# Non-Gaussianity in the distribution of outputs from neural network ensembles

**Joseph M. Antognini**
Whisper AI
San Francisco, California
joe.antognini@gmail.com

## Abstract

There has been a recent surge of interest in modeling neural networks (NNs) as Gaussian processes. In the limit of infinite width a fully connected NN becomes equivalent to a Gaussian process at initialization. Here we demonstrate that for an ensemble of fully connected networks the distribution of outputs at initialization is well described by the Edgeworth expansion, that is, a Gaussian perturbed by Hermite polynomials. The scale of the perturbation is inversely proportional to the number of units in the NN for initializations drawn from a symmetric distribution. By training a large ensemble of fully connected NNs on a one dimensional regression task and a restricted version of MNIST we show through measurements the scale of this perturbation, along with the skew and kurtosis, that the output distribution becomes less Gaussian under the influence of SGD. We find, however, that once training has converged the output distribution remains close enough to a Gaussian that it is still well described by the Edgeworth expansion and the distribution does not acquire highly non-Gaussian features like bimodality or extremely heavy tails. In the restricted MNIST task the output distribution is closest to Gaussian for examples drawn from the training set, and least Gaussian for examples randomly drawn from a uniform distribution. Finally we show that early in training the distribution is least Gaussian for examples with the lowest loss.

## 1 Introduction

Today it is well known that there is a deep connection between modern, highly overparameterized neural networks (NNs) and Gaussian processes. A foundational result by Neal (1996) and Williams (1997) demonstrated that a randomly initialized NN with a single hidden layer of infinite width is equivalent to a Gaussian process so long as the weights of the NN are drawn from a distribution with finite variance. Although the covariance between different hidden units of the NN is zero, these works showed that the covariance between a single hidden unit with different inputs is non-zero, thereby making learning possible.

Although Neal (1996) and Williams (1997) only studied the case of NNs with a single hidden layer, several recent works have extended this insight to show that NNs with multiple, possibly convolutional, layers of infinite width are also Gaussian processes (Lee et al., 2017; Matthews et al., 2018; Novak et al., 2019; Garriga-Alonso et al., 2018). And while these works only studied NNs at initialization, Jacot et al. (2018) showed that gradient descent on a NN corresponds to applying a tangent kernel to an equivalent Gaussian process, and Arora et al. (2019) has recently weakened the conditions on this proof. Yang (2019) extended this neural tangent kernel result to convolutional NNs. Lee et al. (2019) empirically showed that application of the tangent kernel closely match the predicted dynamics of a linearized deep NN.

Preprint. Under review.

Although NNs have grown dramatically in size, they have remained frustratingly finite. In practice, most practitioners tend to choose widths in the range 128–1024. Especially as deep learning models have moved onto mobile devices there has been substantial effort into finding smaller NNs so that they can perform inference quickly and efficiently on low power devices (e.g., Hinton et al., 2015; Han et al., 2015; Howard et al., 2017; Sandler et al., 2018; Howard et al., 2019).

In this paper we attempt to bridge the divide between the infinite NNs of theory and the merely large NNs of practice. We show that for an ensemble of randomly initialized finite NNs the output distribution is not Gaussian, but can instead be well described by the Edgeworth expansion. We use the parameters of this expansion along with the statistical moments to measure the magnitude of the deviation from Gaussianity and find that the output distribution generally becomes less Gaussian after training on a task with stochastic gradient descent (SGD). We show that the different classes of inputs exhibit output distributions with different degrees of non-Gaussianity, with inputs from the training set being the most Gaussian and random inputs being the least. Finally we show that in the early stages of training there is a strong negative correlation between the mean loss of an example across the ensemble and the non-Gaussianity of the output distribution.

## 2 Finite neural networks and deviations from Gaussian processes

In this paper we shall restrict our attention to fully connected NNs with a single hidden layer consisting of $N$ hidden units and a single output. Let us write the output of the NN given an $M$-dimensional input, $\mathbf{x}$, as

$$f(\mathbf{x}) \equiv \sum_{i=0}^{N} h\left(\sum_{j=0}^{M} u_{ij} x_j + b_i\right) v_i, \tag{1}$$

where $u_{ij}$ are the input-to-hidden weights, $b_i$ are the bias parameters of the hidden units, $v_i$ are the hidden-to-output weights, and $h(x)$ is the activation function. At initialization the parameters $\mathbf{u}$, $\mathbf{v}$, and $\mathbf{b}$ are generally sampled independently from a set of probability distributions with finite variance.[1] Neal (1996) and Williams (1997) observed that for fixed input $\mathbf{x}$, the output of the NN is given by the sum of $N$ random variates with finite variance (assuming that the activation function is sufficiently well behaved). By the central limit theorem, as the width of the NN tends towards infinity, the distribution of NN outputs for any given input will be a Gaussian distribution. This implies, then, that the NN is equivalent to a Gaussian process in this limit. For any activation function and set of parameter distributions the covariance between any two inputs can in principle be calculated, and several authors have done this calculation explicitly (e.g., Cho & Saul, 2009; Lee et al., 2017; Arora et al., 2019).

### 2.1 Measurements of the deviation from a Gaussian process

#### 2.1.1 The Edgeworth expansion

The relationship between NNs and Gaussian processes therefore depends critically on the outer summation in Eq. 1 being over an infinite number of independent variates so that the central limit theorem applies. In practice, however, this sum is taken over a large, but finite, number. The limiting distribution of large but finite sums of random variates was first studied by Edgeworth (1908) who derived what is now known to be an asymptotic series of Hermite polynomials. Specifically, a distribution with mean $\mu_0$ and standard deviation $\sigma$ can be well approximated by a Gaussian perturbed by a $K$-term series of Hermite polynomials:

$$f(x) \simeq \mathcal{N}(\mu_0, \sigma^2)\left[1 + \sum_{k=3}^{K} \frac{1}{N^{(k-2)/2}} c_k H_k\left(\frac{x-\mu_0}{\sigma}\right)\right], \tag{2}$$

where $H_k$ is the $k^{\text{th}}$ probabilist's Hermite polynomial, defined by $H_k(x) \equiv (x - D)^k$, with $D$ being the differential operator. The coefficients of the expansion, $c_k$, are related to the moments, $\mu_k$, of the distribution with the first two being given by

$$c_3 = \frac{1}{3!}\kappa_3\mu_3; \qquad c_4 = \frac{1}{4!}\kappa_4\mu_4 + \frac{1}{2 \times 3!^2}\kappa_3^2\mu_6, \tag{3}$$

[1] An important exception is orthogonal initialization (Saxe et al., 2013) which places an orthogonality constraint on the weight matrices thereby causing the individual parameters to lose their independence.

where $\kappa_k$ is the $k^{\text{th}}$ cumulant of the distribution. Being an asymptotic series, the Edgeworth expansion converges to $f(x)$ as $N \to \infty$ for a fixed number of terms (as it must according to the central limit theorem), but the series itself diverges as $k \to \infty$. For a pedagogic, but rigorous, derivation of the Edgeworth expansion see Hansen (2006). Other rigorous treatments include Esseen (1945), Feller (1966), and Blinnikov & Moessner (1998), who translate a Russian derivation by Petrov (1972). It is important to note that the Edgeworth expansion is not, in general, a true probability density function since it is possible for the function to exhibit negative values for arbitrary $c_k$. Nevertheless, for small $c_k$ or large $N$ the distribution will either be a valid probability distribution or asymptotically close to one.

Given a sample from the distribution $f(x)$ we would like to determine the best fit for the first two coefficients, $c_3$ and $c_4$. To do this let us consider the cumulative distribution function (CDF) of $f(x)$:

$$F(x) \simeq \text{erf}(\mu_0, \sigma) + \mathcal{N}(x; \mu_0, \sigma^2) \left[ \frac{c_3}{\sqrt{N}} H_2 \left( \frac{x - \mu_0}{\sigma} \right) + \frac{c_4}{N} H_3 \left( \frac{x - \mu_0}{\sigma} \right) \right]. \tag{4}$$

The coefficients $c_3$ and $c_4$ can then be estimated by subtracting off the CDF of a Gaussian with the sample mean and variance and performing a least-squares estimate on the residuals.

### 2.1.2 Method of moments

A more straightforward way to measure the deviation of a nearly Gaussian distribution from a Gaussian is to measure the moments of the distribution and compare them to a Gaussian since all moments of order three and higher are fixed for a Gaussian. In this paper we will measure the skew and excess kurtosis, which are closely related to the third and fourth moments. In this paper we represent the skew and kurtosis by $\kappa_3$ and $\kappa_4$, respectively, and will refer to them as "shape parameters".

Note that although the first two parameters of the Edgeworth expansion are determined by the third and fourth moments of the underlying distribution, fitting these parameters to a sample measures a different characteristic of the distribution than the sample skew and kurtosis. Due to the exponential damping of the Edgeworth expansion in the tails, fitting the parameters of the Edgeworth expansion will measure the behavior of the distribution close to the mean. By contrast, the sample skew and kurtosis have cubic and quartic dependencies on the distance from the mean and therefore tend to measure the behavior of the tails of the distribution (Darlington, 1970; Westfall, 2014).

### 2.1.3 A note on the generalization to stochastic processes

The Edgeworth expansion and sample skew and kurtosis are properties of a probability distribution. While the distribution of outputs of an ensemble of NNs will follow some probability distribution for fixed input, the behavior of the NN as a whole is in fact a stochastic process since there will generally be some dependence of the distribution given one input with the distribution given another input. In the infinite width limit, this property is captured by treating the NN as a Gaussian process with a non-trivial covariance matrix. Thus, not only will the distribution of outputs for a fixed input be a univariate Gaussian, but the joint distribution for any set of inputs will be described by a multivariate Gaussian with some covariance matrix.

For a finite NN, then, the joint distribution for a set of inputs will be given instead by a multivariate Edgeworth expansion. The multivariate Edgeworth expansion is similar in structure to the univariate case, but consists of a sum of Hermite tensors instead of Hermite polynomials, and the coefficients of this expansion are determined by moment tensors rather than scalars. For an explicit representation see Sellentin et al. (2017) and for proofs concerning its convergence properties see Skovgaard (1986). Similarly, in the method of moments, the skew and kurtosis of the distribution are no longer given by scalars, but instead by rank-3 and rank-4 tensors, respectively (often called the coskewness and cokurtosis). Unfortunately it is challenging enough to train large enough ensembles of NNs to obtain statistically significant deviation measurements for the univariate case. Due to the curse of dimensionality, obtaining these statistics for multivariate distributions requires exponentially larger sample sizes so we do not attempt to measure these coefficients for any joint distributions.

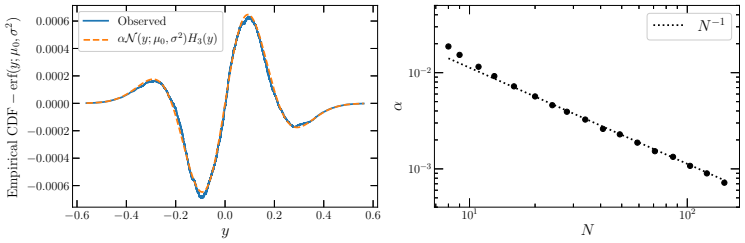


Figure 1: *Left panel:* The difference between the empirical CDF of the output of an ensemble of randomly initialized NNs with the CDF of a Gaussian (solid blue line). The predicted difference from Eq. 4 is shown in the dashed orange line. The empirical CDF was calculated for $10^8$ NNs each with a single hidden layer consisting of 128 units for a fixed input of $x = 1$. *Right panel:* The scaling of the perturbation of the distribution from a Gaussian with the number of hidden units. Each point is the best fit for $\alpha$ of the difference between the empirical CDF with the CDF of the third Hermite polynomial times a Gaussian. The predicted $N^{-1}$ scaling is also shown with a dotted line.

## 3 Deviations from Gaussianity at initialization

To demonstrate the validity of the Edgeworth expansion for a finite NN at initialization, we generate an ensemble of $10^8$ fully connected NNs with a single hidden layer consisting of 128 units and a ReLU activation. We initialize the weights using Glorot uniform initialization (Glorot & Bengio, 2010) and the biases with zeros since this is perhaps the most common initialization distribution and is the default in popular deep learning frameworks like Tensorflow (Abadi et al., 2016) and PyTorch (Paszke et al., 2017). Because we are generating large sets of parameters and performing sensitive statistical tests on the resulting distributions, some care must be taken in choosing an appropriate pseudorandom number generator (PRNG). For our experiments we use the Threefry PRNG (Salmon et al., 2011). Unlike the more popular Mersenne twister (Matsumoto & Nishimura, 1998), Threefry passes all tests in the BigCrush battery from the TestU01 suite (L'Ecuyer & Simard, 2007).

To observe the deviation from Gaussianity at initialization we consider one-dimensional inputs and fix the input to $x = 1$. As detailed in Section 2.1.1 we calculate the empirical CDF and subtract off the CDF of a Gaussian with the sample mean and variance. The resulting residuals are shown in the left panel of Fig. 1 along with the best fit to the expansion coefficients $c_3$ and $c_4$. The symmetry of the initialization distribution imposes the constraint $c_3 = 0$, which we measure to within our statistical uncertainty. By contrast we measure $c_4 \approx 0.106$.

To measure the dependence of the deviation from Gaussianity on the width of the NN, we now initialize a set of NN ensembles identical to the ensemble above, but vary the width of each ensemble from 8 to 148 hidden units and use $10^7$ NNs for each ensemble. We then define $\alpha \equiv c_4/N$ to capture the expected dependence on the NN width. We show the best fit values for $\alpha$ in the right panel of Fig. 1 and find excellent agreement with the expected inverse dependence.

## 4 Deviations from Gaussianity after training with SGD

The results in Section 3 demonstrate that for fixed input, the output distribution of an ensemble of NNs is nearly, but not exactly, Gaussian. We now pursue the question of how this distribution changes over the course of training the NNs in the ensemble. Jacot et al. (2018) proved that if the distribution is Gaussian at initialization, it will remain Gaussian under gradient descent with a squared loss. Although these conditions do not hold for realistic NNs, it is plausible to suppose that under the influence of SGD these small non-Gaussian perturbations decay and the distribution becomes more Gaussian. To study this question we train a NN ensemble on two tasks: fitting a one dimensional curve and a restricted version of MNIST.

### 4.1 A one dimensional regression task

In the first task we train an ensemble of $10^5$ fully connected NNs. As in Section 3 the NNs consist of a single fully connected layer with 128 units and ReLU activations. The weights are initialized

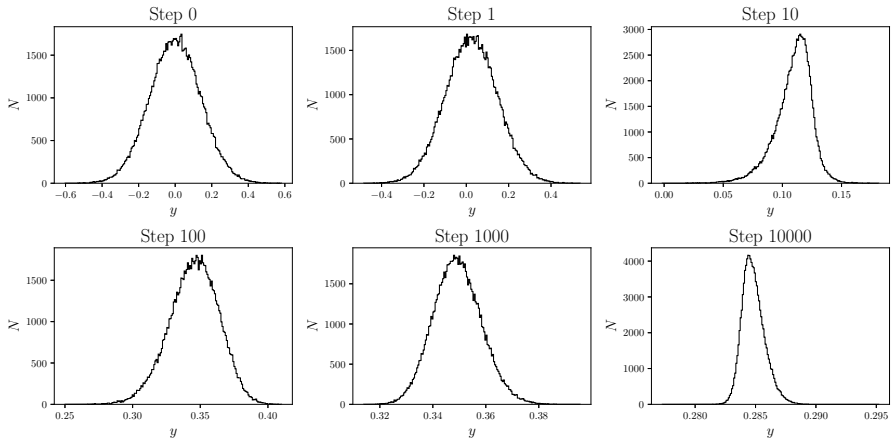


Figure 2: The distribution of outputs of an ensemble of $10^5$ NNs trained to fit the function $e^{-x^2}$ with SGD for a fixed input of $x \approx 1.122$. Although the distribution is quite Gaussian at initialization it becomes noticeably non-Gaussian in the early stages of training before becoming more Gaussian in the intermediate stages of training. In the final stages of training the distribution once again becomes noticeably non-Gaussian.

from the Glorot uniform distribution and the biases are initialized with zeros. We then train the NNs to match the function $e^{-x^2}$ by sampling minibatches of 32 samples in the range $[-3, 3]$ and use a mean square error loss function. We train for 10,000 steps with a learning rate of $10^{-3}$ and capture the distribution of outputs on the range $[-5, 5]$ at steps $10^k$ for $k \in \{0, \ldots, 4\}$. Note that we sample outputs outside the range that the NN has been trained in order to determine if the distributions are qualitatively different in regions where the NN is extrapolating. The distribution of outputs for a fixed input of 1.122 is shown in Fig. 2. As is expected from Neal (1996), at initialization the distribution is quite close to Gaussian. However, in the early stages of training the distribution evolves to become noticeably less Gaussian. At intermediate times during training the distribution becomes more Gaussian, before finally becoming less Gaussian in the final stages of training as the standard deviation of the distribution decreases. We show in Figs. 6 and 7 in the supplementary material the best fits for the Edgeworth and shape parameters across the input distribution and over the course of training. After averaging across the entire input space a general pattern emerges that the output distribution becomes less Gaussian over the course of training.

### 4.2 Restricted MNIST

For a more complicated task we turn to MNIST. Although MNIST is extremely small and simple by the standards of modern deep learning datasets, it is nevertheless expensive to train a sufficiently large ensemble of NNs on the entire problem to obtain statistically significant results. We therefore select the most difficult subtask in the MNIST task of distinguishing between 4s and 9s. So as to be consistent with the assumptions in Jacot et al. (2018) we output a scalar, use a mean squared error loss, and set the target for 4s to be $-1$ and the target for 9s to be 1. We train an ensemble of 60,000 NNs where, as before, each is a single layer fully connected NN with 128 units in the hidden layer and ReLU activations. We train with SGD and a minibatch size of 32 for 10,000 steps using a learning rate of $10^{-3}$. The NNs obtain a mean accuracy on this task of 98.8%.

Unlike the one-dimensional case we cannot obtain the general distribution of outputs across the entire parameter space. Instead we consider four classes of inputs to the NN ensemble: samples from the training set, the test set, random inputs where each pixel is uniformly and independently sampled between $[0, 1]$, and a set of adversarial examples. To construct the adversarial dataset we iterate over the test set, train a NN on the restricted MNIST task, and then construct an adversarial example using the fast gradient sign method (Goodfellow et al., 2014) with $\epsilon = 0.1$. We fix the size of these four classes of inputs to equal the size of the test set (1991 examples).

For each example from each input class we measure the Edgeworth expansion parameters $c_k$; the sample skewness, $\kappa_3$; and kurtosis, $\kappa_4$. We plot the distribution of the absolute value of these

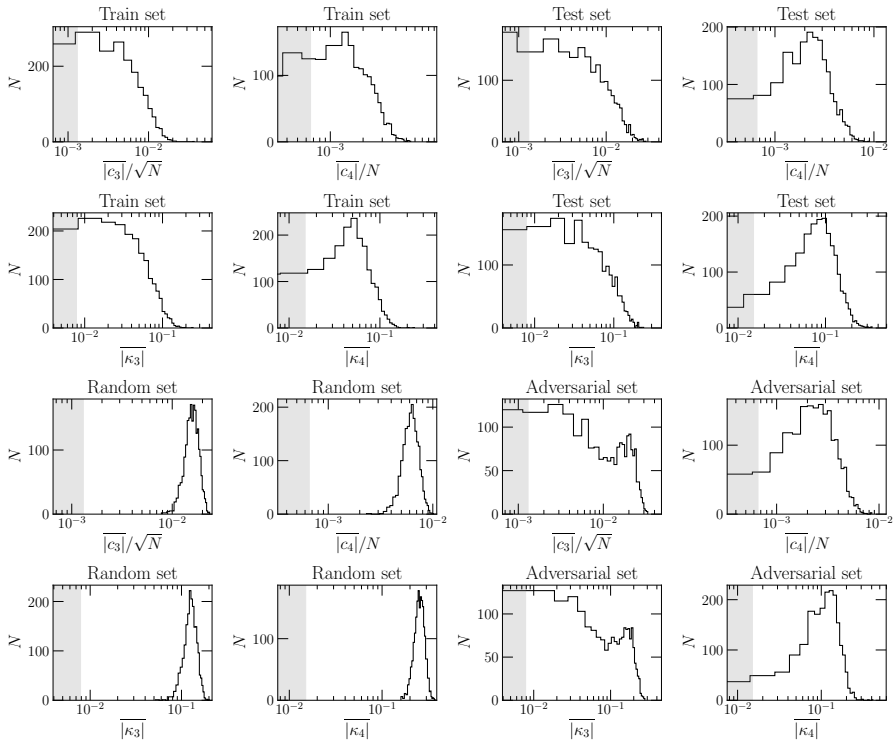


Figure 3: The distribution of measured deviations from Gaussianity for an ensemble of 60,000 NNs trained for 10,000 steps on a restricted MNIST task. For each example from the various input classes we measure the parameters $c_k$ and $\kappa_k$ across the distribution of outputs of the ensemble. The shaded region at the left of each plot is the non-Gaussianity that would be expected from a distribution of 60,000 samples from a Gaussian. Bins near this shaded region are therefore not statistically significant, but bins far to the right are highly statistically significant.

parameters after 10,000 steps of training in Fig. 3. While the measured asymmetry parameters $c_3$ and $\kappa_3$ are generally fairly close to being statistically insignificant, the ensemble exhibits statistically significant values for $\kappa_4$ and $c_4$ for all input classes except perhaps $c_4$ on the training set. Intriguingly the output distribution is furthest from Gaussian for random inputs.

To evaluate how the shape of the output distribution changes over the course of training, we next take the mean of the absolute value of the Edgeworth expansion parameters, sample skewness, and sample kurtosis across all the examples in each dataset class. We perform this procedure at initialization and after training for $10^k$ steps for $k \in \{0, \ldots, 4\}$. We plot the results in Fig. 4.

Consistent with the one-dimensional regression task we find that the distribution becomes more non-Gaussian in the early phases of training (steps 0–10), then becomes more Gaussian from steps 10–100, before gradually becoming less Gaussian by the end of training. The four statistics we measure the exhibit same relative ordering from Gaussian to non-Gaussian across the different input classes at the end of training: the training set is most Gaussian, followed by the test set, then the adversarial set, with the random set being the least Gaussian.

Lastly we examine the correlation between the Edgeworth and shape parameters with the log of the mean squared error loss for every example in the test set after 10 steps of training (when the distribution is least Gaussian). We present the results in Fig. 5. The strongest correlation is between $|c_3|$ and the mean loss, which has a Spearman rank correlation coefficient of $-0.88$ and his highly statistically significant. The parameters $\kappa_3$ and $\kappa_4$ are also negatively correlated with the mean loss. The parameter $|c_4|$ exhibits a small positive correlation with the mean loss (Spearman rank correlation coefficient of 0.078 with a $p \approx 5 \times 10^{-4}$). These results indicate that early in training the output distribution is less Gaussian in regimes of input space that have achieved low loss. We also calculate these correlations at the end of training and plot the results in Fig. 8 in the supplementary material.

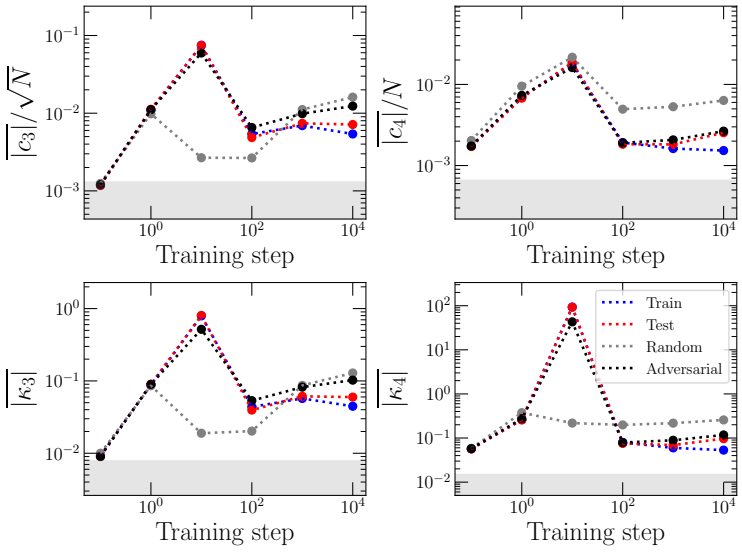


Figure 4: The measured deviations from Gaussianity at various stages during training on a restricted MNIST task. For each example in each input class we calculate the distribution of outputs from the NN ensemble. We then fit this statistic to the ensemble and take the mean of these statistics across every example in the input class. We plot the parameters at initialization at step 0.1. The shaded region at the bottom represents the non-Gaussianity that would be expected from our sample size.

By the end of training the inverse correlation between the Edgeworth and shape parameters with the mean loss has almost entirely disappeared.

## 5 Discussion

In both tasks we study we observe a similar behavior in the shape of the distribution of the ensemble over the course of training. At initialization the deviation from Gaussianity is small in the case of $c_4$ and $\kappa_4$ and statistically equivalent to 0 in the case of $c_3$ and $\kappa_3$. Yet after taking 10 steps of SGD the distribution is much *less* Gaussian than when it began. What drives the distribution away from Gaussianity? This is a difficult question and we cannot answer it in this paper. Nevertheless we can point towards some consequences of this observation. Let us consider the distribution of outputs after a single step of gradient descent. We wrote down the equation of the NN at initialization in Eq. 1. Since biases are commonly initialized with zeros we will ignore the $b_i$ term in the following discussion. The mean squared loss is given by

$$\mathcal{L} = \sum_i \left(f(\mathcal{X}_i) - \mathcal{Y}_i\right)^2, \tag{5}$$

where $\mathcal{X}$ is the training set and $\mathcal{Y}$ is the set of training labels. The gradients with respect to $u_i$ and $v_i$ are

$$\frac{\partial \mathcal{L}}{\partial u_i} = 2\sum_i \left(f(\mathcal{X}_i - \mathcal{Y}_i\right) h'(u_i x) x v_i, \tag{6}$$

$$\frac{\partial \mathcal{L}}{\partial v_i} = 2\sum_i \left(f(\mathcal{X}_i - \mathcal{Y}_i\right) h(u_i x). \tag{7}$$

This implies that the NN after a single step of gradient descent can be written

$$f_1(x) = \sum_i h[(u_i + 2\eta \sum_j (f_0(\mathcal{X}_j) - \mathcal{Y}_j) h'(u_i x) x v_i) x](v_i + 2\eta \sum_j \left(f_0(\mathcal{X}_j) - \mathcal{Y}_j\right) h(u_i x)), \tag{8}$$

where $\eta$ is the learning rate. If we assume that $\eta$ is small, we can linearize the activation function by taking the first term of its Taylor series and drop terms of order $\eta^2$:

$$f_1(x) \simeq \sum_i h(u_i x) v_i + 2\eta \sum_{i,j} \left(f_0(\mathcal{X}_j - \mathcal{Y}_j\right) \left[h'(u_i x)^2 x^2 v_i + h(u_i x)^2\right]. \tag{9}$$

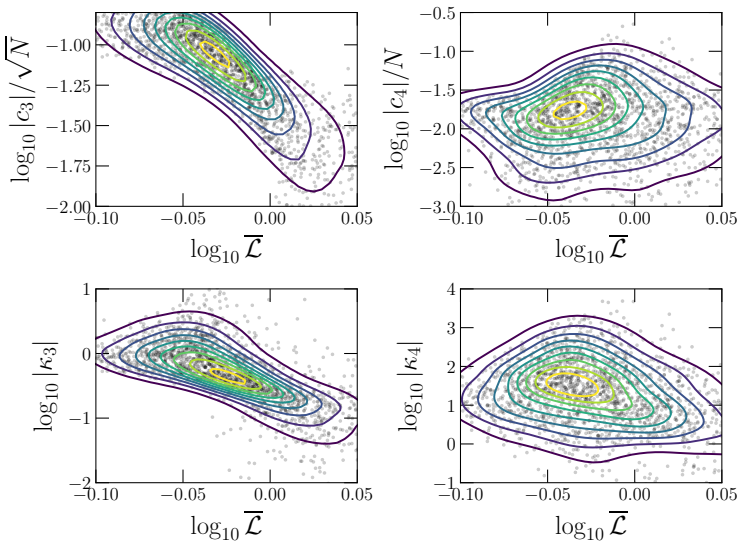


Figure 5: The correlation between the Edgeworth and shape parameters with the log of the mean squared error loss for every example in the test set after 10 steps of training. Contour lines indicate constant number density. The output distribution is less Gaussian for examples from the test set with lower loss.

The first term is, of course, simply $f_0(x)$, the NN at initialization. Because each term of the sum is independent its distribution will tend towards a Gaussian. The second term is more complicated, however. Although this, too, appears to be a sum of independent variates, we must recall that $f_0$ contains a sum over $h(u_k x)v_k$. The product of nonlinear functions of the random variates with each other induces dependencies among all the terms of the sum. This then breaks the independence assumption of the central limit theorem and so this term is no longer guaranteed to have a Gaussian distribution as its limit. Note that this is not inconsistent with the finding of Jacot et al. (2018) that a randomly initialized NN of infinite width will remain Gaussian under the influence of gradient descent. In the infinite width limit, the $f_0(\mathcal{X})$ term can be considered to be an independent normal variate in its own right since each individual $u_i$ and $v_i$ contributes only negligibly. It is only in the case of a NN of finite width that it is possible for the distribution to drift away from Gaussianity.

Despite these increases in the deviations from Gaussianity it is important to note that the distribution always remains close enough to Gaussian that the Edgeworth expansion provides an excellent approximation. We do not, for example, observe any distributions becoming bimodal over the course of training these NN ensembles.

## 6 Summary

We find that the output distribution of randomly initialized finite NNs is not Gaussian, but deviates from a Gaussian distribution consistent with the Edgeworth expansion. For a one-dimensional regression task and a restricted MNIST task the output distribution generally becomes *less* Gaussian under the influence of SGD (as measured by two Edgeworth parameters and two shape parameters). At the end of training on restricted MNIST the output distribution is most Gaussian for inputs from the training set, followed by the test set. Adversarial examples exhibit a less Gaussian distribution and random inputs are the least Gaussian of all. Early in training examples from the test set with a lower mean loss exhibit a less Gaussian output distribution than examples with a higher mean loss. This correlation is strongly reduced, but does not vanish entirely, by the end of training.

## References

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., et al. Tensorflow: A system for large-scale machine learning. In *12th {USENIX}*

*Symposium on Operating Systems Design and Implementation ({OSDI} 16)*, pp. 265–283, 2016.

Arora, S., Du, S. S., Hu, W., Li, Z., Salakhutdinov, R., and Ruosong, W. On exact computation with an infinitely wide neural net. *arXiv preprint arXiv:1904:11955*, 2019.

Blinnikov, S. and Moessner, R. Expansions for nearly Gaussian distributions. *Astronomy & Astrophysics Supplement Series*, 130:193–205, May 1998. doi: 10.1051/aas:1998221.

Cho, Y. and Saul, L. K. Kernel methods for deep learning. In *Advances in neural information processing systems*, pp. 342–350, 2009.

Darlington, R. B. Is kurtosis really "peakedness?". *The American Statistician*, 24(2):19–22, 1970.

Edgeworth, F. Y. The Law of Error. Part I. *Transactions of the Cambridge Philosophical Society*, 20: 36, 1908.

Esseen, C.-G. Fourier analysis of distribution functions. a mathematical study of the laplace-gaussian law. *Acta Mathematica*, 77(1):1–125, 1945.

Feller, W. *An introduction to probability theory and its applications*, volume 2. John Wiley & Sons, 1966.

Garriga-Alonso, A., Aitchison, L., and Rasmussen, C. E. Deep convolutional networks as shallow gaussian processes. *arXiv preprint arXiv:1808.05587*, 2018.

Glorot, X. and Bengio, Y. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pp. 249–256, 2010.

Goodfellow, I. J., Shlens, J., and Szegedy, C. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

Han, S., Pool, J., Tran, J., and Dally, W. Learning both weights and connections for efficient neural network. In *Advances in neural information processing systems*, pp. 1135–1143, 2015.

Hansen, E. Edgeworth expansions, 2006. URL `http://web.math.ku.dk/~erhansen/bootstrap_05/doku/noter/Edgeworth_24_01.pdf`.

Hinton, G., Vinyals, O., and Dean, J. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Howard, A., Sandler, M., Chu, G., Chen, L.-C., Chen, B., Tan, M., Wang, W., Zhu, Y., Pang, R., Vasudevan, V., et al. Searching for mobilenetv3. *arXiv preprint arXiv:1905.02244*, 2019.

Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., and Adam, H. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

Jacot, A., Gabriel, F., and Hongler, C. Neural tangent kernel: Convergence and generalization in neural networks. In *Advances in neural information processing systems*, pp. 8571–8580, 2018.

L'Ecuyer, P. and Simard, R. Testu01: Ac library for empirical testing of random number generators. *ACM Transactions on Mathematical Software (TOMS)*, 33(4):22, 2007.

Lee, J., Bahri, Y., Novak, R., Schoenholz, S. S., Pennington, J., and Sohl-Dickstein, J. Deep neural networks as gaussian processes. *arXiv preprint arXiv:1711.00165*, 2017.

Lee, J., Xiao, L., Schoenholz, S. S., Bahri, Y., Sohl-Dickstein, J., and Pennington, J. Wide neural networks of any depth evolve as linear models under gradient descent. *arXiv preprint arXiv:1902.06720*, 2019.

Matsumoto, M. and Nishimura, T. Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator. *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, 8(1):3–30, 1998.

Matthews, A. G. d. G., Rowland, M., Hron, J., Turner, R. E., and Ghahramani, Z. Gaussian process behaviour in wide deep neural networks. *arXiv preprint arXiv:1804.11271*, 2018.

Neal, R. M. Priors for infinite networks. In *Bayesian Learning for Neural Networks*, pp. 29–53. Springer, 1996.

Novak, R., Xiao, L., Bahri, Y., Lee, J., Yang, G., Abolafia, D. A., Pennington, J., and Sohl-dickstein, J. Bayesian deep convolutional networks with many channels are gaussian processes. In *International Conference on Learning Representations*, 2019. URL `https://openreview.net/forum?id=B1g30j0qF7`.

Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., and Lerer, A. Automatic differentiation in pytorch. In *NIPS 2017 Autodiff Workshop*, 2017. URL `https://openreview.net/forum?id=BJJsrmfCZ`.

Petrov, V. V. *Summy nezavisimykh sluchainykh velichin*. Nauka, 1972.

Salmon, J. K., Moraes, M. A., Dror, R. O., and Shaw, D. E. Parallel random numbers: as easy as 1, 2, 3. In *Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis*, pp. 16. ACM, 2011.

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., and Chen, L.-C. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4510–4520, 2018.

Saxe, A. M., McClelland, J. L., and Ganguli, S. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. *arXiv preprint arXiv:1312.6120*, 2013.

Sellentin, E., Jaffe, A. H., and Heavens, A. F. On the use of the Edgeworth expansion in cosmology I: how to foresee and evade its pitfalls. *arXiv e-prints*, September 2017.

Skovgaard, I. M. On multivariate edgeworth expansions. *International Statistical Review/Revue Internationale de Statistique*, pp. 169–186, 1986.

Westfall, P. H. Kurtosis as peakedness, 1905–2014. rip. *The American Statistician*, 68(3):191–195, 2014.

Williams, C. K. Computing with infinite networks. In *Advances in neural information processing systems*, pp. 295–301, 1997.

Yang, G. Scaling limits of wide neural networks with weight sharing: Gaussian process behavior, gradient independence, and neural tangent kernel derivation. *arXiv preprint arXiv:1902.04760*, 2019.

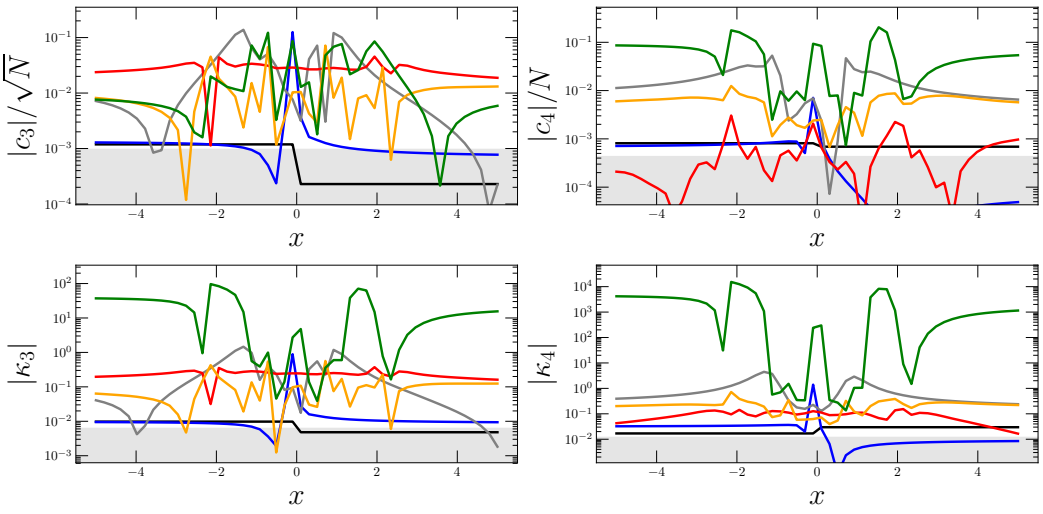


Figure 6: The best fit for the Edgeworth parameters and the shape parameters across the input range for the ensemble of NNs trained to fit the function $e^{-x^2}$. We show these parameters across the range $[-5, 5]$ although the NNs were trained on the range $[-3, 3]$. The NNs are generally more non-Gaussian in the extrapolation regime, but the most non-Gaussian distribution appears in the interpolation regime.

## Supplementary material

We show in Fig. 6 the absolute value of the Edgeworth parameters and the shape parameters, $\kappa_3$ and $\kappa_4$ across the input space of the ensemble of NNs trained to fit the function $e^{-x^2}$. Although we train the ensemble of NNs only on the range $[-3, 3]$ we measure the shape of its output distribution on the range $[-5, 5]$ to observe its behavior in regions where it must extrapolate. We observe that the distributions are generally more non-Gaussian in the regime where the NNs extrapolate than over the regime of interpolation, but this regime does not exhibit the most non-Gaussian distribution across the input space. Fig. 6 demonstrates that there can be substantial variation in the shape of the output distribution across the input space. This is perhaps a consequence of the fact that certain regions of parameter space are harder for the NN to fit than others and the difficulty of a particular region of input space appears to affect the shape of the output distribution (see Fig. 5).

We take the mean of the absolute values of the Edgeworth parameters and the shape parameters across the range of the input space $[-5, 5]$ at various points during training in Fig. 7. Similar to the restricted MNIST task, we observe that the distribution becomes less Gaussian over the course of training, with step 10 also being a particularly non-Gaussian stage of training.

### Further correlations between the Edgeworth and shape parameters with the loss

We show in Fig. 8 the relationship between the Edgeworth and shape parameters with the log of the mean squared error loss on the test set after 10,000 steps of training on the restricted MNIST task. By the end of training the correlation between these parameters with the loss has almost entirely disappeared, although all parameters retain a small, but statistically significant, negative correlation.

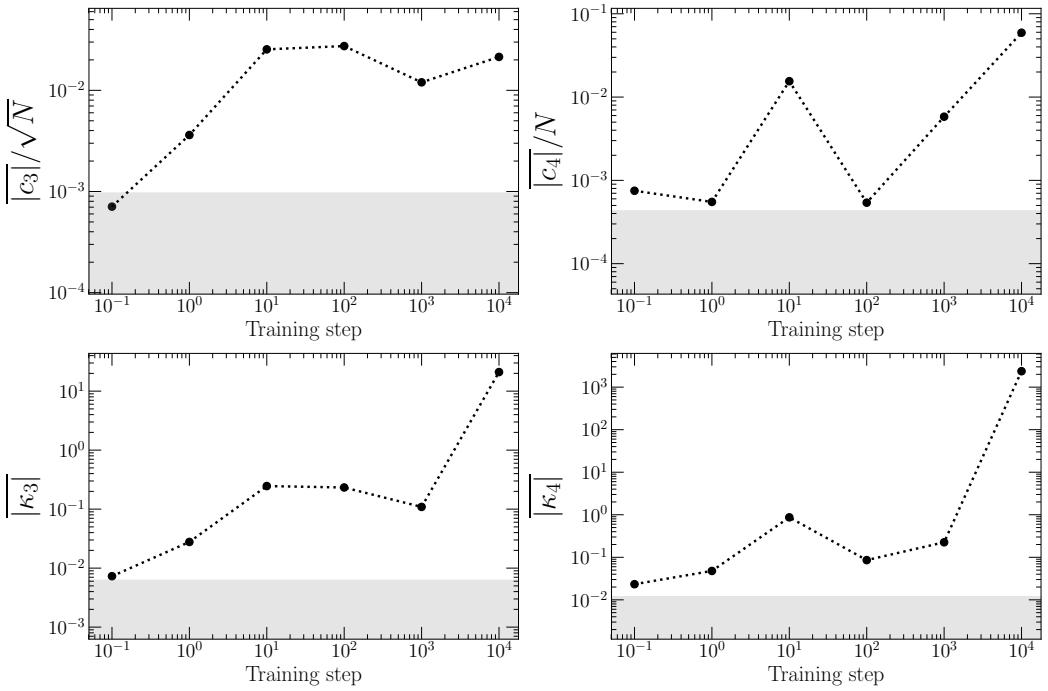


Figure 7: The mean of the absolute value of the Edgeworth parameters and shape parameters across the range $[-5, 5]$ at various stages of training. The shaded region at the left of each plot is the non-Gaussianity that would be expected from a distribution of the same size as our NN ensemble. Points near the shaded region therefore do not deviate from Gaussianity at a statistically significant level. We generally observe that the output distribution becomes more non-Gaussian over the course of training.

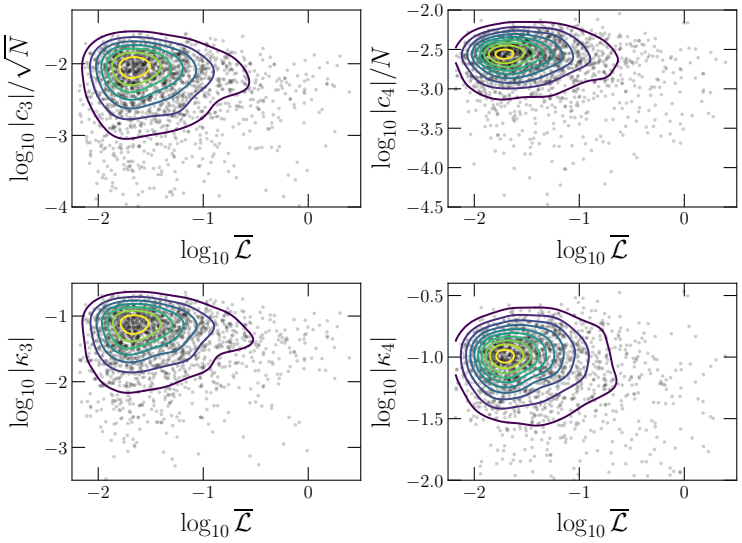


Figure 8: The relationship between the Edgeworth and shape parameters with the log of the mean squared error loss on the test set after 10,000 steps of training. Contours represent lines of constant number density.