

Pullback / vJp rule for broadcast functions (relevant for backprop)

(Neural Network activation functions)

$$f(\underline{x}) = \underline{y}$$

$$\underline{x} \in \mathbb{R}^N$$

$$\underline{y} \in \mathbb{R}^N$$

$$f: \mathbb{R}^N \rightarrow \mathbb{R}^N$$

$$\sigma: \mathbb{R} \rightarrow \mathbb{R}$$

$$f\left(\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix}\right) = \begin{bmatrix} \sigma(x_1) \\ \sigma(x_2) \\ \vdots \\ \sigma(x_N) \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix} = \sigma(\underline{x})$$

task: backpropagate $\underline{\bar{y}} \in \mathbb{R}^N$ to $\underline{\bar{x}} \in \mathbb{R}^N$

$$\underline{\bar{x}}^T = \underline{\bar{y}}^T \frac{\partial f}{\partial \underline{x}} = [\bar{y}_1 \dots \bar{y}_N]$$

$$\begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \dots & \frac{\partial f_1}{\partial x_N} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \dots & \frac{\partial f_2}{\partial x_N} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_N}{\partial x_1} & \frac{\partial f_N}{\partial x_2} & \dots & \frac{\partial f_N}{\partial x_N} \end{bmatrix}$$

Diagram illustrating the Jacobian matrix of the function f . The matrix is $N \times N$. The diagonal elements are circled in orange and labeled $\sigma'(x_i)$. The off-diagonal elements are also circled in orange. The matrix is multiplied by $\underline{\bar{y}}^T$ to get $\underline{\bar{x}}^T$.

$$= \underline{\bar{y}}^T \text{diag}(\underbrace{\sigma'(\underline{x})}_{\text{derivative of the scalar function}})$$

derivative of the scalar function

broadcastedly applied to the primal input

apply $(\cdot)^T$ on both sides

$$\underline{\bar{x}} = (\text{diag}(\sigma'(\underline{x})))^T \underline{\bar{y}}$$

$$= \text{diag}(\sigma'(\underline{x})) \underline{\bar{y}}$$

$$= \sigma'(\underline{x}) \circ \underline{\bar{y}}$$

↑
elementwise multiplication

Full pullback rule

$$\mathcal{B}(f, (\underline{x},), (\underline{\bar{y}},)) = ((\sigma(\underline{x}),), (\sigma'(\underline{x}) \circ \underline{\bar{y}},))$$

for some NN activation functions

- sigmoid: $\sigma'(x) = \sigma(x) \cdot (1 - \sigma(x))$

- tanh: $\sigma'(x) = 1 - \sigma^2(x)$

- relu: $\sigma'(x) = \begin{cases} 1, & x > 0 \\ 0, & \text{else} \end{cases}$