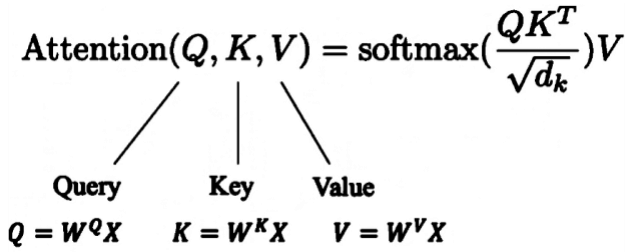
ECS795P Deep Learning and Computer Vision, 2024

Coursework 1: Building a Transformer from Scratch

1. The calculation formula for self-attention is as follows. Assuming the input X has dimensions [768, 6], and W^Q, W^K and W^V all have dimensions [768, 768], calculate the dimensions of K, Q, V, and the output of Attention(Q, K, V). (10% of CW1)



Answer:

(1) Since K=W^KX, Q=W^QX, V=W^VX, and the dimensions of W^K, W^Q *and* W^V *are* [768, 768], the dimension of X is [768, 6], we can get **the dimensions of K, Q, and V are [768, 6]**. In matrix multiplication, when two matrices are multiplied, the dimensions of the resultant matrix are determined by the "outer" dimensions of the multiplication. The "outer" dimensions refer to the number of rows of the first matrix and the number of columns of the second matrix. For instance, if there is an m x n matrix multiplied by an n x p matrix, the resultant matrix will have the dimensions of m x p.

(2) **The output of Attention(Q, K, V) has the dimension of [768, 6]. Here are the reasons:**

a. The transpose of matrix K is denoted by K^T, which inverts its dimensions to [6, 768]. When matrix Q with dimensions [768, 6] is multiplied by K^T, the resulting product is a matrix with dimensions [768, 768]. The softmax function is then applied row-wise to this [768, 768] matrix, which is the result of the operation Q*K^T divided by the square root of d_k, where d_k is the dimensionality of the keys(which is 768 in this case).

b. The multiplication of the output of the softmax function (which has dimensions [768, 768]) and the matrix V (which has dimensions [768, 6]) will have a matrix of dimensions [768, 6].

2. 1) How is the self-attention mechanism implemented?

Answer:

The self-attention mechanism is a core component of sequence-to-sequence models, and it is implemented through the following steps:

a. **Preparation of the Input Matrix**: Initially, there is an input matrix X, which typically represents the embedded representation of each element in a sequence. These embeddings contain semantic information about the elements. Then, three weight matrices are defined: W^Q (for query vectors), W^K (for key vectors), and W^V (for value vectors). These weights are model parameters, usually obtained through training, to extract different information from the sequence. Next, the Q, K, V are computed: Query (Q = W^Q * X), Key (K = W^K * X), and Value (V = W^V * X) are calculated using matrix multiplication. Q represents the information we are interested in, K represents the information available for selection, and V represents the actual content of the information. Q and K are used to calculate matching scores, while V is used to generate the output.

b. **Attention Score Calculation:** Attention scores are calculated by taking the dot product of the query Q and the transpose of the key K^T, which indicates the relevance of each key to a given query. The scores are then scaled by dividing by the square root of the dimension of the key vectors. This scaling is to prevent the dot product from producing excessively large scores, which could lead to gradient vanishing or exploding issues during training, helping to maintain gradient stability.

c. **Application of the Softmax Function:** The softmax function is applied to the attention scores, transforming them into a probability distribution. This indicates the relative importance of each key in the context of all queries.

d. **Weighted Sum Output:** The output of the softmax function (i.e., the probability distribution) is used as weights and multiplied by the value matrix V, resulting in a weighted sum output. This ensures that values with higher attention weights have a greater influence on the final output.

e. F**inal Self-Attention Output:** The final output is a collection of all weighted sums (i.e., the output of the attention mechanism), which merges information from all elements in the input sequence. The output for each element is dynamically weighted based on its relationship with other elements. This dynamic weighting process is at the heart of the self-attention mechanism.

2) What are the primary applications of self-attention? (mention 3 points)

Answer:

a. **Image classification:** The self-attention mechanism can help models identify key features and objects in images. By capturing the relationships between different regions of an image, it improves the accuracy of classification.

b. **Machine translation:** Because the self-attention mechanism helps capture the global dependencies between words in the input sequence, it significantly enhances the performance of machine translation.

c. **Sentiment analysis:** The self-attention mechanism, by capturing the emotional dependencies between different words or phrases in the text, can bring higher precision to sentiment analysis tasks.

3) What advantages does transformer have compared to CNN? (10% of CW1)

Answer:

The Transformer model has several advantages over CNNs as follows:

a. Parallel Processing: CNNs frequently process sequences sequentially by layers, however the Transformer can compute all points in a sequence in parallel, greatly cutting down on model training time.

b. Capturing Long-Distance Dependencies: The Transformer's self-attention mechanism effectively captures long-distance dependencies inside sequences without the need to stack additional convolutional layers, which could reduce CNN performance.

c. Flexibly Handling Sequences of Varying Lengths: The Transformer's position encoding allows it to handle sequences of any length without the requirement for truncation or padding, which CNNs frequently require to adapt sequence lengths.

3. Answer the following questions based on the code:

1) In the section "From attention to transformers - Multi-head attention", what are the shapes of the following variables: *tokens, qis_mh, kis_mh, vis_mh, attmat_mh, zis_mh, zis*?

a. The shape of tokens is [1, 5, 768].

b. The shape of qis_mh is [1, 5, 12, 64].

c. The shape of kis_mh is [1, 5, 12, 64].

d. The shape of vis_mh is [1, 5, 12, 64].

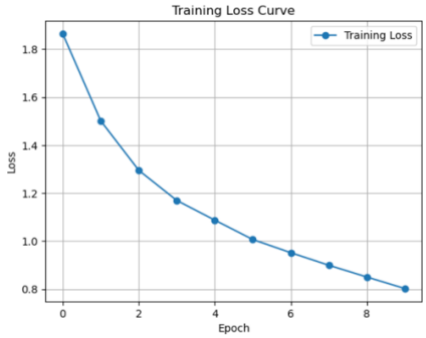
e. The shape of attmat_mh is [1, 12, 5, 5].

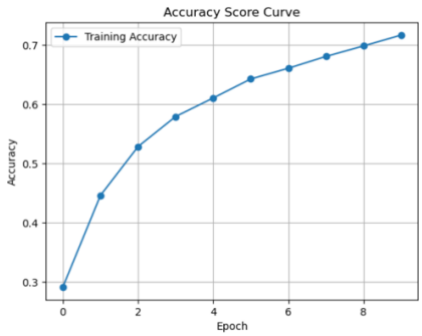
f. The shape of zis_mh is [1, 5, 12, 64].

g. The shape of zis is [1, 5, 768].

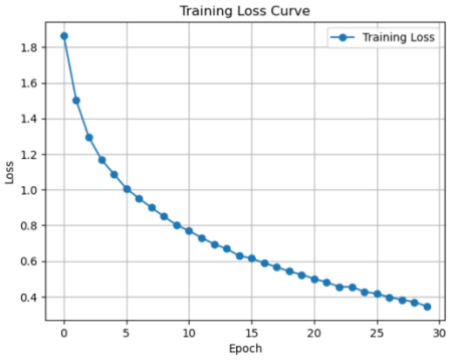
2) In the 'Classification' section, plot the curves of classification accuracy and loss as they vary with epochs for three different settings: 10 epochs, 30 epochs, and 50 epochs. (10% of CW1)

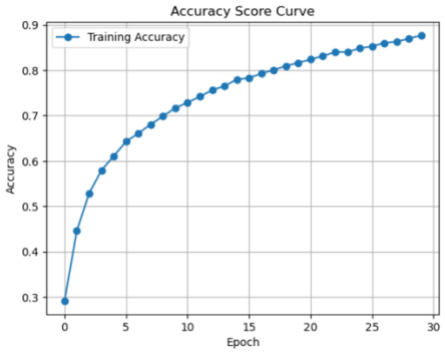
10 epochs:



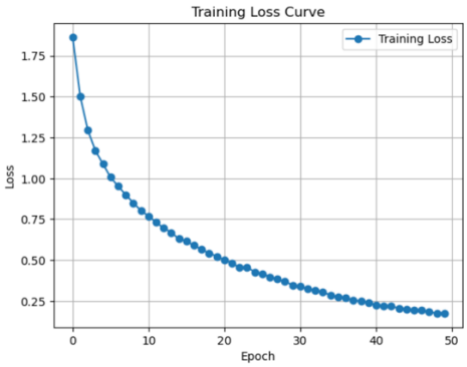


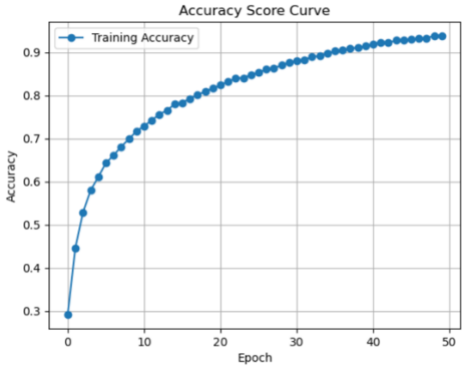
30 epochs:





50 epochs:





As the number of epochs increases, the gradual decrease in loss indicates that the model is learning more and more features from the training data. Meanwhile, the gradual increase in accuracy indicates that the model's performance on the training set is improving, and it also suggests that the model's ability to make accurate predictions on unseen data is getting stronger. However, as the training duration extends, the model may face the risk of overfitting.