Auto-Rotating Perceptrons

Anonymous Author(s)
Affiliation
Address
email

1 Abstract

This paper proposes an improved design of the perceptron unit in order to alleviate the vanishing gradient problem that appears when training large multilayer networks which use these units and bounded activation functions. The new neuron design, namely auto-rotating perceptron (ARP), has a mechanism to ensure that the unit always operates in the dynamic region of the activation function, thus avoiding saturation of the unit. The proposed method does not change the inference structure learned at each neuron. We evaluate the effect of using ARP units in some network architectures with sigmoidal activation function. The results support our hypothesis that networks with ARP units are able to achieve better learning performance than equivalent networks with classic perceptron units.

2 Introduction

Deep neural networks (DNN) models are widely used for inference problems in several areas, such as object detection [1], pattern recognition [2] and image reconstruction [3]. Theoretically, the stacking architecture of these models allows them to learn any mapping function from input to output variables [4–6]. However, in practice, DNN models are very difficult to train when using neural units with bounded activation functions [7]. In such cases, the computation of the error gradients, needed by most learning methods to adjust the network weights, becomes problematic due to the exponential decrease of the gradients as we go down to the initial layers, which can cause slow learning, premature convergence and poor model inference performance [8, 9]. This issue is known as the vanishing gradient problem (VGP) [10, 11]. Researchers have proposed a plethora of unbounded activation functions (e.g. ReLU, PReLU, leaky ReLU, ELU, etc; a review can be found in [12]) as a way to overcome the VGP. Nonetheless, some authors have proposed bounded versions of such activation functions that show to be effective in alleviating the training instability of DNN models [13].

In this paper, we propose a different approach to tackle the VGP in DNN learning. Instead of worrying about the activation function, we design a mechanism for the pre-activation phase. The intuition of this mechanism, called auto-rotation (AR), is to avoid the activation function derivative to take small values (i.e., maintaining the operation of the neural unit in its dynamic region). We show the advantage of this mechanism when applied to multilayer perceptron networks (MLP) by improving their learning performance.

3 Auto-Rotating Perceptron (ARP)

Recalling, a perceptron unit is a function that maps an input vector $\mathbf{x} \in \mathbb{R}^n$ to an output $\hat{y} \in \mathbb{R}$. This mapping is done in two steps. First, a weighted sum of the inputs is computed: $f(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x} + w_0\, x_0$, where $\mathbf{w} \in \mathbb{R}^n$ is the weight vector and $x_0 = 1$. Then, a non-linear function $\sigma(\cdot)$ (i.e., the activation function) is applied to the weighted sum to obtain the neuron output: $\hat{y} = \sigma(f(\mathbf{x}))$.

We define the n-dimensional hyperplane φ in an $(n+1)$-dimensional orthogonal space as the points whose first n dimensions are the input vector coordinates and the additional dimension being $f(\mathbf{x})$.

Submitted to 33rd Conference on Neural Information Processing Systems (NeurIPS 2019). Do not distribute.

In other words, we create the surface $\varphi = \langle \mathbf{x}, f(\mathbf{x}) \rangle \subset \mathbb{R}^{n+1}$. When taking the points of φ where $f(\cdot) = 0$, we generate the boundary $\Gamma \subset \mathbb{R}^n$ that separates the input space into two regions with a different sign for its $f(\cdot)$ value (i.e., the region with $f(\mathbf{x}) > 0$ and the other one with $f(\mathbf{x}) < 0$).

When the neural units are arranged into hierarchical structures, which occurs in an MLP network, the perceptrons of each hidden layer learn an abstract feature from the information given by the previous level. We can interpret that the regions defined by the boundary $\Gamma = \langle \mathbf{x}, 0 \rangle \subset \varphi$ capture the presence or absence of an abstract inferred feature. What is essentially done in the training stage is to learn the non-discrete surface φ, whose intersection with the input space (i.e., the set of points of φ where $f(\mathbf{x}) = 0$) is the boundary Γ that defines the feature extracting capability of the unit. We propose to modify the inclination of the surface φ without changing the boundary Γ. This is done by rotating φ using the boundary Γ as the rotation axis.

This makes sense, since with a rotation of φ we can change the weighted input that goes into the activation function, but at the same time keeping the boundary Γ unchanged. Mathematically, we obtained that a rotation of the perceptron's n-dimensional hyperplane $\varphi = \langle \mathbf{x}, f(\mathbf{x}) \rangle \subset \mathbb{R}^{n+1}$, around an $(n-1)$-dimensional axis $\Gamma = \langle \mathbf{x}, 0 \rangle \subset \mathbb{R}^n$ (where $\Gamma \subset \varphi$), can be achieved by multiplying a real scalar value ρ to all the weights $w_i \in \mathbf{w}$ and the bias w_0. The new n-dimensional hyperplane $\hat{\varphi} = \langle \mathbf{x}, g(\mathbf{x}) \rangle \subset \mathbb{R}^{n+1}$ is defined in terms of the weighted sum $g(\mathbf{x}) = \rho\, \mathbf{w} \cdot \mathbf{x} + \rho\, w_0$, the rotation coefficient $\rho = \frac{L}{|f(\mathbf{x}_Q)|}$ and the hyperparameters $L \in \mathbb{R}$ (which defines the limits of the activation function dynamic range) and $\mathbf{x}_Q \in \mathbb{R}^n$ (which is a point outside the input data range).

Notice that the rotation mechanism acts independently in each neural unit, adjusting dynamically the pre-activation phase (since ρ depends on the perceptron weights) to prevent the activation function to saturate. Because of this auto-adaptation, we named the new unit auto-rotating perceptron.

4 Experimentation, results and future work

We evaluate the effectivity of the ARP unit, with the unipolar sigmoid activation function, in an MLP architecture using three benchmark datasets (MNIST, Fashion MNIST and CIFAR-10). The hyperparameters used were: $L = 4$ (because the sigmoid $\sigma(z)$ is not saturated for $z \leq |4|$) and $\mathbf{x}_Q \in \mathbb{R}^n$ with all its components being 1.1 (because input data is scaled to the range $|x| \leq 1 < x_Q = 1.1$). Figure 1 shows the test prediction accuracy and its corresponding standard deviation (SD). We observe a notorious improvement of test accuracy in CIFAR-10 with ARP units with respect to classic units. In Fashion MNIST, the improvement is also noticeable but at lower extend. In MNIST no improvement in accuracy is observed, but a high variability of results were found. These results show a potential of ARP to deal with the VGP, though further experimentation is needed to confirm this trend and to understand the behavior of the units with different activation functions, hyperparameter configurations, optimizers and network architectures. Source code available in GitHub.

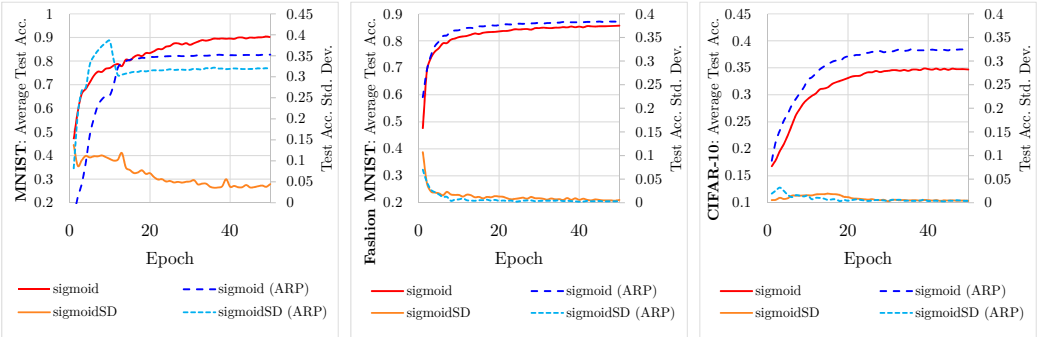

Figure 1: Comparison of the test prediction accuracy. Datasets used: MNIST [14] (left), Fashion MNIST [15] (middle) and CIFAR-10 [16] (right). MLP architecture: (input image size)-50-50-40-30-30-20-10. ARP units only in the hidden layers. Number of executions for each dataset: 30. Epochs per iteration: 50. Same initial weights and biases. Batch size: 64. Optimizer: Adam ($lr = 0.003$).

References

[1] J. Han, D. Zhang, G. Cheng, N. Liu, and D. Xu, "Advanced deep-learning techniques for salient and category-specific object detection: a survey," *IEEE Signal Processing Magazine*, vol. 35, no. 1, pp. 84–100, 2018.

[2] X.-Y. Zhang, F. Yin, Y.-M. Zhang, C.-L. Liu, and Y. Bengio, "Drawing and recognizing chinese characters with recurrent neural network," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 4, pp. 849–862, 2018.

[3] R. A. Rojas, W. Luo, V. Murray, and Y. M. Lu, "Learning optimal parameters for binary sensing image reconstruction algorithms," in *2017 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2017, pp. 2791–2795.

[4] Y. Bengio *et al.*, "Learning deep architectures for AI," *Foundations and trends® in Machine Learning*, vol. 2, no. 1, pp. 1–127, 2009.

[5] Y. LeCun, Y. Bengio, and G. Hinton, "Deep Learning," *Nature*, vol. 521, no. 7553, p. 436, 2015.

[6] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Networks*, vol. 61, pp. 85–117, 2015.

[7] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[8] M. A. Nielsen, *Neural networks and deep learning*. Determination press USA, 2015, vol. 25.

[9] B. Xu, R. Huang, and M. Li, "Revise saturated activation functions," *arXiv preprint arXiv:1602.05980*, 2016.

[10] Y. Bengio, P. Simard, P. Frasconi *et al.*, "Learning long-term dependencies with gradient descent is difficult," *IEEE Transactions on Neural Networks*, vol. 5, no. 2, pp. 157–166, 1994.

[11] S. Hochreiter, Y. Bengio, P. Frasconi, J. Schmidhuber *et al.*, "Gradient flow in recurrent nets: the difficulty of learning long-term dependencies," 2001.

[12] C. Nwankpa, W. Ijomah, A. Gachagan, and S. Marshall, "Activation functions: Comparison of trends in practice and research for deep learning," *ArXiv*, vol. abs/1811.03378, 2018.

[13] S. S. Liew, M. Khalil-Hani, and R. Bakhteri, "Bounded activation functions for enhanced training stability of deep neural networks on visual pattern recognition problems," *Neurocomputing*, vol. 216, pp. 718–734, 2016.

[14] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner *et al.*, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[15] H. Xiao, K. Rasul, and R. Vollgraf. (2017) Fashion MNIST: A Novel Image Dataset for Benchmarking Machine Learning Algorithms.

[16] A. Krizhevsky, G. Hinton *et al.*, "Learning multiple layers of features from tiny images," Citeseer, Tech. Rep., 2009.