**Transformers - Scaled Dot Product attention**

======================================================================

The scaled dot product attention is a crucial component of the Transformer architecture. It's a mechanism that allows the model to weigh the importance of different parts of the input sequence when processing each element. Here's a breakdown of how it works:

--------------------------------------------------------------------------------------------------------------

**Inputs:**

Scaled dot product attention takes three inputs:

**Queries (Q):** These represent the elements in the input sequence for which we want to determine the relevant context. Think of it as "what am I looking for?".

**Keys (K):** These represent all the elements in the input sequence. Think of it as "what information do I have?".

**Values (V):** These also represent all the elements in the input sequence. Think of it as "what is the actual information associated with each element?".

All these inputs (Q, K, V) are typically derived by linearly transforming the input embeddings of the sequence.

-------------------------------------------------------------------------------------------------------------

**Steps:**

1. **Dot Product:** The first step is to calculate the dot product between each query vector and every key vector. This dot product measures the similarity or relevance between the query and each key. A higher dot product score indicates a higher degree of similarity.

Mathematically, if Q is the matrix of all query vectors and K is the matrix of all key vectors, this step is represented as QKT (where KT is the transpose of K). The resulting matrix represents the attention scores between each query and each key.



1. **Scaling:** The dot product scores can become quite large, especially for high-dimensional input vectors. Large values can lead to a very peaked SoftMax function (applied in the next step), where the probability mass concentrates on only a few elements, potentially hindering the learning of the model. To mitigate this, the dot product scores are scaled down by dividing them by the square root of the dimensionality of the key vectors (dk​).



This scaling helps to stabilize the gradients during training.

1. **Softmax:** Next, a softmax function is applied row-wise to the scaled attention scores. This converts the scores into probabilities, representing the weight or importance assigned to each key (and its corresponding value) for each query. The probabilities for each query sum up to 1.



1. **Weighted Sum:** Finally, the attention weights are multiplied by the value vectors (V). This step produces the output of the attention mechanism. The output for each query is a weighted sum of the value vectors, where the weights are the attention probabilities calculated in the previous step. The value vectors of the keys that had higher attention weights (i.e., were more relevant to the query) contribute more to the output.

The final output of the scaled dot product attention is:



-------------------------------------------------------------------------------------------------------------

**Intuition:**

The scaled dot product attention mechanism allows the model to focus on the most relevant parts of the input sequence when processing each word (or token). By comparing each query to all keys, the model identifies which parts of the input are most important for understanding the current query. The value vectors associated with these important keys are then weighted more heavily in the output, effectively allowing the model to gather context-aware representations.

**Why "Scaled"?**

The scaling by dk​​ is crucial for stable training. Without scaling, the variance of the dot products grows with the dimensionality dk​. This can lead to very small gradients after the softmax function, making it difficult for the model to learn. Scaling helps to normalize the variance and keeps the gradients in a reasonable range.

In summary, scaled dot product attention is a fundamental building block of Transformers that enables them to effectively process sequential data by dynamically weighing the importance of different elements in the sequence.