Compositing Units in Neural Networks

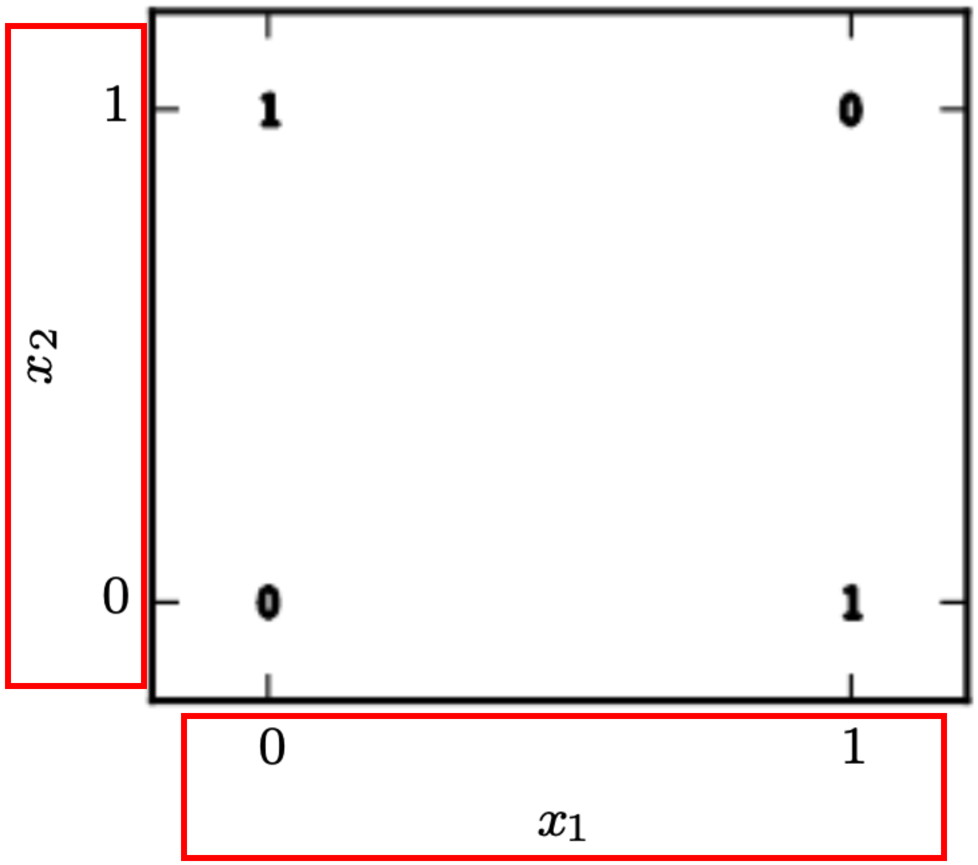
CPT_S 434/534 Neural network design and application

Today's class

- What makes feedforward network different from linear model
  - Understanding its structure
- Units for neural networks
  - Output units → cost function
    - $f_m\left(\ldots\left(f_2\left(f_1(w; x_i)\right)\right)\right) \rightarrow y_i$
  - Hidden units
    - $f_m\left(\ldots\left(f_2\left(f_1(w; x_i)\right)\right)\right) \rightarrow y_i$

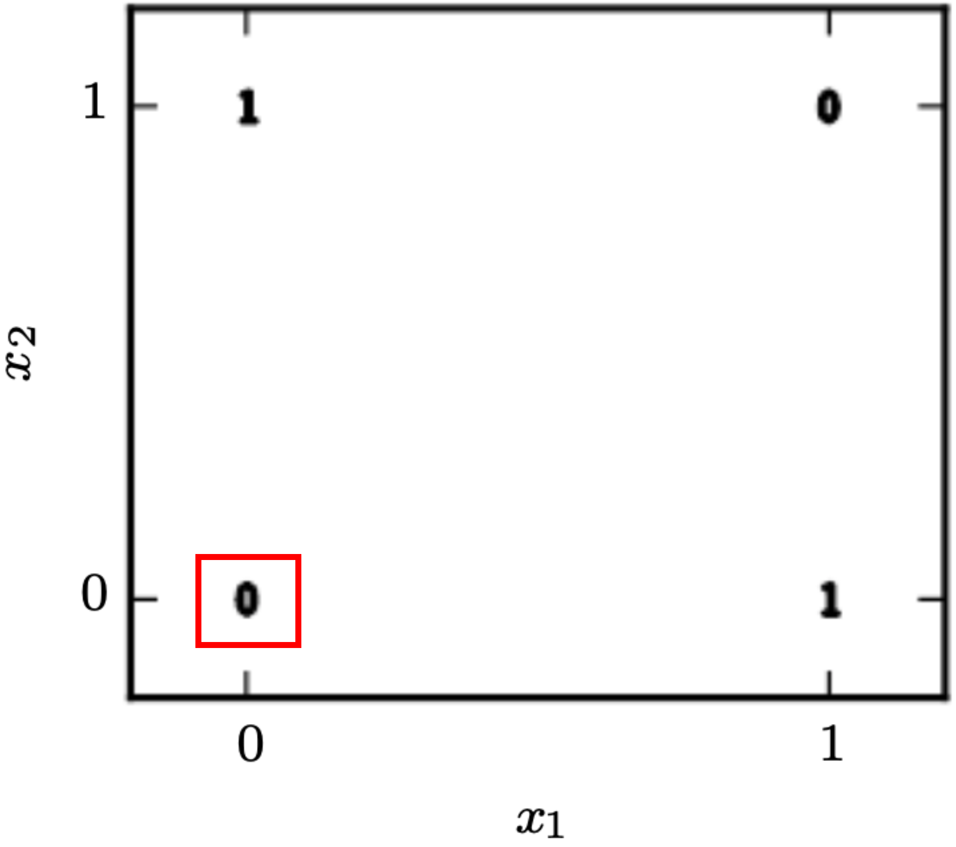
Learning XOR function

$$\boldsymbol{X} = \begin{array}{c} \begin{matrix} x_1 & x_2 \end{matrix} \\ \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 0 \\ 1 & 1 \end{bmatrix} \end{array} \qquad \begin{matrix} 0 \\ 1 \\ 1 \\ 0 \end{matrix}$$



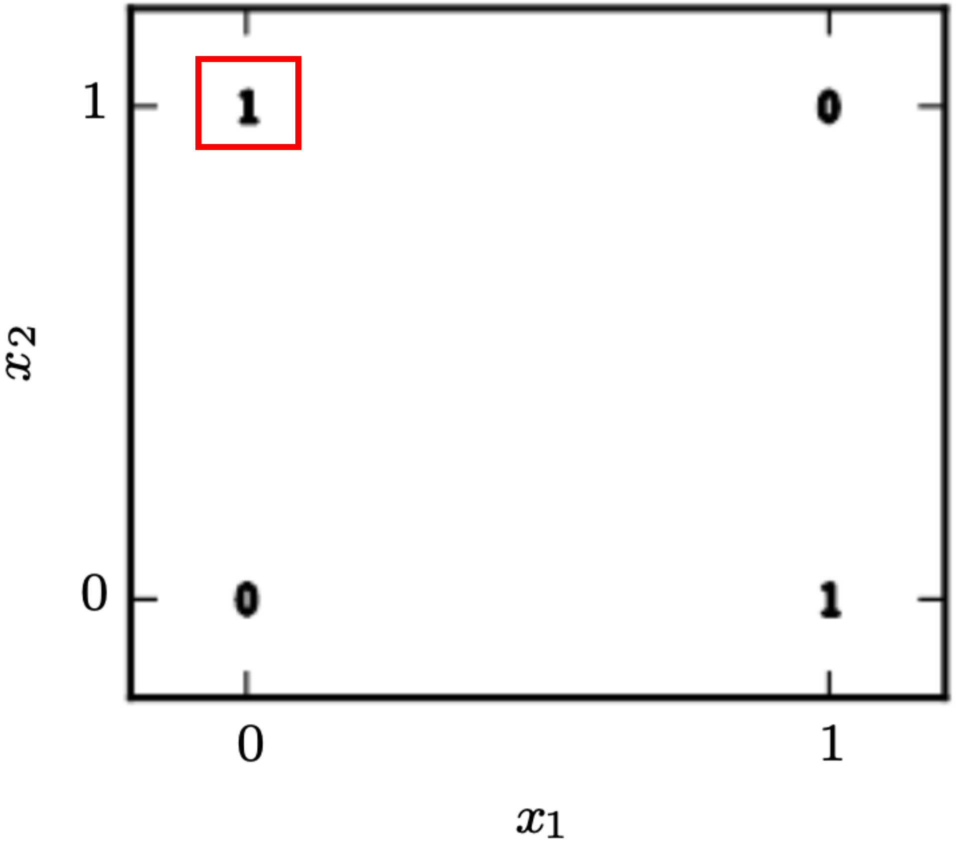
Learning XOR function

$$\begin{array}{cc} & \begin{array}{cc} x_1 & x_2 \end{array} \\ \boldsymbol{X} = & \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 0 \\ 1 & 1 \end{bmatrix} \end{array} \quad \begin{array}{c} 0 \\ 1 \\ 1 \\ 0 \end{array}$$



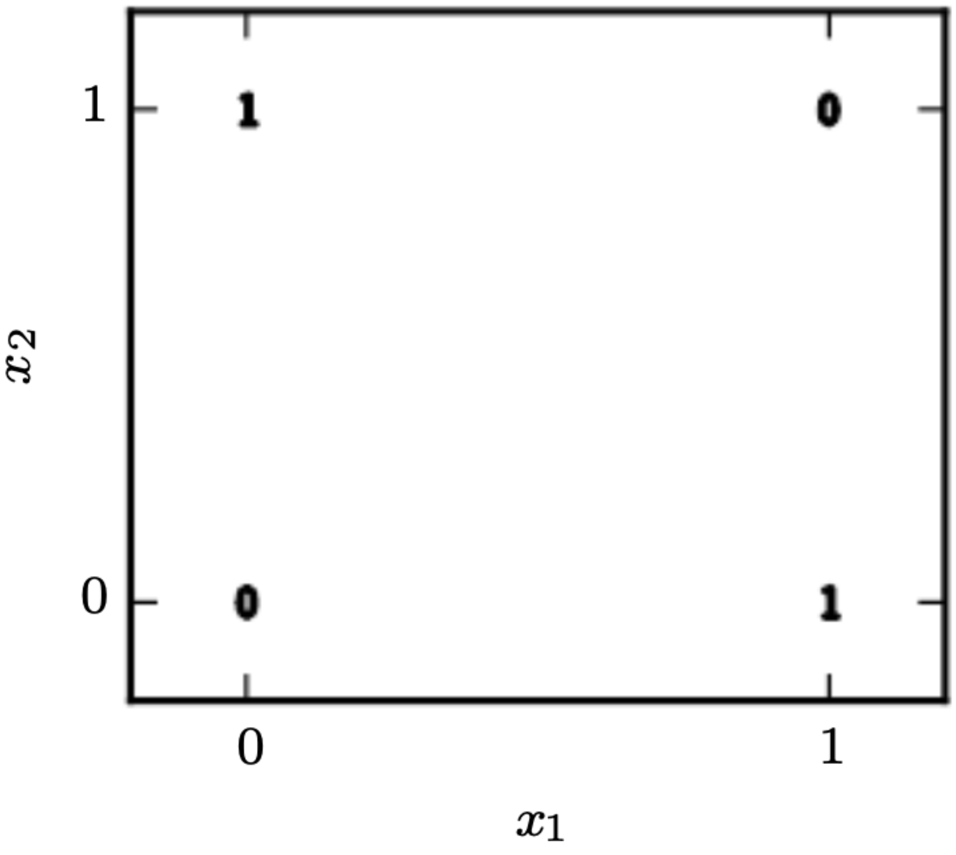
Learning XOR function

$$\boldsymbol{X} = \begin{array}{c} \begin{array}{cc} x_1 & x_2 \end{array} \\ \left[\begin{array}{cc} 0 & 0 \\ 0 & 1 \\ 1 & 0 \\ 1 & 1 \end{array}\right] \end{array} \quad \begin{array}{c} \\ 0 \\ 1 \\ 1 \\ 0 \end{array}$$



Learning XOR function

$$\boldsymbol{X} = \begin{array}{c} \begin{array}{cc} x_1 & x_2 \end{array} \\ \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 0 \\ 1 & 1 \end{bmatrix} \end{array} \quad \begin{array}{c} 0 \\ 1 \\ 1 \\ 0 \end{array}$$



Q: Can we use a linear model to separate 0/1 classes?

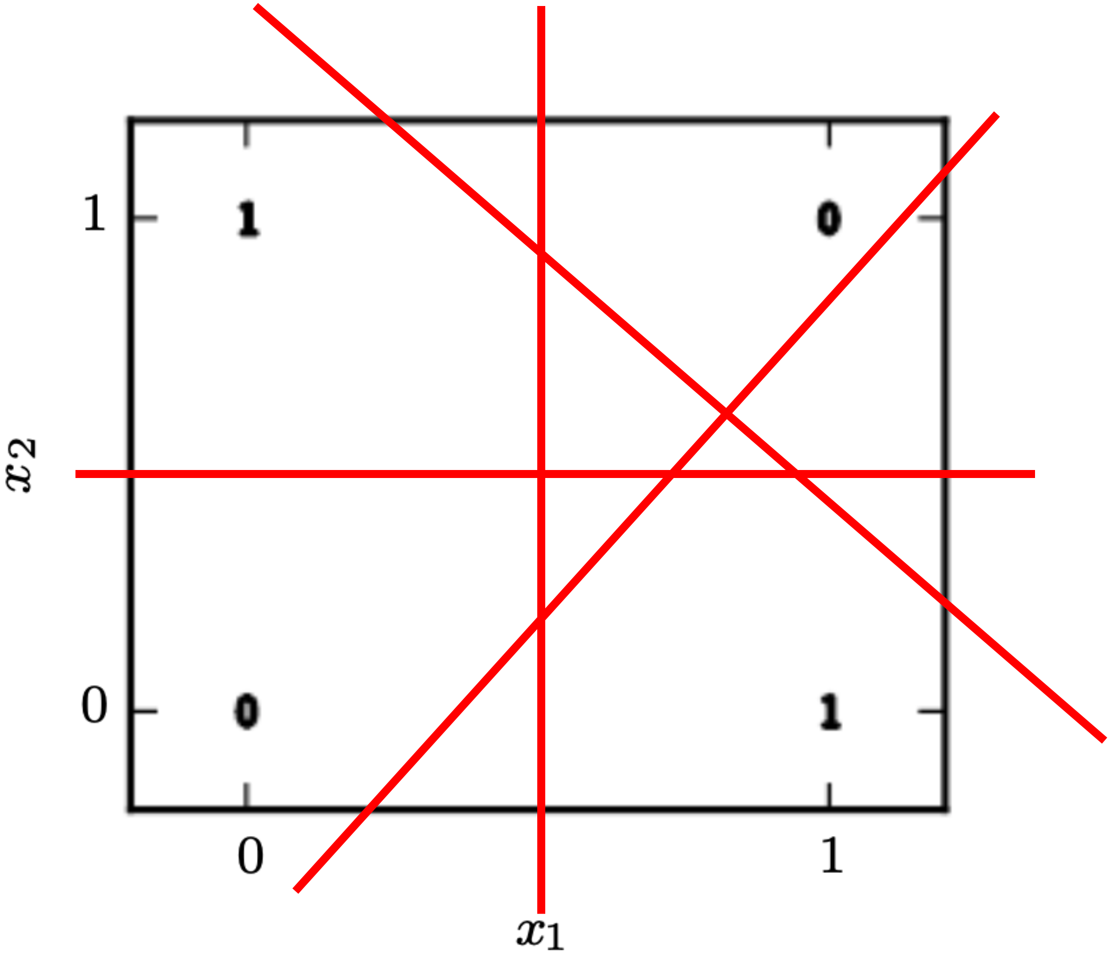
$$\boldsymbol{W}^\top \boldsymbol{x} + \boldsymbol{c}$$

Learning XOR function

$$
\boldsymbol{X} = \begin{array}{c} \begin{array}{cc} x_1 & x_2 \end{array} \\ \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 0 \\ 1 & 1 \end{bmatrix} \end{array} \quad \begin{array}{c} \\ 0 \\ 1 \\ 1 \\ 0 \end{array}
$$

Q: Can we use a linear model to separate 0/1 classes?

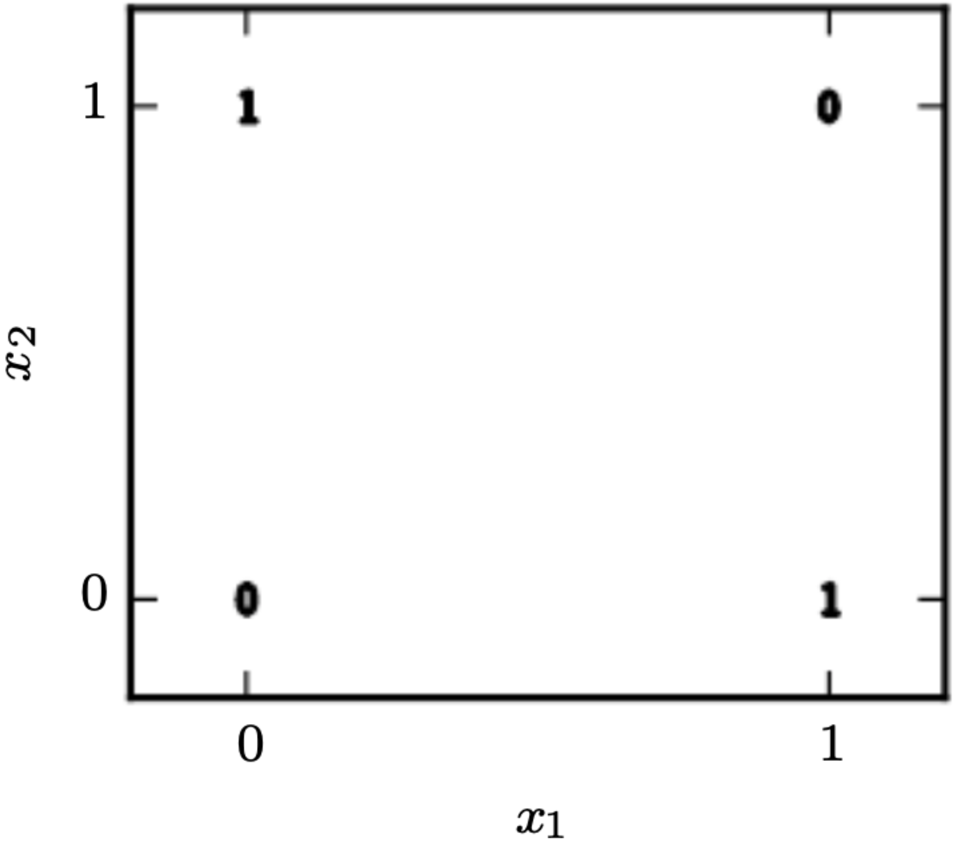
$$\boldsymbol{W}^\top \boldsymbol{x} + \boldsymbol{c}$$



Learning XOR function

$$\boldsymbol{X} = \begin{array}{c} \\ \\ \\ \\ \\ \end{array} \begin{array}{cc} x_1 & x_2 \\ \end{array}$$

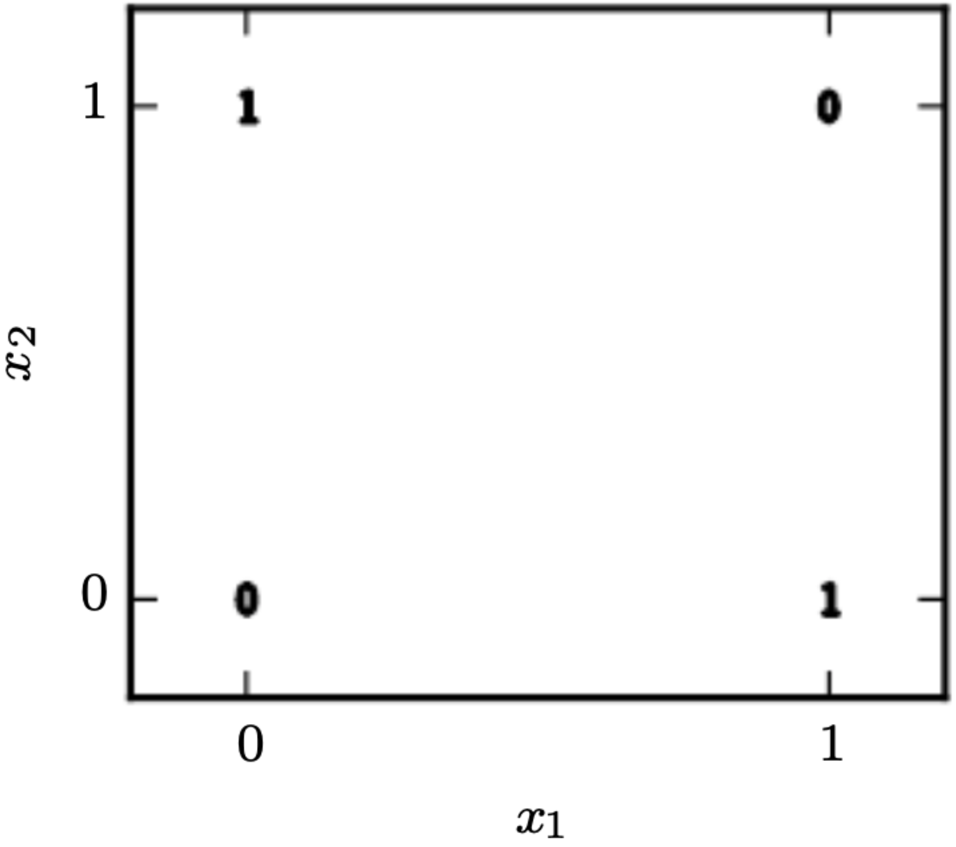
$$\boldsymbol{X} = \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 0 \\ 1 & 1 \end{bmatrix} \quad \begin{matrix} 0 \\ 1 \\ 1 \\ 0 \end{matrix}$$



What if we use a nonlinear function as: $f(\boldsymbol{x}; \boldsymbol{W}, \boldsymbol{c}, \boldsymbol{w}, b) = \boldsymbol{w}^\top \max\{0, \boldsymbol{W}^\top \boldsymbol{x} + \boldsymbol{c}\} + b.$

Learning XOR function

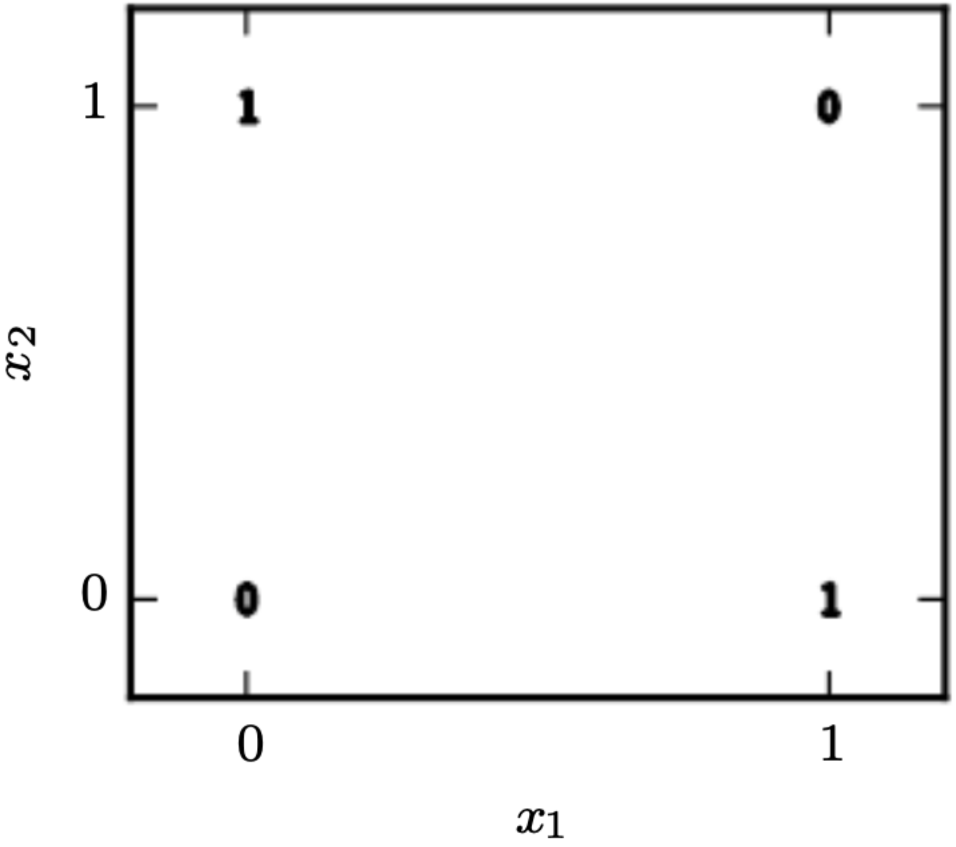
$$
\begin{array}{cc}
 & \begin{array}{cc} x_1 & x_2 \end{array} \\
\boldsymbol{X} = & \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 0 \\ 1 & 1 \end{bmatrix}
\end{array}
\qquad
\begin{array}{c} 0 \\ 1 \\ 1 \\ 0 \end{array}
$$



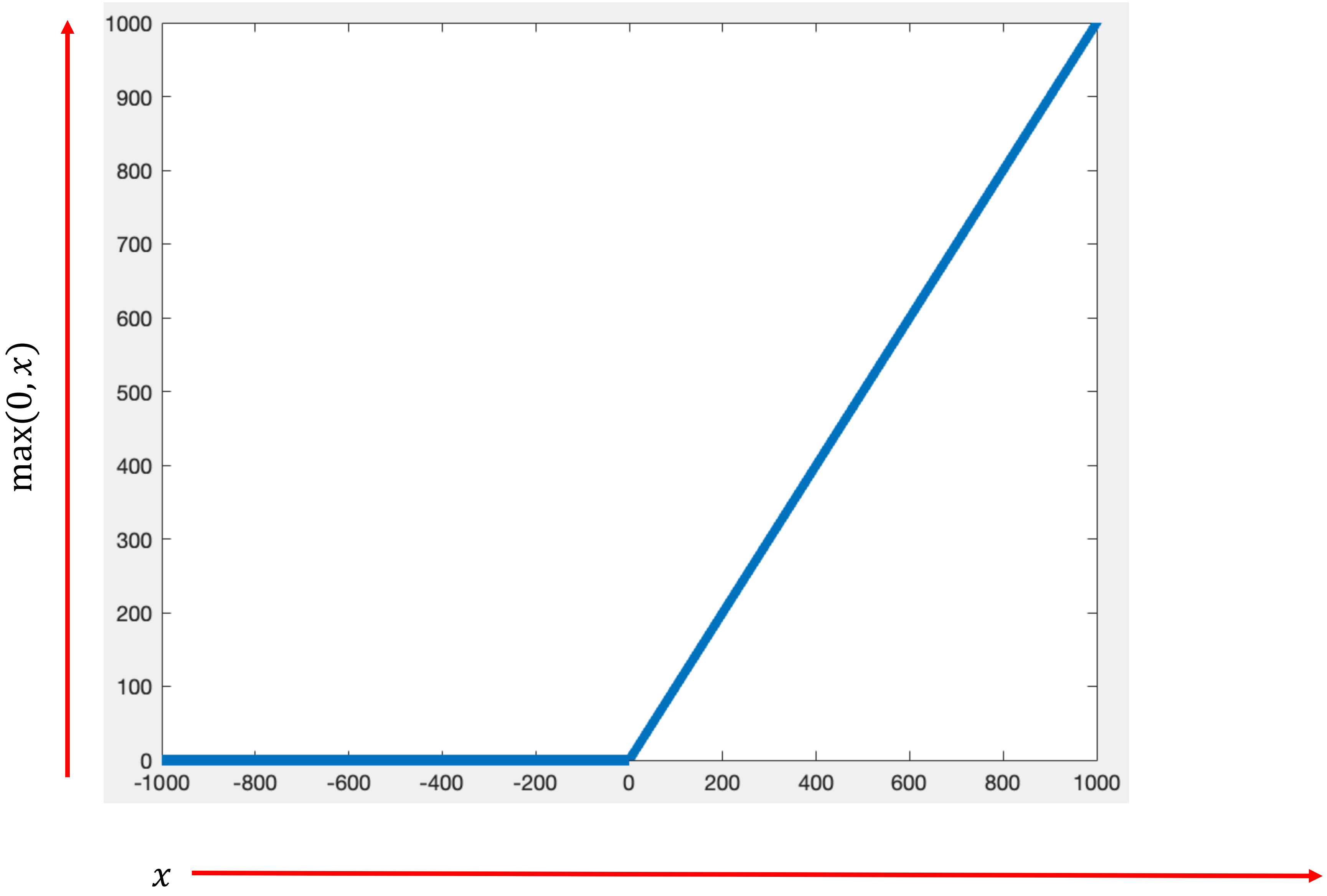
What if we use a nonlinear function as: $f(\boldsymbol{x}; \boldsymbol{W}, \boldsymbol{c}, \boldsymbol{w}, b) = \boldsymbol{w}^\top \max\{0, \boxed{\boldsymbol{W}^\top \boldsymbol{x} + \boldsymbol{c}}\} + b.$

Learning XOR function

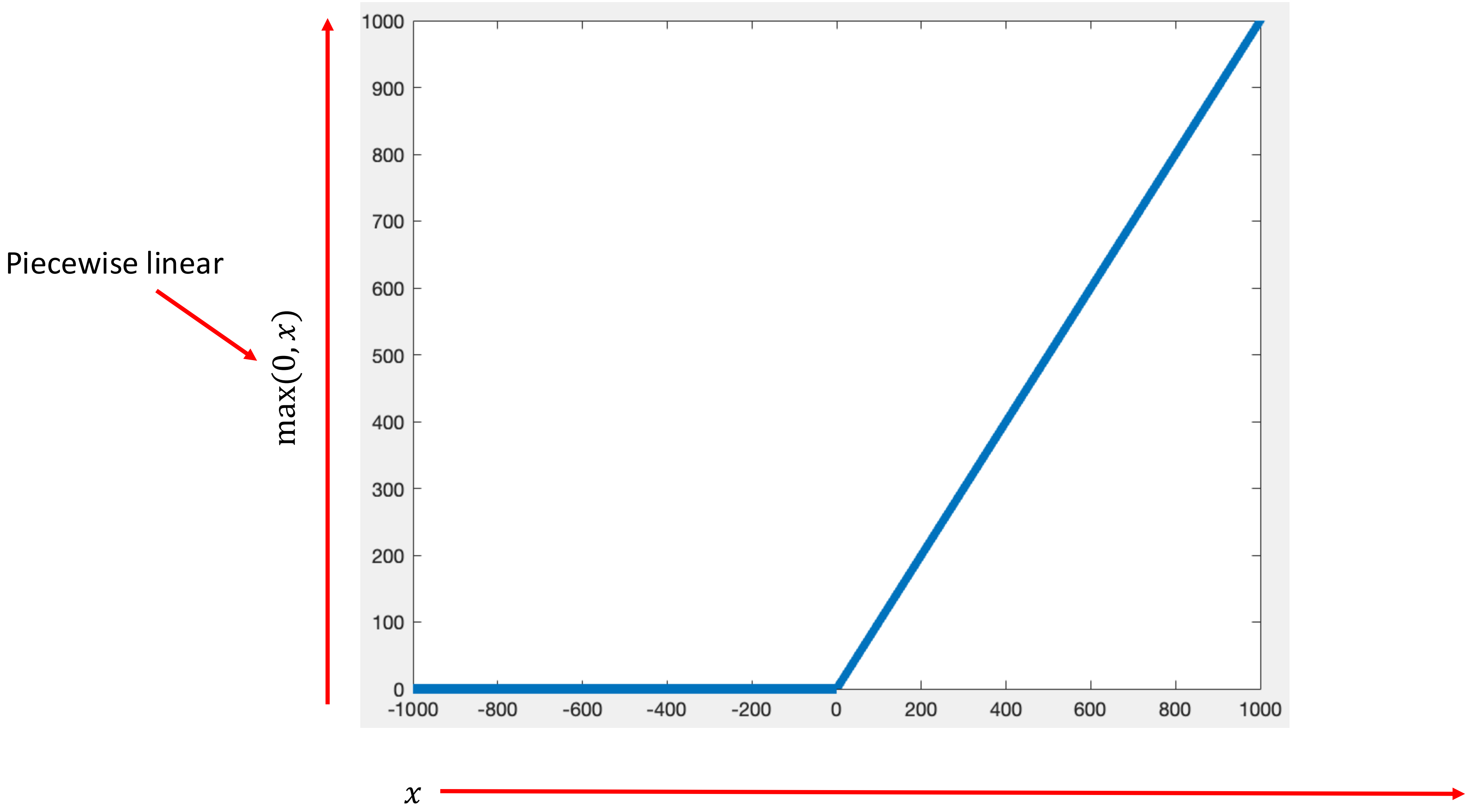
$$
\begin{array}{cc}
 & \begin{array}{cc} x_1 & x_2 \end{array} \\
\boldsymbol{X} = & \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 0 \\ 1 & 1 \end{bmatrix}
\end{array}
\qquad
\begin{array}{c} 0 \\ 1 \\ 1 \\ 0 \end{array}
$$

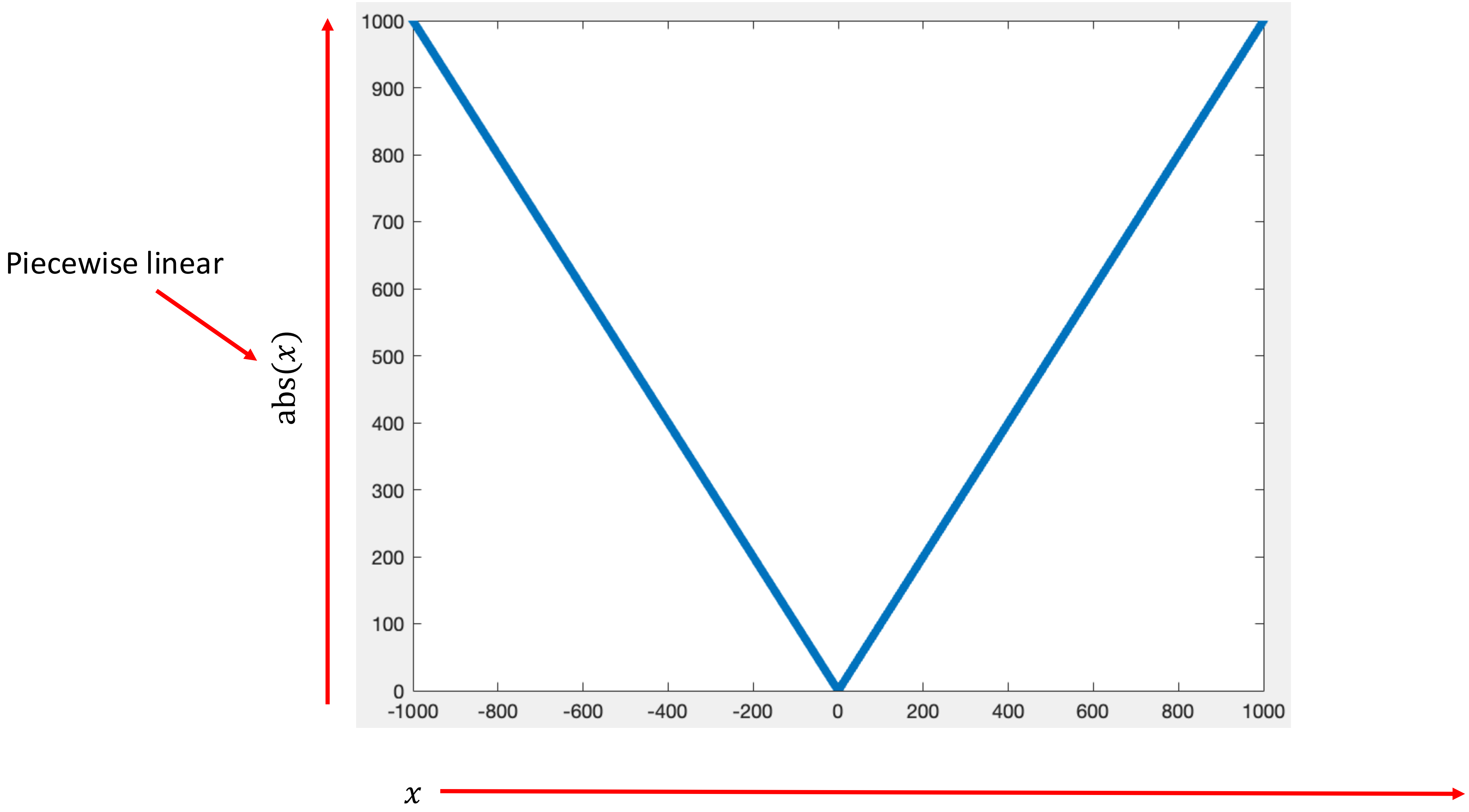


What if we use a nonlinear function as: $f(\boldsymbol{x};\boldsymbol{W},\boldsymbol{c},\boldsymbol{w},b) = \boldsymbol{w}^\top \boxed{\max\{0}, \boldsymbol{W}^\top\boldsymbol{x}+\boldsymbol{c}\}+b.$



max(0, x)
x
1000
900
800
700
600
500
400
300
200
100
0
-1000
-800
-600
-400
-200
0
200
400
600
800
1000

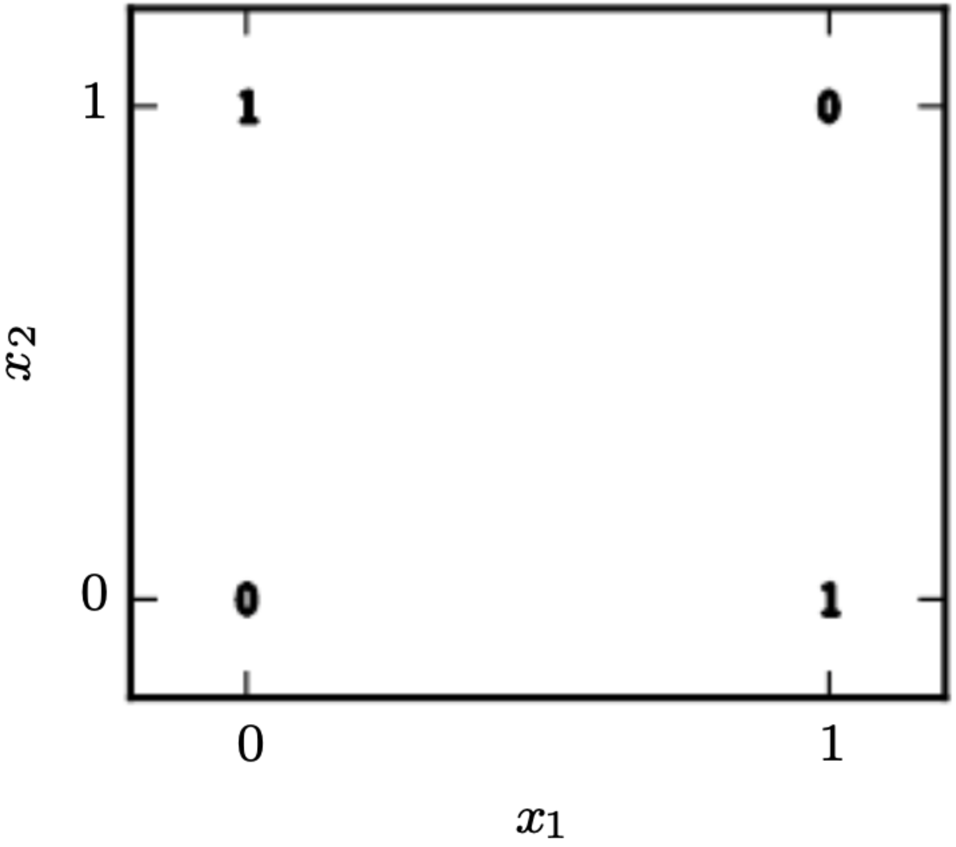

Piecewise linear
max(0, x)
x
1000
900
800
700
600
500
400
300
200
100
0
-1000
-800
-600
-400
-200
0
200
400
600
800
1000



Piecewise linear
abs(x)
x
1000
900
800
700
600
500
400
300
200
100
0
-1000
-800
-600
-400
-200
0
200
400
600
800
1000

Learning XOR function

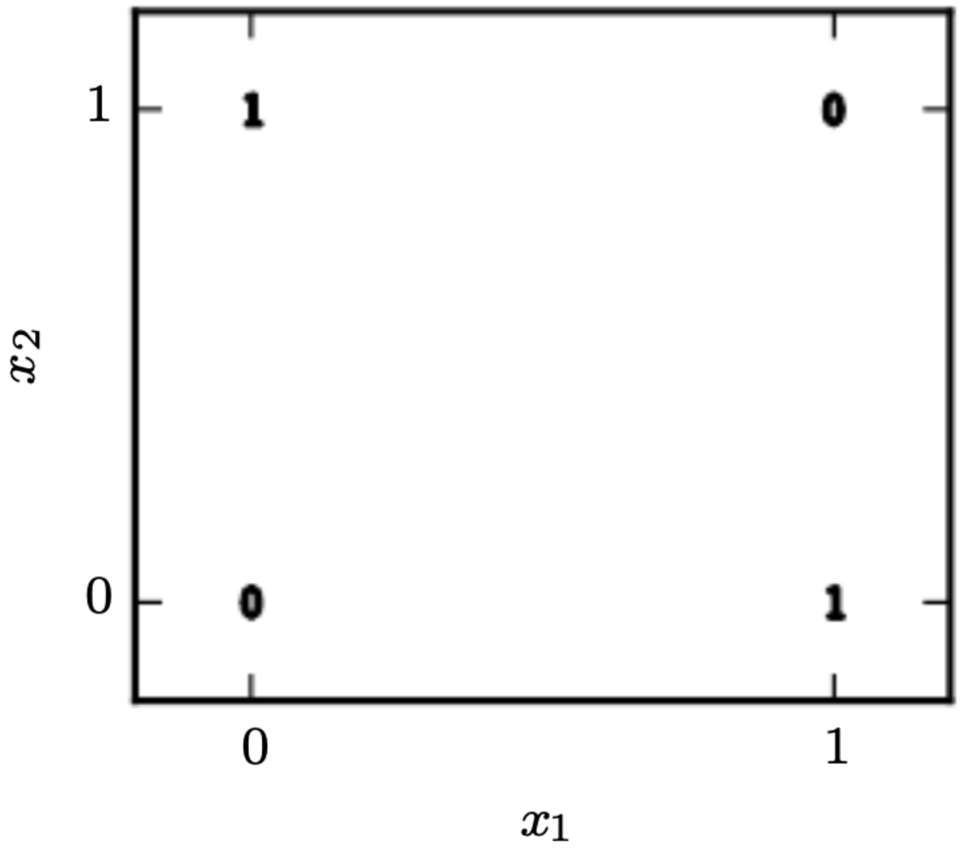
$$\begin{array}{ccc} & \begin{array}{cc} x_1 & x_2 \end{array} & \\ \boldsymbol{X} = & \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 0 \\ 1 & 1 \end{bmatrix} & \begin{array}{c} 0 \\ 1 \\ 1 \\ 0 \end{array} \end{array}$$



What if we use a nonlinear function as: $f(\boldsymbol{x}; \boldsymbol{W}, \boldsymbol{c}, \boldsymbol{w}, b) = \boldsymbol{w}^\top \boxed{\max\{0, \boldsymbol{W}^\top \boldsymbol{x} + \boldsymbol{c}\}} + b.$

Learning XOR function

$$\begin{array}{cc} & \begin{array}{cc} x_1 & x_2 \end{array} \\ \boldsymbol{X} = & \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 0 \\ 1 & 1 \end{bmatrix} \end{array} \quad \begin{array}{c} 0 \\ 1 \\ 1 \\ 0 \end{array}$$

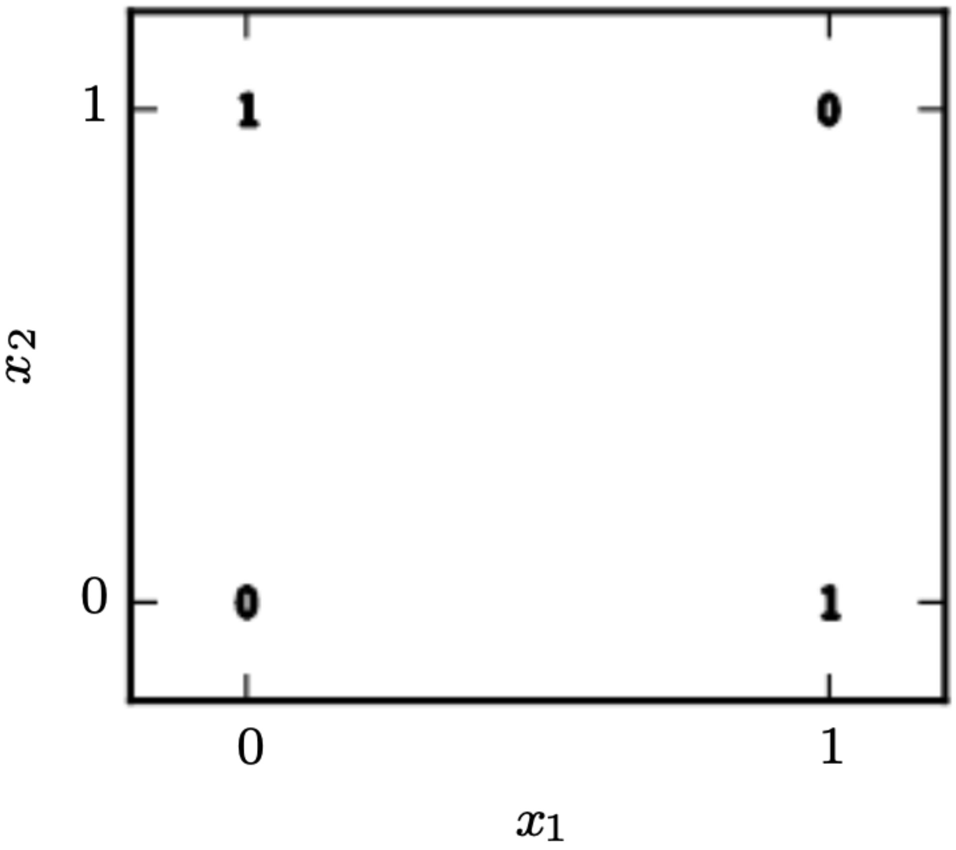


This utput → input of a linear function

What if we use a nonlinear function as: $f(\boldsymbol{x}; \boldsymbol{W}, \boldsymbol{c}, \boldsymbol{w}, b) = \boldsymbol{w}^\top \boxed{\max\{0, \boldsymbol{W}^\top \boldsymbol{x} + \boldsymbol{c}\}} + b.$

Learning XOR function

$$\begin{array}{rcc} & \begin{matrix} x_1 & x_2 \end{matrix} & \\ \boldsymbol{X} = & \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 0 \\ 1 & 1 \end{bmatrix} & \begin{matrix} 0 \\ 1 \\ 1 \\ 0 \end{matrix} \end{array}$$



$$\boldsymbol{W} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix},$$

$$\boldsymbol{c} = \begin{bmatrix} 0 \\ -1 \end{bmatrix},$$

$$\boldsymbol{w} = \begin{bmatrix} 1 \\ -2 \end{bmatrix},$$

What if we use a nonlinear function as: $f(\boldsymbol{x}; \boldsymbol{W}, \boldsymbol{c}, \boldsymbol{w}, b) = \boldsymbol{w}^\top \max\{0, \boldsymbol{W}^\top \boldsymbol{x} + \boldsymbol{c}\} + \cancel{b}$

Learning XOR function

$$\begin{array}{ll} & \begin{array}{cc} x_1 & x_2 \end{array} \\ \boxed{\boldsymbol{X} =} & \boxed{\begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 0 \\ 1 & 1 \end{bmatrix}} \end{array} \quad \begin{matrix} 0 \\ 1 \\ 1 \\ 0 \end{matrix} \quad \rightarrow \boldsymbol{XW} = \begin{bmatrix} 0 & 0 \\ 1 & 1 \\ 1 & 1 \\ 2 & 2 \end{bmatrix}$$

$$\boxed{\boldsymbol{W} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}},$$

$$\boldsymbol{c} = \begin{bmatrix} 0 \\ -1 \end{bmatrix},$$

$$\boldsymbol{w} = \begin{bmatrix} 1 \\ -2 \end{bmatrix},$$

What if we use a nonlinear function as: $f(\boldsymbol{x}; \boldsymbol{W}, \boldsymbol{c}, \boldsymbol{w}, b) = \boldsymbol{w}^\top \max\{0, \boxed{\boldsymbol{W}^\top \boldsymbol{x}} + \boldsymbol{c}\} + \cancel{b}$

Learning XOR function

$$\boldsymbol{X} = \begin{array}{c} \begin{matrix} x_1 & x_2 \end{matrix} \\ \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 0 \\ 1 & 1 \end{bmatrix} \end{array} \quad \begin{matrix} 0 \\ 1 \\ 1 \\ 0 \end{matrix} \quad \rightarrow \boldsymbol{X}\boldsymbol{W} = \begin{bmatrix} \boxed{0} & 0 \\ 1 & 1 \\ 1 & 1 \\ 2 & 2 \end{bmatrix}$$

$$\boldsymbol{W} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix},$$

$$\boldsymbol{c} = \begin{bmatrix} 0 \\ -1 \end{bmatrix},$$

$$\boldsymbol{w} = \begin{bmatrix} 1 \\ -2 \end{bmatrix},$$

What if we use a nonlinear function as: $f(\boldsymbol{x}; \boldsymbol{W}, \boldsymbol{c}, \boldsymbol{w}, b) = \boldsymbol{w}^\top \max\{0, \boxed{\boldsymbol{W}^\top \boldsymbol{x}} + \boldsymbol{c}\} + \cancel{b}$

Learning XOR function

$$
\boldsymbol{X} = \begin{array}{c} \begin{array}{cc} x_1 & x_2 \end{array} \\ \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 0 \\ 1 & 1 \end{bmatrix} \end{array} \quad \begin{array}{c} 0 \\ 1 \\ 1 \\ 0 \end{array} \quad \longrightarrow \boldsymbol{XW} = \begin{bmatrix} 0 & \boxed{0} \\ 1 & 1 \\ 1 & 1 \\ 2 & 2 \end{bmatrix}
$$

$$
\boldsymbol{W} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix},
$$

$$
\boldsymbol{c} = \begin{bmatrix} 0 \\ -1 \end{bmatrix},
$$

$$
\boldsymbol{w} = \begin{bmatrix} 1 \\ -2 \end{bmatrix},
$$

What if we use a nonlinear function as: $f(\boldsymbol{x}; \boldsymbol{W}, \boldsymbol{c}, \boldsymbol{w}, b) = \boldsymbol{w}^\top \max\{0, \boxed{\boldsymbol{W}^\top \boldsymbol{x}} + \boldsymbol{c}\} + \cancel{b}$

Learning XOR function

$$\boldsymbol{X} = \begin{array}{c} \begin{array}{cc} x_1 & x_2 \end{array} \\ \begin{bmatrix} 0 & 0 \\ \boxed{0 \quad 1} \\ 1 & 0 \\ 1 & 1 \end{bmatrix} \end{array} \quad \begin{matrix} 0 \\ 1 \\ 1 \\ 0 \end{matrix} \quad \longrightarrow \boldsymbol{XW} = \begin{bmatrix} 0 & 0 \\ \boxed{1} & 1 \\ 1 & 1 \\ 2 & 2 \end{bmatrix}$$

$$\boldsymbol{W} = \begin{bmatrix} \boxed{\begin{matrix} 1 \\ 1 \end{matrix}} & \begin{matrix} 1 \\ 1 \end{matrix} \end{bmatrix},$$

$$\boldsymbol{c} = \begin{bmatrix} 0 \\ -1 \end{bmatrix},$$

$$\boldsymbol{w} = \begin{bmatrix} 1 \\ -2 \end{bmatrix},$$

What if we use a nonlinear function as: $f(\boldsymbol{x}; \boldsymbol{W}, \boldsymbol{c}, \boldsymbol{w}, b) = \boldsymbol{w}^\top \max\{0, \boxed{\boldsymbol{W}^\top \boldsymbol{x}} + \boldsymbol{c}\} + \cancel{b}$

Learning XOR function

$$\boldsymbol{X} = \begin{array}{c} \begin{array}{cc} x_1 & x_2 \end{array} \\ \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 0 \\ 1 & 1 \end{bmatrix} \end{array} \quad \begin{array}{c} 0 \\ 1 \\ 1 \\ 0 \end{array} \quad \rightarrow \boldsymbol{XW} = \begin{bmatrix} 0 & 0 \\ 1 & \boxed{1} \\ 1 & 1 \\ 2 & 2 \end{bmatrix}$$

$$\boldsymbol{W} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix},$$

$$\boldsymbol{c} = \begin{bmatrix} 0 \\ -1 \end{bmatrix},$$

$$\boldsymbol{w} = \begin{bmatrix} 1 \\ -2 \end{bmatrix},$$

What if we use a nonlinear function as: $f(\boldsymbol{x}; \boldsymbol{W}, \boldsymbol{c}, \boldsymbol{w}, b) = \boldsymbol{w}^\top \max\{0, \boxed{\boldsymbol{W}^\top \boldsymbol{x}} + \boldsymbol{c}\} + \cancel{b}$

Learning XOR function

$$\begin{array}{cc} & \begin{array}{cc} x_1 & x_2 \end{array} \\ \boldsymbol{X} = & \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 0 \\ 1 & 1 \end{bmatrix} \end{array} \quad \begin{array}{c} 0 \\ 1 \\ 1 \\ 0 \end{array} \quad \rightarrow \boldsymbol{XW} = \begin{bmatrix} 0 & 0 \\ 1 & 1 \\ 1 & 1 \\ 2 & 2 \end{bmatrix} + \boldsymbol{c} \rightarrow$$

$$\boldsymbol{W} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix},$$

$$\boxed{\boldsymbol{c} = \begin{bmatrix} 0 \\ -1 \end{bmatrix}},$$

$$\boldsymbol{w} = \begin{bmatrix} 1 \\ -2 \end{bmatrix},$$

What if we use a nonlinear function as: $f(\boldsymbol{x}; \boldsymbol{W}, \boldsymbol{c}, \boldsymbol{w}, b) = \boldsymbol{w}^\top \max\{0, \boxed{\boldsymbol{W}^\top \boldsymbol{x} + \boldsymbol{c}}\} + \cancel{b}$

Learning XOR function

$$\boldsymbol{X} = \begin{array}{c} \begin{array}{cc} x_1 & x_2 \end{array} \\ \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 0 \\ 1 & 1 \end{bmatrix} \end{array} \qquad \begin{array}{c} 0 \\ 1 \\ 1 \\ 0 \end{array} \quad \rightarrow \quad \boldsymbol{X}\boldsymbol{W} = \begin{bmatrix} \boxed{\begin{matrix} 0 & 0 \end{matrix}} \\ \begin{matrix} 1 & 1 \end{matrix} \\ \begin{matrix} 1 & 1 \end{matrix} \\ \begin{matrix} 2 & 2 \end{matrix} \end{bmatrix} \begin{array}{c} + \boldsymbol{c} \\ \rightarrow \end{array}$$

$$\boldsymbol{W} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix},$$

$$\boldsymbol{c} = \begin{bmatrix} \boxed{\begin{matrix} 0 \\ -1 \end{matrix}} \end{bmatrix},$$

$$\boldsymbol{w} = \begin{bmatrix} 1 \\ -2 \end{bmatrix},$$

What if we use a nonlinear function as: $f(\boldsymbol{x}; \boldsymbol{W}, \boldsymbol{c}, \boldsymbol{w}, b) = \boldsymbol{w}^\top \max\{0, \boxed{\boldsymbol{W}^\top \boldsymbol{x} + \boldsymbol{c}}\} + \cancel{b}$

Learning XOR function

$$
\boldsymbol{X} = \begin{array}{c} \begin{matrix} x_1 & x_2 \end{matrix} \\ \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 0 \\ 1 & 1 \end{bmatrix} \end{array} \quad \begin{matrix} 0 \\ 1 \\ 1 \\ 0 \end{matrix}
$$

$$
\boldsymbol{X}\boldsymbol{W} = \begin{bmatrix} 0 & 0 \\ \boxed{1 \quad 1} \\ 1 & 1 \\ 2 & 2 \end{bmatrix} + \boldsymbol{c}
$$

$$
\boldsymbol{W} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix},
$$

$$
\boldsymbol{c} = \begin{bmatrix} \boxed{\begin{matrix} 0 \\ -1 \end{matrix}} \end{bmatrix},
$$

$$
\boldsymbol{w} = \begin{bmatrix} 1 \\ -2 \end{bmatrix},
$$

What if we use a nonlinear function?

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

Learning XOR function

$$\boldsymbol{X} = \begin{array}{c} \begin{array}{cc} x_1 & x_2 \end{array} \\ \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 0 \\ 1 & 1 \end{bmatrix} \end{array} \quad \begin{array}{c} 0 \\ 1 \\ 1 \\ 0 \end{array} \rightarrow \boldsymbol{X}\boldsymbol{W} = \begin{bmatrix} 0 & 0 \\ 1 & 1 \\ 1 & 1 \\ 2 & 2 \end{bmatrix} \xrightarrow{+\boldsymbol{c}} \boxed{\begin{bmatrix} 0 & -1 \\ 1 & 0 \\ 1 & 0 \\ 2 & 1 \end{bmatrix}}$$

$$\boldsymbol{W} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix},$$

$$\boxed{\boldsymbol{c} = \begin{bmatrix} 0 \\ -1 \end{bmatrix}},$$

$$\boldsymbol{w} = \begin{bmatrix} 1 \\ -2 \end{bmatrix},$$

What if we use a nonlinear function? $f(\boldsymbol{x}; \boldsymbol{W}, \boldsymbol{c}, \boldsymbol{w}, b) = \boldsymbol{w}^\top \max\{0, \boxed{\boldsymbol{W}^\top \boldsymbol{x} + \boldsymbol{c}}\} + \cancel{b}$

Learning XOR function

$$\boldsymbol{X} = \begin{array}{c} \begin{array}{cc} x_1 & x_2 \end{array} \\ \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 0 \\ 1 & 1 \end{bmatrix} \end{array} \quad \begin{array}{c} 0 \\ 1 \\ 1 \\ 0 \end{array} \rightarrow \boldsymbol{X}\boldsymbol{W} = \begin{bmatrix} 0 & 0 \\ 1 & 1 \\ 1 & 1 \\ 2 & 2 \end{bmatrix} \xrightarrow{+\boldsymbol{c}} \boxed{\begin{bmatrix} 0 & -1 \\ 1 & 0 \\ 1 & 0 \\ 2 & 1 \end{bmatrix}}$$

$$\boldsymbol{W} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix},$$

$$\boxed{\boldsymbol{c} = \begin{bmatrix} 0 \\ -1 \end{bmatrix}},$$

$$\boldsymbol{w} = \begin{bmatrix} 1 \\ -2 \end{bmatrix},$$

What if we use a nonlinear function as: $f(\boldsymbol{x}; \boldsymbol{W}, \boldsymbol{c}, \boldsymbol{w}, b) = \boldsymbol{w}^\top \max\{0, \overbrace{\boxed{\boldsymbol{W}^\top \boldsymbol{x} + \boldsymbol{c}}}^{\text{The inner linear model}}\} + \cancel{b}$

Learning XOR function

$$\boldsymbol{X} = \begin{array}{c} \begin{matrix} x_1 & x_2 \end{matrix} \\ \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 0 \\ 1 & 1 \end{bmatrix} \end{array} \quad \begin{matrix} 0 \\ 1 \\ 1 \\ 0 \end{matrix} \rightarrow \boldsymbol{XW} = \begin{bmatrix} 0 & 0 \\ 1 & 1 \\ 1 & 1 \\ 2 & 2 \end{bmatrix} \xrightarrow{+\boldsymbol{c}} \boxed{\begin{bmatrix} 0 & -1 \\ 1 & 0 \\ 1 & 0 \\ 2 & 1 \end{bmatrix}}$$

$$\Big\downarrow \max(0, \cdot)$$

$$\boxed{\begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 1 & 0 \\ 2 & 1 \end{bmatrix}}$$

$$\boldsymbol{W} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}, \quad \boldsymbol{c} = \begin{bmatrix} 0 \\ -1 \end{bmatrix}, \quad \boldsymbol{w} = \begin{bmatrix} 1 \\ -2 \end{bmatrix},$$

What if we use a nonlinear function as: $f(\boldsymbol{x}; \boldsymbol{W}, \boldsymbol{c}, \boldsymbol{w}, b) = \boldsymbol{w}^\top \boxed{\max\{0, \boldsymbol{W}^\top \boldsymbol{x} + \boldsymbol{c}\}} + \cancel{b}$

Learning XOR function

$$\boldsymbol{X} = \begin{array}{cc} x_1 & x_2 \\ \end{array}$$
$$\boldsymbol{X} = \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 0 \\ 1 & 1 \end{bmatrix} \quad \begin{matrix} 0 \\ 1 \\ 1 \\ 0 \end{matrix} \rightarrow \boldsymbol{XW} = \begin{bmatrix} 0 & 0 \\ 1 & 1 \\ 1 & 1 \\ 2 & 2 \end{bmatrix} \xrightarrow{+\boldsymbol{c}} \begin{bmatrix} 0 & -1 \\ 1 & 0 \\ 1 & 0 \\ 2 & 1 \end{bmatrix}$$

$$\boldsymbol{W} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix},$$

$$\boldsymbol{c} = \begin{bmatrix} 0 \\ -1 \end{bmatrix},$$

$$\boxed{\boldsymbol{w} = \begin{bmatrix} 1 \\ -2 \end{bmatrix}},$$

$$\begin{bmatrix} 0 & -1 \\ 1 & 0 \\ 1 & 0 \\ 2 & 1 \end{bmatrix} \xrightarrow{\max(0,\, \cdot)} \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 1 & 0 \\ 2 & 1 \end{bmatrix} \xrightarrow{\boldsymbol{w}^\top} \boxed{\begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}}$$

What if we use a nonlinear function as: $f(\boldsymbol{x}; \boldsymbol{W}, \boldsymbol{c}, \boldsymbol{w}, b) = \boxed{\boldsymbol{w}^\top \max\{0, \boldsymbol{W}^\top \boldsymbol{x} + \boldsymbol{c}\}} + \cancel{b}$

Learning XOR function

$$\boldsymbol{X} = \begin{array}{c} \begin{array}{cc} x_1 & x_2 \end{array} \\ \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 0 \\ 1 & 1 \end{bmatrix} \end{array} \quad \boxed{\begin{matrix} 0 \\ 1 \\ 1 \\ 0 \end{matrix}}$$

$$\boldsymbol{X}\boldsymbol{W} = \begin{bmatrix} 0 & 0 \\ 1 & 1 \\ 1 & 1 \\ 2 & 2 \end{bmatrix} \xrightarrow{+\boldsymbol{c}} \begin{bmatrix} 0 & -1 \\ 1 & 0 \\ 1 & 0 \\ 2 & 1 \end{bmatrix}$$

max(0, ·)

$$\begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 1 & 0 \\ 2 & 1 \end{bmatrix} \xrightarrow{\boldsymbol{w}^\top} \boxed{\begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}}$$

$$\boldsymbol{W} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix},$$

$$\boldsymbol{c} = \begin{bmatrix} 0 \\ -1 \end{bmatrix},$$

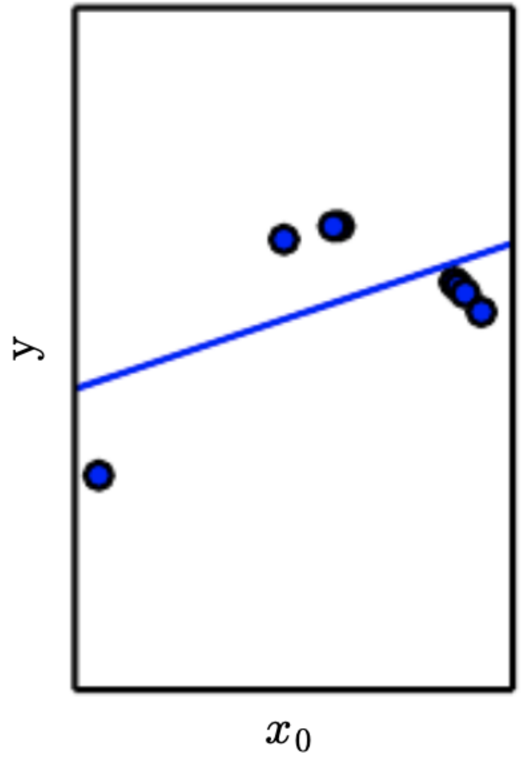
$$\boxed{\boldsymbol{w} = \begin{bmatrix} 1 \\ -2 \end{bmatrix}},$$

What if we use a nonlinear function as: $f(\boldsymbol{x}; \boldsymbol{W}, \boldsymbol{c}, \boldsymbol{w}, b) = \boldsymbol{w}^\top \max\{0, \boldsymbol{W}^\top \boldsymbol{x} + \boldsymbol{c}\} + \cancel{b}$

Learning XOR function

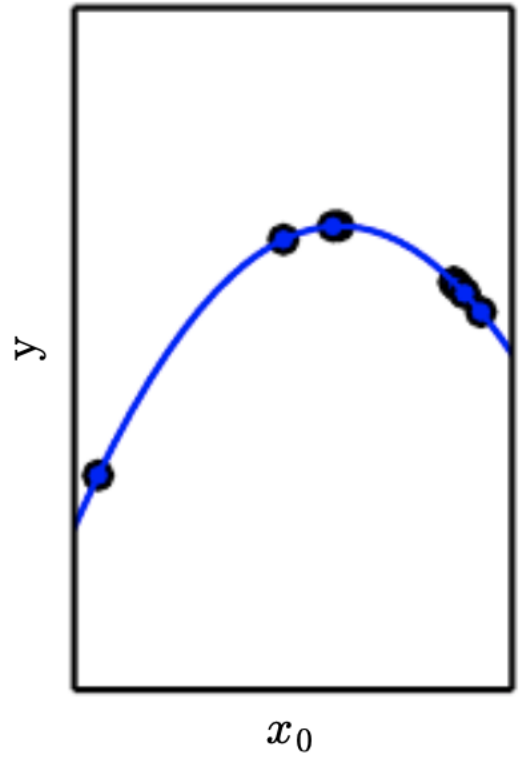
$\boldsymbol{W}^\top \boldsymbol{x} + \boldsymbol{c}$ v.s. $\boldsymbol{w}^\top \max\{0, \boldsymbol{W}^\top \boldsymbol{x} + \boldsymbol{c}\}$

Learning XOR function



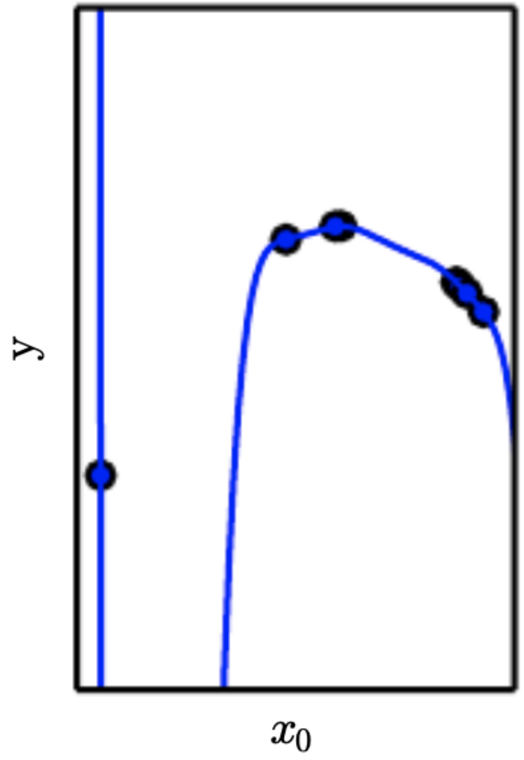
Linear model

$$f(w;x) = w_1 x^1 + w_0$$



Quadratic model

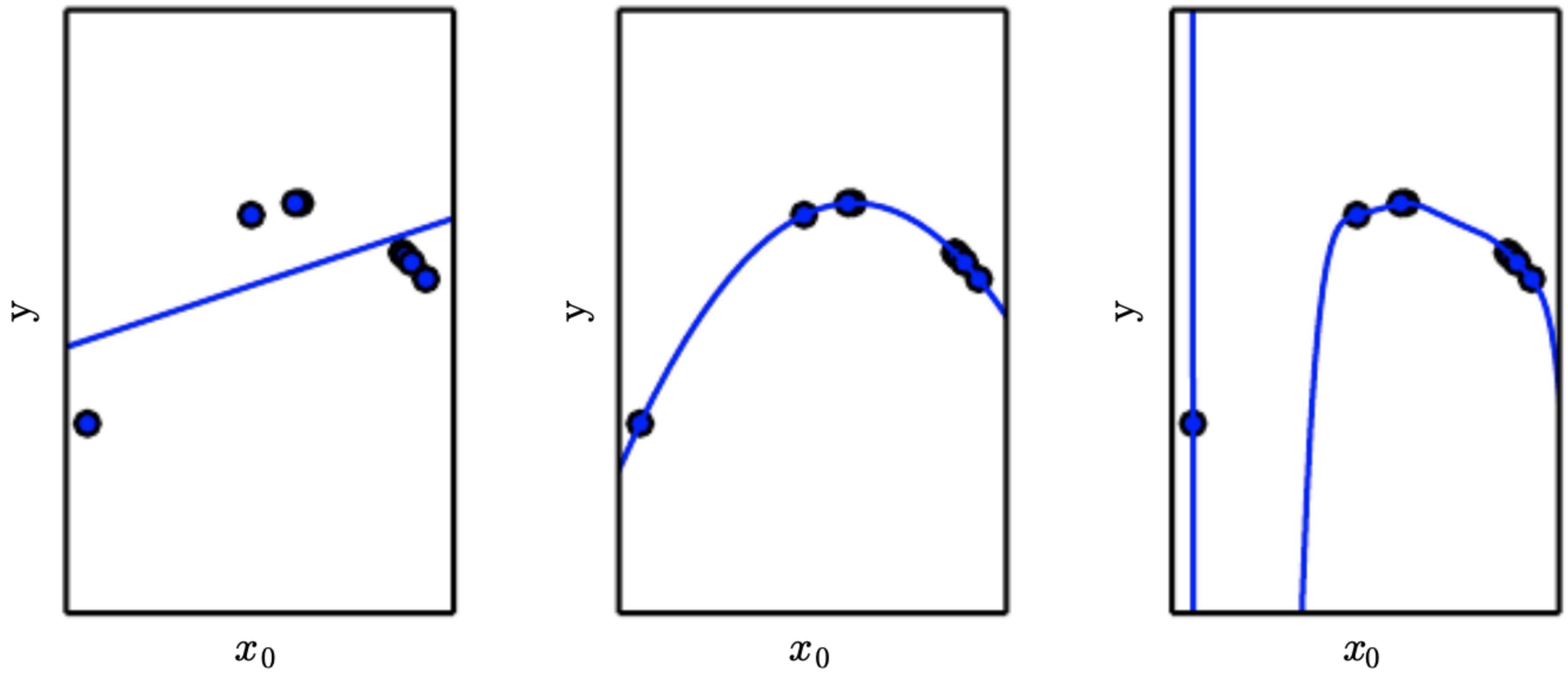
$$f(w;x) = w_2 x^2 + w_1 x^1 + w_0$$



Polynomial model (9 degree)

$$f(w;x) = \sum_{\mathrm{i}=1}^{9} w_i x^i + w_0$$

Learning XOR function



Linear model

$$f(w;x) = w_1 x^1 + w_0$$

$$= \sum_{i=2}^{9} 0 \cdot x^i + \sum_{i=1}^{1} w_i x^i + w_0$$

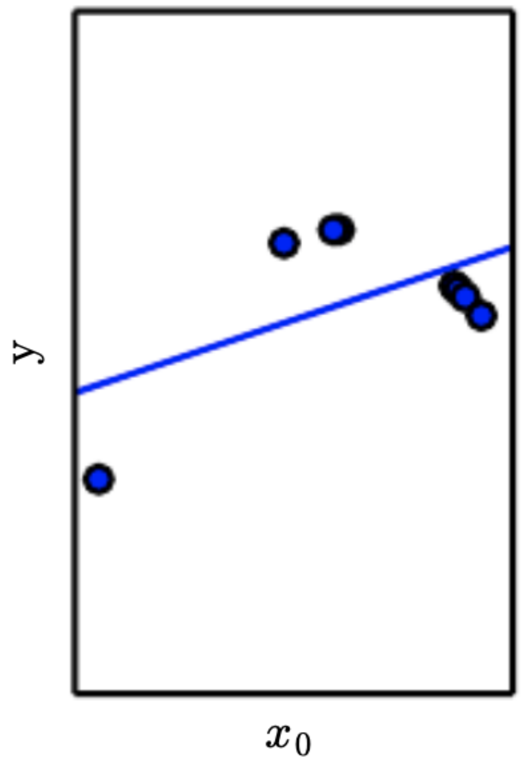
Quadratic model

$$f(w;x) = w_2 x^2 + w_1 x^1 + w_0$$

Polynomial model (9 degree)

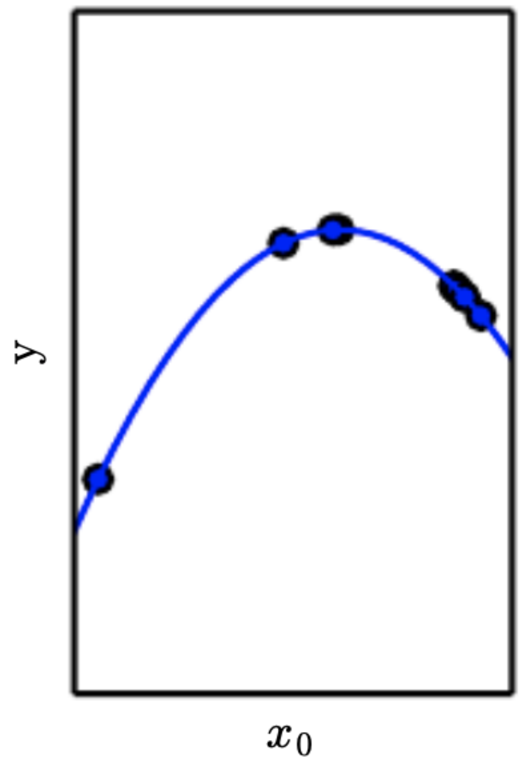
$$f(w;x) = \sum_{i=1}^{9} w_i x^i + w_0$$

Learning XOR function



Linear model

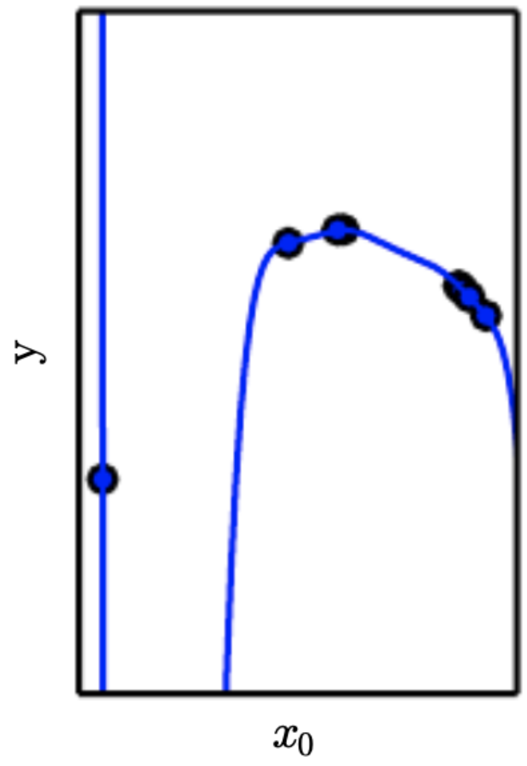
$$f(w;x) = w_1 x^1 + w_0$$



Quadratic model

$$f(w;x) = w_2 x^2 + w_1 x^1 + w_0$$

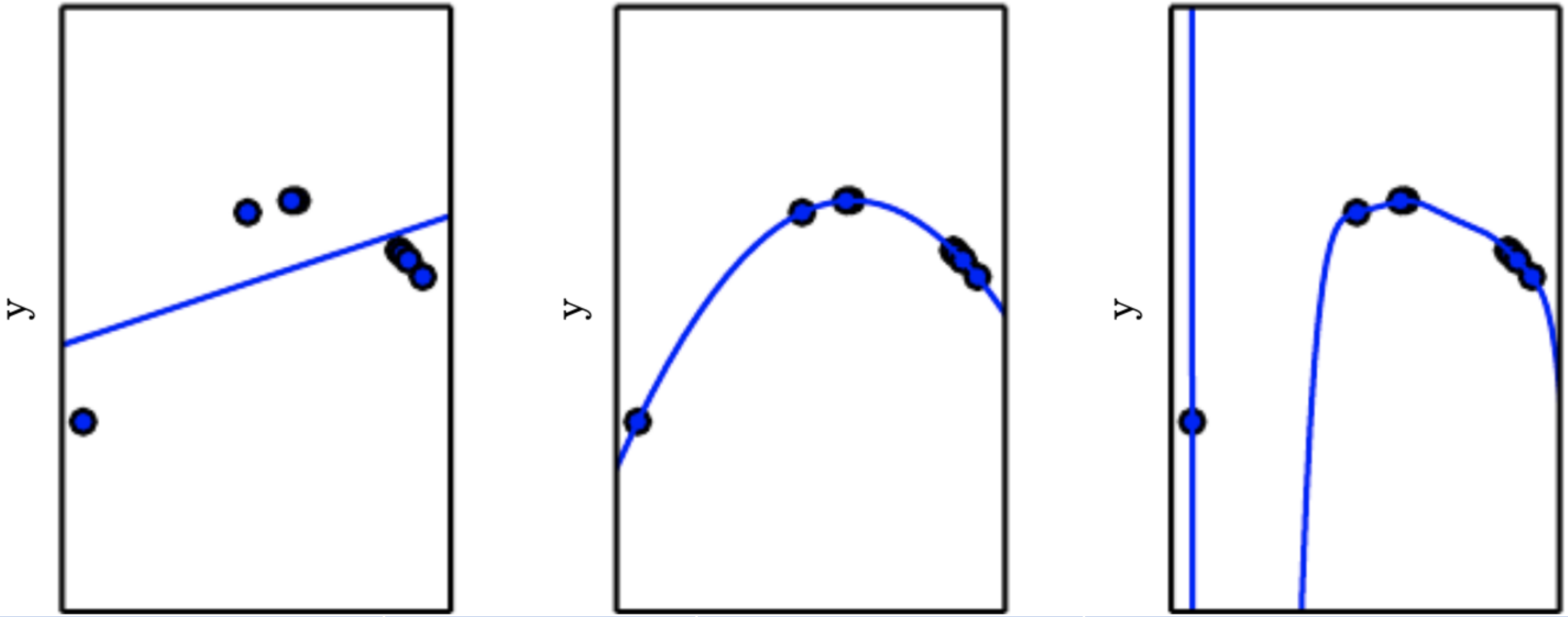
$$= \sum_{i=3}^{9} 0 \cdot x^i + \sum_{i=1}^{2} w_i x^i + w_0$$



Polynomial model (9 degree)

$$f(w;x) = \sum_{i=1}^{9} w_i x^i + w_0$$

Learning XOR function



	Linear function	Quadratic function	9-degree polynomial function
Linear function	Yes	No	No
Quadratic function	Yes	Yes	No
9-degree polynomial function	Yes	Yes	Yes

Best capacity

Learning XOR function

$$\boldsymbol{W}^\top \boldsymbol{x} + \boldsymbol{c} \qquad \text{v.s.} \qquad \boldsymbol{w}^\top \max\{0, \boldsymbol{W}^\top \boldsymbol{x} + \boldsymbol{c}\}$$

nonlinear model: better capacity

	Linear function	Quadratic function	9-degree polynomial function
Linear function	Yes	No	No
Quadratic function	Yes	Yes	No
9-degree polynomial function	Yes	Yes	Yes

Best capacity

Learning XOR function

$$\boldsymbol{W}^\top \boldsymbol{x} + \boldsymbol{c}$$

Raw features

V.S.

$$\boldsymbol{w}^\top \max\{0, \boldsymbol{W}^\top \boldsymbol{x} + \boldsymbol{c}\}$$

nonlinear model: better capacity
(Activation layer)

Learning XOR function

$$\boldsymbol{W}^\top \boldsymbol{x} + \boldsymbol{c}$$

Raw features

V.S.

$$\boldsymbol{w}^\top \max\{0, \boldsymbol{W}^\top \boldsymbol{x} + \boldsymbol{c}\}$$

nonlinear model: better capacity

Raw features

Learning XOR function

$$\boldsymbol{W}^\top \boldsymbol{x} + \boldsymbol{c}$$

Raw features

V.S.

$$\boldsymbol{w}^\top \boxed{\max\{0, \boldsymbol{W}^\top \boldsymbol{x} + \boldsymbol{c}\}}$$

nonlinear model: better capacity

Raw features

Learned features ←output of inner function

Learning XOR function

$$\boldsymbol{W}^\top \boldsymbol{x} + \boldsymbol{c}$$

Raw features

v.s.

$$\boldsymbol{w}^\top \boxed{\max\{0, \boldsymbol{W}^\top \boldsymbol{x} + \boldsymbol{c}\}}$$

nonlinear model: better capacity

Raw features

Learned features ←output of inner function

If W can be learned (determined by training data)

What makes feedforward network different from linear model

- Nonlinear functions in hidden layers

$$f_3\left(f_2\left(f_1(x)\right)\right) = \boldsymbol{w}^\top \max\{0, \boxed{\boldsymbol{W}^\top \boldsymbol{x} + \boldsymbol{c}}\}$$

$$f_1(x) = W'x + c$$

What makes feedforward network different from linear model

- Nonlinear functions in hidden layers

$$f_3\big(f_2(f_1(x))\big) = \boldsymbol{w}^\top \boxed{\max\{0, \boldsymbol{W}^\top \boldsymbol{x} + \boldsymbol{c}\}}$$

$$f_2(x) = \max(0, x)$$

What makes feedforward network different from linear model

- Nonlinear functions in hidden layers

$$f_3\left(f_2\left(f_1(x)\right)\right) = \boxed{\boldsymbol{w}^\top \max\{0, \boldsymbol{W}^\top \boldsymbol{x} + \boldsymbol{c}\}}$$

$$f_3(x) = \mathrm{w}'\mathrm{x}$$

What makes feedforward network different from linear model

- Nonlinear functions in hidden layers

$$f_3\Big(f_2\big(f_1(x)\big)\Big) = \boldsymbol{w}^\top \max\{0, \boldsymbol{W}^\top \boldsymbol{x} + \boldsymbol{c}\}$$

- Q: Why composition makes nonconvexity?

What makes feedforward network different from linear model

- Nonlinear functions in hidden layers

$$f_3\Big(f_2\big(f_1(x)\big)\Big) = \boldsymbol{w}^\top \max\{0, \boldsymbol{W}^\top \boldsymbol{x} + \boldsymbol{c}\}$$

- Q: Why composition makes nonconvexity?

$f(x) = \mathrm{g}(\mathrm{h}(\mathrm{x}))$ where $\mathrm{g}(\mathrm{x}) = \mathrm{h}(\mathrm{x}) = x^2$

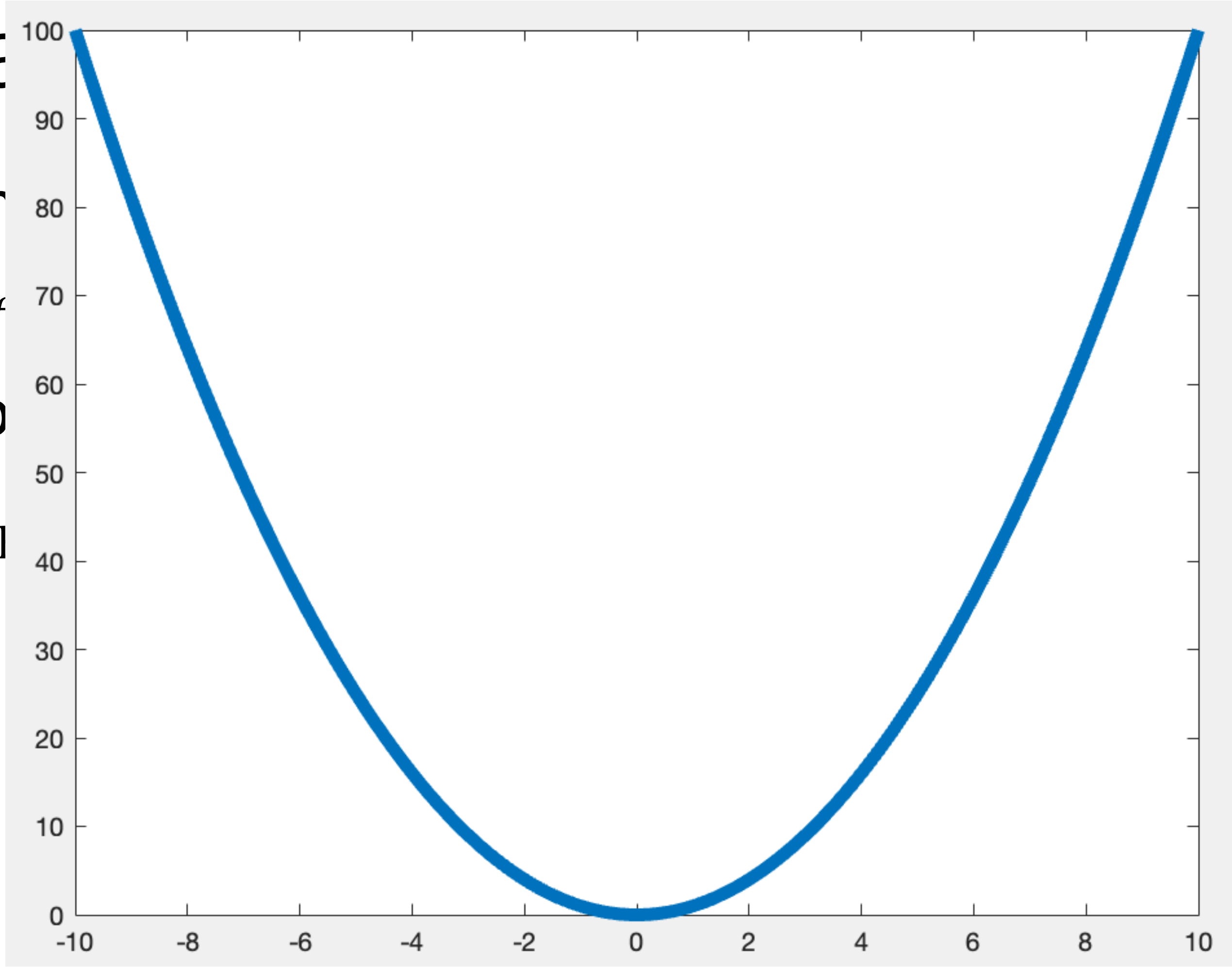
What makes feedforward network different from linea

- Nonlinear fur

$f_3\left(f\right.$

- Q: Why comp

$f(x) = \mathrm{g}($



$f(\mathrm{x}) = x^2$

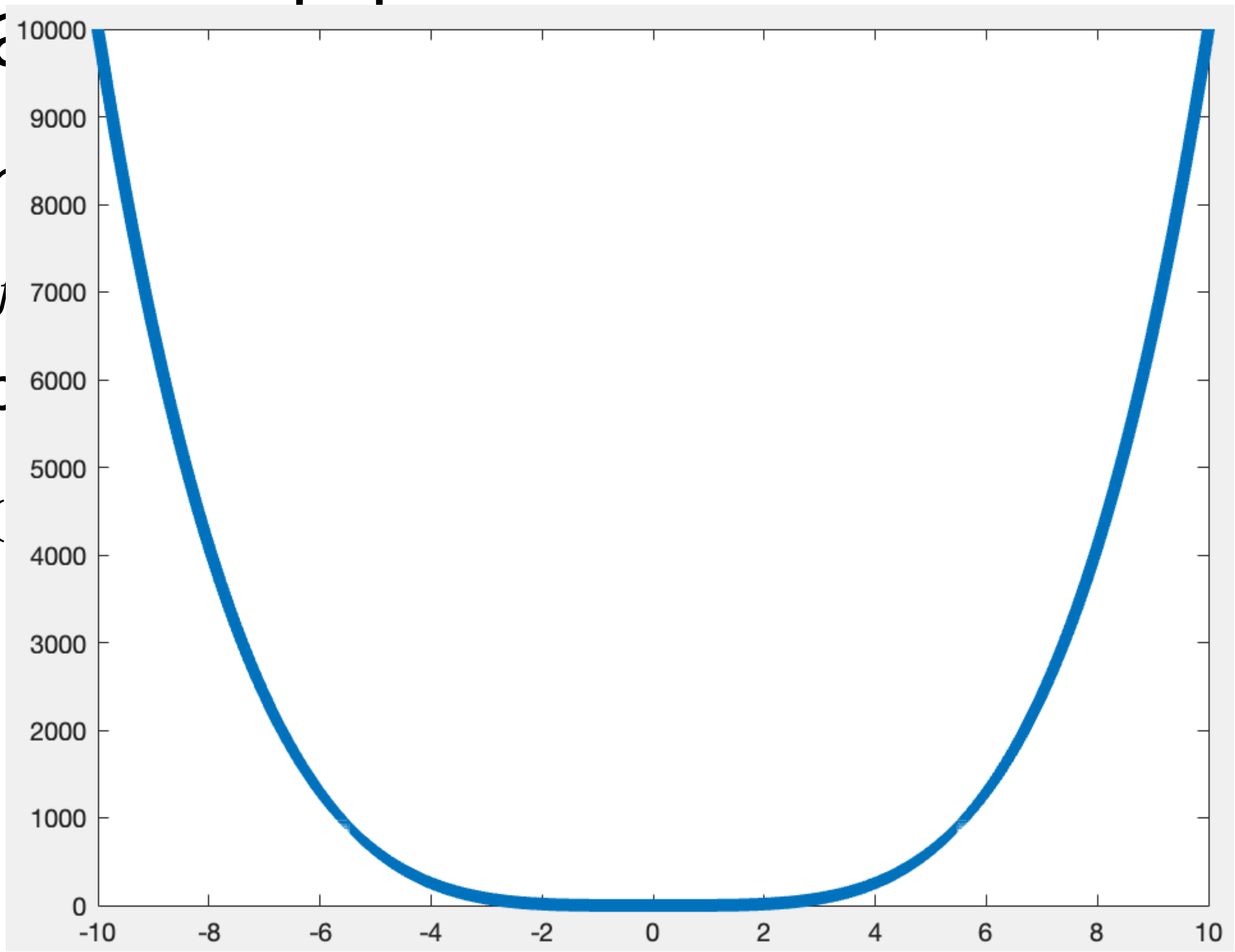
What makes feedforward network different from linea

- Nonlinear fur

$f_3($

- Q: Why comp

$f(x) = g($



$f(x) = g(h(x))$ where $g(x) = h(x) = x^2$

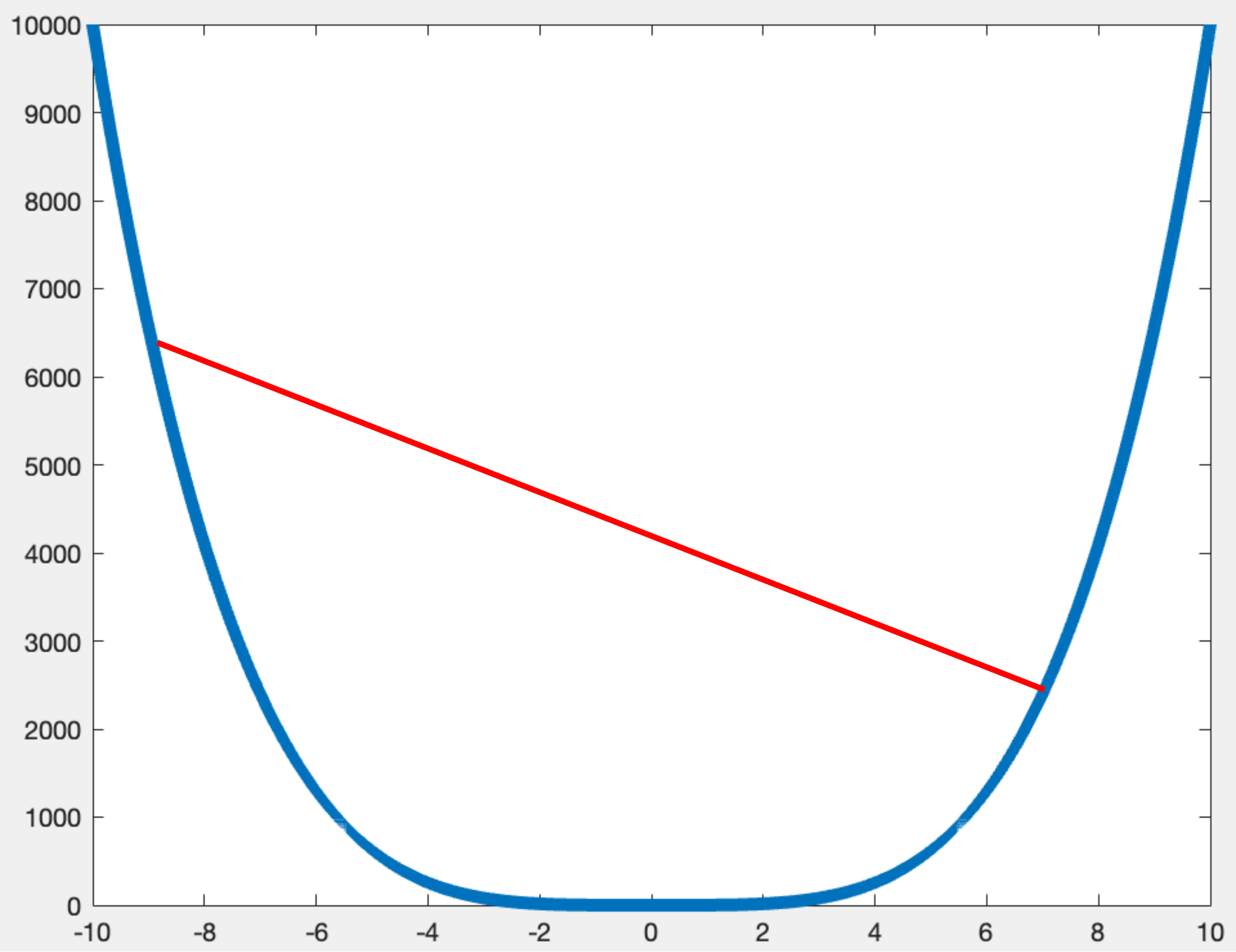
What makes feedforward network different from linea

- Nonlinear fur

$f_3\Big(f$

- Q: Why comp

$f(x) = \mathrm{g}($



$f(x) = \mathrm{g}(\mathrm{h}(\mathrm{x}))$ where $\mathrm{g}(\mathrm{x}) = \mathrm{h}(\mathrm{x}) = x^2$

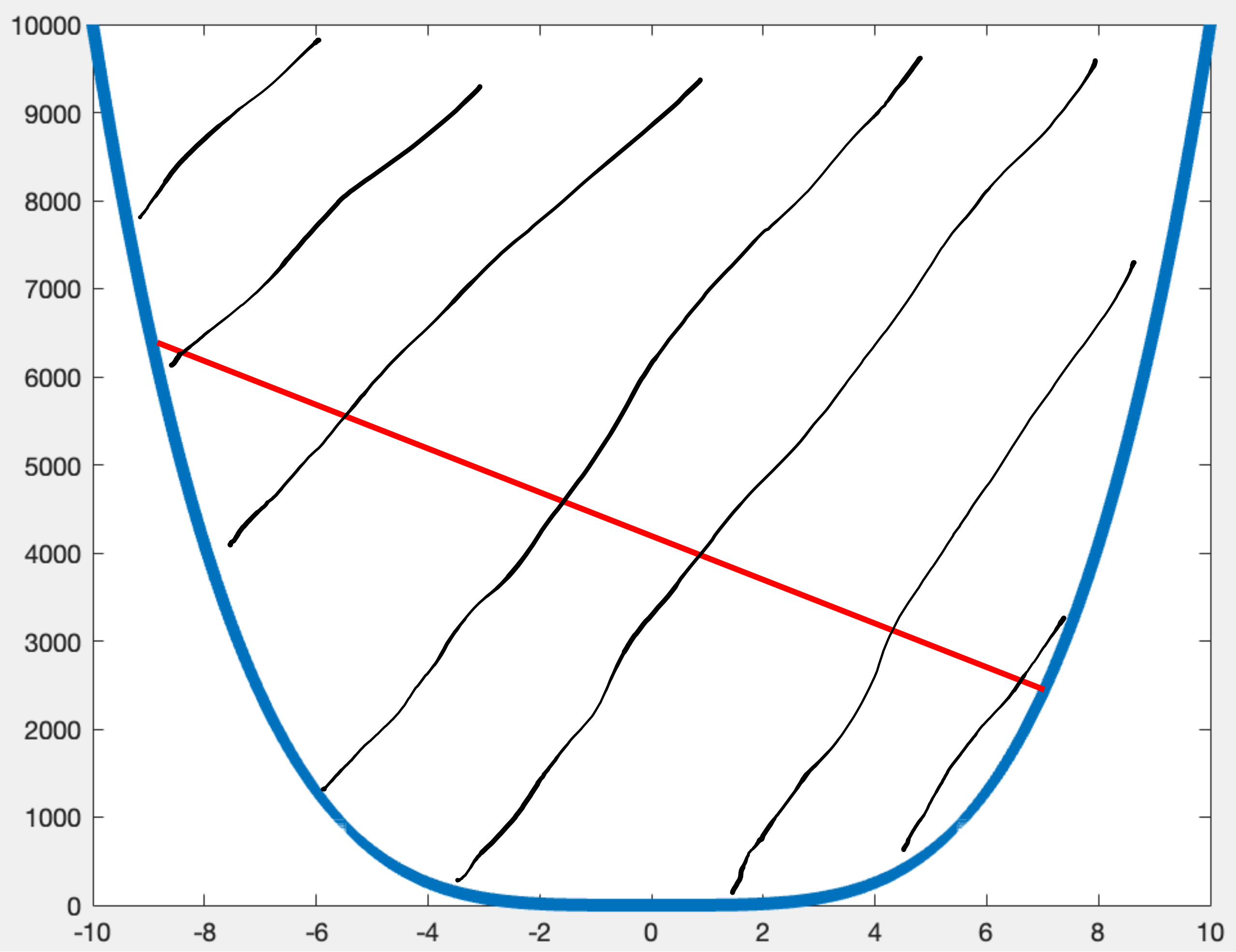
What makes feedforward network different from linea

- Nonlinear fur

$f_3(f$

- Q: Why comp

$f(x) = g($



$f(x) = g(h(x))$ where $g(x) = h(x) = x^2$

What makes feedforward network different from linear model

- Nonlinear functions in hidden layers

$$f_3\left(f_2(f_1(x))\right) = \boldsymbol{w}^\top \max\{0, \boldsymbol{W}^\top \boldsymbol{x} + \boldsymbol{c}\}$$

- Q: Why composition makes nonconvexity?

$f(x) = \mathrm{g}(\mathrm{h}(\mathrm{x}))$ where $\mathrm{g}(\mathrm{x}) = \mathrm{h}(\mathrm{x}) = x^2$

$f(x) = \mathrm{g}(\mathrm{h}(\mathrm{x}))$ where $\mathrm{g}(\mathrm{x}) = \mathrm{h}(\mathrm{x}) = \exp(-\mathrm{x})$

What makes feedforward network different from linea

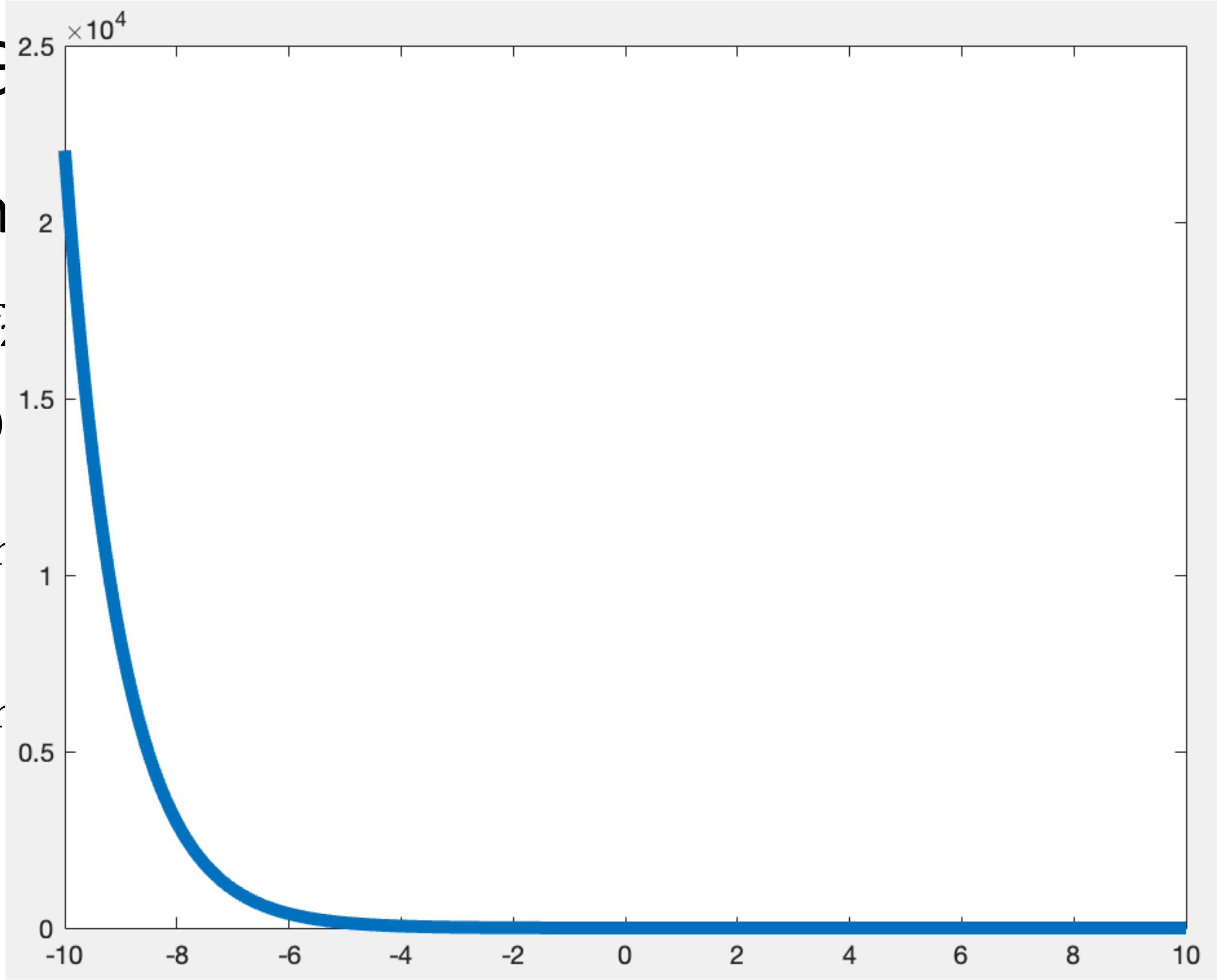
- Nonlinear fun

$f_3(f$

- Q: Why comp

$f(x) = \mathrm{g}($

$f(x) = \mathrm{g}($



$f(\mathrm{x}) = \exp(-\mathrm{x})$

What makes feedforward network different from linea

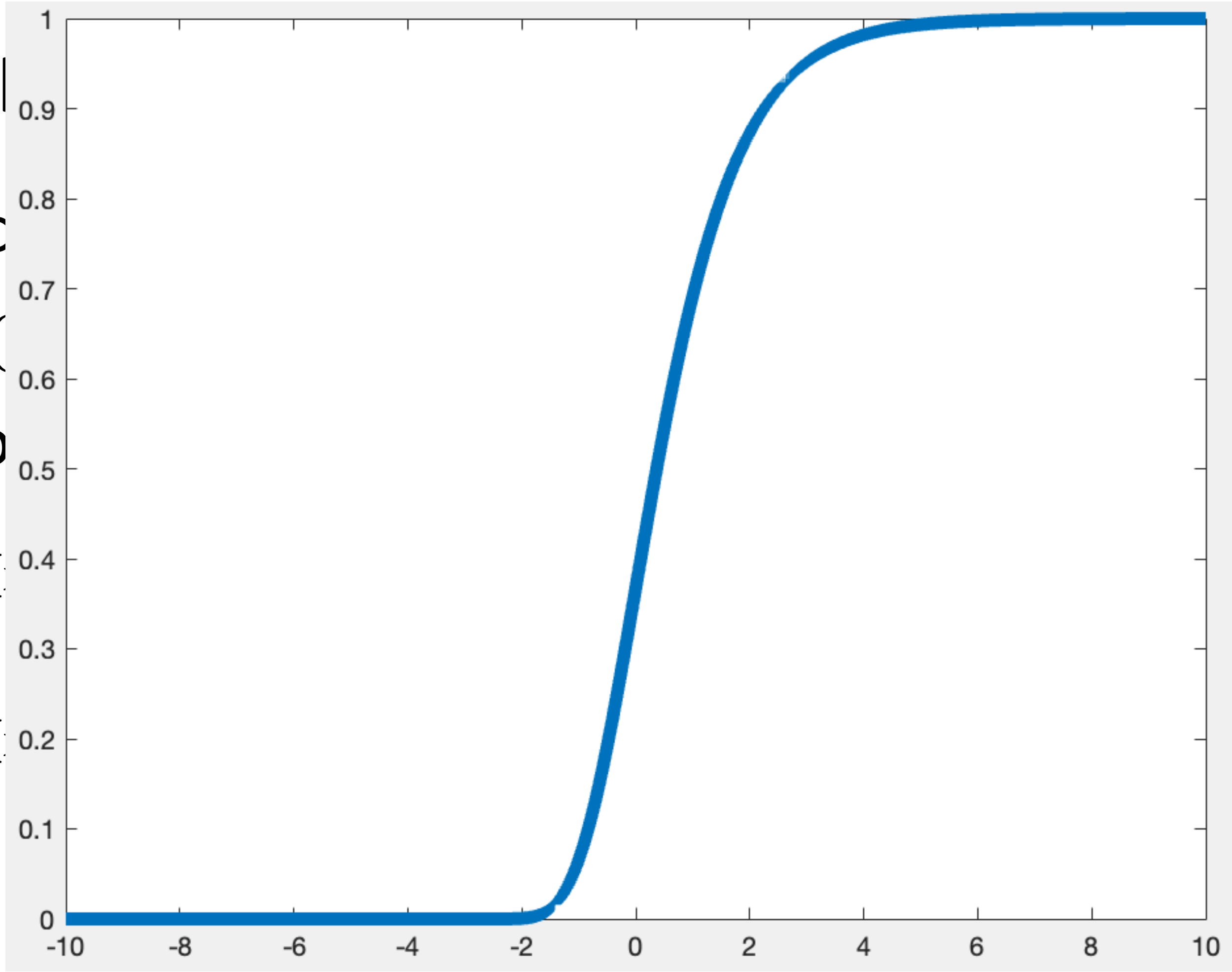
- Nonlinear func

$f_3\left(f_2(\right.$

- Q: Why compo

$f(x) = \mathrm{g}(\mathrm{h}($

$f(x) = \mathrm{g}(\mathrm{h}($



$f(x) = \mathrm{g}(\mathrm{h}(\mathrm{x}))$ where $\mathrm{g}(\mathrm{x}) = \mathrm{h}(\mathrm{x}) = \exp(-\mathrm{x})$

What makes feedforward network different from linear model

- Nonlinear functions in hidden layers

$$f_3\left(f_2(f_1(x))\right) = \boldsymbol{w}^\top \max\{0, \boldsymbol{W}^\top \boldsymbol{x} + \boldsymbol{c}\}$$

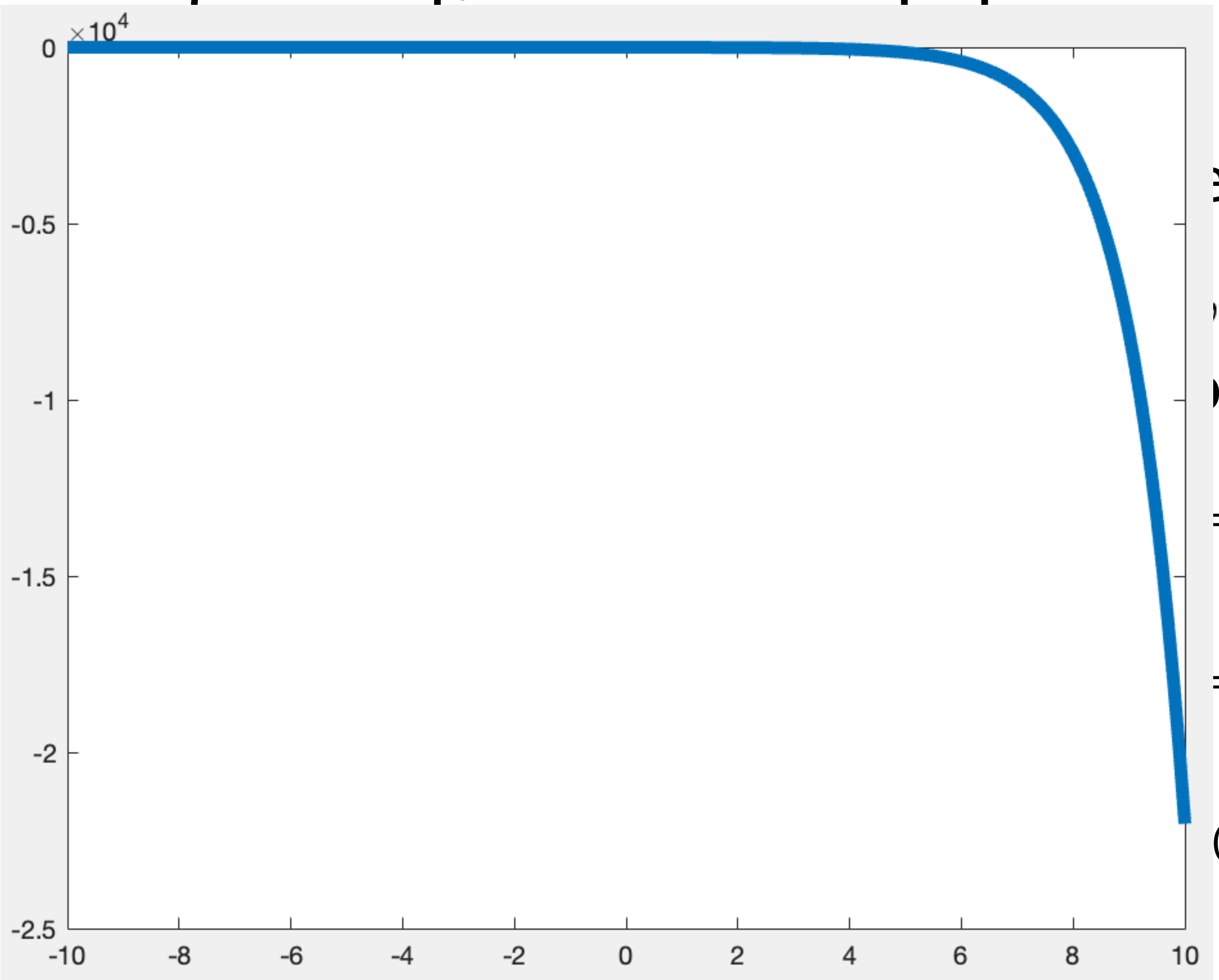
- Q: Why composition makes nonconvexity?

$f(x) = \mathrm{g}(\mathrm{h}(\mathrm{x}))$ where $\mathrm{g}(\mathrm{x}) = \mathrm{h}(\mathrm{x}) = x^2$

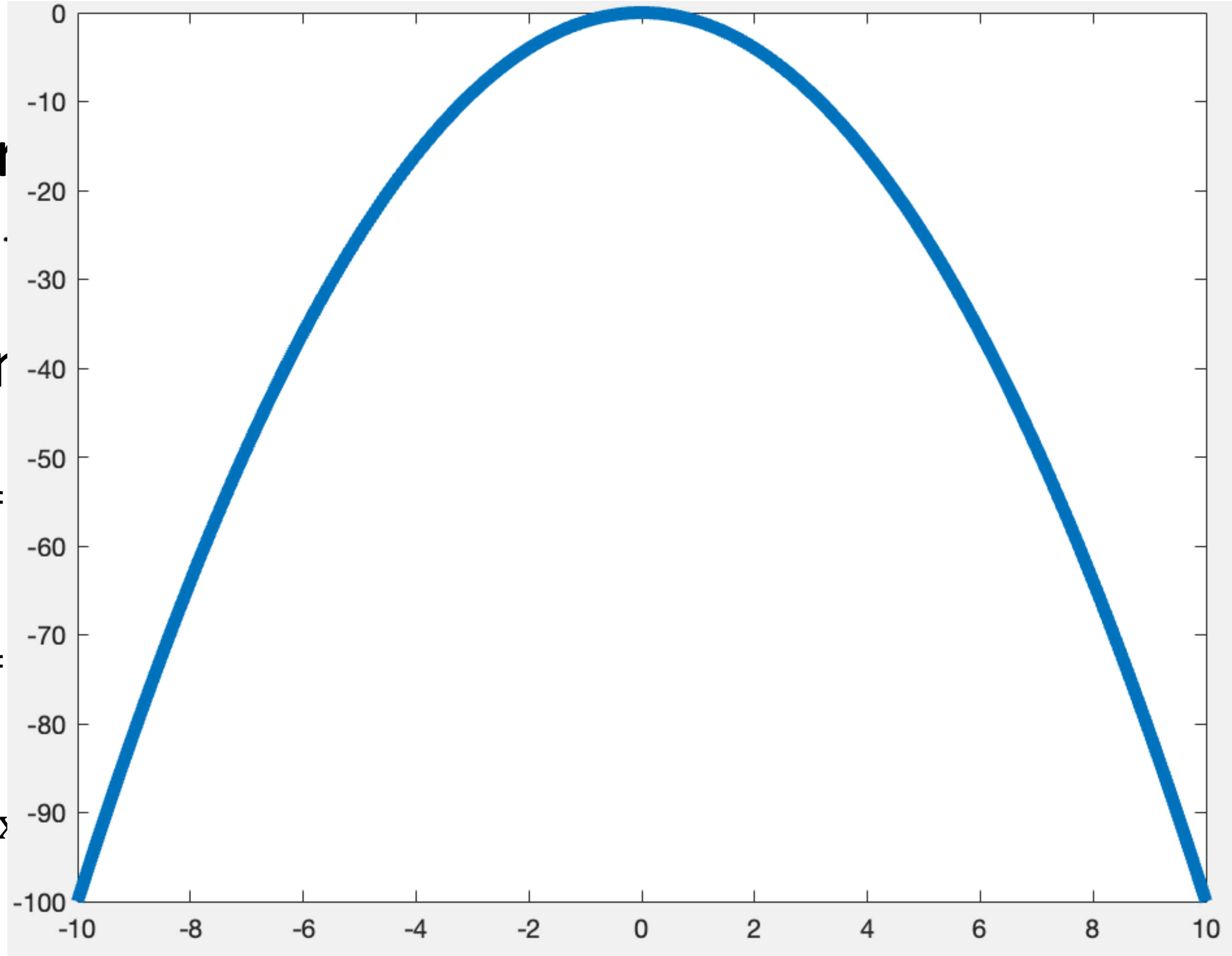
$f(x) = \mathrm{g}(\mathrm{h}(\mathrm{x}))$ where $\mathrm{g}(\mathrm{x}) = \mathrm{h}(\mathrm{x}) = \exp(-\mathrm{x})$

$f(x) = \mathrm{g}(\mathrm{h}(\mathrm{x}))$ where $\mathrm{g}(\mathrm{x}) = -\exp(\mathrm{x}), \mathrm{h}(\mathrm{x}) = -x^2$

What makes feedforward network different



$g(x) = -\exp(x)$



$h(x) = -x^2$

What makes feedforward network different from linear model

- Nonlinear fur

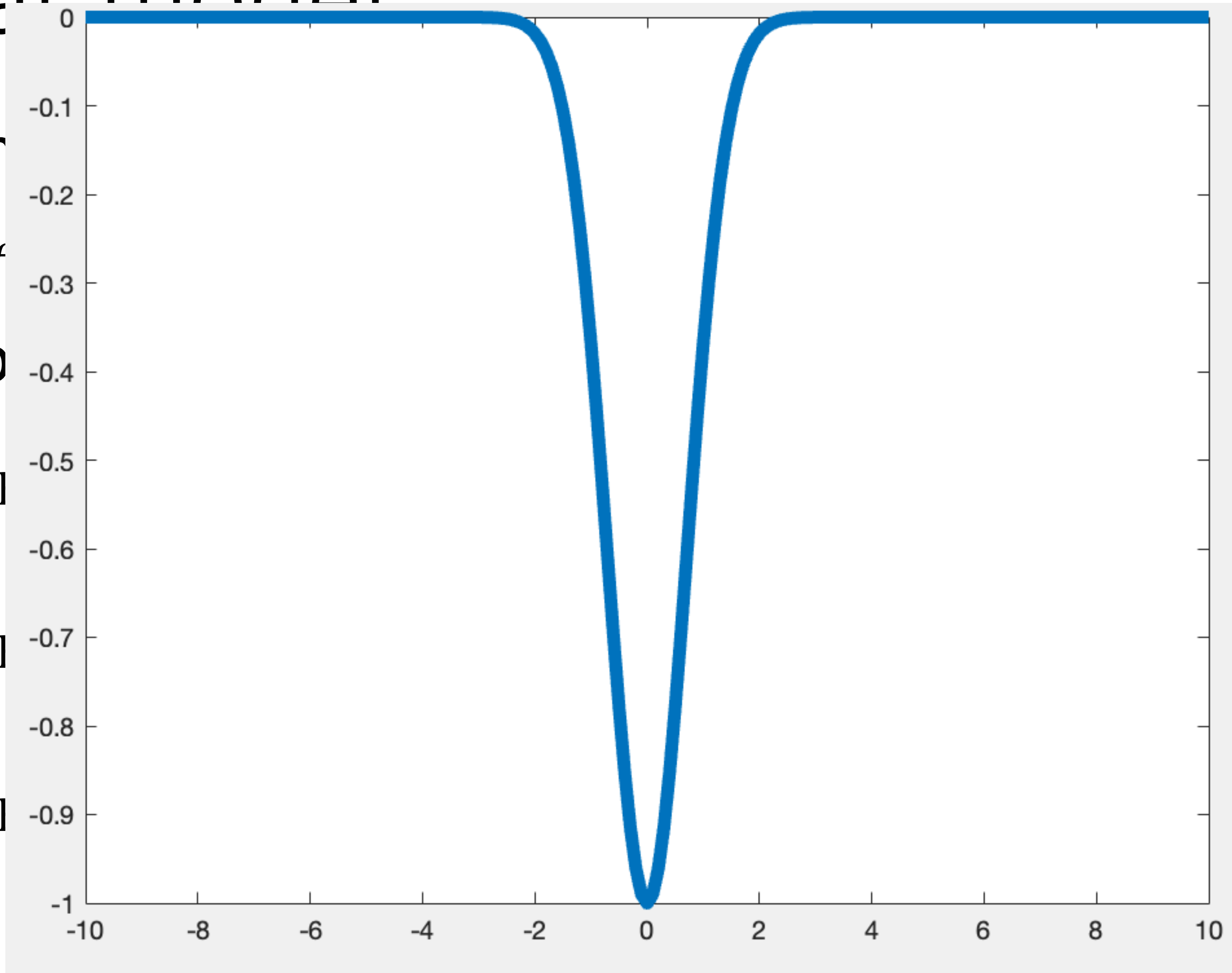
$f_3\left(f\right.$

- Q: Why comp

$f(x) = \mathrm{g}(\mathrm{l}$

$f(x) = \mathrm{g}(\mathrm{l}$

$f(x) = \mathrm{g}(\mathrm{l}$



$f(x) = \mathrm{g}(\mathrm{h}(\mathrm{x}))$ where $\mathrm{g}(\mathrm{x}) = -\exp(\mathrm{x}), \mathrm{h}(\mathrm{x}) = -x^2$

What makes feedforward network different from linear model

- Nonlinear fur

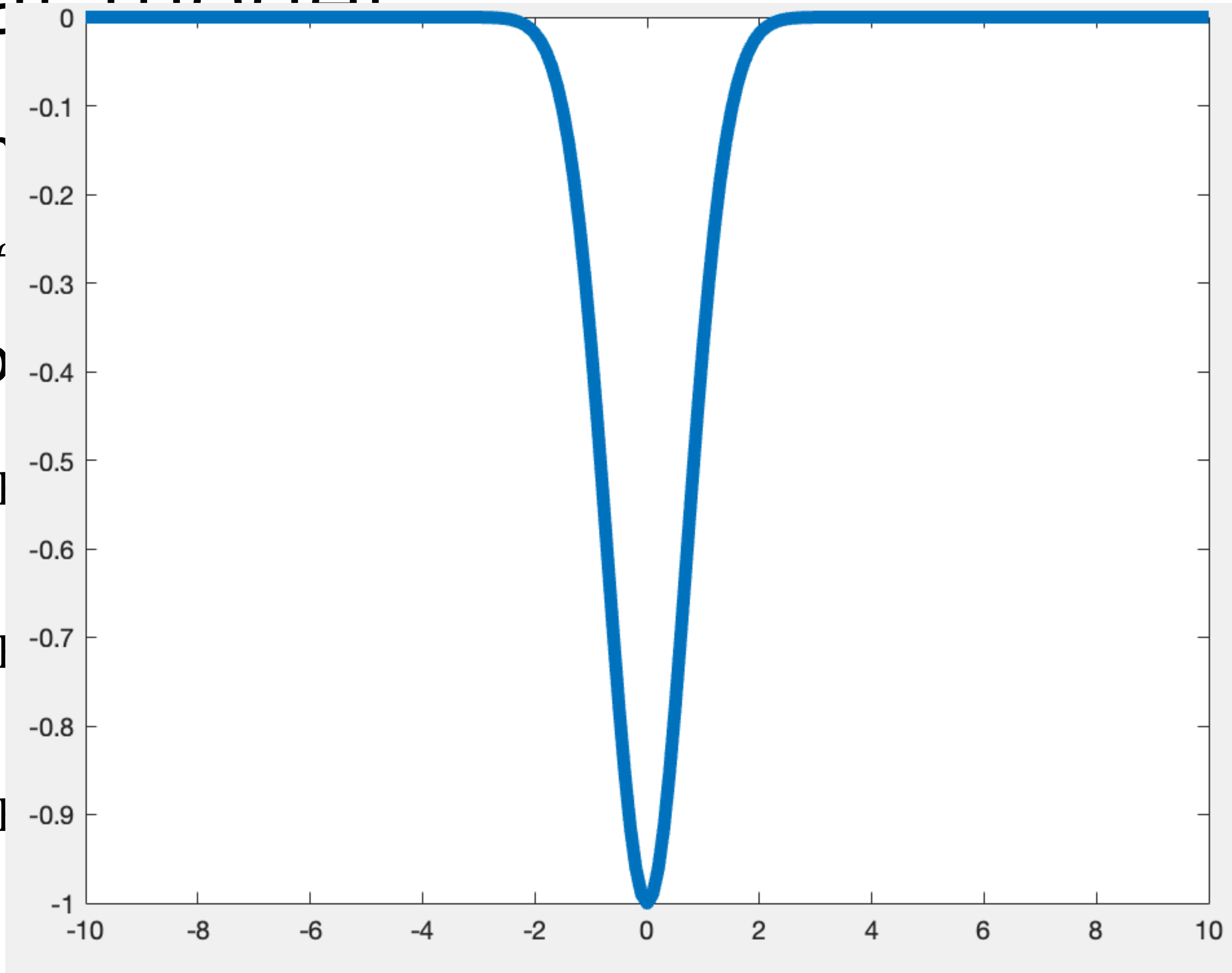
$f_3\big(f$

- Q: Why comp

$f(x) = \mathrm{g}(\mathrm{l}$

$f(x) = \mathrm{g}(\mathrm{l}$

$f(x) = \mathrm{g}(\mathrm{l}$



$f(x) = \mathrm{g}(\mathrm{h}(\mathrm{x}))$ where $\mathrm{g}(\mathrm{x}) = -\exp(\mathrm{x}), \mathrm{h}(\mathrm{x}) = -x^2$

What makes feedforward network different from linear model

- Nonlinear functions in hidden layers

$$f_3\left(f_2(f_1(x))\right) = \boldsymbol{w}^\top \max\{0, \boldsymbol{W}^\top \boldsymbol{x} + \boldsymbol{c}\}$$

- Q: Why composition makes nonconvexity?

$f(x) = \mathrm{g}(\mathrm{h}(\mathrm{x}))$ where $\mathrm{g}(\mathrm{x}) = \mathrm{h}(\mathrm{x}) = x^2$

$f(x) = \mathrm{g}(\mathrm{h}(\mathrm{x}))$ where $\mathrm{g}(\mathrm{x}) = \mathrm{h}(\mathrm{x}) = \exp(-\mathrm{x})$

$f(x) = \mathrm{g}(\mathrm{h}(\mathrm{x}))$ where $\mathrm{g}(\mathrm{x}) = -\exp(\mathrm{x}), \mathrm{h}(\mathrm{x}) = -x^2$

- Special cases

What makes feedforward network different from linear model

- Nonlinear functions in hidden layers

$$f_3\left(f_2(f_1(x))\right) = \boldsymbol{w}^\top \max\{0, \boldsymbol{W}^\top \boldsymbol{x} + \boldsymbol{c}\}$$

- Q: Why composition makes nonconvexity?

$f(x) = \mathrm{g}(\mathrm{h}(\mathrm{x}))$ where $\mathrm{g}(\mathrm{x}) = \mathrm{h}(\mathrm{x}) = x^2$

$f(x) = \mathrm{g}(\mathrm{h}(\mathrm{x}))$ where $\mathrm{g}(\mathrm{x}) = \mathrm{h}(\mathrm{x}) = \exp(-\mathrm{x})$

$f(x) = \mathrm{g}(\mathrm{h}(\mathrm{x}))$ where $\mathrm{g}(\mathrm{x}) = -\exp(\mathrm{x}), \mathrm{h}(\mathrm{x}) = -x^2$

- Special cases

$$f(x) = h(g(x))$$

What makes feedforward network different from linear model

- Nonlinear functions in hidden layers

$$f_3\big(f_2(f_1(x))\big) = \boldsymbol{w}^\top \max\{0, \boldsymbol{W}^\top \boldsymbol{x} + \boldsymbol{c}\}$$

- Q: Why composition makes nonconvexity?

f is convex if h is convex and nondecreasing, and g is convex,
f is convex if h is convex and nonincreasing, and g is concave,
f is concave if h is concave and nondecreasing, and g is concave,
f is concave if h is concave and nonincreasing, and g is convex.

- Special cases

$$f(x) = h(g(x))$$

Boyd, Stephen, Stephen P. Boyd, and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

3.2 Operations that preserve convexity

Today's class

- What makes feedforward network different from linear model
  - Understanding its structure
- Units for neural networks
  - Output units → cost function
    - $f_m\left(\ldots\left(f_2\left(f_1(w; x_i)\right)\right)\right) \rightarrow y_i$
  - Hidden units
    - $f_m\left(\ldots\left(f_2\left(f_1(w; x_i)\right)\right)\right) \rightarrow y_i$

Cost function and output units

Binary	Gray code	One-hot
000	000	00000001
001	001	00000010
010	011	00000100
011	010	00001000
100	110	00010000
101	111	00100000
110	101	01000000
111	100	10000000

$$f_m\Big(\dots\Big(f_2\big(f_1(w;x_i)\big)\Big)\Big) \to y_i$$

- How to interact with groundtruth labels?
- Likelihood function

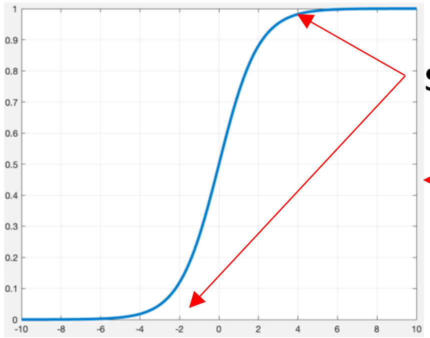
$$L(w) = P_w(X_1 = x_1, \dots, X_n = x_n) = f(w;x_1) \times \cdots \times f(w;x_n) = \prod_{i=1}^{n} \boxed{f(w;x_i)}$$

approximate

Probability mass function
(e.g., Bernoulli distribution)

- Maximum likelihood estimation

$$\max_w \log L(w) = \log \prod_{i=1}^{n} f(w;x_i) = \sum_{i=1}^{n} \log(\boxed{f(w;x_i)})$$



satisfies probability format

$$f_{sigmoid}(z) = \frac{1}{1 + e^{-z}}$$

z: may be unbounded

Cost function and output units

$$f_m\left(\ldots\left(f_2\left(f_1(w; x_i)\right)\right)\right) \rightarrow y_i$$

- How to interact with groundtruth labels?
- Maximum likelihood estimation

$$p_{\text{data}}(\mathbf{x}) \xleftarrow[\text{approximate}]{} p_{\text{model}}(\mathbf{x}; \boldsymbol{\theta})$$

Cost function and output units

$$f_m\left(\dots\left(f_2(f_1(w;x_i))\right)\right) \rightarrow y_i$$

- How to interact with groundtruth labels?
- Maximum likelihood estimation

$$p_{\text{data}}(\mathbf{x}) \xleftarrow{\text{approximate}} p_{\text{model}}(\mathbf{x};\boldsymbol{\theta})$$

Probability for data

Cost function and output units

$$f_m\left(\ldots\left(f_2\left(f_1(w; x_i)\right)\right)\right) \rightarrow y_i$$

- How to interact with groundtruth labels?
- Maximum likelihood estimation

$$p_{\text{data}}(\mathbf{x}) \xleftarrow{\text{approximate}} p_{\text{model}}(\mathbf{x}; \boldsymbol{\theta})$$

Probability for data

Q: Where?

One-hot label:	dog	cat	chair
	1	0	0

Cost function and output units

$$f_m\left(\ldots\left(f_2(f_1(w;x_i))\right)\right) \to y_i$$

- How to interact with groundtruth labels?
- Maximum likelihood estimation

$p_{\text{data}}(\mathbf{x})$ $\xleftarrow{\text{approximate}}$ $p_{\text{model}}(\mathbf{x};\boldsymbol{\theta})$

Probability for data

One-hot label:

dog	cat	chair
1	0	0

Cost function and output units

$$f_m\left(\ldots\left(f_2(f_1(w; x_i))\right)\right) \rightarrow y_i$$

- How to interact with groundtruth labels?
- Maximum likelihood estimation

$p_{\text{data}}(\mathbf{x})$ $\xleftarrow{\text{approximate}}$ $p_{\text{model}}(\mathbf{x}; \boxed{\boldsymbol{\theta}}) \rightarrow \boxed{\boldsymbol{\theta}}$

Probability for data

Assume we know how to construct it

One-hot label:	dog	cat	chair
	1	0	0

Cost function and output units

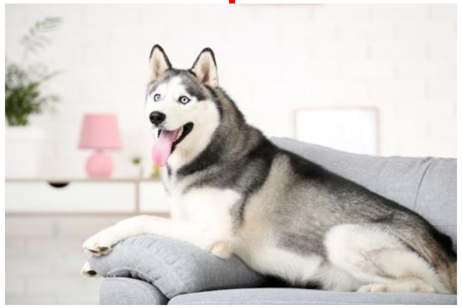
$$f_m\left(\dots\left(f_2\left(f_1(w; x_i)\right)\right)\right) \rightarrow y_i$$

- How to interact with groundtruth labels?
- Maximum likelihood estimation

$$p_{\text{data}}(\mathbf{x}) \xleftarrow{\text{approximate}} p_{\text{model}}(\mathbf{x}; \boxed{\boldsymbol{\theta}}) \rightarrow \boxed{\boldsymbol{\theta}}$$

Probability for data

Assume we know how to construct it



One-hot label:

dog	cat	chair
1	0	0

Cost function and output units

$$f_m\Big(\dots\big(f_2(f_1(w;x_i))\big)\Big) \rightarrow y_i$$

- How to interact with groundtruth labels?
- Maximum likelihood estimation

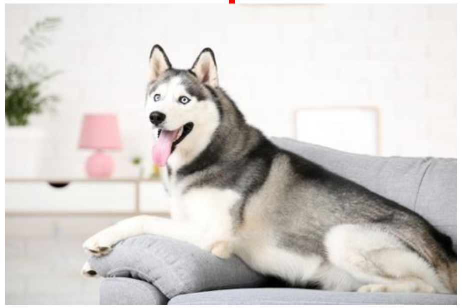
$$p_{\text{data}}(\mathbf{x}) \xleftarrow{\text{approximate}} p_{\text{model}}(\mathbf{x};\boxed{\boldsymbol{\theta}}) \rightarrow \boxed{\boldsymbol{\theta}}$$

Probability for data

Assume we know how to construct it

Probability prediction:

dog	cat	chair
0.98	0.01	0.01



One-hot label:

dog	cat	chair
1	0	0

Cost function and output units

$$f_m\left(\dots\left(f_2\left(f_1(w;x_i)\right)\right)\right) \rightarrow y_i$$

Probability prediction:

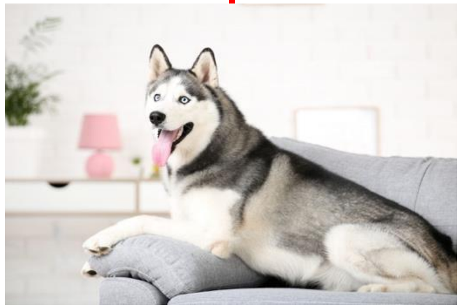
dog	cat	chair
0.98	0.01	0.01

- How to interact with groundtruth labels?
- Maximum likelihood estimation

$$p_{\text{data}}(\mathbf{x}) \xleftarrow{\text{approximate}} p_{\text{model}}(\mathbf{x};\boxed{\boldsymbol{\theta}}) \rightarrow \boxed{\boldsymbol{\theta}}$$

Probability for data

Assume we know how to construct it



One-hot label:

dog	cat	chair
1	0	0

Cost function and output units

$$f_m\Big(\ldots\big(f_2(f_1(w;x_i))\big)\Big) \rightarrow y_i$$

- How to interact with groundtruth labels?
- Maximum likelihood estimation

$p_{\text{data}}(\mathbf{x})$ ← approximate — $p_{\text{model}}(\mathbf{x};\boxed{\boldsymbol{\theta}})$ → $\boxed{\boldsymbol{\theta}}$

Probability for data

Assume we know how to construct it

Probability for “cat”

One-hot label:	dog	cat	chair
	0	1	0

Cost function and output units

$$f_m\Big(\ldots\big(f_2(f_1(w;x_i))\big)\Big) \rightarrow y_i$$

- How to interact with groundtruth labels?
- Maximum likelihood estimation

$$p_{\text{data}}(\mathbf{x}) \xleftarrow{\text{approximate}} p_{\text{model}}(\mathbf{x};\boxed{\boldsymbol{\theta}}) \rightarrow \boxed{\boldsymbol{\theta}}$$

Probability for data

Probability for "cat"

One-hot label:

dog	cat	chair
0	1	0

Assume we know how to construct it

Cost function and output units

$$f_m\left(\ldots\left(f_2(f_1(w;x_i))\right)\right) \rightarrow y_i$$

- How to interact with groundtruth labels?
- Maximum likelihood estimation

Probability prediction:

dog cat chair
0.01 0.98 0.01

$p_{\text{data}}(\mathbf{x}) \xleftarrow{\text{approximate}} p_{\text{model}}(\mathbf{x}; \boxed{\boldsymbol{\theta}}) \rightarrow \boldsymbol{\theta}$

Probability for data

Assume we know how to construct it

Probability for "cat"

One-hot label: dog cat chair
0 1 0

Cost function and output units

$$f_m\left(\ldots\left(f_2\left(f_1(w; x_i)\right)\right)\right) \rightarrow y_i$$

- How to interact with groundtruth labels?
- Maximum likelihood estimation

Probability prediction:

dog	cat	chair
0.01	0.98	0.01

$p_{\text{data}}(\mathbf{x}) \xleftarrow{\text{approximate}} p_{\text{model}}(\mathbf{x}; \boxed{\boldsymbol{\theta}}) \rightarrow \boldsymbol{\theta}$

Probability for data

Assume we know how to construct it

Probability for "cat"

One-hot label:

dog	cat	chair
0	1	0

Cost function and output units

$$f_m\left(\dots\left(f_2\left(f_1(w; x_i)\right)\right)\right) \rightarrow y_i$$

- How to interact with groundtruth labels?
- Maximum likelihood estimation

Probability prediction for "dog/cat/chair" on all data

$$p_{\text{data}}(\mathbf{x}) \xleftarrow{\text{approximate}} p_{\text{model}}(\mathbf{x}; \boxed{\boldsymbol{\theta}}) \rightarrow \boxed{\boldsymbol{\theta}}$$

Probability for data

Assume we know how to construct it

One-hot labels for "dog/cat/chair" on all data

Cost function and output units

$$f_m\left(\ldots\left(f_2\left(f_1(w;x_i)\right)\right)\right) \to y_i$$

- How to interact with groundtruth labels?
- Maximum likelihood estimation

$p_{\text{data}}(\mathbf{x})$ ← approximate — $p_{\text{model}}(\mathbf{x};\boxed{\boldsymbol{\theta}})$ → $\boxed{\boldsymbol{\theta}}$

Probability for data

Assume we know how to construct it

Probability prediction for “dog/cat/chair” on all data

One-hot labels for “dog/cat/chair” on all data

Cost function and output units

- MLE and KL divergence

$$\min_{p_{model}} D_{\mathrm{KL}}\left(\hat{p}_{\mathrm{data}} \| p_{\mathrm{model}}\right) = \mathbb{E}_{\mathbf{x} \sim \hat{p}_{\mathrm{data}}}\left[\log \hat{p}_{\mathrm{data}}(\boldsymbol{x}) - \log p_{\mathrm{model}}(\boldsymbol{x})\right]$$

Cost function and output units

- MLE and KL divergence

Empirical distribution (from training set): cannot scan all possible data $p_{\text{data}}(\mathbf{x})$

$$\min_{p_{model}} D_{\text{KL}}\left(\boxed{\hat{p}_{\text{data}}} \| p_{\text{model}}\right) = \mathbb{E}_{\mathbf{x}\sim\hat{p}_{\text{data}}}\left[\log \hat{p}_{\text{data}}(\boldsymbol{x}) - \log p_{\text{model}}(\boldsymbol{x})\right]$$

Cost function and output units

- MLE and KL divergence

Empirical distribution (from training set): cannot scan all possible data $p_{\text{data}}(\mathbf{x})$

$$\min_{p_{model}} \boxed{D_{\mathrm{KL}}\left(\hat{p}_{\text{data}} \| p_{\text{model}}\right)}=\mathbb{E}_{\mathbf{x} \sim \hat{p}_{\text{data}}}\left[\log \hat{p}_{\text{data}}(\boldsymbol{x})-\log p_{\text{model}}(\boldsymbol{x})\right]$$

If we minimize the KL divergence between the two distributions:

Cost function and output units

- MLE and KL divergence

Empirical distribution (from training set): cannot scan all possible data $p_{\text{data}}(\mathbf{x})$

$$\min_{p_{model}} \boxed{D_{\text{KL}}\left(\hat{p}_{\text{data}} \| p_{\text{model}}\right)} = \mathbb{E}_{\mathbf{x}\sim\hat{p}_{\text{data}}}\left[\log \hat{p}_{\text{data}}(\boldsymbol{x}) - \boxed{\log p_{\text{model}}(\boldsymbol{x})}\right]$$

If we minimize the KL divergence between the two distributions:

Equivalent to

$$\min_{p_{model}} -\mathbb{E}_{\mathbf{x}\sim\hat{p}_{\text{data}}} \boxed{\left[\log p_{\text{model}}(\boldsymbol{x})\right]}$$

Cost function and output units

- MLE and KL divergence

Empirical distribution (from training set): cannot scan all possible data $p_{\text{data}}(\mathbf{x})$

$$\min_{p_{model}} \boxed{D_{\text{KL}}\left(\hat{p}_{\text{data}} \| p_{\text{model}}\right)} = \mathbb{E}_{\mathbf{x}\sim\hat{p}_{\text{data}}}\left[\log \hat{p}_{\text{data}}(\boldsymbol{x}) - \boxed{\log p_{\text{model}}(\boldsymbol{x})}\right]$$

If we minimize the KL divergence between the two distributions:

Equivalent to

$$\min_{p_{model}} -\mathbb{E}_{\mathbf{x}\sim\hat{p}_{\text{data}}}\boxed{\left[\log p_{\text{model}}(\boldsymbol{x})\right]}$$

$$\boldsymbol{\theta}_{\text{ML}} = \arg\max_{\boldsymbol{\theta}} \sum_{i=1}^{m} \log p_{\text{model}}(\boldsymbol{x}^{(i)}; \boldsymbol{\theta})$$

Cost function and output units

- MLE and KL divergence

Empirical distribution (from training set): cannot scan all possible data $p_{\text{data}}(\mathbf{x})$

$$\min_{p_{model}} \boxed{D_{\mathrm{KL}}\left(\hat{p}_{\text{data}} \| p_{\text{model}}\right)} = \mathbb{E}_{\mathbf{x}\sim\hat{p}_{\text{data}}}\left[\log \hat{p}_{\text{data}}(\boldsymbol{x}) - \boxed{\log p_{\text{model}}(\boldsymbol{x})}\right]$$

If we minimize the KL divergence between the two distributions:

Equivalent to

$$\min_{p_{model}} -\mathbb{E}_{\mathbf{x}\sim\hat{p}_{\text{data}}} \boxed{\left[\log p_{\text{model}}(\boldsymbol{x})\right]}$$

$$\boldsymbol{\theta}_{\mathrm{ML}} = \arg\max_{\boldsymbol{\theta}} \sum_{i=1}^{m} \log p_{\text{model}}(\boldsymbol{x}^{(i)}; \boldsymbol{\theta})$$

$$= \boxed{\arg\max_{\boldsymbol{\theta}} \mathbb{E}_{\mathbf{x}\sim\hat{p}_{\text{data}}} \log p_{\text{model}}(\boldsymbol{x}; \boldsymbol{\theta})}$$

Cost function and output units

- MLE and KL divergence

Empirical distribution (from training set): cannot scan all possible data $p_{\text{data}}(\mathbf{x})$

$$\min_{p_{model}} \boxed{D_{\mathrm{KL}}\left(\hat{p}_{\text{data}} \| p_{\text{model}}\right)} = \mathbb{E}_{\mathbf{x} \sim \hat{p}_{\text{data}}}\left[\log \hat{p}_{\text{data}}(\boldsymbol{x}) - \boxed{\log p_{\text{model}}(\boldsymbol{x})}\right]$$

If we minimize the KL divergence between the two distributions:

Equivalent to

$$\min_{p_{model}} -\mathbb{E}_{\mathbf{x} \sim \hat{p}_{\text{data}}}\boxed{\left[\log p_{\text{model}}(\boldsymbol{x})\right]}$$

$$\boldsymbol{\theta}_{\mathrm{ML}} = \underset{\boldsymbol{\theta}}{\arg\max} \sum_{i=1}^{m} \log p_{\text{model}}\left(\boldsymbol{x}^{(i)}; \boldsymbol{\theta}\right)$$

Minimizing KL divergence
=
MLE

$$= \boxed{\underset{\boldsymbol{\theta}}{\arg\max}\, \mathbb{E}_{\mathbf{x} \sim \hat{p}_{\text{data}}} \log p_{\text{model}}(\boldsymbol{x}; \boldsymbol{\theta})}$$

What are hidden units?

$$f_m\left(\dots\left(f_2\left(f_1(w; x_i)\right)\right)\right) \rightarrow y_i$$

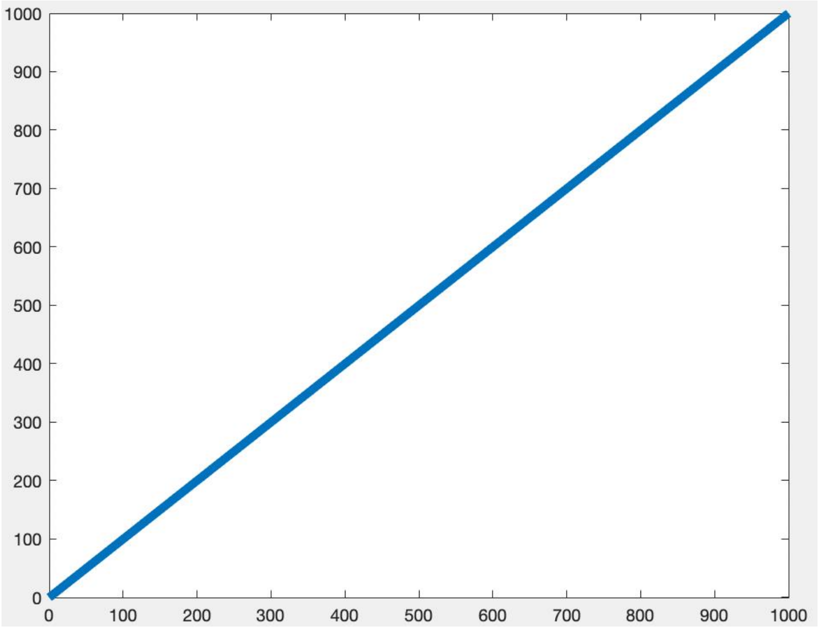
$$\text{Inputs} \longrightarrow \boxed{f} \longrightarrow \text{Outputs}$$

What are hidden units?

$$f_m\left(\ldots\left(f_2\left(f_1(w; x_i)\right)\right)\right) \rightarrow y_i$$

Inputs $\rightarrow \boxed{f} \rightarrow$ Outputs

Q: what if for all layers:
$f(x) = a * x$?

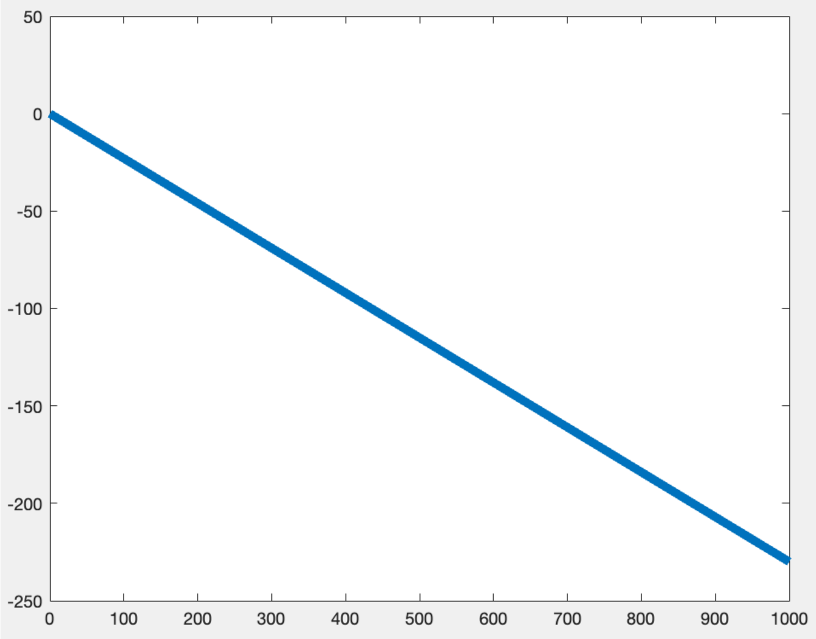


$$f(x) = x$$

What are hidden units?

$$f(x) = a * b * c * d * x$$

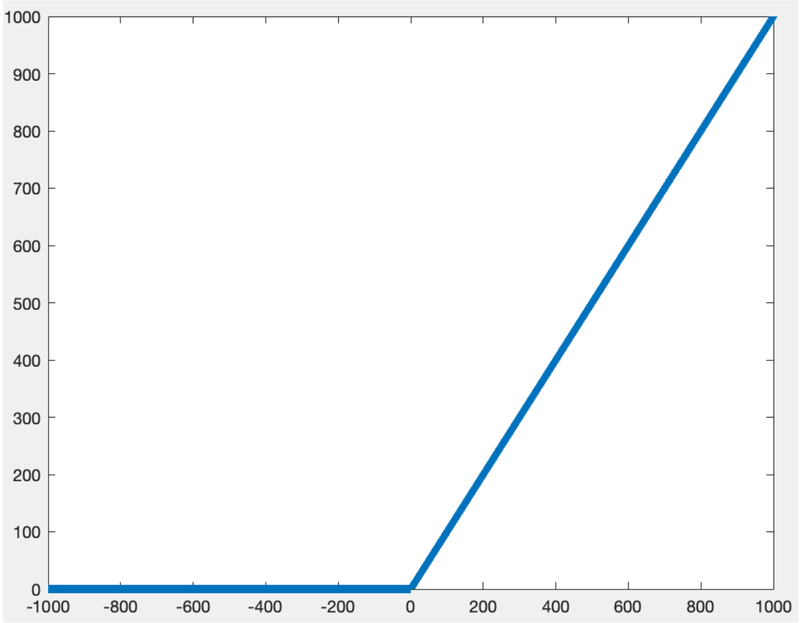
?



Combination of all linear layers is still linear
We are interested in nonlinear layers

ReLU (Rectified Linear Unit)

Activation function

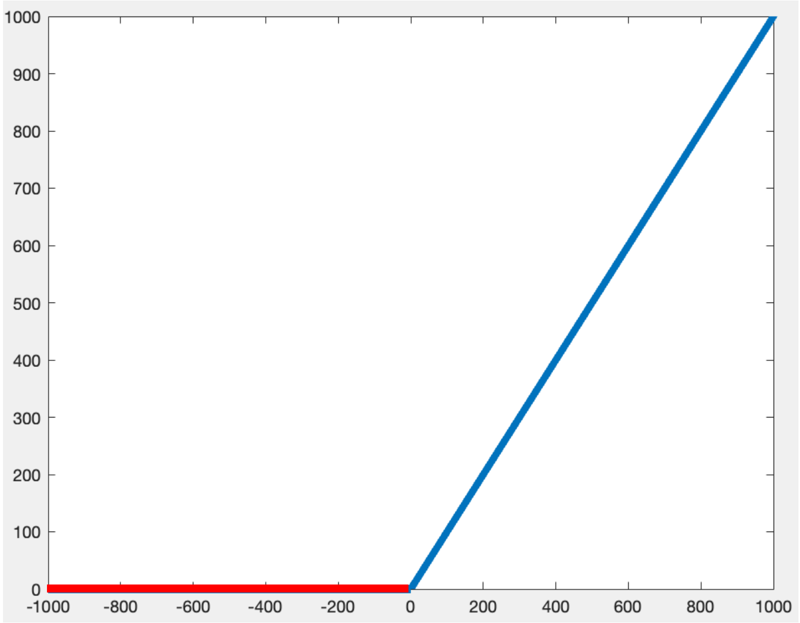


$$f(x) = \max(0, x)$$

ReLU (Rectified Linear Unit)

- Dying ReLU issue

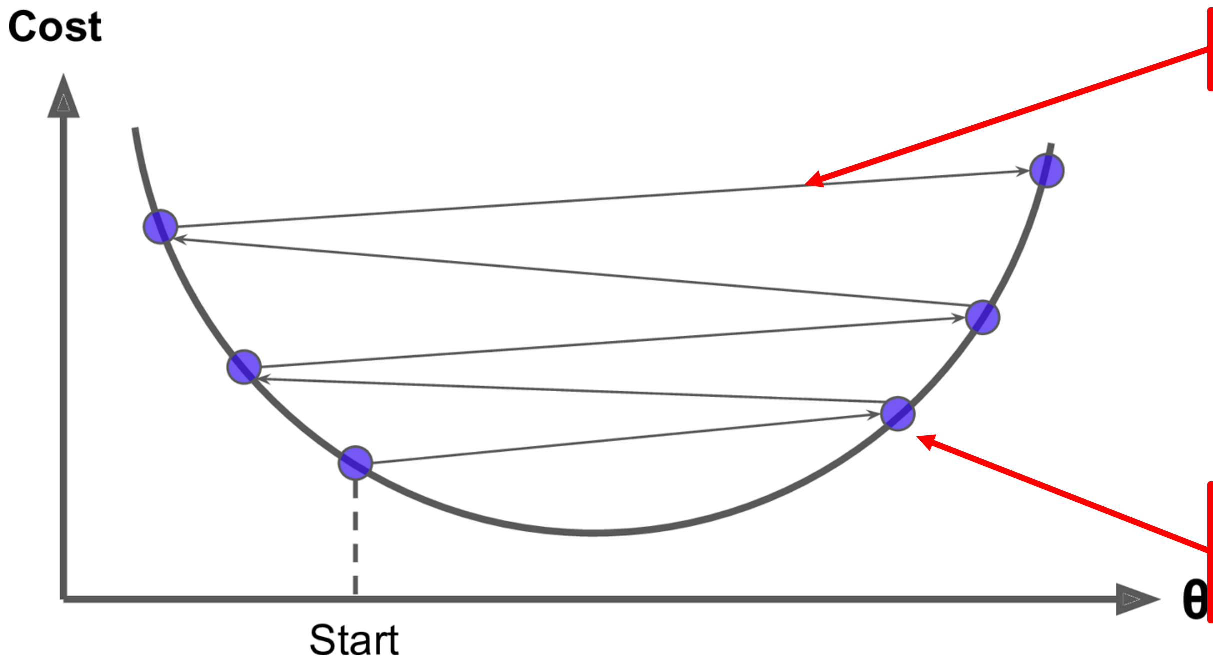
Activation function



$$f(x) = \max(0, x)$$

Determining model parameters

- When to terminate GD (determining T)?
  - Main factors influencing convergence rate?
    - Step size (learning rate)



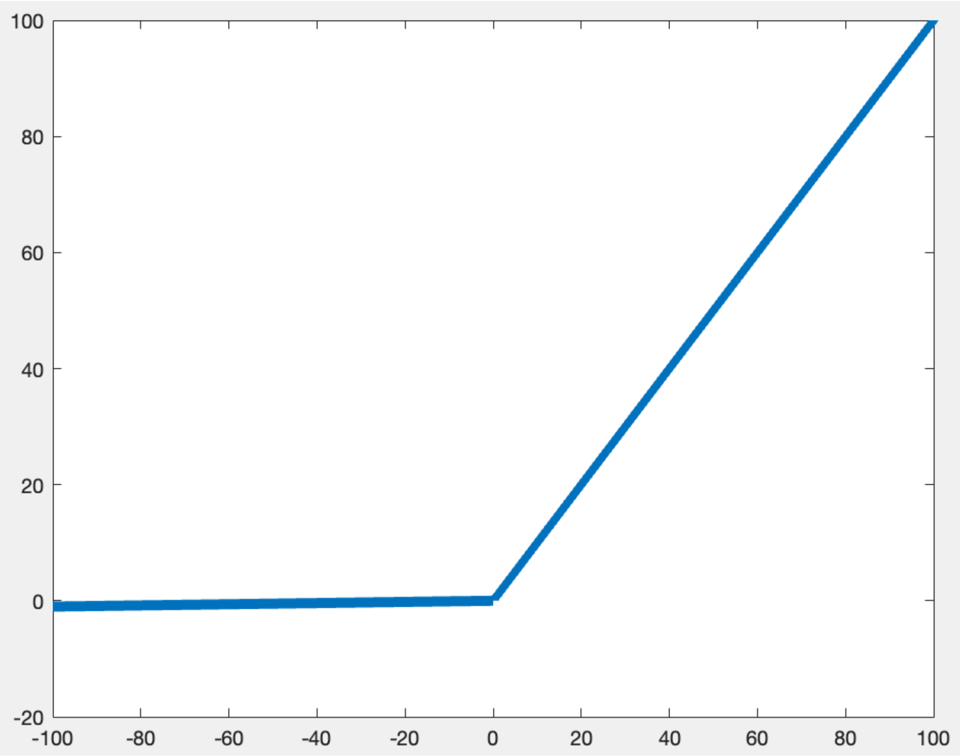
Step size too large

Move the point too far away
May prevent convergence

Image from https://www.oreilly.com/library/view/hands-on-machine-learning/9781491962282/ch04.html

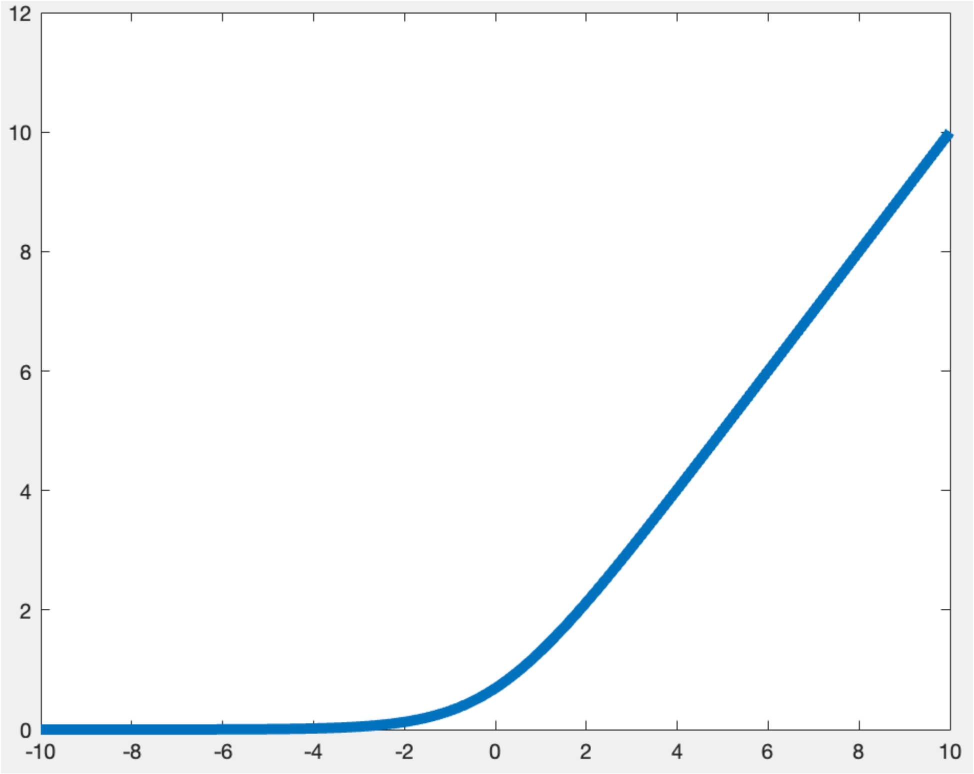
Leaky ReLU

$$f(x_j^i) = \max(0.01x_j^i, x_j^i)$$



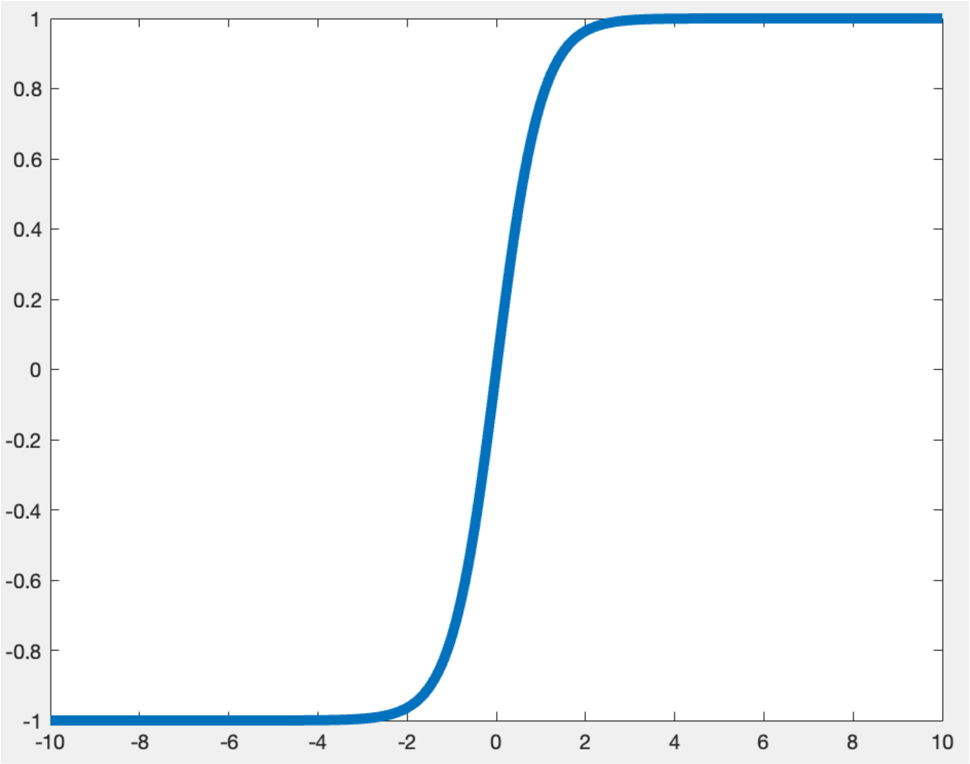
Smooth ReLU/softplus

$$a_j^i = f(x_j^i) = \log\left(1 + \exp(x_j^i)\right)$$



Tanh

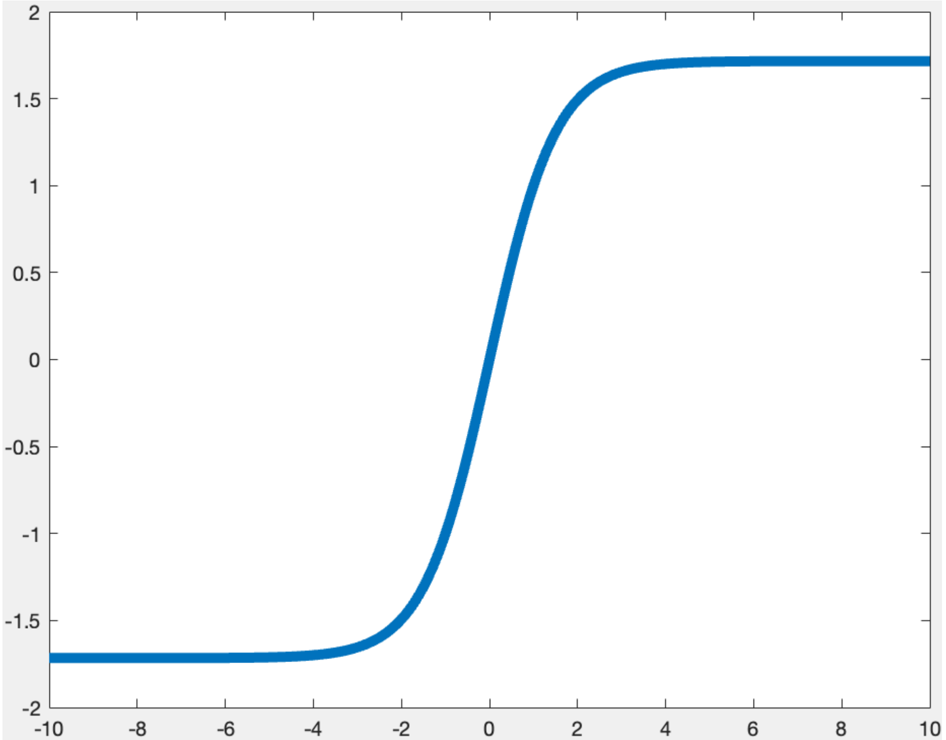
$$f(x_j^i) = \tanh(x_j^i)$$



Credit for https://adl1995.github.io/an-overview-of-activation-functions-used-in-neural-

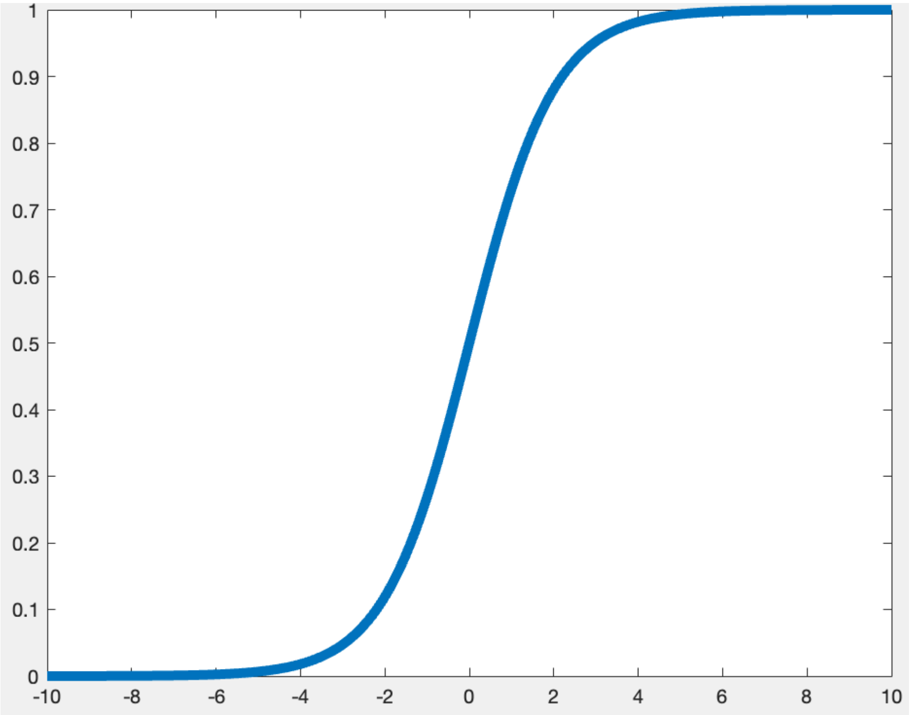
LeCun's Tanh [1]

$$f(x_j^i) = 1.7159 \tanh\left(\frac{2}{3}x_j^i\right)$$



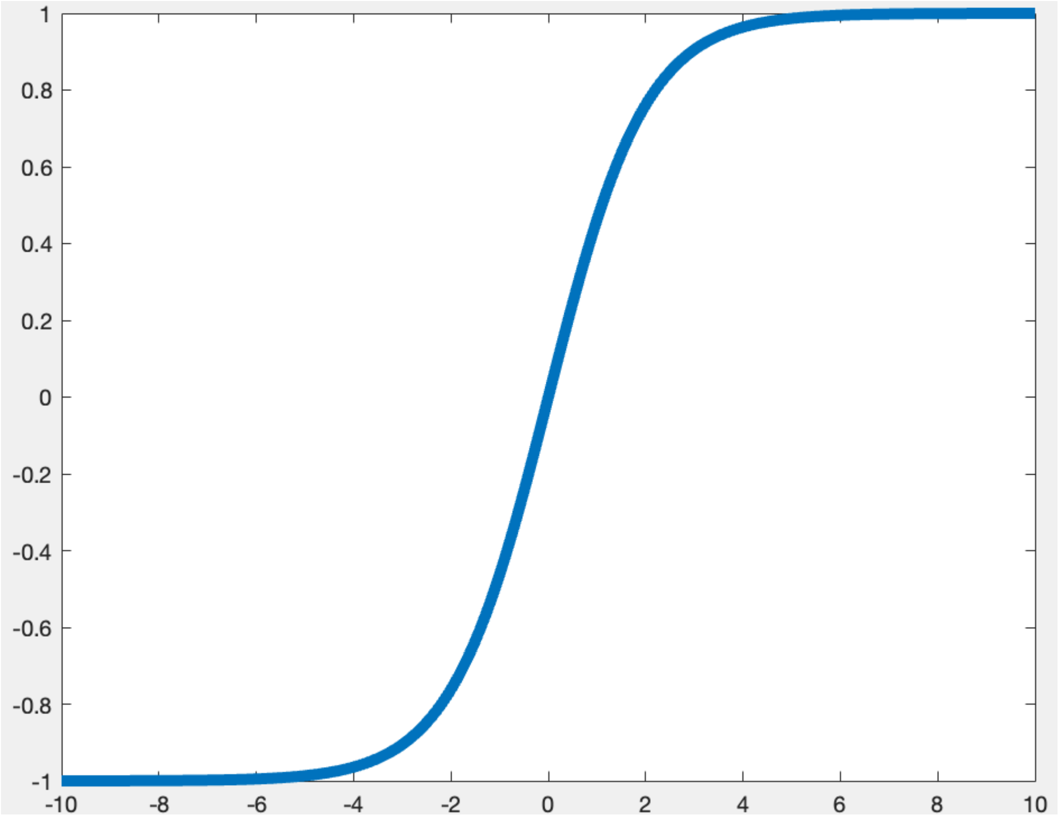
Sigmoid

$$f(x_j^i) = \frac{1}{1 + \exp(-x_j^i)}$$



Bipolar sigmoid

$$f(x_j^i) = \frac{1 - \exp(-x_j^i)}{1 + \exp(-x_j^i)}$$



References

- [1] LeCun, Yann A., Léon Bottou, Genevieve B. Orr, and Klaus-Robert Müller. "Efficient backprop." In *Neural networks: Tricks of the trade*, pp. 9-48. Springer, Berlin, Heidelberg, 2012.

  Pdf online: http://yann.lecun.com/exdb/publis/pdf/lecun-98b.pdf