Symmetric neural networks

YKY

June 3, 2019

1 General case for $y = Ax$

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

Equivariance implies:

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

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:

$$A = \alpha I + \beta 1 1^T. \tag{4}$$

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.

Equivalence implies:

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

$$\sum_{j\neq h,k}\sum_{i\neq h,k} a^k_{ij}x_ix_j + a^k_{hh}x_k^2 + a^k_{kh}x_hx_k + a^k_{hk}x_kx_h + a^k_{kk}x_h^2 = \sum_{j\neq h,k}\sum_{i\neq h,k} a^h_{ij}x_ix_j + a^h_{hh}x_h^2 + a^h_{kh}x_kx_h + a^h_{hk}x_hx_k + a^h_{kk}x_k^2$$

which yields:

$$\begin{aligned} a^h_{ij} &= a^k_{ij} && \forall\ h,k,(i\neq h,k,j\neq h,k) \\ a^h_{hh} &= a^k_{kk} && \forall\ h,k \\ a^k_{hh} &= a^h_{kk} && \forall\ h,k \\ a^k_{kh}+a^k_{hk} &= a^h_{kh}+a^h_{hk} && \forall\ h,k. \end{aligned} \tag{8}$$

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:

$$\boxed{\text{swapped}} \quad y_k\cdot\sigma(x_k\ x_h) = y_h \text{ or } y_{h-N/2} \quad \boxed{\text{original}} \tag{9}$$

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:

$$\boxed{\text{original}} \quad y_j = \sum_i a_{ij}x_i. \tag{10}$$

"Folded" equivariance implies:

$$\begin{aligned} \boxed{\text{swapped}} \quad y_j(\sigma(x_j\ x_k)x) &= y_k \quad \boxed{\text{original}} \\ \sum_{i\neq j,k} a_{ij}x_i + a_{kj}x_j + a_{jj}x_k &= \sum_{i\neq j,k} a_{ik}x_i + a_{jk}x_j + a_{kk}x_k \end{aligned} \tag{11}$$

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:

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

These constraints give rise to a matrix of this form (for the 6×2 case, numbers represent different colors):

$$\begin{matrix} 5 & 1 & 1 & 2 & 3 & 4 \\ 1 & 5 & 1 & 2 & 3 & 4 \\ 1 & 1 & 5 & 2 & 3 & 4 \end{matrix} \quad . \tag{12}$$

This pattern is obtained from my Python code.

3.2 Quadratic case