RWKYV: Reinventing RNNs for the Transformer Era

Overview:

The paper presents the innovative Receptance Weighted Key Value (RWKV) network architecture. By merging the advantages of Recurrent Neural Networks (RNNs) and Transformers, and removing their limitations, it affords computational efficiency and robust expressive capacity. Inspired by ongoing challenges in the scalability and deployment of AI models, this novel architecture targets tasks necessitating large-scale models with billions of parameters, primarily in sequential data processing. Its transformative feature is an attention mechanism reformulation that diminishes the quadratic complexity intrinsic to standard Transformer models, resulting in linear attention. Albeit, this alters the model's capacity to recall detailed information over extensive contexts. Through a series of experiments on benchmark datasets, RWKV has demonstrated competitive performance, efficiency, and scalability. Pretrained models have been released to underline this. Despite its capabilities, the architecture's linear attention presents constraints and its performance is highly dependent on carefully crafted prompts. Further research aims to rectify these limitations.

Detailed Method:

The Receptive Weight Key Value (RWKV) is a unique architecture model primarily composed of four key elements: Receptance, Weight, Key, and Value.

In essence, the RWKV architecture is a series of stacked residual blocks made up of time-mixing and channel-mixing sub-blocks with recurrent structures. It offers a blend of both Transformer and Recurrent Neural Network (RNN) characteristics, striking a balance between efficiency and the capacity to handle complex patterns in data.

Receptance acts as accepting past information, the Weight is a trainable model parameter denoting positional weight decay, the Key is a vector similar to traditional attention, and the Value is also a vector analogous to traditional attention. These interact with each other multiplicatively at every timestep (as shown in Figure 2).

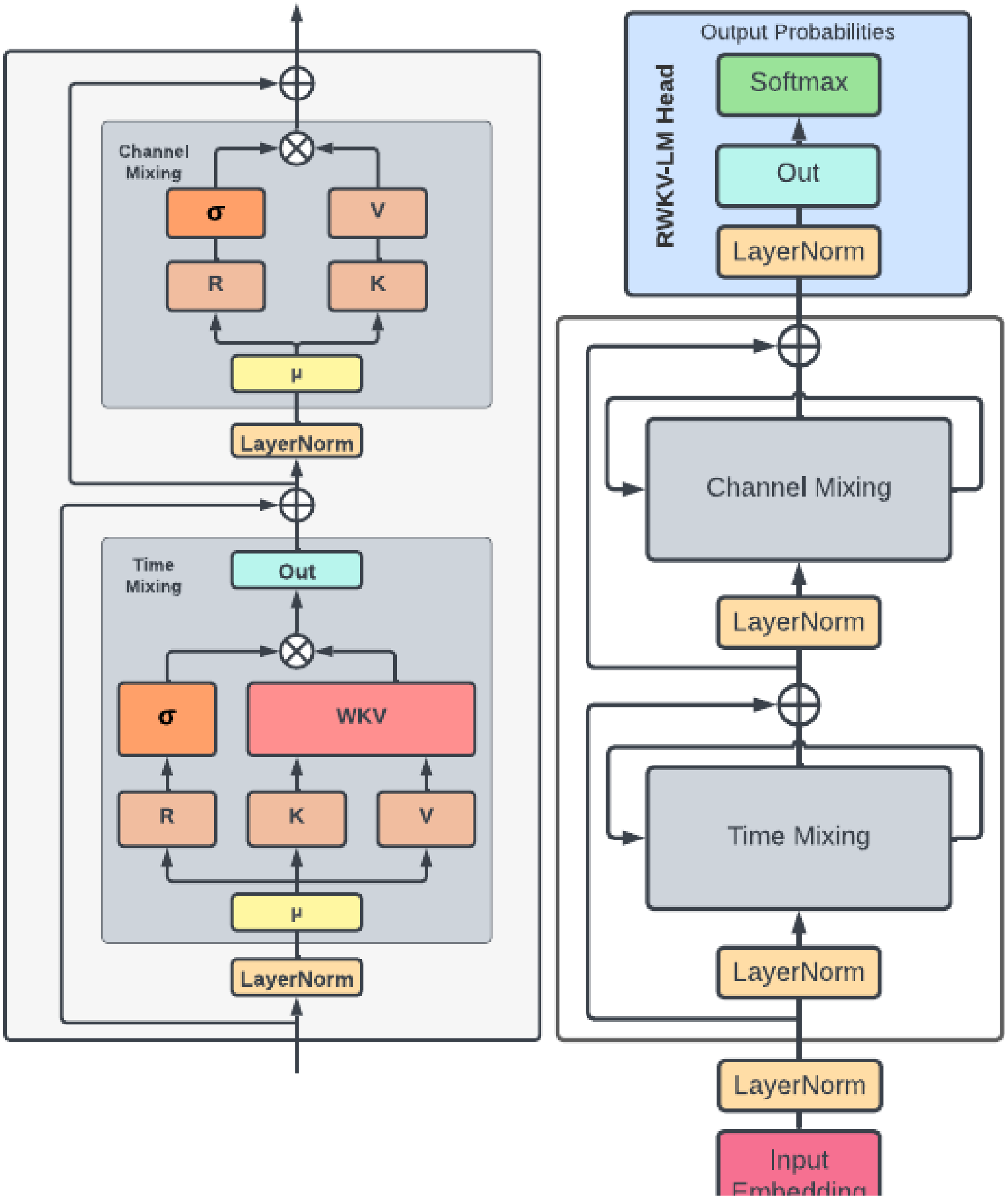

Figure 2: RWKV block elements (left) and RWKV residual block with a final head for language modeling (right) architectures.

.

A notable feature here is time-shift mixing or token shift which appears in the form of diagonal lines in Figure 3.

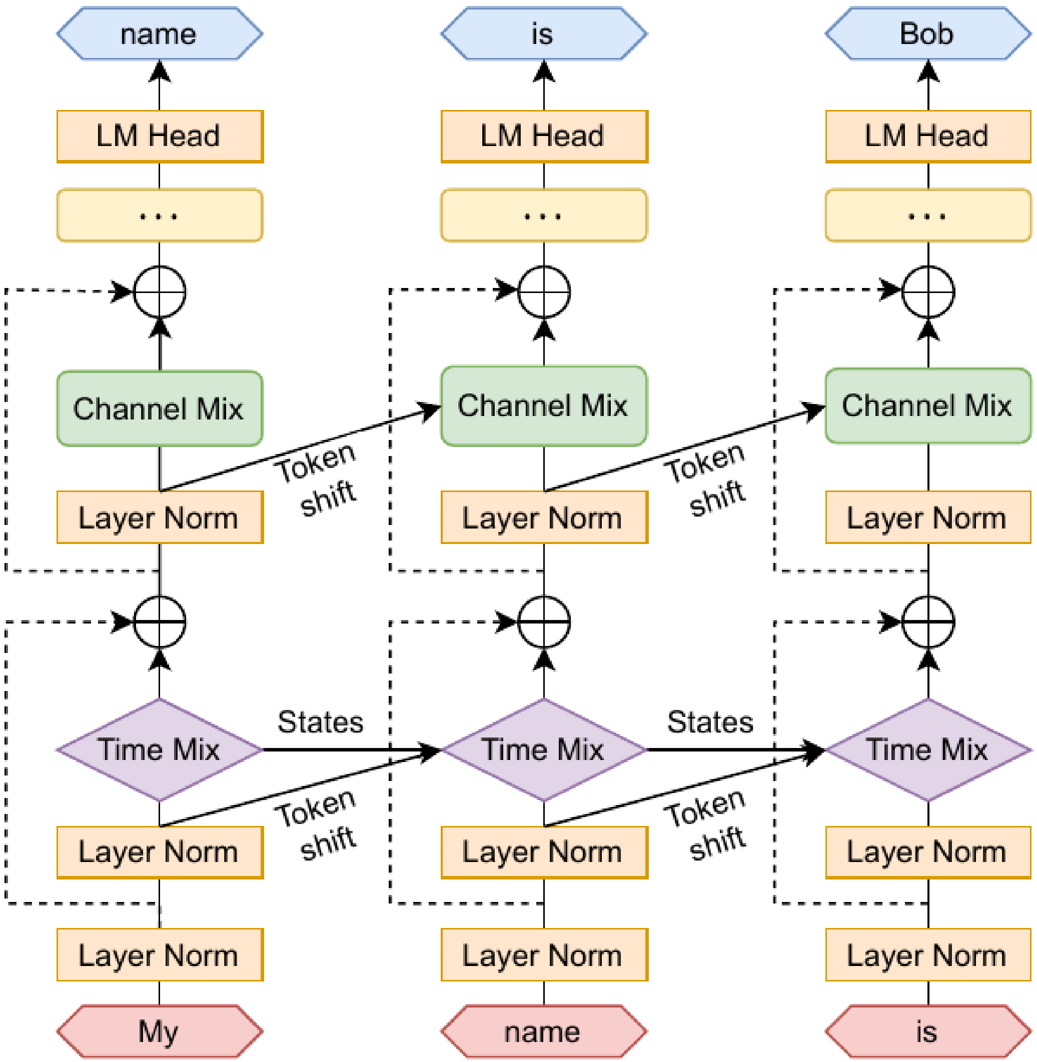

Figure 3: RWKV architecture for language modelling.

. It is a technique that the model uses to linearly interpolate between the current input and the input of the previous time step, allowing for the adjustment of every linear projection of input embedding.

Another essential component is the time-dependent update of the WKV, formulated in equation 14. This computation is similar to AFT (Zhai et al., 2021), but the Weight is now a channel-wise vector multiplied by relative position instead of a pairwise matrix in AFT.

RWKV architecture offers superior performance in language modeling scenarios as seen in Figure 3. Here, the model's time-mixing and channel-mixing blocks use the sigmoid of the receptance to "forget" or eliminate unnecessary historical information.

One of the major advantages of the RWKV structure is its ability to parallelize efficiently in what is termed time-parallel mode, similar to Transformers. This improves the processing time of a batch of sequences, allowing for faster computations.

However, unlike traditional self-attention models, which require a KV cache that grows linearly with the sequence length, the RWKV model functions like an RNN. This attribute enables it to process longer sequences efficiently without degrading performance due to increasing memory footprint or time as the sequence lengthens - a significant benefit especially evident in autoregressive decoding inference of language models.

On a software level, the RWKV model has been implemented using the PyTorch Deep Learning library. While designed as a general recurrent network, the current implementation focuses primarily on language modeling tasks; however, it has the potential for application in a range of NLP tasks.

Finally, one of the key design aspects of the RWKV model is its inherent safeguard against vanishing gradients thanks to its numerical stability and gradient propagation along the most relevant path. This, along with layer normalization and the single-step process for updating attention, enables RWKV to maintain stability, improve learning capabilities and stack multiple layers, surpassing traditional RNNs in capturing more complex data patterns.