Math card on MQ's Self-Reflexive Network

Elements

MQ string vector

$$\vec{S} = (\, a_1a_1,\ a_1a_2,\ a_1a_2a_2,\ a_1a_2a_1,\ a_2a_2,\ a_2a_1,\ a_2a_1a_1,\ a_2a_1a_2)$$

Input matrix

$$\mathbf{I} = \begin{bmatrix} a_1a_2a_2 & a_1a_2a_1 & a_1a_1 & a_1a_2 & a_2a_1 & a_1a_2 & a_1a_2a_1 & a_1a_2a_1 & a_1a_1 & a_1a_1 & a_1a_2a_2 & a_1a_1 & a_1a_2a_2 & a_1a_1 \\ a_2a_2 & a_2a_1 & a_2a_1 & a_1a_2a_2 & a_2a_1a_1 & a_2a_1 & a_2a_1a_2 & a_1a_2 & a_1a_2a_1 & a_1a_2 & a_1a_2a_1 & a_1a_2a_2 & a_2a_1a_1 & a_2a_2 \end{bmatrix}$$

Hidden weight matrix

$$\mathbf{W_h} = \begin{bmatrix} w_{1,1} & w_{1,2} & w_{1,3} & w_{1,4} & w_{1,5} & w_{1,6} & w_{1,7} & w_{1,8} & w_{1,9} & w_{1,10} & w_{1,11} & w_{1,12} & w_{1,13} & w_{1,14} \\ w_{2,1} & w_{2,2} & w_{2,3} & w_{2,4} & w_{2,5} & w_{2,6} & w_{2,7} & w_{2,8} & w_{2,9} & w_{2,10} & w_{2,11} & w_{2,12} & w_{2,13} & w_{2,14} \end{bmatrix}$$

Output weight matrix

$$\mathbf{W_o} = \begin{bmatrix} w_{1,1} & w_{1,2} & \dots & w_{1,14} \\ w_{2,1} & w_{2,2} & \dots & w_{2,14} \\ \dots & \dots & \dots & \dots \\ w_{14,1} & w_{14,2} & \dots & w_{14,14} \end{bmatrix}$$

Generic bias matrix

$$\boldsymbol{\theta} = \begin{bmatrix} b_1 & b_2 & b_3 & b_4 & b_5 & b_6 & b_7 & b_8 & b_9 & b_{10} & b_{11} & b_{12} & b_{13} & b_{14} \end{bmatrix}$$

Hidden activation matrix

$$\mathbf{A_h} = \begin{bmatrix} h_1 & h_2 & h_3 & h_4 & h_5 & h_6 & h_7 & h_8 & h_9 & h_{10} & h_{11} & h_{12} & h_{13} & h_{14} \end{bmatrix}$$

Hidden activation delta matrix

$$\mathbf{\Delta A_h} = \begin{bmatrix} \Delta h_1 & \Delta h_2 & \Delta h_3 & \Delta h_4 & \Delta h_5 & \Delta h_6 & \Delta h_7 & \Delta h_8 & \Delta h_9 & \Delta h_{10} & \Delta h_{11} & \Delta h_{12} & \Delta h_{13} & \Delta h_{14} \end{bmatrix}$$

Output activation matrix

$$\mathbf{A_o} = \begin{bmatrix} o_1 & o_2 & o_3 & o_4 & o_5 & o_6 & o_7 & o_8 & o_9 & o_{10} & o_{11} & o_{12} & o_{13} & o_{14} \end{bmatrix}$$

Output activation delta matrix

$$\mathbf{\Delta A_o} = \begin{bmatrix} \Delta o_1 & \Delta o_2 & \Delta o_3 & \Delta o_4 & \Delta o_5 & \Delta o_6 & \Delta o_7 & \Delta o_8 & \Delta o_9 & \Delta o_{10} & \Delta o_{11} & \Delta o_{12} & \Delta o_{13} & \Delta o_{14} \end{bmatrix}$$

Unitary matrix definition

$$\mathbf{1_{i \times j}} = \begin{bmatrix} 1 & 1 & \dots & 1 \\ 1 & 1 & \dots & 1 \\ \dots & \dots & \dots & \dots \\ 1 & 1 & \dots & 1 \end{bmatrix}$$

i rows and j columns filled with ones

Operations

Hadamard product (also known as the element-wise product)

$$\mathbf{A_{m\times n}} \odot \mathbf{B_{p\times q}} = \begin{bmatrix} a_{1,1} & a_{1,2} & \dots & a_{1,n} \\ a_{2,1} & a_{2,2} & \dots & a_{2,n} \\ \dots & \dots & \dots & \dots \\ a_{m,1} & a_{m,2} & \dots & a_{m,n} \end{bmatrix} \odot \begin{bmatrix} b_{1,1} & b_{1,2} & \dots & b_{1,q} \\ b_{2,1} & b_{2,2} & \dots & b_{2,q} \\ \dots & \dots & \dots & \dots \\ b_{p,1} & b_{p,2} & \dots & b_{p,q} \end{bmatrix} = \begin{bmatrix} a_{1,1}\cdot b_{1,1} & a_{1,2}\cdot b_{1,2} & \dots & a_{1,n}\cdot b_{1,q} \\ a_{2,1}\cdot b_{2,1} & a_{2,2}\cdot b_{2,2} & \dots & a_{2,n}\cdot b_{2,q} \\ \dots & \dots & \dots & \dots \\ a_{m,1}\cdot b_{p,1} & a_{m,2}\cdot b_{p,2} & \dots & a_{m,n}\cdot b_{p,q} \end{bmatrix}$$

$$\mathbf{A} \text{ and } \mathbf{B} \text{ of the same dimension}: \ \mathrm{m} = \mathrm{p} \ \wedge \ \mathrm{n} = \mathrm{q}$$

Penetrating face product

$$\mathbf{B} = [\mathbf{B_n}] = [\mathbf{B}_1 \mid \mathbf{B}_2 \mid \dots \mid \mathbf{B}_n]$$

$$\mathbf{A} \, [\odot] \, \mathbf{B} = [\mathbf{A} \odot \mathbf{B}_1 \mid \mathbf{A} \odot \mathbf{B}_1 \mid \dots \mathbf{A} \odot \mathbf{B}_n]$$

Formulas

Version	***Foward***
Book	$NET_j = \sum_i W_{ij} \cdot U_i$ $F(NET_j) = \frac{1}{1+e^{-\frac{(NET_j + \theta_j)}{Temp}}}$ $Error = \sum_{j=1}^{n} (U_o - U_h)^2$ $Temp = 1 - \frac{1}{1+Error}$
Implementation	$\mathbf{NET_h} = \mathbf{1}_{1\times2}(\mathbf{I} \odot \mathbf{W_h})$ $\mathbf{Net_o} = \mathbf{A_h W_o}$ $\mathbf{F(NET)} = \frac{1}{1+e^{-\frac{(\mathbf{NET} + \boldsymbol{\theta})}{Temp}}}$ $Error = \mathbf{1}_{14\times1}[(\mathbf{A_o} - \mathbf{A_h}) \odot (\mathbf{A_o} - \mathbf{A_h})]$ $Temp = 1 - \frac{1}{1+Error}$
	Backpropagation, weights connected to the output
Book	$\Delta U_{j_{(n+1)}} = U_j \cdot (U_i - Uj) \cdot (1 - Uj) + \Delta U_{j_{(n)}} \cdot (U_i - U_j)$ $\Delta W_{ij} = r \cdot \Delta U_j \cdot U_i$ $W_{ij(n+1)} = W_{ij(n)} + \Delta W_{ij}$
Implementation	$\mathbf{\Delta A_{o_{(n+1)}}} = \mathbf{A_o} \odot (\mathbf{A_h} - \mathbf{A_o}) \odot (\mathbf{1}_{1\times14} - \mathbf{A_o}) + \mathbf{\Delta A_{o_{(n)}}} \odot (\mathbf{A_h} - \mathbf{A_o})$ $\mathbf{\Delta W_o} = \alpha \cdot \mathbf{A_h^T}\, \mathbf{\Delta A_{o_{(n+1)}}}$ $\mathbf{W_{o(n+1)}} = \mathbf{W_{o(n)}} + \mathbf{\Delta W_o}$
	Backpropagation, weights NOT connected to output
Book	$\Delta U_i = U_i \cdot (1 - U_i) \cdot \sum_j (\Delta U_j \cdot W_{ij})$ $\Delta W_{ki} = r \cdot \Delta U_i \cdot U_k$ $W_{ki(n+1)} = W_{ki(n)} + \Delta W_{ki}$
Implementation	$\mathbf{\Delta A_h} = \mathbf{A_h} \odot (\mathbf{1}_{1\times14} - \mathbf{A_h}) + \mathbf{\Delta A_o W_o}$ $\mathbf{\Delta W_h} = \alpha \cdot \mathbf{\Delta A_h}\, [\odot]\, \mathbf{I}$ $\mathbf{W_{h(n+1)}} = \mathbf{W_{h(n)}} + \mathbf{\Delta W_h}$

Table 1: algorithm formulas

Notation

	hidden	*output*	*activation*
Book	i	j	U
Implementation	h	o	A

Table 2: used letters

Code

Op	Math	Python/Tensorflow
scalar multiplication	$a \cdot b$	`A * B`
matrix multiplication	$\mathbf{AB}$	`tf.matmul() # or '@'`
Hadamard product	$\mathbf{A} \odot \mathbf{B}$	`A * B`
penetrating face product	$\mathbf{A} [\odot] \mathbf{B}$	`A * B`
sum of columns	$\mathbf{1}_{i \times j} \mathbf{A}_{j \times i}$	`tf.math.reduce_sum()`

Table 3: math to code

References

1)

Massimo, B. (1993). *Il Modello MQ: Reti Neurali e Percezione Interpersonale / Massimo Buscema Giulia Massini.* (1st ed.). Armando Editore.

(Forulas pag. 158-159)

2)

https://en.wikipedia.org/wiki/Hadamard_product_(matrices)#The_penetrating_face_product