Pullback / vJp rule for broadcast functions

(relevant for backprop)

(Neural Network activation functions)

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

$$f\left(\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix}\right) = \begin{bmatrix} \sigma(x_1) \\ \sigma(x_2) \\ \vdots \\ \sigma(x_N) \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_3 \end{bmatrix}$$

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

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

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

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

$= \sigma.(\underline{x})$

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

$$\bar{\underline{x}}^T = \bar{\underline{y}}^T \frac{\partial f}{\partial \underline{x}} = [\bar{y}_1 \; \cdots \; \bar{y}_N] \begin{bmatrix} \underbrace{\frac{\partial f_1}{\partial x_1}}_{\sigma'(x_1)} & \underbrace{\frac{\partial f_1}{\partial x_2}}_{0} & \cdots & \\ \underbrace{\frac{\partial f_2}{\partial x_1}}_{0} & \underbrace{\frac{\partial f_2}{\partial x_2}}_{\sigma'(x_2)} & & \\ \vdots & & \ddots & \\ & & & \underbrace{\frac{\partial f_N}{\partial x_N}}_{\sigma'(x_N)} \end{bmatrix}$$

$$= \bar{\underline{y}}^T \operatorname{diag}(\underbrace{\sigma'.(\underline{x})}_{})$$

derivative of the scalar function broadcastedly applied to the primal input

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

$$\bar{\underline{x}} = \left(\operatorname{diag}(\sigma'.(\underline{x}))\right)^T \bar{\underline{y}}$$

$$= \operatorname{diag}(\sigma'.(\underline{x})) \, \bar{\underline{y}}$$

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

element wise multiplication

Full pullback rule

$$\boxed{\mathcal{B}(f, (\underline{x},), (\bar{\underline{y}},)) = \left((\sigma.(\underline{x}),), \; (\sigma'.(\underline{x}) \odot \bar{\underline{y}},)\right)}$$

for some NN activation functions

- Sigmoid: $\sigma'(x) = \sigma(x) \cdot (1 - \sigma(x))$
- tanh: $\sigma'(x) = 1 - \sigma^2(x)$
- relu: $\sigma'(x) = \begin{cases} 1, & x > 0 \\ 0, & \text{else} \end{cases}$