1 Recall

A gated deep linear net (GDLN) is defined in terms of an architecture graph, Γ. An input $x_v \in \mathbb{R}^{|v|}$ is specified for all input nodes $v \in \text{In}(\Gamma)$. For these nodes, we set $h_v = x_v$. Activations propagate to subsequent layers according to

$$h_v = g_v \sum_{q \in E : t(q) = v} g_q W_q h_{s(q)}. \tag{1}$$

We call g_v the *node gate* and g_q the *edge gate*. The output of the GDLN is the vector h_v for all $v \in \text{Out}(\Gamma)$.

The dynamics of a weight matrix in the GDLN are given by

$$\tau \frac{d}{dt} W_e = \sum_{p \in \mathcal{P}(e)} W_{\overline{t}(p,e)}^\top \left[\Sigma^{yx}(p) - \sum_{q \in \mathcal{T}(t(p))} W_q \Sigma^{xx}(p,q) \right] W_{\overline{s}(p,e)}^\top, \tag{2}$$

where

$$\Sigma^{yx}(p) \triangleq \left\langle (g_p y_{t(p)} x_{s(p)}^\top) \right\rangle_{x,y,g} \qquad \text{and} \qquad \Sigma^{xx}(p,q) \triangleq \left\langle (g_q x_{s(q)} x_{s(p)}^\top g_p) \right\rangle_{x,y,g} \tag{3}$$

define the second-order statistics of the data and gating architecture.

2 Parity

We want to construct a GDLN that can compute the parity of an n-bit input. We do this by hierarchically computing XOR. There are a few ways to do this. We list two below.

Parallel

We have $\frac{n}{2}$ input nodes and a single output node. The first input node corresponds to the first two bits, the second to the next two bits, and so on.

We compute XOR on each input node to get a single scalar output. We do this using the gating structure defined in the initial GDLN paper. Then, we combine adjacent outputs to form new pairs on which we compute XOR. We do this recursively until we have a single output node. (Note this assumes n is a power of 2. If it is not, do this process on the first 2^k bits, where k is the largest power of 2 less than n. Then, concatenate the remaining bits with the output of the process on the first 2^k bits, and collapse the resulting vector into a single scalar by computing XOR one pair at a time.)

Sequential

Why would we ever do this? But why would we ever do the former, either?

Which is most like the correlation-length discrimination task? The convolutional structure emerges when ... (TODO)

3 Dynamics

There are four types of inputs to each node:

$$x_1 = \begin{bmatrix} -1 \\ -1 \end{bmatrix}, \quad x_2 = \begin{bmatrix} 1 \\ -1 \end{bmatrix}, \quad x_3 = \begin{bmatrix} -1 \\ 1 \end{bmatrix}, \quad x_4 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}. \tag{4}$$

So, there are $\binom{4}{2} + 4 = 10$ possible outer products in Σ^{xx}:

$$x_1 x_1^\top = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}, \quad x_1 x_2^\top = \begin{pmatrix} -1 & 1 \\ -1 & 1 \end{pmatrix}, \quad x_1 x_3^\top = \begin{pmatrix} 1 & -1 \\ 1 & -1 \end{pmatrix}, \quad x_1 x_4^\top = \begin{pmatrix} -1 & -1 \\ -1 & -1 \end{pmatrix} \tag{5}$$

$$x_2 x_2^\top = \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}, \quad x_2 x_3^\top = \begin{pmatrix} -1 & 1 \\ 1 & -1 \end{pmatrix}, \quad x_2 x_4^\top = \begin{pmatrix} 1 & 1 \\ -1 & -1 \end{pmatrix} \tag{6}$$

$$x_3 x_3^\top = \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}, \quad x_3 x_4^\top = \begin{pmatrix} -1 & -1 \\ 1 & 1 \end{pmatrix} \tag{7}$$

$$x_4 x_4^\top = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}. \tag{8}$$

Note the following relations:

- $S_1 \triangleq x_1 x_1^\top = x_4 x_4^\top = -x_1 x_4^\top$,
- $S_2 \triangleq x_1 x_2^\top = -x_1 x_3^\top$,
- $S_3 \triangleq x_2 x_2^\top = x_3 x_3^\top = -x_2 x_3^\top$,
- $S_4 \triangleq x_2 x_4^\top = -x_3 x_4^\top$.

Gating

Consider an input $\mathbf{x} \in \{0,1\}^n$. Let $e_i^{(k,l)}$ be the edge corresponding to x_i in the k-th block of the l-th layer. (Note that there are $\frac{n}{2^l}$ blocks in the l-th layer, and each block is influenced by 2^l inputs.) Define the gate corresponding to $e_i^{(k,l)}$ as

$$g_i^{(k,l)}(\mathbf{x}) = \begin{cases} 1 & \text{if } \text{parity}(\mathbf{x}_{\text{start}(k,l)}^{\text{middle}(k,l)}) = \text{first}(e_i^{(k,l)}) \wedge \text{parity}(\mathbf{x}_{\text{middle}(k,l)}^{\text{end}(k,l)}) = \text{last}(e_i^{(k,l)}) \\ 0 & \text{otherwise} \end{cases}. \tag{9}$$

Note that $\mathbf{x}_{\text{start}(k,l)}^{\text{middle}(k,l)}$ and $\mathbf{x}_{\text{middle}(k,l)}^{\text{end}(k,l)}$ are subsets of $\mathbf{x}$ corresponding to the first and second inputs to the k-th block of the l-th layer, respectively. Each of these vectors has 2^{l-1} entries. Note that there are $2^{2^{l-1}-1}$ possible inputs with a parity of 1 (or 0, respectively) for each of these vectors.

Second-order statistics

Let us consider an edge $e_i^{(k)}$ in the k-th block of the first layer corresponding to the input x_i for that block. Consider an arbitrary path p going through $e_i^{(k)}$. Notice that g_p is nonzero only for a single input $\mathbf{x}_p$, where we know that $x_{s(p)} = x_i$. Then,

$$\Sigma^{yx}(p) = \frac{1}{2^n} y(\mathbf{x}_p) x_i^\top \tag{10}$$

Now, let us consider another path q, not necessarily going through $e_i^{(k)}$. Again, g_q is nonzero only for a single input, which we denote $\mathbf{x}_q$. It corresponds to some x_j. So,

$$\Sigma^{xx}(p,q) = \frac{1}{2^n} x_{s(q)} x_{s(p)}^\top \tag{11}$$