Exchangeability and Kernel Invariance in Trained MLPs

Russell Tsuchida[1] , **Fred Roosta**[1,2] and **Marcus Gallagher**[1]
[1]The University of Queensland
[2]International Computer Science Institute

Abstract

In the analysis of machine learning models, it is often convenient to assume that the parameters are IID. This assumption is not satisfied when the parameters are updated through training processes such as Stochastic Gradient Descent. A relaxation of the IID condition is a probabilistic symmetry known as *exchangeability*. We show the sense in which the weights in MLPs are exchangeable. This yields the result that in certain instances, the layer-wise kernel of fully-connected layers remains approximately constant during training. Our results shed light on such kernel properties throughout training while limiting the use of unrealistic assumptions.

1 Introduction

Despite the widespread usage of deep learning in applications, current theoretical understanding of deep networks continues to lag behind the pursued engineering outcomes. Much recent theory concerns networks in their randomized initial state, or contains assumptions about the parameters or data during training.

For example, Cho and Saul [2009], Daniely *et al.* [2016], Bach [2017] and Tsuchida *et al.* [2018] analyze the kernels of neural networks with random IID weights. Insightful analysis connecting signal propagation in deep networks to chaos have made similar assumptions [Poole *et al.*, 2016; Raghu *et al.*, 2017]. Random matrix theory has recently been applied to neural networks in an attempt to understand the empirical spectral distribution (ESD) of the Hessian [Pennington and Bahri, 2017] and the Gram matrix [Pennington and Worah, 2017], but these works have made strong assumptions on the weight and data distributions.

The most widely used yet unrealistic assumption is that weights remain IID throughout training. Relaxing such unrealistic assumptions makes obtaining meaningful results more challenging. We take a step in this direction by investigating the probabilistic symmetry known as *exchangeability*, which is a generalization of the IID assumption. We uncover the striking result that the layer-wise kernel of MLPs with ReLU activations, trained with many optimizers, *remains constant* up to a scaling factor during training when the network inputs satisfy certain conditions. Otherwise, we are able to bound the absolute difference between layer-wise kernel and the kernel of the network in its random IID state.

2 Background

2.1 Notation

Random variables, vectors and matrices will be denoted by upper case, bold upper case, and bold upper case with overline characters, respectively. Parenthesized superscripts index the layer of the network to which an object belongs. The first and second post-subscripts index the rows and columns of a matrix, respectively. When the row of a matrix is extracted through an index, it will be assumed to be transposed into a column vector. Pre-subscripts will indicate the iteration of an iterative optimizer. Expectation with respect to the distribution of and random variable R is denoted $\mathbb{E}_R$.

Consider an MLP with an input layer and L non-input layers. Denote the number of neurons in layer $0 \leq l \leq L$ by $n^{(l)}$. Denote an input to the network by $\mathbf{x}$. Denote the random weight matrix connecting layer $l-1$ to layer l by $\overline{\mathbf{W}}^{(l)}$. Denote the ℓ^2 norm by $\| \cdot \|$. Denote the activation function by σ. We consider ReLU activations throughout.

2.2 Exchangeability

An *exchangeable* sequence of random variables $(Q_1, Q_2, ...)$ has the property that the joint distribution of the sequence is invariant to finite permutations. That is, a sequence $(Q_i)_{i\geq 1}$ is exchangeable if $(Q_1, Q_2, ...) \overset{d}{=} (Q_{\pi(1)}, Q_{\pi(2)}, ...)$ for all finite permutations π. To aid in readability we will omit the index set in the subscript, so that $(Q_i)_{i\geq 1}$ is the same as $(Q_i)_i$.

Infinite exchangeable sequences are characterized as mixtures of IID random variables through de Finetti's theorem.

Theorem 1. *[Aldous, 1981] An infinite sequence* $\mathbf{Q} = (Q_i)_i$ *is exchangeable if and only if there exists a measurable function* f *such that* $(Q_i)_i \overset{d}{=} \big(f(A, B_i)\big)_i$, *where* A *and* $\boldsymbol{B}$ *are mutually IID random variables uniform on* $[0, 1]$.

Generalizations of Theorem 1 to multi-dimensional arrays exist [Kallenberg, 2006]. A matrix $\overline{\mathbf{Q}}$ is *row and column exchangeable* (RCE) if its joint distribution is invari-

Supplemental material available at the arXiv version https://arxiv.org/abs/1810.08351.

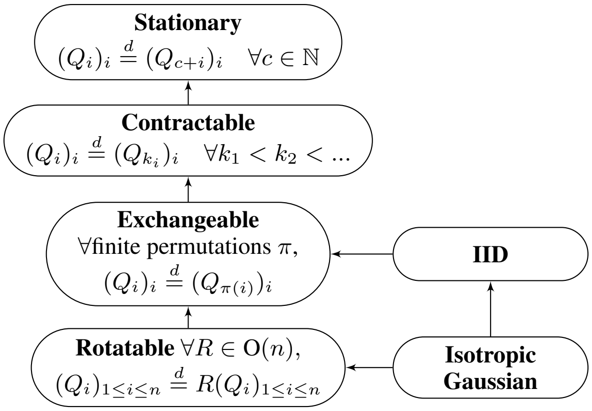
Stationary
$(Q_i)_i \overset{d}{=} (Q_{c+i})_i \quad \forall c \in \mathbb{N}$
Contractable
$(Q_i)_i \overset{d}{=} (Q_{k_i})_i \quad \forall k_1 < k_2 < ...$
Exchangeable
$\forall$finite permutations π,
$(Q_i)_i \overset{d}{=} (Q_{\pi(i)})_i$
IID
Rotatable $\forall R \in \mathrm{O}(n)$,
$(Q_i)_{1\le i\le n} \overset{d}{=} R(Q_i)_{1\le i\le n}$
Isotropic Gaussian

Figure 1: Relative strength of probabilistic symmetries.

ant to row and column permutations. That is, $\overline{\mathbf{Q}}$ is RCE if $(Q_{ji})_{ji} \overset{d}{=} (Q_{\pi_1(j)\pi_2(i)})_{ji}$ for all finite permutations π_1, π_2.

Theorem 2. *[Aldous, 1981] An infinite array $\overline{\mathbf{Q}} = (Q_{ji})_{ji}$ is RCE if and only if there exists a measurable function f such that $(Q_{ji})_{ji} \overset{d}{=} \big(f(A, B_j, C_i, D_{ji})\big)_{ji}$, where $A, \mathbf{B}, \mathbf{C}$, and $\overline{\mathbf{D}}$ are mutually IID uniform on $[0,1]$.*

Intuition concerning the strength of exchangeability in the context of probabilistic symmetries may be aided by the implication graph shown in Figure 1.

2.3 Kernels of Random MLPs

There is a well-studied connection between the feature maps in MLPs (and other neural network architectures) and the kernel of a reproducing kernel Hilbert space (RKHS) [MacKay, 1992; Neal, 1994; Cho and Saul, 2009; Daniely *et al.*, 2016; Bach, 2017; Bietti and Mairal, 2017]. Consider the angle $\theta^{(l)}$ between two random signals $\boldsymbol{\sigma}(\overline{\mathbf{W}}^{(l)}\mathbf{x})$ and $\boldsymbol{\sigma}(\overline{\mathbf{W}}^{(l)}\mathbf{y})$ in the lth hidden layer of an MLP for inputs $\mathbf{x}$ and $\mathbf{y}$. We have

$$\cos\theta^{(l)} = \frac{\sum_{j=1}^{n^{(l)}} \sigma\big(\mathbf{W}_j^{(l)}\cdot\mathbf{x}\big)\sigma\big(\mathbf{W}_j^{(l)}\cdot\mathbf{y}\big)}{\sqrt{\sum_{j=1}^{n^{(l)}} \sigma\big(\mathbf{W}_j^{(l)}\cdot\mathbf{x}\big)\sigma\big(\mathbf{W}_j^{(l)}\cdot\mathbf{x}\big) \sum_{j=1}^{n^{(l)}} \sigma\big(\mathbf{W}_j^{(l)}\cdot\mathbf{y}\big)\sigma\big(\mathbf{W}_j^{(l)}\cdot\mathbf{y}\big)}} \tag{1}$$

where $\mathbf{W}_j^{(l)}$ is the jth row of $\overline{\mathbf{W}}^{(l)}$. We divide the numerator and denominator by $n^{(l)}\|\mathbf{x}\|\|\mathbf{y}\|$ and use the absolute-homogeneity property of the ReLU $\sigma(|a|z) = |a|\sigma(z)$ to consider the scaled numerator

$$\frac{1}{n^{(l)}}\sum_{j=1}^{n^{(l)}} \sigma\Big(\mathbf{W}_j\cdot\mathbf{x}/\|\mathbf{x}\|\Big)\sigma\Big(\mathbf{W}_j\cdot\mathbf{y}/\|\mathbf{y}\|\Big). \tag{2}$$

Let $\hat{\mathbf{x}} = \frac{\mathbf{x}}{\|\mathbf{x}\|}$. Suppose that each row $\mathbf{W}_j^{(l)}$ of $\overline{\mathbf{W}}^{(l)}$ is IID with all other rows (we relax this requirement later) and is defined on some probability space (Ω, Σ, μ). Asymptotically in the number of neurons $n^{(l)}$, the strong law of large numbers implies that (2) converges almost surely to

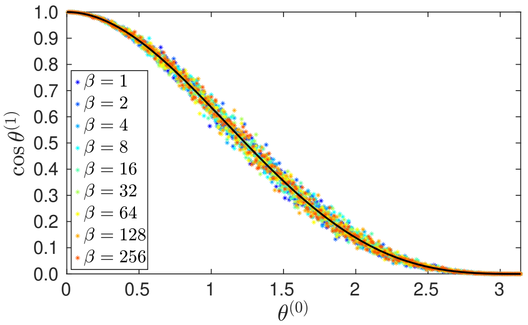

Figure 2: Normalized kernel for a hidden layer with ReLU activations. Samples from a network with 1000 inputs and hidden units are obtained by generating an orthogonal matrix R from a QR decomposition of a random matrix containing IID samples from $\mathcal{U}[0,1]$, then setting $\mathbf{x} = R(1,0,...,0)^T$ and $\mathbf{y} = R(\cos\theta, \sin\theta, 0, ..., 0)^T$.

$$\begin{aligned}&\mathbb{E}\big[\sigma(\mathbf{W}_j^{(l)}\cdot\hat{\mathbf{x}})\sigma(\mathbf{W}_j^{(l)}\cdot\hat{\mathbf{y}})\big] \\ &= \int_\Omega \sigma(\mathbf{W}_j^{(l)}\cdot\hat{\mathbf{x}})\sigma(\mathbf{W}_j^{(l)}\cdot\hat{\mathbf{y}})\,d\mu,\end{aligned} \tag{3}$$

which corresponds to an inner-product in feature space. The kernel is positive semi-definite and uniquely defines an RKHS. When μ is the product measure corresponding to an IID Gaussian with variance $\mathbb{E}\big[(W_{11}^{(l)})^2\big]$ and 0 mean, the kernel has a closed-form expression known as the arc-cosine kernel (of degree 1) [Cho and Saul, 2009], given by

$$\frac{\mathbb{E}\big[(W_{11}^{(l)})^2\big]}{2\pi}\big(\sin\theta^{(l-1)} + (\pi - \theta^{(l-1)})\cos\theta^{(l-1)}\big), \tag{4}$$

where $\theta^{(l-1)}$ is the angle between $\mathbf{x}$ and $\mathbf{y}$. We will refer to (3) as the *layer-wise kernel in layer* l, denoted $k^{(l)}(\mathbf{x},\mathbf{y})$. When (3) is normalized in the same fashion as (1), we will call the resulting quantity the *layer-wise normalized kernel*.

2.4 Layer-wise Kernel in IID MLPs

Our analysis draws upon and extends results concerning the layer-wise normalized kernels of MLPs with IID weights [Tsuchida *et al.*, 2018], which, for completeness, we briefly review here. Construct a sequence $\{\mathbf{x}_{(m)}\}_{m\ge 2}$ such that for all m, $\mathbf{x}_{(m)} \in \mathbb{R}^\infty$ and coordinates $m+1, m+2, ...$ of $\mathbf{x}_{(m)}$ are all 0. Define the sequence $\{\mathbf{y}_{(m)}\}_{m\ge 2}$ in the same way, and additionally require that the angle $\theta^{(l-1)}$ between $\mathbf{x}_{(m)}$ and $\mathbf{y}_{(m)}$ is constant in m. Denote the randomly initialized weight matrix by ${}_0\overline{\mathbf{W}}^{(l)}$. We would like to evaluate

$$\lim_{m\to\infty}\mathbb{E}\big[\sigma({}_0\mathbf{W}_j^{(l)}\cdot\hat{\mathbf{x}}_{(m)})\sigma({}_0\mathbf{W}_j^{(l)}\cdot\hat{\mathbf{y}}_{(m)})\big]. \tag{5}$$

Sufficient conditions for the central limit theorem (CLT) are given below. Let $x_{(m)i}$ denote the ith coordinate of $\mathbf{x}_{(m)}$.

Hypothesis 3. $\lim_{m\to\infty} m^{(1/4)} \max_{i=1}^m \frac{|x_{(m)i}|}{\|\mathbf{x}_{(m)}\|}$ *and* $\lim_{m\to\infty} m^{(1/4)} \max_{i=1}^m \frac{|y_{(m)i}|}{\|\mathbf{y}_{(m)}\|}$ *are both* 0.

This condition is easily satisfied since for data points with many non-zero entries, $\|\mathbf{x}_{(m)}\|$ will grow like $\sqrt{m}$ when compared to $|x_{(m)i}|$. Provided $\mathbb{E}\big[{}_0W_{11}^{(l)}\big] = 0$ and $\mathbb{E}\big|{}_0W_{11}^{(l)}\big|^3 < \infty$, Tsuchida *et al.* [2018] show that under Hypothesis 3,

$$\sigma\Big({}_0\mathbf{W}_1^{(l)} \cdot \hat{\mathbf{x}}_{(m)}\Big)\sigma\Big({}_0\mathbf{W}_1^{(l)} \cdot \hat{\mathbf{y}}_{(m)}\Big) \xrightarrow{d} \sigma(Z_{\mathbf{x}})\sigma(Z_{\mathbf{y}}),$$

$(Z_{\mathbf{x}}, Z_{\mathbf{y}}) \sim \mathcal{N}(\mathbf{0}, \Sigma)$ with $\Sigma = \begin{bmatrix} 1 & \cos\theta^{(0)} \\ \cos\theta^{(0)} & 1 \end{bmatrix}$.

Letting $Z_{\mathbf{x}(m)} = {}_0\mathbf{W}_j^{(l)} \cdot \hat{\mathbf{x}}_{(m)}$, $Z_{\mathbf{y}(m)} = {}_0\mathbf{W}_j^{(l)} \cdot \hat{\mathbf{y}}_{(m)}$,

$$\sigma(Z_{\mathbf{x}(m)})\sigma(Z_{\mathbf{y}(m)}) \leq |Z_{\mathbf{x}(m)}||Z_{\mathbf{y}(m)}| \leq Z_{\mathbf{x}(m)}^2 + Z_{\mathbf{y}(m)}^2.$$

The integral of the RHS is $2\mathbb{E}\big[({}_0W_{11}^{(l)})^2\big]$, so the limit may be brought inside the integral in (5) by Theorem 19 of Royden [2010]. The resulting expectation is (4).

Figure 2 shows the normalized kernels for random weights with PDF $\prod_{i=1}^m \frac{\beta}{2\alpha\Gamma(1/\beta)} e^{-|w_i/\alpha|^\beta}$. This PDF generalizes the isotropic Gaussian PDF ($\beta = 2$) and the Uniform PDF ($\beta \to \infty$). The CLT result says nothing about the kernel of *trained* networks whose weights are not IID. In §4 we extend the CLT result to trained networks. We do this by first exploring exchangeability in MLPs.

3 Exchangeability in MLPs

Suppose that for every l, the matrix $({}_0W_{ji}^{(l)})_{ji}$ is IID and then the weights evolve according to SGD over t iterations. The index j (which corresponds to the jth row of the random weight matrix, or the jth neuron in layer l) is an arbitrary labeling; one may permute these indices along with the corresponding connection in layer $l+1$ without changing the output of the network or the joint distribution of the weights.

We show this for $L = 3$; the generalization to any $L \geq 2$ will be straightforward. To start our argument, it is clear that there is full exchangeability of the weights when the network has been randomly initialized with IID weights and has not yet been trained. More restrictively, we have the following.

Observation 4. *Let $\mathbf{a} \in \mathbb{R}^{n^{(0)}}$ and $\mathbf{b} \in \mathbb{R}^{n^{(3)}}$ be inputs and targets of an MLP. Suppose that the initial weights in each layer ${}_0\overline{\mathbf{W}}^{(l)}$ are IID, and temporarily drop the pre-subscript. Then for any bijective permutations π_1 and π_2,*

$$\begin{aligned}&\Big(\mathbf{a}, \big(W_{\pi_1(i)h}^{(1)}\big)_{ih}, \big(W_{\pi_2(j)\pi_1(i)}^{(2)}\big)_{ji}, \big(W_{k\pi_2(j)}^{(3)}\big)_{kj}, \mathbf{b}\Big) \\ &\overset{d}{=}\Big(\mathbf{a}, \big(W_{ih}^{(1)}\big)_{ih}, \big(W_{ji}^{(2)}\big)_{ji}, \big(W_{kj}^{(3)}\big)_{kj}, \mathbf{b}\Big). \qquad (6)\end{aligned}$$

This generalizes to any network with one or more hidden layers ($L \geq 2$) because the permutation does not affect the non-exchangeable elements $\mathbf{a}$ and/or $\mathbf{b}$. Define $g_{qp}^{(l)}$ to be the function that takes $\mathbf{a}$, $\mathbf{b}$ and realizations of $\big({}_0\overline{\mathbf{W}}^{(m)}\big)_{m\in[L]}$ and calculates realizations of ${}_1W_{qp}^{(l)}$ according to an online (batch size of 1) backpropagation update rule. Let $\overline{\mathbf{g}}^{(l)}$ be a matrix-valued function, whose qpth element is $g_{qp}^{(l)}$. We have

$$\overline{\mathbf{g}}^{(l)}\Big(\mathbf{a}, \mathbf{b}, \big(\overline{\mathbf{w}}^{(r)}\big)_{r\in[L]}\Big) = \overline{\mathbf{w}}^{(l)} - \alpha\frac{\partial E}{\partial \overline{\mathbf{w}}^{(l)}},$$

for some cost function $E\Big(\mathbf{a}, \mathbf{b}; \big(\overline{\mathbf{w}}^{(r)}\big)_{r\in[L]}\Big)$ and step-size α. Denote the LHS of (6) by $\mathbf{U}$ and the RHS of (6) by $\mathbf{U}_\pi$. Then by examining the backpropagation equations,

$$\frac{\partial E}{\partial w_{ji}^{(l)}}\bigg|_{\mathbf{U}_\pi} = \frac{\partial E}{\partial w_{\pi_2(j)\pi_1(i)}^{(l)}}\bigg|_{\mathbf{U}}.$$

By the continuous mapping theorem, we may apply $\mathbf{g}^{(2)}$ to both sides of (6) if $\mathbf{g}^{(2)}$ is almost everywhere (a.e.) continuous. Temporarily dropping the 0 pre-subscripts on the weights,

$$\begin{aligned}&\overline{\mathbf{g}}^{(2)}\Big(\mathbf{a}, \big(W_{ih}^{(1)}\big)_{ih}, \big(W_{ji}^{(2)}\big)_{ji}, \big(W_{kj}^{(3)}\big)_{kj}, \mathbf{b}\Big), \\ &\overset{d}{=}\overline{\mathbf{g}}^{(2)}\Big(\mathbf{a}, \big(W_{\pi_1(i)h}^{(1)}\big)_{ih}, \big(W_{\pi_2(j)\pi_1(i)}^{(2)}\big)_{ji}, \big(W_{k\pi_2(j)}^{(3)}\big)_{kj}, \mathbf{b}\Big), \\ &=\bigg(g_{\pi_2(q)\pi_1(p)}^{(2)}\Big(\mathbf{a}, \big(W_{ih}^{(1)}\big)_{ih}, \big(W_{ji}^{(2)}\big)_{ji}, \big(W_{kj}^{(3)}\big)_{kj}, \mathbf{b}\Big)\bigg)_{qp}, \\ &=\Big({}_1W_{\pi_2(q)\pi_1(p)}^{(2)}\Big)_{qp}, \qquad (7)\end{aligned}$$

and the first line is equal to $\big({}_1W_{qp}^{(2)}\big)_{qp}$. This shows that ${}_1\overline{\mathbf{W}}^{(2)}$ is RCE. ${}_t\overline{\mathbf{W}}^{(1)}$ is row but not column-exchangeable and ${}_t\overline{\mathbf{W}}^{(L)}$ is column but not row-exchangeable.

When any batch size M is used, the inputs $\mathbf{a}$ and $\mathbf{b}$ may be replaced by sets $\{\mathbf{a}_i\}_{i\leq M}$ and $\{\mathbf{b}_i\}_{i\leq M}$ and (7) still holds. If M is the size of the entire finite dataset, this corresponds to gradient descent. We may use any a.e. continuous $\overline{\mathbf{g}}^{(l)}$ whose evaluation commutes with index permutations in the input (such as SGD, Adam [Kingma and Ba, 2015] or RMSprop). Call such an update rule *index commuting*. By redefining $\overline{\mathbf{g}}^{(2)}$ to calculate the weights at the tth iteration of SGD, one can show that ${}_t\overline{\mathbf{W}}^{(2)}$ is RCE for all t.

Theorem 5. *Let $L \geq 3$. Suppose that the initial weights in each layer ${}_0\overline{\mathbf{W}}^{(l)}$ are IID. Suppose the network is trained using an index commuting update rule. Then for all $2 \leq l \leq L-1$ and all optimizer iterations $t \geq 0$, the weight matrices ${}_t\overline{\mathbf{W}}^{(l)}$ are RCE. For $L \geq 2$, ${}_t\overline{\mathbf{W}}^{(1)}$ is row but not column exchangeable and ${}_t\overline{\mathbf{W}}^{(L)}$ is column but not row exchangeable.*

4 Kernels of Trained MLPs

We now extend the results of §2.4 to trained networks using the results of §3. For the remainder of the paper we will drop the pre-subscript t denoting the training iteration on the weights.

4.1 Layer-wise Kernel in Trained MLPs

We examine the limit in m of the layer-wise kernel in layer l for a network with infinitely many RCE weights. By Theorem 1, there exists some measurable function f and some mutually independent A and $\mathbf{B}$ each uniform on $[0,1]$ such that

$$\begin{aligned}&\lim_{m\to\infty}\mathbb{E}\Big[\sigma\Big(\mathbf{W}_1^{(l)} \cdot \hat{\mathbf{x}}_{(m)}\Big)\sigma\Big(\mathbf{W}_1^{(l)} \cdot \hat{\mathbf{y}}_{(m)}\Big)\Big] \\ &= \lim_{m\to\infty}\int_{[0,1]} k_A(\mathbf{x}_{(m)}, \mathbf{y}_{(m)})\, d\mu_A, \qquad (8)\end{aligned}$$

where μ_A is the uniform probability measure on $[0,1]$ with $k_A(\mathbf{x}_{(m)}, \mathbf{y}_{(m)})$ given by

$$\int\limits_{[0,1]^m} \sigma\Big(\mathbf{f}_A(\mathbf{B}) \cdot \hat{\mathbf{x}}_{(m)}\Big) \sigma\Big(\mathbf{f}_A(\mathbf{B}) \cdot \hat{\mathbf{y}}_{(m)}\Big)\, d\mu_{\mathbf{B}},$$

where $\big(\mathbf{f}_A(\mathbf{B})_i\big)_i = \big(f(A, B_i)\big)_i$ and $\mu_{\mathbf{B}}$ is the uniform probability measure. We prove the following in Appendix A.

Proposition 6. *Suppose that* $2 \leq l \leq L-1$*,* $\mathbb{E}\big|W_{11}^{(l)}\big|^3 < \infty$*,* $\mathbb{E}\big|W_{11}^{(l)} W_{12}^{(l)}\big| < \infty$*, Hypothesis 3 is satisfied and* $\lim\limits_{m\to\infty} \sum\limits_{i=1}^{m} \hat{x}_{(m)i} = \lim\limits_{m\to\infty} \sum\limits_{i=1}^{m} \hat{y}_{(m)i} = 0$ *or* $\mathbb{E}\big[W_{11}^{(l)} W_{12}^{(l)}\big] = 0$*. Then* (8) *is given by*

$$\frac{1}{2\pi}\Big(\mathbb{E}\big[(W_{11}^{(l)})^2\big] - \mathbb{E}\big[W_{11}^{(l)} W_{12}^{(l)}\big]\Big)$$
$$\big(\sin\theta^{(l-1)} + (\pi - \theta^{(l-1)})\cos\theta^{(l-1)}\big). \qquad (9)$$

Note that (9) and (4) are the same up to a scaling factor, which cancels out after normalizing.

4.2 The Ergodic Problem

Unfortunately, (8) is not necessarily the inner product in feature space of an infinitely wide network. By Theorem 2,

$$\frac{1}{n}\sum_{j=1}^{n} \sigma\big(\mathbf{W}_j^{(l)} \cdot \hat{\mathbf{x}}_{(m)}\big)\sigma\big(\mathbf{W}_j^{(l)} \cdot \hat{\mathbf{y}}_{(m)}\big)$$
$$\overset{d}{=} \frac{1}{n}\sum_{j=1}^{n} \sigma\big(\mathbf{f}_{A\mathbf{C}}(B_j, \mathbf{D}_j) \cdot \hat{\mathbf{x}}_{(m)}\big)\sigma\big(\mathbf{f}_{A\mathbf{C}}(B_j, \mathbf{D}_j) \cdot \hat{\mathbf{y}}_{(m)}\big),$$

for some measurable $\mathbf{f}_{A\mathbf{C}}(B_1, \mathbf{D}_1) = (f(A, B_j, C_i, D_{ji})_i$, which converges almost surely to the random variable

$$\mathbb{E}_{B_1\mathbf{D}_1}\Big[\sigma\big(\mathbf{f}_{A\mathbf{C}}(B_1, \mathbf{D}_1) \cdot \hat{\mathbf{x}}_{(m)}\big)\sigma\big(\mathbf{f}_{A\mathbf{C}}(B_1, \mathbf{D}_1) \cdot \hat{\mathbf{y}}_{(m)}\big)\Big] \qquad (10)$$

depending on A and $\mathbf{C}$ by the Birkhoff-Khinchin ergodic theorem (see Appendix E). For the purposes of experimenting, we make the following simplifying assumption.

Hypothesis 7. *The following holds:*

$$\frac{1}{n}\sum_{j=1}^{n} \sigma\big(\mathbf{W}_j^{(l)} \cdot \hat{\mathbf{x}}_{(m)}\big)\sigma\big(\mathbf{W}_j^{(l)} \cdot \hat{\mathbf{y}}_{(m)}\big)$$
$$\xrightarrow{p} \mathbb{E}\Big[\sigma\big(\mathbf{W}_1^{(l)} \cdot \hat{\mathbf{x}}_{(m)}\big)\sigma\big(\mathbf{W}_1^{(l)} \cdot \hat{\mathbf{y}}_{(m)}\big)\Big].$$

Hypothesis 7 says that taking averages over j of the products of activations in *one* network is equivalent to taking averages over one fixed neuron of the products of activations in an *ensemble* of independent networks. A sufficient condition is that the measure is *ergodic* with respect to the row-shift transformation. This condition is stronger than necessary. In statistical mechanics, an "approximate ergodicity" applied to *sum functions* is used to compare time averages with phase averages [Khinchin, 1949; Kurth, 2014]. The Ergodic Problem features heavily in the history of statistical mechanics [Moore, 2015]. It is our hope that by introducing this assumption into the analysis of MLPs, we make further progress towards efforts in connecting neural networks to statistical mechanics [Martin and Mahoney, 2017]. In §5 we demonstrate that Hypothesis 7 is not inconsistent with our empirical observations.

5 Experiments

We illustrate our results with selected figures. Other datasets and optimizers are investigated in the supplemental material.

5.1 Verification of Proposition 6

Architecture. We train an autoencoder with 4 layers and 3072 neurons in each layer on CIFAR10 [Krizhevsky and Hinton, 2009] with pixel values normalized to $[0,1]$ using an $\ell 2$ objective. Weights are initialized with a variance of $\frac{2}{n_l}$ [He *et al.*, 2015].

Method. In Figure 3 we plot the empirical layer-wise normalized kernel in each layer. The color of the points moves from blue to red as the training iteration t increases. Each sample is generated using Procedure 1. The numerical steps ensure that the desired angle $\theta^{(l-1)}$ is obtained between $\mathbf{x}$ and $\mathbf{y}$. The alphabetical steps ensure that $\sum x_i = \sum y_i = 0$.

Procedure 1 Sample $\theta^{(l-1)}$

```
    Inputs datapoint $\mathbf{x}$, $\theta^{(l-1)}$ Output $\mathbf{y}$ at angle $\theta^{(l-1)}$ to $\mathbf{x}$.
a   Set the last two coordinates of $\mathbf{x}$ to 0.
1   Sample random vector $\mathbf{p}$ orthogonal to $\mathbf{x}$: Set all coordinates of $\mathbf{p}$ to zero where $\mathbf{x}$ is non-zero and sample remaining coordinates of $\mathbf{p}$ from $\mathcal{U}[0,1]$. Set last two coordinates to 0. Normalize $\mathbf{p}$ so that $\|\mathbf{x}\| = \|\mathbf{p}\|$.
b   Set the second last coordinate of $\mathbf{x}$ to the negative sum of all coordinates of $\mathbf{x}$.
c   Set the last coordinate of $\mathbf{p}$ to the negative sum of all coordinates of $\mathbf{p}$.
2   Return $\mathbf{y} = \cos\theta^{(l-1)}\mathbf{x} + \sin\theta^{(l-1)}\mathbf{p}$.
```

5.2 Inputs with Non-Zero Sums

Consider a modification of the method described in §5.1: the alphabetical steps of Procedure 1 are not performed. This means that the sums $\sum \hat{x}_i$ and $\sum \hat{y}_i$ are no longer 0. However, if $\mathbb{E}\big[W_{11}^{(l)} W_{12}^{(l)}\big] = 0$, Proposition 6 still applies. Note that

$$\mathbb{E}\big[W_{11}^{(l)} W_{12}^{(l)}\big] = \int\limits_{[0,1]^3} f(A, B_1) f(A, B_2)\, d\mu_{B_1 B_2 A}$$
$$= \int\limits_{[0,1]} \Big(\int\limits_{[0,1]} f(A, B_1)\, d\mu_{B_1}\Big)^2 d\mu_A$$
$$= 0 \text{ iff } \int\limits_{[0,1]} f(A, B_1)\, d\mu_{B_1} = 0.$$

Also, by the strong law of large numbers,

$$\frac{1}{n}\sum_{i=1}^{n} W_{1i}^{(l)} \xrightarrow{a.s.} \int_{[0,1]} f(A, B_1)\, d\mu_{B_1}.$$

Therefore, for finite $n^{(l)}$ if $(E^{n^{(l)}})^2 := \max\limits_j \Big(\Big(\frac{1}{n^{(l)}} \sum\limits_{i=1}^{n^{(l)}} W_{ji}^{(l)}\Big)^2\Big)$ is "small", $\mathbb{E}\big[W_{11}^{(l)} W_{12}^{(l)}\big]$ will

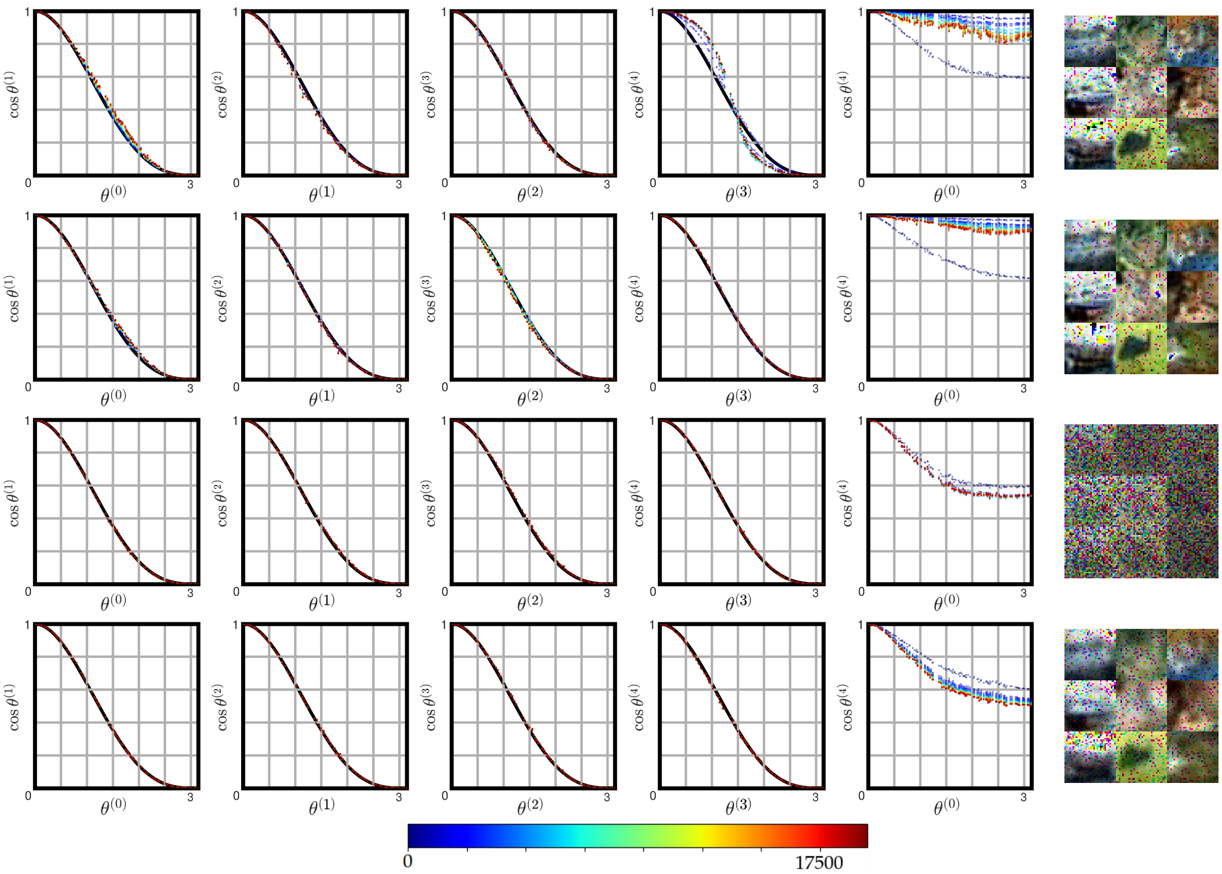

Figure 3: Layer-wise normalized kernels for a trained MLP at iteration t, indicated by color. Batch-size of 256 used. First 4 columns: layers 1 to 4. Fifth column: full network. Last column: sample reconstructions on test data, indicating whether or not training converged. First 3 rows: adam using step size 0.001, $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\varepsilon = [10^{-16}, 10^{-8}, 1]$. Last row: SGD with constant learning rate 0.5.

be "small". We are interested in finding optimizer hyperparameters that result in $(E^{n^{(l)}})^2 \neq 0$, which in turn results in deviations from (9). We make the following observations:

(1) When **Adam**, **RMSProp** or **Nadam** [Dozat, 2016] are used, as the hyperparameter ε decreases there is a sharp change in $(E^{n^{(l)}})^2$ and the mean squared error (MSE) of the observed normalized kernel to the normalized arc-cosine kernel of degree one measured at iteration $t = 19000$. When $(E^{n^{(l)}})^2$ is small the kernel is approximately described by (9). See Figures 4 and 5 and Appendices F and G.

(2) SGD using step sizes α that result in stable training generally have smaller $(E^{n^{(l)}})^2$ than Adam, and thus the normalized kernel agrees more closely with Proposition 6. See Figures 4 and 5 and Appendices F and G.

6 Discussion and Conclusion

We identified that the weights in hidden layers of MLPs are RCE. Using this symmetry, we analyzed the kernels of *trained* networks. Specifically, we found that the normalized kernel remains invariant when the inputs have sums over their coordinates of 0. When the sums are not 0, a bound which depends on $\mathbb{E}[W_{11}^{(l)} W_{12}^{(l)}]$ applies to the residual of the normalized kernel to the normalized arc-cosine kernel. We derived a measure $(E^{(n^{(l)}})^2$ which, when close to 0, indicates whether $\mathbb{E}[W_{11}^{(l)} W_{12}^{(l)}]$ is close to 0 and thus whether the normalized kernel remains approximately invariant during training.

When empirically comparing optimizers, those which result in small $\mathbb{E}[W_{11}^{(l)} W_{12}^{(l)}]$ have kernels which follow the normalized arc-cosine kernel during training. The parameter ε present in Adam and other optimizers can increase $\mathbb{E}[W_{11}^{(l)} W_{12}^{(l)}]$, leading to qualitatively different kernels to the normalized arc-cosine kernel. Changes in other hyperparameters may change $\mathbb{E}[W_{11}^{(l)} W_{12}^{(l)}]$, although we had difficulty finding instances where changing α in SGD resulted in a kernel that did not roughly match the normalized arc-cosine kernel without also resulting in unstable training.

In contrast with works that analyze weight distributions through an approximation of SGD by a stochastic differential equation [Seung *et al.*, 1992; Watkin *et al.*, 1993; Martin and Mahoney, 2017; Chaudhari and Soatto, 2018], we incorporate very little knowledge of the learning rule into our theory. The result is that our theory is perhaps more general than required. Interestingly, our results still hold if we perform (stochastic) gradient *ascent* on the weights. We believe our analysis would benefit from including more knowledge of the update rule.

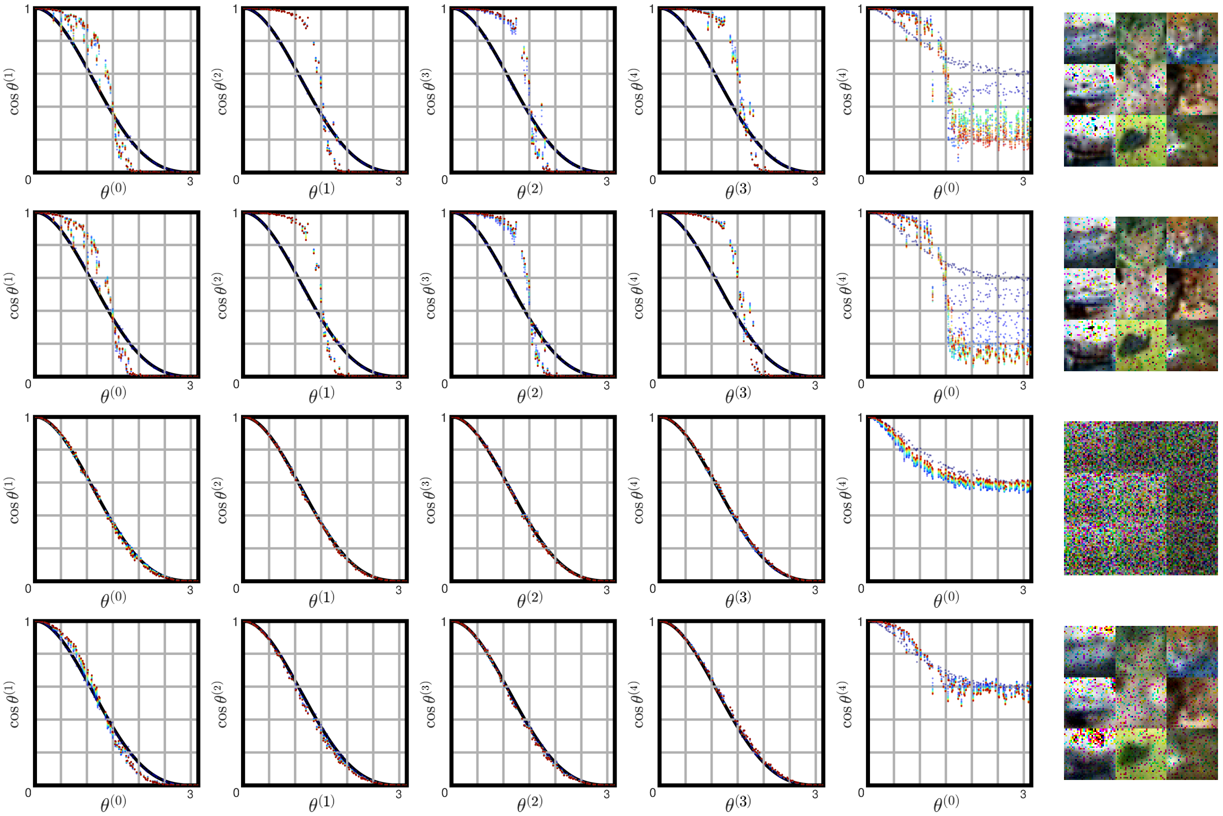

Figure 4: As in Figure 3, but for inputs with non-zero sums as outlined in §5.2.

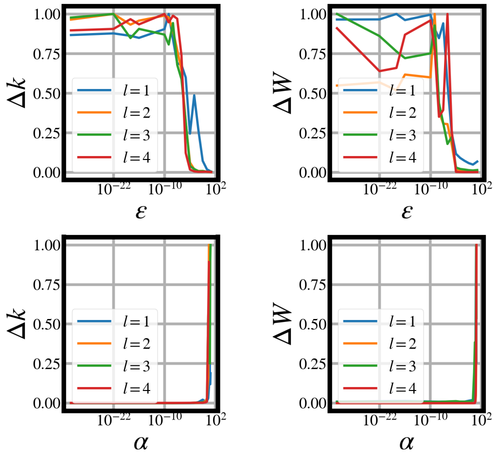

Figure 5: Δk: MSE of kernel to normalized arc-cosine kernel normalized to between 0 and 1. $\Delta W : (E^{n^{(l)}})^2$ normalized to between 0 and 1. Top: Adam using step size 0.001, $\beta_1 = 0.9$, $\beta_2 = 0.999$ varying ε. Bottom: SGD varying α.

Jacot *et al.* [2018] analyse a continuous-time approximation of (not stochastic) gradient descent. In this dynamic, it is shown that MLPs in function space follow a *linear* differential equation in an infinite width limit. A central object of their study is a positive-definite kernel, the neural tangent kernel, which is shown to stay approximately constant during training. Also utilizing a kernel and working in function space, Du *et al.* [2019] bound distances between functions trained in continuous and discrete time dynamics in special cases and investigate the surprising fact that certain models can achieve zero training loss. Chizat and Bach [2018] present a unified framework of these and other works under a training regime called *lazy training*. We remark that it is easy to find networks trained with Adam that do *not* exhibit approximately constant kernels (for example, see the first row of Figure 4). These networks seem to out-perform those that do have constant kernels during training. This is consistent with the view expressed by Chizat and Bach [2018], that "*the most competitive neural networks are not trained in this regime [lazy training].*"

In future work, we would like to examine the ESD of weight and Hessian matrices without normality assumptions as in previous works [Pennington and Bahri, 2017; Pennington and Worah, 2017], perhaps using results concerning the ESD of exchangeable random matrices [Chatterjee, 2006; Adamczak *et al.*, 2016].

Acknowledgements

Russell Tsuchida and Fred Roosta gratefully acknowledge the generous support given by the Australian Research Council Centre of Excellence for Mathematical & Statistical Frontiers (ACEMS). Fred Roosta was partially supported by the Australian Research Council through a Discovery Early Career Researcher Award (DE180100923).

References

[Adamczak *et al.*, 2016] R. Adamczak, D. Chafaï, and P. Wolff. Circular law for random matrices with exchangeable entries. *Random Structures & Algorithms*, 48(3):454–479, 2016.

[Aldous, 1981] D.J. Aldous. Representations for partially exchangeable arrays of random variables. *Journal of Multivariate Analysis*, 11(4):581–598, 1981.

[Bach, 2017] F. Bach. Breaking the curse of dimensionality with convex neural networks. *Journal of Machine Learning Research*, 18(19):1–53, 2017.

[Bietti and Mairal, 2017] A. Bietti and J. Mairal. Invariance and stability of deep convolutional representations. In *Advances in Neural Information Processing Systems*, pages 6210–6220, 2017.

[Chatterjee, 2006] S. Chatterjee. A generalization of the Lindeberg principle. *The Annals of Probability*, 34(6):2061–2076, 2006.

[Chaudhari and Soatto, 2018] P. Chaudhari and S. Soatto. Stochastic gradient descent performs variational inference, converges to limit cycles for deep networks. In *International Conference on Learning Representations*, 2018.

[Chizat and Bach, 2018] L. Chizat and F. Bach. A note on lazy training in supervised differentiable programming. Technical report, 2018.

[Cho and Saul, 2009] Y. Cho and L.K. Saul. Kernel methods for deep learning. In *Advances in Neural Information Processing Systems*, pages 342–350, 2009.

[Daniely *et al.*, 2016] A. Daniely, R. Frostig, and Y. Singer. Toward deeper understanding of neural networks: The power of initialization and a dual view on expressivity. In *Advances In Neural Information Processing Systems*, pages 2253–2261, 2016.

[Dozat, 2016] T. Dozat. Incorporating nesterov momentum into adam. Technical report, 2016.

[Du *et al.*, 2019] S.S. Du, X. Zhai, B. Poczos, and A. Singh. Gradient descent provably optimizes over-parameterized neural networks. *International Conference on Learning Representations*, 2019.

[He *et al.*, 2015] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *International Conference on Computer Vision*, pages 1026–1034, 2015.

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

[Kallenberg, 2006] O. Kallenberg. *Probabilistic symmetries and invariance principles*. Springer Science & Business Media, 2006.

[Khinchin, 1949] A.I. Khinchin. *Mathematical foundations of statistical mechanics*. Courier Corporation, 1949.

[Kingma and Ba, 2015] D.P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

[Krizhevsky and Hinton, 2009] A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images. Technical report, 2009.

[Kurth, 2014] R. Kurth. *Axiomatics of classical statistical mechanics*. Elsevier, 2014.

[MacKay, 1992] D.J.C. MacKay. A practical Bayesian framework for backpropagation networks. *Neural Computation*, 4(3):448–472, 1992.

[Martin and Mahoney, 2017] C.H. Martin and M.W. Mahoney. Rethinking generalization requires revisiting old ideas: statistical mechanics approaches and complex learning behavior. *arXiv preprint arXiv:1710.09553*, 2017.

[Moore, 2015] C.C. Moore. Ergodic theorem, ergodic theory, and statistical mechanics. *Proceedings of the National Academy of Sciences*, 112(7):1907–1911, 2015.

[Neal, 1994] R.M. Neal. *Bayesian Learning for Neural Networks*. PhD thesis, University of Toronto, 1994.

[Pennington and Bahri, 2017] J. Pennington and Y. Bahri. Geometry of neural network loss surfaces via random matrix theory. In *International Conference on Machine Learning*, pages 2798–2806, 2017.

[Pennington and Worah, 2017] J. Pennington and P. Worah. Nonlinear random matrix theory for deep learning. In *Advances in Neural Information Processing Systems*, pages 2634–2643, 2017.

[Poole *et al.*, 2016] B. Poole, S. Lahiri, M. Raghu, J. Sohl-Dickstein, and S. Ganguli. Exponential expressivity in deep neural networks through transient chaos. In *Advances in Neural Information Processing Systems*, pages 3360–3368, 2016.

[Raghu *et al.*, 2017] M. Raghu, B. Poole, J. Kleinberg, S. Ganguli, and J. Sohl-Dickstein. On the expressive power of deep neural networks. In *International Conference on Machine Learning*, pages 2847–2854, 2017.

[Royden and Fitzpatrick, 2010] H.L. Royden and P. Fitzpatrick. *Real Analysis*. Prentice Hall, 2010.

[Seung *et al.*, 1992] H.S. Seung, H. Sompolinsky, and N. Tishby. Statistical mechanics of learning from examples. *Physical Review A*, 45(8):6056, 1992.

[Tsuchida *et al.*, 2018] R. Tsuchida, F. Roosta-Khorasani, and M. Gallagher. Invariance of weight distributions in rectified MLPs. *International Conference on Machine Learning*, 2018.

[Watkin *et al.*, 1993] T.L.H. Watkin, A. Rau, and M. Biehl. The statistical mechanics of learning a rule. *Reviews of Modern Physics*, 65(2):499, 1993.