The Vanishing Empirical Variance in Randomly Initialized Deep ReLU Networks

Michał Grzejdziak-Zdziarski[1], David M.J. Tax[2], and Marco Loog[3]

[1] Nomagic, Warsaw, Poland (✉)
[2] Delft University of Technology, Delft, The Netherlands
[3] Radboud University, Nijmegen, The Netherlands

Abstract. Neural networks are typically initialized such that the hidden pre-activations' theoretical variance remains constant to avoid the vanishing and exploding gradient problem. This condition is necessary to train very deep networks, but numerous analyses show this to be insufficient. We explain this behavior by analyzing the *empirical* variance, which is more meaningful in the practical setting that deals with 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 and find that its kurtosis grows to infinity with increasing depth, even if the theoretical variance stays constant. As a result, the empirical variance vanishes: its asymptotic distribution converges in probability to zero. Our analysis focuses on fully connected ReLU networks with He-initialization, but we hypothesize that many more random weight initialization methods suffer from 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.

Keywords: vanishing gradient · empirical variance · kurtosis · ReLU.

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 [13,3]. This approach has been used to derive two initialization methods: so-called Glorot [8] and He-initialization [11]. However, these methods are still not sufficient to train arbitrarily deep networks. Other statistical properties have been shown to explode or vanish even under constant variance [9,10,5,24], while in the meantime, still no initialization method has been demonstrated to work for very deep networks.

Here, we take another look at the consequences of keeping the theoretical variance constant and analyze distributional properties that go beyond this second central moment. Specifically, with a focus on He-initialization [11], we analyze the dynamics of the kurtosis, the fourth standardized moment, as a signal is propagated through a neural network that is He-initialized. Under mild assumptions, we prove that the 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 to zero (in probability), despite the constant theoretical variance. Consequently, almost all outputs are mapped to zero by an arbitrarily deep network. We call this problem the vanishing empirical variance.

Our analysis suggests that the problem of 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 [27], which is a deterministic method that keeps empirical variance constant, can train very deep and narrow networks, a fact not realized by its authors.

In Section 2, we recall the literature related to our work. In Section 3, we define the basic setup we consider, nuancing the original analysis of He-initialization from [11]. In Section 4, we analyze this setting and come to 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 [4]. It was popularized as a "trick" by [19]. [8] extended it to balance the need to keep the output variance and the gradient variance constant, while [11] analyzed the specific case of ReLU activation. Further extensions of this work to the specific case of highly popular ResNets [12] have been given by [26] and [1]. Other approaches to random weight initialization include orthogonal initialization [22], delta-orthogonal initialization [25], data-dependent LSUV [20] or MetaInit initialization [6], and GSM initialization [5]. Another approach is to initialize the weights deterministically. Examples are identity initialization [2] and ZerO initialization [27]. Although our focus is He-initialization [11], we hypothesize that our claims extend to other random initialization methods, which we corroborate in our experiments.

Various results indicate that controlling the variance is not sufficient to mitigate gradient problems. [9] showed that the empirical variance of gradients grows exponentially with increasing depth, while [10] and [5] showed the same for the empirical variance of the lengths of activations and pre-activations, respectively. [24] 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. [23] showed that for standard random weight initialization methods, the number of iterations required to converge grows exponentially in depth. [7] reached a similar conclusion, while [15] 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, such as orthogonal initialization [22] or GSM initialization [5].

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[4]

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

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 [11] 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 1 (He random vector). *We say that a random vector $\boldsymbol{x} \in \mathbb{R}^{w}$ is a He random vector if all variables in $\boldsymbol{x}$ have mean zero, symmetric*[5]*, uncorrelated, and homoscedastic with some variance σ_x^2.*

Proposition 1 (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.*

[4] 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.

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

A proof for Proposition 1 has been given by [11] under the stronger assumption of preservation of independence of vector entries. For simplification, [11] assumed that independence is preserved through the network, but for our analysis it is important to realize that what actually is preserved is uncorrelatedness. We provide a proof with this modification in the supplementary material.

One may ask how to make sure that the properties of He random vector are satisfied at the input. The Proposition 2 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 2 (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

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

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

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

We will assume that inputs are always He random vector and that networks are initialized according to Proposition 1. Effectively, the outputs for each hidden layer will be He random vectors too.

4 Theory

We present our main theoretical results. We derive the relation between input and output kurtosis in a neural network (Proposition 3) and prove that for bounded-width networks, it goes to infinity with increasing depth (Theorem 1). 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 prove Proposition 3 in which we will derive the exact recursive formula for the dynamics of kurtosis over consecutive layers. 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 2 that the covariance of squared outputs is equal for any two outputs.

In the proof of Proposition 3, we will use two lemmas 1 and 2 that are given first.

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

Proof.

$$\mathbb{E}[\phi_a^4(x)] = \int\limits_{-\infty}^{\infty} \phi_a^4(x)p(x)dx = \int\limits_{-\infty}^{0} a^4x^4p(x)dx + \int\limits_{0}^{\infty} x^4p(x)dx$$

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

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

Proof.

$$\mathbb{E}[\phi_a^2(x)\phi_a^2(y)] = \int\limits_{\mathbb{R}^2} \phi_a^2(x)\phi_a^2(y)p(x,y)dxdy$$

$$= \int\limits_{\mathbb{R}_+^2} x^2y^2p(x,y)dxdy + 2a^2 \int\limits_{\mathbb{R}_+\times\mathbb{R}_-} x^2y^2p(x,y)dxdy$$

$$+ a^4 \int\limits_{\mathbb{R}_-^2} x^2y^2p(x,y)dxdy = \frac{1}{4}(a^2+1)^2\mathbb{E}[x^2y^2]$$

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

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

Proposition 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 $\mathbb{E}[(y_i^{(l+1)})^4] = \mathbb{E}[(\sum_{j=1}^{w_l} W_{ij}^{(l+1)}\phi_a(y_j^{(l)}))^4]$ 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

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

Using Lemma 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 2

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

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

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

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 1, that with the dynamics derived in Proposition 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 3, we will show that any product of such matrices is of a form parameterized with four positive parameters. Next, in Lemma 4, we will show that any matrix $\mathbf{A}^{(l)}$ has a positive eigenvalue that is strictly larger than 1. The proof of Lemma 4 uses the Perron theorem, which we provide with a reference to a proof in the appendix in the supplement. We combine these two properties in Lemma 5 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 tends to infinity. We use this property in the proof of Theorem 1.

Lemma 3. *Consider the product of matrices* $\boldsymbol{B} = \prod_{l=0}^{d} \boldsymbol{A}^{(d-l)}$ *as given in Proposition 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

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

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

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

Lemma 4. *The largest eigenvalue of any matrix* $\boldsymbol{A}^{(l)}$ *from Proposition 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. $\mathbf{A}_-^{(l)}$ 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 show that $\lambda_{max} > 1$.

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

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

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

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

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 $\kappa_w \geq 3$

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

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

Lemma 5. *Consider the matrix $\boldsymbol{A} = \boldsymbol{A}^{(l)}$ from Proposition 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 3 go to infinity.*

Proof. By Lemma 4 $\lambda_{max}(\mathbf{A}) > 1$ so $\lim\limits_{m\to\infty} ||\mathbf{A}^m|| = \infty$, 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 3, $\gamma_{m+1} = \gamma_1\alpha_m + \delta_1\gamma_m > \gamma_1\alpha_m$, 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 3, $\alpha_{m+1} = \alpha_m \alpha_1 + \beta_m \gamma_1$, 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 1. *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 3. 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

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

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 3 for a specific width w, and m_w the number of occurrences of such matrices until depth d. With $d \to \infty$, there is at least one w for which $m_w \to \infty$. For such widths w, $\mathbf{A}_w^{m_w}$ behaves according to Lemma 5. The product $\prod_{w=2}^{w_{max}} \mathbf{A}_w^{m_w}$ consists of a finite number of matrices of the form from Lemma 3 and at least one matrix with all positive parameters from Lemma 3 going to infinity. In effect, the positive parameters from Lemma 3 for this product goes to infinity, which implies that $\lambda_{max}\left(\prod_{w=2}^{w_{max}} \mathbf{A}_w^{m_w}\right)$ goes to infinity. As a result, with $d \to \infty$, $\lambda_{max}(\mathbf{B}^{(d)}) \to \infty$.

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

5 Vanishing empirical variance

Theorem 1 has important consequences for He-initialized networks. That the kurtosis grows to infinity implies that, for any finite sample size, the observed empirical variance will converge in probability to zero. Combined with the other properties preserved through He-initialization, this practically means that virtually all outputs map arbitrarily close to zero for a sufficiently deep network. And this happens despite the theoretical variance being constant.

Let us explain this implication more formally.[6] For variance of the empirical variance S_n^2 and the kurtosis, it holds that $Var[S_n^2] = \left(\kappa - \frac{n-3}{n-1}\right)\frac{\sigma^4}{n}$, where n is the sample size, κ is the kurtosis and σ^2 is the theoretical variance. For large n, we can approximate distribution of the ratio $\frac{S_n^2}{\sigma^2}$ by $\frac{\chi^2(DF_n)}{DF_n}$, where $DF_n = \frac{2\sigma^4}{Var[S_n^2]} = \frac{2n}{\kappa - \frac{n-3}{n-1}}$. This can alternatively be 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 shrinks to zero for any $n \in \mathbb{N}$, 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.

When training sufficiently deep networks on machines with finite precision using finite datasets, we will observe outputs to be zeroed out. As a result, propagated gradients will be zeros and the weights of the network will remain at their initialization values.

6 Experiments

Theorem 1 shows that He-initialization suffers from the vanishing empirical variance problem. We now hypothesize that problems with empirical variance concern all fully random initialization methods that 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. Here, we present empirical evidence supporting this hypothesis. We verify it experimentally for five state-of-the-art random

[6] We refer to [21] for a detailed treatment and proofs.

initialization methods: Glorot by [8], He by [11], orthogonal by [22], GSM by [5], and MetaInit by [6]. We also show that ZerO proposed by [27] is superior to all these methods in very deep regimes.

All experiments are performed on constant-width ReLU networks on MNIST [18] and CIFAR10 [17]. 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 code is available at `https://github.com/Grzejdziok/vanishing-empirical-variance`.

6.1 The necessity of small kurtosis for He-initialization

We performed experiments 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.

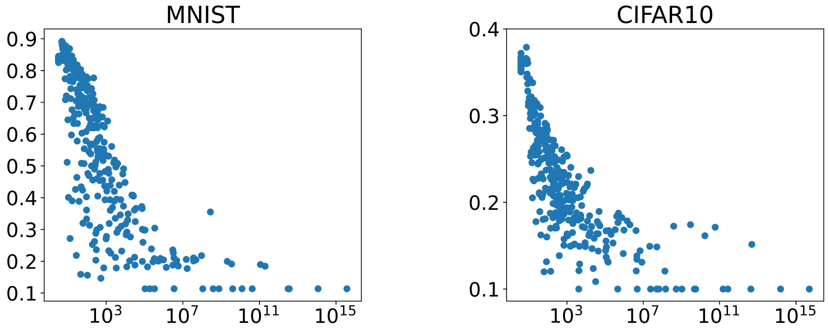

Fig. 1. Test accuracy after 500 gradient steps vs output distribution kurtosis at initialization for networks of varying widths, depths, and initialization distributions trained on MNIST and CIFAR10. Results for 330 experiments per dataset.

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 the 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 [16] 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.

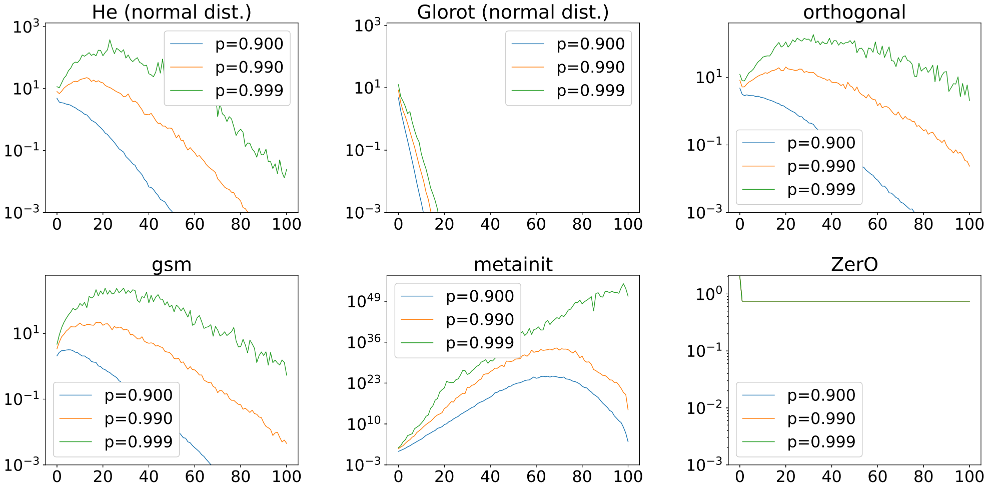

Fig. 2. Estimated quantiles of the output empirical variance distribution over depth given the whole CIFAR10. The plots use a logarithmic scale. ZerO is deterministic, so all its quantiles are equal.

6.2 SOTA initialization and empirical variance problem

To see whether the problems with empirical variance occur for other state-of-the-art initialization methods, we performed an experiment to estimate the empirical variance distribution at the output.

For all random initialization methods considered, we initialized 10,000 neural networks (1,000 for MetaInit) of constant width $w = 10$ and depth $d = 100$ and calculated the empirical variance of a single output given the whole CIFAR10 training set as input. We then calculated quantiles at 0.9, 0.99, and 0.999 over all different initializations and plotted these against layer depth in Figure 2.

We observe that for all random initialization methods except MetaInit 90% 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. We observe a different behavior for ZerO, which initializes most layers to identities. It keeps the empirical variance constant after the first layer.

6.3 Empirical variance problem: practical significance

To evaluate the practical significance of the problems with empirical variances observed in the previous section, we trained neural networks of varying depths on CIFAR10 and evaluated their test accuracy after 500 gradient steps. We used Adam [16] as an optimizer with $\beta_1 = 0.9$, and $\beta_2 = 0.999$ without weight decay. We trained networks for two widths: 1) width 10 and depths from 0 to 100 with

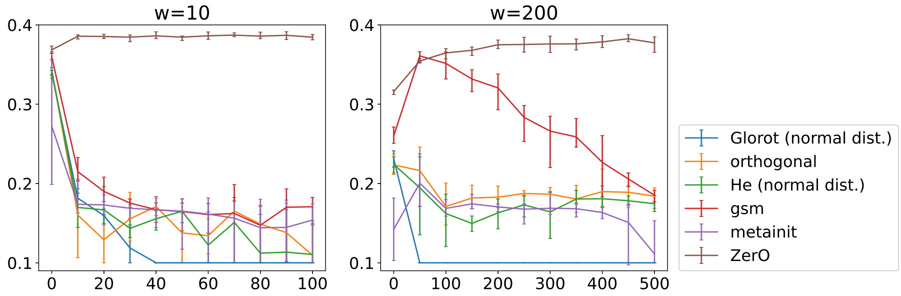

Fig. 3. Test accuracy after 500 gradient steps over network depth for fully connected constant-width ReLU networks trained on CIFAR10. The curves indicate the means and the bars indicate the minima and maxima over 5 repetitions.

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 3. 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.

7 Discussion

By analyzing the dynamics of the output's 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 issues with exploding or vanishing empirical variance concern not only He-initialization but also other state-of-the-art random initialization methods such as Glorot [8], GSM [5], or MetaInit [6]. All of these methods fail to train very deep networks, while our experiments show that with deterministic initialization methods like ZerO [27], successful training is possible.

Our contribution is primarily theoretical and, in our experiments, we analyzed a toy architecture of constant-width fully connected ReLU networks that illustrated our theoretical results. However, we hypothesize that our main result about exploding kurtosis extends to setups that are commonly used in practice, like transformers or convolutional networks. The addition of skip connections cannot stop the growth of kurtosis because kurtosis explodes even for networks without activation function, which is equivalent to setting the negative slope parameter a to 1 in our analysis. Moreover, convolutional layers will induce even more dependency of layer outputs due to parameter sharing, so we expect the output kurtosis to grow even faster.

It is unclear whether using activation functions other than ReLU could mitigate this issue. Bounded activation functions like tanh or sigmoid reduce the theoretical variance of their inputs, yet their impact on kurtosis is unclear.

References

1. Bachlechner, T., Majumder, B.P., Mao, H., Cottrell, G., McAuley, J.: Rezero is all you need: fast convergence at large depth. In: de Campos, C., Maathuis, M.H. (eds.) Proceedings of the Thirty-Seventh Conference on Uncertainty in Artificial Intelligence. vol. 161, pp. 1352–1361. PMLR (27–30 Jul 2021), https://proceedings.mlr.press/v161/bachlechner21a.html
2. Bartlett, P., Helmbold, D., Long, P.: Gradient descent with identity initialization efficiently learns positive definite linear transformations by deep residual networks. In: Dy, J., Krause, A. (eds.) Proceedings of the 35th International Conference on Machine Learning. vol. 80, pp. 521–530 (10–15 Jul 2018), https://proceedings.mlr.press/v80/bartlett18a.html
3. Bengio, Y., Simard, P., Frasconi, P.: Learning long-term dependencies with gradient descent is difficult. IEEE Transactions on Neural Networks **5**(2), 157–166 (1994). https://doi.org/10.1109/72.279181
4. Bishop, C.M.: Neural Networks for Pattern Recognition. OUP, USA (1995)
5. Burkholz, R., Dubatovka, A.: Initialization of relus for dynamical isometry. In: Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., Garnett, R. (eds.) Advances in Neural Information Processing Systems. vol. 32. Curran Associates, Inc. (2019), https://proceedings.neurips.cc/paper/2019/file/d9731321ef4e063ebbee79298fa36f56-Paper.pdf
6. Dauphin, Y.N., Schoenholz, S.: Metainit: Initializing learning by learning to initialize. In: Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., Garnett, R. (eds.) Advances in Neural Information Processing Systems. vol. 32. Curran Associates, Inc. (2019), https://proceedings.neurips.cc/paper/2019/file/876e8108f87eb61877c6263228b67256-Paper.pdf
7. Du, S., Hu, W.: Width provably matters in optimization for deep linear neural networks. In: Chaudhuri, K., Salakhutdinov, R. (eds.) Proceedings of the 36th International Conference on Machine Learning. vol. 97, pp. 1655–1664. PMLR (09–15 Jun 2019), https://proceedings.mlr.press/v97/du19a.html
8. Glorot, X., Bengio, Y.: Understanding the difficulty of training deep feedforward neural networks. In: Teh, Y.W., Titterington, M. (eds.) Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics. vol. 9, pp. 249–256. PMLR, Chia Laguna Resort, Sardinia, Italy (13–15 May 2010), https://proceedings.mlr.press/v9/glorot10a.html
9. Hanin, B.: Which neural net architectures give rise to exploding and vanishing gradients? In: Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., Garnett, R. (eds.) Advances in Neural Information Processing Systems. vol. 31. Curran Associates, Inc. (2018), https://proceedings.neurips.cc/paper/2018/file/13f9896df61279c928f19721878fac41-Paper.pdf
10. Hanin, B., Rolnick, D.: How to start training: The effect of initialization and architecture. In: Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., Garnett, R. (eds.) Advances in Neural Information Processing Systems. vol. 31. Curran Associates, Inc. (2018), https://proceedings.neurips.cc/paper/2018/file/d81f9c1be2e08964bf9f24b15f0e4900-Paper.pdf
11. He, K., Zhang, X., Ren, S., Sun, J.: Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In: 2015 IEEE ICCV. pp. 1026–1034 (2015). https://doi.org/10.1109/ICCV.2015.123
12. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition. pp. 770–778 (2016). https://doi.org/10.1109/CVPR.2016.90

13. Hochreiter, S.: Untersuchungen zu dynamischen neuronalen Netzen. Master's thesis, Institut fur Informatik, Technische Universitat, Munchen **1**, 1–150 (1991)
14. Horn, R.A., Johnson, C.R.: Matrix Analysis. Cambridge University Press, Cambridge, 2nd edn. (2013)
15. Hu, W., Xiao, L., Pennington, J.: Provable benefit of orthogonal initialization in optimizing deep linear networks. In: International Conference on Learning Representations (2020), `https://openreview.net/forum?id=rkgqN1SYvr`
16. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: International Conference on Learning Representations (2015)
17. Krizhevsky, A.: Learning multiple layers of features from tiny images. Tech. rep., University of Toronto (2009), `https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf`
18. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE **86**(11), 2278–2324 (1998). `https://doi.org/10.1109/5.726791`
19. LeCun, Y., Bottou, L., Orr, G.B., Müller, K.R.: Efficient backprop. In: Orr, G.B., Müller, K.R. (eds.) Neural Networks: Tricks of the Trade, pp. 9–50. Springer Berlin Heidelberg, Berlin, Heidelberg (1998). `https://doi.org/10.1007/3-540-49430-8_2`, `https://doi.org/10.1007/3-540-49430-8_2`
20. Mishkin, D., Matas, J.: All you need is a good init. In: International Conference on Learning Representations (2016)
21. O'Neill, B.: Some useful moment results in sampling problems. The American Statistician **68**(4), 282–296 (2014). `https://doi.org/10.1080/00031305.2014.966589`, `https://doi.org/10.1080/00031305.2014.966589`
22. Saxe, A., McClelland, J., Ganguli, S.: Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. In: International Conference on Learning Representations (2014)
23. Shamir, O.: Exponential convergence time of gradient descent for one-dimensional deep linear neural networks. In: Beygelzimer, A., Hsu, D. (eds.) Proceedings of the Thirty-Second Conference on Learning Theory. vol. 99, pp. 2691–2713. PMLR (25–28 Jun 2019), `https://proceedings.mlr.press/v99/shamir19a.html`
24. Vladimirova, M., Verbeek, J., Mesejo, P., Arbel, J.: Understanding priors in Bayesian neural networks at the unit level. In: Chaudhuri, K., Salakhutdinov, R. (eds.) Proceedings of the 36th International Conference on Machine Learning. vol. 97, pp. 6458–6467. PMLR (09–15 Jun 2019), `https://proceedings.mlr.press/v97/vladimirova19a.html`
25. Xiao, L., Bahri, Y., Sohl-Dickstein, J., Schoenholz, S., Pennington, J.: Dynamical isometry and a mean field theory of CNNs: How to train 10,000-layer vanilla convolutional neural networks. In: Dy, J., Krause, A. (eds.) Proceedings of the 35th International Conference on Machine Learning. vol. 80, pp. 5393–5402. PMLR (10–15 Jul 2018), `https://proceedings.mlr.press/v80/xiao18a.html`
26. Zhang, H., Dauphin, Y.N., Ma, T.: Residual learning without normalization via better initialization. In: International Conference on Learning Representations (2019), `https://openreview.net/forum?id=H1gsz30cKX`
27. Zhao, J., Schaefer, F.T., Anandkumar, A.: Zero initialization: Initializing neural networks with only zeros and ones. Transactions on Machine Learning Research (2022), `https://openreview.net/forum?id=1AxQpKmiTc`

A Proof of Proposition 3.2

We first prove the following lemma.

Lemma 6. *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.

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

Below, we prove Proposition 3.2.

Proposition 4 (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

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

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

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

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

B Perron theorem

We provide the Perron theorem as given in [14]. We refer to this book for further details and proofs.

Theorem 2 (Perron). *Let $\boldsymbol{A}$ be a $n \times n$ matrix which is irreducible and non-negative 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* A,
3. *there is a unique real vector* $\boldsymbol{x}$ *such that* $\boldsymbol{A}\boldsymbol{x} = \rho(\boldsymbol{A})\boldsymbol{x}$ *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. $(\rho(\boldsymbol{A})^{-1}\boldsymbol{A})^m \rightarrow \boldsymbol{x}\boldsymbol{y}^T$ *as* $m \rightarrow \infty$.