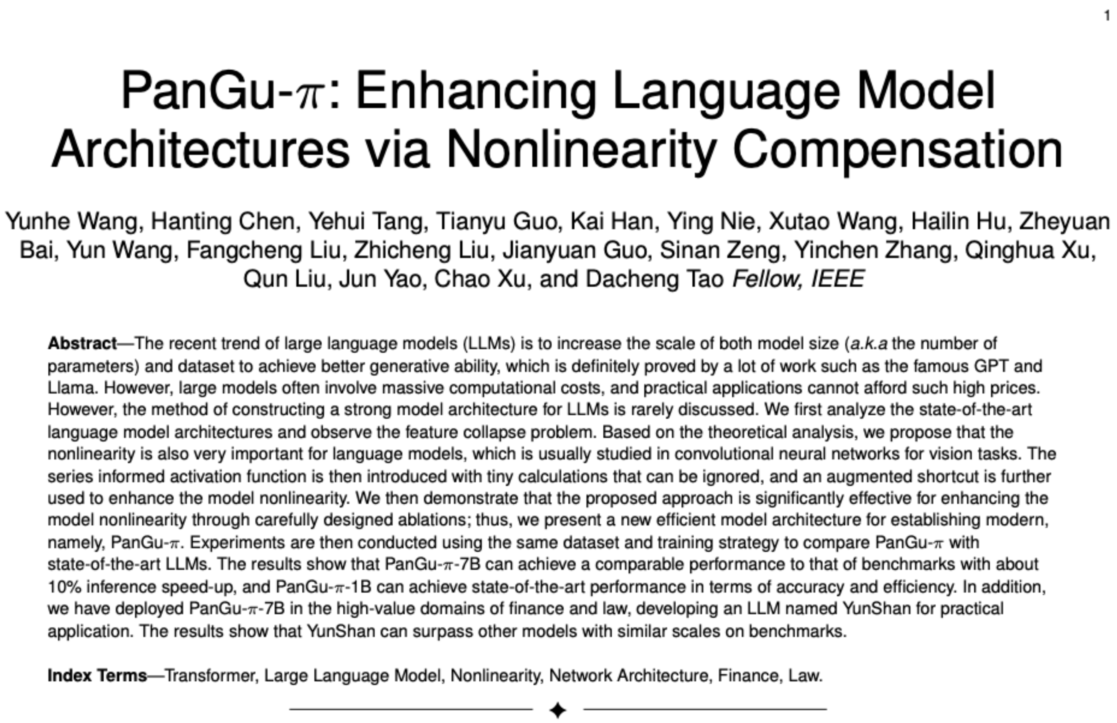

PanGu-π: Enhancing Language Model Architectures via Nonlinearity Compensation

Yunhe Wang, Hanting Chen, Yehui Tang, Tianyu Guo, Kai Han, Ying Nie, Xutao Wang, Hailin Hu, Zheyuan Bai, Yun Wang, Fangcheng Liu, Zhicheng Liu, Jianyuan Guo, Sinan Zeng, Yinchen Zhang, Qinghua Xu, Qun Liu, Jun Yao, Chao Xu, and Dacheng Tao *Fellow, IEEE*

Abstract—The recent trend of large language models (LLMs) is to increase the scale of both model size (*a.k.a* the number of parameters) and dataset to achieve better generative ability, which is definitely proved by a lot of work such as the famous GPT and Llama. However, large models often involve massive computational costs, and practical applications cannot afford such high prices. However, the method of constructing a strong model architecture for LLMs is rarely discussed. We first analyze the state-of-the-art language model architectures and observe the feature collapse problem. Based on the theoretical analysis, we propose that the nonlinearity is also very important for language models, which is usually studied in convolutional neural networks for vision tasks. The series informed activation function is then introduced with tiny calculations that can be ignored, and an augmented shortcut is further used to enhance the model nonlinearity. We then demonstrate that the proposed approach is significantly effective for enhancing the model nonlinearity through carefully designed ablations; thus, we present a new efficient model architecture for establishing modern, namely, PanGu-π. Experiments are then conducted using the same dataset and training strategy to compare PanGu-π with state-of-the-art LLMs. The results show that PanGu-π-7B can achieve a comparable performance to that of benchmarks with about 10% inference speed-up, and PanGu-π-1B can achieve state-of-the-art performance in terms of accuracy and efficiency. In addition, we have deployed PanGu-π-7B in the high-value domains of finance and law, developing an LLM named YunShan for practical application. The results show that YunShan can surpass other models with similar scales on benchmarks.

Index Terms—Transformer, Large Language Model, Nonlinearity, Network Architecture, Finance, Law.

Today we're reviewing a paper that proposes an upgrade to the transformer architecture. As you probably know, a transformer block consists of two main parts (aside from normalization layers):

- Multi-head self-attention mechanism (MSA)
- A fully connected (MLP) layer composed of a linear layer with a non-linear activation function followed by another linear layer (without an activation function).

Recall that the goal of a transformer block is to generate context-dependent or contextualized representations of the input tokens. That is, the representation of each token takes into account the tokens within its context. The authors analyze the properties of the contextualized token representations generated by transformers by comparing them with the token representations fed into the first block of the transformer (i.e., the token representations from the language model's embedding matrix). The improvements proposed in the paper aim to prevent a situation where the contextualized representations of tokens are very similar to each other.

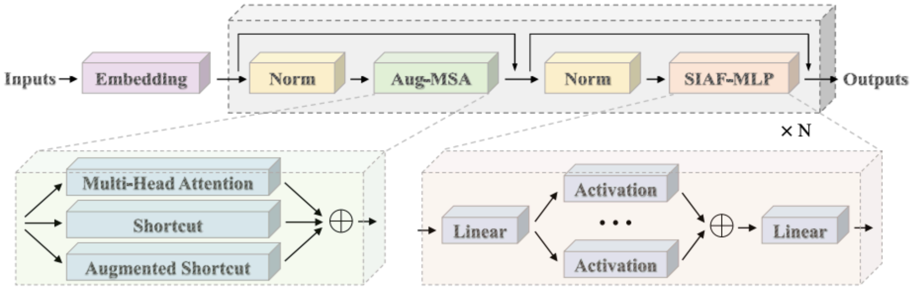

Fig. 2: The diagram of the proposed PanGu-π architecture. The series activation function is adapted to FFN, and the augmented shortcuts are integrated into MSA, which effectively introduces more nonlinearity into the Transformer architecture.

A phenomenon similar to the one described in the end of the previous paragraph is called over-smoothing in Graph Neural Networks (GNN). It happens when there are too many MSA layers, leading to very similar representations of the vertices that might cause the representations to "collapse" into a small subspace of the input representation space. The attention weights matrix in transformers can be seen as a normalized adjacency matrix of a complete graph.

But how do we measure the diversity between token representations? The paper defines the diversity of a matrix M (which is actually a set of vectors) as the minimum Frobenius norm of the difference of M-A over all rank 1 matrices A (all vectors constructing this matrix are linearly dependent).

The authors show that for a model composed of only l stacked MSA blocks (without MLP), the diversity of the output representations can be bounded by the product of the maximal singular values (a generalization of eigenvalues for non-square matrices) of the different weight matrices in the MSA mechanism and the diversity of the input representation (taken from the language model's embedding matrix). Without MLP layers, these representations tend to degenerate and become linearly dependent as the number of blocks l increases. This is the main reason for the existence of MLP in transformers.

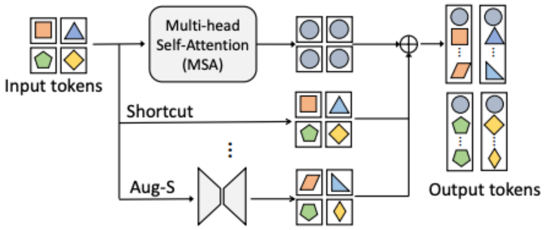

Fig. 3: The diagram of MSA module equipped with augmented shortcuts, where different patterns (rectangle, triangle, etc.) denote different features from various tokens. The original identity shortcut copies the input feature while the augmented shortcuts (Aug-S) project features of each input token to diverse representations.

Furthermore, for the model composed of stacked MLP blocks only, the paper proves that the output representation diversity is a product of the input representation diversity, the maximal singular values of the weight matrices, and the Lipschitz constants of the MLP activation functions.

To improve the properties of the output token representations of the transformer, the paper suggests two modifications, one for MSA and the other for MLP. Recall that in the transformer block, we have a residual connection - that is, the output of MSA is connected to the input representations fed to MSA. The authors propose to develop additional "shortcut" connections. Each such connection is essentially a linear layer with a learned matrix and a non-linear activation function. To not overly increase the computational load added as a result (the weight matrices in these shortcut connections can be 4096x4096, which is quite a lot if you want to use several such shortcuts), they use matrices of low rank. The authors prove that adding such layers to the original transformer blocks contributes to reducing the damage to the diversity of output representations.

In addition, the paper suggests upgrading the activation function, which is an essential part of the transformer mechanism in addition to MSA. Instead of using standard activation functions (like sigmoid or ReLU), the paper proposes to combine (additively) n activation functions with linearly coupled parameters a_i and b_i:

$$\sum_{i=1}^{n} \sigma_i(a_i x + b_i),$$

Of course, there is proof that such a modification of activation functions contributes to increasing the diversity between the output representations.

In addition, the proposed improvements were tested on several benchmarks and showed not bad performance.