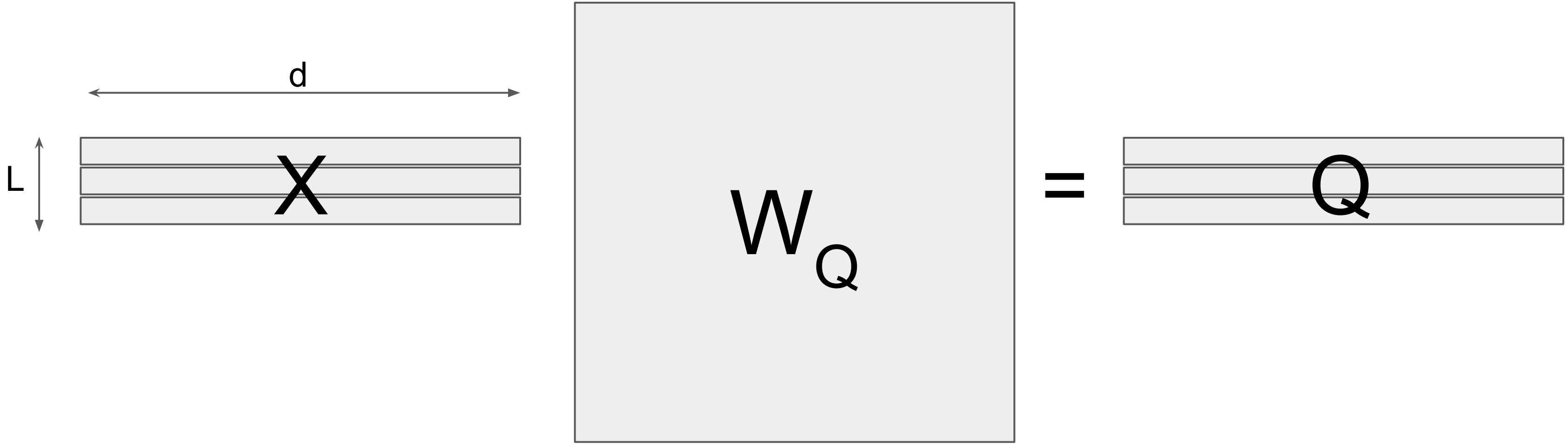


Putting all together



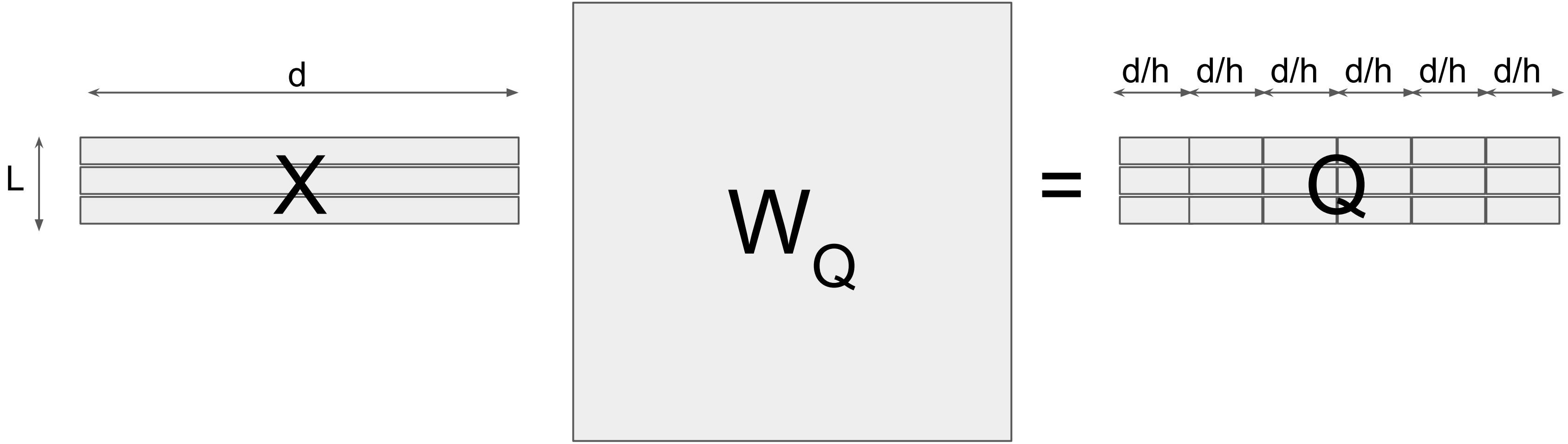
The multi-headed SA

- Hidden dimension *d* is treated as a composition of *h*x*(d/h)* subrepresentations
  - E.g. the 768 dimensional vectors of BERT-base can be viewed as a concatenation of 12 independent 64 dimensional representations
  - With slicing all the attention heads are calculated simultaneously



The multi-headed SA

- Hidden dimension *d* is treated as a composition of *h*x*(d/h)* subrepresentations
  - E.g. the 768 dimensional vectors of BERT-base can be viewed as a concatenation of 12 independent 64 dimensional representations
  - With slicing all the attention heads are calculated simultaneously



The multi-headed SA and the need for FFNNs

- Why we might need multiple heads?
  - Having multiple smaller attention heads in the same layer can be a surrogate to doing multiple attentions performed in subsequent layers
  - There are multiple independent aspects at the current semantic granularity level which can be incorporated into the representation of the next layer

The multi-headed SA and the need for FFNNs

- Why we might need multiple heads?
  - Having multiple smaller attention heads in the same layer can be a surrogate to doing multiple attentions performed in subsequent layers
  - There are multiple independent aspects at the current semantic granularity level which can be incorporated into the representation of the next layer
- As the results of MHSA is treated independently (using concatenation), they shall be blended together ⇒a FF layer is used for upscaling (typically to *4d*)

The multi-headed SA and the need for FFNNs

- Why we might need multiple heads?
  - Having multiple smaller attention heads in the same layer can be a surrogate to doing multiple attentions performed in subsequent layers
  - There are multiple independent aspects at the current semantic granularity level which can be incorporated into the representation of the next layer
- As the results of MHSA is treated independently (using concatenation), they shall be blended together ⇒a FF layer is used for upscaling (typically to *4d*), then another FF layer is used for downscaling to *d*

The multi-headed SA and the need for FFNNs

- Why we might need multiple heads?
  - Having multiple smaller attention heads in the same layer can be a surrogate to doing multiple attentions performed in subsequent layers
  - There are multiple independent aspects at the current semantic granularity level which can be incorporated into the representation of the next layer
- As the results of MHSA is treated independently (using concatenation), they shall be blended together ⇒a FF layer is used for upscaling (typically to *4d*), then another FF layer is used for downscaling to *d*
  - Question: Could another attention module maybe decide how to mix the individual SA heads?

The case of MHSA

- Actually, most of the SAs can be omitted, only a few does the ‘heavy lifting’
  - “*pruning 38 out of 48 encoder heads results in a [marginal] drop*” (Voita et al., 2019)
  - “*the number of attention heads doesn’t have a significant effect*” (K et al., 2020)

			XNLI		
Parameters (in Millions)	**Depth**	**Multi-head Attention**	**Fake-English**	**Russian**	**Δ**
132.78	12	1	77.4	63.2	14.2
132.78	12	2	78.3	62.8	15.5
132.78	12	3	79.5	65.3	14.2
132.78	12	6	78.9	66.7	12.2
132.78	12	16	77.9	64.9	13.0
132.78	12	24	77.9	63.9	14.0
132.78	12	12	79.0	65.7	13.3

The case of MHSA

- Actually, most of the SAs can be omitted, only a few does the ‘heavy lifting’
  - “*pruning 38 out of 48 encoder heads results in a [marginal] drop*” (Voita et al., 2019)
  - “*the number of attention heads doesn’t have a significant effect*” (K et al., 2020)
- The MH also fits the Lottery Ticket Hypothesis

			XNLI		
Parameters (in Millions)	**Depth**	**Multi-head Attention**	**Fake-English**	**Russian**	**Δ**
132.78	12	1	77.4	63.2	14.2
132.78	12	2	78.3	62.8	15.5
132.78	12	3	79.5	65.3	14.2
132.78	12	6	78.9	66.7	12.2
132.78	12	16	77.9	64.9	13.0
132.78	12	24	77.9	63.9	14.0
132.78	12	12	79.0	65.7	13.3

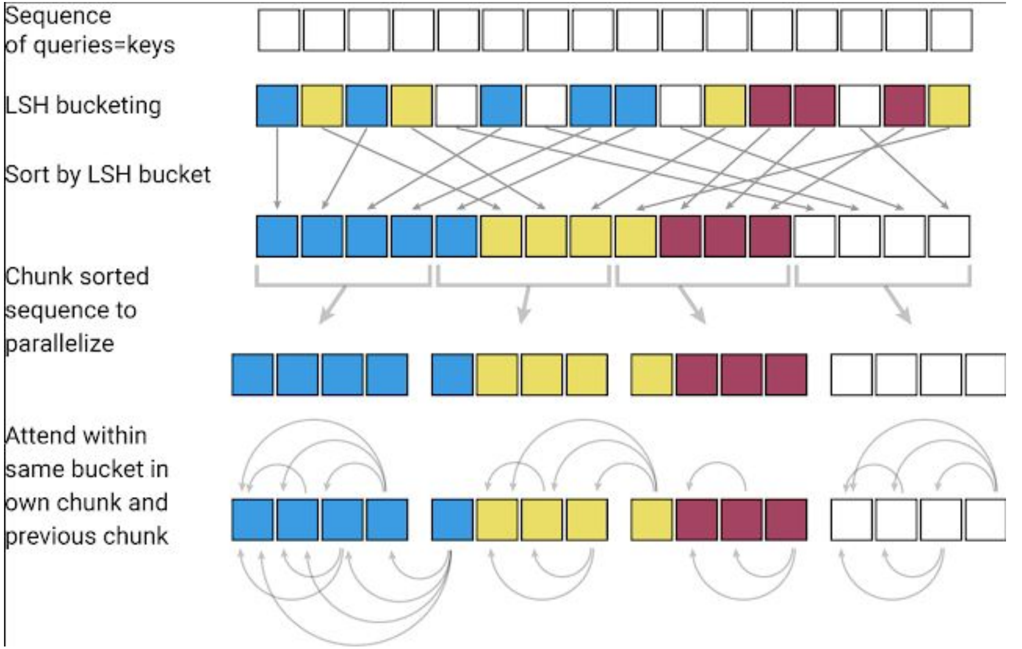
The case of MHSA

- Actually, most of the SAs can be omitted, only a few does the ‘heavy lifting’
    - “*pruning 38 out of 48 encoder heads results in a [marginal] drop*” (Voita et al., 2019)
    - “*the number of attention heads doesn’t have a significant effect*” (K et al., 2020)
- The MH also fits the Lottery Ticket Hypothesis
- OTOH, “reducing the amount of heads also decreases finetuning performance” (Geiping & Goldstein, 2022)

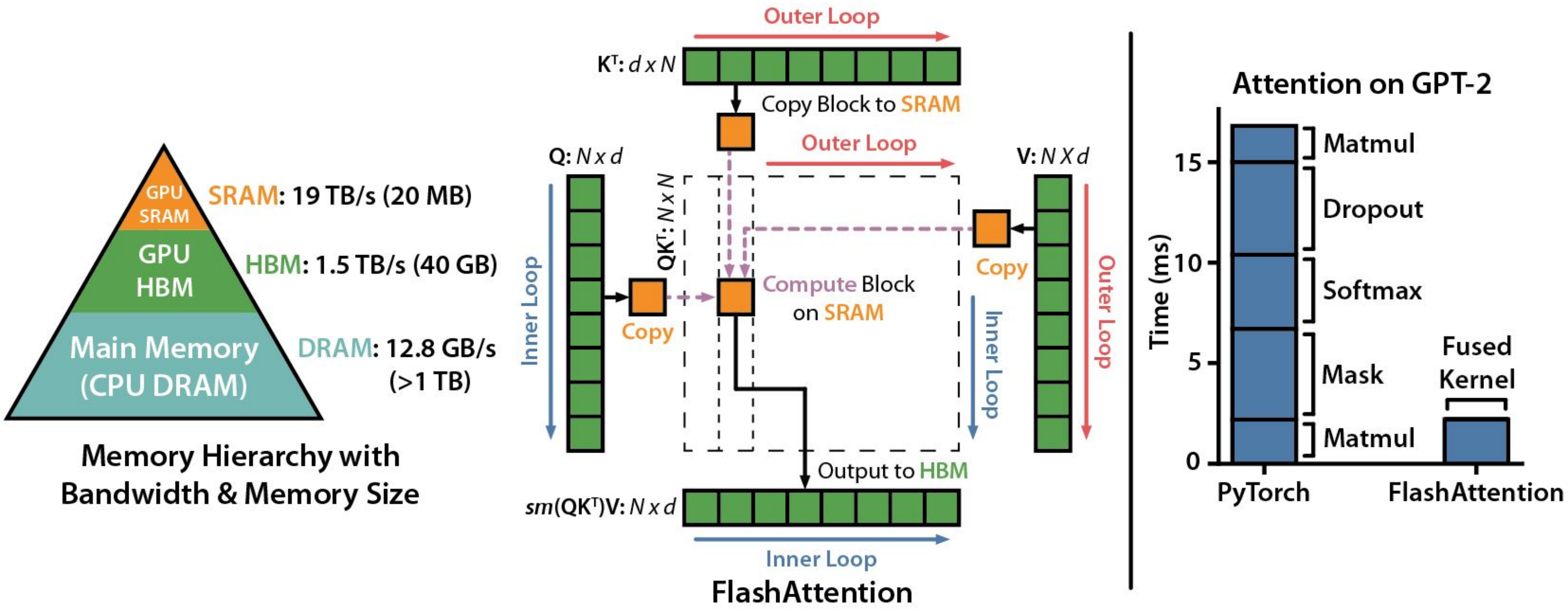
			XNLI		
Parameters (in Millions)	**Depth**	**Multi-head Attention**	**Fake-English**	**Russian**	**Δ**
132.78	12	1	77.4	63.2	14.2
132.78	12	2	78.3	62.8	15.5
132.78	12	3	79.5	65.3	14.2
132.78	12	6	78.9	66.7	12.2
132.78	12	16	77.9	64.9	13.0
132.78	12	24	77.9	63.9	14.0
132.78	12	12	79.0	65.7	13.3

Extensions to SA – Reformer

- For an input of length L, there are L^2 a_{ij} scores to compute :(
- Reformer: LSH to the rescue

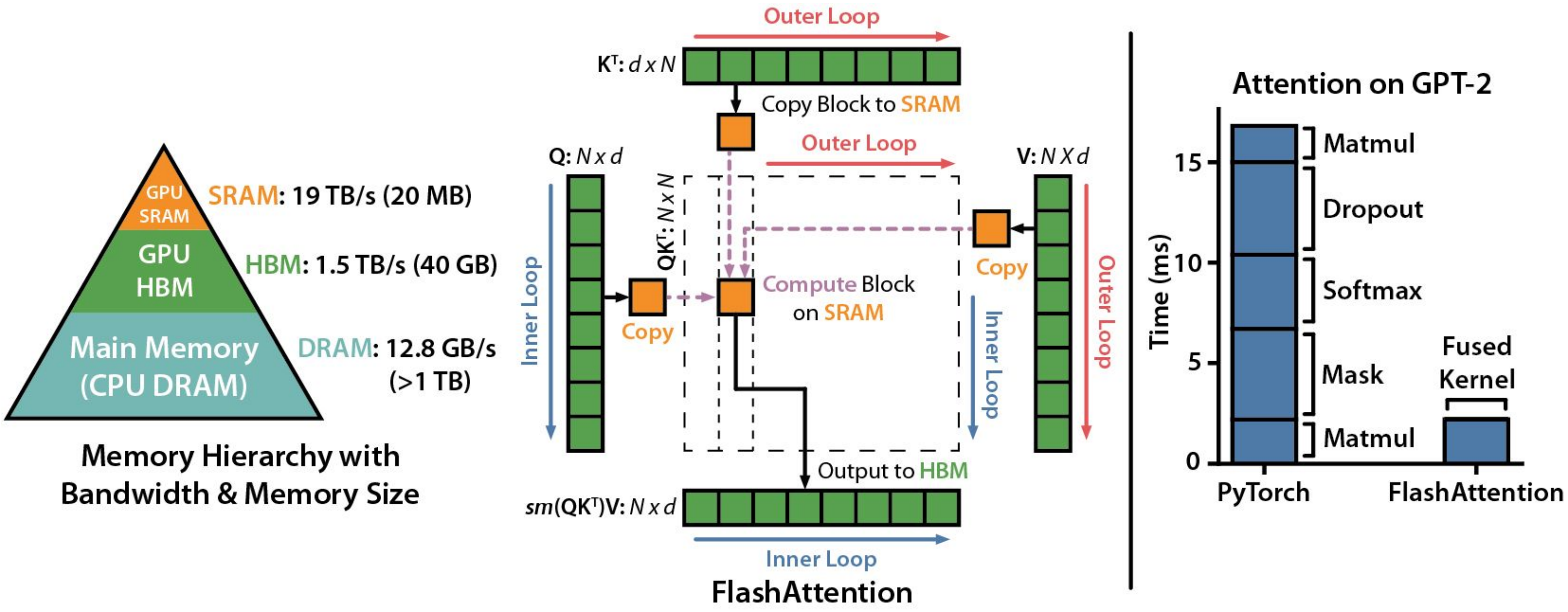


Extensions to SA – FlashAttention (Hua et al., 2022)



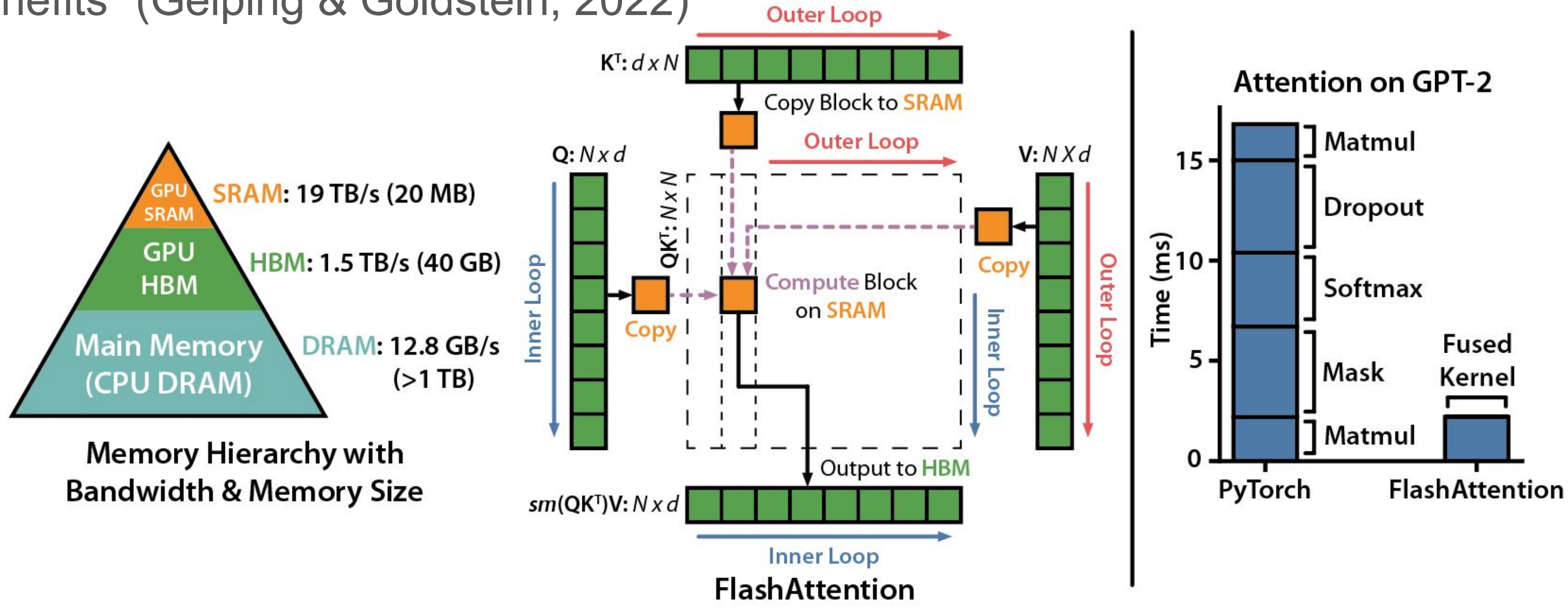
Extensions to SA – FlashAttention (Hua et al., 2022)

- Performs more FLOPs actually, but faster I/O, causing a speedup near up to 10x



Extensions to SA – FlashAttention (Hua et al., 2022)

- Performs more FLOPs actually, but faster I/O, causing a speedup near up to 10x
- OTOH, "we implement the recently proposed FLASH mechanism, but find no benefits" (Geiping & Goldstein, 2022)

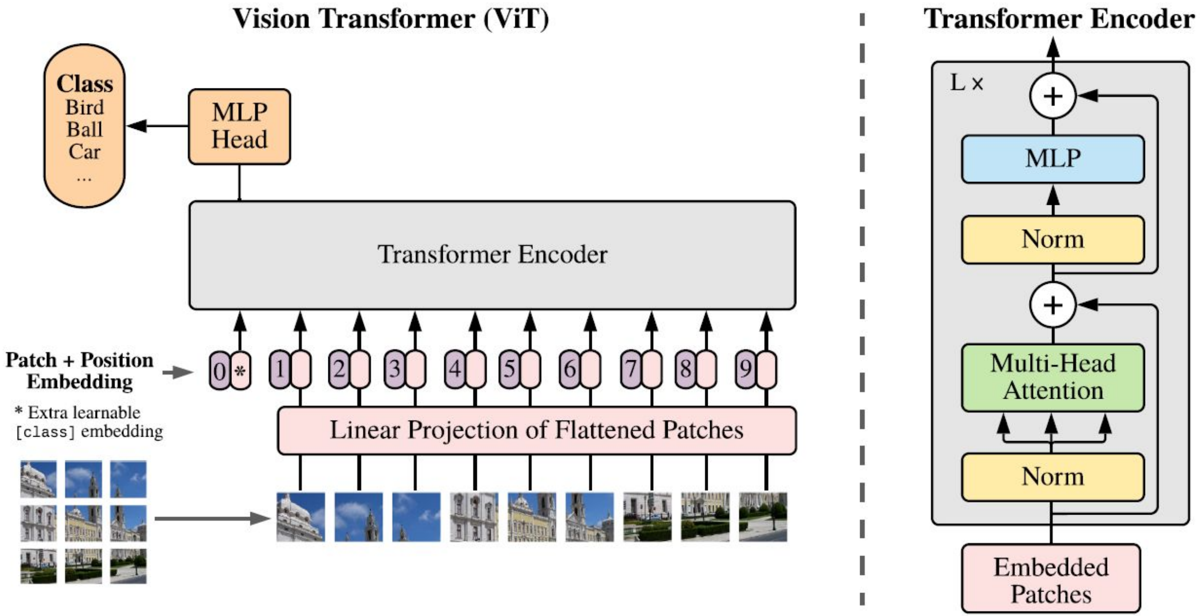


Further extensions

- DeBERTa: Relies on a disentangled attention mechanism
- RWKV: a combination of RNNs and transformers (with linear attention)
- Linformer
- Nystromformer
- Longformer
- Performer
- *former
- …

Attention in ViT

- Transformers (w/o positional embeddings) are meant for sets not sequences
    - There is (certain) evidence that the matter of the order words does not much



Further useful readings

- https://nlp.seas.harvard.edu/2018/04/03/attention.html
- https://lilianweng.github.io/posts/2023-01-27-the-transformer-family-v2
- https://jalammar.github.io/illustrated-transformer/
- https://stats.stackexchange.com/questions/421935/what-exactly-are-keys-queries-and-values-in-attention-mechanisms
- **Cross-Lingual Ability of Multilingual BERT: An Empirical Study**
- **Analyzing Multi-Head Self-Attention: Specialized Heads Do the Heavy Lifting, the Rest Can Be Pruned**