CBMM Memo No. 143 **September 18, 2023**

A Homogeneous Transformer Architecture

Yulu Gan
and
Tomaso Poggio
Center for Brains, Minds, and Machines, MIT, Cambridge, MA, USA

Abstract

While the Transformer architecture has made a substantial impact in the field of machine learning, it is unclear what purpose each component serves in the overall architecture. Heterogeneous nonlinear circuits such as multi-layer RELU networks are interleaved with layers of soft-max units. We introduce here a homogeneous architecture based on Hyper Radial Basis Function (HyperBF) units. Evaluations on CIFAR10, CIFAR100, and Tiny ImageNet demonstrate a performance comparable to standard vision transformers.

This material is based upon work supported by the Center for Brains, Minds and Machines (CBMM), funded by NSF STC award CCF-1231216.

1 Introduction

Why are different types of operations such as soft-max and RELU necessary in the Transformer [1] architecture? While conventional explanations highlight the self-attention's role in weighting sequence positions and the feedforward network's role in information transformation [1], an alternative interpretation suggests that self-attention discerns the sparse graph structure of the compositional regression function [2]. Viewed this way, the subsequent MLP learns the constituent function linking two nodes in the graph. The theoretical results on compositional sparsity suggest that learning of a sparse compositional function could be replicated using nonlinearities that are not RELUs since they can be used to approximate non-linear functions. A classical example consists of Radial Basis Functions (RBF) and its variant, HyperBF. Intriguingly, under a normalization assumption HyperBF networks are exactly equivalent to self-attention based on the soft-max. This insight suggests a unified architecture composed of HyperBFs units only.

2 Method

Our proposed unified architecture can be written as:

$$y = x + \mathrm{HyperBF}(\ \mathrm{LayerNorm}\ (x + \mathrm{HyperBF}(\mathrm{LayerNorm}(x))) \tag{1}$$

A succinct overview of HyperBF is presented in Sec. 2.1. Following this, the role of HyperBF in each layer is detailed in Sec. 2.2.1 and Sec. 2.2.2. Specifically, in Sec. 2.2.1, we delve into the relationship between HyperBF and the self-attention mechanism, drawing parallels with associative memory where both the query and key originate from the data itself. Afterwards, in Sec. 2.2.2, we offer insights into how HyperBF functions in the second layer, acting as associative memory with keys derived from learnable centers.

2.1 Preliminary

2.1.1 Normalized RBF

The Normalized Radial Basis Function, abbreviated as Normalized RBF, is characterized by the following equation:

$$f(\mathrm{x}) = \frac{\sum_{i=1}^{N} y_i K\left(\|\mathrm{x} - \mathrm{x}_i\|\right)}{\sum_{i=1}^{N} K\left(\|\mathrm{x} - \mathrm{x}_i)\ \|\right.} \tag{2}$$

Distinct from the traditional Radial Basis Functions, the Normalized RBF incorporates a normalization factor in its denominator. This factor serves as an approximation of the data's probability distribution. The relationship between this approximation method and regularization theory is well-established (see [3]).

$$f(\mathrm{x}) = \sum_{i=1}^{N} y_i K\left(\|\mathrm{x} - \mathrm{x}_i\|\right). \tag{3}$$

Often referred to as kernel regression or the Nadaraya-Watson estimator, this approximation technique has garnered significant attention within the statistical realm. The equation embodies the quintessential structure of normalized radial basis functions. Here, the centers align with the examples, and the coefficients c_i equate to the function values y_i at the respective data points $\mathbf{x}_i$. It's worth noting that this estimator bears similarities to Parzen windows.

2.1.2 HyperBF

HyperBF [4, 5] is a generalization of RBF networks, where the Mahalanobis-like distance is used instead of the Euclidean distance.

$$\phi(\boldsymbol{x}) = \sum_{\alpha=1}^{N} c_\alpha K\left(\|\boldsymbol{x} - \boldsymbol{t}_\alpha\|_W^2\right) \tag{4}$$

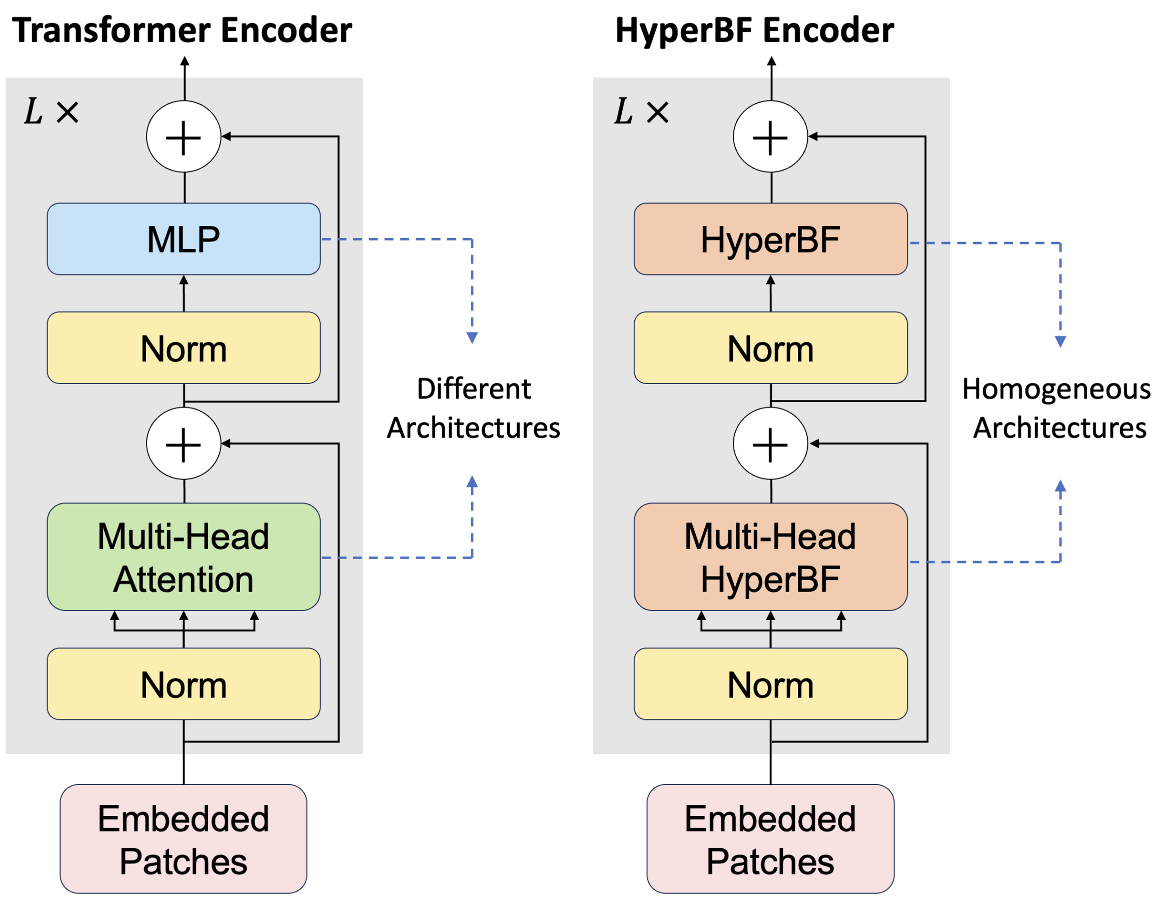

Figure 1: **Our architecture vs. Transformer.** The components of our model integrate seamlessly with HyperBF, forming an architecture reminiscent of a unified associative memory.

Note that t_α and c_α are the center and weight of neuron α. The activation function $K\left(\|\boldsymbol{x}-\boldsymbol{t}_\alpha\|_W^2\right)$ represents a Radial Basis Function such as the Gaussian[1] defined as $K(\cdot)=\exp\left(-\frac{\|x-z\|_W^2}{L}\right)$. In this context, the Mahalanobis-like distance is given by the expression $\|x-z\|_W^2=(x-z_i)^T W^T W (x-z_i)$.

2.2 The Architecture of HyperBF

We have introduced a multilayer architecture consisting of HyperBF units. Our structure can be better comprehended when viewed through the lens of similarity.

2.2.1 Evaluating Sample-to-Sample Similarity in the First Layer

The attention mechanism has been widely used in many sequence modeling tasks. Its dot-product variant is the key building block for the state-of-the-art transformer architectures [1]. Let $\mathbf{q}_t$ denote a query vector, that attends to sequences of L pairs $\mathbf{k}_i, \mathbf{v}_i$ of key and value vectors (see Figure 2). At each timestep, the attention linearly combines the values weighted by the outputs of a Softmax:

$$\operatorname{attn}\left(\mathbf{q}_t, \{\mathbf{k}_i\}, \{\mathbf{v}_i\}\right)=\sum_i \frac{\exp\left(\mathbf{q}_t \cdot \mathbf{k}_i/\sigma^2\right)}{\sum_j \exp\left(\mathbf{q}_t \cdot \mathbf{k}_j/\sigma^2\right)} \mathbf{v}_i^\top \tag{5}$$

As observed in[2], a normalized HyperBF unit is equivalent to soft-max based self attention. A normalized HyperBF requires the assumption that $\|\mathbf{q}_t\|=1$ and $\|\mathbf{k}_i\|=1$. Then $1-\frac{\|\mathbf{q}_t-\mathbf{k}_i\|^2}{2}=\mathbf{q}_t \cdot \mathbf{k}_i$.

[1]There are several other choices including the Laplacian.

In practice, the assumptions that $\|\mathbf{q}_t\| = 1$ and $\|\mathbf{k}_i\| = 1$ in the self-attention layer are not always met in transformers in common use. However, we can remove this assumption (see details in Appendix 6.1) so the equivalence between self-attention and HyperBF can be more general.

$$\operatorname{attn}\left(\mathbf{q}_t, \{\mathbf{k}_i\}, \{\mathbf{v}_i\}\right) = \sum_i \frac{\exp\left(-\frac{\|\mathbf{q}_t - \mathbf{k}_i\|^2}{2\sigma^2}\right)}{\sum_j \exp\left(-\frac{\|\mathbf{q}_t - \mathbf{k}_j\|^2}{2\sigma^2}\right)} \mathbf{v}_i^\top \tag{6}$$

2.2.2 Assessing Similarity Between Centers and Samples in the Second Layer

We can rewrite the Eq. 4 of HyperBF as follows:

$$\phi(\mathrm{x}) = \sum_{i=1}^{N} K\left(\|\mathbf{q}_i - \mathbf{k}_i\|_W^2\right) \mathbf{v}_i^\top \tag{7}$$

where $\mathbf{q}_i$ is the output of the first layer and $\mathbf{k}_i$ is learnable centers. The second layer of the network also has an associate memory structure. Here, the distance between the output of the first layer and the centers is used as the weight.

Typically, the feedforward network in a transformer layer has the following form:

$$\phi(x) = W_2 \sigma\left(W_1 x + b_1\right) + b_2 \tag{8}$$

where W_1 is the weight of the first fully connected layer, b_1 is its bias, σ represents an activation function like ReLU, W_2 is the weight of the second fully connected layer, and b_2 is its bias.

3 Experiments

3.1 Experimental Setup

Dataset. As shown in Tab. 1, CIFAR10 [6] consists of 60,000 32x32 color images, divided into 10 classes, with 6,000 images per class. The 10 classes are airplane, automobile, bird, cat, deer, dog, frog, horse, ship, and truck. The dataset is split into 50,000 training images and 10,000 test images.

CIFAR100 [6] is Similar in structure to CIFAR10, the CIFAR100 dataset contains 60,000 32x32 color images. However, it is divided into 100 classes, with each class containing 600 images. These 100 classes are further grouped into 20 superclasses, providing a hierarchical structure to the dataset.

Tiny ImageNet [7] is a scaled-down version of the renowned ImageNet [8], featuring 100,000 images of 64x64 resolution across 200 classes. Designed for educational use and quick experimentation, it offers a balance between computational feasibility and real-world vision challenges.

Table 1: Vision task statistics and descriptions.

Dataset	Train	Test	Classes	Metric	Domain
CIFAR10	50K	10K	10	Acc.	32×32
CIFAR100	50K	10K	100	Acc.	32×32
Tiny ImageNet	100K	10K	200	Acc.	64×64

3.1.1 Implementation Details.

We implement our code based on Pytorch [2]. We train our multilayer HyperBF and all baselines for 100 epochs on 1 NVIDIA A100 GPUs. The training involves images with a resolution of 32×32 and incorporates data augmentation including filp with a batch size of 256. The proposed model is trained with a learning rate 10^{-4} without any warm-up stage. Both the Vision Transformer and our Multilayer HyperBF utilize 4 heads and 4 blocks. For a fair comparison, we maintain consistent parameters across both models.

[2]Our code will be available at https://github.com/sunrainyg/Unified_architecture

Table 2: Experimental results on several benchmark datasets. For all of them, we train the model on the training dataset and then test on the test dataset using accuracy as the meric. Results show that the multilayer HyperBF network achieves almost the same performance as transformers.

Dataset	Epoch	Method	Acc (%)
CIFAR10	100	ViT	75.61
		HyperBF	75.12
CIFAR100		ViT	49.90
		HyperBF	48.30
Tiny ImageNet		ViT	32.03
		HyperBF	31.14

3.1.2 Experimental Results

We tested the performance of our model on CIFAR10, CIFAR100, and Tiny Imagenet. On these three datasets, our model underperformed the Vision Transformer by 0.49%, 1.6%, and 0.89% respectively. We believe this is within an acceptable range, and our model's performance is comparable to that of the Vision Transformer but more homogeneous. Additionally, there are methods to enhance our model's performance (as detailed in Appendix 6.3). However, these are beyond the scope of this paper's objectives.

4 Conclusions

The experiments in this paper suggest an alternative Transformer architecture based on Hyperbf units. The architecture is homogeneous since the same machinery is used in the self-attention layers and in the "MLP" layers. Among our preliminary observations we mention:

- for vision transformers HyperBF units can be assumed to have $M \approx \frac{I}{\sigma^2}$;
- the self-attention units learn automatically a smaller σ than the units replacing the MLP layers.

We will discuss in later publications how to leverage the new homogeneous architecture for

- efficient and different optimization techniques;
- a better understanding of the principles underlying transformer's properties, including connections to ideas about how the brain might work, see [9];
- studying its characteristics on language datasets;
- the connection between these networks and associative memories.

5 Acknowledgement

We are deeply grateful to Brian Cheung and Mengjia Xu for the discussions and proofreading.

References

[1] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[2] Tomaso A. Poggio. How deep sparse networks avoid the curse of dimensionality: Efficiently computable functions are compositionally sparse. *CBMM Memo*, 10/2022 2022.

[3] Federico Girosi, Michael Jones, and Tomaso Poggio. Regularization Theory and Neural Networks Architectures. *Neural Computation*, 7(2):219–269, 03 1995.

[4] T. Poggio and F. Girosi. Networks for approximation and learning. *Proceedings of the IEEE*, 78(9):1481–1497, 1990.

[5] Roberto Brunelli and Tomaso Poggio. Hyberbf networks for gender classification. *DARPA Image Understanding Workshop*, 02 1995.

[6] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[7] Ya Le and Xuan S. Yang. Tiny imagenet visual recognition challenge. 2015.

[8] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.

[9] T Poggio. A theory of how the brain might work. *Cold Spring Harb Symp Quant Biol*, 1990.

6 Appendix

6.1 General Case of Equivalence between Self-Attention and HyperBF.

Let's start from revisit the definitions: $\operatorname{Softmax}(x_i) = \frac{\exp(x_i)}{\sum_j \exp(x_j)}$. Where x_i is the i-th element of the input vector. Considering the characteristics of Softmax, it normalizes the input vector such that the sum of all elements in the output vector equals 1. Therefore, for any constant offset in the input vector, the output of the Softmax will not change. To demonstrate this, we can consider adding a constant c to the input vector. We can have:

$$\operatorname{Softmax}(x_i + c) = \frac{\exp(c) \cdot \exp(x_i)}{\exp(c) \cdot \sum_j \exp(x_j)} \tag{9}$$

As can be seen, $\exp(c)$ will be canceled out in both the numerator and the denominator, so adding the constant c will not change the output of the Softmax.

If we set $\|\mathbf{q}_t\| = c_1$ $\|\mathbf{k}_i\| = c_2$, we can have $c_1^2 + c_2^2 - \frac{\|\mathbf{q}_t - \mathbf{k}_i\|^2}{2} = \mathbf{q}_t \cdot \mathbf{k}_i$. When calculating the Softmax weights for attention, the constant term of $c_1^2 + c_2^2$ in this expression won't affect the outcome, as the Softmax will automatically normalize all the weights. Therefore, we can safely remove this constant term without affecting the results of the attention computation.

6.2 Additional information on the two layers.

We observed that $W^T W \approx I$. Therefore, special case of Hyperbf with the $W^T W$ in Eq. 4 as a identity that is RBF is acceptable for vision transformer HyperBF; σ are found to be different in the different layers depending on their function in the architecture

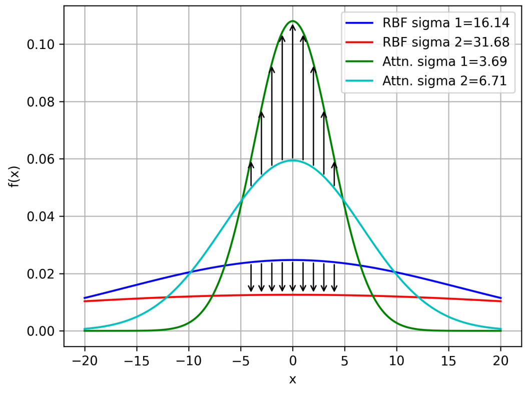

Figure 2: We visualized the Gaussian kernel functions in the first and second layers of HyperBF. Since we only want to focus on the size of σ, we assume $\mu = 0$ in this visualization.

6.3 Improvements of our model

There are several straightforward methods to enhance our model.

Firstly, we employ the same scale factor found in the self-attention layer. The scale factor $\sqrt{d}$ in this layer ensures that the inner product of $\mathbf{q}$ and $\mathbf{k}$ doesn't become excessively large. This is crucial because if the dimensions of $\mathbf{q}$ and $\mathbf{k}$ are too expansive, their inner product will consequently be substantial.

Secondly, a larger inner product indicates greater similarity, while a larger Euclidean or Mahalanobis distance suggests less similarity. Consequently, we introduced a negative sign before the Euclidean or Mahalanobis distance. However, the softmax function tends to be less responsive to negative numbers, leading to less pronounced differences in its output values. This aspect could potentially affect performance, suggesting an avenue for further refinement.

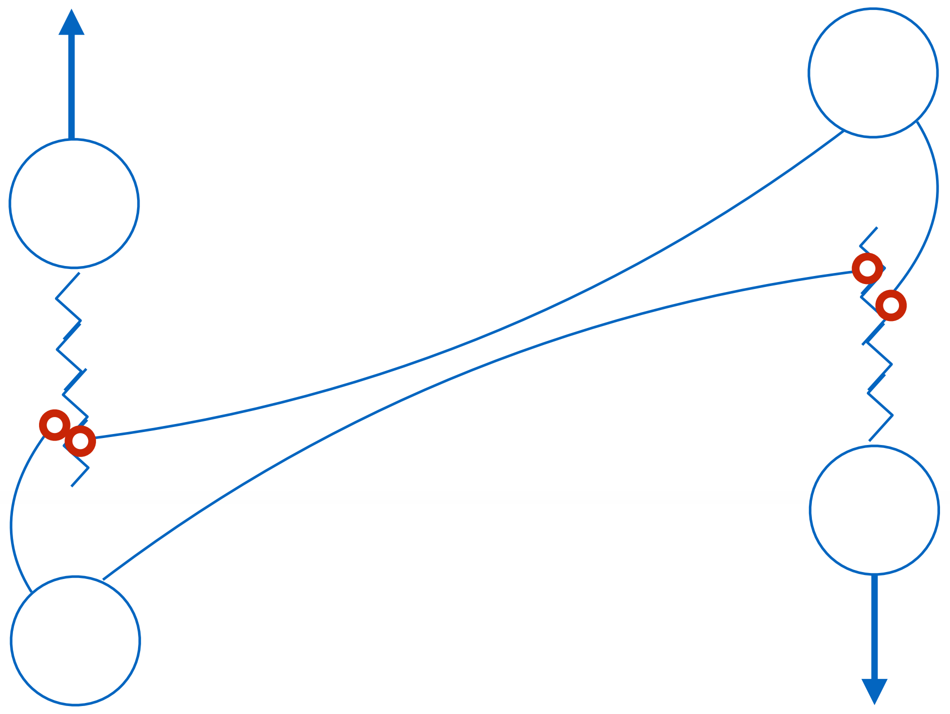

Figure 3: The key synaptic motif in the ascending and the descending streams, possibly repeating in the vertical and horizontal directions. Note symmetry.

7 New PseudoAppendix on work by Brian Cheung, Qianli Liao, Liu Ziyin, Yulu Gan and Tomaso Poggio

This section is a way to put a time stamp on new and ongoing work. We wish to report about a neural circuit which is a biologically plausible implementation of SGD.

7.1 Summary of proposed *NeuroSGD* circuit

Over the last four decades the amazing success of deep learning has been driven by a simple but powerful optimization technique, called Stochastic Gradient Descent (SGD). The default implementation of SGD is backpropagation, which is used to this day in essentially all computer implementations. From the perspective of neuroscience, however, it seems very unlikely that backpropagation could be used by the brain. Though several alternatives have been analyzed, none is supported by both computational and experimental evidence. Here we propose a SGD algorithm that is biologically plausible, works well and lead to experimentally verifiable predictions about a specific synaptic motif of connections between the ascending and the descending streams in cortex (see Figure 3). Perhaps the most interesting aspect of our proposal is a surprising self-assembly property of the basic circuit, using only heterosynaptic Hebb-type plasticity rules. We plan to communicate soon the details of our model that works as well as backpropagation on datasets such as CIFAR10.