### SPINDLE: Self-Pretrainable In-situ Normalizer for Deep Learning Error Function

Jyun-Xin Ke, Siang-En Ciou, Ting-Yu Chen and Tsung-Chu Huang

Department of Electronics Engineering, National Changhua University of Education, Changhua, Taiwan

*Abstract*—For new datasets with distinct distributions from that of pretrained features, not only the model weights but also the activation function shapes should be fine-tuned for accelerating convergence, optimizing successive quantization and promoting cross-validation accuracy. In this paper, we propose a spindle-like ReLU-based radial basis error function called SPINDLE for self-pretrainable in-situ normalizer for deep leaning. Both off-line and on-line algorithms are also proposed for efficient training. Finally the proposed SPINDLE can save the logistic in a normalizing activation function, and the square-exponential operations in radial basis functions.

1. Introduction

Neural network (NN) has become the major mathematical model in artificial intelligence (AI) and dramatically applied in ubiquitous consumer electronics (CE). Owing to diversity of sensors, environmental transplantation from pretrained sea-data backbone feature extractor, and combination of modularized NNs, not only is the input probability density function (pdf) of an activation function no longer a normal distribution, it may even be an arbitrary distribution required to fine-tune. For non-linear signals or non-normal-distributed dataset *x* in a pdf(*x*) as shown in Fig.1(a), the associate cumulative distribution function (CDF) can map *x* to a *N* discrete data in a uniform distribution, which can then be optimized for -bit quantization. Only for small redundant NNs in an existing precision computing system, people like to apply the rectified linear unit (ReLU) as an activation function. For the activation functions with the sum of many inputs, the pdf approaches to a normal distribution according to the law of large numbers, therefore its CDF, an error function (*erf*) is applied as the activation function. For efficient gradient operations, the erf is usually approximated to hyper-tangent tanh(*x*) or logistic, (tanh(*x*)+1)/2, functions, as shown in Fig.1(b).

(a)(b)

Fig. 1 (a) A CDF as an activation function and (b) a special case in .

Recently, several companies and institutes have devoted great resources to develop a backbone NN for sea data object recognition, such as the Transformer [1][2] and ViT [3][4]. The time-consuming pretrained feature extractors can be applied in consumers’ environment just after a while of an in-situ fine-tuning for their specific classifier as shown in Fig.2(a). However, usually only the weights of the classifier are tuned but the shapes of the activation functions not. This will crucially limit the accuracy improvement.

(a)  (b) 

Fig. 2 (a) Fine-tuning extracted features and (b) a RBF in [12].

The shape of activation functions can be also tuned using look-up tables (LUT), but they either consume a lot of memory [5][6]or suffer tuning [7][8]. Recently, the authors in [9][10] started to tune the shapes of activation functions using Gaussian kernel-basis functions (KBF) for improving supervised backpropagation. That is, the centers and deviations are trained by gradient-decent learning. It is not supervised that the Gaussian kernel has been studied for a long time in estimating a non-normal distribution by kernel density estimation (KDE) [11]. Although the KBF is efficient in unsupervised forward clustering networks including KBF network (KBFN) [12] and support vector machine (SVM) [13], however, for tuning the shape of each activation function [14], the computing cost of so many radial basis functions (RBF) will be a crucial burden. Therefore in this paper, we propose a spindle-like self-pretrainable in-situ normalizer for deep learning error (SPINDLE) function for fine-tuning any pdf inputs efficiently. The name is also due to its spindle shape like [15], but they have no direct relation.

1. Proposed SPINDLE Function

Fig.3 shows four generalized ReLU functions can be specified by two signs, , and ,

(1)

Since any continuous functions can be approximated by a range-addressable look-up table (RALUT) [8], the slope of the *i-*th piecewise line (PWL), can be represented by a ternary-coded binary number with a fixed-point unit ,

, (2)

where .



Fig. 3 Four generalized ReLU functions.

Since all continuous CDFs are monotonically ascending functions such that only is considered, a SPINDLE can be implemented as

(3)

where and operation () denotes a sign-extension left (right) shift. Fig.4 shows an architecture. Note that for constant shifts, only wires are conducted for addition without hardware shifters.



Fig. 4 The proposed SPINDEL architecture.

1. Reconfiguration Algorithms

The reconfiguration for proposed SPINDER can be in-situ training algorithm, and off-line algorithms. The off-line algorithms can further be sampling ranking and the light-slope PWL searching (LS-PWL) algorithm in [16].

*A. On-Line Backpropagation*

Initially all biases are set to the minimum values and gradually moved right or adapted by training. From Eq.3, we have the gradient according to the linearity.

(4)

The operations are similar to the gradient of a ReLU. Preventing from oscillating, a batch learning with each batch of size 16 or 32samples are trained. Note that the trained function will be the generalized error function, namely, the approximate CDF of the input distribution.

*B. Off-Line Sample Ranking and Searched by the LS-PWL*

An N-sample part of dataset are randomly selected from the dataset and expected to have an approximate pdf similar to distribution of the associate input. The part can be also the total dataset. The samples are then sorted and annotated the fraction r/N, where r is the rank. The PWL can be constructed by the bin-based accumulation or batch-based accumulation. For each equal-bin, the sizes in each bin will show a histogram, and the approximated PWL-pdf can be constructed. For the batch-based accumulation with a batch size B from to , the slope will be B/( and taken to selecting proper ReLUs for constructing the PWL.

1. Evaluation and Experimental Results

To verify the learning acceleration and convergent error, a 10,000-sample dataset is generated in a bimodal distribution, *f2* = + and a twin-peak *f2* = + , where is a triangle distribution within a and c with a peak at b. The convergence rate and non-normal kernel proximity are briefly evaluated in Table I due to limited space.

1. Comparison of AN-Decoders with Previous Work.

|  |  |  |
| --- | --- | --- |
| Functions | RBF | SPINDLE |
| Convergence | Fast | Slow due to small learning rate |
| Non-normal kernel proximity | Poor | Best |

Since there are multiple freedoms in the superposition of SPINDLE, its backpropagation should be adapted randomly and slowly. However, the SPINDLE can approximate any function including triangle and sinusoidal pdfs well. For a *k*-peak pdf, bin and/or batch spreading is still required for separating the peaks for RBF-based KDE with a complexity of about ***O***(*N*), however, the error and difficult will be increased for a large *k*. As for sample ranking for SPINDLE, the complexity is in ***O***(*N*log*N*) and it usually needs about 4k to 6k pieces of ReLUs.

Both *k*-kernel RBF and *k*-ReLU SPIDLE are implemented to an all-programmable SoC zynq-7020 in the Xilinx Pynq-z2 system. Table II shows the required elements, where DSP, FF, LUT and CY are separately numbers of elements DSP48E, flip-flop, LUT2 and CarryLogic in FPGA zynq-7020. Note that the DSP48E is a 48-bit versatile MAC with pre-addition and post-logic. Obviously, with the same number of kernels or ReLU, *k*, the proposed SPIDLE can save more than 86% of FFs and all DSP48Es. Even for a k-peak distribution approximated by *k* kernels for a RBF and about 6*k* segments of ReLUs for SPINDLE, we can reduce a lot of area overhead with power dissipation.

1. Comparison of AN-Decoders with Previous Work.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Func. | RBF | | | SPINDLE | | |
| *k* | DSP | FF | LUT | DSP | FF | LUT |
| 1 | 10 | 2,244 | 3,466 | 0 | 434 | 583 |
| 2 | 22 | 4,316 | 6,106 | 0 | 574 | 792 |
| 4 | 28 | 6,720 | 7,709 | 0 | 951 | 1,202 |
| 8 | 40 | 11,464 | 10,791 | 0 | 1,575 | 2,024 |
| 16 | 64 | 20,952 | 16,981 | 0 | 2,984 | 3,659 |

1. Conclusions and Future Work

In this paper we propose a spindle-like ReLU array structure for online fine-tuning or off-line sample-ranking to approximate any CDF as the proper activation function. Since RBFs suffer crucially heavy cost, the SPINDLE will be the best solution for fine-tuning variable distributed features or signal inputs of environment-changeable NN systems.

References

1. Dosovitskiy, A., “An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale”, *arXiv e-prints*, 2020.
2. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., and Zagoruyko, S., “End-to-End Object Detection with Transformers”, *arXiv e-prints*, 2020.
3. Chen, T., Cheng, Y., Gan, Z., Yuan, L., Zhang, L., and Wang, Z., “Chasing Sparsity in Vision Transformers: An End-to-End Exploration”, arXiv e-prints, 2021.
4. He, K., Chen, X., Xie, S., Li, Y., Dollár, P., and Girshick, R., "Masked Autoencoders Are Scalable Vision Learners", *arXiv e-prints*, 2021.
5. M. Zhang, S. Vassiliadis and J. Delgado-Frias (1996) 'Sigmoid generators for neural computing using piecewise approximations', IEEE Trans. Comp., vol. 45, no. 9, pp. 1045-1049
6. B. Zamanlooy and M. Mirhassani, "Efficient VLSI Implementation of Neural Networks With Hyperbolic Tangent Activation Function," in IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 22, no. 1, pp. 39-48, Jan. 2014.
7. S. Saranya and B. Elango, "Implementation of PWL and LUT based approximation for hyperbolic tangent activation function in VLSI," 2014 Int'l Conf. Comm. and Signal Proc., 2014, pp. 1778-1782.
8. W.-C. Yang *et al*. "Range-Lookup Approximate Computing Acceleration for Any Activation Functions in Low-Power Neural Network," IEEE Int'l Conf. Consumer Electronics - Taiwan, 2020.
9. Y. Bodyanskiy, A. Pirus and A. Deineko, "Multilayer Radial-basis Function Network and its Learning," 2020 IEEE 15th Inter’l Conf. Computer Sci. and Info. Technologies (CSIT), 2020, pp. 92-95.
10. Jiang, Q., Zhu, L., Shu, C. et al. Multilayer perceptron neural network activated by adaptive Gaussian radial basis function and its application to predict lid-driven cavity flow. Acta Mech. Sin., 2022.
11. Parzen, E. "On Estimation of a Probability Density Function and Mode". The Annals of Math. Statistics. 33 (3): 1065–1076, 1962.
12. J. Moody,C. J. Darken,“Fast learning in networks of locally-tuned processing units”,Neural Computation, Vol. 1, 1989,pp.281-294.
13. Matteo Fischetti. "Fast training of Support Vector Machines with Gaussian kernel" Discrete Optimiz., 22(A), 2016, pp. 183-194, 2016.
14. J.A.Leonard, M. A.Kramer, L. H. Ungar,“Using radial basis functions to approximate a function and its error bounds”, IEEE Trans. on Neural Networks, Vol. 3, 1992, pp.614-627.
15. H. Zhao et al., "Spindle Net: Person Re-identification with Human Body Region Guided Feature Decomposition and Fusion," 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 907-915.
16. T.-Y. Chen, C.-D Tsai, H.-W. Fu, Y.-C. Yang and T.-C. Huang. "Error Correctable Range-Addressable Lookup for Activation and Quantization in AI Automotive Electronics," 2021 IEEE ICCE-TW, Penghu, Taiwan, June 17, 2021.