The Effect of Target Normalization and Momentum on Dying ReLU

Isac Arnekvist[1], J. Frederico Carvalho[2], Danica Kragic[1], and Johannes A. Stork[3]

[1] KTH Royal Institute of Technology
[2] Univrses AB
[3] Örebro University
Correspondence to isacar@kth.se

Abstract. Optimizing parameters with momentum, normalizing data values, and using rectified linear units (ReLUs) are popular choices in neural network (NN) regression. Although ReLUs are popular, they can collapse to a constant zero and "die", effectively removing their contribution from the model. In this paper, we consider the effects of target normalization and momentum on dying ReLUs. We show how optimization of a single ReLU model is influenced by the properties of two mirrored cones in parameter space, one corresponding to dying ReLU, and the other sharing properties with an oscillating discrete-time linear autonomous system when using momentum[4].

1 Introduction

Gradient-based optimization enables learning of powerful deep NN models [1,2]. However, most learning algorithms remain sensitive to learning rate, scale of data values, and the choice of activation function—making deep NN models hard to train [3,4]. Stochastic gradient descent with momentum [5,6], normalizing data values to have zero mean and unit variance [7], and employing rectified linear units (ReLUs) in NNs [8,9,10] have emerged as an empirically motivated and popular practice. In this paper, we analyze a specific failure case of this practice, referred to as "dying" ReLU.

The ReLU activation function, $y = \max\{x, 0\}$ is a popular choice of activation function and has been shown to have superior training performance in various domains [11,12]. ReLUs can, however, be initialized "dead" as a constant zero function [13] or die during optimization, the latter being a major obstacle in training deep NNs [14,15]. Once dead, gradients are zero making recovery possible only if inputs change distribution.

Mitigations include modifying the ReLU to also activate for negative inputs [16,17,18], training procedures with normalization steps [19,20], and initialization methods [13]. The underlying cause for ReLUs dying during optimization still remains poorly understood.

[4] Full manuscript with more results and derivations: arxiv.org/abs/2005.06195

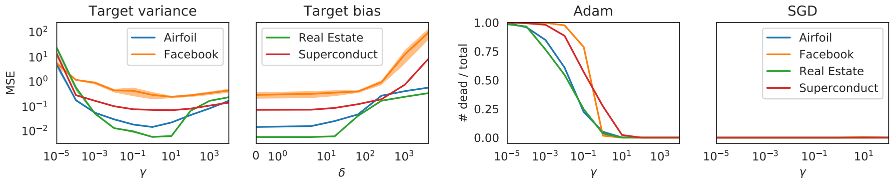

Fig. 1. The performance of models, in terms of fit to the training data, degrades as the variance (γ) of target values deviate from 1, or the mean (δ) deviates from 0. Here shown in terms of mean-squared error (MSE) on four different datasets [21]. Furthermore, decreasing γ leads to more dying ReLUs only with the use of momentum optimization [6].

In this paper, we analyze the observation illustrated in Figure 1 that regression performance degrades with smaller target variances, potentially caused by dying ReLU because of the use of momentum. Although target normalization is a common pre-processing step, we believe an understanding of *why* it is important is missing, especially with the connection to momentum optimization.

2 Preliminaries

The basic building block of a neural network is an affine transformation $\mathbb{R}^n \mapsto \mathbb{R}$

$$\hat{y} = \mathbf{w}^\top \mathbf{x} + b \tag{1}$$

followed by a non-linear activation function

$$\hat{y}_f = f(\mathbf{w}^\top \mathbf{x} + b). \tag{2}$$

In this paper, we consider the ReLU activation function $f(x) = \max\{0, x\}$. Parameters are abbreviated $\theta = (\mathbf{w}^\top, b)^\top$. We consider the ReLU dead when

$$P(\mathbf{w}^\top \mathbf{x} + b) < \varepsilon \tag{3}$$

for some small ε, i.e., that the output is non-zero with probability smaller than ε.

This building block — affine transformation and ReLU — is what we aim to understand in this paper. Although in a deep neural network, the output of the ReLU is not compared explicitly against some ground truth, we will assume the positive part is fitted against a simple target affine transformation $y = \mathit{\Gamma}^\top \mathbf{x} + \mathit{\Delta}$. If we also assume $\mathbf{x} \sim \mathcal{N}(0, I)$, we can w.l.o.g. set $\mathit{\Gamma} = (0, 0, \ldots, \gamma)^\top$, where γ^2 will equal the variance of the target variable. We assume a squared error loss

$$\mathcal{L}(\theta) = \mathbb{E}_{\mathbf{x}} \left[\frac{1}{2} (\hat{y}(\mathbf{x}) - y(\mathbf{x}))^2 \right] \tag{4}$$

and perform momentum optimization with momentum parameter β and learning rate η.

3 Investigating only the affine model

Gradient descent with momentum, using Eq. (1) as approximator, can be written closed form as

$$\begin{pmatrix} [\mathbf{a}']_i \\ [\mathbf{m}']_i \end{pmatrix} = \underbrace{\begin{pmatrix} \beta & 1-\beta \\ -\eta\beta & 1-\eta(1-\beta) \end{pmatrix}}_{=\mathbf{A}} \begin{pmatrix} [\mathbf{a}]_i \\ [\mathbf{m}]_i \end{pmatrix} \tag{5}$$

where $\mathbf{a} = \left((\mathbf{w} - \mathit{\Gamma})^\top, b - \mathit{\Delta}\right)^\top$, subscript i denotes the i:th element, the prime ($'$) denotes the value after multiplication with the matrix $\mathbf{A}$, and $\mathbf{m}$ is the momentum vector. The behavior of the parameter evolution can be well described by the eigenvalues of $\mathbf{A}$. If all eigenvalues lies inside the complex unit circle, we are guaranteed to converge to the zero vector in the limit, i.e., where $\mathbf{w} = \mathit{\Gamma}$ and $b = \mathit{\Delta}$. More interestingly, for $\beta = 0$, all eigenvalues are real, but as we increase β, eventually eigenvalues become complex and parameters will oscillate, see Figure 2. In the next section we will show how these properties translate to the ReLU case, and how this can lead to dead ReLU.

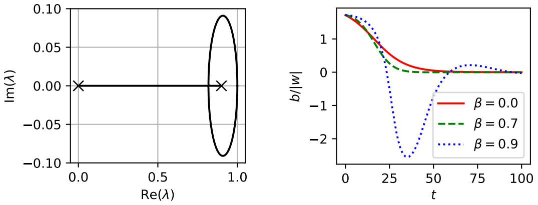

Fig. 2. Left: Eigenvalues of $\mathbf{A}$ as β is increased from 0 (marked as ×) to 1, and learning rate is kept constant at $\eta = 0.1$. At $\beta \approx 0.7$ eigenvalues the become complex and take the value 1 for $\beta = 1$. Right: As the eigenvalues become complex, gradient descent produces oscillations. These are three examples all originating from the same parameter coordinate.

4 ReLU model

The definition of a dead ReLU, using our assumption on $\mathbf{x}$, can be rewritten as $\mathit{\Phi}\left(\frac{b}{\|\mathbf{w}\|}\right) < \varepsilon$, where $\mathit{\Phi}$ is the cumulative distribution function of the standard normal. Rewriting in terms of the inverse, we get $\frac{b}{\|\mathbf{w}\|} < \mathit{\Phi}^{-1}(\varepsilon)$ which defines a "dead" cone in parameter space. Likewise, we can define a "linear" cone, $\frac{b}{\|\mathbf{w}\|} > -\mathit{\Phi}^{-1}(\varepsilon)$, where the probability of values being set to zero by the ReLU is smaller than ε.

We calculated the analytical gradients when integration is performed over $\mathbf{x}$, Eq. (4), and illustrate how the resulting vector fields in two dimensions change with a reduced target variance in Figure 3.

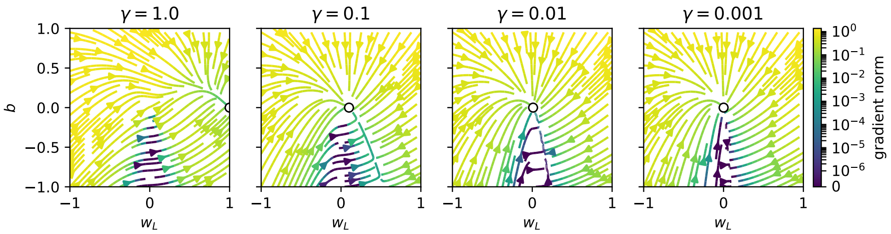

Fig. 3. Vector fields showing evolution of parameters $\theta = (w_L, b)^\top$ for different target variances γ^2 when $\beta = 0$. The dead cone is clearly visible in the lower part. The linear cone is the dead cone mirrored along the b-axis. As γ becomes smaller, parameters evolve either into the dead cone, or into the linear cone where oscillations become prevalent when introducing momentum. Ground truth also moves closer to the dead cone for smaller variances.

Parameters evolve into the linear cone (if not to the dead cone) for small γ, then approach the ground truth now close to the dead cone. In Figure 2, we observed complex eigenvalues and oscillations starting at around $\beta = 0.7$. Because of these oscillations in the linear cone, parameters overshoot into the dead cone and get stuck there. In Figure 4, we show how larger and larger areas of the parameters space converges in the dead cone as β is increased $\beta > 0.7$.

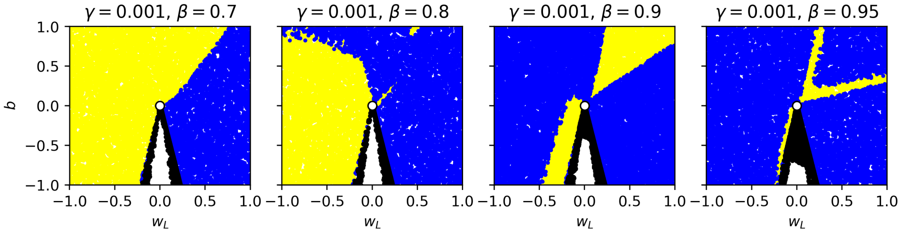

Fig. 4. Blue regions correspond to parameter initializations that converge in the dead cone. Black dots show converged parameters when two consecutive updates have norm $< 10^{-6}$. As β increases, eventually very small regions of the parameter space (yellow) are able to converge at the ground truth (white dot). Once again, the dead cone is clearly visible in the bottom part.

5 Further analysis and future work

The full manuscript presents further analysis of gradients to further support the claim of the linear and dead cone, including more thorough derivations and motivations of the claims in this abstract. Finally, we also show that the problem of dying ReLU persists in deeper architectures, including deep residual networks with batch normalization [20,22]. Future work is to synthesize advice or novel algorithms for use by researchers and practitioners.

References

1. Shaveta Dargan, Munish Kumar, Maruthi Rohit Ayyagari, and Gulshan Kumar. A survey of deep learning and its applications: A new paradigm to machine learning. *Archives of Computational Methods in Engineering*, 2019.
2. David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, 1986.
3. Rupesh K Srivastava, Klaus Greff, and Jürgen Schmidhuber. Training very deep networks. In *Advances in neural information processing systems*, pages 2377–2385, 2015.
4. Simon Du, Jason Lee, Haochuan Li, Liwei Wang, and Xiyu Zhai. Gradient descent finds global minima of deep neural networks. In *International Conference on Machine Learning*, pages 1675–1685, 2019.
5. Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton. On the importance of initialization and momentum in deep learning. In *International conference on machine learning*, pages 1139–1147, 2013.
6. Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.
7. Yann A LeCun, Léon Bottou, Genevieve B Orr, and Klaus-Robert Müller. Efficient backprop. In *Neural networks: Tricks of the trade*, pages 9–48. Springer, 2012.
8. Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436, 2015.
9. Prajit Ramachandran, Barret Zoph, and Quoc V Le. Searching for activation functions. *arXiv preprint arXiv:1710.05941*, 2017.
10. Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 807–814, 2010.
11. Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Deep sparse rectifier neural networks. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 315–323, 2011.
12. Yi Sun, Xiaogang Wang, and Xiaoou Tang. Deeply learned face representations are sparse, selective, and robust. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2892–2900, 2015.
13. Lu Lu, Yeonjong Shin, Yanhui Su, and George Em Karniadakis. Dying relu and initialization: Theory and numerical examples. *arXiv preprint arXiv:1903.06733*, 2019.
14. Ludovic Trottier, Philippe Gigu, Brahim Chaib-draa, et al. Parametric exponential linear unit for deep convolutional neural networks. In *2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 207–214. IEEE, 2017.
15. Abien Fred Agarap. Deep learning using rectified linear units (relu). *arXiv preprint arXiv:1803.08375*, 2018.
16. Andrew L Maas, Awni Y Hannun, and Andrew Y Ng. Rectifier nonlinearities improve neural network acoustic models. In *Proc. icml*, volume 30, page 3, 2013.
17. Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. Fast and accurate deep network learning by exponential linear units (elus). In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016.

18. Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 1026–1034. IEEE, 2015.
19. Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.
20. Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.
21. Dheeru Dua and Casey Graff. UCI machine learning repository, 2017.
22. Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.