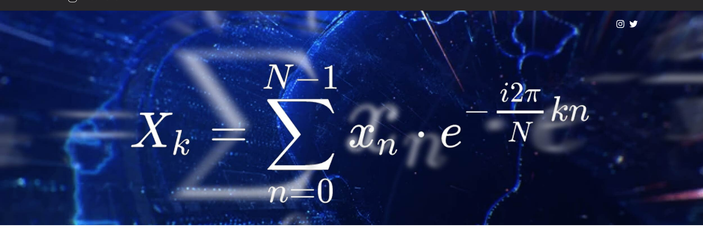


Attention: Fourier Transforms. A giant leap in transformer efficiency.

[7 minute read]

In this article we will discuss a promising neural network architecture, FNet, proposed recently by Google research. FNet is a interesting approach to neural network design, in that it replaces the traditional attention mechanism in transformer networks with a novel operation: the Fourier transform (FT).

Google AI researchers Lee-Thorpe et al. claim that FNet can be trained up to 80% faster than analogous BERT models, while only sacrificing 3–8% in accuracy measured on the GLUE benchmark.

This work is of particular interest to us at Optalysys because we are currently developing the world's fastest and most efficient FT core using optical processing. Optical computing can accelerate the FT layers of these networks by several orders of magnitude, meaning that these already-efficient networks could be further optimised.

The contents of the article are as follows:

Attention

Transformer networks utilise the attention mechanism, which is calculated using a series of linear transformations. Attention-based approaches have proven themselves to be exceptionally useful for problems that feature global interaction. For example, the recent AlphaFold 2 network that cracked the protein folding problem makes extensive use of attention.

The aim of the attention mechanism is to model global interactions between tokens in a sequence. Attention is a sequence to sequence transformation with inputs and outputs represented as matrices. At the first layer in a typical transformer network, raw inputs are converted into matrix representations via an embedding function. For instance, this could convert a sentence, a sequence of word tokens, into a matrix where each row is a vector representation of a word.

To calculate the output matrix of the attention layer, the input matrix X is multiplied with the key K, query Q and value V matrices. A scalar normalisation factor n is added to stabilise gradients:

$$attention(X, Q, K, V, n) = softmax\left(\frac{XQ \cdot (XK)^T}{n}\right) XV$$

The above equation is the key to a transformer's representative power. Matrices are used to transform the *entire* input sequence, so attention is able to model global interactions in data.

The work on FNets presents us with an insightful formalism to describe attention, as a case of *parameterised token mixing*.

- Attention is parameterised because it is calculated via application of learnable parameters in each matrix.
- Attention mixes tokens because each vector in the output matrix contains a representation of each input token in the context of all tokens in the sequence.

Fourier Transforms

The FT is a highly useful mathematical function that is used extensively in fields as diverse as signals processing, differential equation solving, and quantum mechanics.

We have previously discussed specific applications of the FT, as well as the Optalysys approach to accelerating the function in articles such as:

- *Optalysys: What we do (And why we do it)*
- Fourier-optical computing for AI: Acceleration, squared
- Deep learning and fluid dynamics: Solving the Navier-Stokes equations with an optical neural network

The FT is a mapping between two different ways of representing a function. Specifically the FT decomposes a function of either time or space into a function of temporal or spatial frequency. For instance we could retrieve the notes (frequencies) that make up a musical chord by applying the FT to an audio signal (which is an amplitude that oscillates as a function of time). The inverse FT (IFT) can be used to reverse this process, mapping a frequency domain signal back into the temporal or spatial domain.

The mathematical definition of the discrete FT (DFT) is as follows:

- The input signal is denoted x, this may be spatial or temporal, for instance a sound wave or an image.
- The output of the DFT is a complex valued vector X.
- Each element of X encodes a frequency term, and is obtained by multiplying every element of x with a complex exponential then summing up these products.

$$X_k = \sum_{n=0}^{N-1} x_n \cdot e^{-\frac{i2\pi}{N}kn}$$

The relevant observation, at least for this article, is that each element in the FT of a function depends on all elements of the input.

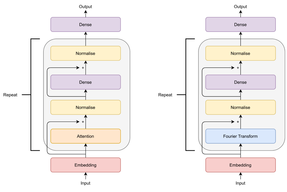
We can therefore view the FT as non-parameterised token mixing.

- It is token mixing because all elements in the input function contribute to each element in the output function.
- It is not parameterised because the output only depends on the input, there are no other parameters such as the matrices used in attention.

FNets use FTs as a way of mixing information so that every element of an input influences every other element, thus achieving the same kind of global interaction that the more expensive (see next section) attention mechanism provides.

Fourier Transforms as a replacement for attention

FTs and attention mechanisms have clear similarities when viewed from the perspective of token mixing. This is the motivation behind exploring the possibility of replacing attention layers with FTs. Lee-Thorpe et al. proposed the following modification to the transformer block used in networks like BERT to create FNet:



Left: standard transformer architecture. Right: FNet architecture.

All else being equal, there are two computational advantages to FTs vs. attention.

1. FTs can be calculated on graphics processing units (GPUs) fairly efficiently thanks to the Fast Fourier transform (FFT) algorithm. This allows the quadratic algorithm written above to run in $O(n \, log(n))$ time using a divide-and-conquer approach. Attention cannot be accelerated in this way and thus has undesirable scaling properties of the order$O(n^2)$.
2. FTs are not parameterised, meaning that we can reduce a model's memory footprint by replacing attention with FTs, as we do not need to store key, query and value matrices.

Results

The researchers reported that FNet trained faster (80% on GPUs, 70% on TPUs), and with much higher stability than BERT. The peak performance as measured by accuracy on the GLUE benchmark was 92% as good as BERT.

They also proposed a hybrid architecture that replaced all but the last two attention layers in BERT with FT layers, which reached 97% relative accuracy, with only a minor penalty in training time and stability. This is a highly encouraging result, seemingly the undesirable scaling properties of transformers can be avoided if we employ more tactful approaches to token mixing. This is especially important for resource constrained environments.

How optical FTs can change the game

Optalysys have created the world's most efficient FT core, based on an optical process. It not only reduces the algorithmic complexity of the FT from the FFT's *O(n log(n)) scaling* to constant time: O(1), it also can be achieved with a fraction of the power consumption of a digital electronic circuit. When compared to the most efficient GPU implementation we have seen: the Nvidia A100, the optical approach is two orders of magnitude better.

We ran some experiments on the Huggingface implementation of FNet and found that on an Nvidia Quadro P6000 GPU, the FT was responsible for up to 30% of the inference time on the FNet architecture. Taking this into account, any GPU that integrates optical FT functionality would receive an instant speedup/efficiency gain on these networks of around 40%... and this is on an architecture that has been designed with traditional GPUs in mind.

Interestingly, Google's TPU was not able to run FNet as efficiently as a GPU (relative to BERT). The GPU trained FNet 80% faster than BERT, the TPU's, improvement was 70%. This is because the TPU is not at all optimised for FTs.

The TPU's primary objective is to accelerate multiply and accumulate operations (MACs) used for matrix multiplication. The TPU is so inefficient at FTs that the researchers did not use the FFT algorithm on sequences < 4096 elements, instead opting for a quadratic-scaling FT implementation using a pre-computed DFT matrix. If the next generation TPU was to integrate something like the technology we are developing at Optalysys, then its FT processing efficiency could be raised by a factor of 1000.

Optalysys vision and Beta program

We see a huge potential to combine the strengths of free-space optical computing with most existing AI accelerators. Indeed, optical hardware can work in tandem with a whole range of electronic processors, providing faster and more efficient processing solutions.

In order to realise this ambition, we are interested in working with third parties to create next-generation AI and encryption systems leveraging the optical FT. Access to the Optalysys optical systems is not yet open to everyone, but for those interested, please contact us at optalysys.com for access to the beta program for bench-marking and evaluation.

Telephone: +44 (0) 1977 551615 Email: info@optalysys.com