Design of Model architecture

Jaehyeon Shin

In the last report, I analyzed each of Components in model architecture. I am going to explain why the model architecture is designed this way.

1. Convolutional Neural Network

- First, I will write the roles of each component in the basic structure of a CNN(Convolutional Neural Network). In the basic, CNN typically consists of an input layer, convolutional layer, activation function, pooling layer, fully connected layer, and output layer. Briefly, the input layer resizes the image so it can be used in the subsequent layers, while the convolutional layer extracts features, such as edges. The activation function is used to capture non-linear patterns, as the data is typically complex and non-linearly separable. The pooling layer preserves important features while reducing the spatial dimensions. In the case of classification, the fully connected layer integrates all the information up to this point to classify the input. After that, output layer (e.g. softmax) produces the final prediction. In summary, the CNN extracts features, passes them through activation functions to capture more complex characteristics, and summarizes this information, which is then fed into the FC layer to produce the output.
- This architecture is efficiently and well-designed. If any component is missing, or if the order of components within the architecture is altered, the design is likely to perform less effectively compared to the original structure.
- If the convolutional layer is omitted, the model will be unable to extract local spatial features. Additionally, without the weight sharing property, the number of parameters increases, leading to issues with memory and computational resources. If the activation function is omitted, since only linear operations are performed within the model, it becomes a simple model incapable of capturing non-linear relationships. As a result, it will show poor performance when faced with complex tasks. If pooling layer omitted, model loses down-sampling. Then, the model uses more memory than the original, and the amount of computation increases accordingly. The risk of overfitting also increases. If FC layer omitted, the model loses the ability to combine information. And FC layers usually reduce the final into a fixed size vector. Without this, it is hard to link the learned features to specific output classes.
- If we switches the order of convolutional layer and activation function, it may lose important information. For example, when using ReLU, all negative values are turned into zeros, which means the model may not extract features as well as before. If the position of the pooling layer is changed, and it comes before the activation function, the summarizing happen before the activation is applied, which could reduce the ability to capture complex pattern Also, if it comes before the convolutional layer(Pooling -> Convolutional layer -> Activation function), summarizing from the start could result in information loss. Finally, if the Fully connected layer comes first instead of last, the raw image, which is multi-dimensional data, would need to be transformed into one-dimensional input vector. This can cause the FC layer to ignore important spatial relationships. This make it unsuitable for tasks like image recognition.
-

2. Transformer

- The component of the Transformer model consists of input embedding, positional encoding, Multi-head Attention, Feed-forward, Masked Multi-head Attention, output generation using linear and Softmax layer.
- Briefly, I will first describe the roles of each component. The embedding layer convert discrete token into the vector representations. The Positional encoding adds positional information to the input embeddings. The Multi-head attention captures contextual information such as relevance from multiple views. The Masked Multi-head attention does same role but future information is masked to maintain

causality. The Residual connection and layer normalization help preserve the original input while incorporating new information, preventing information loss and mitigating the vanishing gradient problem, thereby facilitating faster convergence. The Feed Forward layer applies non-linear transformations to the representation at each position, improving capacity to capture complex patterns. Linear and softmax generate output probabilities.

- Upon receiving an input sentence, the embedding layer transforms it into a vector representation, with positional encoding added to incorporate word order information. In the encoder, multi-head attention captures the relationships between words, while residual connections and normalization stabilize the learning process. The feed-forward layer further refines the representations, and this encoder process is repeated iteratively to generate context-rich representations. The output sequence, which serve as input to the decoder, does a similar transformation through the embedding layer, with positional encoding added to discern the relationships between previously generated words. Encoder-decoder attention then aligns the input and output sequences by learning contextual relationships between the encoder's output and the decoder's current state. The feed-forward layer enhances these representations, and this process is repeated multiple times to produce a deeply contextualized output. Finally, the output is passed through linear and softmax layers to convert it into a probability, where the word with the highest probability is selected. This process is repeated to generate the entire sequence.

- This architecture is efficiently and well-designed. If any component is missing, or if the order of components within the architecture is altered, the design is likely to perform less effectively compared to the original structure.

- First, the input embedding layer must come at the beginning. Since input token need to be transformed into vectors, it naturally appears at the front. As the Transformer processes all tokens simultaneously, without positional encoding, the model cannot account for word order or understand sentence structure, which would lead to suboptimal performance. Additionally, it is logical to include positional information before other transformations, making it appropriate to place positional encoding right after embedding layer.

- If Multi-head attention is omitted, there will be a lack of contextual understanding, as word relationships in a sentence are crucial. Without it, model will show poor performance. Moreover, if the process of capturing context occurs after transformation (feed-forward layer), it become less efficient in capturing relationships compared to the original design. If the feed-forward layer is omitted, the model's ability to capture intricate patterns and relations through non-linearity is severely limited, significantly reducing its representatives. Higher-order information is lost, leading to a drop in overall performance. This layer refines each word's representation based on the context provided by multi-head attention. While the attention mechanism learns global relationships, the feed-forward layer focuses on local details, making it an essential component for the model to learn balanced representations. So, this component is necessary. If the order between the two is reversed, as mentioned earlier, attention would be applied to transformed information, which is not optimal.

- Additionally, residual connections and normalization are present after multi-head attention and feed-forward layer. These mechanisms ensure that the model can process and learn information stably at each stage, maintaining a consistent learning process regardless of depth. Without them, the model is prone to vanishing or exploding gradient problem, and the output distributions across layers could become inconsistent, leading to instability during training.

- The shifted output involves shifting the actual target sequence one word to the right, enabling the model to predict the next word based on the preceding words at each step. By employing the teacher forcing method, the training process utilizes the target sequence as input, thereby facilitating more stable model training. In the absence of this method, the model relies on previously generated words as input, and any errors in these generated words can influence the generation of subsequent words, resulting in the accumulation of errors. Consequently, the quality of the final generated sequence deteriorates, and the model may fail to achieve the expected performance through training.

- Consequently, if masking is not applied in the multi-head attention, the multi-head attention mechanism

would incorporate contextual information from future words, thereby preventing the generation of natural and coherent words. Therefore, masking is indispensable. Through it, the decoder captures the relationship with the previously generated words in the output sequence, enabling the production of accurate and consistent outputs. Although one might think that since attention mechanisms capture context, Encoder-Decoder attention alone could suffice without Masked multi-head attention, the two attention mechanisms serve complementary roles and are essential for generating high-quality, and meaningful output sequences. The combination of both attention mechanisms allows the Transformer decoder to effectively learn both the global context and the internal relationships within the output sequence. In contrast, utilizing one of these would result in suboptimal performance.

- Residual connections and normalization in the decoder perform roles analogous to those in the encoder, ensuring stable. These are also crucial. The feed-forward layer is critical as it integrates information from the two attention mechanisms and refines the representations of output words, thereby enhancing the model's expressive capacity. Placing the feed-forward layer in its designated position maximizes efficiency by facilitating the integration and transformation of contextual information derived from both attention mechanisms. Lastly, the linear layer is essential as it transforms the decoder's output to match the dimension of the vocabulary, and softmax is necessary to compute the probability of each word appearing in the sequence.

- In summary, the Transformer architecture effectively implements a natural methodology where output words are generated based on the contextual information of the input sequence. By integrating the context of the generated words with that of input sequence and using attention mechanism for enabling parallel computation and addressing global dependencies, the model successfully produces a accurate output sequence.

3. Mamba.

- The architecture of the Mamba model consists of projection, conv1d, the SiLU activation function and multiplication. Briefly, the roles of each component is as follows. The Linear projection is located at the beginning and tend of the block to transform the input into an easier-to-process format. Conv1d learns local patterns from sequence data. The SiLU activation function introduces nonlinearity, allowing the model to learn more complex patters. Selective SSM efficiently captures temporal dependencies in sequence data and selectively processes important information. Multiplication acts combining information. To outline the flow, the sequence is first represented as an embedding vector, and Conv1D is used to capture the semantic relationships between adjacent words. The SiLU activation allows the model to learn beyond simple linear transformations. This result is then passed to the selective SSM, which learns temporal patterns and dependencies. Following this, the input is combined with the SiLU-activated result through multiplication, reinforcing important information while suppressing less relevant data. Finally, a linear projection is applied to transform the result into the final output.

- This architecture is efficiently and well-designed. If any component is missing, or if the order of components within the architecture is altered, the design is likely to perform less effectively compared to the original structure.

- In the Mamba paper, the section on model ablations explores how adjusting specific components impacts performance with perplexity used as the evaluation metric. A lower perplexity indicates that the model predicts the data more accurately. When comparing non-selective (LTI) SSMs to selective SSMs, the latter showed a significant performance improvement because it captures better and process temporal dependencies. Additionally, selective parameter Δ, B, and C are used. Experimental results show that Δ is the most important parameter. This is because Δ plays a central role in deciding which information the model should focus on, significantly influencing the entire learning process. B and C synergize with Δ because the model is designed to adjust the interactions between the input and state, as well as between the state and output.

- If conv1d is removed, the model would struggle to effectively learn local patterns. It is difficult for the

SSM alone to capture short-range patterns. If the activation function is omitted, the model would not be able to learn non-linearities, making it challenging to capture complex pattens. If the selective SSM is removed, the model would struggle to capture long-term dependencies, the model would be unable to selectively emphasize important information, leading to increased noise and reduced predictive performance.

- It the order of components is altered, the model's performance may decrease compared to the original design. If the activation function appears after the SSM instead of after conv1d, the SSM would learn pattern form linearly transformed data, limiting its ability to capture complex relationship. If the activation is placed before both conv1d and the SSM, non-linearity would be applied, distorting the local patterns processed by conv1d. It may make difficult to learn fundamental patterns. Therefore, the natural flow is to first learn short-term patterns through conv1d, apply non-linearity, and then selectively learn long-term patterns, after which important information is emphasized.
- If RNNs were used instead of Conv1D, the sequential nature of RNNs would make parallel processing difficult, reducing computational efficiency when handling long sequencesFurthermore, using an activation function like ReLU instead of SiLU could lead to the dead neurons problem. And unlike other functions, SiLU is not monotonic, allowing it to capture more complex relationships in the data. The multiplication operation is used because it provides efficient results with a simple mechanism.