CRITICAL INITIALISATION IN CONTINUOUS APPROXIMATIONS OF BINARY NEURAL NETWORKS

Anonymous authors
Paper under double-blind review

ABSTRACT

The training of stochastic neural network models with binary (± 1) weights and activations via continuous surrogate networks is investigated. We derive, using mean field theory, a set of scalar equations describing how input signals propagate through surrogate networks. The equations reveal that depending on the choice of surrogate model, the networks may or may not exhibit an order to chaos transition, and the presence of depth scales that limit the maximum trainable depth. Specifically, in solving the equations for edge of chaos conditions, we show that surrogates derived using the Gaussian local reparameterisation trick have no critical initialisation, whereas a deterministic surrogates based on analytic Gaussian integration do. The theory is applied to a range of binary neuron and weight design choices, such as different neuron noise models, allowing the categorisation of algorithms in terms of their behaviour at initialisation. Moreover, we predict theoretically and confirm numerically, that common weight initialization schemes used in standard continuous networks, when applied to the mean values of the stochastic binary weights, yield poor training performance. This study shows that, contrary to common intuition, the means of the stochastic binary weights should be initialised close to close to ± 1 for deeper networks to be trainable.

1 INTRODUCTION

Recent work in deep learning has used a mean field formalism to explain the empirically well known impact of initialization on the dynamics of learning Saxe et al. (2013), Poole et al. (2016), Schoenholz et al. (2016). From one perspective Poole et al. (2016), Schoenholz et al. (2016), the formalism studies how signals propagate forward and backward in wide, random neural networks, by measuring how the variance and correlation of input signals evolve from layer to layer, knowing the distributions of the weights and biases of the network. By studying these moments the authors in Schoenholz et al. (2016) were able to explain how heuristic initialization schemes avoid the "vanishing and exploding gradients problem" Glorot & Bengio (2010), establishing that for neural networks of arbirary depth to be trainable they must be initialised at "criticality", which corresponds to initial correlation being preserved to any depth. Practically, this line of work provides maximum trainable depth scales, as well as insight into how different initialization schemes will affect the speed of learning at the initial stages of training.

In this paper we extend this mean field formalism to two binary neural network approximations Soudry et al. (2014), Shayer et al. (2017), each of which acts as a smooth surrogate model suitable for the application of continuous optimization techniques. The problem of learning when the activations and weights of a neural network are of low precision has seen renewed interest in recent years, in part due to the promise of on-chip learning and the deployment of low-power applications Courbariaux & Bengio (2016). Recent work has opted to train discrete variable networks directly via backpropagation on a differentiable surrogate network, thus leveraging automatic differentiation libraries and GPUs. A key to this approach is in defining an appropriate surrogate network as an approximation to the discrete model, and various algorithms have been proposed Baldassi et al. (2018), Soudry et al. (2014), Courbariaux & Bengio (2016), Shayer et al. (2017).

Unfortunately, comparisons are difficult to make, since different algorithms may perform better under specific combinations of optimisation algorithms, initialisations, and heuristics such as drop out and batch normalization. Therefore a theoretical understanding of the various components of

the algorithms is desirable. To date, the initialisation of any binary neural network algorithm has not been studied, although the affect of quantization levels has been explored through this perspective Blumenfeld et al. (2019). Since all approximations still retain the basic neural network structure of layerwise processing, crucially applying backpropagation for optimisation, it is reasonable to expect that signal propagation will also be an important concept for these methods.

The two continuous surrogate models of binary networks that we study make use of the application of the central limit theorem (CLT) at the receptive fields of each neuron, assuming the binary weights are *stochastic*. Specifically, the fields are written in terms of the continuous means of stochastic binary weights, but with more complicated expressions than for standard continuous networks. The first approximation, presented in Soudry et al. (2014), and studied in the case of the perceptron in Baldassi et al. (2018), yields a deterministic surrogate via analytic integration. The ideas behind the approximation are old Spiegelhalter & Lauritzen (1990) but have seen renewed use in the current context from Bayesian Ribeiro & Opper (2011) Hernández-Lobato & Adams (2015) and non-Bayesian perspectives Soudry et al. (2014). The second approximation is based on the so called "local reparameterisation trick", which combines Monte Carlo sampling with the CLT to yield a differentiable network Shayer et al. (2017), Peters & Welling (2018). Note that the algorithm presented in Shayer et al. (2017) did not consider binary neurons, which we show here to severely limit this approach.

Our contribution is to successfully apply, in the spirit of Poole et al. (2016), a second level of mean field theory to analyse two surrogate models. The application of this mean field theory hinges on the use of self-averaging arguments Mezard et al. (1987). We demonstrate via simulation that the recursive equations derived for signal propagation accurately describe the behaviour of randomly initialised networks. Unlike standard continuous networks, it is not always the case that a binary neural network will have an edge of chaos (EOC). Therefore, for each surrogate, we attempt to solve the equations for this condition. As we will see, in the case that both neurons and weights are stochastic and binary (the most difficult case), we will see that an EOC exists for deterministic surrogate, while it does not exist for the reparameterisation trick based surrogate. We explore other choices or combinations of binary weights and neurons as well.

In the case that critical initialisations exist, we are also able to derive the depth scales that limit the maximum trainable depth, similarly to Schoenholz et al. (2016). These scales increase as the networks are initialised closer to criticality, similarly to standard neural networks. In the stochastic binary weight models, initialising close to criticality corresponds to the means of the weights being initialised with strongly broken symmetry, close to ± 1. Finally, we demonstrate experimentally that trainability is indeed delivered with this initialisation, making it possible to train deeper binary neural networks.

We also discuss the equivalent perspective to signal propagation, as first established in Saxe et al. (2013), that we are effectively studying how to control the singular value distribution of the input-output Jacobian matrix of the neural network Pennington et al. (2017) Pennington et al. (2018), specifically its mean. While for standard continuous neural networks the mean squared singular value of the Jacobian is directly related to the derivative of the correlation recursion equation, in the surrogates studied here this is not so. We show that in this case the derivative calculated is only an approximation of the Jacobian mean squared singular value, but that the approximation error approaches zero as the layer width goes to infinity. We consider the possibilities in pursuing this line of work, and other important questions, in the discussion.

2 BACKGROUND

2.1 CONTINUOUS NEURAL NETWORKS AND APPROXIMATIONS TO BINARY NETWORKS

A neural network model is typically defined as a deterministic non-linear function. We consider a fully connected feedforward model, which is composed of $N^\ell \times N^{\ell-1}$ weight matrices W^ℓ and bias vectors b^ℓ in each layer $\ell \in \{0, \dots, L\}$, with elements $W^\ell_{ij} \in \mathbb{R}$ and $b^\ell_i \in \mathbb{R}$. Given an input vector $x^0 \in \mathbb{R}^{N_0}$, the network is defined in terms of the following vector equations,

$$x^\ell = \phi^\ell(h^\ell_{\text{cts}}), \qquad h^\ell_{\text{cts}} = W^\ell x^{\ell-1} + b^\ell \tag{1}$$

where the pointwise non-linearity is, for example, $\phi^\ell(\cdot) = \tanh(\cdot)$. We refer to the input to a neuron, such as h^ℓ_{cts}, as the pre-activation field.

In the binary neural networks we study, we instead consider *stochastic* binary weight matrices and neurons. The idea is to leverage this stochasticity in deriving continuous surrogates. We denote the matrices as $\mathbf{S}^\ell$ with all weights[1] $\mathbf{S}^\ell_{ij} \in \{\pm 1\}$ being independently sampled Bernoulli variables: $\mathbf{S}^\ell_{ij} \sim Bernoulli(M^\ell_{ij})$, where the probability of flipping is controlled by the mean $M^\ell_{ij} := \mathbb{E}\mathbf{S}^\ell_{ij}$. The neurons in this model are also Bernoulli variables, controlled by the incoming field $\mathbf{h}^\ell_{\text{SB}} = \mathbf{S}^\ell \mathbf{x}^{\ell-1} + b^\ell$ (SB denoting "stochastic binary"). The idea behind several recent papers Soudry et al. (2014) Baldassi et al. (2018), Shayer et al. (2017), Peters & Welling (2018) is to adapt the mean of the Bernoulli weights, with the stochastic model essentially used to "smooth out" the discrete variables and arrive at a differentiable function, open to the application of continuous optimisation techniques.

The algorithms we study here take the limit of large layer width to model the field $\mathbf{h}^\ell_{\text{SB}}$ as a Gaussian, with mean $\bar{h}^\ell_i := \sum_j M^\ell_{ij} x^{\ell-1}_j + b^\ell_i$ and variance $\Sigma^\ell_{ii} = \sum_j 1 - (M^\ell_{ij} x^{\ell-1}_j)^2$. This is the first level of mean field theory, which we can apply successively from layer to layer by propagating means and variances to eventually obtain a differentiable function of the M^ℓ_{ij}.

Briefly, the deterministic surrogate of Soudry et al. (2014) and Baldassi et al. (2018) can be derived as follows. For a finite dataset $\mathcal{D} = \{x_\mu, y_\mu\}$, with y_μ the label, we define a cost via

$$\mathcal{L}_\mathcal{D}(f; M, b) = \sum_{\mu \in \mathcal{D}} \log \mathbb{E}_{\mathbf{S},\mathbf{x}} \big[p(y_\mu = f(x_\mu; \mathbf{S}, b, \mathbf{x})) \big] \tag{2}$$

with the expectation $\mathbb{E}_{\mathbf{S},\mathbf{x}}[\cdot]$ over all weights and neurons. This objective might also be recognised as a marginal likelihood, and so it is reasonable to describe this method as Type II maximum likelihood, or empirical Bayes. In any case, it is possible to take the expectation via approximate analytic integration, leaving us with a completely deterministic neural network with, for example, $\tanh(\cdot)$ non-linearities, but with more complicated pre-activation fields than a standard neural network.

The starting point for this approximation comes from rewriting the expectation $\mathbb{E}_{\mathbf{S},\mathbf{x}} \big[p(y_\mu = f(x_\mu; \mathbf{S}, b, \mathbf{x})) \big]$ in terms of nested conditional expectations, similarly to a Markov chain, with layers corresponding to time indices,

$$\begin{aligned}
\mathbb{E}_{\mathbf{S},\mathbf{x}}[p(y_\mu = f(x_\mu; \mathbf{S}, b, \mathbf{x}))] &= \sum_{\mathbf{S}^\ell, \mathbf{x}^\ell \; \forall \ell} p(y_\mu = f(x_\mu; \mathbf{S}, b, \mathbf{x})) p(\mathbf{x}^\ell | \mathbf{x}^{\ell-1}, \mathbf{S}^\ell) p(\mathbf{S}^\ell) \\
&= \sum_{\mathbf{S}^{L+1}} p(y_\mu = \mathbf{S}^{L+1}\mathbf{x}^L + b^L | \mathbf{x}^L) \prod_{\ell=0}^{L-1} \sum_{\mathbf{x}^\ell} \sum_{\mathbf{S}^\ell} p(\mathbf{x}^{\ell+1} | \mathbf{x}^\ell, \mathbf{S}^\ell) p(\mathbf{S}^\ell)
\end{aligned}$$

with the distribution of neurons factorising across the layer, given the previous layer, $p(\mathbf{x}^{\ell+1}|\mathbf{x}^\ell) = \prod_i p(\mathbf{x}^{\ell+1}_i | \mathbf{x}^\ell, \mathbf{S}^\ell_i)$. The basic idea is to successively marginalise over the stochastic inputs to each neuron, calculating an approximation of each neuron's probability distribution, $\hat{p}(\mathbf{x}^\ell_i)$. The approximation is based on the well known Gaussian integral of the Gaussian cumulative distribution function[2], see the appendices for details. The steps of the approximation can be written for illustration as,

$$\begin{aligned}
p(\mathbf{x}^\ell_i) = \sum_{\mathbf{x}^{\ell-1}} \sum_{\mathbf{S}^\ell} p(\mathbf{x}^\ell_i | \mathbf{x}^{\ell-1}, \mathbf{S}^\ell) p(\mathbf{S}^{\ell-1}) \hat{p}(\mathbf{x}^\ell) &\approx \int_{h^\ell_i} \sigma(\mathbf{h}^\ell_i \mathbf{x}^{\ell+1}_i) \mathcal{N}(\mathbf{h}^\ell_i | \bar{h}^\ell_i, (\Sigma^\ell_{MF})_{ii}) \\
&\approx \sigma(\kappa \frac{\bar{h}^\ell_i}{(1 + \Sigma^\ell_{MF})^{1/2}_{ii}} \mathbf{x}^\ell_i) := \hat{p}(\mathbf{x}^\ell_i)
\end{aligned} \tag{3}$$

[1] We follow the convention in physics models for 'spin' sites $\mathbf{S}^\ell_{ij} \in \{\pm 1\}$, and also denote a stochastic *binary* random variable with bold font.

[2] This is a slightly more general formulation than that in Soudry et al. (2014), which considered sign activations, but is otherwise equivalent. We note that the final algorithm derived in Soudry et al. (2014) did not backpropagate through the variance terms, whereas this was done properly in Baldassi et al. (2018) for binary networks, and earlier by Hernández-Lobato & Adams (2015) for Bayesian estimation of continuous neural networks.

with κ a constant of the integration, approximate or exact. The sigmoidal function $\sigma(\cdot)$ is typically cumulative distribution function of either the Gaussian, or the logistic distribution. We discuss this in more detail shortly, since it determines the form of the neuron non-linearity.

The term Σ_{MF} is the mean field approximation to the covariance between the stochastic binary pre-activations,

$$(\Sigma_{MF})_{ij} = Cov(\mathbf{h}^{\ell}_{\text{SB}}, \mathbf{h}^{\ell}_{\text{SB}})_{ij}\delta_{ij} \tag{4}$$

that is, a diagonal approximation to the full covariance (δ_{ij} is the Kronecker delta). This approximate probability is then used as part of the Gaussian CLT applied at the next layer. Importantly, we can write out the network forward equations analogously to the continuous case,

$$x^{\ell}_i = \phi^{\ell}(\kappa h^{\ell}), \quad h^{\ell} = \Sigma^{-\frac{1}{2}}_{MF}\bar{h}^{\ell}, \quad \bar{h}^{\ell} = M^{\ell}x^{\ell-1} + b^{\ell} \tag{5}$$

We note that the backpropagation algorithm derived in Soudry et al. (2014) was derived from a Bayesian message passing scheme, but removes all cavity arguments without corrections. As we have shown this algorithm is easier to derive from an empirical Bayes or maximum marginal likelihood formulation. Furthermore, in Soudry et al. (2014) the authors chose not to "backpropagate through" the variance terms, based on Taylor approximation and large layer width arguments.

The authors of Shayer et al. (2017), Peters & Welling (2018) utilise instead the local reparameterisation trick Kingma & Welling (2013) to obtain differentiable networks. The basic idea here is to rewrite the incoming field $\mathbf{h} \sim \mathcal{N}(\mu, \sigma^2)$ as $\mathbf{h} = \mu + \sigma\epsilon$ where $\epsilon \sim \mathcal{N}(0, 1)$. Thus any expectation over h can be written instead as an expectation over ϵ. The resulting networks are thus differentiable (with respect to the means and variances forming each Gaussian), albeit not deterministic. The forward propagation equations for this surrogate are

$$x^{\ell}_i = \phi^{\ell}(h^{\ell}), \quad h^{\ell} = \bar{h}^{\ell} + \epsilon^{\ell}\Sigma^{-\frac{1}{2}}_{MF}, \quad \bar{h}^{\ell} = M^{\ell}x^{\ell-1} + b^{\ell} \tag{6}$$

Given the either the approximately analytically integrated loss function, or the reparameterisation trick based surrogate, it is possible to perform gradient descent with respect to the M^{ℓ}_{ij} and b^{ℓ}_i.

In the next section move on to a second level of mean field, in order to study how a signal propagates on average in these continuous models, given *random* initialisation of the M^{ℓ} and b^{ℓ}. This is analogous to the approach of Poole et al. (2016) who studied random W^{ℓ} and b^{ℓ} in the standard continuous case. The motivation for considering this perspective is that, despite having a very different pre-activation fields, the surrogate models still maintain the same basic architecture, as seen clearly from the equations equation 31 and equation 6. Therefore, the surrogates are likely to inherit the same "training problems" of standard neural networks, such as the vanishing and exploding gradient problems Glorot & Bengio (2010). Since the dynamic mean field theory of Poole et al. (2016) provides a compelling explanation of the dynamics of the early stages of learning, via signal propagation, it is worthwhile to see if this theory can be extended to the non-standard network definitions.

2.1.1 A NOTE ON THE NON-LINEARITY $\phi()$ AND NEURON NOISE MODELS

The form of each neuron's probability distribution, $\sigma(\cdot)$ in Equation equation 3 depends on the underlying noise model. We can express a Bernoulli random variable $\mathbf{S} \in \{\pm 1\}$ with $\mathbf{S} \sim p(\mathbf{S}; \theta)$ via its latent variable formulation $\mathbf{S} = \text{sign}(\theta + \alpha\mathbf{L})$. In this form θ is referred to as a "natural" parameter, and the term $\mathbf{L}$ is a latent random noise, which determines the form of the probability distribution $\sigma(\cdot)$. In turn, this determines the form of the non-linearity since $\phi(\cdot) = 2\sigma(\cdot) - 1$. In general the form of $\phi(\cdot)$ will impact on the surrogates' performance, including within and beyond the mean field description presented here. However, a result following from the analysis in the next section is that choosing a deterministic binary neuron, ie. the $\text{sign}(\cdot)$ function, or a stochastic binary neuron, reduces to the same propagation equations, up to a scaling constant.

2.2 FORWARD SIGNAL PROPAGATION FOR STANDARD CONTINUOUS NETWORKS

We first recount the formalism developed in Poole et al. (2016). Assume the weights of a standard continuous network are initialised with $W^{\ell}_{ij} \sim \mathcal{N}(0, \sigma^2_w)$, biases $b^{\ell} \sim \mathcal{N}(0, \sigma^2_b)$, and input signal

x_a^0 has zero mean $\mathbb{E}x^0 = 0$ and variance $\mathbb{E}[x_a^0 \cdot x_a^0] = q_{aa}^0$, and with a denoting a particular input pattern. As before, the signal propagates via equation equation 1 from layer to layer.

The particular mean field approximation used here replaces each element in the pre-activation field h_i^ℓ by a Gaussian random variable whose moments are matched. So we are interested in computing, from layer to layer, the variance $q_{aa}^\ell = \frac{1}{N_\ell}\sum_i (h_{i;a}^\ell)^2$ from a particular input x_a^0, and also the covariance between the pre-activations $q_{ab}^\ell = \frac{1}{N_\ell}\sum_i h_{i;a}^\ell h_{i;b}^\ell$, arising from two different inputs x_a^0 and x_b^0 with given covariance q_{ab}^0. As explained in Poole et al. (2016), assuming the independence within a layer; $\mathbb{E}h_{i;a}^\ell h_{j;a}^\ell = q_{aa}^\ell \delta_{ij}$ and $\mathbb{E}h_{i;a}^\ell h_{j;b}^\ell = q_{ab}^\ell \delta_{ij}$, it is possible to derive recurrence relations from layer to layer

$$q_{aa}^\ell = \sigma_w^2 \int Dz \phi^2(\sqrt{q_{aa}^{\ell-1}} z) + \sigma_b^2 := \sigma_w^2 \mathbb{E}\phi^2(h_{j,a}^{\ell-1}) + \sigma_b^2 \tag{7}$$

with $Dz = \frac{dz}{\sqrt{2\pi}} e^{-\frac{z^2}{2}}$ the standard Gaussian measure. The recursion for the covariance is given by

$$q_{ab}^\ell = \sigma_w^2 \int Dz_1 Dz_2 \phi(u_a)\phi(u_b) + \sigma_b^2 := \sigma_w^2 \mathbb{E}\big[\phi(h_{j,a}^{\ell-1})\phi(h_{j,b}^{\ell-1})\big] + \sigma_b^2 \tag{8}$$

where

$$u_a = \sqrt{q_{aa}^{\ell-1}} z_1, \; u_b = \sqrt{q_{bb}^{\ell-1}}\big(c_{ab}^{\ell-1} z_1 + \sqrt{1-(c_{ab}^{\ell-1})^2} z_2\big)$$

and we identify c_{ab}^ℓ as the correlation in layer ℓ. Arguably the most important quantity is the the slope of the correlation recursion equation or mapping from layer to layer, denoted as χ, which is given by:

$$\chi = \frac{\partial c_{ab}^\ell}{\partial c_{ab}^{\ell-1}} = \sigma_w^2 \int Dz_1 Dz_2 \phi'(u_a)\phi'(u_b) \tag{9}$$

As discussed Poole et al. (2016), when $\chi_{c^*} = 1$ the system is at a critical point where correlations can propagate to arbitrary depth, corresponding to the edge of chaos. In continuous networks, χ is equivalent to the mean square singular value of the Jacobian matrix for a single layer $J_{ij} = \frac{\partial h_i^\ell}{\partial h_j^{\ell-1}}$, as explained in Poole et al. (2016). Therefore controlling χ will prevent the gradients from either vanishing or growing exponentially with depth.

In Schoenholz et al. (2016) explicit depth scales for standard neural networks are derived, which diverge corresponding when $\chi_{c^*} = 1$, thus providing the bounds on maximum trainable depth. We will not rewrite these continuous depth scales, since these resemble those in this case with which we now proceed.

3 THEORETICAL RESULTS FOR DETERMINISTIC SURROGATES

3.1 FORWARD SIGNAL PROPAGATION

For the deterinistic surrogate model we assume means initialised from some bounded distribution $M_{ij}^\ell \sim P(\mathcal{M} = M_{ij})$, with mean zero and variance of the means g?iven by σ_m^2. For instance, a valid distribution could be a clipped Gaussian[3], or another Bernoulli, for example $P(\mathcal{M}) = \frac{1}{2}\delta(\mathcal{M} = +\sigma_m) + \frac{1}{2}\delta(\mathcal{M} = -\sigma_m)$, whose variance is σ_m^2. The biases are distributed as $b^\ell \sim \mathcal{N}(0, N_{\ell-1}\sigma_b^2)$, with the variance scaled by the previous layer width $N^{\ell-1}$ since the denominator of the pre-activation scales with $N^{\ell-1}$ as seen from the definition equation 31. Once again we have input signal x_a^0, with zero mean $\mathbb{E}x^0 = 0$, and with a denoting a particular input pattern. Assume we have a binary neuron averaged appropriately, such that its mean $\bar{x}_i^\ell := \mathbb{E}_{p(x_i)} x_i^\ell = \phi(h_i^{\ell-1})$, where the field is given by:

$$h_i^\ell = \frac{\sum_j M_{ij}^\ell \phi(h_i^{\ell-1}) + b_i^\ell}{\sqrt{\sum_j [1 - (M_{ij}^\ell)^2 \phi^2(h_i^{\ell-1})]}} \tag{10}$$

[3]That is, sample from a Gaussian then pass the sample through a function bounded on the interval $[-1, 1]$.

which we can read from the vector equation equation 31. Note that this corresponds to the deterministic $\text{sign}(\cdot)$ neuron case. We actually show in Appendix B that the stochastic and deterministic binary neuron cases reduce to the same signal propagation equations.

As in the continuous case we are interested in computing the variance $q^\ell_{aa} = \frac{1}{N_\ell}\sum_i (h^\ell_{i;a})^2$ and covariance $\mathbb{E}h^\ell_{i;a}h^\ell_{j;b} = q^\ell_{ab}\delta_{ij}$, via recursive formulae. The key to the derivation is recognising that the denominator is a self-averaging quantity,

$$\lim_{N\to\infty}\frac{1}{N}\sum_j 1-(M^\ell_{ij})^2\phi^2(h^{\ell-1}_i) = 1-\mathbb{E}[(M^\ell_{ij})^2\phi^2(h^{\ell-1}_i)] = 1-\sigma^2_m\mathbb{E}\phi^2(h^{l-1}_{j,a}) \tag{11}$$

where we have used the properties that the M^ℓ_{ij} and $h^{\ell-1}_i$ are each i.i.d. random variables at initialisation, and independent Mezard et al. (1987). Following this self-averaging argument, we can take expectations more readily as shown in the appendices, finding the variance recursion

$$q^\ell_{aa} = \frac{\sigma^2_m\mathbb{E}\phi^2(h^{l-1}_{j,a})+\sigma^2_b}{1-\sigma^2_m\mathbb{E}\phi^2(h^{l-1}_{j,a})} \tag{12}$$

and then based on this expression for q^ℓ_{aa}, and assuming $q_{aa} = q_{bb}$, the correlation recursion can be written as

$$c^\ell_{ab} = \frac{1+q^\ell_{aa}}{q^\ell_{aa}}\frac{\sigma^2_m\mathbb{E}\phi(h^{l-1}_{j,a})\phi(h^{l-1}_{j,b})+\sigma^2_b}{1+\sigma^2_b} \tag{13}$$

The slope of the correlation mapping from layer to layer, when the normalized length of each input is at its fixed point $q^\ell_{aa} = q^\ell_{bb} = q^*(\sigma_m,\sigma_b)$, denoted as χ, is given by:

$$\chi = \frac{\partial c^\ell_{ab}}{\partial c^{\ell-1}_{ab}} = \frac{1+q^*}{1+\sigma^2_b}\sigma^2_m\int Dz_1Dz_2\phi'(u_a)\phi'(u_b) \tag{14}$$

where u_a and u_b are defined exactly as in the continuous case. Refer to the appendices for full details of the derivation.

3.2 Edge of chaos conditions

The edge of chaos in the hyper-parameter space (σ^2_b,σ^2_m), for the dynamical equations of the network, is determined as being the condition $\chi_1 = 1$, since this determines the stability of the correlation map fixed point $c^* = 1$. Note that for the deterministic surrogate this is always a fixed point. Following the straightforward arguments in Hayou et al. (2019) we take $\chi_1 = 1$ we can rearrange for σ^2_m,

$$\chi_1 = \frac{\sigma^2_m\mathbb{E}[\left(\phi'(\sqrt{q^*}z)\right)^2]}{1-\sigma^2_m\mathbb{E}[\phi^2(\sqrt{q^*}z)]} = 1 \implies \sigma^2_m = \frac{1}{\mathbb{E}[\left(\phi'(\sqrt{q^*}z)\right)^2]+\mathbb{E}[\phi^2(\sqrt{q^*}z)]} \tag{15}$$

We can then substitute this into the expression for the variance map,

$$q^\ell_{aa} = \sigma^2_b + (\sigma^2_b+1)\frac{\mathbb{E}\phi^2(h^{l-1}_{j,a})}{\mathbb{E}[\left(\phi'(\sqrt{q^*}z)\right)^2]} \tag{16}$$

Thus, in order to find the edge of chaos, as a function of the parameters σ^2_m and σ^2_b, one must simply find a value of σ^2_b which satisfies the variance map. We solve this numerically, as shown in Figure 4, for different neuron noise models and hence non-linearities $\phi(\cdot)$. We find that the edge of chaos for all these design choices is close to the point $(\sigma^2_m,\sigma^2_b) = (1,0)$. However, it is not just the singleton point, as for example in Hayou et al. (2019) for the ReLu case. We plot these edges of chaos in Appendix

3.3 ASYMPTOTIC EXPANSIONS AND DEPTH SCALES

In the continuous case, when χ approaches 1, we approach criticality and the rate of convergence to any fixed point slows. The depth scales, as derived in Schoenholz et al. (2016) provide a quantitative indicator to the number of layers correlations will survive for, and thus how trainable a network is. We show here that similar depth scales can be derived for these deterministic surrogates. According to Schoenholz et al. (2016) it should hold asymptotically that $|q^{\ell}_{aa} - q^*| \sim \exp(-\frac{\ell}{\xi_q})$ and $|c^{\ell}_{ab} - c^*| \sim \exp(-\frac{\ell}{\xi_c})$ for sufficiently large ℓ (the network depth), where ξ_q and ξ_c define the depth scales over which the variance and correlations of signals may propagate. Writing $q^{\ell}_{aa} = q^* + \epsilon^{\ell}$, we can show that:

$$\epsilon^{\ell+1} = \frac{\epsilon^{\ell}}{1+q^*}\big[\chi_1 + \frac{1+q^*}{1+\sigma_b^2}\sigma_w^2 \int Dz\phi''(\sqrt{q^*}z)\phi(\sqrt{q^*}z)\big] + \mathcal{O}((\epsilon^{\ell})^2) \tag{17}$$

We can similarly expand for the correlation $c^{\ell}_{ab} = c^* + \epsilon^{\ell}$, and if we assume $q^{\ell}_{aa} = q^*$, we can write

$$\epsilon^{\ell+1} = \epsilon^{\ell}\big[\frac{1+q^*}{1+\sigma_b^2}\sigma_m^2 \int Dz\phi'(u_1)\phi'(u_2)\big] + \mathcal{O}((\epsilon^{\ell})^2) \tag{18}$$

The depth scales we are interested in are given by the log ratio $\log \frac{\epsilon^{\ell+1}}{\epsilon^{\ell}}$. As discussed in Schoenholz et al. (2016), we are most interested in the correlation depth scale,

$$\xi_c^{-1} = -\log\big[\frac{1+q^*}{1+\sigma_b^2}\sigma_m^2 \int Dz\phi'(u_1)\phi'(u_2)\big] = -\log\chi \tag{19}$$

The arguments used in the original derivation Schoenholz et al. (2016) carry over to this case in a straightforward manner, albeit with more tedious algebra.

4 THEORETICAL RESULTS FOR REPARAMETERIZATION TRICK SURROGATES

4.1 FORWARD SIGNAL PROPAGATION

The pre-activation field for the perturbed surrogate with both stochastic binary weights and neurons is given by,

$$h^l_{i,a} = \frac{1}{\sqrt{N}}\sum_j M^l_{ij}\phi(h^{l-1}_{j,a}) + b^l_i + \epsilon^{\ell}_{i,a}\frac{1}{\sqrt{N}}\sqrt{\sum_j 1 - (M^l_{ij})^2\phi^2(h^{l-1}_{j,a})} \tag{20}$$

where we recall that $\epsilon \sim \mathcal{N}(0,1)$. The non-linearity $\phi(\cdot)$ can of course be derived from any valid binary Bernoulli neuron model. Appealing to the same self-averaging arguments used in the previous section, we find the variance map to be

$$q^{\ell}_{aa} = \mathbb{E}\left[(h^l_{i,a})^2\right] = \mathbb{E}\left[\left(\frac{1}{\sqrt{N}}\sum_j m^l_{ij}\phi(h^{l-1}_{j,a}) + b^l_i + \frac{1}{\sqrt{N}}\epsilon^{\ell}_{i,a}\sqrt{\sum_j 1 - (m^l_{ij})^2\phi^2(h^{l-1}_{j,a})}\right)^2\right] \tag{21}$$

$$= \sigma_m^2\mathbb{E}\phi^2(h^{l-1}_{j,a}) + \sigma_b^2 + (1 - \sigma_m^2\mathbb{E}\phi^2(h^{l-1}_{j,a})) = 1 + \sigma_b^2 \tag{22}$$

Interestingly, we see that the variance map does not depend on the variance of the means of the binary weights. This is a counter intuitive result, not immediately obvious from the pre-activation field definition. In the covariance map however, we do not have such simplification, since the perturbation $\epsilon_{i,a}$ in uncorrelated between inputs a and b, thus we recover Equation equation 8 similarly for the standard continuous case. Thus the correlation map is given by

$$c^l_{ab} = \frac{\sigma_m^2\mathbb{E}\phi(h^{l-1}_{j,a})\phi(h^{l-1}_{j,a}) + \sigma_b^2}{1+\sigma_b^2} \tag{23}$$

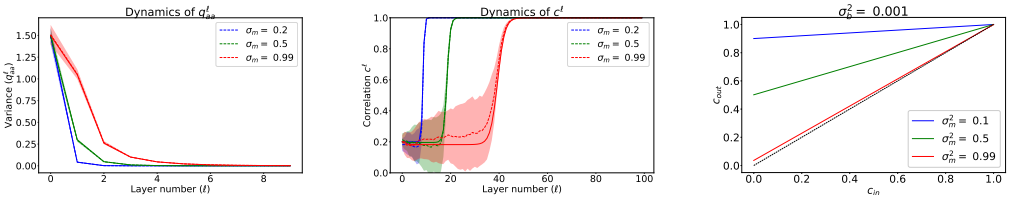

Figure 1: Dynamics of the variance and correlation maps, with simulations of a network of width $N = 1000$, 50 realisations, for various hyperparameter settings: $\sigma_m^2 \in \{0.2, 0.5, 0.99\}$ (blue, green and red respectively). (a) variance evolution, (b) correlation evolution. (c) correlation mapping (c_{in} to c_{out}), with $\sigma_b^2 = 0.001$

4.2 Edge of chaos conditions

For an edge of chaos to exist, we of course require that $c^* = 1$ to be a fixed point, as well as for the system to be marginally stable, $\chi_1 = 1$. Here we argue that these conditions cannot be met simultaneously. Specifically, from the correlation map we have a fixed point $c^* = 1$ if and only if

$$\sigma_m^2 = \frac{1}{\mathbb{E}[\phi^2(h_{j,a}^{l-1})]} \tag{24}$$

However, for any valid function $\phi(z)$, the expectation $\mathbb{E}[\phi^2(z)] \leq 1$. For example, consider $\phi(z) = \tanh(\kappa z)$ for any finite κ. This means that $c^* = 1$ can not be a fixed point, and thus there is no edge of chaos for this model. Of course, as $\kappa \to \infty$, and $\phi(z)$ becomes the $\text{sign}(z)$ function, $c* = 1$ is in fact always a fixed point, however the $\text{sign}(z)$ function does not have a derivative defined appropriately for a gradient descent procedure.

Likewise, since we have for χ the same expression as Equation equation 9, then considering the condition $\chi_1 = 1$, we find

$$\sigma_m^2 = \frac{1}{\mathbb{E}[(\phi'(h_{j,a}^{l-1}))^2]} \tag{25}$$

this expression cannot be satisfied unless $\phi(z)$, which is bounded between ± 1, has derivative identically equal to one (recall the preactivations are assumed to be zero mean Gaussian). Thus, neither condition can be met and there is no edge of chaos. In the appendices we include the case of continuous neurons and binary weights, where an edge does exist.

5 Numerical and experimental results

5.1 Simulations

We now move on to simulations of random networks, of the deterministic surrogate. In the appendices we present results for the reparameterisation trick based surrogate, but for the remainder of the paper we focus on the approximation which has an edge of chaos. We first verify that the theory accurately predicts the average behaviour of randomly initialised networks. In Figure 1 we see that the average behaviour of random networks are well predicted by the mean field theory. The estimates of the variance and correlation from simulations of random neural networks provided some input signals are plotted. The dotted lines correspond to empirical means, the shaded area corresponds to one standard deviation, and solid lines are the theoretical prediction. We see strong agreement in both the variance and correlation plots. In Appendix D we present the variance and correlation depth scales as a function of σ_m, and different curves corresponding to different bias variance values σ_b.

5.2 Training performance for different mean initialisation σ_m^2

Here we test experimentally the predictions of the mean field theory by training networks to overfit a dataset in the supervised learning setting, having arbitrary depth and different initialisations. We consider first the performance of the deterministic *surrogate*, not its corresponding binary network.

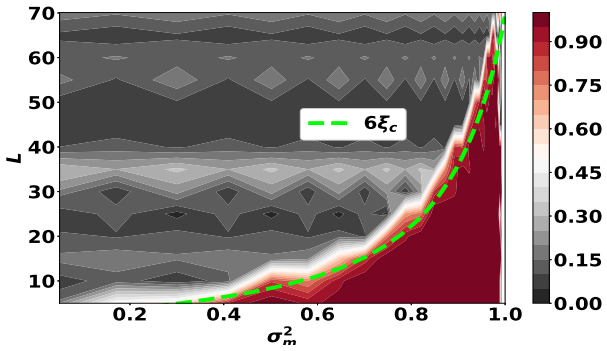

Figure 2: Training performance of the continuous surrogate network, for different depth (in steps of 5 layers, up to $L = 70$), against the variance of the means σ_m^2. Overlaid is a curve proportional to the correlation depth scale.

We use the MNIST dataset with reduced training set size (25%) and record the training performance (percentage of the training set correctly labeled) after 20 epochs of gradient descent over the training set, for various network depths $L < 70$ and different mean variances $\sigma_m^2 \in [0, 1)$. The optimizer used was Adam Kingma & Ba (2014) with learning rate of 2×10^{-4} chosen after simple grid search, and a batch size of 64.

We see that the experimental results match the correlation depth scale derived, with a similar proportion to the standard continuous case of $6\xi_c$ being the maximum possible attenuation in signal strength before trainability becomes difficult, as described in Schoenholz et al. (2016).

The reason we see the trainability not diverging in Figure 2 is that training time increases with depth, on top of requiring smaller learning rates for deeper networks, as described in detail in Saxe et al. (2013). The experiment here used the same number of epochs regardless of depth, meaning shallower networks actually had an advantage over deeper networks.

We should note that this theory does not specify for how many steps of training the effects of the initialisation will persist, that is, for how long the network remains close to criticality. Therefore, the number of steps we trained the network for is an arbitrary choice, and thus the experiments validate the theory in a more qualitative than quantitative way. Results were similar for other optimizers, including SGD, SGD with momentum, and RMSprop. Note that these networks were trained without dropout, batchnorm or any other heuristics.

In Figure 3 we present the training performance for the deterministic surrogate and its counterpart binary networks, both deterministic and stochastic. Once again, we test our algorithms on the MNIST dataset and plot results after 5 epochs. We see that the performance of the stochastic network matches more closely the performance of the continuous surrogate, especially as the number of samples increases, from $N = 5$ to $N = 100$ samples.

We can report that the number of samples necessary to achieve better classification, at least for more shallow networks, appears to depends on the number of training epochs. In some way, this is a sensible relationship, since during the course of training we might expect the means of the weights to polarise, moving closer to the bounds ± 1. Likewise, from experience continuous with neural networks, the neurons, which initially have zero mean pre-activations, are expected to "saturate" during training, that is, they become either always "on" $(+1)$ or "off" (-1). A stochastic network being "closer" to deterministic would require fewer samples overall. We can again report that this phenomena was observed. In the discussion we elaborate on what further experiments and analysis may be required to understand this problem.

6 Discussion

In this paper we have theoretically studied, based on self-averaging arguments, binary neural network algorithms using dynamic mean field theory, following the analysis recently developed for

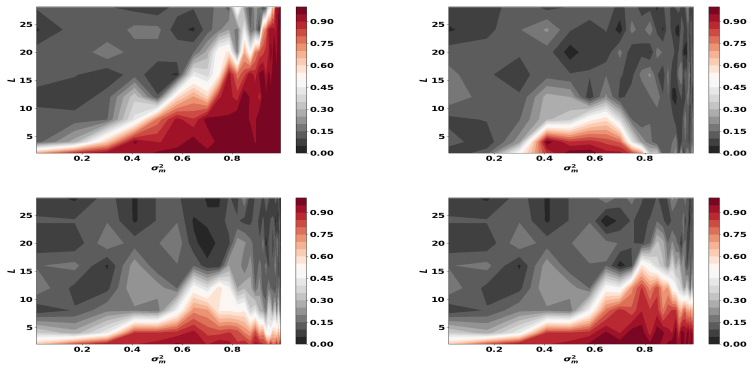

Figure 3: Training performance of the continuous surrogate and its binary counterparts after training on the MNIST dataset for 5 epochs. Top left: performance of the continuous surrogate. Top right: deterministic binary network. Bottom row: the performance of the stochastic binary network, averaged over 5 and 100 Monte Carlo samples (left and right, respectively).

standard continuous neural networks Schoenholz et al. (2016). This first study of a continuous surrogate networks has yielded results of practical significance, revealing that these networks have poor trainability when initialised away from ± 1, as is common practice.

One interesting problem this paper opens up is in understanding the relationship between the surrogate networks and the binary counterparts. Interesting results were uncovered for the binary neural networks corresponding to the trained surrogate, both binary and stochastic. It was seen that during training, when evaluating the deterministic and stochastic binary counterparts concurrently with the surrogate, the performance of both binary networks is worse than the continuous model, especially as depth increases. The stochastic binary network was seen to outperform the deterministic binary network, which makes sense since the objective optimised is the expectation over an ensemble of stochastic binary networks.

A study of random binary networks, included in the Appendices, and published recently Blumenfeld et al. (2019) for a different problem, showed that binary networks are always in a chaotic phase. However, when evaluating any binary network which is *trained* by some algorithm (eg. gradient descent on a given surrogate model), signals will of course propagate forwards through the corresponding binary network. This network will either be deterministic or stochastic. In either case, it makes sense that the closer one is to the early stages of the training process, the closer the signal propagation behaviour is to the randomly initialised case. Consider for a moment the signal propagation behaviour of a continuous network that has been trained, and this is not in its initially random state. This means that, as far as the mean field theory is concerned, the self-averaging behaviour, including any central limit behaviour, cannot be assumed to hold. However, clearly the networks are still performing some useful information processing, and thus are not in either the completely ordered case (asymptotic correlation $c^\infty = 1$) nor the chaotic case ($c^\infty = 0$). As said, it makes sense that the closer one is to the early stages of the training process, the closer the signal propagation behaviour will reflect the randomly initialised case. That is, correlations do not propagate, since there is no edge of chaos condition. However, it is possible that as training progresses the signal propagation behaviour binary counterparts of these surrogates might approach the signal propagation of the trained surrogate model. This may explain the difference in the performance between the surrogate model and its binary counterparts (deterministic or stochastic) early in training, a difference which appears to decrease as training progresses.

REFERENCES

Carlo Baldassi, Federica Gerace, Hilbert J. Kappen, Carlo Lucibello, Luca Saglietti, Enzo Tartaglione, and Riccardo Zecchina. Role of synaptic stochasticity in training low-precision neural networks. *Phys. Rev. Lett.*, 120:268103, Jun 2018. doi: 10.1103/PhysRevLett.120.268103. URL `https://link.aps.org/doi/10.1103/PhysRevLett.120.268103`.

Yaniv Blumenfeld, Dar Gilboa, and Daniel Soudry. A mean field theory of quantized deep networks: The quantization-depth trade-off, 2019.

Matthieu Courbariaux and Yoshua Bengio. Binarynet: Training deep neural networks with weights and activations constrained to +1 or -1. *CoRR*, abs/1602.02830, 2016. URL http://arxiv.org/abs/1602.02830.

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

Soufiane Hayou, Arnaud Doucet, and Judith Rousseau. On the impact of the activation function on deep neural networks training. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2672–2680, Long Beach, California, USA, 09–15 Jun 2019. PMLR. URL http://proceedings.mlr.press/v97/hayou19a.html.

José Miguel Hernández-Lobato and Ryan P. Adams. Probabilistic backpropagation for scalable learning of bayesian neural networks. In *Proceedings of the 32Nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, pp. 1861–1869. JMLR.org, 2015. URL http://dl.acm.org/citation.cfm?id=3045118.3045316.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014. URL http://arxiv.org/abs/1412.6980.

Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. *CoRR*, abs/1312.6114, 2013.

Marc Mezard, Giorgio Parisi, and Miguel Virasoro. *Spin Glass Theory and Beyond*, volume 9. 01 1987. doi: 10.1063/1.2811676.

Jeffrey Pennington, Samuel S. Schoenholz, and Surya Ganguli. Resurrecting the sigmoid in deep learning through dynamical isometry: theory and practice. *CoRR*, abs/1711.04735, 2017. URL http://arxiv.org/abs/1711.04735.

Jeffrey Pennington, Samuel Schoenholz, and Surya Ganguli. The emergence of spectral universality in deep networks. In Amos Storkey and Fernando Perez-Cruz (eds.), *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, volume 84 of *Proceedings of Machine Learning Research*, pp. 1924–1932, Playa Blanca, Lanzarote, Canary Islands, 09–11 Apr 2018. PMLR. URL http://proceedings.mlr.press/v84/pennington18a.html.

Jorn W. T. Peters and Max Welling. Probabilistic binary neural networks. *CoRR*, abs/1809.03368, 2018. URL http://arxiv.org/abs/1809.03368.

Ben Poole, Subhaneil Lahiri, Maithra Raghu, Jascha Sohl-Dickstein, and Surya Ganguli. Exponential expressivity in deep neural networks through transient chaos. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett (eds.), *Advances in Neural Information Processing Systems 29*, pp. 3360–3368. Curran Associates, Inc., 2016.

Fabiano Ribeiro and Manfred Opper. Expectation propagation with factorizing distributions: A gaussian approximation and performance results for simple models. *Neural Computation*, 23(4): 1047–1069, 2011. doi: 10.1162/NECO_a_00104. URL https://doi.org/10.1162/NECO_a_00104. PMID: 21222527.

Andrew M. Saxe, James L. McClelland, and Surya Ganguli. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. *CoRR*, abs/1312.6120, 2013. URL http://arxiv.org/abs/1312.6120.

Samuel S. Schoenholz, Justin Gilmer, Surya Ganguli, and Jascha Sohl-Dickstein. Deep information propagation. *CoRR*, abs/1611.01232, 2016. URL http://arxiv.org/abs/1611.01232.

Oran Shayer, Dan Levi, and Ethan Fetaya. Learning discrete weights using the local reparameterization trick. *CoRR*, abs/1710.07739, 2017. URL http://arxiv.org/abs/1710.07739.

Daniel Soudry, Itay Hubara, and Ron Meir. Expectation backpropagation: Parameter-free training of multilayer neural networks with continuous or discrete weights. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger (eds.), *Advances in Neural Information Processing Systems 27*, pp. 963–971. Curran Associates, Inc., 2014.

David J. Spiegelhalter and Steffen L. Lauritzen. Sequential updating of conditional probabilities on directed graphical structures. *Networks*, 20(5):579–605, 1990. doi: 10.1002/net.3230200507. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/net.3230200507.

A Derivation of deterministic surrogate networks

A.1 Integrating over stochastic or deterministic binary neurons

The form of each neuron's probability distribution depends on the underlying noise model. We can express a Bernoulli random variable $\mathbf{S} \in \{\pm 1\}$ with $\mathbf{S} \sim p(\mathbf{S}; \theta)$ via its latent variable formulation,

$$\mathbf{S} = \text{sign}(\theta + \alpha \mathbf{L}) \tag{26}$$

In this form θ is referred to as a "natural" parameter, from the statistics literature on exponential families. The term $\mathbf{L}$ is a latent random noise, which determines the form of the probability distribution. We also introduce a scaling α to control the variance of the noise, so that as $\alpha \to 0$ the neuron becomes a deterministic sign function. Letting $\alpha = 1$ for simplicity, we see that the probability of the Bernoulli variable taking a positive value is

$$p(\mathbf{S} = +1) = \int_{-\infty}^{-\theta} p(\mathbf{L}) d\mathbf{L} \tag{27}$$

where $p(\mathbf{L})$ is the known probability density function for the noise $\mathbf{L}$. The two common choices of noise models are Gaussian or logistic noise. The Gaussian of course has shifted and scaled $\text{erf}(\cdot)$ function as its cumulative distribution. The logistic random variable has the classic "sigmoid" or logistic function as its CDF, $\sigma(z) = \frac{1}{1+e^{-z}}$.

Thus, the probability of a the variable being positive is a function of the CDF. In the Gaussian case, this is $\Phi(\theta)$. By symmetry, the probability of $p(\mathbf{S} = -1) = \Phi(-\theta)$. Thus, we see the probability distribution for the Bernoulli random variable in general is the CDF of the noise $\mathbf{L}$, and we write $p(\mathbf{S}) = \Phi(\mathbf{S}\theta)$. In the logistic noise case, we have $p(\mathbf{S}) = \sigma(\mathbf{S}\theta)$

For the stochastic neurons, the natural parameter is the incoming field $\mathbf{h}_i^\ell = \sum_j \mathbf{S}_{i,j}^\ell \mathbf{x}_j^{\ell-1} + b_i^\ell$. Assuming this is approximately Gaussian in the large layer width limit, we can successively marginalise over the stochastic inputs to each neuron, calculating an approximation of each neuron's probability distribution, $\hat{p}(\mathbf{x}_i^\ell)$. This approximation is then used in the central limit theorem for the next layer, and so on.

For the case of neurons with latent Gaussian noise as part of the Bernoulli model, the integration over the pre-activation field (assumed to be Gaussian) is exact. Explicitly,

$$\begin{aligned} p(\mathbf{x}_i^\ell) &= \sum_{\mathbf{x}^{\ell-1}} \sum_{\mathbf{S}^\ell} p(\mathbf{x}_i^\ell | \mathbf{x}^{\ell-1}, \mathbf{S}^\ell) p(\mathbf{S}^{\ell-1}) \hat{p}(\mathbf{x}^\ell) \\ &\approx \int \Phi(\mathbf{x}_i^\ell \mathbf{h}_i^\ell) \mathcal{N}(\mathbf{h}_i^\ell | \bar{h}_i^\ell, (\Sigma_{MF}^\ell)_{ii}) \\ &= \Phi\left(\frac{\bar{h}_i^\ell}{\sqrt{1 + 2(\Sigma_{MF}^\ell)_{ii}}} \mathbf{x}_i^\ell \right) := \hat{p}(\mathbf{x}_i^\ell) \end{aligned} \tag{28}$$

where $\Phi(\cdot)$ is the CDF of the Gaussian distribution. We have again Σ_{MF} denoting the mean field approximation to the covariance between the stochastic binary pre-activations. The Gaussian expectation of the Gaussian CDF is a known identity, which we state in more generality in the next section, where we also consider neurons with logistic noise.

This new approximate probability distribution $\hat{p}(\mathbf{x}_i^\ell)$ can then used as part of the Gaussian CLT applied at the next layer, since it determines the means of the neurons in the next layer,

$$\mathbb{E}\mathbf{x}_i^\ell = 2\Phi\left(\frac{\bar{h}_i^\ell}{\sqrt{1+(\Sigma_{MF}^\ell)_{ii}}}\mathbf{x}_i^\ell\right) - 1 \tag{29}$$

If we follow these setps from layer to layer, we see that we are actually propagating approximate means for the neurons, combined non-linearly with the means of the weights. Given the approximately analytically integrated loss function, it is possible to perform gradient descent with respect to the means and biases, M_{ij}^ℓ and b_i^ℓ.

In the case of deterministic $\mathrm{sign}()$ neurons we obtain particularly simple expressions. In this case the "probability" of a neuron taking, for instance, positive is just Heaviside step function of the incoming field. Denoting the Heaviside with $\Theta(\cdot)$, we have

$$\begin{aligned} p(\mathbf{x}_i^\ell) &= \sum_{\mathbf{x}^{\ell-1}} \sum_{\mathbf{S}^\ell} p(\mathbf{x}_i^\ell|\mathbf{x}^{\ell-1}, \mathbf{S}^\ell) p(\mathbf{S}^{\ell-1}) \hat{p}(\mathbf{x}^{\ell-1}) \\ &\approx \int \Theta(\mathbf{x}_i^\ell \mathbf{h}_i^\ell) \mathcal{N}(\mathbf{h}_i^\ell|\bar{h}_i^\ell, (\Sigma_{MF}^\ell)_{ii}) \\ &\approx \Phi\left(\frac{\bar{h}_i^\ell}{(\Sigma_{MF}^\ell)_{ii}^{-\frac{1}{2}}}\mathbf{x}_i^\ell\right) := \hat{p}(\mathbf{x}_i^\ell) \end{aligned} \tag{30}$$

We can write out the network forward equations for the case of deterministic binary neurons, since it is a particularly elegant result. In general we have

$$\bar{x}_i^\ell = \phi(\eta h^\ell), \quad h^\ell = \Sigma_{MF}^{-\frac{1}{2}} \bar{h}^\ell, \quad \bar{h}^\ell = M^\ell x^{\ell-1} + b^\ell \tag{31}$$

where $\phi(\cdot) = \mathrm{erf}(\cdot)$ is the mean of the next layer of neurons, being a scaled and shifted version of the neuron's noise model CDF. The constant is $\eta = \frac{1}{\sqrt{2}}$, standard for the Gaussian CDF to error functin conversion.

A.2 Exact and approximate Gaussian integration of sigmoidal functions

We now present the integration of stochastic neurons with logistic as well as Gaussian noise as part of their latent variable models. The logistic case is an approximation built on the Gaussian case, motivated by approximating the logistic CDF with the Gaussian CDF. The reason we may be interested in using logistic CDFs, rather than just considering latent Gaussian noise models which integrate exactly, is not justified in any rigorous or experimental way. Any such analysis would likely consider the effect of the tails of the logistic versus the Gaussian distributions, where the logistic tails are much heavier than those of the Gaussian. One historic reason for considering the logistic function, we note, is the prevalence of logistic-type functions (such as $\tanh(\cdot)$) in the neural network literature. The computational cost of evaluating either logistic or error functions is similar, so there is no motivation from the efficiency side. Instead it seems a historic preference to have logistic type functions used with neural networks.

As we saw in the previous subsection, the integration over the analytic probability distribution for each neuron gave a function which allows us to calculate the means of the neurons in the next layer. Therefore, we directly calculate the expression for the means.

The Gaussian integral of the Gaussian CDF was used in the previous section to derive the exact probability distribution for the Bernoulli neuron in the next layer. The result is well known, and can be stated in generality as follows,

$$\int_{-\infty}^{\infty} \Phi(ay) \frac{e^{-\frac{(y-x)^2}{2\sigma^2}}}{\sqrt{2\pi\sigma^2}} dy = \Phi(\frac{x}{\sqrt{1+a^2\sigma^2}}) \tag{32}$$

We can integrate a logistic noise Bernoulli neuron using this result as well. The idea is to approximate the logistic noise with a suitably scaled Gaussian noise. However, since the overall network

approximation results in propagating means from layer to layer, we can equivalently need to approximate the $\tanh(\cdot)$ with the with the erf. Specifically, if we have $f(x;\alpha) = \tanh(\frac{x}{\alpha})$, an approximation is $g(x;\alpha) = \mathrm{erf}(\frac{\sqrt{\pi}}{2\alpha}x)$, by requiring equality of derivatives at the origin. In order to establish this, consider

$$f'(0;\alpha) = (1 - \tanh^2(0/\alpha)\frac{1}{\alpha} = \frac{1}{\alpha} \tag{33}$$

and

$$\frac{d\,\mathrm{erf}(x;\sigma)}{dx}|_{x=0} = \frac{2}{\sqrt{\pi\sigma^2}}e^{-x^2/\sigma^2}|_{x=0} = \frac{2}{\sqrt{\pi\sigma^2}} \tag{34}$$

Equating these, gives $\sigma^2 = \frac{4\alpha^2}{\pi}$, thus $\sigma = \frac{2\alpha}{\sqrt{\pi}}$.

The approximate integral over the Bernoulli neuron mean is then

$$\int_{-\infty}^{\infty} f(y;\alpha)\frac{e^{-\frac{(y-x)^2}{2\sigma^2}}}{\sqrt{2\pi\sigma^2}}dy \approx \int_{-\infty}^{\infty} \mathrm{erf}(\frac{\sqrt{\pi}}{2\alpha}y)\frac{e^{-\frac{(y-x)^2}{2\sigma^2}}}{\sqrt{2\pi\sigma^2}}dy \tag{35}$$

$$= \mathrm{erf}(\frac{\sqrt{\pi}}{2\alpha\gamma}x) \tag{36}$$

$$\text{with } \gamma = \sqrt{1 + \frac{\pi\sigma^2}{2\alpha^2}} \tag{37}$$

If we so desire, we can approximate this again with a $\tanh(\cdot)$ using the $\tanh(\cdot)$ to $\mathrm{erf}(\cdot)$ approximation in reverse. The scale parameter of this $\tanh(\cdot)$ will be $\alpha_2 = \frac{\pi}{4\alpha\gamma}$. If $\alpha = 1$ as is standard, then

$$\mathrm{erf}(\frac{\sqrt{\pi}}{2\gamma}x) \approx \tanh(\frac{\pi x}{4\gamma}) \tag{38}$$

B EQUIVALENCE OF DETERMINISTIC AND STOCHASTIC NEURONS FOR DETERMINISTIC SURROGATE

Assume a stochastic neuron with some latent noise, as per the previous appendix, with mean $\bar{x}_i^\ell := \mathbb{E}_{p(x_i)}x_i^\ell = \phi(h_i^{\ell-1})$. The field is given by

$$h_i^\ell = \frac{1}{\sqrt{2}}\frac{\sum_j M_{ij}^\ell \phi(h_i^{\ell-1}) + b_i^\ell}{\sqrt{1 + 2\sum_j[1 - (M_{ij}^\ell)^2\phi^2(h_i^{\ell-1})]}} \tag{39}$$

We see that the expression for the variance of the field simplifies as follows,

$$q_{aa}^\ell = \mathbb{E}(h_i^\ell)^2 = \frac{1}{2}\frac{\sum_j M_{ij}^\ell \phi(h_i^{\ell-1}) + b_i^\ell}{1 + 2\sum_j[1 - (M_{ij}^\ell)^2\phi^2(h_i^{\ell-1})]} \tag{40}$$

$$= \frac{1}{2}\frac{N(\sigma_m^2\mathbb{E}\phi^2(h_{j,a}^{l-1}) + \sigma_b^2)}{1 + 2(N - N\sigma_m^2\mathbb{E}\phi^2(h_{j,a}^{l-1}))} \tag{41}$$

$$= \frac{1}{2}\frac{\sigma_m^2\mathbb{E}\phi^2(h_{j,a}^{l-1}) + \sigma_b^2}{2(1 - \sigma_m^2\mathbb{E}\phi^2(h_{j,a}^{l-1}))} \tag{42}$$

By similar steps, we find that in the deterministic binary neuron case, we would obtain the same expression, albeit with a different scaling constant. This is easily seen by inspection of the field term in the deterministic neuron case,

$$h_i^\ell = \frac{1}{\sqrt{2}}\frac{\sum_j M_{ij}^\ell \phi(h_i^{\ell-1}) + b_i^\ell}{\sqrt{\sum_j[1 - (M_{ij}^\ell)^2\phi^2(h_i^{\ell-1})]}} \tag{43}$$

which again was derived in the previous appendix.

C DERIVATION OF SIGNAL PROPAGATION EQUATIONS IN DETERMINISTIC SURROGATE NETWORKS

Here we present the derivations for the signal propagation in the continuous network models studied in the paper.

C.1 VARIANCE PROPAGATION

We first calculate the variance given a signal:

$$q_{aa}^l = \frac{1}{N_l}\sum_i \left(h_{i,a}^l\right)^2 = E\left[\left(h_{i,a}^l\right)^2\right] \tag{44}$$

Where for us:

$$h_{i,a}^l = \frac{\sum_j m_{ij}^l \phi\left(h_{j,a}^{l-1}\right) + b_i^l}{\sqrt{\sum_j \left(1 - \left(m_{ij}^l\right)^2 \phi^2\left(h_{j,a}^{l-1}\right)\right)}} \tag{45}$$

and

$$m_{ij} \sim N\left(0, \sigma_m^2\right) b_i \sim N\left(0, N_{l-1}\sigma_b^2\right) \tag{46}$$

$$\begin{aligned}
\mathbb{E}\left[\left(h_{i,a}^l\right)^2\right] &= \mathbb{E}\left[\left(\frac{\sum_j m_{ij}^l \phi\left(h_{j,a}^{l-1}\right) + b_i^l}{\sqrt{\sum_j \left(1 - \left(m_{ij}^l\right)^2 \phi^2\left(h_{j,a}^{l-1}\right)\right)}}\right)^2\right] = \frac{\mathbb{E}\left[\left(\sum_j m_{ij}^l \phi\left(h_{j,a}^{l-1}\right) + b_i^l\right)^2\right]}{N_{l-1} - \sum_j \left(m_{ij}^l\right)^2 \phi^2\left(h_{j,a}^{l-1}\right)} \\
&= \frac{\sum_j \sigma_m^2 \mathbb{E}\phi^2\left(h_{j,a}^{l-1}\right) + N_{l-1}\sigma_b^2}{N_{l-1}\left(1 - \frac{1}{N_{l-1}}\sum_j \left(m_{ij}^l\right)^2 \phi^2\left(h_{j,a}^{l-1}\right)\right)} = \frac{N_{l-1}\sigma_m^2 \mathbb{E}\phi^2\left(h_{j,a}^{l-1}\right) + N_{l-1}\sigma_b^2}{N_{l-1}\left(1 - \sigma_m^2 \mathbb{E}\phi^2\left(h_{j,a}^{l-1}\right)\right)} \\
&= \frac{\sigma_m^2 \mathbb{E}\phi^2\left(h_{j,a}^{l-1}\right) + \sigma_b^2}{1 - \sigma_m^2 \mathbb{E}\phi^2\left(h_{j,a}^{l-1}\right)}
\end{aligned} \tag{47}$$

Where, $\mathbb{E}\phi^2\left(h_{j,a}^{l-1}\right)$ can be written explicitly, taking into account that $h_{j,a}^{l-1} \sim N\left(0, q_{aa}\right)$:

$$\begin{aligned}
\mathbb{E}\left[\phi^2\left(h_{j,a}^l\right)\right] = \int \mathcal{D}h_{j,a}^l \phi^2\left(h_{j,a}^l\right) &= \int dh_{j,a}^l \frac{1}{\sqrt{2\pi}\mathbb{E}\left[\left(h_{j,a}^l\right)^2\right]} \exp\left(-\frac{\left(h_{j,a}^l\right)^2}{2\mathbb{E}\left[\left(h_{j,a}^l\right)^2\right]}\right) \phi^2\left(h_{j,a}^l\right) \\
&= \int dh_{j,a}^l \frac{1}{\sqrt{2\pi q_{aa}^l}} \exp\left(-\frac{\left(h_{j,a}^l\right)^2}{2q_{aa}^l}\right) \phi^2\left(h_{j,a}^l\right)
\end{aligned} \tag{48}$$

We can now perform the following change of variable:

$$z_{j,a}^l = \frac{h_{j,a}^l}{\sqrt{q_{aa}^l}} \tag{49}$$

Then:

$$\begin{aligned}
\mathbb{E}\left[\phi^2\left(h_{j,a}^l\right)\right] &= \frac{1}{\sqrt{2\pi q_{aa}^l}} \sqrt{q_{aa}^l} \int dz_{j,a}^l \exp\left(-\frac{\left(z_{j,a}^l\right)^2}{2}\right) \phi^2\left(\sqrt{q_{aa}^l} z_{j,a}^l\right) \\
&= \frac{1}{\sqrt{2\pi}} \int dz \exp\left(-\frac{z^2}{2}\right) \phi^2\left(\sqrt{q_{aa}^l} z\right) \\
&= \int \mathcal{D}z \phi^2\left(\sqrt{q_{aa}^l} z\right)
\end{aligned} \tag{50}$$

$$q_{aa}^l = \mathbb{E}\left[\left(h_{i,a}^l\right)^2\right] = \frac{\sigma_m^2 \int \mathcal{D}z\phi^2\left(\sqrt{q_{aa}^{l-1}}z\right) + \sigma_b^2}{1 - \sigma_m^2 \int \mathcal{D}z\phi^2\left(\sqrt{q_{aa}^{l-1}}z\right)} \tag{51}$$

In the first layer, input neurons are not stochastic: they are samples drawn from the Gaussian distribution $x^0 \sim N\left(0, q^0\right)$:

C.1.1 CORRELATION PROPAGATION

To determine the correlation recursion we start from its definition:

$$c_{ab}^l = \frac{q_{a,b}^l}{\sqrt{q_{aa}^l q_{bb}^l}}, \tag{52}$$

where q_{ab}^l represents the covariance of the pre-activations $h_{i,a}^l$ and $h_{i,b}^l$, related to two distinct input signals and therefore defined as:

$$q_{ab}^l = \frac{1}{N_l}\sum_i h_{i,a}^l h_{i,b}^l = \mathbb{E}\left[h_{i,a}^l h_{i,b}^l\right]. \tag{53}$$

Replacing the pre-activations with their expressions provided in eq. (45) and taking advantage of the self-averaging argument, we can then write:

$$c_{ab}^l = \frac{\sigma_m^2 \mathbb{E}\left[\phi\left(h_{j,a}^{l-1}\right)\phi\left(h_{j,b}^{l-1}\right)\right] + \sigma_b^2}{\sqrt{q_{aa}^l\left(1 - \sigma_m^2 \mathbb{E}\left[\phi^2\left(h_{j,a}^{l-1}\right)\right]\right)}\sqrt{q_{bb}^l\left(1 - \sigma_m^2 \mathbb{E}\left[\phi^2\left(h_{j,b}^{l-1}\right)\right]\right)}}. \tag{54}$$

At this point, given that q_{aa}^l and q_{bb}^l quite quickly approach the fixed point, we can conveniently assume $q_{aa}^l = q_{bb}^l$. Moreover, exploiting eq.(51), we can finally write the expression for the correlation recursion:

$$c_{ab}^l = \frac{1 + q_{aa}^l}{q_{aa}^l}\frac{\sigma_m^2 \mathbb{E}\left[\phi\left(h_{j,a}^{l-1}\right)\phi\left(h_{j,b}^{l-1}\right)\right] + \sigma_b^2}{1 + \sigma_b^2}. \tag{55}$$

C.2 DERIVATION OF THE SLOPE OF THE CORRELATIONS AT THE FIXED POINT

To check the stability at the fixed point, we need to compute the slope of the correlations mapping from layer to layer at the fixed point:

$$\begin{aligned}
\chi|_{q_*} &= \frac{\partial c_{ab}^l}{\partial c_{ab}^{l-1}} \\
&= \frac{1 + q_*}{q_*}\frac{\sigma_m^2}{1 + \sigma_b^2}\frac{\partial}{\partial c_{ab}^{l-1}}\mathbb{E}\left[\phi\left(h_{j,a}^{l-1}\right)\phi\left(h_{j,b}^{l-1}\right)\right]|_{q_*} \quad , \\
&= \frac{1 + q_*}{q_*}\frac{\sigma_m^2}{1 + \sigma_b^2}\frac{\partial}{\partial c_{ab}^{l-1}}\int \mathcal{D}z_a\mathcal{D}z_b\phi\left(u_a\right)\phi\left(u_b\right)|_{q_*}
\end{aligned} \tag{56}$$

where we get rid of σ_b because independent from c_{ab}^{l-1}. Replacing the definition of u_a and u_b provided in the continuous model, we can explicitly compute the derivative with respect to c_{ab}^{l-1}:

$$\chi = \frac{1+q_*}{q_*}\frac{\sigma_m^2}{1+\sigma_b^2}\left(A-B\right), \tag{57}$$

where we have defined A and B as:

$$\begin{aligned} A &= \sqrt{q_*}\int \mathcal{D}z_a \mathcal{D}z_b \phi\left(\sqrt{q_{aa}^{l-1}}z_a\right)\phi'\left(\sqrt{q_{bb}^{l-1}}\left(c_{ab}^{l-1}z_a + \sqrt{1-\left(c_{ab}^{l-1}\right)^2}z_b\right)\right)z_a \\ B &= \sqrt{q_*}\int \mathcal{D}z_a \mathcal{D}z_b \phi\left(\sqrt{q_{aa}^{l-1}}z_a\right)\phi'\left(\sqrt{q_{bb}^{l-1}}\left(c_{ab}^{l-1}z_a + \sqrt{1-\left(c_{ab}^{l-1}\right)^2}z_b\right)\right)\frac{c_{ab}^{l-1}}{\sqrt{1-\left(c_{ab}^{l-1}\right)^2}}z_b. \end{aligned} \tag{58}$$

We can focus on B first. Integrating by parts over z_b we get:

$$B = \sqrt{q_*}\int \mathcal{D}z_a \mathcal{D}z_b \phi\left(\sqrt{q_{aa}^{l-1}}z_a\right)\frac{\partial}{\partial z_a}\phi'\left(\sqrt{q_{bb}^{l-1}}\left(c_{ab}^{l-1}z_a + \sqrt{1-\left(c_{ab}^{l-1}\right)^2}z_b\right)\right). \tag{59}$$

Then, integrating by parts over z_a, we the get:

$$\begin{aligned} B = \sqrt{q_*}\int \mathcal{D}z_a \mathcal{D}z_b \phi\left(\sqrt{q_{aa}^{l-1}}z_a\right)\phi'\left(\sqrt{q_{bb}^{l-1}}\left(c_{ab}^{l-1}z_a + \sqrt{1-\left(c_{ab}^{l-1}\right)^2}z_b\right)\right)z_a + \\ - q_*\int \mathcal{D}z_a \mathcal{D}z_b \phi'\left(\sqrt{q_{aa}^{l-1}}z_a\right)\phi'\left(\sqrt{q_{bb}^{l-1}}\left(c_{ab}^{l-1}z_a + \sqrt{1-\left(c_{ab}^{l-1}\right)^2}z_b\right)\right). \end{aligned} \tag{60}$$

Replacing A and B in eq. (57), we then obtain the closest expression for the stability at the variance fixed point, namely:

$$\chi|_{q_*} = \frac{1+q_*}{1+\sigma_b^2}\sigma_m^2\int \mathcal{D}z_a \mathcal{D}z_b \phi'\left(u_a\right)\phi'\left(u_b\right) \tag{61}$$

C.3 VARIANCE DEPTH SCALE

As pointed out in the main text, it should hold asymptotically that:

$$|q_{aa}^{l+1} - q_*| \sim \exp\left(-\frac{l+1}{\xi_q},\right) \tag{62}$$

with ξ_q defining the variance depth scale. To compute it we can expand over small perturbations around the fixed point, namely:

$$\begin{aligned} q_{aa}^{l+1} &= q_* + \epsilon^l \\ &= \frac{\sigma_m^2\int \mathcal{D}z\phi^2\left(\sqrt{q_*+\epsilon^l}z\right)+\sigma_b^2}{1-\sigma_m^2\int \mathcal{D}z\phi^2\left(\sqrt{q_*+\epsilon^l}z\right)}. \end{aligned} \tag{63}$$

Expanding the square root for small ϵ^l, we can then write:

$$q_{aa}^{l+1} \simeq \frac{\sigma_m^2 \int \mathcal{D}z \phi^2 \left(\sqrt{q_*}z + \frac{\epsilon^l}{2\sqrt{q_*}}z\right) + \sigma_b^2}{1 - \sigma_m^2 \int \mathcal{D}z \phi^2 \left(\sqrt{q_*}z + \frac{\epsilon^l}{2\sqrt{q_*}}z.\right)} \tag{64}$$

We can now expand the activation function ϕ around small perturbations and then computing the square getting rid of higher order terms in ϵ^l, thus finally obtaining:

$$q_{aa}^{l+1} \simeq q_* + \frac{1+q_*}{\sqrt{q_*}} \frac{\sigma_m^2 \int \mathcal{D}z \phi\left(\sqrt{q_*}z\right) \phi'\left(\sqrt{q_*}z\right) z}{1 - \sigma_m^2 \int \mathcal{D}z \phi^2\left(\sqrt{q_*}z\right)} \epsilon^l \tag{65}$$

Comparing this expression with the one in eq. (63), we can then write:

$$\epsilon^{l+1} \simeq \frac{1+q_*}{\sqrt{q_*}} \frac{\sigma_m^2 \int \mathcal{D}z \phi\left(\sqrt{q_*}z\right) \phi'\left(\sqrt{q_*}z\right) z}{1 - \sigma_m^2 \int \mathcal{D}z \phi^2\left(\sqrt{q_*}z\right)} \epsilon^l. \tag{66}$$

Integrating by parts over z, we then obtain:

$$\epsilon^{l+1} \simeq \left[(1+q_*) \frac{\sigma_m^2 \int \mathcal{D}z \phi'\left(\sqrt{q_*}z\right) \phi'\left(\sqrt{q_*}z\right) + \int \mathcal{D}z \phi''\left(\sqrt{q_*}z\right) \phi\left(\sqrt{q_*}z\right)}{1 - \sigma_m^2 \int \mathcal{D}z \phi^2\left(\sqrt{q_*}z\right)}\right] \epsilon^l. \tag{67}$$

Given that it holds eq. (51), and noticing that χ evaluated at the correlation fixed point $c_* = 1$ is given by:

$$\chi|_{c_*=1} = \frac{\sigma_m^2}{1+\sigma_b^2} (1+q_*) \int \mathcal{D}z \left[\phi'\left(\sqrt{q_*}z\right)\right]^2, \tag{68}$$

we can finally get:

$$\epsilon^{l+1} \simeq \left[\chi|_{c_*=1} + \frac{\sigma_m^2 (1+q_*)}{1+\sigma_b^2} \int \mathcal{D}z \phi''\left(\sqrt{q_*}z\right) \phi\left(\sqrt{q_*}z\right)\right] \frac{\epsilon^l}{1+q_*}. \tag{69}$$

Given that we expect (62) to hold asymptotically, that is:

$$\epsilon^{l+1} \sim \exp\left(-\frac{l+1}{\xi_q}\right), \tag{70}$$

we can finally obtain the variance depth scale:

$$\xi_q^{-1} = \log(1+q_*) - \log\left(\chi|_{c_*=1} + \frac{\sigma_m^2 (1+q_*)}{1+\sigma_b} \int \mathcal{D}z \phi''\left(\sqrt{q_*}z\right) \phi\left(\sqrt{q_*}z\right)\right). \tag{71}$$

D SUPPLEMENTARY FIGURES

D.1 EDGE OF CHAOS SIMULATIONS: DETERMINISTIC SURROGATE CASE

We see in Figure 4 that the edges exist in the plane, for but $\sigma_b^2 > 10^{-20}$ all the corresponding mean variances $\sigma + m^2 > 1$ which is not possible.

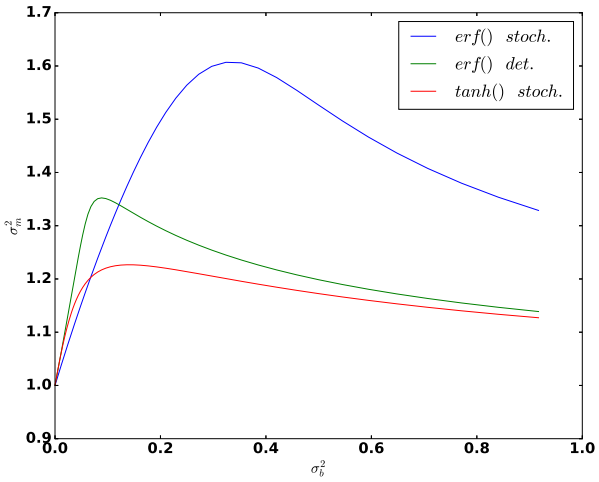

Figure 4: Edges of chaos for the deterministic surrogate model, for stochastic binary weights and stochastic or deterministic binary neurons. Presented is the edge of chaos in the (σ_m^2, σ_b^2), for both the a) stochastic neuron case with $\phi(z) = \text{erf}(\frac{1}{4}\ z)$, b) the deterministic sign neuron case with $\phi(z) = \text{erf}(\frac{1}{2}\cdot)$, and (c) the logistic based stochastic neuron, with $\tanh()$ approximation. We see all edges are above $\sigma^2 = 1$ for all but small $\sigma_b^2 << 1$.

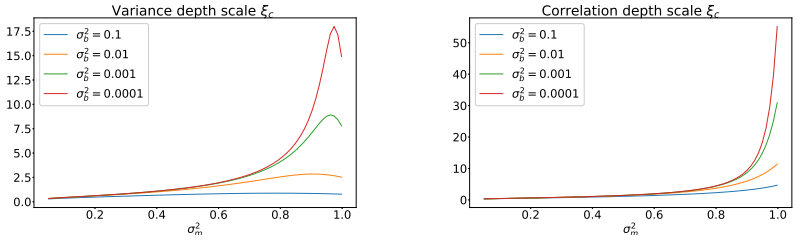

Figure 5: Depth scales as σ_m^2 is varied. (a) The depth scale controlling the variance propagation of a signal (b) The depth scale controlling correlation propagation of two signals. Notice that the correlation depth scale ξ_c only diverges as $\sigma_m^2 \to 1$, whereas for standard continuous networks, there are an infinite number of such points, corresponding to various combinations of the weight and bias variances.

D.2 DEPTH SCALES

We see in Figure 5 the depth scales for the deterministic surrogate. Note the divergence as one expects following the simulations in Figure 4.

D.3 JACOBIAN MEAN SQUARED SINGULAR VALUE AND MEAN FIELD GRADIENT BACKPROPAGATION

An alternative perspective on critical initialisation, to be contrasted with the forward signal propagation theory, is that we are simply attempting to control the mean squared singular value of the input-output Jacobain matrix of the entire network, which we can decompose into the product of single layer Jacobian matrices. In standard networks, the single layer Jacobian mean squared singular value is equal to the derivative of the correlation mapping χ as established in Poole et al. (2016). For the Gaussian model studied here this is not true, and corrections must be made to calculate the true mean squared singular value. This can be seen by observing the terms arising from denominator of the pre-activation field,

$$J_{ij}^{\ell} = \frac{\partial h_{i,a}^{\ell}}{\partial h_{j,a}^{\ell-1}} = \frac{\partial}{\partial h_j^{\ell}} \left(\frac{\bar{h}_{i,a}^{\ell}}{\sqrt{\Sigma_{ii}^{\ell}}} \right) = \phi'(h_{i,a}^{\ell}) \big[\frac{M_{ij}^{\ell}}{\sqrt{\Sigma_{ii}^{\ell}}} + (M_{ij}^{\ell})^2 \frac{\bar{h}_{i,a}^{\ell}}{(\Sigma_{ii}^{\ell})^{3/2}} \phi(h_{i,a}^{\ell}) \big] \tag{72}$$

Since Σ_{ii} is a quantity that scales with the layer width N_ℓ, it is clear that when we consider squared quantities, such as the mean squared singular value, the second term, from the derivative of the denominator, will vanish in the large layer width limit. Thus the mean squared singular value of the single layer Jacobian approaches χ. We will proceed as if χ is the exact quantity we are interested in controlling.The analysis involved in determining whether the mean squared singular value is well approximated by χ essentially takes us through the mean field gradient backpropagation theory as described in Schoenholz et al. (2016). This idea provides complementary depth scales for gradient signals travelling backwards.

E REPARAMETERISATION TRICK SURROGATE

E.1 SIGNAL PROPAGATION EQUATIONS

The signal propagation equations for the case of continuous neurons and stochastic binary weights yields the variance map,

$$q_{aa} = \mathbb{E}\phi^2(h_{j,a}^{l-1}) + \sigma_b^2 \tag{73}$$

Thus, once again, the variance map does not depend on the variance of the means of the binary weights. The covariance map however does retain a dependence on σ_m^2,

$$q_{ab}^{l} = \sigma_m^2 \mathbb{E}\phi(h_{j,a}^{l-1})\phi(h_{j,a}^{l-1}) + \sigma_b^2 \tag{74}$$

with the same expression as before. The correlation map is given by

$$c_{ab}^{l} = \frac{\sigma_m^2 \mathbb{E}\phi(h_{j,a}^{l-1})\phi(h_{j,a}^{l-1}) + \sigma_b^2}{\mathbb{E}\phi^2(h_{j,a}^{l-1}) + \sigma_b^2} \tag{75}$$

and we have the derivative of the correlation map given by

$$\chi = \sigma_m^2 \mathbb{E}\phi'(h_{j,a}^{l-1})\phi'(h_{j,b}^{l-1}) \tag{76}$$

E.2 DETERMINING THE EDGE OF CHAOS

Since the mean variance σ_m^2 does not appear in the variance map, we must once again consider different conditions for the edge of chaos. Specifically, from the correlation map we have a fixed point $c^* = 1$ if and only if

$$\sigma_m^2 = 1 \tag{77}$$

Figure 6: Training performance of the perturbed surrogate networks: a) evaluation of continuous surrogate, b) evaluation of corresponding the binary model (non-stochastic). Maximum depth $L = 30$, steps of $L = 2$, after ten epochs on reduced MNIST training set (10%), using SGD with momentum. Non-linearity used was $\tanh(\cdot)$ (with $\kappa = 1$), and divergence in trainability of continuous surrogate is observed for hyperparameter setting of $(\sigma_m^2, \sigma_b^2) = (1, 0)$

In turn, the edge of chaos condition $\chi_1 = 1$ holds if

$$\mathbb{E}[(\phi'(h_{j,a}^{l-1}))^2] = \frac{1}{\sigma_m^2} = 1 \tag{78}$$

Thus, to find the edge of chaos, we need to find a value of $q_{aa} = \mathbb{E}\phi^2(h_{j,a}^{l-1}) + \sigma_b^2$ that satisfies this final condition. In the case that $\phi(\cdot) = \tanh(\cdot)$, then the function $(\phi'(h_{j,a}^{l-1}))^2 \leq 1$, taking the value 1 at the origin only, this requires $q_{aa} \to 0$. Thus the 'edge of chaos' is the singleton point $(\sigma_b^2, \sigma_m^2) = (0, 1)$. This is confirmed by experiment, as we report in the subsequent sections.

It is of course possible to investigate this perturbed surrogate for different Bernoulli noise models. For example, given different noise scaling κ, as in the previous chapter, there will be a corresponding σ_b^2 that satisfy the edge of chaos condition. We leave such an investigation to future work, given the case of binary weights and continuous neurons does not appear to be of a particular interest over the binary neuron case. In the next section we present experiments confirming the success of the 'edge of chaos' initialisation for the perturbed surrogate with continuous neurons.

E.3 Experiments

As we see in Figures 6 and 7, the edge of chaos for the $\tanh(\cdot)$ non-linearity occurs only at the singleton point $(\sigma_b^2, \sigma_m^2) = (0, 1)$. Presented are simulations for varying σ_m^2, with fixed $\sigma_b^2 = 0$.

As in the previous chapter, we compare the continuous and binary surrogate performance, at training time Figure 6, and test time Figure 7. Once again, we observe the depth scale divergence in the continuous surrogate, but the binary network corresponding to the adapted means does not perform to similar depths. This effect was observed for different conditions, such as longer training time, larger network width and different gradient step sizes.

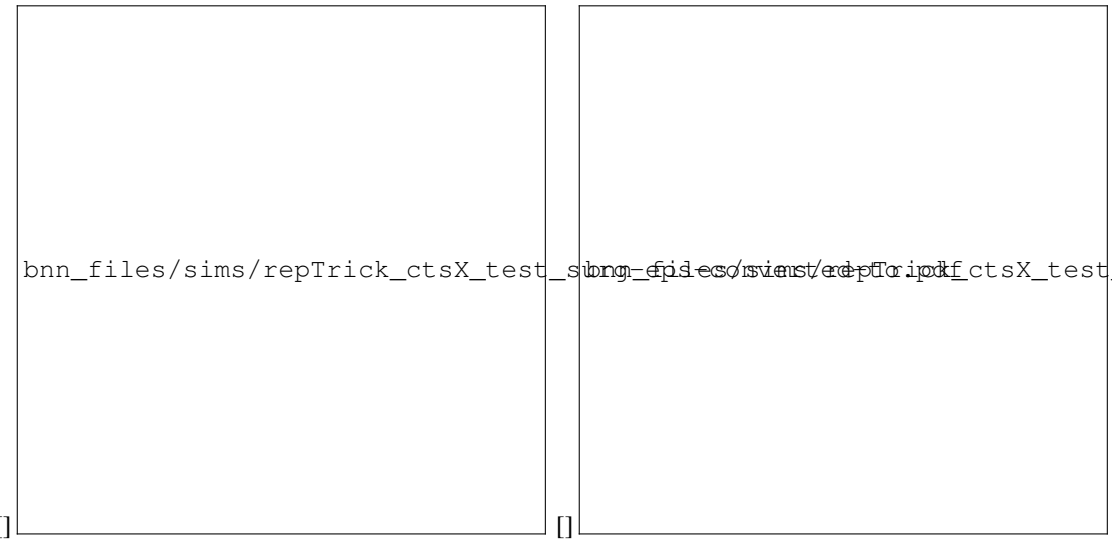

Figure 7: Test performance of the perturbed surrogate networks: a) evaluation of continuous surrogate, b) evaluation of corresponding the binary model (non-stochastic). Results corresponding to experiment presented in Figure 6.

F SIGNAL PROPAGATION OF BINARY NETWORKS

F.1 FORWARD SIGNAL PROPAGATION

In this neural network, it should be understood that all neurons are simply $\mathrm{sign}(\cdot)$ functions of their input, and all weights $W_{ij}^{\ell} \in \{\pm 1\}$ are randomly distributed according to

$$P(W_{ij}^{\ell} = +1) = 0.5 \tag{79}$$

$$\tag{80}$$

thus maintaining a zero mean.

The pre-activation field is given by

$$h_i^{\ell} = \frac{1}{\sqrt{N_{\ell-1}}} \sum_j W_{ij}^{\ell} \,\mathrm{sign}(h_j^{\ell-1}) + b_i^{\ell} \tag{81}$$

So, the length map is:

$$q_{aa}^{\ell} = \int Dz (\mathrm{sign}(\sqrt{q_{aa}^{\ell-1}} z)^2) + \sigma_b^2 \tag{82}$$

$$= 1 + \sigma_b^2 \tag{83}$$

Interestingly, this is the same value as for the perturbed Gaussian with stochastic binary weights and neurons.

The covariance evolves as

$$q_{ab}^{\ell} = \int Dz_1 Dz_2 \,\mathrm{sign}(u_a)\,\mathrm{sign}(u_b) + \sigma_b^2 \tag{84}$$

we again have a correlation map:

$$c_{ab}^{\ell} = (c_{ab}^{\ell-1}, q_{aa}^{\ell-1}, q_{bb}^{\ell-1}, b, \sigma_b) = \frac{\int Dz_1 Dz_2 \,\mathrm{sign}(u_a)\,\mathrm{sign}(u_b) + \sigma_b^2}{\sqrt{q_{aa}^{\ell-1} q_{bb}^{\ell-1}}} \tag{85}$$

We can find this correlation in closed form. First we rewrite our integral with h, for a joint density $p(h_a, h_b)$, and then rescale the h_a such that the variance is 1, so that $dh_a = \sqrt{q_{aa}} dv_a$

$$\int dh_a dh_b \operatorname{sign}(h_a) \operatorname{sign}(h_b) p(h_a, h_b) = \int dv_a dv_b \operatorname{sign}(v_a) \operatorname{sign}(v_b) p(v_a, v_b) \tag{86}$$

$$= \big(2P(v_1 > 0, v_2 > 0) - 2P(v_1 > 0, v_2 < 0)\big) \tag{87}$$

where $p(v_a, v_b)$ is a joint with the same correlation c_{ab} (which is now equal to its covariance), and the capital $P(v_1, v_2)$ corresponds to the (cumulative) distribution function. A standard result for standard bivariate normal distributions with correlation ρ,

$$P(v_1 > 0, v_2 > 0) = \frac{1}{4} + \frac{\sin^{-1}(\rho)}{2\pi}, \qquad P(v_1 > 0, v_2 < 0) = \frac{\cos^{-1}(\rho)}{2\pi} \tag{88}$$

So we then have that

$$\int dh_a dh_b \phi(h_a)\phi(h_b) p(h_a, h_b) = \sqrt{q_{aa} q_{bb}}\big(\frac{1}{2} + \frac{\sin^{-1}(c_{ab}^{\ell-1})}{\pi} - \frac{\cos^{-1}(c_{ab}^{\ell-1})}{\pi}\big) \tag{89}$$

Thus the correlation map is:

$$c_{ab}^{\ell} = \frac{\big(\frac{1}{2} + \frac{\sin^{-1}(c_{ab}^{\ell-1})}{\pi} - \frac{\cos^{-1}(c_{ab}^{\ell-1})}{\pi}\big) + \sigma_b^2}{\sqrt{q_{aa}^{\ell-1} q_{bb}^{\ell-1}}} \tag{90}$$

$$= \frac{\frac{2}{\pi}\sin^{-1}(c_{ab}^{\ell-1}) + \sigma_b^2}{\sqrt{q_{aa}^{\ell-1} q_{bb}^{\ell-1}}} \tag{91}$$

Since, from before we have $q_{aa} = 1 + \sigma_b^2$, we then obtain

$$c_{ab}^{\ell} = \frac{\frac{2}{\pi}\sin^{-1}(c_{ab}^{\ell-1}) + \sigma_b^2}{1 + \sigma_b^2} \tag{92}$$

Recall that $\sin^{-1}(1) = \frac{\pi}{2}$, so we have that $c^* = 1$ is a fixed point always.

We will now derive its slope, denoted as $\chi = \frac{\partial c_{ab}^{\ell}}{\partial c_{ab}^{\ell-1}}$, but by first integrating over the $\phi() = \operatorname{sign}()$ non-linearities, and then taking the derivative.

Now we are in a place to take the derivative :

$$\chi = \frac{\partial c_{ab}^{\ell}}{\partial c_{ab}^{\ell-1}} = \frac{2}{\pi} \frac{1}{\sqrt{q_{aa}^{\ell-1} q_{bb}^{\ell-1}}} \frac{1}{\sqrt{1 - (c_{ab}^{\ell-1})^2}} = \frac{2}{\pi} \frac{1}{(1 + \sigma_b^2)} \frac{1}{\sqrt{1 - (c_{ab}^{\ell-1})^2}} \tag{93}$$

We can see that the derivative χ diverges at $c_{ab}^{\ell} = 1$, meaning that there is no 'edge of chaos' for this system. This of course means that correlations will not propagate to arbitrary depth in deterministic binary networks, as one might have expected.

F.2 Stochastic weights and neurons

We begin again with the variance map,

$$q_{aa}^{l} = \mathbb{E}\left[(h_{i,a}^{l})^2\right] \tag{94}$$

where in this the field is given by

$$h_{i,a}^{l} = \frac{1}{\sqrt{N}} \sum_j W_{ij}^{l} x_{h_{j,a}^{l-1}} + b_i^{l} \tag{95}$$

where $x_{h_{j,a}^{l-1}}$ denotes a Bernoulli neuron whose natural parameter is the pre-activation from the previous layer.

The expectation for the length map is defined in terms of nested conditional expectations, since we wish to average over all random elements in the forward pass,

$$q_{aa}^{\ell} = \mathbb{E}_h \mathbb{E}_{x|h} x_{h_{j,a}^{l-1}} + \sigma_b^2 \tag{96}$$

$$= 1 + \sigma_b^2 \tag{97}$$

Once again, this is the same value as for the perturbed Gaussian with stochastic binary weights and neurons.

Similarly, the covariance map gives us,

$$q_{ab}^{l} = \mathbb{E}\left[h_{i,a}^{l} h_{i,b}^{l}\right] \tag{98}$$

$$= \mathbb{E}_{h_a,h_b} \mathbb{E}_{x_b|h_a} \mathbb{E}_{x_b|h_b} x_{h_{j,a}^{l-1}} x_{h_{j,b}^{l-1}} + \sigma_b^2 \qquad = \mathbb{E}\phi(h_{j,a}^{l-1})\phi(h_{j,a}^{l-1}) + \sigma_b^2 \tag{99}$$

with $phi(\cdot)$ being the mean function, or a shifted and scaled version of the cumulative distribution function for the Bernoulli neurons, just as in previous Chapters. This expression is equivalent to the perturbed surrogate for stochastic binary weights and neurons, with a mean variance of $\sigma_m^2 = 1$. Following the arguments for that surrogate, no edge of chaos exists.

F.3 Stochastic binary weights and continuous neurons

In this case, as we show in the appendix, the resulting equations are

$$q_{aa}^{\ell} = \mathbb{E}\phi^2(h_{j,a}^{l-1}) + \sigma_b^2 \tag{100}$$

$$q_{ab}^{l} = \mathbb{E}\phi(h_{j,a}^{l-1})\phi(h_{j,a}^{l-1}) + \sigma_b^2 \tag{101}$$

which are, once again, the same as for the perturbed surrogate in this case, with $\sigma_m^2 = 1$. This means that this model *does* have an edge of case, at the point $(\sigma_m^2, \sigma_b^2) = (1, 0)$.

F.4 Continuous weights and stochastic binary neurons

Similar arguments to the above show that the equations for this case are exactly equivalent to the perturbed surrogate model. This means that no edge of chaos exists in this case either.

G Miscellaneous comments

G.1 Remark: Valdity of the CLT for the first level of mean field

A legitimate immediate concern with initialisations that send $\sigma_m^2 \to 1$ may be that the binary stochastic weights $\mathbf{S}_{ij}^{\ell}$ are no longer stochastic, and that the variance of the Gaussian under the central limit theorem would no longer be correct. First recall the CLT's variance is given by $\text{Var}(\mathbf{h}_{\text{SB}}^{\ell}) = \sum_j (1 - m_j^2 x_j^2)$. If the means $m_j \to \pm 1$ then variance is equal in value to $\sum_j m_j^2 (1 - x_j^2)$, which is the central limit variance in the case of only Bernoulli neurons at initialisation. Therefore, the applicability of the CLT is invariant to the stochasticity of the weights. This is not so of course if *both* neurons and weights are deterministic, for example if neurons are just $\tanh()$ functions.