16.1) Deep Feedforward Networks

Vitor Kamada

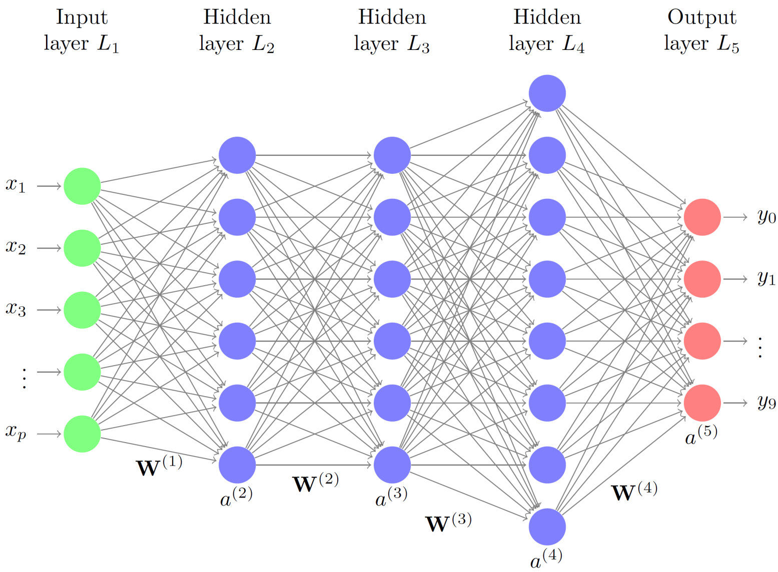
August 2019

Reference

Goodfellow et al. (2016): Ch 6.

https://www.deeplearningbook.org/

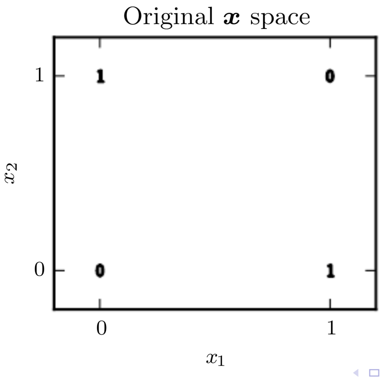
Deep Feedforward



Efron & Hastie (2016: 354)

XOR function ("exclusive or")

$$X = \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 0 \\ 1 & 1 \end{bmatrix} \qquad Y = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$



Goodfellow et al. (2016: 169)

MSE Loss Function

$$y = f^*(\mathbf{x})$$

$$J(\boldsymbol{\theta}) = \frac{1}{4} \sum_{x \in \mathbb{X}} (f^*(\mathbf{x}) - f(\mathbf{x}; \boldsymbol{\theta}))^2$$

$$f(\mathbf{x}; \mathbf{w}, \boldsymbol{b}) = \mathbf{x}^{\mathbf{T}} \mathbf{w} + b$$

$$\mathbf{w} = 0 \text{ and } b = \frac{1}{2}$$

Two Functions Chained Together

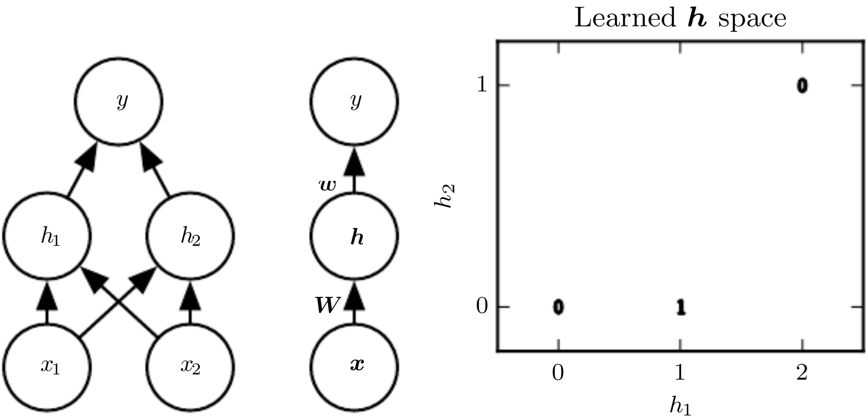
$$f(\mathbf{x}; \mathbf{W}, \mathbf{c}, \mathbf{w}, \boldsymbol{b}) = f^{(2)}(f^{(1)}(\mathbf{x}))$$

$$h = f^{(1)}(\mathbf{x}; \mathbf{W}, c)$$

$$y = f^{(2)}(\mathbf{h}; \mathbf{w}, b)$$

$$f(\mathbf{x}; \mathbf{W}, \mathbf{c}, \mathbf{w}, \boldsymbol{b}) = \mathbf{w}^{\mathbf{T}} max\{0, \mathbf{W}^{\mathbf{T}}\mathbf{x} + \mathbf{c}\} + b$$

Solving the XOR by Learning Representation



Goodfellow et al. (2016: 169-170)

$\mathbf{w}^\mathsf{T} max\{0, \mathbf{W}^\mathsf{T}\mathbf{x} + \mathbf{c}\} + b$

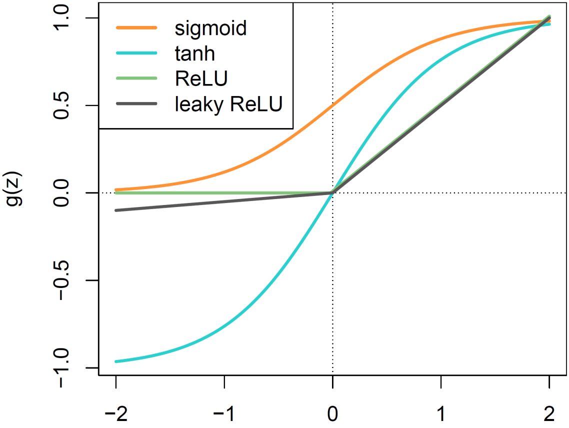
$$\mathbf{W} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}_{2\times 2} \quad \mathbf{c} = \begin{bmatrix} 0 \\ -1 \end{bmatrix}_{2\times 1} \quad \mathbf{w} = \begin{bmatrix} 1 \\ -2 \end{bmatrix}_{2\times 1} \quad b = 0$$

$$\mathbf{X} = \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 0 \\ 1 & 1 \end{bmatrix}_{4\times 2} \quad \mathbf{XW} = \begin{bmatrix} 0 & 0 \\ 1 & 1 \\ 1 & 1 \\ 2 & 2 \end{bmatrix}_{4\times 2} \quad \mathbf{XW} + \mathbf{c} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \\ 1 & 0 \\ 2 & 1 \end{bmatrix}_{4\times 2}$$

$$max\{0, \mathbf{XW} + \mathbf{c}\} = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 1 & 0 \\ 2 & 1 \end{bmatrix}_{4\times 2} \quad max\{0, \mathbf{XW} + \mathbf{c}\}\mathbf{w} = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}_{4\times 1} = Y$$

Rectified Linear Unit (ReLU) vs Leaky ReLU

$$g(z) = max(0, z)$$



Efron & Hastie (2016:361)

$$g(z, \alpha) = max(0, z) + \alpha min(0, z), \ \alpha = 0.01$$

Generalized ReLU

$$g(z, \alpha) = max(0, z) + \alpha min(0, z)$$

$$\text{If } \alpha = -1 \rightarrow g(z) = |z|$$

Absolute Value Rectification

If α is learnable parameter, then

Parametric ReLU or PReLU