Vanishing Empirical Variance in Randomly Initialized Networks

Michał Grzejdziak
Student number: 5266440
MSc Computer Science
Faculty of Electrical Engineering, Mathematics & Computer Science
Delft University of Technology

12 June 2023

Preface

The tremendous progress in neural network research over the past decade has been framed by many as "The Deep Learning Revolution". The seminal 2012 AlexNet paper suggested that that the increase of the number of layers is everything what is needed to improve the generalization performance of neural network models. The initial successes with scaling depths to hundreds of layers were followed by empirical results showing diminishing returns of further increases. These were joined by theoretical results that demonstrated that deep networks at initialization are difficult to optimize. At the same time, favorable properties of large widths were discovered. The most recent breakthroughs, including Large Language Models like GPT-4 or LLaMA, were achieved with relatively shallow but very wide networks.

In my thesis project, I explore whether we can go deep again. I study statistical properties of randomly-initialized fully-connected neural networks and find out that they suffer from a previously unknown problem of the vanishing empirical variance of the network outputs. I show that despite keeping the theoretical variance constant over all layers, the empirical variance converges in probability to zero. I demonstrate it theoretically for the specific case of He initialization and show empirical evidence for the same behavior in other state-of-the-art random initialization methods. The practical consequence of my results is that arbitrarily deep networks cannot be trained when initialized randomly with state-of-the-art methods. I show, however, that they can be trained when initialized with a deterministic method that avoids the problem of the vanishing empirical variance.

The thesis report consists of three parts and is based on a paper submitted to the NeurIPS 2023 conference which is currently under review. The paper and its supplementary material form the first two parts. The third part contains additional findings that were made within the work on the thesis but were not included in the NeurIPS submission.

I would like to thank my supervisors David M. J. Tax and Marco Loog for their guidance, support, and exchange of ideas throughout the thesis project. I would also like to thank the thesis committee members Marcel J. T. Reinders, Wendelin Böhmer, and David M. J. Tax.

Michał Grzejdziak, 28 May 2023

Contents

Vanishing Empirical Variance in Randomly Initialized Networks

Michał Grzejdziak
Delft University of Technology
Delft, The Netherlands
m.a.grzejdziak@student.tudelft.nl

Marco Loog
Radboud University
Nijmegen, The Netherlands
marco.loog@ru.nl

David M. J. Tax
Delft University of Technology
Delft, The Netherlands
d.m.j.tax@tudelft.nl

Abstract

Neural networks are commonly initialized to keep the theoretical variance of the hidden pre-activations constant, in order to avoid the vanishing and exploding gradient problem. Though this condition is necessary to train very deep networks, numerous analyses showed that it is not sufficient. We explain this fact by analyzing the behavior of the empirical variance which is more meaningful in practice of data sets of finite size. We demonstrate its discrepancy with the theoretical variance which grows with depth. We study the output distribution of neural networks at initialization in terms of its kurtosis which we find to grow to infinity with increasing depth even if the theoretical variance stays constant. The result of this is that the empirical variance vanishes: its asymptotic distribution converges in probability to zero. Our analysis, which studies increased dependence of outputs, focuses on fully-connected ReLU networks with He initialization, but we hypothesize that many more random weight initialization methods suffer from either vanishing or exploding empirical variance. We support this hypothesis experimentally and demonstrate the failure of state-of-the-art random initialization methods in very deep regimes.

1 Introduction

The main heuristic for deriving initialization methods for deep neural networks is to keep the theoretical variance of the output or gradient distribution constant over all hidden layers. The idea is that this ensures proper propagation of the input signal through the network and therefore mitigates the vanishing gradient problem [Hochreiter, 1991, Bengio et al., 1994]. This approach has been used to derive two initialization methods: so-called Glorot [Glorot and Bengio, 2010] and He initialization [He et al., 2015]. However, these methods are still not sufficient to train arbitrarily deep networks. Other statistical properties were and demonstrated to explode or vanish even when keeping variance constant [Hanin, 2018, Hanin and Rolnick, 2018, Burkholz and Dubatovka, 2019, Vladimirova et al., 2019], but still no initialization method has been demonstrated to work for very deep networks.

In this paper, we take another look at the consequences of keeping the theoretical variance constant and analyze distributional properties beyond it. Specifically, we analyze the dynamics of kurtosis, the fourth standardized moment, as a signal is propagated through a neural network that is He-initialized. We prove that, under mild assumptions, kurtosis of the output distribution grows to infinity with increasing depth. As we will show, the surprising effect of this is that the empirical variance has to go

Preprint. Under review.

to zero (in probability), despite the constant theoretical variance. Consequently, almost all outputs are mapped to zero by an arbitrarily deep network. Our analysis, which studies increased dependence of outputs, suggests vanishing empirical variance may concern many more random initialization schemes. We demonstrate this empirically for state-of-the-art random initialization methods for fully-connected ReLU networks. We also show that ZerO [Zhao et al., 2022], which is a deterministic method that keeps empirical variance constant, can train very deep and narrow networks. To our knowledge, we are first to demonstrate the trainability of fully-connected ReLU networks of such large depths and small widths.

In Section 2, we recall literature related to our paper. In Section 3, we define the setup which we analyze in Section 4, which also contains our main theoretical result. In Section 5, we show its practical consequences for the empirical variance of the output distribution at initialization. In Section 6, we present our experimental results. Finally, in Section 7, we discuss how our analysis extends to other types of layers and other activation functions.

2 Related work

The idea to initialize the weights by sampling them i.i.d. from a zero-mean symmetric distribution such that the variance is kept constant over all layers is known at least since the work by Bishop [1995]. Later, it was popularized later as a "trick" by LeCun et al. [1998]. Glorot and Bengio [2010] extended it to balance the need to keep the output variance and the gradient variance constant, while He et al. [2015] analyzed the specific case of ReLU activation. Further extensions of this work to the specific case of the highly popular ResNets [He et al., 2016] has been given by Zhang et al. [2019] and Bachlechner et al. [2021]. Other approaches to random weight initialization include orthogonal initialization [Saxe et al., 2014], delta-orthogonal initialization [Xiao et al., 2018], data-dependent LSUV [Mishkin and Matas, 2016] or MetaInit initialization [Dauphin and Schoenholz, 2019], and GSM initialization [Burkholz and Dubatovka, 2019]. Another approach is to initialize the weights deterministically. Examples are identity initialization [Bartlett et al., 2018] and ZerO initialization [Zhao et al., 2022]. In our paper we analyze the setup of random weight initialization, focusing on the case of He initialization [He et al., 2015]. We hypothesize that our claims extend to other random initialization methods and we demonstrate it in our experiments.

Various results indicate that controlling the variance is not sufficient to mitigate gradient problems. Hanin [2018] showed that the empirical variance of gradients grows exponentially with increasing depth, while Hanin and Rolnick [2018] and Burkholz and Dubatovka [2019] showed the same for the empirical variance of the lengths of activations and pre-activations respectively. Vladimirova et al. [2019] demonstrated that with increasing depth, the output distribution has increasingly heavy tails. We add to this line of research by studying kurtosis of the output distribution, which directly relates to its empirical variance. Our analysis shows that even if we keep the theoretical variance constant, the empirical variance will tend to zero.

Proper initialization of neural networks is only a prerequisite to ensure fast convergence to a good solution of the given optimization problem. Shamir [2019] showed that for standard random weight initialization methods, the number of iterations required to convergence grows exponentially in depth. Du and Hu [2019] reached a similar conclusion, while Hu et al. [2020] showed that the convergence speed is independent of depth for the case of orthogonal initialization. However, our work questions the possibility of convergence of very deep randomly initialized networks in practice even with initialization schemes designed to overcome the problem of large depth like orthogonal initialization [Saxe et al., 2014] or GSM initialization [Burkholz and Dubatovka, 2019].

3 Preliminaries

We consider fully-connected networks with leaky ReLU nonlinearities. For an input $\mathbf{x} \in \mathbb{R}^{w_0}$, and a neural network with depth $d \in \mathbb{N}$, widths $(w_l)_{l=0}^{d} \subset \mathbb{N}$, and negative slope $a \in \mathbb{R}$, the output

$\mathbf{y}^{(l)} \in \mathbb{R}^{w_l}$ of the lth layer is recursively defined as[1]

$$\mathbf{y}^{(0)} = \mathbf{x}, \quad \mathbf{y}^{(l)} = \mathbf{W}^{(l)}\phi_a(\mathbf{y}^{(l-1)})$$

where for all $l = 1, ..., d$ $\mathbf{W}^{(l)} \in \mathbb{R}^{w_l \times w_{l-1}}$ is a weight matrix and $\phi_a : \mathbb{R} \to \mathbb{R}$ is leaky ReLU with the negative slope parameter $a \in \mathbb{R}$, applied entry-wise

$$\phi_a(x) = \begin{cases} ax, & \text{if } x < 0, \\ x, & \text{otherwise.} \end{cases}$$

We treat $\mathbf{x}$ and $(\mathbf{W}^{(l)})_{l=1}^{d}$ as random variables and analyze distributional properties of $\mathbf{y}^{(d)}$ with increasing d. We study the initialization method by He et al. [2015] which takes the entries of each weight matrix $\mathbf{W}^{(l)}$ to be i.i.d. symmetric variables with variance $\frac{2}{w_l(a^2+1)}$. This method preserves several distributional properties of the input random vectors.

Definition 3.1 (He random vector)**.** We say that a random vector $\mathbf{x} \in \mathbb{R}^w$ is a He random vector if all variables in $\mathbf{x}$ are zero-mean, symmetric[2], uncorrelated, and homoscedastic with some variance σ_x^2.

Proposition 3.2 (He initialization)**.** *If for all $l = 1, ..., d$ weight matrices $\boldsymbol{W}^{(l)}$ are i.i.d., zero-mean, and symmetric with variance equal to $\frac{2}{w_l(a^2+1)}$, then for an input He random vector $\boldsymbol{x}$ with variance σ_x^2 the output random vector $\mathbf{y}^{(d)}$ is a He random vector with variance σ_x^2.*

A proof for Proposition 3.2 has been given by He et al. [2015] under the stronger assumption of preservation of independence of vector entries. We point out a mistake in the reasoning by He et al. [2015], which does not change their main contribution in the form of variance-preserving initialization method. He et al. [2015] assumed that independence is preserved through the network, but what actually is preserved is uncorrelatedness. With this correction, we give our proof in the supplement.

One may ask how to make sure that the properties of He random vector are satisfied at the input. The Proposition 3.3 below shows that, if we include an additional weight matrix before the first activation, it transforms any input random vector to a He random vector.

Proposition 3.3 (Any random vector can be transformed to a He random vector)**.** *For any finite-variance random vector $\boldsymbol{x} \in \mathbb{R}^w$, if $\boldsymbol{W} \in \mathbb{R}^{w \times w}$ is a random matrix of i.i.d. zero-mean, symmetric variables with finite variance such that $\boldsymbol{W}$ and $\boldsymbol{x}$ are mutually independent, then $\boldsymbol{z} = \boldsymbol{W}\boldsymbol{x}$ is a He random vector with some variance σ_z^2.*

Proof. Consider a specific entry z_i in $\mathbf{z}$, $z_i = \sum_{k=1}^{w} W_{ik}x_k$. For any i, k, W_{ik} is symmetric and zero-mean, and so must be $W_{ik}x_k$. Because z_i is a sum of zero-mean and symmetric random variables, it is zero-mean and symmetric. All entries of $\mathbf{z}$ have the same variance because it is expressed with the same formula, so they are homoscedastic with some variance σ_z^2. Lastly, we will show that z_i, z_j are uncorrelated for any i, j, $i \neq j$. Consider covariance of two entries z_i, z_j

$$\text{Cov}[z_i, z_j] = \mathbb{E}[z_i z_j] - \mathbb{E}[z_i]\mathbb{E}[z_j].$$

Because $\mathbb{E}[z_i]$ is equal to zero, it simplifies to

$$\begin{aligned}\text{Cov}[z_i, z_j] = \mathbb{E}[z_i z_j] &= \mathbb{E}[(\sum_{k=1}^{w} W_{ik}x_k)(\sum_{k=1}^{w} W_{jk}x_k)] \\ &= \mathbb{E}[\sum_{k_1=1}^{w}\sum_{k_2=1}^{w} W_{ik_1}x_{k_1}W_{jk_2}x_{k_2}] = \sum_{k_1=1}^{w}\sum_{k_2=1}^{w} \mathbb{E}[W_{ik_1}]\mathbb{E}[x_{k_1}W_{jk_2}x_{k_2}] = 0.\end{aligned}$$

□

We will assume that inputs are always He random vector and that networks are initialized according to Proposition 3.2. Effectively, the outputs for each hidden layer will be He random vectors too.

[1]Throughout the paper, for vectors and matrices we use upper indices to indicate the layer, and lower indices to refer to entries. For scalars we use the lower indices to indicate the layer. For the function ϕ_a we use the lower index to indicate the negative slope parameter a.

[2]By a symmetric random variable we mean a random variable with a probability distribution symmetric around its mean.

4 Theory

In this section we present our main theoretical results. We derive the relation between the input and the output kurtosis in a neural network (Proposition 4.3) and then prove that for bounded-width networks it grows to infinity with increasing depth (Theorem 4.7).

Here, kurtosis of a random variable x is defined as $Kurt[x] = \mathbb{E}[\frac{(x-\mathbb{E}[x])^4}{Var[x]^2}]$. We will analyze the case with $\mathbb{E}[x] = 0$ which simplifies it to $Kurt[x] = \frac{1}{Var[x]^2}\mathbb{E}[x^4]$.

First, we will prove Proposition 4.3 in which we will derive the exact recursive formula for dynamics of kurtosis over consecutive layer. The derived formula tracks two statistical properties in a linear matrix difference equation: kurtosis and covariance of squared outputs. We take the mild assumption which is satisfied with Proposition 3.3 that covariance of squared outputs is equal for any two outputs.

In the proof of Proposition 4.3 we will use two lemmas 4.1 and 4.2 that are given first. Their proofs are given in supplementary material.

Lemma 4.1. *Let x be a zero-mean, symmetric random variable with $Var[x] = \sigma_x^2$ and $Kurt[x] = \kappa_x$. Then $\mathbb{E}[\phi_a^4(x)] = \frac{(a^4+1)}{2}\sigma_x^4\kappa_x$.*

Lemma 4.2. *Let x, y be identically distributed, uncorrelated, zero-mean, symmetric random variables with variances σ_x^2, kurtoses κ_x and $Cov[x^2, y^2] = c$. Then $\mathbb{E}[\phi_a^2(x)\phi_a^2(y)] = \frac{(a^2+1)^2}{4}(\sigma_x^4 + c)$.*

Proposition 4.3. *Consider a network that is He-initialized with a distribution that has kurtosis κ_w and that has output random vectors $\boldsymbol{y}^{(l)}$ at every layer l with variance σ_x^2. Let*

$$c_l = Cov[(y_i^{(l)})^2, (y_j^{(l)})^2]$$

be the covariance between any two squared entries from $\boldsymbol{y}^{(l)}$ and let $\kappa_l = Kurt[y_i^{(l)}]$ be the kurtosis of every entry i in $\boldsymbol{y}^{(l)}$. Then the kurtoses of consecutive layers are recursively related through the linear matrix difference equation

$$\boldsymbol{k}^{(l+1)} = \boldsymbol{A}^{(l)}\boldsymbol{k}^{(l)}$$

where $\boldsymbol{k}^{(l)} = [\kappa_l, c_l, 1]^T$ and

$$\boldsymbol{A}^{(l)} = \begin{bmatrix} \frac{2(a^4+1)\kappa_w}{w_l(a^2+1)^2} & \frac{3(w_l-1)}{w_l\sigma_x^4} & \frac{3(w_l-1)}{w_l} \\ \frac{2(a^4+1)\sigma_x^4}{w_l(a^2+1)^2} & \frac{w_l-1}{w_l} & \frac{-\sigma_x^4}{w_l} \\ 0 & 0 & 1 \end{bmatrix}.$$

Consequently, the relation between the input kurtosis and the output kurtosis at depth $d+1$ is

$$\boldsymbol{k}^{(d+1)} = \left(\prod_{l=0}^{d} \boldsymbol{A}^{(d-l)}\right)\boldsymbol{k}^{(0)}.$$

Proof. We will derive the formula for κ_{l+1}, then for c_{l+1}. We have $\mathbb{E}[y_i^{(l+1)}] = 0$, $Var[y_i^{(l+1)}] = \sigma_x^2$, so $\kappa_{l+1} = Kurt[y_i^{(l+1)}] = \frac{1}{\sigma_x^4}\mathbb{E}[(y_i^{(l+1)})^4]$. We can expand $\mathbb{E}[(y_i^{(l+1)})^4] = \mathbb{E}[(\sum_{j=1}^{w_l} W_{ij}^{(l+1)}\phi_a^4(y_j^{(l)}))]$ using the multinomial theorem. Because the weight matrix entries are i.i.d., zero-mean and symmetric, the terms with the odd powers vanish. We get

$$\mathbb{E}[(y_i^{(l+1)})^4] = \sum_{j=1}^{w_l}\mathbb{E}[(W_{ij}^{(l+1)}\phi_a^4(y_j^{(l)}))] + \sum_{\substack{j,k=1 \\ j\neq k}}^{w_l}\binom{4}{2,2}\mathbb{E}[(W_{ij}^{(l+1)}\phi_a(y_j^{(l)}))^2(W_{ik}^{(l+1)}\phi_a(y_k^{(l)}))^2].$$

Using Lemma 4.1, we find that

$$\mathbb{E}[(W_{ij}^{(l+1)}\phi_a(y_j^{(l)}))^4] = \frac{4}{w_l^2(a^2+1)^2}\kappa_w\frac{(a^4+1)}{2}\sigma_x^4\kappa_l = \frac{2(a^4+1)}{w_l^2(a^2+1)^2}\kappa_w\sigma_x^4\kappa_l.$$

We can get a closed-form formula for $\mathbb{E}[(W_{ij}^{(l+1)}\phi_a(y_j^{(l)}))^2(W_{ik}^{(l+1)}\phi_a(y_k^{(l)}))^2]$ using Lemma 4.2

$$\mathbb{E}[(W_{ij}^{(l+1)}\phi_a(y_j^{(l)}))^2(W_{ik}^{(l+1)}\phi_a(y_k^{(l)}))^2] = \left(\frac{2}{w_l(a^2+1)}\right)^2 \frac{(a^2+1)^2}{4}(\sigma_x^4+c_l) = \frac{\sigma_x^4+c_l}{w_l^2}.$$

Putting the two above to the multinomial expansion of $\mathbb{E}[(y_i^{(l+1)})^4]$ given in the beginning, we get

$$\begin{aligned}\mathbb{E}[(y_i^{(l+1)})^4] &= \frac{2(a^4+1)}{w_l(a^2+1)^2}\kappa_w\sigma_x^4\kappa_l + 6\binom{w_l}{2}\frac{\sigma_x^4+c_l}{w_l^2} = \frac{2(a^4+1)}{w_l(a^2+1)^2}\kappa_w\sigma_x^4\kappa_l + 3w_l(w_l-1)\frac{\sigma_x^4+c_l}{w_l^2} \\ &= \frac{2(a^4+1)}{w_l(a^2+1)^2}\kappa_w\sigma_x^4\kappa_l + \frac{3(w_l-1)}{w_l}c_l + \frac{3(w_l-1)\sigma_x^4}{w_l}.\end{aligned}$$

Finally, we should divide $\mathbb{E}[(y_i^{(l+1)})^4]$ by σ_x^4 to get

$$\kappa_{l+1} = \frac{2(a^4+1)\kappa_w}{w_l(a^2+1)^2}\kappa_l + \frac{3(w_l-1)}{w_l\sigma_x^4}c_l + \frac{3(w_l-1)}{w_l}.$$

Next, consider $c_{l+1} = Cov[(y_i^{(l+1)})^2, (y_j^{(l+1)})^2]$ for any $i, j, i \neq j$

$$\begin{aligned}c_{l+1} &= Cov[(y_i^{(l+1)})^2, (y_j^{(l+1)})^2] = \mathbb{E}[(y_i^{(l+1)})^2(y_j^{(l+1)})^2] - \mathbb{E}[(y_i^{(l+1)})^2]\mathbb{E}[(y_j^{(l+1)})^2] \\ &= \mathbb{E}[(\sum_{k=1}^{w_l} W_{ik}^{(l+1)}\phi_a(y_k^{(l)}))^2(\sum_{k=1}^{w_l} W_{jk}^{(l+1)}\phi_a(y_k^{(l)}))^2] - \sigma_x^4 \\ &= \frac{4}{w_l^2(a^2+1)^2}\sum_{k_1=1}^{w_l}\sum_{k_2=1}^{w_l}\mathbb{E}[\phi_a^2(y_{k_1}^{(l)})\phi_a^2(y_{k_2}^{(l)})] - \sigma_x^4\end{aligned}$$

where the sum $\sum_{k_1=1}^{w_l}\sum_{k_2=1}^{w_l}\mathbb{E}[\phi_a^2(y_{k_1}^{(l)})\phi_a^2(y_{k_2}^{(l)})]$ is

$$\begin{aligned}\sum_{k_1=1}^{w_l}\sum_{k_2=1}^{w_l}\mathbb{E}[\phi_a^2(y_{k_1}^{(l)})\phi_a^2(y_{k_2}^{(l)})] &= \sum_{\substack{k_2=1 \\ k_2\neq k_1}}^{w_l}\frac{(a^2+1)^2}{4}(\sigma_x^4+c_l) + \sum_{k=1}^{w_l}\frac{a^4+1}{2}\sigma_x^4\kappa_l \\ &= \frac{w_l(w_l-1)(a^2+1)^2}{4}(\sigma_x^4+c_l) + \frac{w_l(a^4+1)}{2}\sigma_x^4\kappa_l.\end{aligned}$$

Putting it all together, we get

$$c_{l+1} = \frac{2(a^4+1)\sigma_x^4}{w_l(a^2+1)^2}\kappa_l + \frac{w_l-1}{w_l}c_l - \frac{\sigma_x^4}{w_l}.$$

and $\mathbf{k}^{(l+1)} = [\kappa_{l+1}, c_{l+1}, 1]^T$ is of the desired form. □

Now, we will show in Theorem 4.7, that with the dynamics derived in Proposition 4.3, for any valid $\mathbf{k}^{(0)}$, κ_d will grow to infinity. To this end, we will first prove three lemmas that describe the properties of matrices $\mathbf{A}^{(l)}$ and their products. In Lemma 4.4, we will show that any product of such matrices is of a form parameterized with four positive parameters. Next, in Lemma 4.5, we will show that any matrix $\mathbf{A}^{(l)}$ has a positive eigenvalue that is strictly larger than 1. The proof of Lemma 4.5 uses the Perron theorem which we provide with a reference to a proof in the supplementary material. We will combine these two properties in Lemma 4.6 to show that for any $\mathbf{A} = \mathbf{A}^{(l)}$ raised to a power m, all its positive parameters will tend to infinity with $m \to \infty$ and so its norm will tend to infinity. We will use this property in the proof of Theorem 4.7.

Lemma 4.4. *Consider the product of matrices $\boldsymbol{B} = \prod_{l=0}^{d}\boldsymbol{A}^{(d-l)}$ as given in Proposition 4.3, with $w > 1$. $\boldsymbol{B}$ is of the form*

$$\boldsymbol{B} = \begin{bmatrix}\alpha & \frac{\beta}{\sigma_x^4} & \beta \\ \gamma\sigma_x^4 & \delta & \sigma_x^4(\delta-1) \\ 0 & 0 & 1\end{bmatrix} \tag{1}$$

with $\gamma > 0$, $\alpha \geq \gamma$, $\delta > 0$, $\beta \geq \delta$.

Proof. We prove the lemma by induction on d. For $d = 0$ we have $\mathbf{B} = \mathbf{A}^{(0)}$ which is satisfied by the definition of $\mathbf{A}^{(0)}$. Assume that (1) is satisfied for some $d \in \mathbb{N}$. Denote $\mathbf{C} = \prod_{l=0}^{d} \mathbf{A}^{(d-l)}$ and $\mathbf{B} = \mathbf{A}^{(d+1)}\mathbf{C}$. We can write

$$\mathbf{A}^{(d+1)} = \begin{bmatrix} \alpha_1 & \frac{\beta_1}{\sigma_x^4} & \beta_1 \\ \gamma_1\sigma_x^4 & \delta_1 & \sigma_x^4(\delta_1 - 1) \\ 0 & 0 & 1 \end{bmatrix}, \mathbf{C} = \begin{bmatrix} \alpha_2 & \frac{\beta_2}{\sigma_x^4} & \beta_2 \\ \gamma_2\sigma_x^4 & \delta_2 & \sigma_x^4(\delta_2 - 1) \\ 0 & 0 & 1 \end{bmatrix}$$

with $\forall_{i=1,2}$, $\gamma_i > 0$, $\alpha_i \geq \gamma_i$, $\delta_i > 0$, $\beta_i \geq \delta_i$. $\mathbf{A}^{(d+1)}\mathbf{C}$ is equal to

$$\begin{bmatrix} \alpha_1\alpha_2 + \beta_1\gamma_2 & \frac{\alpha_1\beta_2 + \beta_1\delta_2}{\sigma_x^4} & \alpha_1\beta_2 + \beta_1\delta_2 \\ (\gamma_1\alpha_2 + \delta_1\gamma_2)\sigma_x^4 & \gamma_1\beta_2 + \delta_1\delta_2 & \sigma_x^4(\gamma_1\beta_2 + \delta_1\delta_2 - 1) \\ 0 & 0 & 1 \end{bmatrix}.$$

If we set $\alpha = \alpha_1\alpha_2 + \beta_1\gamma_2$, $\beta = \alpha_1\beta_2 + \beta_1\delta_2$, $\gamma = \gamma_1\alpha_2 + \delta_1\gamma_2$, $\delta = \gamma_1\beta_2 + \delta_1\delta_2$, we get that $\mathbf{B} = \mathbf{A}^{(l+1)}\mathbf{C}$ is of the desired form with

$$\begin{aligned} \gamma = \gamma_1\alpha_2 + \delta_1\gamma_2 > 0, \quad \alpha = \alpha_1\alpha_2 + \beta_1\gamma_2 \geq \gamma > 0, \\ \delta = \gamma_1\beta_2 + \delta_1\delta_2 > 0, \quad \beta = \alpha_1\beta_2 + \beta_1\delta_2 \geq \delta > 0. \end{aligned}$$

□

Lemma 4.5. *The largest eigenvalue of any matrix $\boldsymbol{A}^{(l)}$ from Proposition 4.3 is larger than 1.*

Proof. Consider the matrix $\mathbf{A}^{(l)}$ for some $l = 0, ..., d$. Its characteristic polynomial is of the form

$$\det(\mathbf{A}^{(l)} - \lambda I) = (\lambda^2 + b\lambda + c)(1 - \lambda)$$

where $\lambda^2 + b\lambda + c$ is the characteristic polynomial of a matrix $\mathbf{A}_{-}^{(l)}$ equal to $\mathbf{A}^{(l)}$ but with row 3 and column 3 removed. Matrix $\mathbf{A}_{-}^{(l)}$ is positive and by the Perron theorem it has two distinct real eigenvalues λ_{max} and λ_{min} such that $\lambda_{max} > 0$ and $\lambda_{max} > |\lambda_{min}|$. We will now show that $\lambda_{max} > 1$.

Express λ_{max} using trace and determinant of $\mathbf{A}_{-}^{(l)}$, $\lambda_{max} = \frac{\text{tr}(\mathbf{A}_{-}^{(l)}) + \sqrt{\text{tr}^2(\mathbf{A}_{-}^{(l)}) - 4\det(\mathbf{A}_{-}^{(l)})}}{2}$, where $\text{tr}(\mathbf{A}_{-}^{(l)})$ and $\det(\mathbf{A}_{-}^{(l)})$ are

$$\text{tr}(\mathbf{A}_{-}^{(l)}) = \frac{2(a^4+1)\kappa_w}{w_l(a^2+1)^2} - \frac{1}{w_l} + 1, \quad \det(\mathbf{A}_{-}^{(l)}) = \frac{2(a^4+1)(w_l - 1)(\kappa_w - 3)}{w_l^2(a^2+1)^2}.$$

We consider two cases for $\text{tr}(\mathbf{A}_{-}^{(l)})$ and show that in both of them $\lambda_{max} > 1$. If $\text{tr}(\mathbf{A}_{-}^{(l)}) \geq 2$, then $\lambda_{max} > 1$ because $\lambda_{max} > \frac{\text{tr}(\mathbf{A}_{-}^{(l)})}{2}$. Otherwise, if $1 \leq \text{tr}(\mathbf{A}_{-}^{(l)}) < 2$, then

$$\begin{aligned} \lambda_{max}(\mathbf{A}_{-}^{(l)}) > 1 &\Leftrightarrow \sqrt{\text{tr}^2(\mathbf{A}_{-}^{(l)}) - 4\det(\mathbf{A}_{-}^{(l)})} > 2 - \text{tr}(\mathbf{A}_{-}^{(l)}) \\ &\Leftrightarrow \text{tr}^2(\mathbf{A}_{-}^{(l)}) - 4\det(\mathbf{A}_{-}^{(l)}) > 4 - 4\text{tr}(\mathbf{A}_{-}^{(l)}) + \text{tr}^2(\mathbf{A}_{-}^{(l)}) \Leftrightarrow \text{tr}(\mathbf{A}_{-}^{(l)}) > \det(\mathbf{A}_{-}^{(l)}) + 1 \end{aligned}$$

which is always satisfied, because for $\kappa_w < 3$, we have $\det(\mathbf{A}_{-}^{(l)}) + 1 < 1 \leq \text{tr}(\mathbf{A}_{-}^{(l)})$, and for $\kappa_w \geq 3$

$$\det(\mathbf{A}_{-}^{(l)}) = \frac{2(a^4+1)(w_l - 1)(\kappa_w - 3)}{w_l^2(a^2+1)^2} < \frac{2(a^4+1)(\kappa_w - 3)}{w_l(a^2+1)^2} < \frac{2(a^4+1)(\kappa_w - 1)}{w_l(a^2+1)^2} \leq \text{tr}(\mathbf{A}_{-}^{(l)}) - 1.$$

This proves that $\lambda_{max} > 1$. □

Lemma 4.6. *Consider the matrix $\boldsymbol{A} = \boldsymbol{A}^{(l)}$ from Proposition 4.3 raised to the power m. If $m \to \infty$, then α_m, β_m, γ_m, δ_m from the representation of $\boldsymbol{A}^m$ in the form from Lemma 4.4 go to infinity.*

Proof. By Lemma 4.5 $\lambda_{max}(\mathbf{A}) > 1$ so $\lim_{m\to\infty} ||\mathbf{A}^m|| = \infty$, so it must be that at least one of α_m, β_m, γ_m, δ_m goes to infinity. Consider four cases:

1. Assume α_m tends to infinity. From the proof of Lemma 4.4, $\gamma_{m+1} = \gamma_1\alpha_m + \delta_1\gamma_m > \gamma_1\alpha_m$, so γ_m must tend to infinity. In the same way, $\delta_{m+1} = \gamma_m\beta_1 + \delta_m\delta_2 > \gamma_m\beta_1$, so δ_m must tend to infinity too. Because $\beta_m > \delta_m$, β_m must tend to infinity as well.

2. Assume β_m tends to infinity. From the proof of Lemma 4.4, $\alpha_{m+1} = \alpha_m\alpha_1 + \beta_m\gamma_1$, so α_m must tend to infinity, and so γ_m and δ_m as shown above in 1.

3. Assume γ_m tends to infinity. Then α_m must tend to infinity because $\alpha_m \geq \gamma_m$ for any m, and so β_m and δ_m must tend to infinity as shown above in 1.

4. Assume δ_m tends to infinity. Then β_m must tend to infinity because $\beta_m \geq \delta_m$ for any m, and so α_m and γ_m must tend to infinity as shown above in 2.

□

Theorem 4.7. *For any He-initialized network with widths bounded from below by 2 and from above by some w_{max}, the output distribution kurtosis grows to infinity with increasing depth for any input He random vector.*

Proof. We can express the vector $\mathbf{k}^{(d+1)}$ at depth $d+1$ as $\mathbf{k}^{(d+1)} = \mathbf{B}^{(d)}\mathbf{k}^{(0)}$ with $\mathbf{B}^{(d)} = \prod_{l=0}^{d}\mathbf{A}^{(d-l)}$ parameterized by α_d, β_d, γ_d, δ_d from Lemma 4.4. We can write that

$$\kappa_{d+1} = \alpha_d\kappa_0 + \frac{\beta_d}{\sigma_x^4}c_0 + \beta_d.$$

Note that it must be that $c_0 \geq -\sigma_x^4$, because

$$c_0 = Cov[(y_i^{(0)})^2(y_j^{(0)})^2] = \mathbb{E}[(y_i^{(0)})^2(y_j^{(0)})^2] - \mathbb{E}[y_i^{(0)}]^2\mathbb{E}[y_j^{(0)}]^2 = \mathbb{E}[(y_i^{(0)})^2(y_j^{(0)})^2] - \sigma_x^4 \geq -\sigma_x^4.$$

We can consider the output of the first layer as the actual input, so we can even say that $c_0 = \sigma_x^4(-1+\epsilon)$ for some $\epsilon > 0$, because $c_1 = \gamma_1\sigma_x^4\kappa_0 + \delta_1 c_0 + \sigma_x^4(\delta_1 - 1) > \gamma_1\sigma_x^4\kappa_0 - \sigma_x^4$ for some $\gamma_1 > 0$ and $\delta_1 > 0$.

We can write then that

$$\kappa_{d+1} = \alpha_d\kappa_0 + \frac{\beta_d}{\sigma_x^4}c_0 + \beta_d = \alpha_d\kappa_0 + \beta_d\epsilon.$$

To know that κ_{d+1} goes to infinity with $d \to \infty$ it is enough to show that $\lim_{d\to\infty}||\mathbf{B}^{(d)}|| = \infty$, because it would imply that one of α_d, β_d, γ_d or δ_d goes to infinity, in which case κ_{d+1} goes to infinity. We will show that $\lim_{d\to\infty}\lambda_{max}(\mathbf{B}^{(d)}) = \infty$ which implies that $\lim_{d\to\infty}||\mathbf{B}^{(d)}|| = \infty$. Because for any two matrices $\mathbf{M}_1$ and $\mathbf{M}_2$ $\lambda_{max}(\mathbf{M}_1\mathbf{M}_2) = \lambda_{max}(\mathbf{M}_2\mathbf{M}_1)$, we can consider λ_{max} of a rearranged matrix product

$$\lambda_{max}(\mathbf{B}^{(d)}) = \lambda_{max}\left(\prod_{l=0}^{d}\mathbf{A}^{(l)}\right) = \lambda_{max}\left(\prod_{w=2}^{w_{max}}\mathbf{A}_w^{m_w}\right)$$

where $\mathbf{A}_w$ denotes a matrix $\mathbf{A}^{(l)}$ from Proposition 4.3 for a specific width w, and m_w the number of occurrences of such matrices until depth d. With $d \to \infty$ there will be at least one w for which $m_w \to \infty$. For such widths w, $\mathbf{A}_w^{m_w}$ will behave according to Lemma 4.6. The product $\prod_{w=2}^{w_{max}}\mathbf{A}_w^{m_w}$ will consist of a finite number of matrices of the form from Lemma 4.4 and at least one matrix with all positive parameters from Lemma 4.4 going to infinity. In effect, the positive parameters from Lemma 4.4 for this product will go to infinity, which implies that $\lambda_{max}\left(\prod_{w=2}^{w_{max}}\mathbf{A}_w^{m_w}\right)$ will go to infinity. As a result, with $d \to \infty$, $\lambda_{max}(\mathbf{B}^{(d)})$ will go to infinity. □

We set the width to satisfy $w > 1$, but the same can be proven allowing for $w = 1$. This requires another assumption that either $|a| \neq 1$ or $\kappa_w \neq 1$.

5 Exploding kurtosis and vanishing empirical variance

Theorem 4.7 has important consequences for He-initialized networks. Although the theoretical variance is kept constant over all hidden layers, for deep enough networks we will typically observe the empirical variance at the output to be close to zero. This stems from the relation between the kurtosis κ and the empirical variance distribution S_n^2. The variance of S_n^2 depends on kurtosis as[3] $Var[S_n^2] = \left(\kappa - \frac{n-3}{n-1}\right)\frac{\sigma^4}{n}$ where n is the sample size, κ is kurtosis and σ^2 is the theoretical variance. For a large n, we can approximate the ratio distribution of $\frac{S_n^2}{\sigma^2}$ as $\frac{S_n^2}{\sigma^2} \sim \frac{\chi^2(DF_n)}{DF_n}$ with $DF_n = \frac{2\sigma^4}{Var[S_n^2]} = \frac{2n}{\kappa - \frac{n-3}{n-1}}$. This can be alternatively expressed in terms of the gamma distribution $\frac{S_n^2}{\sigma^2} \sim \Gamma(k = \frac{DF_n}{2}, \theta = \frac{2}{DF_n})$. With kurtosis κ growing to infinity, DF_n for any $n \in \mathbb{N}$ shrinks to zero so the shape parameter k shrinks to zero and the scale parameter $\theta = \frac{1}{k}$ grows to infinity. The probability density function for this distribution is given as $f(x; k, \theta) = f(x; k, \frac{1}{k}) = \frac{x^{k-1}e^{-kx}k^k}{\Gamma(k)}$. For any $x > 0$, with $k \to 0$, this converges to zero because the numerator converges to a constant and the denominator grows to infinity. The speed of convergence is faster for large x.

Despite keeping the theoretical variance constant, the empirical variance distribution converges in probability to zero, because as shown by Theorem 4.7 the output kurtosis grows to infinity with increasing depth. In practice, all inputs will be mapped arbitrarily close to zero. This will make training impossible as no gradient will be propagated through networks with outputs zeroed-out.

6 Experiments

We proved in Theorem 4.7 that He initialization suffers from the vanishing empirical variance problem. We hypothesize that problems with empirical variance concern all fully random initialization methods which initialize weight matrices with off-diagonal entries, because this induces increased dependence of outputs. If the theoretical variance is kept constant or decreases, the empirical variance vanishes, otherwise it explodes. In the next sections, we present empirical evidence that supports this hypothesis. We verify it experimentally for five state-of-the-art random initialization methods: Glorot by Glorot and Bengio [2010], He by He et al. [2015], orthogonal by Saxe et al. [2014], GSM by Burkholz and Dubatovka [2019], and MetaInit by Dauphin and Schoenholz [2019]. We also demonstrate that ZerO proposed by Zhao et al. [2022] is superior over all these methods in very deep regimes.

All experiments are performed on constant-width ReLU networks on CIFAR10 [Krizhevsky, 2009]. The inputs are preprocessed so that the means of all channels are zero and the variances are one.

The experiments were run on a machine with a single Intel i7-11850H CPU. The anonymized code is available at `https://drive.google.com/file/d/1GIH4WOgJCCjreLW6v0MPyRMb9xGKarqv/view?usp=share_link`.

6.1 Empirical evidence for the problems with empirical variance

We estimate quantiles of the output empirical variance distribution for different initialization methods over 10,000 neural networks (1,000 for MetaInit) of constant width $w = 10$ and depth $d = 100$, given the whole CIFAR10 training set as input. The quantile plots for probabilities 0.9, 0.99 and 0.999 over layer depth are given in Figure 1. For all considered random initialization methods except MetaInit, we observe that 90% randomly initialized networks will have empirical variance lower than 10^{-3} after 80 layers and all quantiles monotonously decrease after 40 layers. For MetaInit, empirical variances explode. On the other hand, ZerO, which initializes most layers to identities, keeps empirical variance constant after the first layer.

6.2 Practical significance of the problems with empirical variance

We trained neural networks at varying depths to verify that random initialization methods suffer from the problems with empirical variance in practice. We evaluated their test accuracy after 500 gradient steps. As an optimizer we used Adam [Kingma and Ba, 2015] with $\beta_1 = 0.9$, and $\beta_2 = 0.999$ with

[3] We refer to [O'Neill, 2014] for a detailed treatment and proofs.

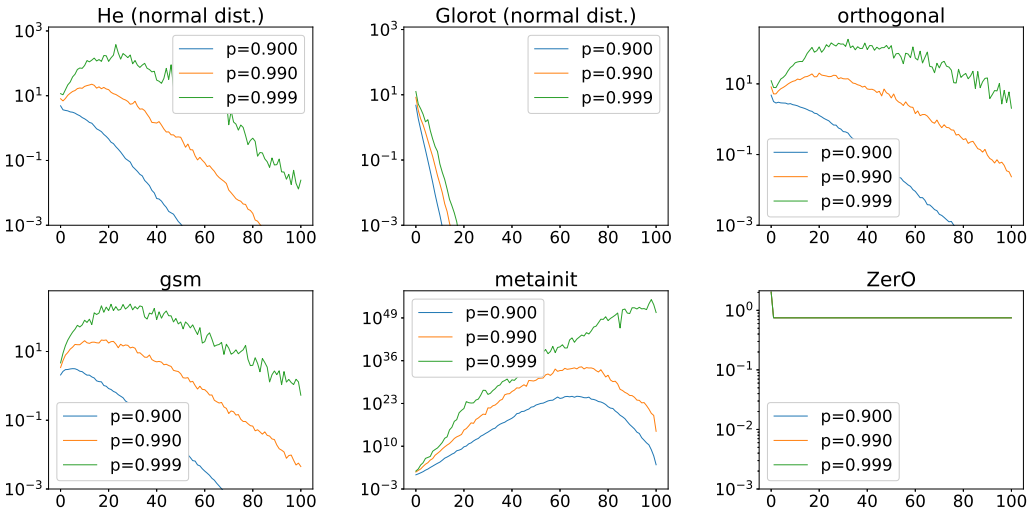

Figure 1: Estimated quantiles of the output empirical variance distribution over depth given the whole CIFAR10. The plots use logarithmic scale. ZerO is deterministic, so all its quantiles are equal.

no weight decay. We trained networks for two widths: 1) width 10 and depths from 0 to 100 with a step of 10 and learning rate of 10^{-4}, 2) width 200 and depths from 0 to 500 with a step of 50 and learning rate of 10^{-5}. The results are given in Figure 2. We can see that all random initialization methods fail to train in very deep regimes and are inferior to ZerO which does not suffer from the problems with empirical variance.

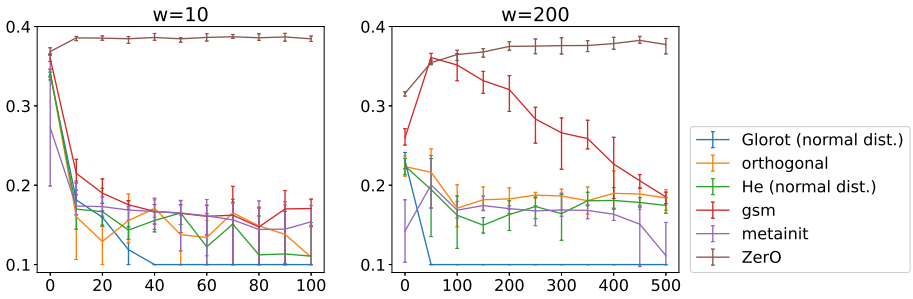

Figure 2: Test accuracy after 500 gradient steps over network depth for fully-connected constant-width ReLU networks trained on CIFAR10. The curves indicate the means and the bars indicate the minima and maxima over 5 repetitions.

7 Discussion

By analysing the dynamics of kurtosis in He-initialized networks, we identified two new problems in very deep neural networks: the exploding kurtosis and the vanishing empirical variance problems. Our experiments show that problems with either exploding or vanishing empirical variance concern not only He initialization but also other state-of-the-art random initialization methods like Glorot [Glorot and Bengio, 2010], GSM [Burkholz and Dubatovka, 2019], or MetaInit [Dauphin and Schoenholz, 2019]. All these methods fail to train very deep networks. Our experiments show that it is possible to train very deep networks with deterministic initialization methods like ZerO [Zhao et al., 2022].

We analyzed fully-connected ReLU networks, but we hypothesize that our main result about exploding kurtosis extends to many other setups. Addition of skip connections cannot stop the growth of kurtosis,

because kurtosis explodes even for the networks with no activation function, which is equivalent to setting the negative slope parameter a to 1 in our analysis. Convolutional layers induce even more dependence of layer outputs due to parameter sharing, so we expect the output kurtosis to grow at even faster pace. It is unclear whether using activation functions other than leaky ReLU could mitigate the issue. Bounded activation functions like tanh or sigmoid reduce the theoretical variance of their inputs. The impact of these activations on kurtosis is unclear. Whether there exist an architecture which by its design would prevent the growth of kurtosis requires further research.

References

Thomas Bachlechner, Bodhisattwa Prasad Majumder, Henry Mao, Gary Cottrell, and Julian McAuley. Rezero is all you need: fast convergence at large depth. In Cassio de Campos and Marloes H. Maathuis, editors, *Proceedings of the Thirty-Seventh Conference on Uncertainty in Artificial Intelligence*, volume 161 of *Proceedings of Machine Learning Research*, pages 1352–1361. PMLR, 27–30 Jul 2021. URL https://proceedings.mlr.press/v161/bachlechner21a.html.

Peter Bartlett, Dave Helmbold, and Philip Long. Gradient descent with identity initialization efficiently learns positive definite linear transformations by deep residual networks. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 521–530. PMLR, 10–15 Jul 2018. URL https://proceedings.mlr.press/v80/bartlett18a.html.

Y. Bengio, P. Simard, and P. Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, 5(2):157–166, 1994. doi: 10.1109/72.279181.

Christopher M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, Inc., USA, 1995. ISBN 0198538642.

Rebekka Burkholz and Alina Dubatovka. Initialization of relus for dynamical isometry. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/file/d9731321ef4e063ebbee79298fa36f56-Paper.pdf.

Yann N Dauphin and Samuel Schoenholz. Metainit: Initializing learning by learning to initialize. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/file/876e8108f87eb61877c6263228b67256-Paper.pdf.

Simon Du and Wei Hu. Width provably matters in optimization for deep linear neural networks. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 1655–1664. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/du19a.html.

Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In Yee Whye Teh and Mike Titterington, editors, *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pages 249–256, Chia Laguna Resort, Sardinia, Italy, 13–15 May 2010. PMLR. URL https://proceedings.mlr.press/v9/glorot10a.html.

Boris Hanin. Which neural net architectures give rise to exploding and vanishing gradients? In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper/2018/file/13f9896df61279c928f19721878fac41-Paper.pdf.

Boris Hanin and David Rolnick. How to start training: The effect of initialization and architecture. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper/2018/file/d81f9c1be2e08964bf9f24b15f0e4900-Paper.pdf.

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, 2015. doi: 10.1109/ICCV.2015.123.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. doi: 10.1109/CVPR.2016.90.

Sepp Hochreiter. Untersuchungen zu dynamischen neuronalen Netzen. *Master's thesis, Institut fur Informatik, Technische Universitat, Munchen*, 1:1–150, 1991.

Wei Hu, Lechao Xiao, and Jeffrey Pennington. Provable benefit of orthogonal initialization in optimizing deep linear networks. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=rkgqN1SYvr.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009. URL https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf.

Yann LeCun, Leon Bottou, Genevieve B. Orr, and Klaus Robert Müller. Efficient backprop. In Genevieve B. Orr and Klaus-Robert Müller, editors, *Neural Networks: Tricks of the Trade*, pages 9–50. Springer Berlin Heidelberg, Berlin, Heidelberg, 1998. ISBN 978-3-540-49430-0. doi: 10.1007/3-540-49430-8_2. URL https://doi.org/10.1007/3-540-49430-8_2.

Dmytro Mishkin and Jiri Matas. All you need is a good init. In *International Conference on Learning Representations*, 2016.

B. O'Neill. Some useful moment results in sampling problems. *The American Statistician*, 68 (4):282–296, 2014. doi: 10.1080/00031305.2014.966589. URL https://doi.org/10.1080/00031305.2014.966589.

A Saxe, J McClelland, and S Ganguli. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. In *International Conference on Learning Representations*, 2014.

Ohad Shamir. Exponential convergence time of gradient descent for one-dimensional deep linear neural networks. In Alina Beygelzimer and Daniel Hsu, editors, *Proceedings of the Thirty-Second Conference on Learning Theory*, volume 99 of *Proceedings of Machine Learning Research*, pages 2691–2713. PMLR, 25–28 Jun 2019. URL https://proceedings.mlr.press/v99/shamir19a.html.

Mariia Vladimirova, Jakob Verbeek, Pablo Mesejo, and Julyan Arbel. Understanding priors in Bayesian neural networks at the unit level. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 6458–6467. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/vladimirova19a.html.

Lechao Xiao, Yasaman Bahri, Jascha Sohl-Dickstein, Samuel Schoenholz, and Jeffrey Pennington. Dynamical isometry and a mean field theory of CNNs: How to train 10,000-layer vanilla convolutional neural networks. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 5393–5402. PMLR, 10–15 Jul 2018. URL https://proceedings.mlr.press/v80/xiao18a.html.

Hongyi Zhang, Yann N. Dauphin, and Tengyu Ma. Residual learning without normalization via better initialization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=H1gsz30cKX.

Jiawei Zhao, Florian Tobias Schaefer, and Anima Anandkumar. Zero initialization: Initializing neural networks with only zeros and ones. *Transactions on Machine Learning Research*, 2022. URL https://openreview.net/forum?id=1AxQpKmiTc.

Vanishing Empirical Variance in Randomly Initialized Networks: Supplementary Material

A Proof of Proposition 3.2

We first prove the following lemma.

Lemma. *Let x be a zero-mean, symmetric random variable with $Var[x] = \sigma_x^2$. Then $\mathbb{E}[\phi_a^2(x)] = \frac{(a^2+1)}{2}\sigma_x^2$.*

Proof.

$$\begin{aligned}\mathbb{E}[\phi_a^2(x)] &= \int_{-\infty}^{\infty} \phi_a^2(x)p(x)dx = \int_{-\infty}^{0} a^2x^2p(x)dx + \int_{0}^{\infty} x^2p(x)dx \\ &= a^2\int_{-\infty}^{0} x^2p(x)dx + \int_{0}^{\infty} x^2p(x)dx = \frac{1}{2}a^2\int_{-\infty}^{\infty} x^2p(x)dx + \frac{1}{2}\int_{-\infty}^{\infty} x^2p(x)dx \\ &= \frac{a^2+1}{2}\sigma_x^2\end{aligned}$$

□

Below, we prove Proposition 3.2.

Proposition (He initialization)**.** *If for all $l = 1, ..., d$ weight matrices $\boldsymbol{W}^{(l)}$ are i.i.d., zero-mean, and symmetric with variance equal to $\frac{2}{w_l(a^2+1)}$, then for an input He random vector $\boldsymbol{x}$ with variance σ_x^2 the output random vector $\boldsymbol{y}^{(d)}$ is a He random vector with variance σ_x^2.*

Proof. Consider a He random vector $\mathbf{x}$ with variance σ_x^2 as input. We prove the proposition by induction on d starting from the base case of $d = 0$ which is satisfied by assumptions on the input vector. Assume that it holds for some l. For $l+1$, we have $\mathbf{y}^{(l+1)} = \mathbf{W}^{(l+1)}\phi_a(\mathbf{y}^{(l)})$. Consider a specific entry $y_k^{(l+1)} = \sum_{i=1}^{w_l} W_{ki}^{(l+1)}\phi_a(y_i^{(l)})$. It is symmetric as it is a sum of symmetric random variables. As it is a sum of uncorrelated variables, its variance is sum of variances of the summands

$$\begin{aligned}Var[y_k^{(l+1)}] &= \sum_{i=1}^{w_l} Var[W_{ki}^{(l+1)}\phi_a(y_i^{(l)})] = \sum_{i=1}^{w_l} \mathbb{E}[(W_{ki}^{(l+1)})^2]\mathbb{E}[\phi_a^2(y_i^{(l)})] - \mathbb{E}[W_{ki}^{(l+1)}]^2\mathbb{E}[\phi_a(y_i^{(l)})]^2 \\ &= \sum_{i=1}^{w_l} Var[W_{ki}^{(l+1)}]\frac{(a^2+1)}{2}\sigma_x^2 = \sum_{i=1}^{w_l} \frac{2}{(a^2+1)w_l}\frac{(a^2+1)}{2}\sigma_x^2 = \sum_{i=1}^{w_l}\frac{\sigma_x^2}{w_l} = \sigma_x^2.\end{aligned}$$

Preprint. Under review.

Lastly, consider $Cov[y_k^{(l+1)}, y_j^{(l+1)}]$ for $k \neq j$

$$Cov[y_k^{(l+1)}, y_j^{(l+1)}] = \mathbb{E}[y_k^{(l+1)} y_j^{(l+1)}] - \mathbb{E}[y_k^{(l+1)}]\mathbb{E}[y_j^{(l+1)}] = \mathbb{E}[(\sum_{i=1}^{w_l} W_{ki}^{(l+1)} \phi_a(y_i^{(l)}))(\sum_{i=1}^{w_l} W_{ji}^{(l+1)} \phi_a(y_i^{(l)}))]$$
$$= \mathbb{E}[\sum_{i_0=1}^{w_l} \sum_{i_1=1}^{w_l} W_{ki_0}^{(l+1)} \phi_a(y_{i_0}^{(l)}) W_{ji_1}^{(l+1)} \phi_a(y_{i_1}^{(l)})] = \sum_{i_0=1}^{w_l} \sum_{i_1=1}^{w_l} \mathbb{E}[W_{ki_0}^{(l+1)} \phi_a(y_{i_0}^{(l)}) W_{ji_1}^{(l+1)} \phi_a(y_{i_1}^{(l)})]$$
$$= \sum_{i_0=1}^{w_l} \sum_{i_1=1}^{w_l} \mathbb{E}[W_{ki_0}^{(l+1)}]\mathbb{E}[\phi_a(y_{i_0}^{(l)}) W_{ji_1}^{(l+1)} \phi_a(y_{i_1}^{(l)})] = 0.$$

So all entries in $\mathbf{y}_{l+1}$ are uncorrelated. □

B Proof of Lemma 4.1

Lemma. *Let x be a zero-mean, symmetric random variable with $Var[x] = \sigma_x^2$ and $Kurt[x] = \kappa_x$. Then $\mathbb{E}[\phi_a^4(x)] = \frac{(a^4+1)}{2}\sigma_x^4 \kappa_x$.*

Proof.

$$\mathbb{E}[\phi_a^4(x)] = \int_{-\infty}^{\infty} \phi_a^4(x) p(x) dx = \int_{-\infty}^{0} a^4 x^4 p(x) dx + \int_{0}^{\infty} x^4 p(x) dx$$
$$= \frac{1}{2} a^4 \int_{-\infty}^{\infty} x^4 p(x) dx + \frac{1}{2} \int_{-\infty}^{\infty} x^4 p(x) dx = \frac{a^4+1}{2} \mathbb{E}[x^4] = \frac{a^4+1}{2} \sigma_x^4 \kappa_x.$$

□

C Proof of Lemma 4.2

Lemma. *Let x, y be identically distributed, uncorrelated, zero-mean, symmetric random variables with variances σ_x^2, kurtoses κ_x and $Cov[x^2, y^2] = c$. Then $\mathbb{E}[\phi_a^2(x)\phi_a^2(y)] = \frac{(a^2+1)^2}{4}(\sigma_x^4 + c)$.*

Proof.

$$\mathbb{E}[\phi_a^2(x)\phi_a^2(y)] = \int_{\mathbb{R}_+^2} x^2 y^2 p(x,y) dx dy + 2a^2 \int_{\mathbb{R}_+ \times \mathbb{R}_-} x^2 y^2 p(x,y) dx dy + a^4 \int_{\mathbb{R}_-^2} x^2 y^2 p(x,y) dx dy$$
$$= \frac{1}{4}(a^2+1)^2 \mathbb{E}[x^2 y^2]$$

Recall that $Cov[x^2, y^2] = \mathbb{E}[x^2 y^2] - \mathbb{E}[x^2]\mathbb{E}[y^2]$ so $\mathbb{E}[x^2 y^2] = Cov[x^2, y^2] + \mathbb{E}[x^2]\mathbb{E}[y^2]$. We get as a result

$$\mathbb{E}[\phi_a^2(x)\phi_a^2(y)] = \frac{(a^2+1)^2}{4}(\sigma_x^4 + c).$$

□

D Perron theorem

We provide the Perron theorem as given in Horn and Johnson [2013]. We refer to this book for further details and proofs.

Theorem (Perron)**.** *Let $\boldsymbol{A}$ be a $n \times n$ matrix which is irreducible and nonnegative and $n \geq 2$. Let $\rho(\boldsymbol{A})$ denote the spectral radius of $\boldsymbol{A}$. Then:*

1. $\rho(\boldsymbol{A}) > 0$,
2. $\rho(\boldsymbol{A})$ *is an algebraically simple eigenvalue of* $\boldsymbol{A}$,
3. *there is a unique real vector* $\boldsymbol{x}$ *such that* $\boldsymbol{Ax} = \rho(\boldsymbol{A})\boldsymbol{x}$ *and* $x_1 + x_2 + ... + x_n = 1$*; this vector is positive,*
4. *there is a unique real vector* $\boldsymbol{y}$ *such that* $\boldsymbol{y}^T\boldsymbol{A} = \boldsymbol{y}^T\rho(\boldsymbol{A})$ *and* $y_1 + y_2 + ... + y_n = 1$*; this vector is positive,*
5. $|\lambda| < \rho(\boldsymbol{A})$ *for every eigenvalue* λ *of* $\boldsymbol{A}$ *such that* $\lambda \neq \rho(\boldsymbol{A})$,
6. $(\rho(\boldsymbol{A})^{-1}\boldsymbol{A})^m \to \boldsymbol{xy}^T$ *as* $m \to \infty$.

E Negative correlation between kurtosis at initialization and test performance

We performed additional experiments on MNIST [LeCun et al., 1998] and CIFAR10 [Krizhevsky, 2009] to illustrate the negative impact of high output kurtosis at initialization on training. In the case of He initialization, it is possible to calculate the theoretical output kurtosis recursively applying the formula from Proposition 4.3, given κ_0 and c_0 for the input He random vector. We estimated the values for MNIST and CIFAR10 using 10^7 random samples. For MNIST we got $c_0 = 0.71$, $\kappa_0 = 3.95$ and for CIFAR10 we got $c_0 = 0.10$, $\kappa_0 = 3.28$. We trained networks of various depths and widths to observe relation between different values of kurtosis at initialization and test accuracy. We trained constant-width He-initialized networks twice for each of all tuples (width, depth, initialization distribution) for widths and depths from 5 to 50 with step of 5, and the weight initialization distributions Bernoulli ($\kappa_w = 1$), uniform ($\kappa_w = 1.8$), normal ($\kappa_w = 3$). We used Adam optimizer [Kingma and Ba, 2015] with learning rate of 10^{-4}, $\beta_1 = 0.9$, $\beta_2 = 0.999$ and no weight decay. We plotted test accuracy after 500 gradient steps over output kurtosis at initialization. The results are given in Figure 1. From the plots we can see that networks with large output kurtosis at initialization cannot be effectively trained.

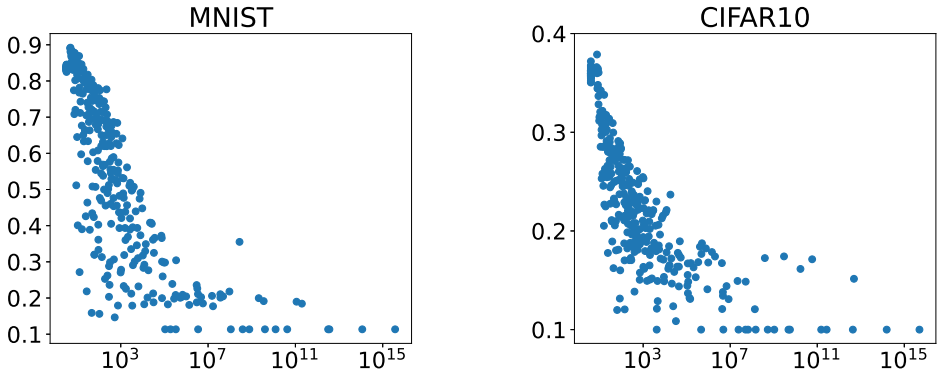

Figure 1: Test accuracy after 500 gradient steps vs output distribution kurtosis at initialization for networks of varying widths, depths and initialization distributions trained on MNIST and CIFAR10. Results for 330 experiments per dataset.

References

Roger A. Horn and Charles R. Johnson. *Matrix Analysis*. Cambridge University Press, Cambridge, 2nd edition, 2013. ISBN 9780521839402.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009. URL `https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf`.

Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. doi: 10.1109/5.726791.

Vanishing Empirical Variance in Randomly Initialized Networks: Supplementary Material Beyond the NeurIPS submission

A ZerO*

We performed the same experiments as in Figure 2 in the NeurIPS paper, but on the MNIST [LeCun et al., 1998] dataset instead of CIFAR10 [Krizhevsky, 2009]. The results are given in Figure 1. We can see that ZerO fails in this case for very deep networks. We hypothesize that the reason for is that ZerO initializes dimension-decreasing layers to truncated identities, so it does not propagate all meaningful information from the MNIST input at initialization.

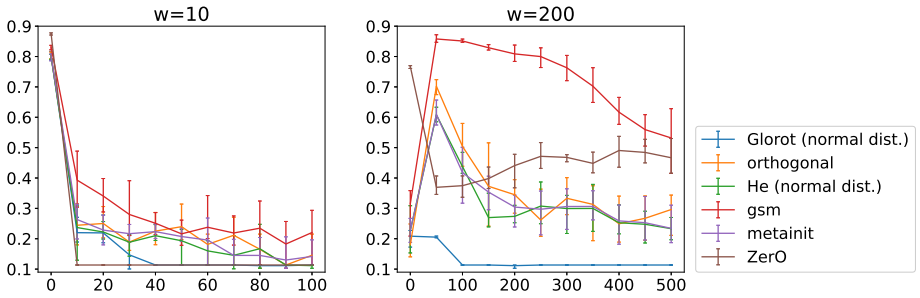

Figure 1: Test accuracy after 500 gradient steps over network depth for fully-connected constant-width ReLU networks trained on MNIST. The curves indicate the means and the bars indicate the minima and maxima over 5 repetitions.

To mitigate this issue, we propose a slight modification to ZerO, which we call ZerO*. In ZerO*, we initialize the first layer randomly by sampling weights i.i.d. from a zero-mean symmetric distribution with the variance of $\frac{1}{w_0}$. All other layers are initialized as in ZerO. ZerO* is not purely deterministic, but its random part changes empirical variance and kurtosis over a single layer only, so ZerO* does not suffer from neither exploding nor vanishing empirical variance problem.

We repeated experiments from Figure 1 to compare ZerO and ZerO*. The results are given in Figure 2 for MNIST. We can see that ZerO* is superior to ZerO and random initialization methods considered in Figure 1.

We also repeated experiments for ZerO and ZerO* from Figure 2 from the paper on CIFAR10. The results are given in Figure 3. We can see that ZerO* has more randomness, but its performance is not degraded compared to ZerO.

B ReLU networks converge to Dirac-delta function

For the special case of ReLU networks with negative slope $a = 0$, we can prove that they converge to Dirac-delta function in probability with increasing depth. This is an independent result from the one concerning the dynamics of kurtosis which was discussed in the paper, but its consequence for very deep networks is practically the same: they cannot be trained because their output is zero for almost all outputs.

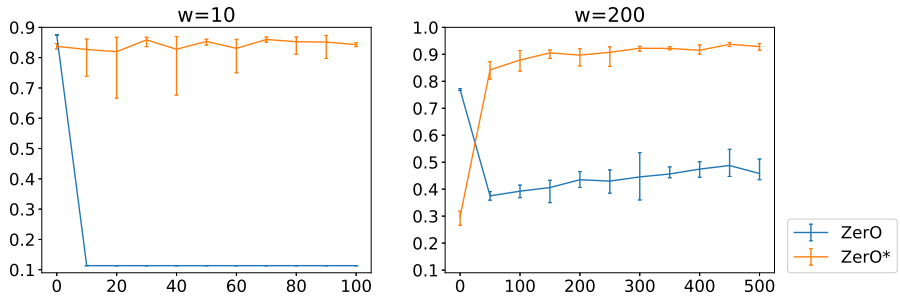

Figure 2: Test accuracy after 500 gradient steps over network depth for fully-connected constant-width ReLU networks trained on MNIST. The curves indicate the means and the bars indicate the minima and maxima over 5 repetitions.

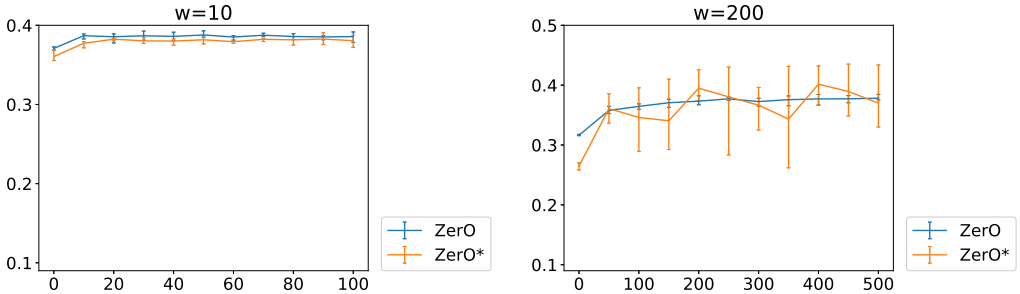

Figure 3: Test accuracy after 500 gradient steps over network depth for fully-connected constant-width ReLU networks trained on CIFAR10. The curves indicate the means and the bars indicate the minima and maxima over 5 repetitions.

Consider the simplest case of a deep feedforward network with one input dimension and one output dimension with all linear layers containing a single parameter. If we initialize the parameters according to He initialization, the variance of the output is kept constant across the layers as expected. But at the same time, the probability that the output is non-zero decreases with increasing number of layers. With the number of layers approaching infinity, the output distribution of such a network converges in probability to the Dirac delta function, which is expressed formally below.

Proposition B.1 (ReLU networks of width 1 converge to Dirac-delta distribution)**.** *Consider a neural network of depth d with constant width $w = 1$ and negative slope $a = 0$ that is He-initialized. Then for input symmetric random variable x the output $y^{(d)}$ satisfies $p(y^{(d)} \neq 0) = (\frac{1}{2})^d$ and $\lim_{d\to\infty} p(y^{(d)} \neq 0) = 0$.*

Proof. We prove the observation by induction on d. The base case for $y^{(0)}$ is satisfied by definition of x. Now if the observation holds for an l, we have $p(y^{(l)} \neq 0) = (\frac{1}{2})^l$. Take $y^{(l+1)} = w^{(l+1)}\phi_0(y^{(l)})$. $y^{(l)}$ is non-zero with probability $(\frac{1}{2})^l$, and because of symmetry, it is positive with probability $\frac{1}{2}(\frac{1}{2})^l = (\frac{1}{2})^{l+1}$, and so $\phi_0(y^{(l)})$ is non-zero with probability $(\frac{1}{2})^{l+1}$. Because $w^{(l+1)}$ is He-initialized, $y^{(l+1)} = w^{(l+1)}\phi_0(y^{(l)})$ is non-zero if and only if $\phi_0(y^{(l)})$ is non-zero. □

The above observation is a special case of a more general result which holds for neural networks with ReLU activations with arbitrary width.

Proposition B.2 (ReLU networks converge to Dirac-delta distribution)**.** *Consider a He-initialized network of layer widths $\{w_l\}_{l=1}^d$, negative slope of 0 and depth d. Then for input He random vector $\boldsymbol{x}$ the output $\boldsymbol{y}^{(d)}$ satisfies $p(\boldsymbol{y}^{(d)} \neq \boldsymbol{0}) = \prod_{i=0}^{d-1} \frac{2^{w_i}-1}{2^{w_i}}$ and so $\lim_{d\to\infty} p(\boldsymbol{y}^{(d)} \neq \boldsymbol{0}) = 0$.*

Proof. We prove the observation by induction on d. The base case for $\mathbf{y}_0$ is satisfied by definition of input as He random vector. If the observation holds for an l we have $p(\mathbf{y}^{(l)} \neq \mathbf{0}) = \prod_{i=0}^{l-1} \frac{2^{w_i}-1}{2^{w_i}}$. Note that it must hold that either all elements of $\mathbf{y}^{(l)}$ are non-zero or all are zero so $p(\mathbf{y}^{(l+1)} \neq \mathbf{0}) = p(\phi_0(\mathbf{y}^{(l)}) \neq \mathbf{0})$. $\phi_0(\mathbf{y}^{(l)}) \neq \mathbf{0}$ if two conditions are met - $\mathbf{y}^{(l)}$ is non-zero and at least one its element is positive. The probability that all elements of $\mathbf{y}^{(l)}$ are negative given that it is non-zero is $(\frac{1}{2})^{w_l}$ because they are symmetric and uncorrelated. So the probability that at least one element of $\mathbf{y}_l$ is positive given that $\mathbf{y}_l$ is non-zero is $(1 - \frac{1}{2^{w_l}})$. By law of total probability we have $p(\mathbf{y}^{(l+1)} \neq \mathbf{0}) = (1 - \frac{1}{2^{w_l}}) \prod_{i=0}^{l-1} \frac{2^{w_i}-1}{2^{w_i}} = (\frac{2^{w_l}-1}{2^{w_l}}) \prod_{i=0}^{l-1} \frac{2^{w_i}-1}{2^{w_i}} = \prod_{i=0}^{l} \frac{2^{w_i}-1}{2^{w_i}}$. □

References

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009. URL `https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf`.

Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. doi: 10.1109/5.726791.