DeepLABNet: End-to-end Learning of Deep Radial Basis Networks

Andrew Hryniowski[1,2,3]
Alexander Wong[1,2,3]
Email: {apphryniowski, a28wong}@uwaterloo.ca

[1] Vision and Image Processing Research Group
[2] Waterloo Artificial Intelligence Institute, University of Waterloo
[3] DarwinAI Corp.

Abstract

Radial basis function (RBF) networks provide an interesting mechanism for learning complex non-linear activation functions in a neural network. However, the interest in RBF networks has waned due to the difficulty of integrating RBFs into deep neural network architectures in a tractable and stable manner. In this work, we present a novel approach that enables end-to-end learning of deep RBF networks with fully learnable activation basis functions in a tractable manner. We demonstrate that our approach for enabling the use of learnable activation basis functions in deep neural networks, which we will refer to as DeepLABNet, is an effective tool for automated activation function learning within complex network architectures.

1 Method

RBF networks [1] have three main sets of parameters: i) basis function parameters, ii) the radial distance parameters, and iii) the kernel coefficients. Normally, the basis function parameters are predetermined, and the radial distance parameters are initialized either uniformly or to a sub-set of training samples. The kernel coefficients are then learned through linear optimization. These constraints make it difficult to use RBF networks in a hierarchical context. To solve these challenges, we introduce DeepLABNet, a novel approach that facilitates end-to-end learning of fully trainable deep RBF networks with fully learnable activation functions. Our proposed approach unlocks the power of RBF networks in a hierarchical environment by i) decoupling inter-channel dependencies within each sub-RBF network, ii) using polyharmonic radial basis functions, and iii) strategic regularization and initialization for improved stability during training.

Decoupling Inter-Channel Dependencies. RBF networks utilize *global* basis functions where each basis function is dependent on every input feature and affects every output feature. Such a relationship results in a high computational cost during training and inference. Furthermore, without a good initialization, a fully learnable RBF network will have difficulty learning and may be unstable, particularly for the case of deep RBF networks. To tackle this issue, we decouple the input features within an RBF, thus significantly reducing the computational cost during training and inference, allowing for better initialization strategies to be utilized and increased model stability during training, and allows additional basis functions to be utilized on a per-feature-basis. More specifically, DeepLABNet's feature-level RBF networks leverage the following design:

$$y = f(x) = \sum_{i=1}^{s} \lambda_i \phi(|x - c_i|) + v_0 x + v_1 \qquad (1)$$

where x and y are the input and output of any given feature-level RBF network, respectively. $\phi(\cdot)$ is the radial basis function, and c_i is the x coordinate of the i^{th} control point. v_0 and v_1 are scalar components added to the RBF network.

Polyharmonic Spline Radial Basis Function. Common practice when using RBF networks is to use Gaussian basis functions [1]. However, such networks tend to fail when input samples are too distant from a basis function centroid as all basis function output's approach zero. Another class of basis functions are the polyharmonic spline functions, which can be defined as

$$U_k(r) = \begin{cases} r^k, & \text{if } k \text{ is odd} \\ r^k \log r, & \text{if } k \text{ is even} \end{cases} \qquad (2)$$

where r is the output of some radial distance function (e.g., Euclidean), and k is the degree of the kernel. Note for all k's that $U_k(0) = 0$. Unlike the Gaussian basis function, polyharmonic basis functions implicitly handle outliers as inputs further from a basis centroid are given additional weighting.

Strategic Initialization and Regularization. Random initialization of an RBF network can result in a function whose gradients quickly explode (depending on the kernel), especially when the basis centroids are positioned near one another, or set everything to zero. To avoid such issues, we utilize a initialization scheme where control point pairs $\{(x,y)_{i\in s}\}$ are uniformly placed along a 'hockey stick' curve and the parameters $\{\lambda_{i\in s}\}$, v_0, and v_1 are solved for using linear regression. To ensure stability when training the kernel coefficients, $\{\lambda_{i\in s}\}$ must remain balanced, otherwise the RBF output can become overly sensitive to updates to λ_i's during learning and result in drastically different non-linear functions. We use the following loss function to aid in stability

$$L_{\text{DeepLABNet}} = L_{model} + \lambda_{sum} \frac{1}{A} \sum_{a}^{A} |\sum_{i}^{s} \lambda_{ai}| \qquad (3)$$

where $L_{\text{DeepLABNet}}$ is the loss function for a network using DeepLABNet, L_{model} is the loss function for a given model without the added RBF functions, λ_{sum} is the regularization strength, and A is the number of RBF functions in a DNN.

2 DeepLABNet vs RBF Networks

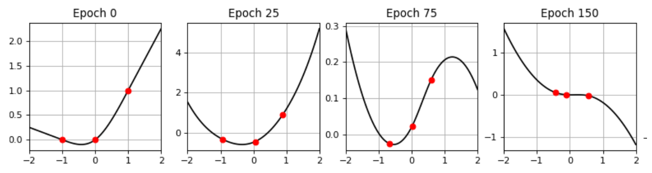

Fig. 1: A DeepLABNet RBF activation function throughout training. The RBF function centroids are marked with red dots.

We analyze DeepLABNet by comparing to two other network designs: i) a traditional single layer RBF network, and ii) a deep convolutional neural network that uses traditional RBF networks for activation (which we call RBF activation network). A standard deep convolutional neural network with ReLU activation is used as a baseline reference. Each design is tested against the following three benchmark datasets: MNIST [2], CIFAR-10 [3], and CIFAR-100 [3]. The baseline network, RBF activation network, and DeepLABNet use LeNet-5 [4] and ResNet-20 [5] architectures for MNIST and CIFAR-10/CIFAR-100 datasets, respectively. The single layer RBF networks use 250 hidden neurons for the MNIST dataset, and 1000 hidden neurons for both the CIFAR-10 and CIFAR-100 datasets. The validation accuracy of each model is shown in the table below.

Table 1: Comparison of the different types of RBF networks

	Baseline	RBF Network	RBF Activation Network	DeepLABNet
MNIST	97.9	94.3	69.7	**98.9**
CIFAR-10	90.6	51.5	33.4	**91.5**
CIFAR-100	65.4	22.8	3.55	**66.3**

It can be observed that DeepLABNet has the best performance on all three datasets. DeepLABNet outperforms the baseline network by 1.0%, 0.9%, and 0.9% on MNIST, CIFAR-10, and CIFAR-100, respectively. Comparatively, the RBF network has far greater number of parameters to the other models but was still unable to reach performance of the base line model and DeepLABNet. This result is not surprising as the RBF network is single layer. On the other hand, the RBF activation network is as deep as DeepLABNet while having more parameters but still fails to outperform even the single layer RBF network. The poor performance of the RBF activation network is mostly due to the network getting stuck in a local minimum during training. Moreover, this lack in performance effectively highlights the difficulty of naively integrating RBF networks directly in a deep learning pipeline and demonstrates DeepLABNet's merits.

References

[1] D. S. Broomhead and D. Lowe, "Radial basis functions, multi-variable functional interpolation and adaptive networks," tech. rep., Royal Signals and Radar Establishment Malvern, 1988.

[2] Y. LeCun, "The mnist database of handwritten digits," *http://yann.lecun.com/exdb/mnist/*, 1998.

[3] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," tech. rep., 2009.

[4] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, 1998.

[5] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016.