**FNet: Mixing Tokens with Fourier Transforms**

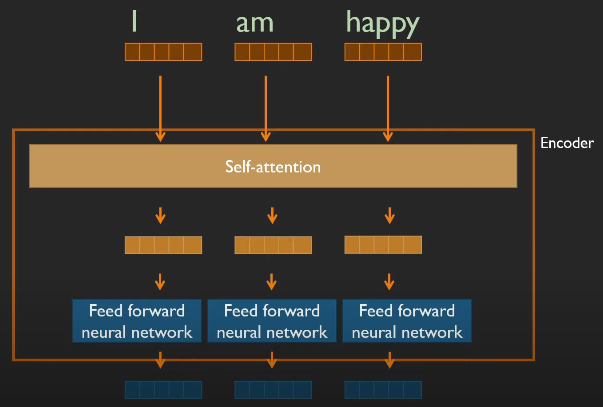
**Why Fourier?**

By using Fourier transform, the embedding gets converted into the frequency domain (decompose a signal into its constituent frequencies). Each output from Fourier transform has a component of each of the input embedding. Thus, facilitating the information across all embeddings. So the Fourier transforms act as an effective mechanism for mixing tokens.

## Why not self-attention layers?

Self-attention blocks result in higher accuracy, and provide useful contextual meaning to the network, they are computationally expensive. Self-attention is quite powerful, but it is essentially becoming a bottleneck and there is a race to replace it without affecting performance.

We can also say that, The Transformer is composed of self-attention, followed by fully connected layers along with residual connections and layer normalizations. This had been proven to be extremely successful on multiple data modalities. It started with text, became mainstream on both image and text, then on the image only and then it moved on to speech and even further than that.



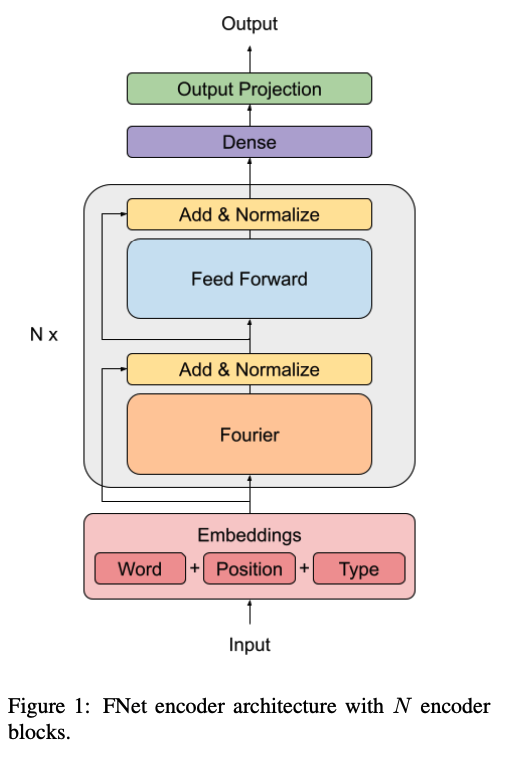
Transformer architecture

But the powerful as it is, the transformer has a problem: The self-attention mechanism scales quadratically in time and memory with increasing sequence length. In other words, there is a limit to what the length of your text might be.

## FNet architecture

FNet take the classical transformer block and modify it by replacing the self-attention layer containing learnable parameters with an unparameterized, thus not a learnt transformation, being the Fourier transform. Only learnable parameters of the whole block are the weights of the feedforward network.

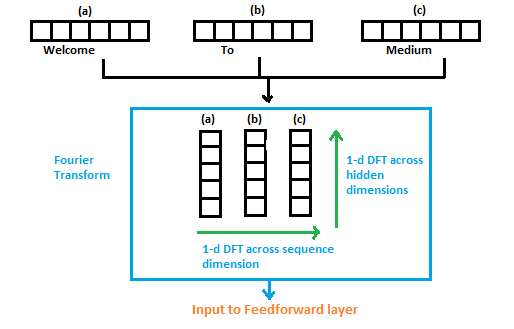
We can also say that difference between this architecture and the classic architecture is that the self-attention layer of the encoder is replaced with a Fourier layer [1] that applies a 2D Fourier transformation to the input sequence length and the hidden dimension



Fourier transform block applies a 2D DFT to its embedding input — one 1D DFT along the sequence dimension and one 1D DFT along the hidden dimension.

Image showing Fourier transform block equation

Operation happening inside Fourier block



DFT operations on input embeddings (Source: Author)

The output of the Fourier transform is in the frequency domain. As we know the convolution in the time domain is equivalent to multiplication in the frequency domain. The feedforward network is performing kind of convolution operations on the embeddings.

Then for the next block, the frequency domain embeddings go as input and get converted into time-domain again by Fourier transform. This conversion from time-domain to frequency-domain and vice-versa happens N times. N is the number of blocks in a transformer. Also, it can be thought of as alternating between multiplication and convolutions.