Symmetric neural networks

YKY

June 9, 2019

1 General case for $y = Ax$

$$\boxed{\text{original}} \quad y_j = \sum_i a_{ij} x_i. \tag{1}$$

Equivariance implies:

$$\begin{aligned} \boxed{\text{swapped}} \quad y_j(\sigma(x_j\ x_k)x) &= \sigma \cdot y_j = y_k \quad \boxed{\text{original}} \\ \sum_{i \neq j,k} a_{ij} x_i + a_{kj} x_j + a_{jj} x_k &= \sum_{i \neq j,k} a_{ik} x_i + a_{jk} x_j + a_{kk} x_k. \end{aligned} \tag{2}$$

Comparing coefficients yields:

$$\begin{aligned} a_{ij} &= a_{ik} \qquad \forall\, j,k,(i \neq j,k) \\ a_{kj} &= a_{jk} \qquad \forall\, j,k \\ a_{jj} &= a_{kk} \qquad \forall\, j,k. \end{aligned} \tag{3}$$

In other words, the matrix A is of the form:

$$A = \alpha I + \beta 11^T. \tag{4}$$

2 Case for $y_k = A_k x \cdot x$

The general form of a "quadratic" vector function is:

$$y = (Ax) \cdot x + Bx + C. \tag{5}$$

We just focus on the quadratic term $(Ax) \cdot x$:

$$\boxed{\text{original}} \quad y_k = \sum_j \left[\sum_i a_{ij}^k x_i \right] x_j. \tag{6}$$

Note that the matrix A is "3D" and has $N \times N \times N$ entries.

Equivalance implies:

$$\boxed{\text{swapped}} \quad y_k(\sigma(x_k\ x_h)\cdot x) = \sigma \cdot y_k = y_h \quad \boxed{\text{original}} \tag{7}$$

$$\begin{aligned}
LHS &= \sum_j \left[\sum_{i\neq h,k} a^k_{ij}x_i + a^k_{hj}x_k + a^k_{kj}x_h\right]\sigma\cdot x_j \\
&= \sum_{j\neq h,k}\left[\sum_{i\neq h,k} a^k_{ij}x_i + a^k_{hj}x_k + a^k_{kj}x_h\right]x_j + \left[\sum_{i\neq h,k} a^k_{ih}x_i + a^k_{hh}x_k + a^k_{kh}x_h\right]x_k + \left[\sum_{i\neq h,k} a^k_{ik}x_i + a^k_{hk}x_k + a^k_{kk}x_h\right]x_h \\
&= \sum_{j\neq h,k}\sum_{i\neq h,k} a^k_{ij}x_ix_j + \sum_{j\neq h,k} a^k_{hj}x_kx_j + \sum_{j\neq h,k} a^k_{kj}x_hx_j \\
&\quad + \sum_{i\neq h,k} a^k_{ih}x_ix_k + a^k_{hh}x_k^2 + a^k_{kh}x_hx_k \\
&\quad + \sum_{i\neq h,k} a^k_{ik}x_ix_h + a^k_{hk}x_kx_h + a^k_{kk}x_h^2 \\
RHS &= \sum_j\left[\sum_i a^h_{ij}x_i\right]x_j \\
&= \sum_{j\neq h,k}\sum_{i\neq h,k} a^h_{ij}x_ix_j + \sum_{j\neq h,k} a^h_{kj}x_kx_j + \sum_{j\neq h,k} a^h_{hj}x_hx_j \\
&\quad + \sum_{i\neq h,k} a^h_{ik}x_ix_k + a^h_{kk}x_k^2 + a^h_{hk}x_hx_k \\
&\quad + \sum_{i\neq h,k} a^h_{ih}x_ix_h + a^h_{kh}x_kx_h + a^h_{hh}x_h^2
\end{aligned} \tag{8}$$

Comparing coefficients yields:

$$\begin{aligned}
a^h_{ij} &= a^k_{ij} && \forall\ h,k,(i\neq h,k,j\neq h,k) \\
a^h_{kj} &= a^k_{hj} && \forall\ h,k,(j\neq h,k) \\
a^h_{hj} &= a^k_{kj} && \forall\ h,k,(j\neq h,k) \\
a^h_{ik} &= a^k_{ih} && \forall\ h,k,(i\neq h,k) \\
a^h_{ih} &= a^k_{ik} && \forall\ h,k,(i\neq h,k) \\
a^h_{kk} &= a^k_{hh} && \forall\ h,k \\
a^h_{hh} &= a^k_{kk} && \forall\ h,k \\
a^h_{hk} + a^h_{kh} &= a^k_{hk} + a^k_{kh} && \forall\ h,k.
\end{aligned} \tag{9}$$

How many different colors?

$$\begin{array}{llll}
N=2\ \ldots. & 6 & /\ 8 & = 75\% \\
N=3\ \ldots. & 9 & /\ 27 & = 33.3\% \\
N=4\ \ldots. & 11 & /\ 64 & = 17/2\% \\
N=5\ \ldots. & 13 & /\ 125 & = 10.4\% \\
N=6\ \ldots. & 15 & /\ 216 & = 6.9\%
\end{array} \tag{10}$$

There would be N **blocks** of $N\times N$ matrices.

All diagonals consists of 2 colors, regardless of N (from 2nd and 3rd equations). This leaves $N(N-1)$ non-diagonal entries per block.

Non-diagonal entries of different blocks are equal, if the block indices are different from the row and column indices. Out of N blocks there would be 2 different sets of non-diagonal weights. (This comes from the 1st equation.)

The last equation causes non-diagonal weights to have a certain symmetry about the diagonal.

3 With output space "folded in half"

Now suppose the output is only 1/2 the dimension of the input. Define a new form of equivariance such that the input permutation would act on the output as "folded in half".

In other words, equivariance is changed to:

$$\boxed{\text{swapped}} \quad y_k \cdot \sigma(x_k \; x_h) = y_h \text{ or } y_{h-N/2} \quad \boxed{\text{original}} \tag{11}$$

where τ is σ acting on y as double its length and identifying $y_i = y_{i+N/2}$.

3.1 Linear case

Just notice that the dimension of y is halved:

$$\boxed{\text{original}} \quad y_j = \sum_i a_{ij} x_i. \tag{12}$$

"Folded" equivariance implies:

$$\begin{aligned} \boxed{\text{swapped}} \quad y_j(\sigma(x_j \; x_k)x) &= \sigma \cdot y_j = y_k \quad \boxed{\text{original}} \\ \sum_{i \neq j,k} a_{ij} x_i + a_{kj} x_j + a_{jj} x_k &= \sum_{i \neq j,k} a_{ik} x_i + a_{jk} x_j + a_{kk} x_k \end{aligned} \tag{13}$$

with the restriction $j \in \{1, ..., N/2\}$, and $k \in \{1, ..., N\}$.

The constraints obtained are same as before, except that index ranges are different:

$$\begin{aligned} a_{ij} &= a_{ik} \qquad \forall\, j,k, (i \neq j,k) \\ a_{kj} &= a_{jk} \qquad \forall\, j,k \\ a_{jj} &= a_{kk} \qquad \forall\, j,k \end{aligned}$$

These constraints give rise to a matrix of this form (for the 6×3 case, numbers represent different colors):

$$\begin{matrix} 5 & 1 & 1 & 2 & 3 & 4 \\ 1 & 5 & 1 & 2 & 3 & 4 \\ 1 & 1 & 5 & 2 & 3 & 4 \end{matrix} \; . \tag{14}$$

This pattern is obtained from my Python code.

3.2 Quadratic case