# Modern AI as the compositional approach to function approximation

*Zehua Lai and Lek-Heng Lim*

The singular value decomposition of a linear operator $F$ takes two well-known forms: As a *sum* of separable linear operators

$$F = \sigma_1 u_1 \otimes v_1 + \sigma_2 u_2 \otimes v_2 + \cdots + \sigma_r u_r \otimes v_r \tag{1}$$

or a *composition* of three structured linear operators

$$F = U \Sigma V^*.$$

These are of course equivalent for linear operators but when $F$ is a nonlinear operator, an additive model

$$F = f_1 + f_2 + \cdots + f_r \tag{2}$$

and a compositional model

$$F = F_1 \circ F_2 \circ \cdots \circ F_s \tag{3}$$

would produce vastly different results. The traditional approach taken towards the approximation, interpolation, or regression of a target function in approximation theory, harmonic analysis, machine learning, signal processing, statistics, etc, are mainly variations of the additive model in (2). Nevertheless the deep neural network revolution and transformer revolution have now brought the composition model (3) to the fore.

If selected, our tutorial will discuss why the many admirable innovations with the additive model (2) such as

- compressed sensing: ensuring $r$ is small through sparsity or low-rank;
- kernel methods: linearizing (2) to (1) through embedding in feature space;
- tensor networks: imposing separability on $f_i$'s and graph structures on the sum in (2);
- wavelets: exploiting multiscale and localization properties of $F$ in (2);

and more, have all fallen short to overcome the curse-of-dimensionality. We will then discuss how the compositional model (3) provides a remarkably effective way to alleviate the curse.

We hold the opinion that this simple idea of replacing sums by compositions is, more than anything else, the key to the phenomenal success of recent AI models. While we have alluded to this briefly in [3], in the proposed tutorial we will elaborate in greater detail and furnish more examples to make this point.

The example of splines serves as a fitting illustration. Traditionally, a vector-valued spline $F : \mathbb{R}^n \to \mathbb{R}^m$ of degree $d$ is regarded simply as

$$F = \xi_1 \mathbb{I}_{A_1} + \xi_2 \mathbb{I}_{A_2} + \cdots + \xi_r \mathbb{I}_{A_r}, \tag{4}$$

where $\xi_1, \dots, \xi_r \in \mathbb{R}[x_1, \dots, x_n]$ are polynomials of degree $d$ and

$$\mathbb{R}^n = A_1 \cup \cdots \cup A_r \tag{5}$$

a partition of the domain (here $\mathbb{I}_A$ is the indicator function on $A$). This is the picture of splines in the additive model (2).

In the case of linear splines, i.e., $d = 1$, the work of Arora et al [1] established that a ReLU-activated $l$-layer feed forward neural network is nothing more than a picture of splines in the compositional model (3):

$$F(x) = A_{l+1}\sigma_l A_l \cdots \sigma_2 A_2 \sigma_1 A_1 x \tag{6}$$

for any input $x \in \mathbb{R}^n$, weight matrix $A_i \in \mathbb{R}^{n_i \times n_{i-1}}$, $n_0 = n$ and $n_{l+1} = m$,

$$\sigma_i(x) \coloneqq \mathrm{ReLU}(x + b_i)$$

with bias vector $b_i \in \mathbb{R}^{n_i}$. The activation $\mathrm{ReLU}(x) \coloneqq \max(x, 0)$ is always applied coordinatewise when its input is a vector or a matrix. In other words, for $d = 1$, neural networks are exactly linear splines and vice versa.

We extended this work to splines of higher degree in [3]. Since neural networks are linear splines, compositions of neural networks just give neural networks with more layers and they remain linear splines. So one needs a new element in order to obtain splines of higher degree. We showed that the attention module in (7) fulfills such a role and is all one needs to generate splines of arbitrarily high degrees as compositions. In other words, for higher values of $d$, transformers are exactly splines of degree $d$ and vice versa with one caveat.

A ReLU-activated *attention module* [4] is a map between matrix spaces defined by $\alpha : \mathbb{R}^{n\times p} \to \mathbb{R}^{m\times p}$,

$$\alpha(X) \coloneqq V(X)\,\mathrm{ReLU}\big(K(X)^{\mathsf{T}}Q(X)\big), \tag{7}$$

where $Q : \mathbb{R}^{n\times p} \to \mathbb{R}^{d\times p}$, $K : \mathbb{R}^{n\times p} \to \mathbb{R}^{d\times p}$, $V : \mathbb{R}^{n\times p} \to \mathbb{R}^{m\times p}$ are given by affine maps

$$Q(X) = A_Q X + B_Q, \qquad K(X) = A_K X + B_K, \qquad V(X) = A_V X + B_V,$$

with weight matrices $A_Q, A_K \in \mathbb{R}^{d\times n}$, $A_V \in \mathbb{R}^{m\times n}$, and bias matrices $B_Q, B_K \in \mathbb{R}^{d\times p}$, $B_V \in \mathbb{R}^{m\times p}$.

It is not difficult to see that $\alpha$ is a cubic spline, in the sense of a function defined piecewise by cubic polynomials (we will describe how to obtain differentiability in this context later). Let $\varphi$ denote a neural network (6) applied to a matrix $X \in \mathbb{R}^{n\times p}$ columnwise to each of the $p$ columns of $X$. So $\varphi : \mathbb{R}^{n\times p} \to \mathbb{R}^{m\times p}$. We showed in [3, Section 3.1] that when attention modules are alternately composed with neural networks as in the *encoder* of a transformer

$$\varepsilon_t = \varphi_t \circ \alpha_t \circ \varphi_{t-1} \circ \alpha_{t-1} \circ \cdots \circ \varphi_1 \circ \alpha_1,$$

we obtain a spline of degree $3^t$. Furthermore, the same applies to the *decoder* of a transformer

$$\delta_s = \varphi_s \circ \beta_s \circ \varphi_{s-1} \circ \beta_{s-1} \circ \cdots \circ \varphi_1 \circ \beta_1,$$

where the attention modules $\alpha_i$'s are replaced by so-called *masked* attention modules $\beta_i$'s, which are essentially "upper triangular" analogs of attention modules. Finally, a transformer constructed out of a $t$-layer encoder and $s$-layer decoder is then a spline of degree $3^{t+s} + 3^t - 3^s$. We also showed that the converse — every spline can be represented as a transformer — holds true if and only if the Pierce–Birkhoff conjecture holds true [3, Section 3.3]; and we recently proved enough of this conjecture to show that as long as the partition in (5) is piecewise linear (i.e., $A_i$'s are polyhedra), then a spline of any degree on such a partition is a transformer.

The observations in the previous paragraph lead to several insights:

(i) A transformer is a spline $F$ written in the compositional form (3).

(ii) Expressing a spline as a composition affords a straightforward way to impose differentiability to any desired degree, namely, by replacing the $C^0$ activation ReLU with any $C^k$ activation. Indeed, if we use the $C^\infty$ activation SoftMax as in the original paper [4] where the transformer architecture was proposed, we obtain a "smoothed spline," impossible in the traditional way of constructing a spline.

(iii) Most importantly, this compositional form allows one to generate splines of exceeding high degrees that cannot be realistically represented in the traditional additive form (4) because of curse-of-dimensionality. For example, even the most rudimentary transformer in [4] has $s = t = 6$, which produces a spline of degree 531,441. The transformer in OpenAI's GPT likely uses far higher values of $s$ and $t$. The earliest version of Google's BERT [2] involves an encoder with $s = 24$, which produces a spline of degree exceeding 280 billion.

It is safe to say that splines of such insanely high degree have never before been used in any application before the appearance of transformers. We posit that therein lies the real novelty of the transformer technology.

If selected, our tutorial will also survey more recent works where we applied the same insights to functions traditionally written in an additive form (2) with atoms $f_i$'s such as wavelets or sinusoids to construct transformer-like models by expressing them in a compositional form (3).

## References


[1] R. Arora, A. Basu, P. Mianjy, and A. Mukherjee, "Understanding deep neural networks with rectified linear units," in *International Conference on Learning Representations*, 2018.

[2] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," in *Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 4171–4186, 2019.

[3] Z. Lai, L.-H. Lim, and Y. Liu, "Attention is a smoothed cubic spline," `arXiv:2408.09624`, 2024.

[4] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, volume 30, 2017.