Les Cahiers du GERAD ISSN: 0711–2440

Random bias initialization improves quantized training

X. Li,
V. Partovi Nia

G–2020–23–EIW07

April 2020

Citation suggérée : X. Li, V. Partovi Nia (Avril 2020). Random bias initialization improves quantized training, *In* C. Audet, S. Le Digabel, A. Lodi, D. Orban and V. Partovi Nia, (Eds.). Proceedings of the Edge Intelligence Workshop 2020, Montréal, Canada, 2–3 Mars, 2020, pages 44–50. Les Cahiers du GERAD G–2020–23, GERAD, HEC Montréal, Canada.

Avant de citer ce rapport technique, veuillez visiter notre site Web (https://www.gerad.ca/fr/papers/G-2020-23-EIW07) afin de mettre à jour vos données de référence, s'il a été publié dans une revue scientifique.

Suggested citation: X. Li, V. Partovi Nia (April 2020). Random bias initialization improves quantized training, *In* C. Audet, S. Le Digabel, A. Lodi, D. Orban and V. Partovi Nia, (Eds.). Proceedings of the Edge Intelligence Workshop 2020, Montreal, Canada, March 2–3, 2020, pages 44–50. Les Cahiers du GERAD G–2020–23, GERAD, HEC Montréal, Canada.

Before citing this technical report, please visit our website (https://www.gerad.ca/en/papers/G-2020-23-EIW07) to update your reference data, if it has been published in a scientific journal.

La publication de ces rapports de recherche est rendue possible grâce au soutien de HEC Montréal, Polytechnique Montréal, Université McGill, Université du Québec à Montréal, ainsi que du Fonds de recherche du Québec – Nature et technologies.

Dépôt légal – Bibliothèque et Archives nationales du Québec, 2020
– Bibliothèque et Archives Canada, 2020

The publication of these research reports is made possible thanks to the support of HEC Montréal, Polytechnique Montréal, McGill University, Université du Québec à Montréal, as well as the Fonds de recherche du Québec – Nature et technologies.

Legal deposit – Bibliothèque et Archives nationales du Québec, 2020
– Library and Archives Canada, 2020

GERAD HEC Montréal
3000, chemin de la Côte-Sainte-Catherine
Montréal (Québec) Canada H3T 2A7

Tél. : 514 340-6053
Téléc. : 514 340-5665
info@gerad.ca
www.gerad.ca

Random bias initialization improves quantized training

Xinlin Li [a]
Vahid Partovi Nia [b]

[a] *Huawei Noah's Ark Lab, Montréal (Québec), Canada, H3N 1X9*

[b] *GERAD, HEC Montréal, Montréal (Québec), Canada, H3T 2A7*

xinlin.li1@huawei.com
vahid.partovinia@huawei.com

April 2020
Les Cahiers du GERAD
G–2020–23–EIW07

Abstract: *Binary neural networks improve computationally efficiency of deep models with a large margin. However, there is still a performance gap between a successful full-precision training and binary training. We bring some insights about why this accuracy drop exists and call for a better understanding of binary network geometry. We start with analyzing full-precision neural networks with ReLU activation and compare it with its binarized version. This comparison suggests to initialize networks with random bias, a counter-intuitive remedy.*

1 Introduction

It is common to use low-bit quantized networks such as Binary Neural Networks (BNNs) [1] to implement deep neural networks on edge devices such as cell phones, smart wearables, etc. BNNs only keep the sign of weights and compute the sign of activations $\{-1, +1\}$ by applying the sign function in the forward pass. In backward propagation, BNN uses Straight-Through-Estimator (STE) to estimate the backward gradient through the sign function and update on full-precision weights. The forward and backward loop of a BNN, therefore, becomes similar to the full-precision neural network with hard hyperbolic tangent *htanh* activation. The htanh function is a piece-wise linear version of the nonlinear hyper-bolic tangent, and is known to be inferior in terms of accuracy compared to ReLU-like activation function. Although the analysis is based on htanh function, this conclusion equally applies to BNNs that use STE, a htanh-like, back propagation scheme. Other saturating activations like hyperbolic tangent and sigmoid commonly applied in recurrent neural networks and attention-based models may benefit from this resolution as well. Among others, [3] recommends an initialization scheme for binary weights but ignores the bias term. [2] utilized automatic search techniques on searching different activation functions. Most top novel activation functions found by the searches have an asymmetric saturating regime, which is similar to ReLU.

2 Full-precision networks

A typical full-precision neural network block can be described by

$$\begin{gathered} x^{i+1} = \mathrm{ReLU}(W^i x^i + b^i) \\ W^i \in \mathbb{R}^{m\times n}, b^i \in \mathbb{R}^m, x^i \in \mathbb{R}^n, x^{i+1} \in \mathbb{R}^m. \end{gathered} \tag{1}$$

Neural networks are trained using the back-propagation algorithm. Back propagation is composed of two components i) forward pass and ii) backward propagation. In the forward pass, the loss function $\mathcal{L}(.)$ is evaluated on the current weights, and in backward propagation, gradients and then weights are updated sequentially.

Assume weight vectors W^i_j have unit norm. It is a reasonable assumption when the network has batch normalization layers in which all neuron responses are normalized, as the magnitude of the weight vectors does not affect the layer output. The j^{th} neuron response in the $(i+1)^{th}$ layer are computed as

$$x^{i+1}_j = \begin{cases} W^i_j x^i + b^i_j & W^i_j x^i + b^i_j > 0 \\ 0 & W^i_j x^i + b^i_j \leq 0 \end{cases} \tag{2}$$

First, the input data points x^i are projected to the j^{th} row vector of the weight matrix. The dot product of W^i_j and x^i are cut by the corresponding bias term b^i_j, i.e. the output x^{i+1}_j is set to zero if the dot product is smaller than the threshold, see Figure 2 (left panel). A hyper-plane whose normal direction defined by W^i_j divides the input space into two parts: i) activated region (non-saturated regime) and ii) non-activated region (saturated regime), see Figure 1. If the data point x^i falls on the positive side of a hyper-plane (activated region), the hyper-plane is activated by x^i. Consequently, x^{i+1}_j is positive. Otherwise, x^{i+1}_j equals zero and remains inactive.

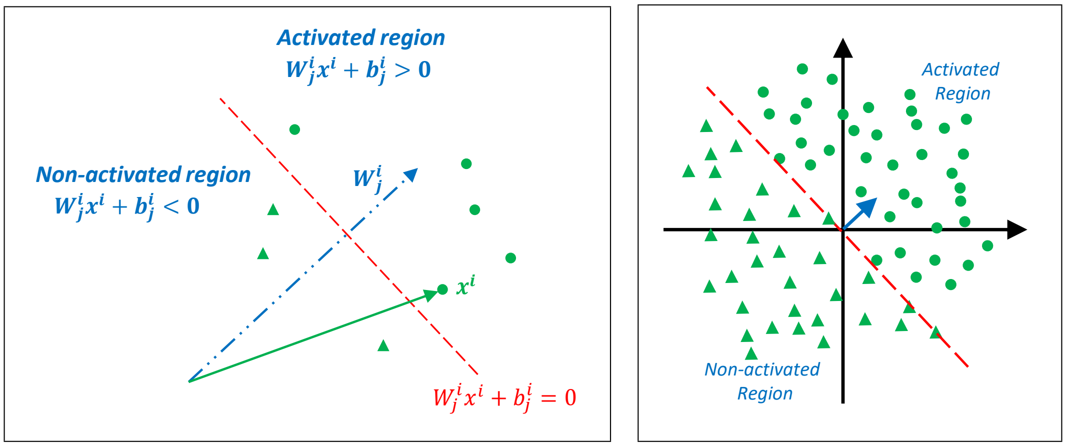

Figure 1: Activated and non-activated regions of ReLU (left panel). Activated region of ReLU at initialization (right panel)

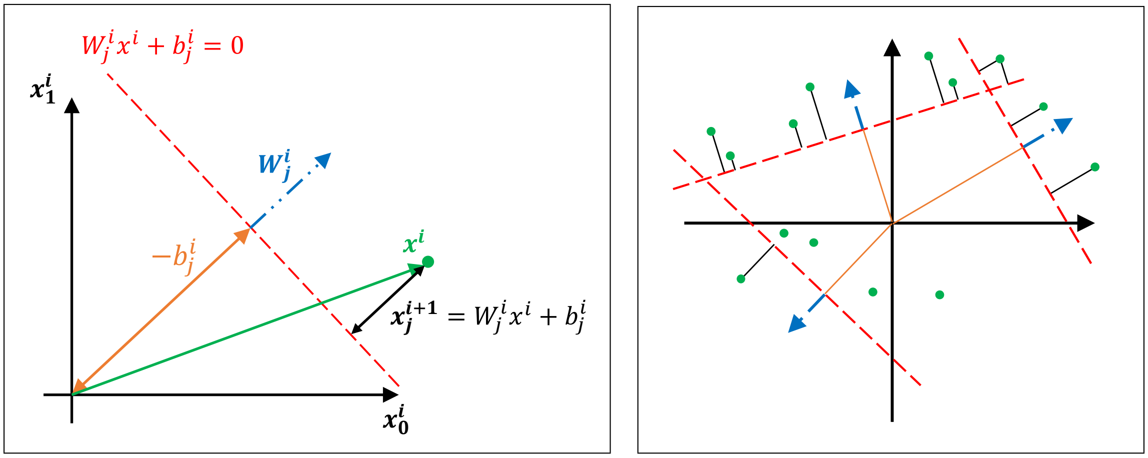

Figure 2: Geometric behavior of ReLu during forward pass, trained hyperplanes (left panel) and their geometry (right panel)

The weight matrix W^i of size $m \times n$ and the bias vector b^i of size $m \times 1$ define m hyper-planes in the n-dimensional input space, see Figure 2 (right panel).

During backward propagation, the backward gradient update on W_j^i and x^i are computed using

$$\begin{cases} \frac{d\mathcal{L}}{dW_j^i} = \frac{d\mathcal{L}}{dx_j^{i+1}} * \frac{dx_j^{i+1}}{dW_j^i} \\ \frac{d\mathcal{L}}{db_j^i} = \frac{d\mathcal{L}}{dx_j^{i+1}} * \frac{dx_j^{i+1}}{db_j^i} \\ \frac{d\mathcal{L}}{dx^i} = \frac{d\mathcal{L}}{dx_j^{i+1}} * \frac{dx_j^{i+1}}{dx^i} \end{cases} \tag{3}$$

For the case of ReLU activation

$$\frac{dx_j^{i+1}}{dW_j^i} = \begin{cases} x^i & W_j^i x^i + b_j^i > 0 \\ 0 & W_j^i x^i + b_j^i \leq 0 \end{cases} \tag{4}$$

$$\frac{dx_j^{i+1}}{db_j^i} = \begin{cases} 1 & W_j^i x^i + b_j^i > 0 \\ 0 & W_j^i x^i + b_j^i \leq 0 \end{cases} \tag{5}$$

$$\frac{dx_j^{i+1}}{dx^i} = \begin{cases} W_j^i & W_j^i x^i + b_j^i > 0 \\ 0 & W_j^i x^i + b_j^i \leq 0 \end{cases} \tag{6}$$

The activation function only allows the gradients from data point on the activated region to backward propagate and update the hyper-plane (4).

From the hyper-plane analysis, we realize that ReLU activation has three ideal properties i) the diversity of activated regions at initialization, ii) The equality of data points at initialization, iii) The equality of hyper-planes at initialization which we discuss each property in more details later. These may explain why ReLU activation outperforms the traditional Hyperbolic tangent or sigmoid activations. To argue each property, let us suppose that the distribution of the dot products is zero-centered. This assumption is automatically preserved in neural networks with batch normalization layer.

i) Region diversity: the activated regions of hyper-planes solely depend on the direction of the weight vector, which is randomly initialized. This allows different hyper-planes to learn from a different subset of data points, and ultimately diversifies the backward gradient signal. ii) Data equality: an arbitrary data point x^i, is located on activated regions of approximately half of the total hyper-planes in a layer. In other words, the backward gradients from all data points can pass through the approximately same amount of activation function, update hyper-planes, and propagate the gradient. iii) Hyperplane equality: an arbitrary hyper-plane W^i_j, is affected by the backward gradients from approximately 50% of the total data points. All hyper-planes on average receive the same amount of backward gradients. Hyper-plane equality speeds up the convergence and facilitates model optimization, see Figure 1 (right panel).

The performance gap between ReLU activation and htanh activation is caused by their different activated region distribution, see Figure 3. Clearly, htanh activation is not as good as ReLU in defining balanced and fair activated regions. However, we analyze each property for htanh as well.

i) Region diversity: activated regions of htanh are not as diverse as ReLU. Activated regions of htanh cover only the area close to the origin. Assuming Gaussian data, this is a dense area that the majority of data points are located in.

ii) Data equality: data points are not treated fairly htanh activation function. Data points that closer to the origin can activate more hyper-planes than the data points far from the origin. If the magnitude of a data point x^i is small enough, it can activate all hyper-planes in the same layer, see the deep-red region of Figure 3 (right panel). As a consequence, in backward gradients, few data instances affect all hyper-planes. In other words, the backward gradients from a part of the training data have a larger impact on model than others. This imbalance ultimately affects model generalization problem since the model training focuses only on a subset of the training data points close to the origin.

iii) Hyperplane equality: The initial activated regions should cover a similar-sized subset of the training data points overall, and this property is shared in both ReLU and htanh activations. Similar analysis also applies to other activation functions with the zero-centered activated region, like sigmoid or tanh.

3 Training acceleration

Here we proposed a simple initialization strategy to alleviate the data inequality issue and improve activated region diversity for the htanh activation relying on our geometric insight described earlier. We argue bias initialization with a uniform distribution between $[-\lambda, \lambda]$, where λ is a hyper-parameter is a quick remedy. With random bias initialization, the data points that far from the origin can activate more hyper-planes. If $\lambda > \max(\|x\|) + 1$, all data points activate approximately the same number of hyper-planes during backward propagation, so data equality can be achieved. Also, with the diverse initial activated region, different hyper-planes learn from different subset of training data points.

However, this initialization strategy comes with a drawback. Hyper-plane equality no longer holds when the biases are not set to zero. Hyper-planes with larger initial bias have less activated data.

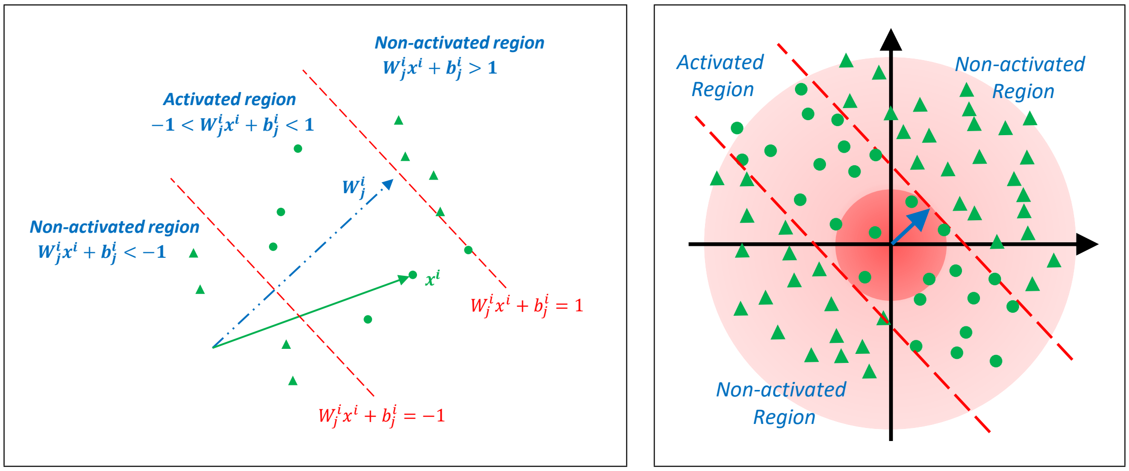

Figure 3: Activated region and non-activated region of htanh activation function(left panel). Activated region of Hard Tanh at initialization (right panel)

Therefore, choosing the optimal value of λ is a trade-off between the hyper-plane equality and the data equality. Experiments below shows that the validation curve becomes unsteady if λ value set to too high. Empirically, with a batch normalization layer, $\lambda \approx 2$ provide a good initial estimate. In this case, the activated regions covering from -3 to $+3$, so it allows the gradients from almost all data points to propagate. Our experiments shows small λ also helps to improve the performance of ResNet architecture.

4 Numerical Results

The proposed bias initialization method is evaluated on the CIFAR-10. The network architectures are based on the original implementation of the BNN [1]. We choose the VGG-7 architecture and the ResNet architecture.

The VGG-7 architecture, is a simple and over-parameterized model for CIFAR 10. This is an ideal architecture to compare the performance between different activations. Figure 4 confirms that the random bias initialization strategy helps to reduce the performance gap between htanh and ReLU activation. A similar effect is observed for ResNet type architectures.

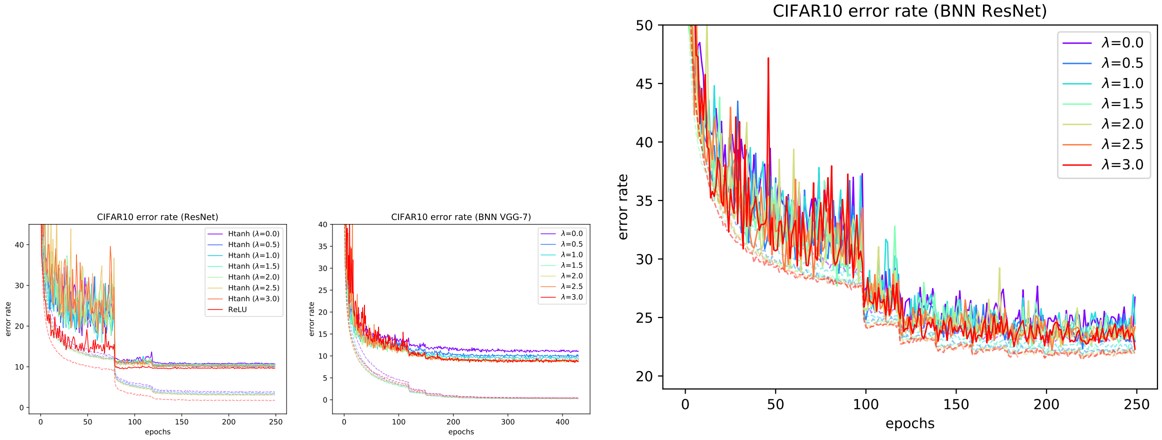

Figure 4: Training of full-precision ResNet architecture (top left panel) Binary VGG-7 architecture (top right), and Binary ResNet (bottom panel)

We also tested the proposed bias initialization on the ResNet-like architecture. The results are depicted in Figure 4 re-assures that bias initialization improves htanh and pushes it toward ReLU accuracy, see Table 1.

Table 1: Validation error rate % for full-precision training, $\lambda = 0$ coincides with common deterministic initialization

Activations	VGG-7	ResNet
ReLU (Baseline)	6.98	9.45
htanh	10.91	10.63
htanh (λ=0.5)	9.99	9.87
htanh (λ=1.0)	9.15	10.47
htanh (λ=1.5)	8.36	10.13
htanh (λ=2.0)	7.98	10.23
htanh (λ=2.5)	**7.83**	**9.84**

Binary training that use STE is similar to htanh activation. We expect to observe a similar effect in BNN training with STE gradient approximator. The validation error rate is summarized in Table 2. In the Binary VGG-7 experiments, we reduced the accuracy gap between full-precision network with ReLU activation and BNN from 4% to 1.5%. The bias initialization strategy is effective to close the gap on binary ResNet architecture by almost 1%, even while the full-precision model even under-fits on CIFAR10 data.

Table 2: Validation error rate % for Binary training, $\lambda = 0$ coincides with common deterministic initialization

λ	Binary VGG-7	Binary ResNet
0.0	10.77	23.11
0.5	9.57	22.31
1.0	9.17	22.83
1.5	8.57	22.56
2.0	8.56	22.47
2.5	8.53	**22.19**
3.0	**8.48**	22.30

5 Conclusion

We analyzed different geometric behaviour of ReLU activated and hard tanh activated full-precision neural network. The analysis implies the superior performance of ReLU activation may come from its three preferred geometric properties, region diversity, data equality and hyper-plane equality. However, a theoretical investigation is required to prove this claim. We proposed to use random bias initialization in hard tanh activated neural network to micmic the geometric properties of ReLU. The same analysis can also apply to binary neural networks with Straight-Through-Estimator back-propagation scheme. Our numerical experiments confirm our geometric intuition. The CIFAR10 experiments show the proposed random bias initialization reduces the performance gap between ReLU activation and hard tanh activation on ResNet and VGG architectures. This initialization strategy improves the binary neural network performance as well.

6 Acknowledgement

We would like to thank Huawei CBG Software Shanghai colleagues, especially Mohan Liu and Li Zhou for their fruitful technical discussions. We also thank Yanhui Geng and Jin Tang for their support throughoutthe project, and Masoud Asgharian for reviewing the manuscript.

References

[1] Matthieu Courbariaux, Itay Hubara, Daniel Soudry, Ran El-Yaniv, and Yoshua Bengio. Binarized neural networks: Training deep neural networks with weights and activations constrained to $+1$ or -1. 2016.

[2] Prajit Ramachandran, Barret Zoph, and Quoc V Le. Searching for activation functions. arXiv preprint arXiv:1710.05941, 2017.

[3] Eyyüb Sari, Mouloud Belbahri, and Vahid Partovi Nia. How does batch normalization help binary training? arXiv preprint arXiv:1909.09139v2, 2019.