Supplementary Material for Dynamics of learning in deep linear neural networks

Andrew M. Saxe (asaxe@stanford.edu)
Department of Electrical Engineering

James L. McClelland (mcclelland@stanford.edu)
Department of Psychology

Surya Ganguli (sganguli@stanford.edu)
Department of Applied Physics
Stanford University, Stanford, CA 94305 USA

1 Hyberbolic dynamics of learning

In Section 1.3 of the main text we treat the dynamics of learning in three layer networks where mode strengths in each layer are equal, i.e, $a = b$, a reasonable limit when starting with small random initial conditions. More generally, though, we are interested in how long it takes for ab to approach s from any given initial condition. To access this, given the hyperbolic nature of the dynamics, it is useful to make the hyperbolic change of coordinates,

$$a = \sqrt{c_0}\cosh\frac{\theta}{2} \quad b = \sqrt{c_0}\sinh\frac{\theta}{2} \quad \text{for } a^2 > b^2 \tag{1}$$

$$a = \sqrt{c_0}\sinh\frac{\theta}{2} \quad b = \sqrt{c_0}\cosh\frac{\theta}{2} \quad \text{for } a^2 < b^2. \tag{2}$$

Thus θ parametrizes the dynamically invariant manifolds $a^2 - b^2 = \pm c_0$. For any c_0 and θ, this coordinate system covers the region $a + b > 0$, which is the basin of attraction of the upper right component of the hyperbola $ab = s$. A symmetric situation exists for $a + b < 0$, which is attracted to the lower left component of $ab = s$. We use θ as a coordinate to follow the dynamics of the product ab, and using the relations $ab = c_0 \sinh\theta$ and $a^2 + b^2 = c_0 \cosh\theta$, we obtain

$$\tau\frac{d\theta}{dt} = s - c_0\sinh\theta. \tag{3}$$

This differential equation is separable in θ and t and can be integrated to yield

$$t = \tau\int_{\theta_0}^{\theta_f}\frac{d\theta}{s - c_0\sinh\theta} = \frac{\tau}{\sqrt{c_0^2 + s^2}}\left[\ln\frac{\sqrt{c_0^2 + s^2} + c_0 + s\tanh\frac{\theta}{2}}{\sqrt{c_0^2 + s^2} - c_0 - s\tanh\frac{\theta}{2}}\right]_{\theta_0}^{\theta_f}. \tag{4}$$

Here t is the amount of time it takes to travel from θ_0 to θ_f along the hyperbola $a^2 - b^2 = \pm c_0$. The fixed point lies at $\theta = \sinh^{-1} s/c_0$, but the dynamics cannot reach the fixed point in finite time. Therefore we

introduce a cutoff ϵ to mark the endpoint of learning, so that θ_f obeys $\sinh\theta_f = (1-\epsilon)s/c_0$ (i.e. ab is close to s by a factor $1-\epsilon$). We can then average over the initial conditions c_0 and θ_0 to obtain the expected learning time of an input-output relation that has a correlation strength s. Rather than doing this, it is easier to obtain a rough estimate of the timescale of learning under the assumption that the initial weights are small, so that c_0 and θ_0 are close to 0. In this case $t = O(\tau/s)$ (with a weak logarithmic dependence on the cutoff (i.e. $\ln(1/\epsilon)$). This modestly generalizes the result given in the main text: the timescale of learning of each input-output mode α of the correlation matrix Σ^{31} is inversely proportional to the correlation strength s_α of the mode even when a and b differ slightly, i.e., c_0 small. This is not an unreasonable limit for random initial conditions because $|c_0| = |a \cdot a - b \cdot b|$ where a and b are random vectors of N_2 synaptic weights into and out of the hidden units. Thus we expect the lengths of the two random vectors to be approximately equal and therefore c_0 will be small relative to the length of each vector.

2 Optimal discrete time learning rates

In Section 2 we state results on the optimal learning rate as a function of depth in a deep linear network, which we derive here. Starting from the decoupled initial conditions given in the main text, the dynamics arise from gradient descent on

$$E(a_1, \cdots, a_{N_l-1}) = \frac{1}{2\tau}\left(s - \prod_{k=1}^{N_l-1} a_k\right). \tag{5}$$

Hence for each a_i we have

$$\frac{\partial E}{\partial a_i} = -\frac{1}{\tau}\left(s - \prod_{k=1}^{N_l-1} a_k\right)\left(\prod_{k\neq i}^{N_l-1} a_k\right) \equiv f(a_i) \tag{6}$$

The elements of the Hessian are thus

$$\begin{aligned} \frac{\partial^2 E}{\partial a_i a_j} &= \frac{1}{\tau}\left(\prod_{k\neq j}^{N_l-1} a_k\right)\left(\prod_{k\neq i}^{N_l-1} a_k\right) - \frac{1}{\tau}\left(s - \prod_{k=1}^{N_l-1} a_k\right)\left(\prod_{k\neq i,j}^{N_l-1} a_k\right) &(7)\\ &\equiv g(a_i, a_j) &(8)\end{aligned}$$

for $i \neq j$, and

$$\frac{\partial^2 E}{\partial a_i^2} = \frac{1}{\tau}\left(\prod_{k\neq i}^{N_l-1} a_k\right)^2 \equiv h(a_i) \tag{9}$$

for $i = j$.

We now assume that we start on the symmetric manifold, such that $a_i = a_j = a$ for all i, j. Thus we have

$$\begin{aligned} E(a) &= \frac{1}{2\tau}\left(s - a^{N_l-1}\right), &(10)\\ f(a) &= -\frac{1}{\tau}\left(s - a^{N_l-1}\right)a^{N_l-2}, &(11)\\ g(a) &= \frac{2}{\tau}a^{2N_l-4} - \frac{1}{\tau}sa^{N_l-3} &(12)\\ h(a) &= \frac{1}{\tau}a^{2N_l-4} &(13)\end{aligned}$$

The Hessian is

$$H(a) = \begin{bmatrix} h & g & \cdots & g & g \\ g & h & \cdots & g & g \\ \vdots & & \ddots & & \vdots \\ g & g & \cdots & h & g \\ g & g & \cdots & g & h \end{bmatrix}. \tag{14}$$

One eigenvector is $v_1 = [11\cdots1]^T$ with eigenvalue $\lambda_1 = h + (N_l - 2)g$, or

$$\lambda_1 = (2N_l - 3)\frac{1}{\tau}a^{2N_l-4} - (N_l - 2)\frac{1}{\tau}sa^{N_l-3}. \tag{15}$$

Now consider the second order update (Newton-Raphson) (here we use 1 to denote a vector of ones)

$$\begin{aligned} a^{t+1}1 &= a^t 1 - H^{-1}f(a^t)1 && (16)\\ &= a^t 1 - f(a^t)H^{-1}1 && (17)\\ a^{t+1} &= a^t - f(a^t)/\lambda_1(a^t) && (18) \end{aligned}$$

Note that the basin of attraction does not include small initial conditions, because for small a the Hessian is not positive definite.

To determine the optimal learning rate for first order gradient descent, we compute the maximum of λ_1 over the range of mode strengths that can be visited during learning, i.e., $a \in [0, s^{1/(N_l-1)}]$. This occurs at the optimum, $a_{opt} = s^{1/(N_l-1)}$. Hence substituting this into (15) we have

$$\lambda_1(a_{opt}) = (N_l - 1)\frac{1}{\tau}s^{\frac{2N_l-4}{N_l-1}}. \tag{19}$$

The optimal learning rate α is proportional to $1/\lambda_1(a_{opt})$, and hence scales as

$$\alpha \sim O\left(\frac{1}{N_l s^2}\right) \tag{20}$$

for large N_l.

2.1 Learning speeds with optimized learning rate

How does the optimal learning rate impact learning speeds? We compare the three layer learning time to the infinite depth learning time, with learning rate set inversely proportional to Eqn. (19) with proportionality constant c.

This yields a three layer learning time t_3 of

$$t_3 = c\ln\frac{u_f(s-u_0)}{u_0(s-u_f)} \tag{21}$$

and an infinite layer learning time t_∞ of

$$t_\infty = c\left[\log\left(\frac{u_f(u_0-s)}{u_0(u_f-s)}\right) + \frac{s}{u_0} - \frac{s}{u_f}\right], \tag{22}$$

Hence the difference is

$$t_\infty - t_3 = \frac{cs}{u_0} - \frac{cs}{u_f} \approx \frac{cs}{\epsilon} \tag{23}$$

where the final approximation is for $u_0 = \epsilon, u_f = s - \epsilon$, and ϵ small. Thus very deep networks incur only a finite delay relative to shallow networks.