Pattern Recognition Letters
journal homepage: www.elsevier.com

If dropout limits trainable depth, does critical initialisation still matter? A large-scale statistical analysis on ReLU networks

Arnu Pretorius[a], Elan van Biljon[a], Benjamin van Niekerk[b], Ryan Eloff[b], Matthew Reynard[b], Steve James[c], Benjamin Rosman[c], Herman Kamper[b,**], Steve Kroon[a]

[a] *Computer Science Division, Stellenbosch University, South Africa*
[b] *Department of Electrical and Electronic Engineering, Stellenbosch University, South Africa*
[c] *Department of Computer Science and Applied Mathematics, University of the Witwatersrand, South Africa*

ABSTRACT

Recent work in signal propagation theory has shown that dropout limits the depth to which information can propagate through a neural network. In this paper, we investigate the effect of initialisation on training speed and generalisation for ReLU networks within this depth limit. We ask the following research question: given that critical initialisation is crucial for training at large depth, if dropout limits the depth at which networks are trainable, does initialising critically still matter? We conduct a large-scale controlled experiment, and perform a statistical analysis of over 12 000 trained networks. We find that (1) trainable networks show no statistically significant difference in performance over a wide range of non-critical initialisations; (2) for initialisations that show a statistically significant difference, the net effect on performance is small; (3) only extreme initialisations (very small or very large) perform worse than criticality. These findings also apply to standard ReLU networks of moderate depth as a special case of zero dropout. Our results therefore suggest that, in the shallow-to-moderate depth setting, critical initialisation provides zero performance gains when compared to off-critical initialisations and that searching for off-critical initialisations that might improve training speed or generalisation, is likely to be a fruitless endeavour.

1. Introduction

Dropout is arguably one of the most popular and successful forms of regularisation for deep neural networks (Srivastava et al., 2014). This has sparked research into analysing dropout's effects (Wang and Manning, 2013; Wager et al., 2013; Baldi and Sadowski, 2013), extending dropout's mechanism of regularisation (Wan et al., 2013; Gal et al., 2017; Gomez et al., 2018; Ghiasi et al., 2018) and connecting dropout to different Bayesian inference methods (Kingma et al., 2015; Gal and Ghahramani, 2016; Molchanov et al., 2017). Despite its success, dropout has also been shown to limit the *trainable depth* of a neural network (Schoenholz et al., 2017).

At initialisation, the random weight projection at each layer combined with dropout may cause inputs to become uniformly correlated beyond a certain depth. Discriminatory information in the inputs may therefore vanish before reaching the output. The trainable depth of a network is the maximum depth to which this information is able to propagate forward without completely vanishing in this way. Schoenholz et al. (2017) arrive at this result through a *mean field* analysis of dropout at initialisation.

Mean field theory provides a powerful approach to analysing neural networks and has become a cornerstone of recent discoveries in improved initialisation schemes. These schemes, often referred to as *critical initialisations*, ensure stable signal propagation dynamics by preserving second moment input statistics during the forward pass, even at infinite depth. Critical initialisation has made it possible to train *extremely deep* networks (sometimes up to 10 000 layers) for a variety of different architectures (Pennington et al., 2017; Xiao et al., 2018; Chen et al., 2018). Using the tools of mean field theory, Pretorius et al. (2018) extend these results to fully-connected ReLU networks with multiplicative noise regularisation. These results hold for a general class of noise distributions, while earlier

**Corresponding author: *e-mail:* kamperh@sun.ac.za (Herman Kamper)

work (Hendrycks and Gimpel, 2016) describes dropout-specific initialisation schemes.

For non-critical initialisation, signal propagation can become unstable and result in the saturation of activation functions. In the particular case of ReLU activations, numerical instability (overflow or underflow) can arise when training very deep networks. Despite this, when training ReLU networks of a finite depth, there is a range of trainable but non-critical initialisations. That is, there exists a "band" of valid initialisations around the critical point. It is conceivable that using these alternative, non-critical initialisations may confer some benefits. For example, Saxe et al. (2014) note that just off of criticality, the spectrum of the input-output Jacobian can be well behaved, which has been linked to improvements in training and generalisation (Pennington et al., 2017). This leads us to the following question.

Question: *If dropout limits the depth to which networks can train, does critical initialisation still matter?* Given that stable signal propagation at extreme depths is no longer a concern, are there alternative initialisations that might perform better than the critical initialisation?

To investigate the above research question, we conduct a large-scale randomised control trial (RCT)—an approach borrowed from the medical community—to compare training speed and generalisation for ReLU neural networks with dropout for different initialisations. We consider multiple datasets, training algorithms, dropout rates and combinations of hyperparameters to avoid confounding effects. To the best of our knowledge, this is the first application of RCTs in a deep learning context. A statistical analysis of our results leads to the following insight.

Answer: *There is no statistically significant difference between the critical initialisation and a wide neighbourhood of non-critical initialisations, as measured by training speed and generalisation.* In our experiment we find that this also applies to standard ReLU networks without dropout, for which the critical initialisation is the popular "He" initialisation (He et al., 2015). Our findings seem to indicate that networks of moderate depth (less than 20 layers) are in fact very insensitive to initialisation. In addition, we conclude that exploring the initialisation landscape around criticality in the hope of finding previously undiscovered benefits, is unlikely to be a fruitful enterprise.

2. Background

We model the expected value of a target variable $\mathbf{y}$ conditioned on an input $\mathbf{x}$, i.e. $\mathbb{E}(\mathbf{y}|\mathbf{x})$, using a fully-connected feedforward neural network with dropout. Given an input $\mathbf{x}^0 \in \mathbb{R}^{D_0}$, we can define this neural network recursively as

$$\mathbf{x}^l = \phi(\tilde{\mathbf{h}}^l), \qquad \tilde{\mathbf{h}}^l = W^l\left(\mathbf{x}^{l-1} \odot \frac{\epsilon^{l-1}}{1-\theta}\right) + \mathbf{b}^l, \tag{1}$$

for $l = 1, ..., L$, where L is the total number of hidden layers, $\odot$ denotes element-wise multiplication, and $\epsilon^l \sim \text{Bern}(1-\theta)$ is a Bernoulli noise vector, corresponding to a dropout rate θ. The dimensionality of hidden layer l is denoted as D_l, and activations at each layer are computed element-wise using $\phi(a) = \text{ReLU}(a) = \max(0, a)$. The initial weights $W^l \in \mathbb{R}^{D_l \times D_{l-1}}$ and biases $\mathbf{b}^l \in \mathbb{R}^{D_l}$ are sampled i.i.d. from zero-mean Gaussian distributions with variances σ_w^2/D_{l-1} and σ_b^2, respectively.

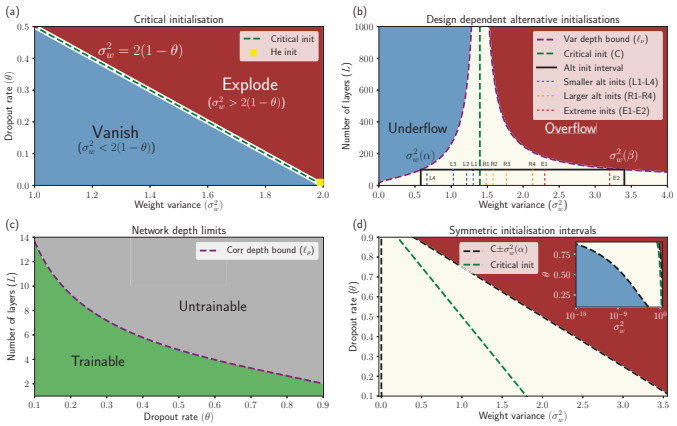


Fig. 1. ***Choosing network depth and initialisation*****. (a): Critical initialisation boundary separating regimes of vanishing or exploding variance signal propagation at large depth. (b): An illustration of the region that allows stable variance (beige) and correlation (black bordered beige) information to propagate through the entire network. (We choose a depth $L = 100$ for ease of visualisation, although this depth is too deep for training with dropout). Alternative non-critical initialisations L1–L4 (blue), C (green), R1–R4 (orange), E1–E2 (red) are sampled from this region. (c): The depth to which networks with dropout are trainable for different dropout rates (dashed purple line). (d) Symmetrical interval containing L1–L4 and R1–R4 as a function of the dropout rate: the (beige) region around criticality represents the set of trainable initialisations.**

We focus on ReLU because of its widespread use and empirical success and consider the fully-connected setting since conclusions for these networks often generalise to other architectures, e.g. convolutional networks (He et al., 2015; Xiao et al., 2018). Schoenholz et al. (2017) also hypothesise that the signal propagation behaviour of many different architectures is likely to be governed by the fully-connected case.

2.1. Mean field theory for signal propagation

Poole et al. (2016), Schoenholz et al. (2017) and Pretorius et al. (2018) use mean field theory to analyse fully-connected feedforward neural networks at initialisation. For large layer widths, each pre-activation (the linear combination of the incoming connections from the previous layer) at initialisation in any given layer of the network represents a large sum of i.i.d. random variables. According to the central limit theorem, this sum will tend to a Gaussian distribution in the limit of infinite width. Using the above observation, the mean field approach is to fit Gaussian distributions over all the pre-activation units through moment matching to describe the behaviour of wide random neural networks at initialisation.

In more detail, consider two inputs $\mathbf{x}_1^0$ and $\mathbf{x}_2^0$. Denote the scalar pre-activation at unit j in layer l for input $\mathbf{x}_1^0$ as $\tilde{h}_j^{l,1}$. For fully-connected ReLU networks with dropout, Pretorius et al. (2018) derive the joint distribution over the pre-activations in expectation over the network parameters and the noise as

$$p\left(\tilde{h}_j^{l,1}, \tilde{h}_j^{l,2}\right) = \mathcal{N}(\mathbf{0}, \tilde{\Phi}^l),$$

where

$$\tilde{\Phi}^l = \begin{bmatrix} \nu_1^l & \kappa^l \\ \kappa^l & \nu_2^l \end{bmatrix}.$$

The layer-wise evolution of the terms in the covariance matrix $\tilde{\Phi}^l$, are given by

$$\nu_1^l = \frac{\sigma_w^2}{2(1-\theta)} \nu_1^{l-1} + \sigma_b^2 \tag{2}$$

$$\kappa^l = \frac{\sigma_w^2}{2} \kappa^{l-1} \left(g(\rho^{l-1}) + \frac{1}{2} \right) + \sigma_b^2 \tag{3}$$

$$\rho^l = \kappa^l / \sqrt{\nu_1^l \nu_2^l} \tag{4}$$

where

$$g(\rho^{l-1}) = \frac{1}{\pi \rho^{l-1}} \left(\rho^{l-1} \sin^{-1}\left(\rho^{l-1}\right) + \sqrt{1 - (\rho^{l-1})^2} \right)$$

with initial variance $\nu_1^0 = \frac{\mathbf{x}_1^0 \cdot \mathbf{x}_1^0}{D^0}$ and covariance $\kappa^0 = \frac{\mathbf{x}_1^0 \cdot \mathbf{x}_2^0}{D^0}$. The above quantities are derived in the large width limit, but in practice tend to hold for finite widths of moderate size (Poole et al., 2016; Schoenholz et al., 2017; Pretorius et al., 2018).

Critical initialisation. A fixed point of the variance recurrence in (2) is given by

$$\{\sigma_w^2, \sigma_b^2\} = \{2(1-\theta), 0\},$$

which ensures that signal propagation variances are preserved during the forward pass of a ReLU network with dropout (Pretorius et al., 2018). This network parameter setting is referred to as the *critical initialisation*. Figure 1(a) shows the relationship between the critical initialisation and the dropout rate. Away from criticality, the variance signal tends to vanish or explode. If the dropout rate θ is zero, the initialisation reduces to the popular "He" initialisation for ReLU networks (He et al., 2015).

Trainable depth. Consider the following proposition due to Schoenholz et al. (2017):

> **Proposition 1**: *At initialisation, a necessary condition for training any neural network is that the information from the input layer should be able to reach the output layer.*

Pretorius et al. (2018) analyse the evolution of the input variances and correlations, as given in (2) and (4), to establish when this information propagation requirement is violated. Let α and β represent the smallest and largest positive values representable on a modern machine. The depth at which numerical instability issues (underflow or overflow) arise from the variances described in (2) for non-critical initialisations is bounded by

$$\ell_\nu = \begin{cases} \frac{\ln\left(\frac{\alpha}{\nu_0^1}\right)}{\ln\left(\frac{\sigma_w^2}{2(1-\theta)}\right)}, & \text{if } \sigma_w^2 < 2(1-\theta) \\ \frac{\ln\left(\frac{\beta}{\nu_0^1}\right)}{\ln\left(\frac{\sigma_w^2}{2(1-\theta)}\right)}, & \text{if } \sigma_w^2 > 2(1-\theta). \end{cases} \tag{5}$$

An example of these bounds are shown in Figure 1(b) as purple dashed lines. The depth bounds around criticality are finite but large, exceeding typical depths for most modern deep neural networks used in practice. However, even if single input information can propagate to large depths, the correlation between inputs, described in (4), converge to degenerate levels over a much shorter depth horizon (Pretorius et al., 2018). This can limit the network's ability to train, since all discriminatory information is lost during forward propagation. Furthermore, the rate of convergence in correlation is invariant to initialisation, but increases as more dropout is applied. As a result, inputs to a dropout network tend to convey similar information at shallower depths compared to unregularised networks. The bound that characterises convergence in correlation is

$$\ell_\rho = -6 / \ln \left[\frac{(1-\theta)}{\pi} \left(\sin^{-1}(\rho^*) + \frac{\pi}{2} \right) \right], \tag{6}$$

where ρ^* denotes the converged correlation and the factor 6 is an ad-hoc factor, which seems to provide a good fit to experimental data, but is as yet unexplained (Schoenholz et al., 2017; Pretorius et al., 2018). Figure 1(c) plots the theoretically predicted trainable depth using (6) for different dropout rates. Note that these depths are much shallower than those for variance dynamics.

3. Experimental setup

We conduct a large-scale controlled experiment using networks of trainable depth to compare the effect of initialisation on training speed and generalisation for ReLU networks with dropout. We explore the space around criticality by selecting alternative initialisations whose values theoretically satisfy Proposition 1. Our final aim is to test whether there exists a statistically significant difference, as measured by training speed and generalisation, between the different initialisations. To answer this question, we use a systematic randomised control trial methodology with hypothesis testing.

3.1. Controlled experiments using neural networks: a randomised control trial approach

Inspired by causal discovery in medical research, we consider a hypothesis a priori and conduct a "randomised control trial" (RCT) (Kendall, 2003) using neural networks. In an ordinary randomised control trial a random sample, representative of the full population, is split into two groups. One group receives some form of an intervention, such as a new drug. The other group, referred to as the control group, receives no intervention. The purpose of the two groups is to control for all confounding effects that are unrelated to the intervention of interest. The groups are then monitored by collecting data over time. Once the study has been completed, a test for statistical significance can be applied to ascertain if there exists a difference between the two groups, as measured by a quantitative metric of interest. If a statistical significant difference is detected, the intervention is confirmed as being the cause. In this paper, we aim to test for differences in initialisation of fully-connected ReLU neural networks with dropout.

To begin, consider the following *design space*:

Ω-design space: *We define the neural network design space Ω as the space consisting of different possible combinations of design components used to construct an algorithm for classification using a fully-connected ReLU neural network with dropout. Specifically, the design space is given by the following Cartesian product*

$$\Omega = \mathcal{X} \times \mathcal{D} \times \mathcal{W} \times \mathcal{R} \times \mathcal{B} \times \mathcal{O} \times \mathcal{M} \times \mathcal{L}$$

where the component sets divide into (1) dataset $\mathcal{X}$, (2) network topology: depth $\mathcal{D}$, width $\mathcal{W}$, (3) dropout rate $\mathcal{R}$, and (4) training procedure: batch size $\mathcal{B}$, optimiser $\mathcal{O}$, momentum $\mathcal{M}$, and learning rate $\mathcal{L}$.

We adapt the RCT approach for analysing neural network initialisation as follows. First, we randomly generate a collection of different neural network algorithms by sampling from the design space, or "population," of possible neural networks. For example, a 10-layer ReLU network trained on MNIST, where each layer is 256 units wide, with a dropout rate of 0.5, optimised using RMSprop with zero momentum and a learning rate of 5×10^{-4} and batches of size 128, corresponds to the 8-tuple: $(\text{MNIST}, 10, 256, 0.5, 128, \text{RMSprop}, 0, 5 \times 10^{-4})$. Next, we construct identical "groups" by using multiple copies of the sampled designs. Each group in the experiment is then assigned a different initialisation scheme. Finally, we test the following hypothesis related to a given metric:

Null hypothesis: *Given a metric τ, let $\mu_{crit}(\tau)$ denote the group mean associated with the critical initialisation and $\mu_a(\tau)$, the mean associated with an alternative initialisation $a \in \mathcal{A} \subset \mathcal{I}$, where $\mathcal{I}$ is the set of all possible initialisations, and $\mathcal{A}$ is our chosen set of alternative initialisations. Then the null hypothesis to be tested is*

$$\text{H}_0 : \mu_{\text{crit}}(\tau) = \mu_a(\tau), \forall a \in \mathcal{A}. \tag{7}$$

We discuss our methodology for selecting alternative initialisations in Section 3.2.

If the null hypothesis is rejected, we have strong evidence to indicate that the performance of the critical initialisation is significantly different from those of alternative initialisations. If H_0 cannot be rejected, the perceived difference is not considered statistically significant. Figure 2 summarises this approach to studying the effect of initialisation in neural networks.

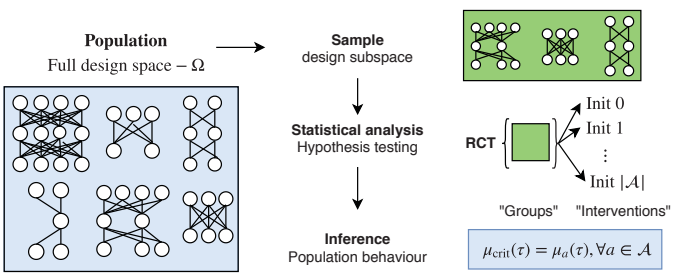


Fig. 2. Randomised control trial approach to analysing the effect of initialisation in neural networks.

Metrics. Our metrics of interest are training speed and generalisation performance. Specifically, we define **training speed** (τ_s) as the accuracy achieved on the training set at the 100$^{\text{th}}$ epoch, and **generalisation** (τ_g) as the highest accuracy achieved on the test set over the course of training. For example, $\mu_{\text{crit}}(\tau_s)$ denotes the mean training speed associated with the critical initialisation, where a higher mean accuracy at epoch 100 indicates faster training.

Sampling algorithm designs. For our experiment groups, we sample 1120 different designs. These designs are drawn randomly from Ω, which we construct by forming the Cartesian product of the following discrete sets for dataset, depth, width, dropout rate, batch size, optimiser, momentum and learning rate:

$$\begin{aligned}
\mathcal{X} &= \{\text{MNIST}, \text{FashionMNIST}, \text{CIFAR-10}, \text{CIFAR-100}\} \\
\mathcal{D} &= \{2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 15, 20\} \\
\mathcal{W} &= \{400, 600, 800\} \\
\mathcal{R} &= \{0, 0.1, 0.3, 0.5\} \\
\mathcal{B} &= \{32, 64, 128, 256\} \\
\mathcal{O} &= \{\text{SGD}, \text{Adam}, \text{RMSprop}\} \\
\mathcal{M} &= \{0, 0.5, 0.9\} \\
\mathcal{L} &= \{10^{-3}, 10^{-4}, 10^{-5}, 10^{-6}\}
\end{aligned}$$

For a given dropout rate, we limit the sampled depths in $\mathcal{D}$ to only the settings that would allow useful correlation information to reach the output layer, i.e. $d \leq \ell_\rho, \forall d \in \mathcal{D}$. We also include network depths of 15 and 20 when no dropout is being applied. We sample 70 designs for each dropout rate and dataset combination for a large enough diversity in network architecture and optimisation. To ensure a balanced set of network designs, we simply duplicate each group of designs for every dropout rate in $\mathcal{R}$, as well as for each dataset. A full description of this process is presented in Appendix A. Finally, each network is trained for 500 epochs on MNIST (LeCun et al., 1998), FashionMNIST (Xiao et al., 2017), CIFAR-10 and CIFAR-100 (Krizhevsky and Hinton, 2009), using the full training set for each.

3.2. A network design dependent set of alternative initialisations

We ensure that our networks preserve short range correlation information by limiting their depth. To develop a principled approach towards exploring the initialisation space around criticality, we now ask: for a fixed depth, what is the range of initialisations around criticality that will remain numerically stable until the output layer? In other words, for which σ_w^2 in (5) is $\ell_\nu \geq \ell_\rho$. We can find these bounds for alternative initialisations by solving for σ_w^2 in (5), which gives

$$\begin{aligned}
\sigma_w^2(\alpha) &= \inf\left\{\sigma_w^2 \in \mathbb{R}_{>0} \middle| \ell_\nu \geq \ell_\rho, \sigma_w^2 < 2(1-\theta)\right\} \\
&= 2(1-\theta)\left(\frac{\alpha}{\nu^0}\right)^{1/\ell_\rho} \quad \text{[lower bound]} \qquad (8) \\
\sigma_w^2(\beta) &= \sup\left\{\sigma_w^2 \in \mathbb{R}_{>0} \middle| \ell_\nu \geq \ell_\rho, \sigma_w^2 > 2(1-\theta)\right\} \\
&= 2(1-\theta)\left(\frac{\beta}{\nu^0}\right)^{1/\ell_\rho} \quad \text{[upper bound]} \qquad (9)
\end{aligned}$$

In our experiments, we use 32-bit floating point precision, such that $\alpha = 1.1754944 \times 10^{-38}$ in (8) and $\beta = 3.4028235 \times 10^{38}$ in (9). An example interval for possible alternative initialisations bounded by $\sigma_w^2(\alpha)$ and $\sigma_w^2(\beta)$ is shown in Figure 1(b). Note that the interval is not symmetric. This is because although the signal vanishes and explodes at the same rate, the critical initialisation is typically much closer to α than to β. This causes the interval to be wider to the right. To cover this entire space around criticality would be computationally infeasible. Therefore, we focus on sampling alternative initialisations around criticality as a function of the dropout rate.

Specifically, we first sample a core set of initialisations within the interval $\mathrm{C} \pm \sigma_w^2(\alpha)$ centred around the critical initialisation (C), with logarithmic spacing between samples (see Appendices B and C for more detail). This symmetric interval is illustrated by the dashed black lines in Figure 1(d) for different dropout rates. Note that the interval becomes narrower for larger dropout rates. The inset in Figure 1(d) plots the left side of the interval close to zero on a log-scale. The core set of alternative initialisations for the fixed depth in Figure 1(b) are shown as blue dashed lines (below criticality, marked L1-L4) and orange dashed lines (above criticality, marked R1-R4). Finally, we explore further to the right by sampling halfway, as well as close to the end of the interval, between criticality and $\sigma_w^2(\beta)$. These more extreme initialisations are depicted in red (marked E1 and E2).

3.3. Statistical comparison methodology

The null hypothesis H_0 can be tested using an *omnibus test*, which is specifically designed for multiple comparisons (Demšar, 2006). If the null hypothesis is rejected in this setting, there is evidence that at least one of the competing initialisations is significantly different from the rest. We use the Iman-Davenport extension (Iman and Davenport, 1980) of the non-parametric Friedman rank test (Friedman, 1937) as recommended by Demšar (2006) and García et al. (2010). We describe this test below in the context of comparing different initialisations.

Friedman (Friedman, 1937): For a given metric τ and a set of competing initialisations $\mathcal{I}$, the Friedman test first ranks initialisations $i \in \mathcal{I}$ in terms of their mean performances $\mu_i(\tau)$ and then computes a test statistic using these ranks. An average rank is assigned to tied initialisations. In more detail, let r_{di} denote the rank for a specific design d (sampled from Ω) using initialisation i. We denote the mean rank over the set of all sampled designs $\Delta \subset \Omega$, as $\bar{R}_i = \frac{1}{|\Delta|} \sum_{d=1}^{|\Delta|} r_{di}$. The Friedman test statistic under the null hypothesis of no difference is

$$\chi_F^2 = \frac{12|\Delta|}{|\mathcal{I}|(|\mathcal{I}|+1)} \left(\sum_{i=1}^{|\mathcal{I}|} \bar{R}_i^2 - \frac{|\mathcal{I}|(|\mathcal{I}|+1)^2}{4} \right),$$

and is approximately χ^2 distributed with $|\mathcal{I}| - 1$ degrees of freedom.

Iman-Davenport (Iman and Davenport, 1980): It has been shown that the Friedman test can be a poor approximation to the χ^2 distribution. Therefore, the Iman-Davenport test modifies the Friedman test as follows

$$F_{ID} = \frac{(|\Delta| - 1)\chi_F^2}{|\Delta|(|\mathcal{I}| - 1) - \chi_F^2},$$

to more accurately approximate an F distribution with $(|\mathcal{I}| - 1)$ and $(|\mathcal{I}| - 1)(|\Delta| - 1)$ degrees of freedom.

If we reject H_0, we may next ask whether there exists specific differences between the critical and the alternative initialisations. For this purpose we perform multiple pairwise tests.

It is important to note, however, that when conducting multiple pairwise comparisons with popular two-sample tests, a significant difference might be detected simply by chance. To illustrate this, consider the probability of rejecting the null hypothesis when it is in fact true. This is known as a type I error. The null hypothesis is usually rejected if the probability of a type I error—the p-value—is less than some specified *significance level*, typically set at 5%. However, it is insufficient to separately control for type I errors for each individual pairwise comparison. In our case, pairwise comparisons between the critical and the alternative initialisations (L1–L4, R1–R4, E1, E2) result in a total of 10 comparisons. At a significance level of 5%, a satisfactory probability of not making a type I error in a *single* comparison is $\gamma = 1 - p(\text{reject } \mathrm{H}_0 | \mathrm{H}_0 \text{ is true}) = 95\%$. However, the probability of not making a type I error *across all comparisons* is actually $\gamma^{10} \approx 60\%$, which is much lower than what was previously considered acceptable. Therefore, we guard against type I errors in multiple tests by using *post-hoc* tests that aim to adjust the significance level to control the *family-wise* error—the probability that at least one type I error is made among multiple tests (García et al., 2010; Santafe et al., 2015). The specific post-hoc test we use is the Finner test (Finner, 1993) as recommended by Garcia and Herrera (2008) and García et al. (2010). The specifics of this test are given below.

Finner (Finner, 1993): Let p_i, $i = 1, ..., i^*, ..., |\mathcal{I}|$, denote ordered p-values obtained from multiple pairwise comparisons corresponding to the null hypotheses of no mean difference $\mathrm{H}_{01}, ..., \mathrm{H}_{0i^*}, ..., \mathrm{H}_{0|\mathcal{I}|}$. Using the Finner test, we reject $\mathrm{H}_{01}, ..., \mathrm{H}_{0i^*}$, where

$$i^* = \min \left\{ i | p_i > 1 - \gamma^{i/(|\mathcal{I}|-1)} \right\}.$$

3.4. Summary of experimental setup

We aim to test for differences in initialisation by conducting a large scale randomised control trial experiment using neural networks. We begin by sampling 70 neural network algorithm designs from the design space Ω for each dropout rate and dataset combination, for a total of 1120 designs. To form groups in our experiment, we make 11 identical copies of the 1120 designs. Note that this is a core aspect of our approach. We ensure *within-group variation* by sampling *different designs*, but then duplicate this collection of designs to form identical groups, one for each "intervention", i.e. initialisation. For each group, we assign a different initialisation—either critical initialisation (C), or one of the 10 alternative initialisations (L1–L4, R1–R4, E1–E2). All designs are then trained, resulting in a total of $70 \times 4 \times 4 \times 11 = 12320$ trained neural networks. Using these results, we test our hypothesis—that no difference exists between the various initialisations in terms of training speed and generalisation—using omnibus and post-hoc statistical tests.

To provide an analogy in the context of drug testing: our approach is akin to selecting a large random sample of human

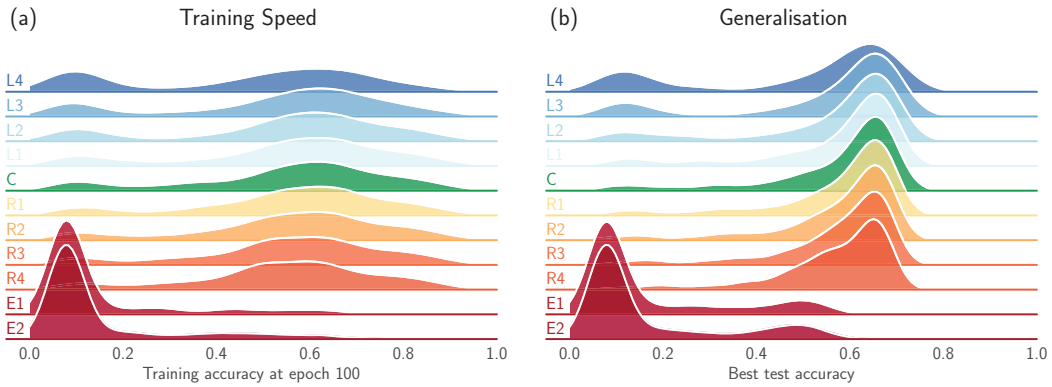


Fig. 3. ***Density fits for the different initialisations*****. (a): training speed. (b): generalisation.**

Table 1. Post-hoc tests for training speed and generalisation. The symbol * indicates a significant p-value (less than 5%) and † indicates a large effect size.

TRAINING SPEED – $H_0 : \mu_{CRIT}(\tau_s) = \mu_a(\tau_s), a \in \{L_i, R_i, E_1, E_2\}_{i=1}^4$

	L4	L3	L2	L1	R1	R2	R3	R4	E1	E2
ADJUSTED p-VALUE	*0	0.1338	0.857	0.4003	0.6479	0.0677	$^*\mathbf{3.95 \times 10^{-6}}$	$^*\mathbf{3.33 \times 10^{-15}}$	$^*\mathbf{2.4 \times 10^{-9}}$	$^*\mathbf{2.4 \times 10^{-9}}$
EFFECT SIZE	−0.2512	−0.0648	−0.0287	−0.0069	0.0094	0.0202	0.0147	−0.0048	†**−1.1055**	†**−1.1019**

GENERALISATION – $H_0 : \mu_{CRIT}(\tau_g) = \mu_a(\tau_g), a \in \{L_i, R_i, E_1, E_2\}_{i=1}^4$

	L4	L3	L2	L1	R1	R2	R3	R4	E1	E2
ADJUSTED p-VALUE	*0	0.0545	*0.0033	*0.0418	0.1226	$^*\mathbf{9.23 \times 10^{-6}}$	$^*\mathbf{4.44 \times 10^{-16}}$	*0	*0	*0
EFFECT SIZE	−0.2389	−0.0716	−0.0412	−0.0103	−0.0016	0.0102	0.0049	−0.0145	†**−1.1708**	†**−1.1787**

participants, duplicating (cloning) them to form identical groups, and then administering a different drug to each group. To have exact copies of a representative sample to test on is an ideal case for an experimenter, since (1) within-group variation controls for confounding effects, and (2) having identical groups ensures that if differences between groups are detected, it can only be as a result of the drug. Here we are fortunate to be dealing with software entities and not human beings, which allows us the luxury of this ideal setup (see Appendix E for a further discussion on the validity of the RCT approach).

4. Results

A visualisation of our findings is presented in Figure 3. For each initialisation, we plot densities summarising the results for (a) training speed and (b) generalisation. Visually our analysis seems to indicate that the average effect on training speed and generalisation for the critical initialisation is quite similar to the average effect of alternative initialisations, except at the extremes. To make this conclusion more concrete, we conduct a statistical analysis of the results.

Statistical analysis.[1] Using the Iman-Davenport omnibus test, we reject the null hypothesis of no difference between the different initialisations for both training speed and generalisation, with a p-value equal to 2.2×10^{-16} (practically zero). This is somewhat unsurprising, since there are clear differences between the initialisations closer to criticality and those at the extremes (E1 and E2). Therefore, given that we have rejected H_0, we also conduct post-hoc tests.

Table 1 provides pairwise comparisons between the critical initialisation and the alternatives. For mean training speed, we find that only the initialisations at the extremes, i.e. close to $\sigma_w^2(\alpha)$ and $\sigma_w^2(\beta)$, give significantly different results. These include initialising very close to zero (L4) and very large initialisations (E1 and E2). For the initialisations around criticality the differences are not statistically significant. For generalisation, it seems that the alternative initialisations are more sensitive to deviations from criticality (only R1 and L3 indicate no statistically significant difference). However, given the large scale of our study we are able to detect very fine differences. Therefore, even when differences are significant, it is important to consider the sizes of their effects.

Effect sizes. The purpose of computing effect sizes is to gauge whether statistically significant differences in effects are actually meaningful as measured by their magnitude. For a metric τ, we define the effect size for an alternative initialisation $a \in \mathcal{A}$, as $d_a(\tau) = [\mu_a(\tau) - \mu_{\text{crit}}(\tau)]/\text{sd}_{\text{crit}}(\tau)$, where $\text{sd}_{\text{crit}}(\tau)$ is the standard deviation of τ for the critical initialisation. This definition of effect size for a given quantity is often referred to as Cohen's d (Cohen, 1988). In our context, a value of $d = 1$ can be interpreted as a difference in effects equal to one standard deviation away from the mean of criticality. Effect sizes are typically considered to be large, i.e. meaningful, for $d \geq 0.8$. The effect sizes for all the alternative initialisations are given in Table 1, where the direction of an effect is indicated by its

[1] Although the results appear to be multimodal, our non-parametric tests are based on ranking and therefore do not make assumptions regarding the underlying distribution. Our tests are therefore still appropriate.

sign (negative indicating a worse performance when compared to criticality). Effect sizes larger than 0.8 in absolute value are marked with the † symbol. As suggested by the plots in Figure 3, the majority of initialisations around criticality with statistically significant differences in generalisation, as shown in Table 1, have negligible effects sizes. The only meaningful effects are again at the extremes. Finally, we note that the above findings also hold when just considering standard ReLU networks without dropout (shown in Appendix D).

5. Conclusion

At large depth, critical initialisation for neural networks is often considered crucial for success in training and generalisation. However, recent work has shown that dropout, a popular regularisation strategy, limits the depth to which networks can be trained. Given this depth limit, we explore whether initialising at criticality still matters or whether it is possible that alternative, non-critical initialisations (less suited for stable signal propagation at extreme depth) provide any previously undiscovered benefits over critical initialisation. We conducted a large-scale controlled experiment by training over 12000 neural networks. A systematic statistical analysis of training speed and generalisation performance showed that, for a wide range of alternative initialisations around criticality, there is no statistically significant difference between these initialisations and the critical initialisation. Our analysis provides strong evidence that, for moderately deep feedforward ReLU networks (as well as those whose depth is constrained by dropout), there is little to be gained by searching for alternative initialisation schemes.

We emphasize the value of the methodology presented in this paper. Initialisation aside, the methodology can be followed in a generic way to rigorously test the effects of any design component of interest associated with a particular machine learning algorithm. Since statistical rigour is often lacking in empirical machine learning research, we hope that this approach might serve as a useful template for more rigorous investigations.

References

Baldi, P., Sadowski, P.J., 2013. Understanding dropout, in: Advances in Neural Information Processing Systems, pp. 2814–2822.

Chen, M., Pennington, J., Schoenholz, S.S., 2018. Dynamical isometry and a mean field theory of RNNs: Gating enables signal propagation in recurrent neural networks. Proceedings of the International Conference on Machine Learning .

Cohen, J., 1988. Statistical power analysis for the behavioral sciences. 2d ed. Hillsdale, N.J.: Lawrence Erlbaum.

Demšar, J., 2006. Statistical comparisons of classifiers over multiple data sets. Journal of Machine learning research 7, 1–30.

Finner, H., 1993. On a monotonicity problem in step-down multiple test procedures. Journal of the American Statistical Association 88, 920–923.

Friedman, M., 1937. The use of ranks to avoid the assumption of normality implicit in the analysis of variance. Journal of the american statistical association 32, 675–701.

Gal, Y., Ghahramani, Z., 2016. Dropout as a bayesian approximation: Representing model uncertainty in deep learning, in: international conference on machine learning, pp. 1050–1059.

Gal, Y., Hron, J., Kendall, A., 2017. Concrete dropout, in: Advances in Neural Information Processing Systems, pp. 3581–3590.

García, S., Fernández, A., Luengo, J., Herrera, F., 2010. Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power. Information Sciences 180, 2044–2064.

Garcia, S., Herrera, F., 2008. An extension on"statistical comparisons of classifiers over multiple data sets"for all pairwise comparisons. Journal of Machine Learning Research 9, 2677–2694.

Ghiasi, G., Lin, T.Y., Le, Q.V., 2018. Dropblock: A regularization method for convolutional networks, in: Advances in Neural Information Processing Systems, pp. 10727–10737.

Glorot, X., Bengio, Y., 2010. Understanding the difficulty of training deep feedforward neural networks, in: Proceedings of the International Conference on Artificial Intelligence and Statistics, pp. 249–256.

Gomez, A.N., Zhang, I., Swersky, K., Gal, Y., Hinton, G.E., 2018. Targeted dropout .

He, K., Zhang, X., Ren, S., Sun, J., 2015. Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification, in: Proceedings of the IEEE International Conference on Computer Vision, pp. 1026–1034.

Hendrycks, D., Gimpel, K., 2016. Adjusting for dropout variance in batch normalization and weight initialization. arXiv preprint arXiv:1607.02488 .

Iman, R.L., Davenport, J.M., 1980. Approximations of the critical region of the fbietkan statistic. Communications in Statistics-Theory and Methods 9, 571–595.

Kendall, J., 2003. Designing a research project: randomised controlled trials and their principles. Emergency Medicine Journal 20, 164–168.

Kingma, D.P., Salimans, T., Welling, M., 2015. Variational dropout and the local reparameterization trick, in: Advances in Neural Information Processing Systems, pp. 2575–2583.

Krizhevsky, A., Hinton, G., 2009. Learning multiple layers of features from tiny images. Technical Report. University of Toronto.

LeCun, Y., Bottou, L., Bengio, Y., Haffner, P., 1998. Gradient-based learning applied to document recognition. Proceedings of the IEEE 86, 2278–2324.

Molchanov, D., Ashukha, A., Vetrov, D., 2017. Variational dropout sparsifies deep neural networks, in: Proceedings of the 34th International Conference on Machine Learning-Volume 70, JMLR. org. pp. 2498–2507.

Pennington, J., Schoenholz, S., Ganguli, S., 2017. Resurrecting the sigmoid in deep learning through dynamical isometry: theory and practice, in: Advances in Neural Information Processing Systems, pp. 4788–4798.

Poole, B., Lahiri, S., Raghu, M., Sohl-Dickstein, J., Ganguli, S., 2016. Exponential expressivity in deep neural networks through transient chaos, in: Advances in Neural Information Processing Systems, pp. 3360–3368.

Pretorius, A., Van Biljon, E., Kroon, S., Kamper, H., 2018. Critical initialisation for deep signal propagation in noisy rectifier neural networks, in: Advances in Neural Information Processing Systems, pp. 5722–5731.

Santafe, G., Inza, I., Lozano, J.A., 2015. Dealing with the evaluation of supervised classification algorithms. Artificial Intelligence Review 44, 467–508.

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

Schoenholz, S.S., Gilmer, J., Ganguli, S., Sohl-Dickstein, J., 2017. Deep information propagation. Proceedings of the International Conference on Learning Representations .

Srivastava, N., Hinton, G.E., Krizhevsky, A., Sutskever, I., Salakhutdinov, R., 2014. Dropout: a simple way to prevent neural networks from overfitting. Journal of Machine Learning Research 15, 1929–1958.

Wager, S., Wang, S., Liang, P.S., 2013. Dropout training as adaptive regularization, in: Advances in Neural Information Processing Systems, pp. 351–359.

Wan, L., Zeiler, M., Zhang, S., Cun, Y.L., Fergus, R., 2013. Regularization of neural networks using dropconnect, in: Proceedings of the 30th International Conference on Machine Learning, pp. 1058–1066.

Wang, S., Manning, C., 2013. Fast dropout training, in: Proceedings of the 30th International Conference on Machine Learning, pp. 118–126.

Xiao, H., Rasul, K., Vollgraf, R., 2017. Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms. arXiv preprint arXiv:1708.07747 .

Xiao, L., Bahri, Y., Sohl-Dickstein, J., Schoenholz, S.S., Pennington, J., 2018. Dynamical isometry and a mean field theory of CNNs: How to train 10,000-layer vanilla convolutional neural networks. Proceedings of the International Conference on Machine Learning .

Appendix

A. *Pseudo random design construction*

A.1. *Initial / stage-one designs*

First we construct incomplete neural network designs by taking the random Cartesian product of the form:

$$\Omega_* = \mathcal{W} \times \mathcal{B} \times \mathcal{O} \times \mathcal{M} \times \mathcal{L}. \tag{A.1}$$

To ensure balanced designs are created (in the sense that the design space contains an equal number of each item in any of the above sets), we do not sample from the sets with a uniform random probability of selecting each element, but rather concatenate random permutations of each set and pair configurations across this. This process is best illustrated with a small example:

Suppose we only have two hyper-parameter choices: the width of the hidden layers and the learning rate. We would then want to generate pairs of layer-width-learning-rate samples. Our method is to:

1. concatenate random permutations of each set:
   $\widehat{\mathcal{W}} = [\text{permute}(\mathcal{W}), \ldots, \text{permute}(\mathcal{W})] =$
   $[600, 400, 800, 800, 600, 400, 600, 400]$ (for example)
   $\widehat{\mathcal{L}} = [\text{permute}(\mathcal{L}), \ldots, \text{permute}(\mathcal{L})] =$
   $\left[10^{-4}, 10^{-3}, 10^{-5}, 10^{-6}, 10^{-3}, 10^{-6}, 10^{-5}, 10^{-4}\right]$ (for example)
2. sequentially pair these concatenated sets:
   $\Omega_*^{(i)} = \left(\widehat{\mathcal{W}}^{(i)}, \widehat{\mathcal{L}}^{(i)}\right)$
   $\therefore \Omega_*^{(0)} = \left(600, 10^{-4}\right); \Omega_i^{(1)} = \left(400, 10^{-3}\right)$; etc

We then duplicate these combinations for each dropout rate we wish to test. Subsequently, we generate the set of viable correlation information preserving depths based on the dropout rate that is present in each configuration. We use the same setup as above to pair viable depths with incomplete combinations to form complete combinations.

Note that Adam does not support momentum. Thus, when Adam and momentum values were paired, the momentum parameter was ignored when creating the network.

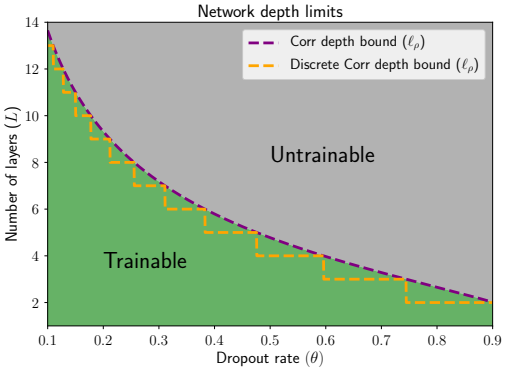


Fig. A.1. ***Discrete trainable depth boundary*****. The depth to which networks with dropout are trainable for different dropout rates (dashed purple line). Discrete depths for each dropout rate is shown by the dashed orange line. When creating networks in practice, the discrete bound should be considered.**

A.2. *Complete / stage-two designs*

A process identical to the above is followed to match incomplete designs Ω_* with viable network depths based on the dropout rate of the group. We construct depth sets $\mathcal{D}_\theta \subset \mathcal{D}$ such that $\mathcal{D}_\theta$ only contains depths to which correlation information can propagate for the given dropout rate θ –see Figure A.1 for a graphical representation of this and note that, since we are working with networks with discrete numbers of layers, we follow the discretised boundary. We then concatenate permutations of this set and use this to complete the designs and form our final designs Ω. Let us illustrate this by continuing the above example for $\theta = 0.7$:

1. construct $\mathcal{D}_{0.7} = \{2, 3, 4, 5, 6, 7\}$ using (6) (we omit networks with 0 and 1 hidden layers to ensure some measure of expressiblity)
2. concatenate random permutations of $\mathcal{D}_{0.7}$:
   $\widehat{\mathcal{D}}_{0.7} = [\text{permute}(\mathcal{D}_{0.7}), \ldots, \text{permute}(\mathcal{D}_{0.7})] =$
   $[6, 3, 7, 2, 5, 4, 2, 3, 6, 5, 4, 7]$ (for example)
3. sequentially pair this concatenated set with the incomplete designs:
   $\Omega^{(i)} = \left(\Omega_*^{(i)}, \widehat{\mathcal{D}}_{0.7}^{(i)}, 0.7\right)$
   $\therefore \Omega^{(0)} = \left(600, 10^{-4}, 6, 0.7\right); \Omega^{(1)} = \left(400, 10^{-3}, 3, 0.7\right)$; etc

These four design sets, one for each value in $\mathcal{R}$, are then duplicated 44 times (once for each combination of initialisation candidate in $\mathcal{A}$ and dataset in $\mathcal{X}$).

B. *Method for generating network design dependent initialisations*

Inputs:

- θ: dropout rate
- L: depth of the network
- S: the number of candidates on either side of critical, within the core group, to be generated
- E: the number of "extreme" candidates (those far greater than critical) to be generated
- β: the largest positive value that can be represented given the floating point precision of the current computer
- α: the smallest positive value that can be represented given the floating point precision of the current computer

Steps:

1. calculate $\sigma^2_{\text{critical}} = 2(1 - \theta)$
2. calculate $\sigma_w^2(\beta)$ using (10)
3. generate the set of "extreme" samples, $\boldsymbol{\sigma^2_{\text{extreme}}}$:
   3..1 select the first "extreme" candidate such that it is within the depth boundary:
   $\sigma^2_{\text{extreme},E} = 0.9\sigma_w^2(\beta)$
   3..2 recursively calculate the subsequent "extreme" candidates such that they are logarithmically spaced:
   $\sigma^2_{\text{extreme},e} = \frac{1}{2}\sigma^2_{\text{extreme},(e+1)}$ for $e \in \{1, 2, ..., E-1\}$
4. calculate $\sigma_w^2(\alpha)$ using (9)

5. generate the set of logarithmically spaced samples less than critical, $\sigma^2_{\textbf{left}}$:
$\sigma^2_{\text{left},s} = \sigma^2_{\text{critical}} - \frac{0.9}{2^{s-1}}(\sigma^2_{\text{critical}} - \sigma^2_w(\alpha))$ for $s \in \{1, 2, ..., S\}$
6. generate the set of samples just greater than critical, $\sigma^2_{\textbf{right}}$, by reflecting $\sigma^2_{\textbf{left}}$ about the critical initialisation:
$\sigma^2_{right,s} = \sigma^2_{\text{critical}} - (\sigma^2_{left,s} - \sigma^2_{\text{critical}}) = \sigma^2_{\text{critical}} + \frac{0.9}{2^{s-1}}(\sigma^2_{\text{critical}} - \sigma^2_w(\alpha))$ for $s \in \{1, 2, ..., S\}$

The set of candidate initialisations is then $\left\{\sigma^2_{\textbf{left}}, \sigma^2_{\text{critical}}, \sigma^2_{\textbf{right}}, \sigma^2_{\textbf{extreme}}\right\}$.

C. Design and corresponding initialisation examples

Table C.1 shows 12 sampled designs and their corresponding initialisations. These samples are representative of our full set of design samples and give a good idea of typical network parameters. While there appears to be no difference between core initialisation values across samples with the same dropout rate, this is actually not the case. Changes are simply typically too small to be seen with only 3 decimal places. This is due to the rate of change of $\sigma^2_w(\alpha)$ being very low for networks of shallow to moderate depth (roughly 20 hidden layers or less).

D. Additional results

In this section we provide additional statistical analyses per dropout rate as well as with zero dropout. These results are given in Tables D.1, D.2, D.3 and D.4.

E. On the validity of the RCT approach

We performed two auxiliary studies to ensure the effectiveness of the RCT setup.

Firstly, we wanted to ensure the methodology was set up correctly and could identify known performance differences. To achieve this, we created a smaller scale scenario very similar to the study described in the main text but instead using initialisation as "intervention", we use the activation function. Networks with non-linear activation functions are more expressive and should be able to outperform their linear counterparts. Furthermore, the ReLU activation function does not suffer from saturation or vanishing gradients and typically outperforms the sigmoid when using random Gaussian initialisation. These are well established results, frequently demonstrated in the literature. Therefore, we decided to construct an RCT where the interventions are the following activations: linear, sigmoid, and ReLU. Each network was initialised critically and trained on MNIST for 1955 iterations using a batch size of 128. This RCT consisted of 1761 trained networks.

The results of the above experiment are exactly as expected and are given in Figure E.1. ReLU networks performed best overall. The performance of linear networks were capped significantly below that of ReLU networks. Finally, the sigmoidal networks were able to perform better than linear networks and as well as ReLU networks in the best case. However, the distribution over performance for the sigmoid exhibits a long tail towards low accuracy due to vanishing gradient issues, causing network training to stall. Furthermore, pairwise comparisons with post-hoc tests between ReLU and the other activations yield

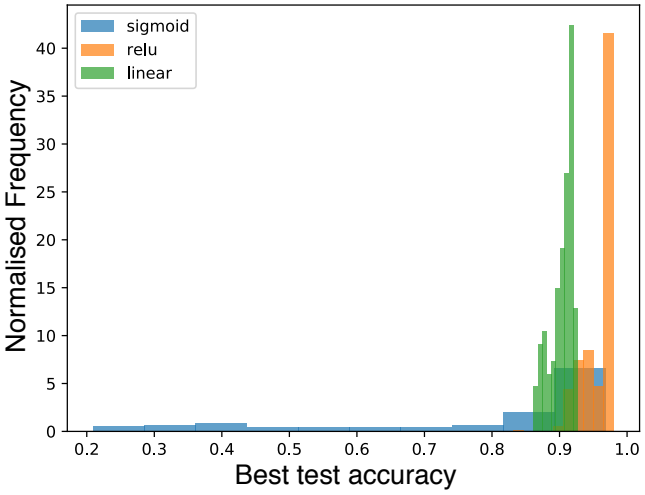


Fig. E.1. ***Best test accuracy distribution per activation function.*** **The test RCT confirms the validity of this setup as it clearly confirms well known results: (1) ReLU networks typically outperform linear and sigmoidal networks, (2) the best performing sigmoidal networks outperform linear networks and perform comparatively to ReLU networks, but are typically more difficult to train due to vanishing gradients.**

p-values that are practically zero, indicating significantly different mean performances with meaningful effects sizes (-2.65 for linear and -8.42 for sigmoid).

After confirming the validity of our RCT setup, we aimed to control for more sophisticated training procedures, such as learning rate decay. Learning rate decay is expected to have a complex temporal interaction with other design components during training and poses a challenge as a potential confounder for the RCT to control. We theorise that adding any mechanism that generally improves performance, such as learning rate decay, should have an average effect (taken over sampled designs) that is roughly equal across groups.

To test this, we construct two RCTs nearly identical to the above. All networks make use of the ReLU activation function and the interventions in this case are initialisation schemes: He (He et al., 2015), Xavier (Glorot and Bengio, 2010) and Orthogonal (Saxe et al., 2014). One of these RCTs makes use of learning rate decay and the other does not. In this way, we can compare the statistical findings of each and confirm whether they agree. If the test results do agree, it means that the RCT has successfully controlled for the confounding effects of learning rate in both cases, i.e. with and without decay. Figure E.2 shows best test accuracy distributions (a) without and (b) with learning rate decay. It is clear that although learning rate decay may improve the overall performance of all groups, the relative performance differences between groups remain roughly the same. Table E.1 gives the test results for each RCT. The conclusions closely match between the two trials. Therefore, we conclude that the RCT as described and performed in the main text provides a very general approach to isolating the effects of a particular intervention.

A final possible objection to this setup is that samples might not always be independent due to correlations between selected hyperparameters. This could be the case when sampling across a coarse grid for a single hyperparameter. However, in our setup,

Table C.1. 12 example sets of sampled designs and their corresponding initialisations.

– Example designs –

design index	dataset	depth	width	rate	batch size	optimiser	momentum	learning rate
0	CIFAR-10	12	800	0.0	256	Adam	0.0	10^{-5}
1	FashionMNIST	15	800	0.0	64	SGD	0.5	10^{-6}
2	MNIST	4	400	0.0	256	RMSprop	0.0	10^{-5}
3	CIFAR-10	2	400	0.5	256	RMSprop	0.0	10^{-5}
4	CIFAR-100	3	800	0.5	64	SGD	0.5	10^{-6}
5	FashionMNIST	4	800	0.5	256	Adam	0.0	10^{-5}
6	CIFAR-10	7	600	0.3	256	Adam	0.0	10^{-3}
7	CIFAR-100	4	600	0.3	64	SGD	0.5	10^{-6}
8	FashionMNIST	5	400	0.3	256	RMSprop	0.0	10^{-5}
9	CIFAR-10	3	600	0.1	32	SGD	0.5	10^{-4}
10	MNIST	8	600	0.1	32	SGD	0.5	10^{-4}
11	CIFAR-10	4	600	0.1	128	Adam	0.0	10^{-3}

– Initialisations –

design index	L4	L3	L2	L1	C	R1	R2	R3	R4	E1	E2
0	0.201	1.101	1.550	1.775	2.000	2.225	2.450	2.899	3.799	1.464×10^{3}	2.926×10^{3}
1	0.205	1.103	1.551	1.776	2.000	2.224	2.449	2.897	3.795	3.346×10^{2}	6.671×10^{2}
2	0.200	1.100	1.550	1.775	2.000	2.225	2.450	2.900	3.800	3.865×10^{9}	7.731×10^{9}
3	0.100	0.550	0.775	0.887	1.000	1.113	1.225	1.450	1.900	8.301×10^{18}	1.660×10^{19}
4	0.100	0.550	0.775	0.888	1.000	1.112	1.225	1.450	1.900	3.142×10^{12}	6.283×10^{12}
5	0.100	0.550	0.775	0.888	1.000	1.112	1.225	1.450	1.900	1.933×10^{9}	3.865×10^{9}
6	0.140	0.770	1.085	1.243	1.400	1.557	1.715	2.030	2.660	2.013×10^{5}	4.026×10^{5}
7	0.140	0.770	1.085	1.243	1.400	1.557	1.715	2.030	2.660	2.706×10^{9}	5.412×10^{9}
8	0.140	0.770	1.085	1.243	1.400	1.557	1.7154	2.030	2.660	3.204×10^{7}	6.408×10^{7}
9	0.180	0.990	1.395	1.598	1.800	2.002	2.205	2.610	3.4204	5.655×10^{12}	1.131×10^{13}
10	0.180	0.990	1.395	1.598	1.800	2.002	2.205	2.610	3.420	5.309×10^{4}	1.062×10^{5}
11	0.180	0.990	1.395	1.598	1.800	2.002	2.205	2.610	3.420	3.479×10^{9}	6.958×10^{9}

Table D.1. No dropout – $\theta = 0$: Post-hoc tests for training speed and generalisation for no dropout ($\theta = 0$). The symbols * indicate a significant p-value and † a large effect size.

Training speed – $H_0 : \mu_{\text{crit}}(\tau_s) = \mu_a(\tau_s), a \in \{L_i, R_i, E_1, E_2\}_{i=1}^{4}$

	L4	L3	L2	L1	R1	R2	R3	R4	E1	E2
Adjusted p-value	$^{*}\mathbf{0}$	$\mathbf{1.99 \times 10^{-11}}$	$\mathbf{3.07 \times 10^{-5}}$	0.0864	0.0529	$\mathbf{9.68 \times 10^{-5}}$	$\mathbf{8.89 \times 10^{-6}}$	$\mathbf{4.71 \times 10^{-7}}$	$^{*}\mathbf{0}$	$^{*}\mathbf{0}$
Effect size	−0.6024	−0.1854	−0.1124	−0.0434	0.0382	0.0716	0.0774	0.0602	$^{\dagger}\mathbf{-1.1085}$	$^{\dagger}\mathbf{-1.1009}$

Generalisation – $H_0 : \mu_{\text{crit}}(\tau_g) = \mu_a(\tau_g), a \in \{L_i, R_i, E_1, E_2\}_{i=1}^{4}$

	L4	L3	L2	L1	R1	R2	R3	R4	E1	E2
Adjusted p-value	$^{*}\mathbf{0}$	$^{*}\mathbf{0.0033}$	0.8549	0.7927	0.1846	$^{*}\mathbf{0.0018}$	$^{*}\mathbf{7.20 \times 10^{-6}}$	$^{*}\mathbf{3.49 \times 10^{-14}}$	$^{*}\mathbf{0}$	$\mathbf{0}$
Effect size	−0.5224	−0.1592	−0.0975	−0.0325	−0.0027	0.0077	−0.0032	−0.0374	$^{\dagger}\mathbf{-1.0008}$	$^{\dagger}\mathbf{-1.0191}$

Table D.2. Dropout – $\theta = 0.1$: Post-hoc tests for training speed and generalisation for dropout with rate $\theta = 0.5$. The symbols * indicate a significant p-value and † a large effect size.

Training speed – $H_0 : \mu_{crit}(\tau_s) = \mu_a(\tau_s), a \in \{L_i, R_i, E_1, E_2\}_{i=1}^4$

	L4	L3	L2	L1	R1	R2	R3	R4	E1	E2
Adjusted p-value	$^*\mathbf{0}$	$^*\mathbf{0.0203}$	0.9219	0.6552	0.9395	0.2258	$^*\mathbf{0.0021}$	$^*\mathbf{2.59 \times 10^{-8}}$	$^*\mathbf{0}$	$^*\mathbf{0}$
Effect size	−0.2578	−0.0723	−0.0189	0.0015	0.0311	0.0261	0.0202	−0.0009	$^\dagger\mathbf{-1.1328}$	$^\dagger\mathbf{-1.1209}$

Generalisation – $H_0 : \mu_{crit}(\tau_g) = \mu_a(\tau_g), a \in \{L_i, R_i, E_1, E_2\}_{i=1}^4$

	L4	L3	L2	L1	R1	R2	R3	R4	E1	E2
Adjusted p-value	$^*\mathbf{0}$	0.4475	0.8743	0.7033	0.7874	0.5100	$^*\mathbf{0.0193}$	$^*\mathbf{1.41 \times 10^{-6}}$	$^*\mathbf{0}$	$^*\mathbf{0}$
Effect size	−0.3091	−0.0976	−0.0650	−0.0237	0.0139	0.0236	0.0221	0.0167	$^\dagger\mathbf{-1.1503}$	$^\dagger\mathbf{-1.1572}$

Table D.3. Dropout – $\theta = 0.3$: Post-hoc tests for training speed and generalisation for dropout with rate $\theta = 0$. The symbols * indicate a significant p-value and † a large effect size.

Training speed – $H_0 : \mu_{crit}(\tau_s) = \mu_a(\tau_s), a \in \{L_i, R_i, E_1, E_2\}_{i=1}^4$

	L4	L3	L2	L1	R1	R2	R3	R4	E1	E2
Adjusted p-value	$^*\mathbf{0.0051}$	0.0727	0.0727	0.2097	0.1885	$^*\mathbf{0.0090}$	$^*\mathbf{1.49 \times 10^{-6}}$	$^*\mathbf{4.88 \times 10^{-14}}$	$^*\mathbf{0}$	$^*\mathbf{0}$
Effect size	−0.1348	−0.0281	−0.0020	−0.0009	−0.0066	−0.0047	−0.0124	−0.0336	$^\dagger\mathbf{-1.1267}$	$^\dagger\mathbf{-1.1304}$

Generalisation – $H_0 : \mu_{crit}(\tau_g) = \mu_a(\tau_g), a \in \{L_i, R_i, E_1, E_2\}_{i=1}^4$

	L4	L3	L2	L1	R1	R2	R3	R4	E1	E2
Adjusted p-value	$^*\mathbf{0.0409}$	$^*\mathbf{0.0018}$	$^*\mathbf{0.0075}$	$^*\mathbf{0.0409}$	0.3975	$^*\mathbf{0.0296}$	$^*\mathbf{1.03 \times 10^{-5}}$	$^*\mathbf{6.88 \times 10^{-14}}$	$^*\mathbf{0}$	$^*\mathbf{0}$
Effect size	−0.1144	−0.0416	−0.0074	0.0039	0.0035	0.0093	0.0128	−0.0113	$^\dagger\mathbf{-1.2549}$	$^\dagger\mathbf{-1.2575}$

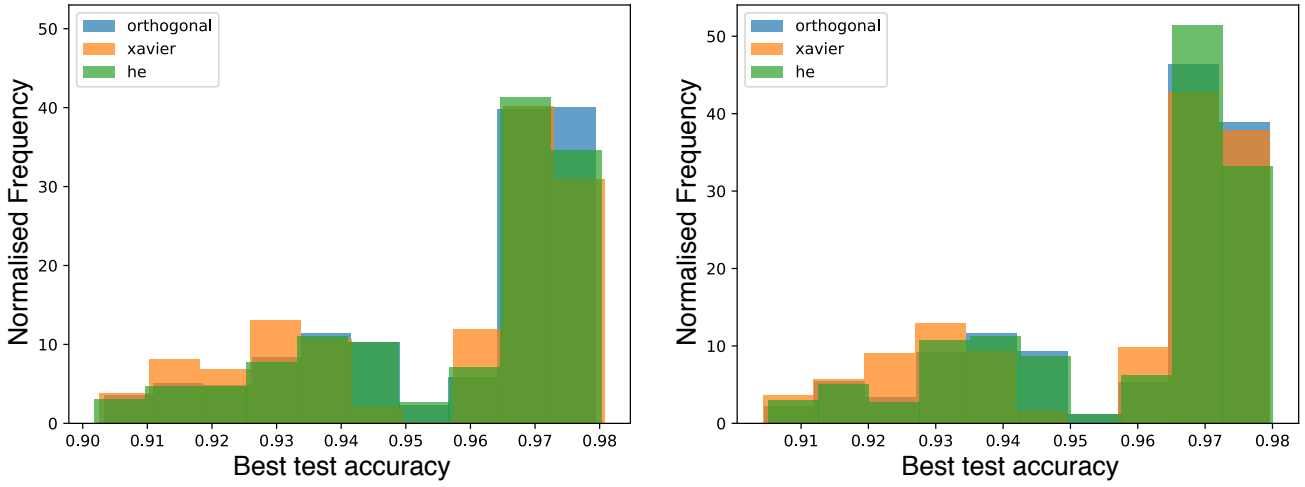
Table D.4. Dropout – $\theta = 0.5$: Post-hoc tests for training speed and generalisation for dropout with rate $\theta = 0$. The symbols * indicate a significant p-value and † a large effect size.

Training speed – $H_0 : \mu_{crit}(\tau_s) = \mu_a(\tau_s), a \in \{L_i, R_i, E_1, E_2\}_{i=1}^4$

	L4	L3	L2	L1	R1	R2	R3	R4	E1	E2
Adjusted p-value	0.1170	$^*\mathbf{7.46 \times 10^{-5}}$	$^*\mathbf{0.0075}$	0.0843	0.1188	$^*\mathbf{0.0002}$	$^*\mathbf{1.63 \times 10^{-8}}$	$^*\mathbf{2.66 \times 10^{-14}}$	$^*\mathbf{0}$	$^*\mathbf{0}$
Effect size	−0.0491	0.01421	0.0105	0.0119	−0.0227	−0.0073	−0.0210	−0.0404	$^\dagger\mathbf{-1.1158}$	$^\dagger\mathbf{-1.1171}$

Generalisation – $H_0 : \mu_{crit}(\tau_g) = \mu_a(\tau_g), a \in \{L_i, R_i, E_1, E_2\}_{i=1}^4$

	L4	L3	L2	L1	R1	R2	R3	R4	E1	E2
Adjusted p-value	$^*\mathbf{0.0009}$	$^*\mathbf{6.18 \times 10^{-6}}$	$^*\mathbf{0.0016}$	0.0591	0.2637	$^*\mathbf{0.0010}$	$^*\mathbf{1.03 \times 10^{-5}}$	$^*\mathbf{5.08 \times 10^{-12}}$	$^*\mathbf{0}$	$^*\mathbf{0}$
Effect size	-3.37×10^{-2}	4.44×10^{-3}	1.01×10^{-5}	8.92×10^{-3}	-2.04×10^{-2}	4.63×10^{-4}	-1.18×10^{-2}	$-2.64e-02$	$^\dagger\mathbf{-1.2796}$	$^\dagger\mathbf{-1.2843}$



(a) no learning rate decay

(b) learning rate decay

Fig. E.2. ***Comparison of RCTs with and without learning rate decay*****. Best test accuracy distributions per initialisation are shown. It is clear that although learning rate decay improves the overall performance of all groups, the relative performance between groups remains the same.**

Table E.1. Statistical tests using learning rates with and without decay: Post-hoc tests for generalisation using and not using learning rate decay. Comparisons are between He and Xavier and He and orthogonal initialisation with the null hypothesis H_0 of no difference. The relative differences as detected by the tests remain the same between the two approaches, thus the RCT has successfully isolated only the effects of the initialisation.

Generalisation		
With decay	Orthogonal	Xavier
Adjusted p-value	0.1972	***0.0904**
Effect size	0.0124	−0.1176
Without decay	Orthogonal	Xavier
Adjusted p-value	0.9183	***0.0067**
Effect size	0.0156	−0.1130

we randomly sample over multiple grids of hyperparameters for each design ("participant in our study"). Thus for correlations between *designs* to persist, they must do so simultaneously across multiple *hyperparameters* (dimensions of the design space) to influence the results. Given the high-dimensionality of the design space, we feel it safe to treat each design as an independent sample from the population.