1 The Problem

We consider a feedforward neural network with a single hidden layer and activation function σ. It receives an input $x \in \mathbb{R}^n$ and produces a scalar output $\hat{y} \in \mathbb{R}$. The hidden layer has K units. The weights for the first and second layer are $W_1 \in \mathbb{R}^{K\times n}$ and $W_2 \in \mathbb{R}^{1\times K}$, respectively, and the corresponding biases are $b_1 \in \mathbb{R}^K$ and $b_2 \in \mathbb{R}$.

$$\hat{y} = W_2\sigma(W_1x + b_1) + b_2. \tag{1}$$

Our data x are sampled from a mixture of two translation-invariant distributions in some family $\{p_\xi\}_\xi$ parameterized by a correlation length-scale ξ. That is, we sample $x \sim p_{\xi_1}$ with probability $\frac{1}{2}$ and $x \sim p_{\xi_2}$ otherwise. If x is sampled from p_{ξ_1}, then $y(x) = 1$; otherwise, $y(x) = 0$. We can train using either mean-squared error or cross-entropy loss, though we primarily consider the former.

Alessandro's paper primarily considers the case where $W_2 = \frac{1}{K}\mathbf{1}^\top$ (take the mean of the hidden activations) is fixed and $\sigma(h) = \mathrm{erf}(\frac{h}{\sqrt{2}})$. I have also tried $\sigma = \mathrm{sigmoid}, \mathrm{ReLU}$. For the former, the results are qualitatively identical, while for the latter we get localization if $\xi_1 > \xi_2$ and short-range oscillations otherwise. For $\sigma = \mathrm{ReLU}$, one can further remove the bias terms b_1 and b_2 (though not for sigmoid).

We consider two types of datasets: the nonlinear Gaussian process (NLGP) and the single pulse (SP). We explain them in more detail later. They differ primarily in that the former has continuous support on $\mathbb{R}^n$, while the latter has discrete support on a subset of $\{0, 1\}^n$. The former also has a gain parameter that controls the degree of localization, while the latter does not.

Motivated by localization with the ReLU model, we tried to see how well a gated deep linear network (GDLN) could do. Next, we'll provide some experimental support for why this might work, and then do some analysis. We'll conclude with some questions, concerns, and ideas.

2 Have You Tried Making It Linear?

Gating lets us decompose the ReLU post-activation in terms of the pre-activation's sign and magnitude.

$$\mathrm{ReLU}(\langle w_1(t), x\rangle) = g(t, x)\langle w_1(t), x\rangle \qquad \text{where} \qquad g(t, x) = \mathbb{1}(\langle w_1(t), x\rangle \geq 0). \tag{2}$$

We generally assume that g does not vary during learning, even though w_1 may. Later on, we'll try to analyze what happens when this does not hold.

To assess the validity of this assumption, we need to see how much $g(x)$, as defined above, changes during learning. Additionally, post-hoc, we can usually pick a somewhat sensible gating structure that mimics a specific run's behavior. But we'd like to be able to determine this gating upfront. We explore all this in the following subsections.

2.1 Sign Flipping

Note that g is invariant to the scale of w_1. We've observed in previous experiments that w_1 appears to grow uniformly in size during much of its training. (There is, importantly, a phase where it goes from Gaussian to non-Gaussian, but the localization seems to be more likely to occur around its mode.) This suggests that

$g(x)$ may be relatively constant during learning. If this is so, then it would be reasonable to try using a standard GDLN to model the ReLU network.

We will model the ReLU network as in equation (2), focusing on how g varies with time for each hidden neuron. We will look at the metrics

$$p(t) = \mathbb{P}_x(g(t,x) = g(t+\delta t, x)) \qquad \text{for all } t \tag{3}$$

$$p_{\text{unif}} = \mathbb{P}_x(\{g(t,x) = g(t',x)\ \forall t, t'\}) \tag{4}$$

2.2 Predicting Loca(liza)tion

2.3 Evolving Gates?

3 Let's Consider a Single Layer with Linear Activation...

3.1 Model

Our GDLN model is defined as follows:

$$\hat{y}(x) = \frac{1}{K}\left(\sum_{k\in[K]} g_k(x) w_k^\top\right) x, \tag{5}$$

where g_k are (node) gates, and $w_k \in \mathbb{R}^n$ are the rows of the first-layer weight matrix $W_1 \in \mathbb{R}^{K\times n}$. That is,

$$W_1 = \begin{pmatrix} w_1^\top \\ \vdots \\ w_K^\top \end{pmatrix} \tag{6}$$

3.2 Gradient Flow

Recalling the GDLN paper, the gradient flow for w_1 is given by

$$\tau \frac{d}{dt} w_1^\top = \frac{1}{K}\left[\Sigma^{yx}(p_1) - \sum_{k\in[K]} w_k^\top \Sigma^{xx}(p_1, p_k)\right], \tag{7}$$

where

$$\Sigma^{yx}(p_i) = \left\langle g_i y x^\top \right\rangle_{g,x,y} \tag{8}$$

$$\Sigma^{xx}(p_i, p_j) = \left\langle g_i g_j x x^\top \right\rangle_{g,x}. \tag{9}$$

3.3 General Case

Let us relabel $b_i = \Sigma^{yx}(p_i)^\top$ and $A_{ij} = \Sigma^{xx}(p_i, p_j)$. Note that A_{ij} is symmetric and $A_{ij} = A_{ji}$. Then, we can write the gradient flow for all weights as

$$K\tau \frac{d}{dt} \underbrace{\begin{bmatrix} w_1 \\ \vdots \\ w_K \end{bmatrix}}_{w\in\mathbb{R}^{Kn}} = \underbrace{\begin{bmatrix} b_1 \\ \vdots \\ b_k \end{bmatrix}}_{b\in\mathbb{R}^{Kn}} - \underbrace{\begin{bmatrix} A_{11} & \cdots & A_{1K} \\ \vdots & \ddots & \vdots \\ A_{K1} & \cdots & A_{KK} \end{bmatrix}}_{A\in\mathbb{R}^{Kn\times Kn}} \begin{bmatrix} w_1 \\ \vdots \\ w_K \end{bmatrix}. \tag{10}$$

Observe that w is the vectorized form of our $K \times n$ first-layer weight matrix. Note also that A is a symmetric real matrix, so we can diagonalize it as $A = P\Lambda P^\top$, where the columns of P are the eigenvectors of A and the diagonal entries of Λ are the corresponding (nonnegative) eigenvalues. (It is symmetric because A is block symmetric with blocks A_{ij}, and the blocks are also symmetric.) To see this more clearly, let us write

$$\tilde{g}(x) = \begin{bmatrix} g_1(x) \\ \vdots \\ g_K(x) \end{bmatrix}. \tag{11}$$

Then,

$$A = \left\langle (\tilde{g} \otimes x)(\tilde{g} \otimes x)^\top \right\rangle_{g,x}, \tag{12}$$

which is clearly a symmetric matrix. (Interjection: Whatever distribution we have over g should satisfy that $\tilde{g} \sim \Pi\tilde{g}$, where Π is some permutation matrix on K elements. That is, the distribution should be invariant to the ordering of the gates, since this is what we want empirically.)

We can reparameterize in terms of $u = P^\top w$ and $c = P^\top b$.

$$K\tau \frac{d}{dt} u = -\Lambda u + c \implies u(t) = \Lambda^{-1} e^{-\frac{t}{K\tau}\Lambda + C}\mathbf{1} + \Lambda^{-1} c, \tag{13}$$

where C is a constant diagonal matrix that defines the initial condition. So,

$$w(t) = P\Lambda^{-1} e^{-\frac{t}{K\tau}\Lambda + C}\mathbf{1} + P\Lambda^{-1} c \tag{14}$$

$$= A^{-1} P e^{-\frac{t}{K\tau}\Lambda + C}\mathbf{1} + A^{-1} b. \tag{15}$$

3.4 Winning Gating Structure

Can we read off the winning gating structure from the gradient flow? For simplicity, let us assume we are sampling uniformly from a finite set of G gates, $\{g\}$. Then, we can write equation (10) as

$$K\tau \frac{d}{dt} w = \frac{1}{G} \sum_{g \in \{g\}} \left[b_{x,y|g} - A_{x,y|g} w \right], \tag{16}$$

where the subscript on b and A indicates the conditioning on a specific gating structure g. Intuitively, a gating structure that minimizes the norm of A will shrink the slowest. This is somewhat equivalent to minimizing the eigenvalues of A, since they are all nonnegative. (What happens if an eigenvalue is zero?)

Let us consider a single block in A:

$$A_{ij} = \langle g_i g_j x x^\top \rangle_{g,x} = \mathbb{P}(g_i = 1, g_j = 1) \langle x x^\top \rangle_{x|g_i = g_j = 1}. \tag{17}$$

Let us quickly ask: Does what we observe empirically match this intuition? We see that receptive fields come in pairs and tile the space.

TODO: empirically look at dominating eigenvalues for finite case!

3.4.1 Early Dynamics

For small t, but sufficiently large to see separation among different eigenvalues, can we predict the leading structure?

3.4.2 Limiting Behavior

If none of the eigenvalues are zero, then $w(\infty) = A^{-1}b$. If we write

$$\tilde{x} = \begin{bmatrix} g_1(x)x \\ \vdots \\ g_K(x)x \end{bmatrix} \in \mathbb{R}^{Kn}, \tag{18}$$

then $A = \langle \tilde{x}\tilde{x}^\top \rangle_{x,y,g}$ and $b = \langle \tilde{x}y \rangle_{x,y,g}$. Then, it is clear that this is the population solution to the OLS problem of regressing y on $\tilde{x}$, averaging across the distributions of the data *and* the gating architectures.

In this context, one might ask, which gating structure minimizes the MSE loss? The loss is

$$\begin{aligned}
\mathcal{L}_{OLS} &= \left\langle \left(\tilde{x}'^\top (\langle \tilde{x}\tilde{x}^\top \rangle_{x,y,g})^{-1} \langle \tilde{x}y \rangle_{x,y,g} - y' \right)^2 \right\rangle_{x',y',g'} && (19) \\
&= \left\langle (\tilde{x}'^\top (\langle \tilde{x}\tilde{x}^\top \rangle_{x,y,g})^{-1} \langle \tilde{x}y \rangle_{x,y,g})^2 - 2(y'\tilde{x}'^\top (\langle \tilde{x}\tilde{x}^\top \rangle_{x,y,g})^{-1} \langle \tilde{x}y \rangle_{x,y,g}) + (y')^2 \right\rangle_{x',y',g'} && (20) \\
&= \frac{1}{2} - \langle y\tilde{x}^\top \rangle_{x,y,g} (\langle \tilde{x}\tilde{x}^\top \rangle_{x,y,g})^{-1} \langle \tilde{x}y \rangle_{x,y,g} && (21) \\
&= \frac{1}{2} - \frac{1}{2} \langle \tilde{x} \rangle_{x,g|y=1}^\top (\langle \tilde{x}\tilde{x}^\top \rangle_{x,g|y=1} + \langle \tilde{x}\tilde{x}^\top \rangle_{x,g|y=0})^{-1} \langle \tilde{x} \rangle_{x,g|y=1}. && (22)
\end{aligned}$$

In the final step, we assumed (WLOG) that the negative class is $y = 0$ and the positive class is $y = 1$. (Throughout, we also assume that the classes are balanced.) The question is: For fixed p_{ξ_1} and p_{ξ_2}, how do we choose the gates g_k to minimize equation (22)?

It may be useful to write this in terms of Kronecker products. Let

$$\tilde{g}(x) = \begin{bmatrix} g_1(x) \\ \vdots \\ g_K(x) \end{bmatrix} \in \{0,1\}^K. \tag{23}$$

Then, minimizing equation (22) is equivalent to maximizing

$$\mathcal{L}^*(\tilde{g}) = \langle \tilde{g} \otimes x \rangle_{x,g|y=1}^\top (\langle (\tilde{g} \otimes x)(\tilde{g} \otimes x)^\top \rangle_{x,g|y=1} + \langle (\tilde{g} \otimes x)(\tilde{g} \otimes x)^\top \rangle_{x,g|y=0})^{-1} \langle \tilde{g} \otimes x \rangle_{x,g|y=1} \tag{24}$$

over $\tilde{g} : \mathrm{supp}(p_{\xi_1}) \cup \mathrm{supp}(p_{\xi_2}) \to \{0,1\}^K$. *I will have to think more about this.*

After a bit more thinking... I think that the best precision matrix would be maximally diagonal (no clue if this is actually true! but maybe it holds empirically?). For Gaussian data (at least), this mean that the blocks are independent conditioned on all the other blocks. Gates that tile the space without overlap would achieve this (I think?). But tbh I haven't got the slightest clue!!

3.5 Exclusive Gates

Let us assume that the gates are exclusive, that is, only one gate is active at a time. Then, $\Sigma^{xx}(p,q) = 0$ for $p \neq q$.

Then A becomes block diagonal. We can write the gradient flow for w_1 as

$$K\tau \frac{d}{dt} w_1 = -A_{11} w_1 + b_1. \tag{25}$$

Note that $A_{11} = \Sigma^{xx}(p_1, p_1)$ is always symmetric (and real). So, we can diagonalize it as $A_{11} = P\Lambda P^\top$, where the columns of P are $v_1, \dots, v_n$ and the diagonal entries of Λ are $\lambda_1, \dots, \lambda_n$. Let us introduce $u_1 = P^\top w_1$ and $c_1 = P^\top b_1$. Then,

$$K\tau \frac{d}{dt} u_1 = -\Lambda u_1 + c_1. \tag{26}$$

This ODE is solved by

$$u_1(t) = \Lambda^{-1} e^{-\frac{t}{K\tau}\Lambda + C}\mathbf{1} + \Lambda^{-1} c_1, \tag{27}$$

where C is a constant diagonal matrix that defines the initial condition. Then,

$$w_1(t) = P\left(\Lambda^{-1} e^{-\frac{t}{K\tau}\Lambda + C}\mathbf{1} + \Lambda^{-1} c\right) \tag{28}$$

$$= P\Lambda^{-1} e^{-\frac{t}{K\tau}\Lambda + C}\mathbf{1} + P\Lambda^{-1} c_1 \tag{29}$$

$$= A_{11}^{-1}\left(P e^{-\frac{t}{K\tau}\Lambda + C}\mathbf{1} + b_1\right). \tag{30}$$

Recalling A_{11} and b_1,

$$w_1(t) = (\Sigma^{xx}(p_1, p_1))^{-1}\left(P e^{-\frac{t}{\tau}\Lambda + C}\mathbf{1} + \Sigma^{yx}(p_1)^\top\right). \tag{31}$$

So,

$$w_1(\infty) = (\Sigma^{xx}(p_1, p_1))^{-1}\left(P e^{-\frac{t}{\tau}\Lambda + C}\mathbf{1} + \Sigma^{yx}(p_1)^\top\right). \tag{32}$$

As with above, this is the population solution to OLS, $(X^\top X)^{-1} X^\top y = (\langle xx^\top \rangle)^{-1}(\langle xy \rangle)$.

So, each weight matrix converges to the OLS solution on the subset of the data determined by its gate.

3.6 Redundant Gates

What if $g_1 = g_2$? Then, $b_1 = b_2$ and $A_{11} = A_{12} = A_{22}$. So,

$$K\tau \frac{d}{dt} w_1 = -A_{11}(w_1 + w_2) + b_1, \tag{33}$$

$$K\tau \frac{d}{dt} w_2 = -A_{11}(w_1 + w_2) + b_1. \tag{34}$$

Clearly, then, $w_1 - w_2$ is a constant vector. Moreover, $\frac{1}{2}(w_1 + w_2)$ evolves according to

$$\frac{K\tau}{2}\frac{d}{dt}(w_1 + w_2) = -A_{11}(w_1 + w_2) + b_1. \tag{35}$$

Writing $2\Delta = w_1 - w_2$ and $w_1 + w_2 = 2(w_1 - \Delta)$, we have

$$K\tau \frac{d}{dt} w_1 = K\tau \frac{d}{dt}(w_1 - \Delta) = -2A_{11}(w_1 - \Delta) + b_1. \tag{36}$$

We can plug this into our solution from the previous section to get

$$w_1(t) = \frac{1}{2} A_{11}^{-1}\left(P e^{-\frac{t}{K\tau}\Lambda + C}\mathbf{1} + b_1\right). \tag{37}$$

So,

$$w_2(t) = \frac{1}{2} A_{11}^{-1}\left(P e^{-\frac{t}{K\tau}\Lambda + C}\mathbf{1} + b_1\right) - (w_1(0) - w_2(0)). \tag{38}$$

4 Theory-driven Experiments

4.1 General Case

4.2 Single Gate

4.3 Redundant Gates

5 Next Steps

Concerns:

Questions:

Some ideas:

1. Let the gates vary during training. That is, g depends on t and x. So, Σ^{yx} and Σ^{xx} will depend on t as well. We can still solve the differential equation, and we may be able to use a single basis to diagonalize them across time (Toeplitz/circulant properties? DFT?).

5.1 Time-dependent Gating

Let us now consider the case where g depends on x as well as t. That is, $g = g(t, x)$. Then, Σ^{yx} and Σ^{xx} will depend on t as well. In terms of equation (10), b and A depend on t rather than being fixed. That is,

$$K\tau \frac{d}{dt} w = b(t) - A(t)w. \tag{39}$$

Note that $A(t)$ is always real symmetric for each t. Let us momentarily assume that it is diagonalizable in the same basis across time, that is $A(t) = P\Lambda(t)P^\top$, where $\Lambda(t)$ is diagonal.

When is $A(t)$ diagonalizable in the same basis across time? Recall that two matrices are simultaneously diagonalize iff they commute. Let us consider two time points, t and t'.

$$\begin{aligned}
A(t)A(t') &= \left[\langle g_i(t,x) g_j(t,x) x x^\top \rangle_{x,y,g}\right]_{ij} \left[\langle g_i(t,x') g_j(t,x') x' x'^\top \rangle_{x',y',g}\right]_{jk} && (40)\\
&= \left[\textstyle\sum_l \langle g_i(t,x) g_l(t,x) x x^\top \rangle_{x,y,g} \langle g_l(t',x') g_j(t',x') x' x'^\top \rangle_{x',y',g}\right]_{ij} && (41)\\
&= \left[\textstyle\sum_l \left\langle g_i(t,x) g_l(t,x) x x^\top g_l(t',x') g_j(t',x') x' x'^\top \right\rangle_{x,y,x',y',g}\right]_{ij} && (42)\\
&= \left[\left\langle g_i(t,x) g_j(t',x') \textstyle\sum_l g_l(t,x) g_l(t',x') x x^\top x' x'^\top \right\rangle_{x,y,x',y',g}\right]_{ij} && (43)
\end{aligned}$$

5.2 Single step in a ReLU Network

Let's consider what happens in a single step of a network with ReLU activation. We make predictions with

$$\hat{y}(x) = \frac{1}{K} \sum_{k \in [K]} \mathrm{ReLU}(\langle w_k, x \rangle). \tag{44}$$

We consider MSE loss,

$$\mathcal{L} = \frac{1}{2} \mathbb{E}_{X,Y} \left[(\hat{y}(X) - Y)^2 \right]. \tag{45}$$

The gradient flow for w_1 is given by

$$\tau \frac{d}{dt} w_1 = -\mathbb{E}_{X,Y}\left[(\mathrm{ReLU}(\langle w_1(t), X\rangle) - Y) \frac{\partial}{\partial w_1} [\mathrm{ReLU}(\langle w_1(t), X\rangle)] \right] \tag{46}$$

$$= \big(\underbrace{\mathbb{E}_{X,Y|\langle w_1(t),X\rangle>0}[YX]}_{\equiv (I)} - \underbrace{\mathbb{E}_{X,Y|\langle w_1(t),X\rangle>0}[\langle w_1(t), X\rangle X]}_{\equiv (II)} \big) \, \mathbb{P}(\langle w_1(t), X\rangle > 0). \tag{47}$$

Recall that X is a mixture of $X \mid Y = 1$ and $X \mid Y = 0$. So, we will compute these expectations separately. Let us write $S = \langle w_1(t), X\rangle$. Then, we can use the law of total expectation to write

$$\mathbb{E}_{X|Y=1,\langle w_1(t),X\rangle>0}[f(X)] = \mathbb{E}_{S>0|y=1}\left[\mathbb{E}_{X|S,y=1}[f(X)] \right]. \tag{48}$$

So, we need to find the distribution of X conditioned on $S \equiv \langle w_1(t), X\rangle = s$. In general, this is very challenging. We will split this into two terms, one of which disappears when X is Gaussian. Let Σ be the covariance of x (recall it has mean 0). We write

$$X = AX + Sv, \tag{49}$$

where

$$v = \frac{1}{w_1(t)^\top \Sigma w_1(t)} \Sigma w_1(t), \tag{50}$$

$$A = I_n - v w_1(t)^\top. \tag{51}$$

Thus, $\mathbb{E}_{X|S=s}[X] = \mathbb{E}_{X|S=s}[AX] + sv$. Our choice of A and v implies that AX and S have zero *covariance* (see this post). When X is Gaussian, this implies that AX and S are independent, so we'd have $\mathbb{E}_{X|S=s}[X] = A\,\mathbb{E}[X] + sv = sv$.

With this representation,

$$\mathbb{E}_{X|Y=1,S>0}[YX] = \mathbb{E}_{S>0|Y=1}\left[\mathbb{E}_{X|S,Y=1}[AX] + Sv \right] \tag{52}$$

$$= \mathbb{E}_{X|Y=1,S>0}[AX] + \frac{\mathbb{E}_{S>0|Y=1}[S]}{w_1(t)^\top \Sigma_1 w_1(t)} \Sigma_1 w_1(t), \tag{53}$$

$$\mathbb{E}_{X|Y=1,S>0}[SX] = \mathbb{E}_{S>0|Y=1}\left[S\,\mathbb{E}_{X|S,Y=1}[AX] + S^2 v \right] \tag{54}$$

$$= \mathbb{E}_{X|Y=1,S>0}[SAX] + \mathbb{E}_{S>0|Y=1}\left[S^2\right] v \tag{55}$$

$$= \mathbb{E}_{X|Y=1,S>0}[SAX] + w_1(t)^\top \Sigma_1 w_1(t) v, \tag{56}$$

$$= \mathbb{E}_{X|Y=1,S>0}[SAX] + \Sigma_1 w_1(t). \tag{57}$$

Note that if X were Gaussian, then the first terms in equations (53) and (57) would be zero.

Now, we evaluate (I) and (II).

$$(I) = \mathbb{E}_{X|S>0}[SAX] + [\mathbb{P}(Y = 1 \mid S > 0)\Sigma_1 + \mathbb{P}(Y = 0 \mid S > 0)\Sigma_0]\, w_1(t), \tag{58}$$

$$(II) = \mathbb{E}_{X|S>0}[AX] + \mathbb{P}(Y = 1 \mid S > 0) \frac{\mathbb{E}_{S>0|Y=1}[S]}{w_1(t)^\top \Sigma_1 w_1(t)} \Sigma_1 w_1(t) \tag{59}$$

Then,

$$[(I) - (II)]\,\mathbb{P}(S > 0) \tag{60}$$

$$= -\mathbb{E}_{X|S>0}[(S-1)AX] - \left[\mathbb{P}(Y = 1 \mid S > 0) \left(\frac{\mathbb{E}_{S>0|Y=1}[S]}{w_1(t)^\top \Sigma_1 w_1(t)} - 1 \right) \Sigma_1 + \mathbb{P}(Y = 0 \mid S > 0) \frac{\mathbb{E}_{S>0|Y=0}[S]}{w_1(t)^\top \Sigma_0 w_1(t)} \Sigma_0 \right] w_1(t). \tag{61}$$

By symmetry, $\mathbb{P}(Y = 1 \mid S > 0) = \frac{1}{2}$ and $PR(S > 0) = \frac{1}{2}$. Then,

$$4\tau \frac{d}{dt} w_1 = -\mathbb{E}_{X|S>0}\left[(S-1)AX\right] - \left[\left(\frac{\mathbb{E}_{S>0|Y=1}\left[S\right]}{w_1(t)^\top \Sigma_1 w_1(t)} - 1\right)\Sigma_1 + \frac{\mathbb{E}_{S>0|Y=0}\left[S\right]}{w_1(t)^\top \Sigma_0 w_1(t)}\Sigma_0\right] w_1(t) \tag{62}$$

Also recall that Σ_1 and Σ_0 both diagonalize in the discrete Fourier basis, which we denote with P, and their corresponding diagonal matrices of eigenvalues Λ_1 and Λ_0. Write $u_1 = P^\top w_1$.

$$4\tau \frac{d}{dt} u_1 = -\mathbb{E}_{X|S>0}\left[(S-1)P^\top AX\right] - \left[\left(\frac{\mathbb{E}_{S>0|Y=1}\left[S\right]}{u_1(t)^\top \Lambda_1 u_1(t)} - 1\right)\Lambda_1 + \frac{\mathbb{E}_{S>0|Y=0}\left[S\right]}{u_1(t)^\top \Lambda_0 u_1(t)}\Lambda_0\right] u_1(t). \tag{63}$$

Let us expand the first term for $Y = 1$. Define $\Xi = P^\top X$.

$$\mathbb{E}_{X|Y=1,S>0}\left[(S-1)P^\top AX\right] = \mathbb{E}_{X|Y=1,S>0}\left[(S-1)P^\top\left(I_n - \frac{\Sigma_1 w_1(t) w_1(t)^\top}{w_1(t)^\top \Sigma_1 w_1(t)}\right)X\right] \tag{64}$$

$$= \mathbb{E}_{\Xi|Y=1,\langle u_1(t),\Xi\rangle>0}\left[(\langle u_1(t),\Xi\rangle - 1)\left(I_n - \frac{\Lambda_1 u_1(t) u_1(t)^\top}{u_1(t)^\top \Lambda_1 u_1(t)}\right)\Xi\right]. \tag{65}$$

Now, we must stop and ask: what is Ξ? Recall that P is the discrete Fourier basis. Importantly, it is actually the *real* part of the discrete Fourier basis since Σ_1 is symmetric. That is, with $\omega = e^{-\frac{2\pi i}{n}}$,

$$P_{:,j} = \Re\frac{1}{\sqrt{n}}\begin{bmatrix}1\\ \omega^j\\ \omega^{2j}\\ \vdots\\ \omega^{(n-1)j}\end{bmatrix} = \frac{1}{\sqrt{n}}\begin{bmatrix}1\\ \cos(\frac{2\pi}{n}j)\\ \cos(\frac{2\pi}{n}2j)\\ \vdots\\ \cos(\frac{2\pi}{n}(n-1)j)\end{bmatrix}. \tag{66}$$

Ξ is the discrete Fourier transform (DFT) of X.

TRY COMPUTING DENSITY OF NON-GAUSSIAN USING TRICK; DON'T THINK FOURIER APPROACH WILL BE VERY HELPFUL TBH.

5.2.1 Gaussian X

Now, let us assume that X is Gaussian. Then, $\mathbb{E}_{X|S>0}\left[(S-1)P^\top AX\right] = 0$. Furthermore,

$$\mathbb{E}_{S>0|Y=1}\left[S\right] = \sqrt{\frac{2}{\pi}}\left(w_1(t)^\top \Sigma_1 w_1(t)\right)^{\frac{1}{2}} = \sqrt{\frac{2}{\pi}}\left(u_1(t)^\top \Lambda_1 u_1(t)\right)^{\frac{1}{2}}. \tag{67}$$

Then, the gradient flow becomes

$$4\tau \frac{d}{dt} u_1 = -\left[\left(\frac{1}{\sqrt{u_1(t)^\top \Lambda_1 u_1(t)}} - 1\right)\Lambda_1 + \frac{1}{\sqrt{u_1(t)^\top \Lambda_0 u_1(t)}}\Lambda_0\right] u_1(t). \tag{68}$$

So,

$$\mathbb{E}_{S>0|y=1}\left[\mathbb{E}_{x|S,y=1}\left[x\right]\right] = \mathbb{E}_{S>0|y=1}\left[sv_1\right] = \mathbb{E}_{S>0|y=1}\left[s\right]v_1, \tag{69}$$

$$\mathbb{E}_{S>0|y=1}\left[\mathbb{E}_{x|S,y=1}\left[xx^\top\right]\right] = \mathbb{E}_{S>0|y=1}\left[A_1\Sigma_1 A_1^\top\right] = A_1\Sigma_1 A_1^\top. \tag{70}$$

Note that $S \sim \mathcal{N}\left(0, w_1(t)^\top \Sigma w_1(t)\right)$. So, $\mathbb{E}_{S>0|y=1}\left[s\right] = \left(\frac{2}{\pi} w_1(t)^\top \Sigma w_1(t)\right)^{\frac{1}{2}}$. In summary,

$$\mathbb{E}_{x|y=1,\langle w_1(t),x\rangle>0}\left[x\right] = \frac{\left(\frac{2}{\pi} w_1(t)^\top \Sigma w_1(t)\right)^{\frac{1}{2}}}{w_1(t)^\top \Sigma w_1(t)} \Sigma w_1(t) = \sqrt{\frac{2}{\pi}} \left(w_1(t)^\top \Sigma w_1(t)\right)^{-\frac{1}{2}} \Sigma w_1(t), \tag{71}$$

$$\mathbb{E}_{x|y=1,\langle w_1(t),x\rangle>0}\left[xx^\top\right] = A_1\Sigma_1 A_1^\top = \left(I_n - \frac{1}{w_1(t)^\top \Sigma_1 w_1(t)} \Sigma_1 w_1(t) w_1(t)^\top\right)\Sigma_1. \tag{72}$$

Rewriting the gradient flow,

$$\frac{\tau}{\mathbb{P}(\langle w_1(t), x\rangle > 0)} \frac{d}{dt} w_1 \tag{73}$$

$$= \left[\left(I_n - \frac{1}{w_1(t)^\top \Sigma_1 w_1(t)} \Sigma_1 w_1(t) w_1(t)^\top\right)\Sigma_1 + \left(I_n - \frac{1}{w_1(t)^\top \Sigma_0 w_1(t)} \Sigma_0 w_1(t) w_1(t)^\top\right)\Sigma_0\right] w_1(t) - \left[\sqrt{\frac{2}{\pi}} \left(w_1(t)^\top \Sigma w_1(t)\right)^{-\frac{1}{2}} \Sigma w_1(t)\right] \tag{74}$$

$$= -\sqrt{\frac{2}{\pi}} \left(w_1(t)^\top \Sigma_1 w_1(t)\right)^{-\frac{1}{2}} \Sigma_1 w_1(t). \tag{75}$$

By symmetry, $\mathbb{P}(\langle w_1(t), x\rangle > 0) = \frac{1}{2}$.

$$\tau \frac{d}{dt} w_1 = -2\sqrt{\frac{2}{\pi}} \left[w_1(t)^\top \Sigma_1 w_1(t)\right]^{-\frac{1}{2}} \Sigma_1 w_1(t). \tag{76}$$

Let us write $\Sigma_1 = P\Lambda P^\top$, where Λ is diagonal and P is orthogonal, and $u_1 = P^\top w_1$. Then,

$$\tau \frac{d}{dt} u_1 = -2\sqrt{\frac{2}{\pi}} \left[u_1(t)^\top \Lambda u_1(t)\right]^{-\frac{1}{2}} \Lambda u_1(t). \tag{77}$$

So it appears that this shrinks u_1 to 0, and this is accelerated as u_1 gets small by the weighted norm. This is consistent with what we observe for the Gaussian case. But what happens when x is non-Gaussian. Analytically, it's hard to say exactly. But it seems like the input-input covariance terms will *not* cancel, and so we will have another term that hopefully does more than just shrink u_1 to 0. It is surprising that $w_1(t)$ pops out of the input-output term. This *does not happen in the gated linear network*, and this seems like a key difference. What does this say about gating early during training? I should double-check this to make sure it's right. If this weren't the case though, we would get a bias term that probably yields localization (or some nonzero structure).

What we see in equation (77) is that the largest eigenvalues are shrunk fastest. This corresponds to the longest-frequency signals disappearing quickly. So, we see that the Gaussian noise quickly turns into a short-range oscillation, which is then damped out to 0. This seems consistent with my ReLU simulations, but I need to make sure the results are perfectly comparable (same learning rate, using exact same data at each time step, etc.)