Composition of functions

CPT_S 434/534 Neural network design and application

In last class

- Bag-of-words features (hand-crafted)
- History of convolutional neural networks
- Feedforward networks

Bag-of-words features

- TF-IDF (term frequency–inverse document frequency)

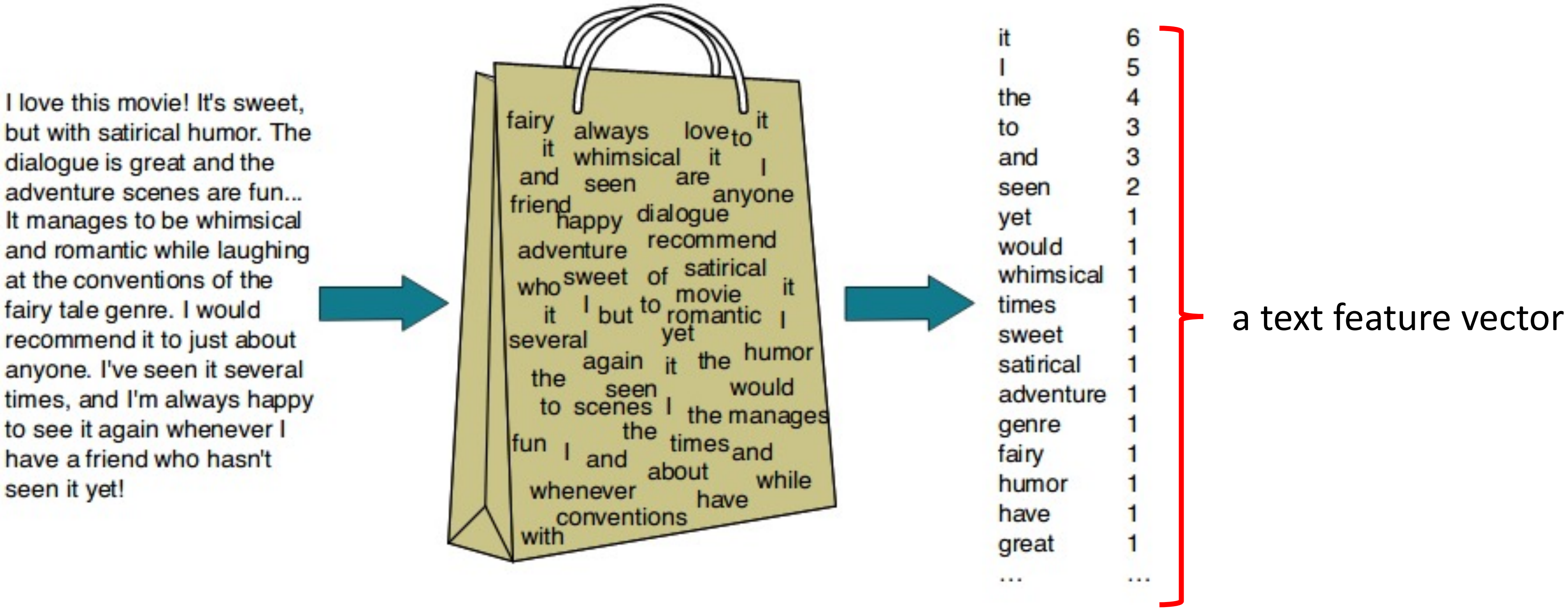


Image from https://heartbeat.fritz.ai/spam-filtering-using-bag-of-words-1c5484ff07f1

Feedforward networks

- Or multilayer perceptrons (MLPs)

Features must be extracted/generated (hand-crafted)

$$f(w; x_i) = x_i{}'w + b \rightarrow y_i$$

Linear model

Weight

Bias

$$f(w; x_i)$$

If we composite another function

$$f_2(f_1(w; x_i))$$

If we composite other functions

1st layer

$$f_m\underbrace{\left(\dots\left(f_2(f_1(w; x_i))\right)\right)}_{\text{Hidden layers}} \rightarrow y_i$$

Feature extraction/generation is not necessary

Features may be learned by hidden layers

final layer or output layer

Feedforward networks

- Or multilayer perceptrons (MLPs)

Features must be extracted/generated (hand-crafted)

$$f(w; x_i) = x_i{}'w + b \rightarrow y_i$$

Linear model

Weight

Bias

$$f(w; x_i)$$

If we composite another function

$$f_2(f_1(w; x_i))$$

If we composite other functions

1st layer

$$f_m\underbrace{\left(\dots\left(f_2\left(f_1(w; x_i)\right)\right)\right)}_{\text{Hidden layers}} \rightarrow y_i$$

Hidden layers $\rightarrow$ Not directly interact with y_i

Feature extraction/generation is not necessary
Features may be learned by hidden layers

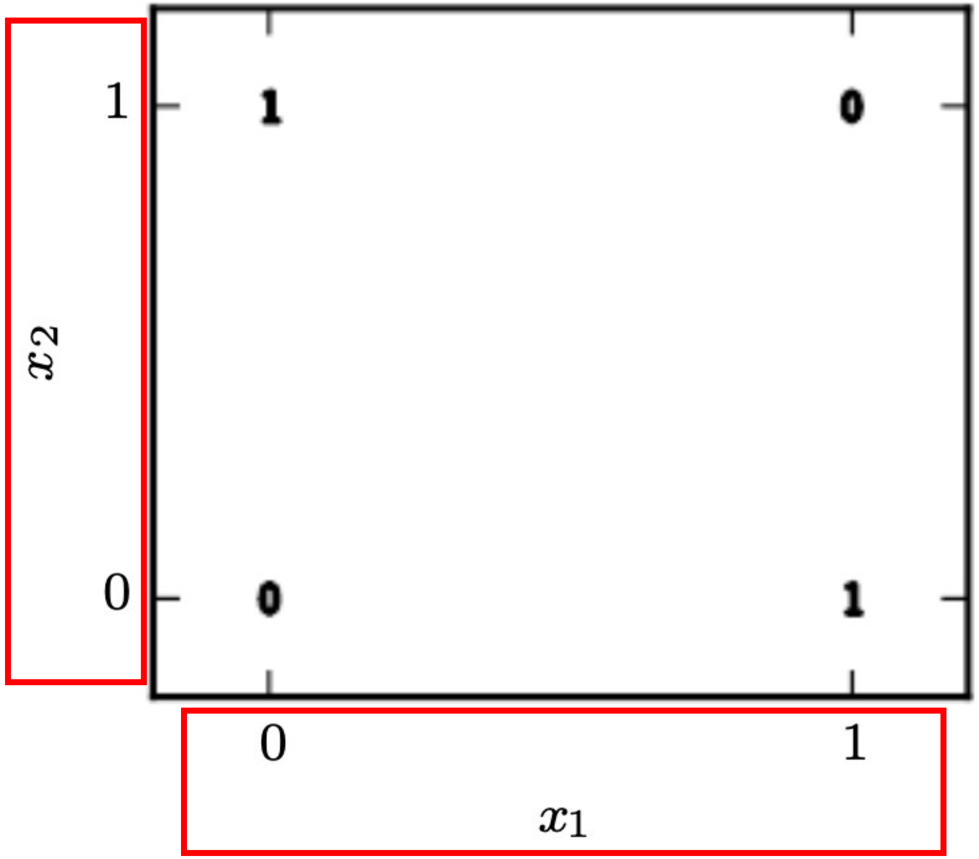
final layer or output layer

Today’s class

- What makes feedforward network different from linear model
  - Understanding its structure

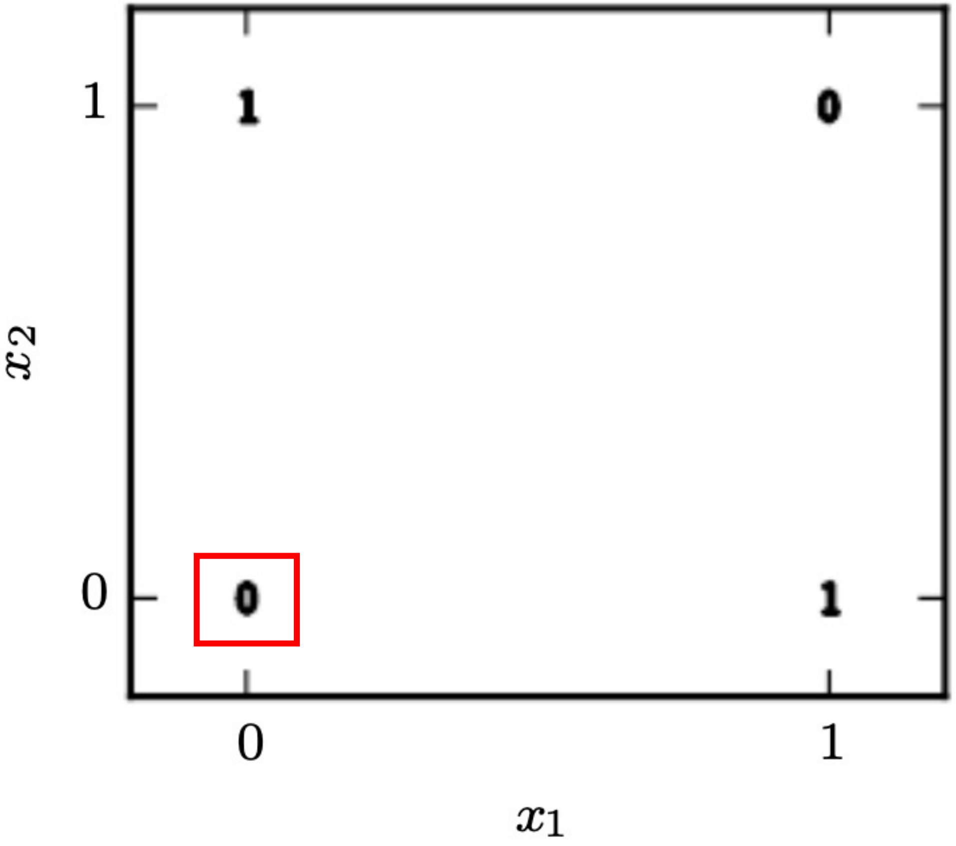
Learning XOR function

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



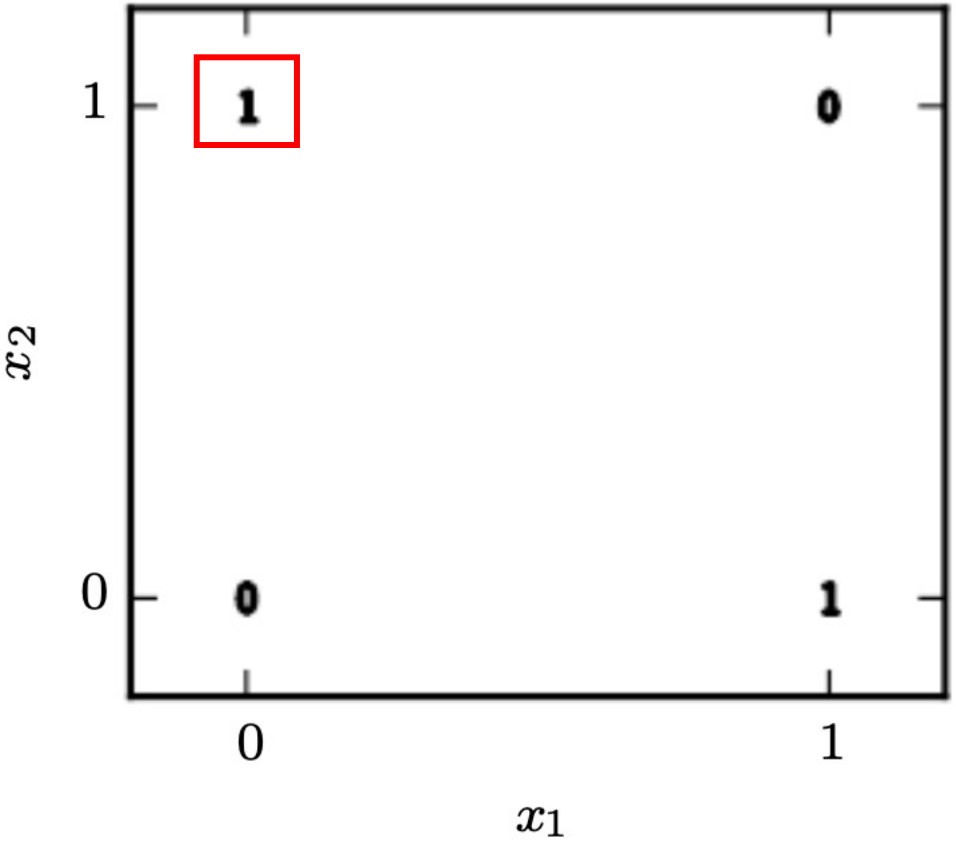
Learning XOR function

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



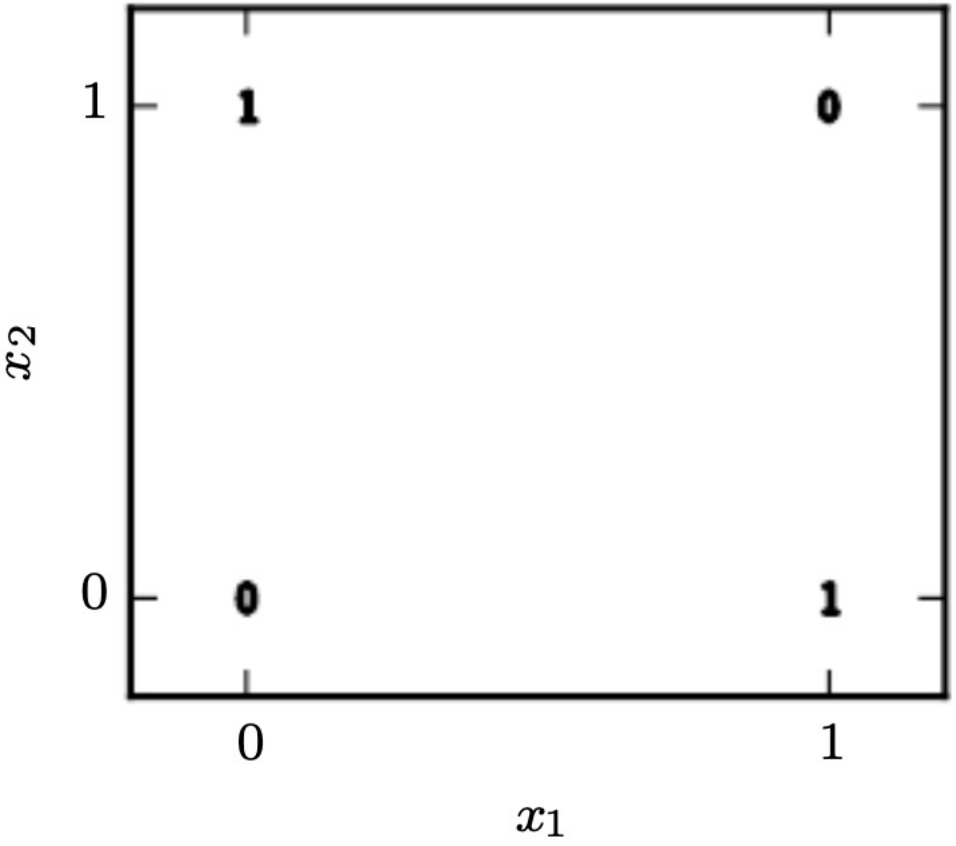
Learning XOR function

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



Learning XOR function

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



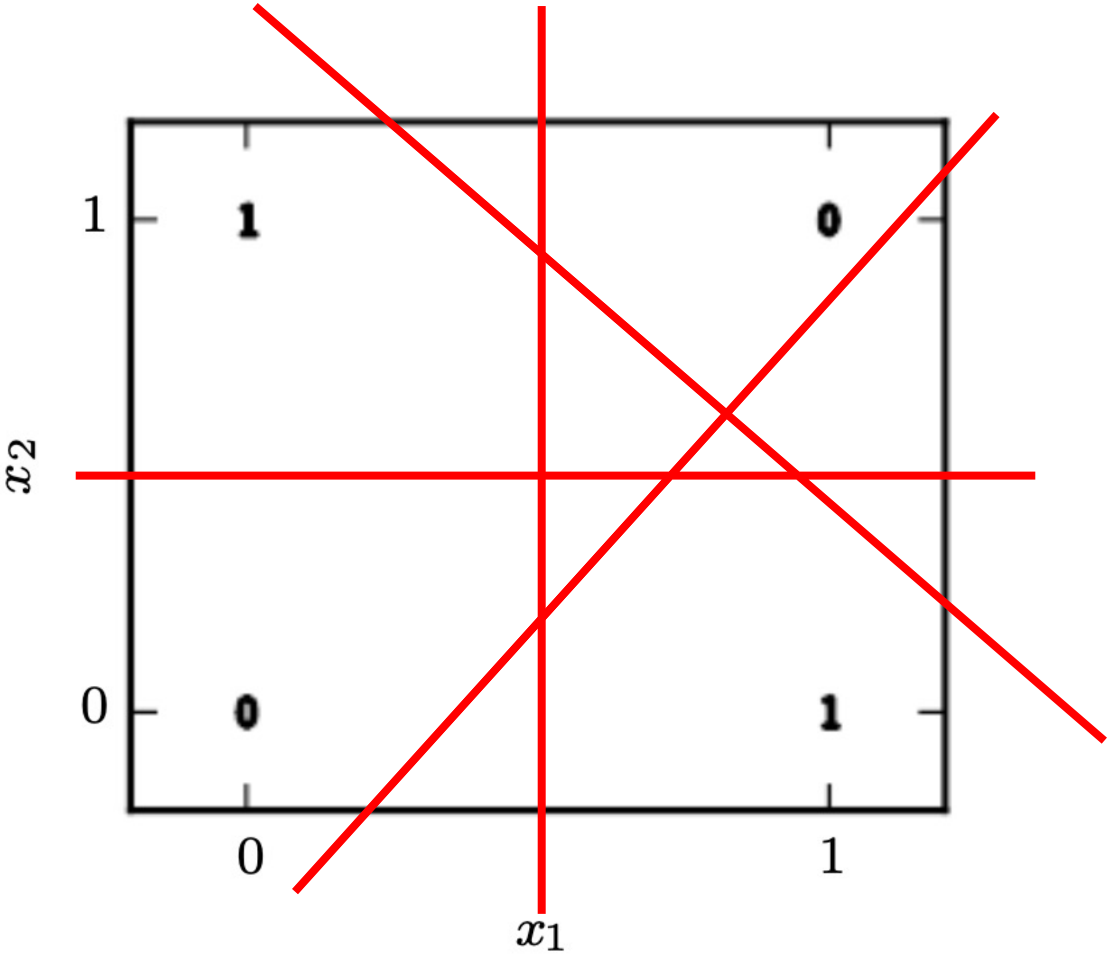
Q: Can we use a linear model to separate 0/1 classes?

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

Learning XOR function

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

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

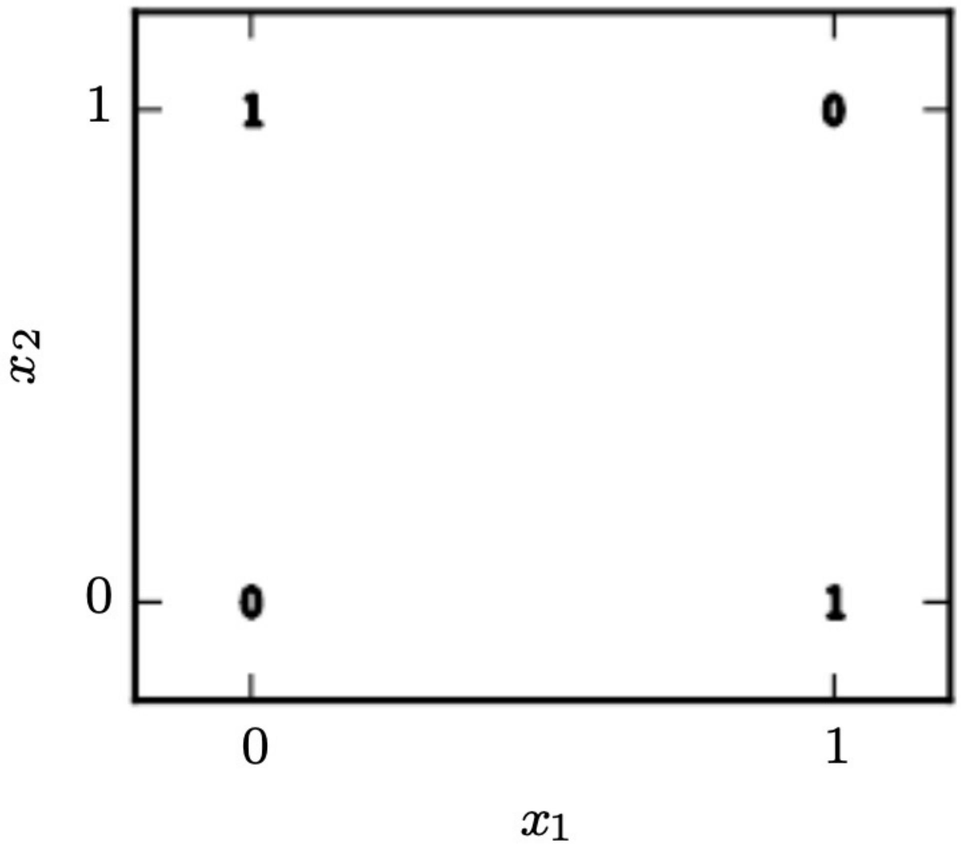


Q: Can we use a linear model to separate 0/1 classes?

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

Learning XOR function

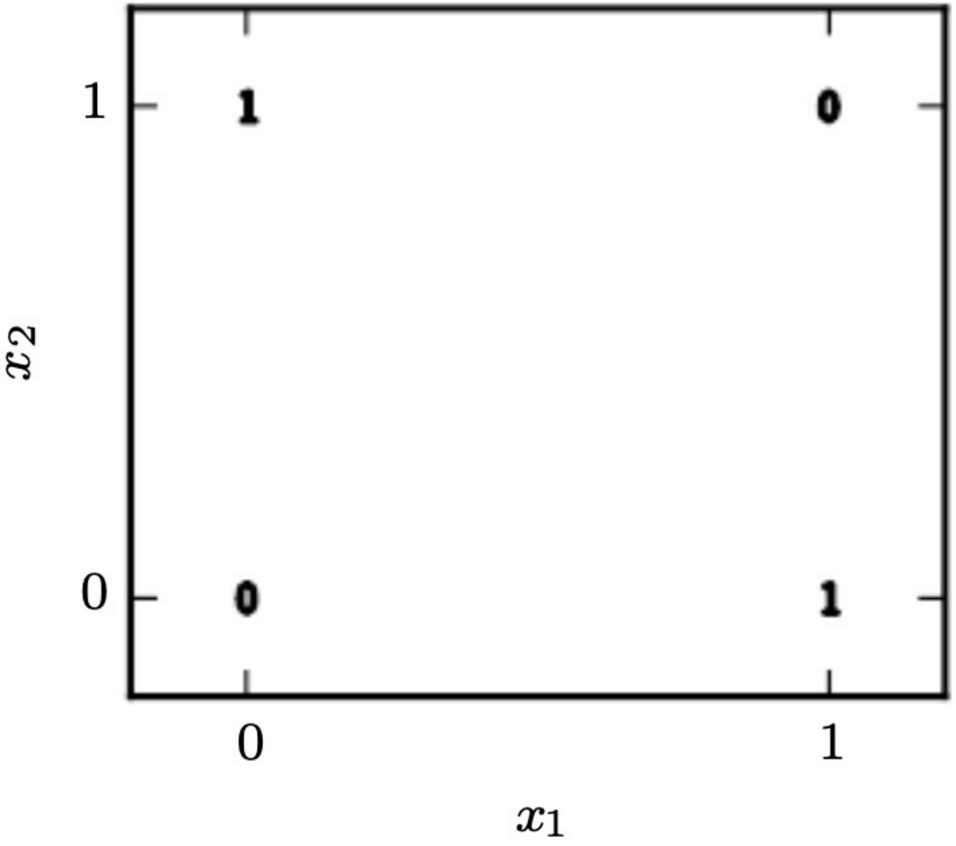
$$
\begin{array}{cc}
 & \begin{array}{cc} x_1 & x_2 \end{array} \\
\boldsymbol{X} = & \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 0 \\ 1 & 1 \end{bmatrix}
\end{array}
\quad
\begin{array}{c} 0 \\ 1 \\ 1 \\ 0 \end{array}
$$



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

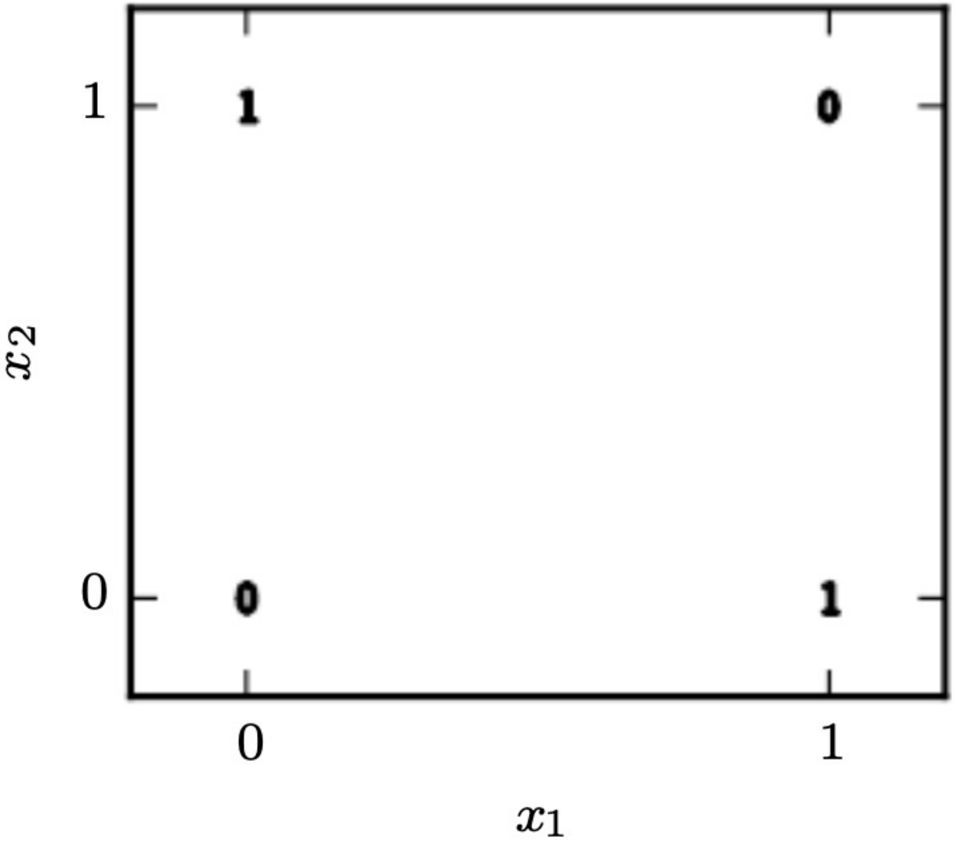
$$\boldsymbol{X} = \begin{array}{c} \begin{matrix} x_1 & x_2 \end{matrix} \\ \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 0 \\ 1 & 1 \end{bmatrix} \end{array} \qquad \begin{matrix} 0 \\ 1 \\ 1 \\ 0 \end{matrix}$$



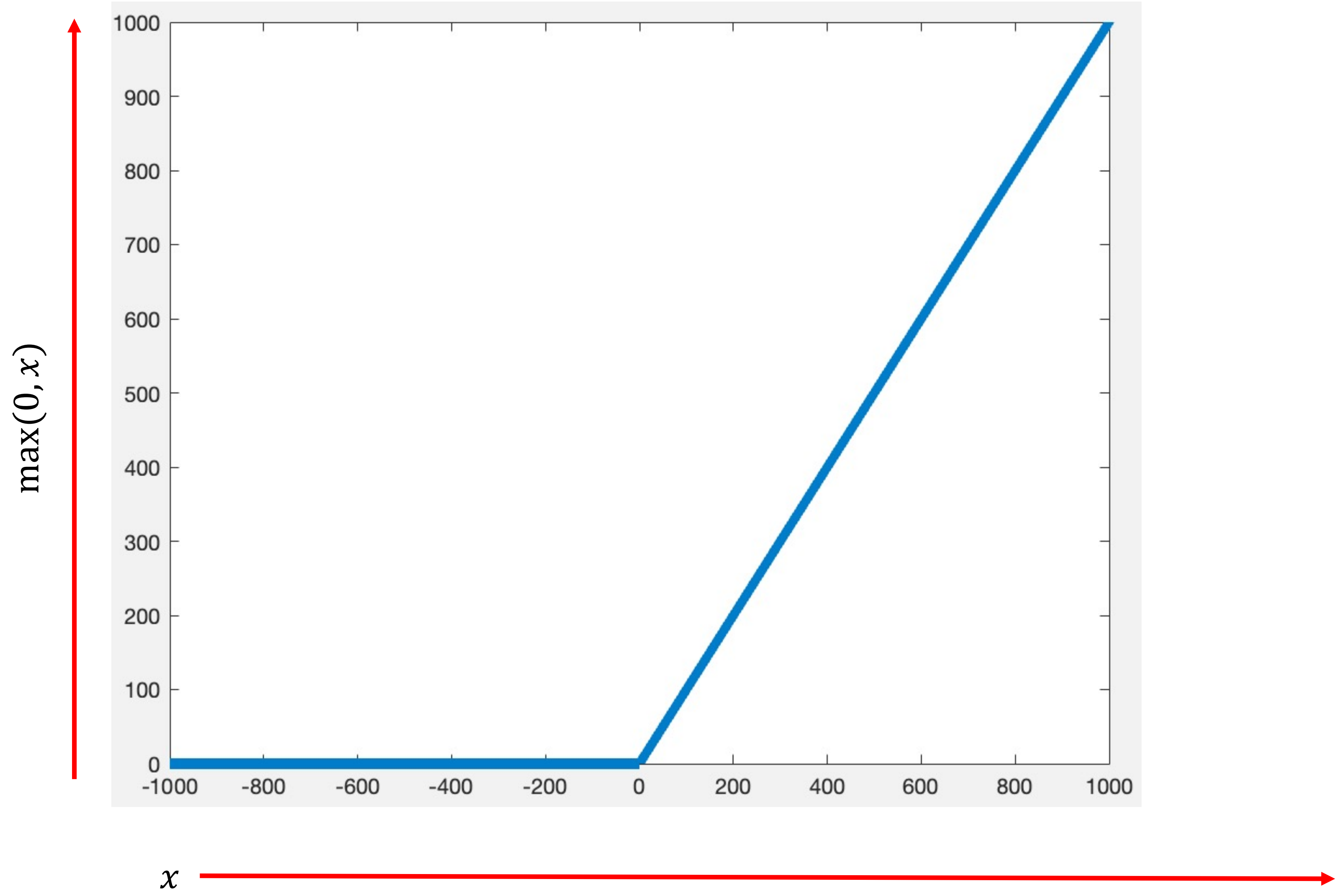
What if we use a nonlinear function as: $f(\boldsymbol{x}; \boldsymbol{W}, \boldsymbol{c}, \boldsymbol{w}, b) = \boldsymbol{w}^\top \max\{0, \boxed{\boldsymbol{W}^\top \boldsymbol{x} + \boldsymbol{c}}\} + b.$

Learning XOR function

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

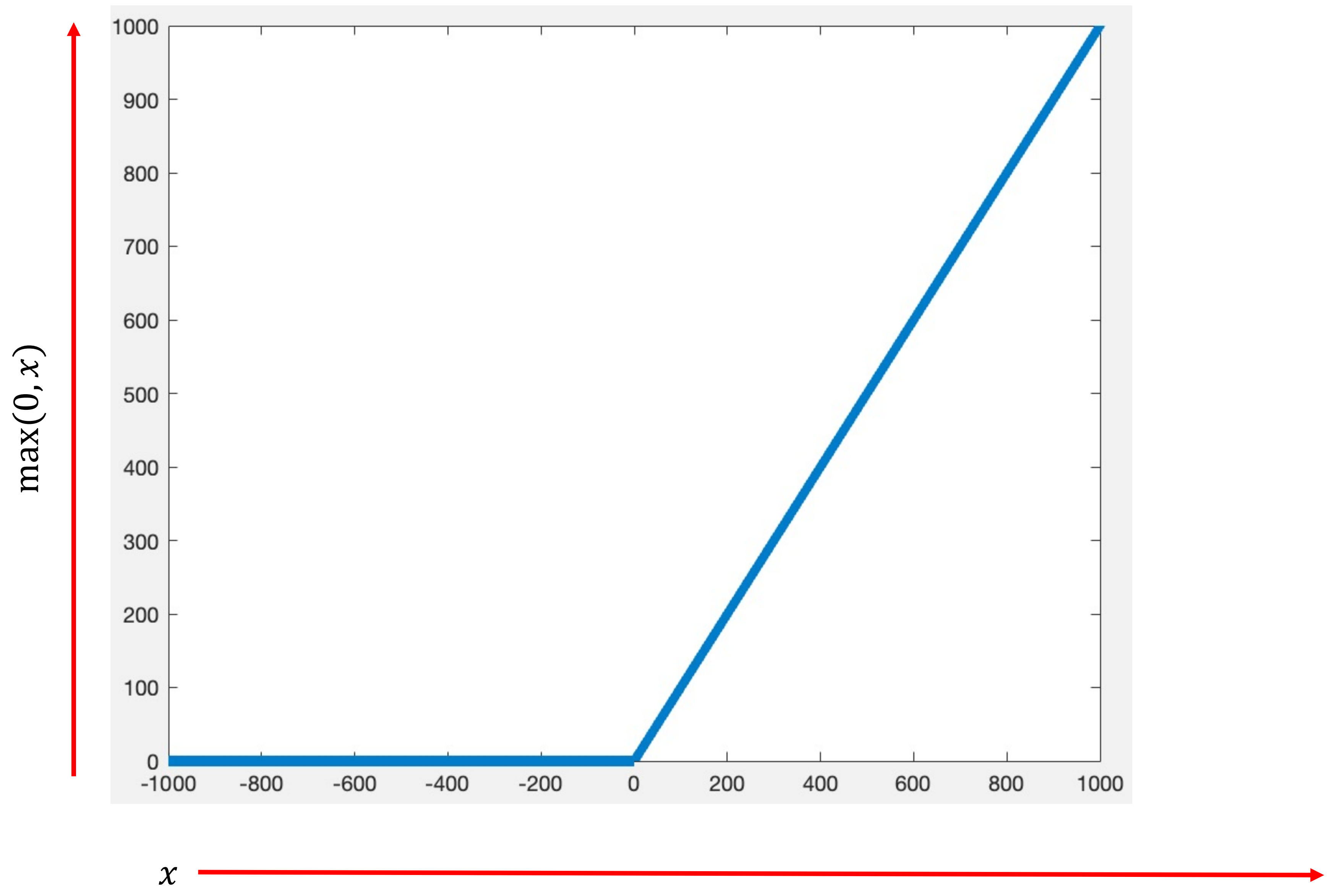


What if we use a nonlinear function as: $f(\boldsymbol{x}; \boldsymbol{W}, \boldsymbol{c}, \boldsymbol{w}, b) = \boldsymbol{w}^\top \boxed{\max\{0}, \boldsymbol{W}^\top \boldsymbol{x} + \boldsymbol{c}\} + b.$

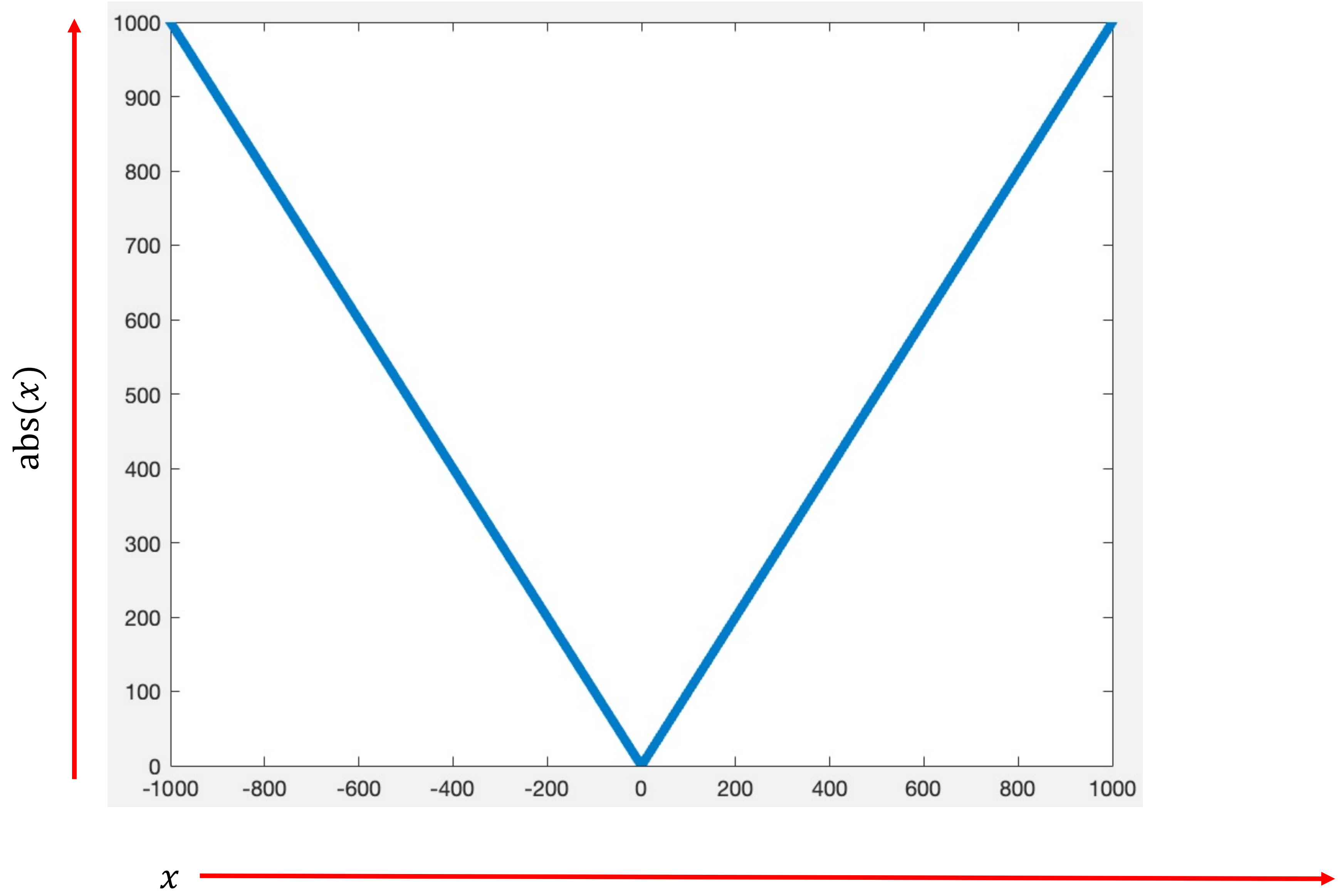


1000
900
800
700
600
500
400
300
200
100
0
-1000
-800
-600
-400
-200
0
200
400
600
800
1000
$\max(0, x)$
x

Piecewise linear

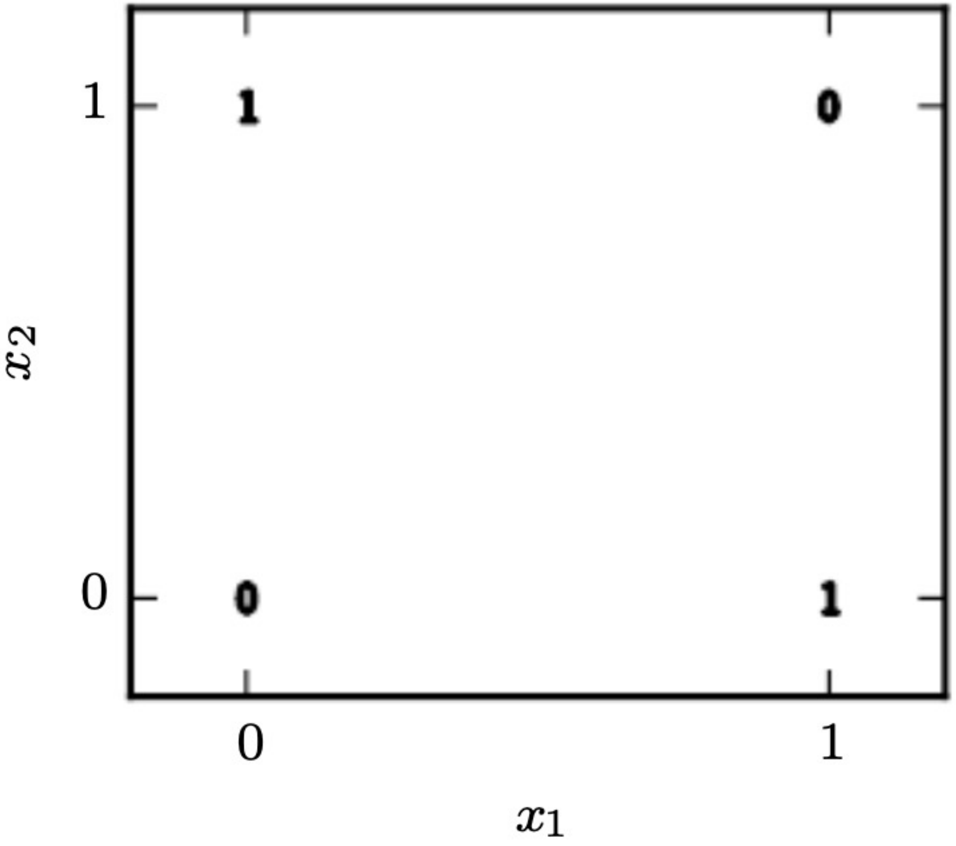


Piecewise linear



Learning XOR function

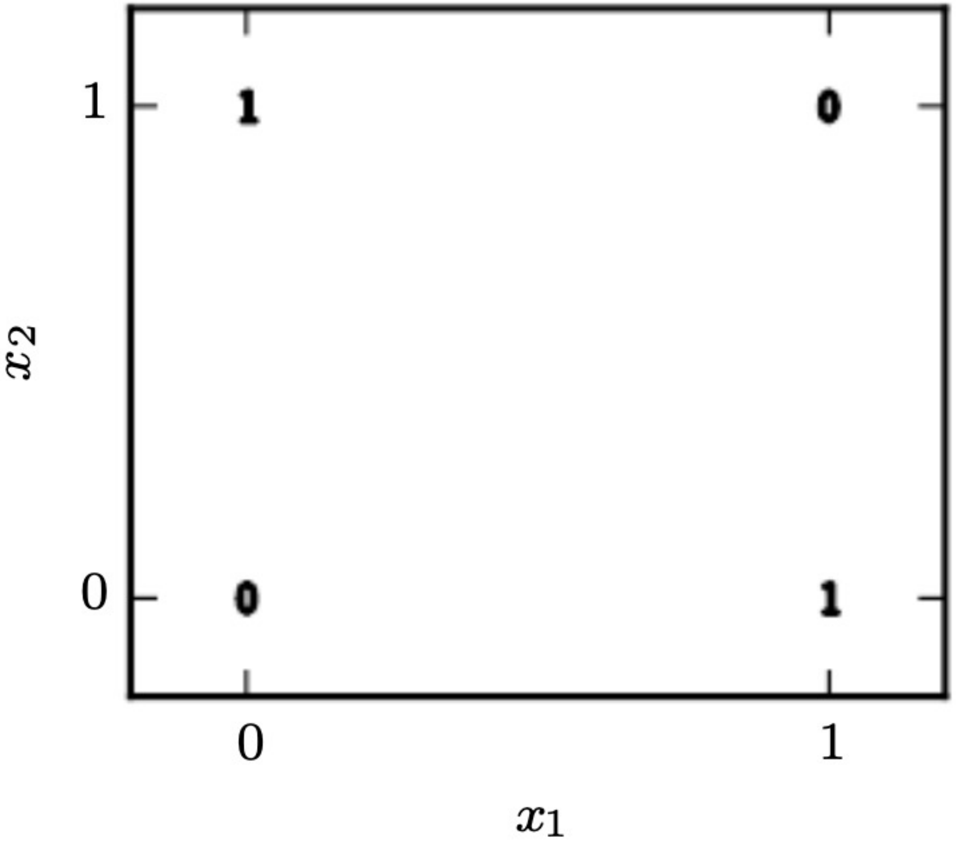
$$\begin{array}{ccc} & \begin{array}{cc} x_1 & x_2 \end{array} & \\ \boldsymbol{X} = & \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 0 \\ 1 & 1 \end{bmatrix} & \begin{array}{c} 0 \\ 1 \\ 1 \\ 0 \end{array} \end{array}$$



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

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

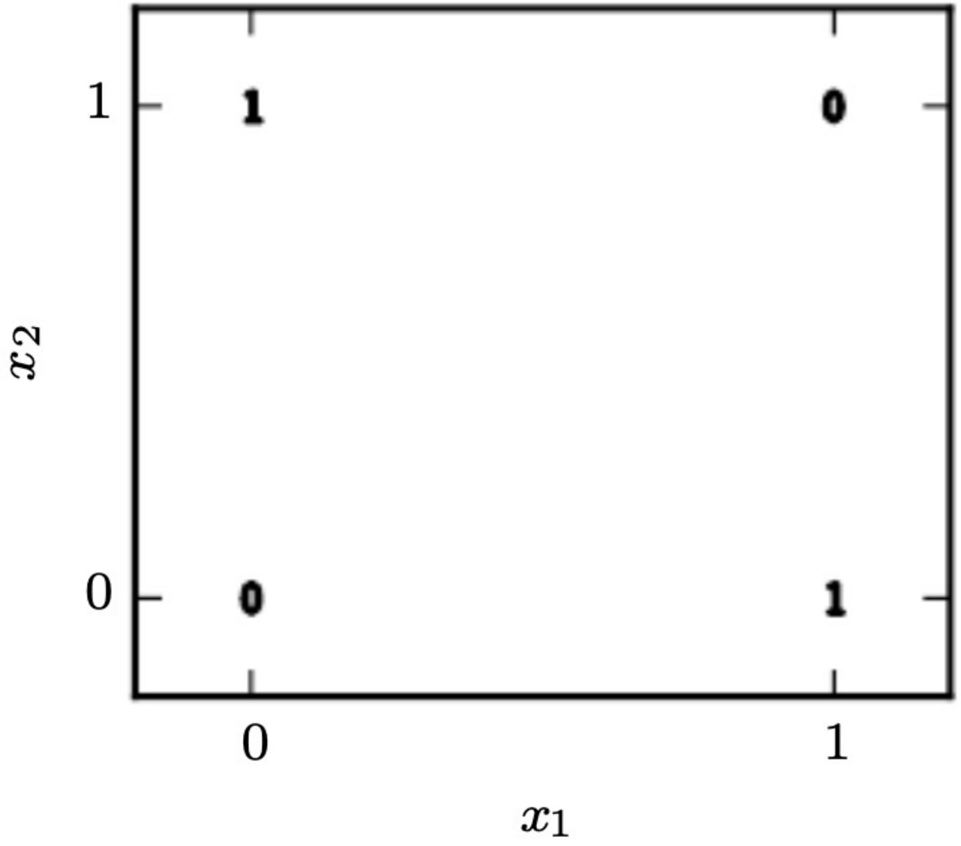


This utput → input of a linear function

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

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



$$\boxed{\begin{array}{c} \boldsymbol{W} = \left[\begin{array}{cc} 1 & 1 \\ 1 & 1 \end{array}\right], \\ \boldsymbol{c} = \left[\begin{array}{c} 0 \\ -1 \end{array}\right], \\ \boldsymbol{w} = \left[\begin{array}{c} 1 \\ -2 \end{array}\right], \end{array}}$$

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

Learning XOR function

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

The inner linear model

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

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

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

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

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

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

Learning XOR function

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

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

$$\begin{bmatrix} 0 & -1 \\ 1 & 0 \\ 1 & 0 \\ 2 & 1 \end{bmatrix} \xrightarrow{\max(0,\, \cdot)} \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 1 & 0 \\ 2 & 1 \end{bmatrix}$$

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

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

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

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

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

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

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

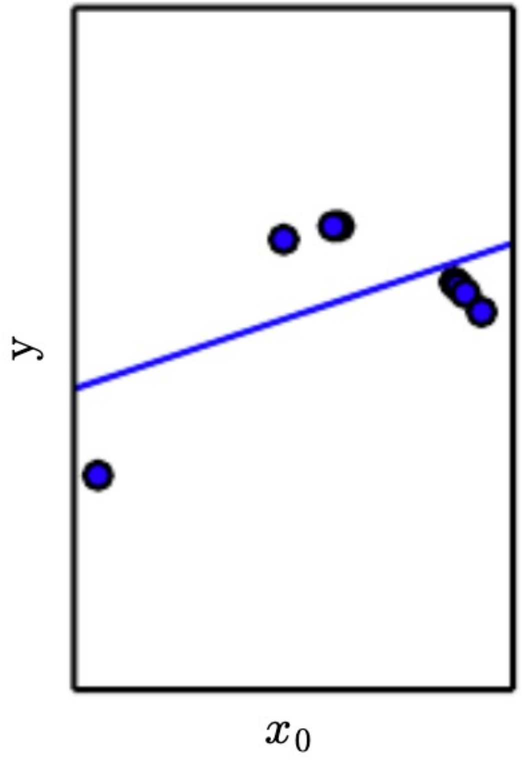
$$\boxed{\boldsymbol{w} = \begin{bmatrix} 1 \\ -2 \end{bmatrix}},$$

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

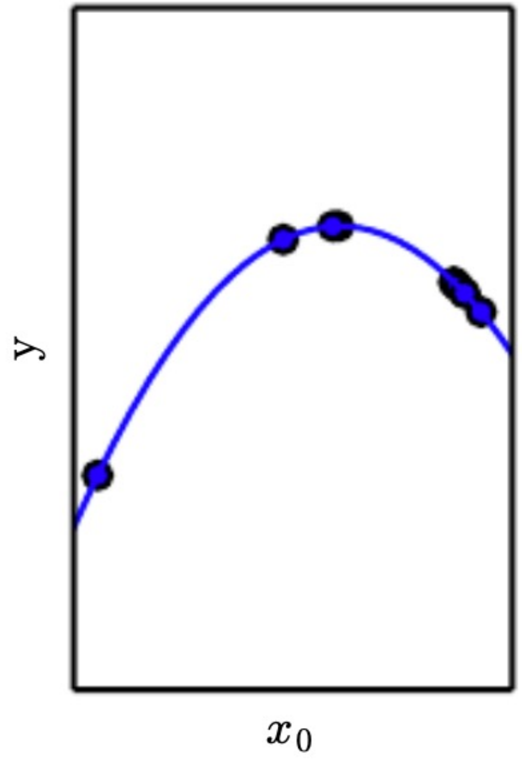
Learning XOR function

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

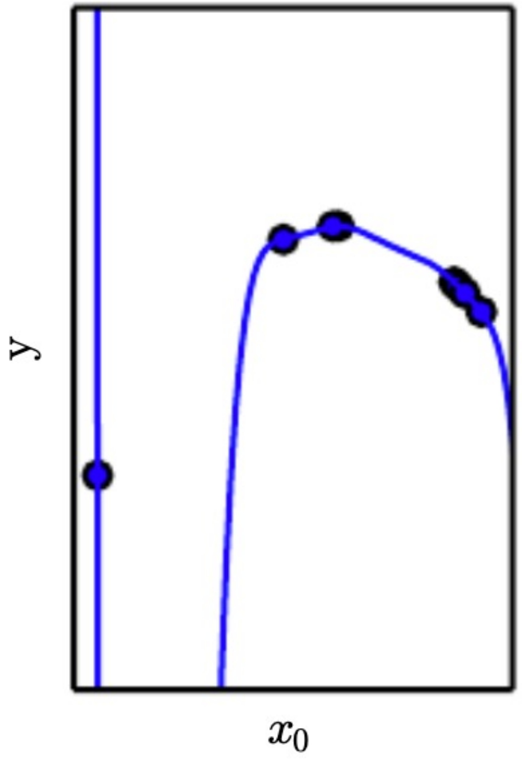
Learning XOR function



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



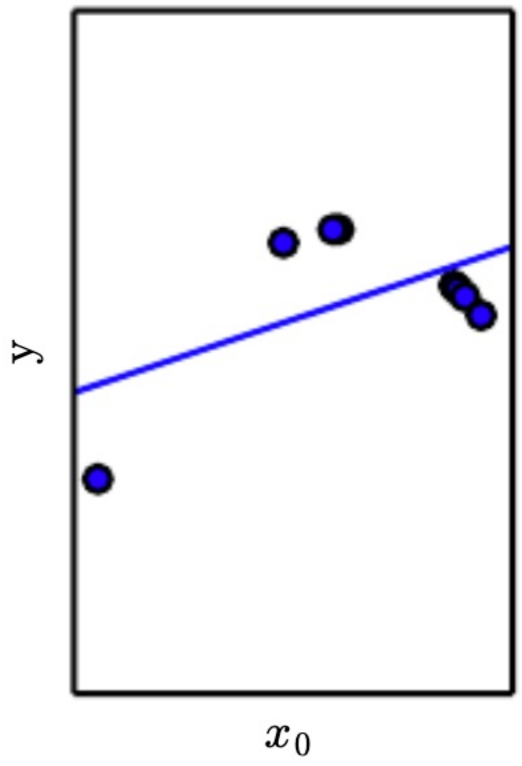
Quadratic model
$f(w;x) = w_2 x^2 + w_1 x^1 + w_0$



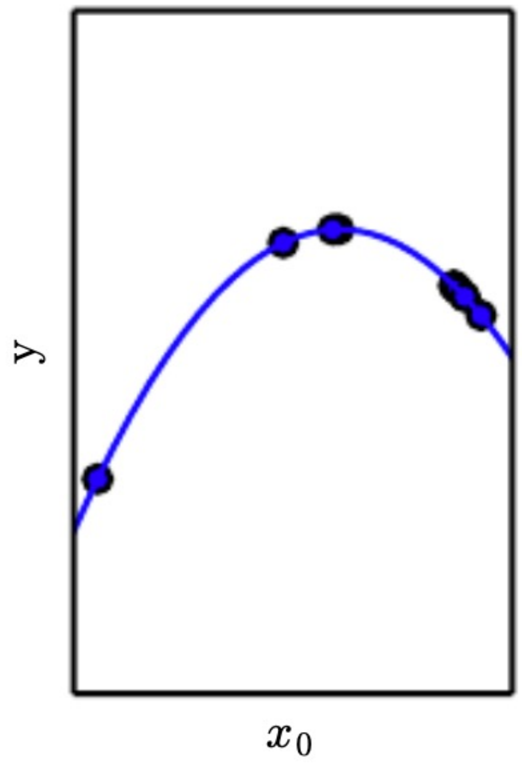
Polynomial model (9 degree)

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

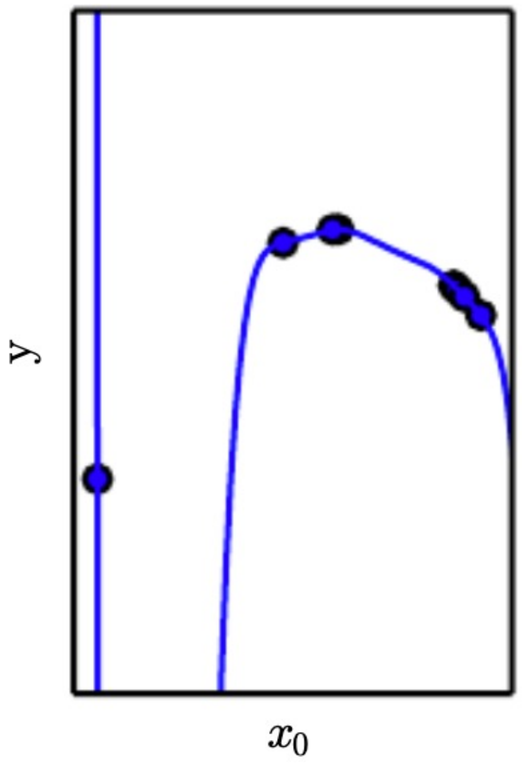
Learning XOR function



Linear model
$$f(w;x) = w_1 x^1 + w_0$$
$$= \sum_{i=2}^{9} 0 \cdot x^i + \sum_{i=1}^{1} w_i x^i + w_0$$

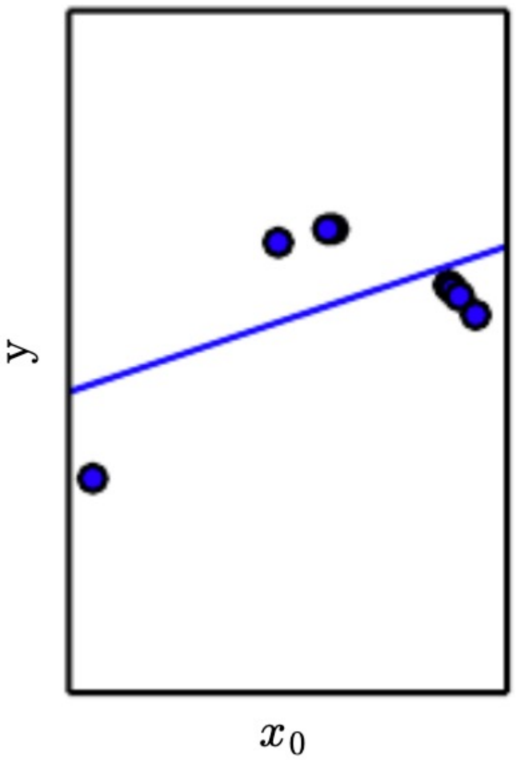


Quadratic model
$$f(w;x) = w_2 x^2 + w_1 x^1 + w_0$$



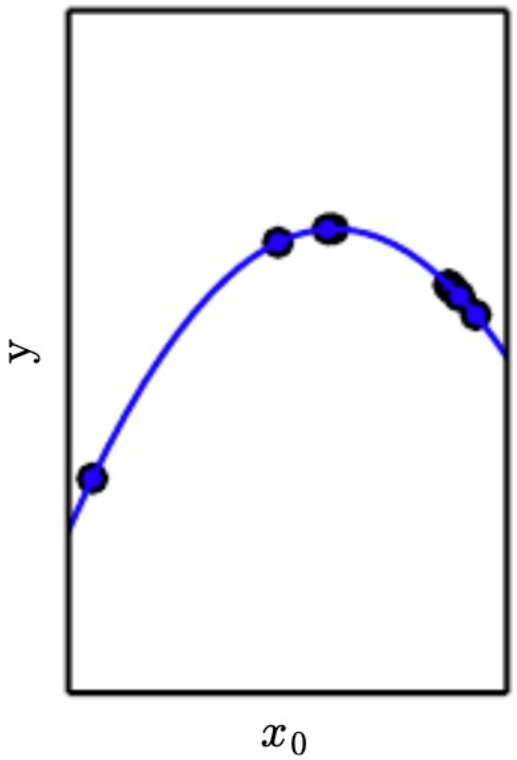
Polynomial model (9 degree)
$$f(w;x) = \sum_{i=1}^{9} w_i x^i + w_0$$

Learning XOR function



Linear model

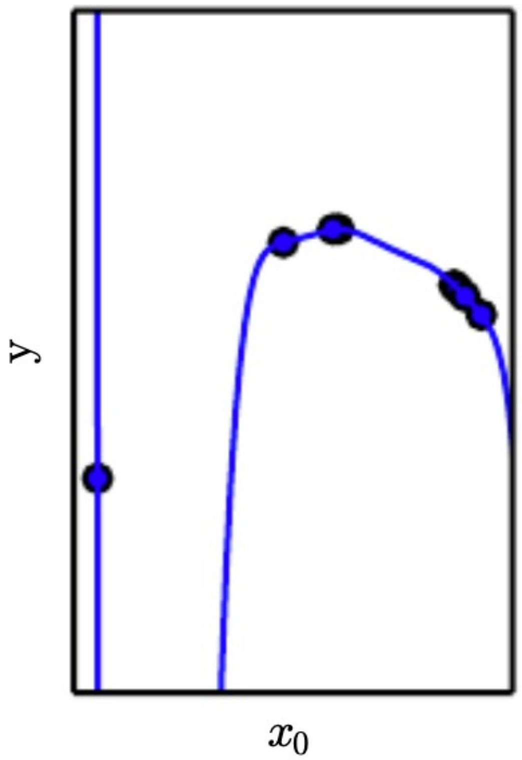
$$f(w;x) = w_1 x^1 + w_0$$



Quadratic model

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

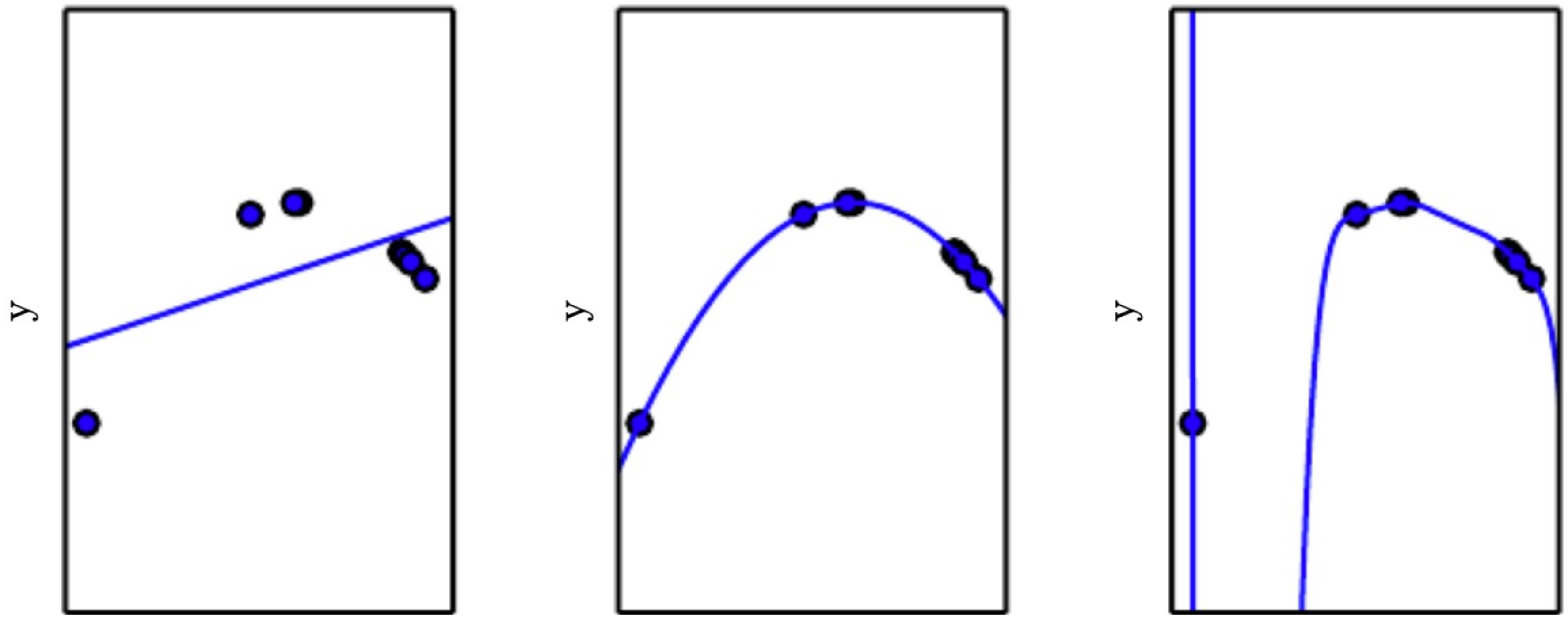
$$= \sum_{i=3}^{9} 0 \cdot x^i + \sum_{i=1}^{2} w_i x^i + w_0$$



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

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

v.s.

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

nonlinear model: better capacity

Raw features

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

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

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

What makes feedforward network different from linear model

- Nonlinear functions in hidden layers

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

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

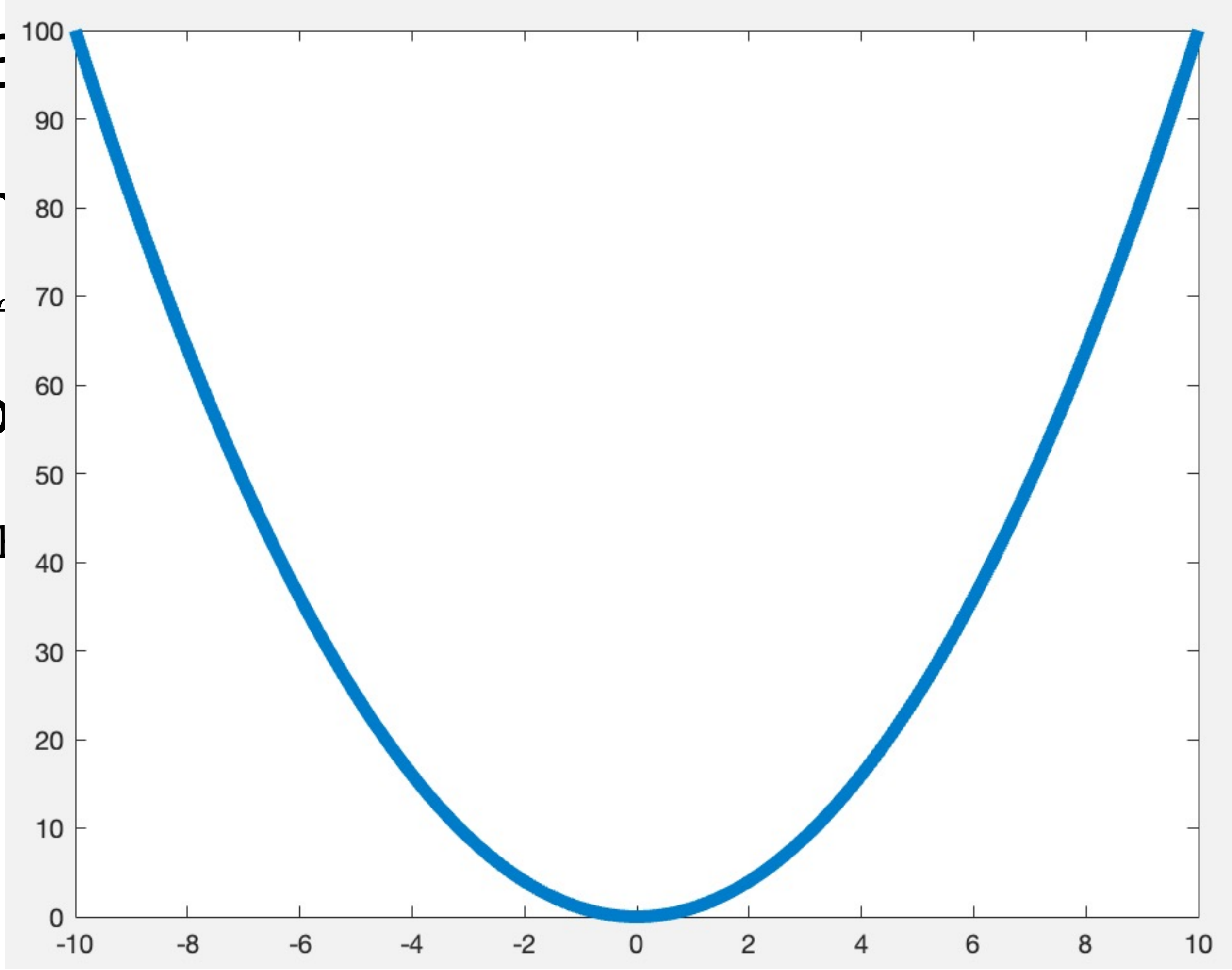
What makes feedforward network different from linea

- Nonlinear fur

$f_3(f$

- Q: Why comp

$f(x) = g($



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

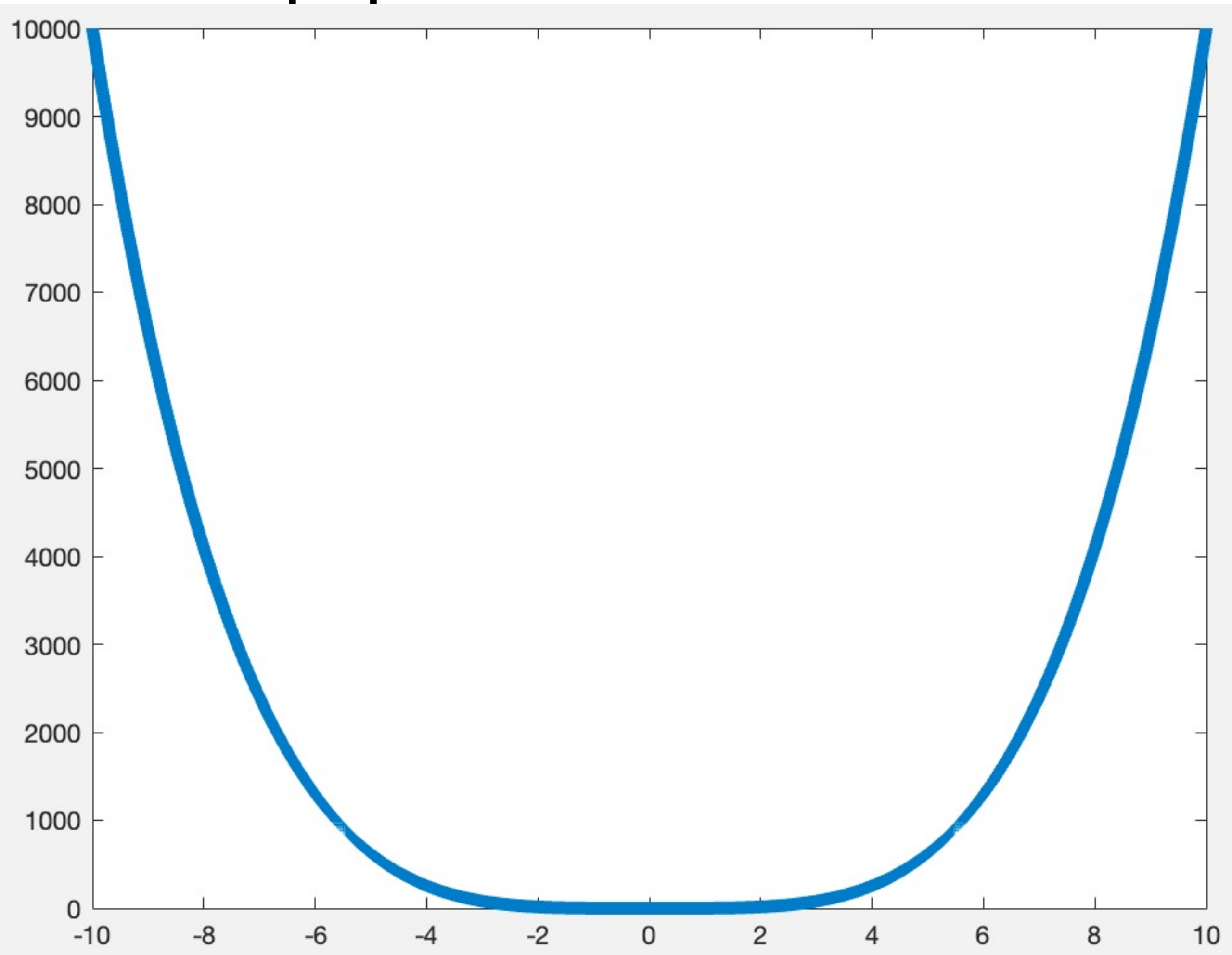
What makes feedforward network different from linea

- Nonlinear fur

$f_3 ($

- Q: Why comp

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



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

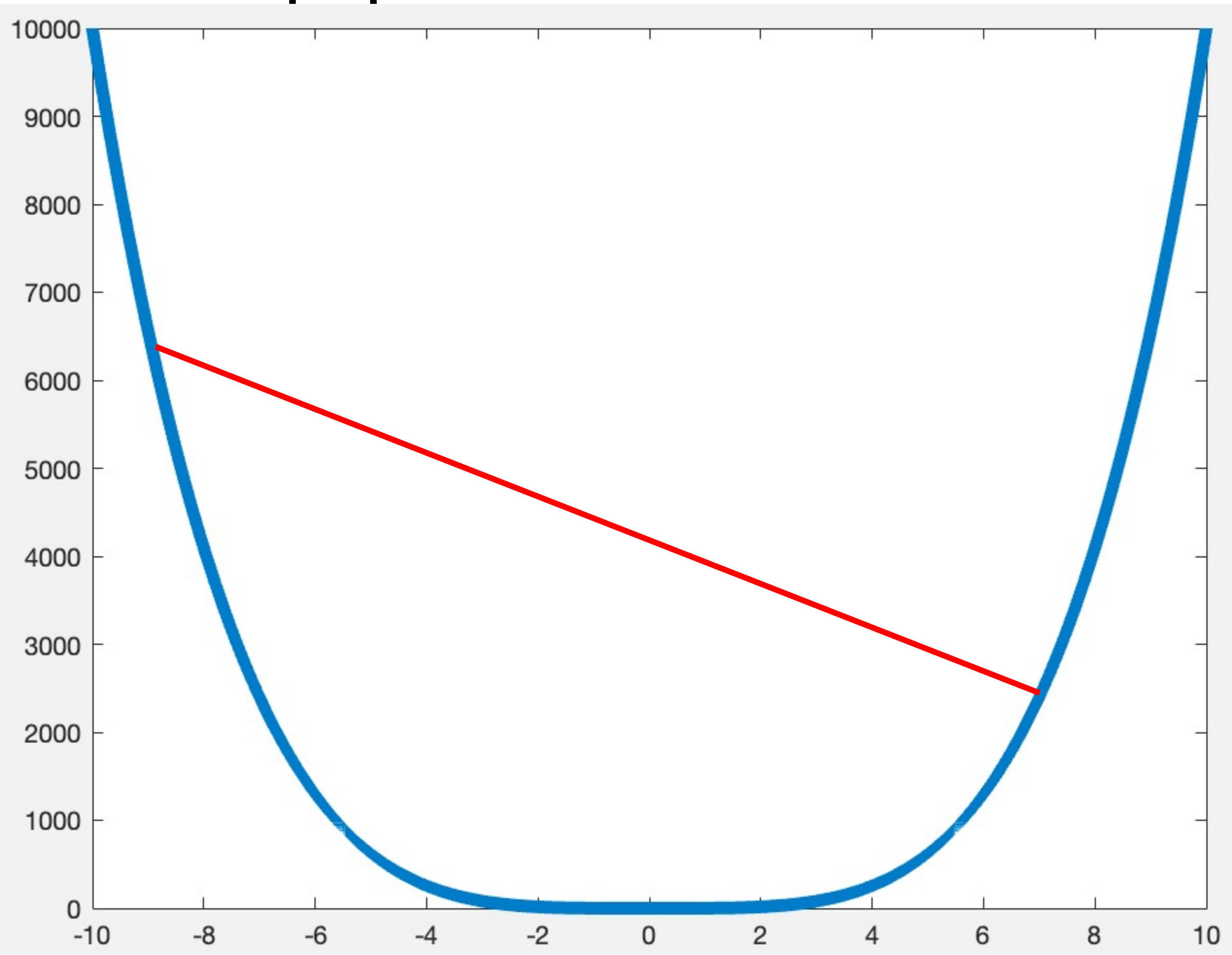
What makes feedforward network different from linea

- Nonlinear fur

$f_3($

- Q: Why comp

$f(x) = g($



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

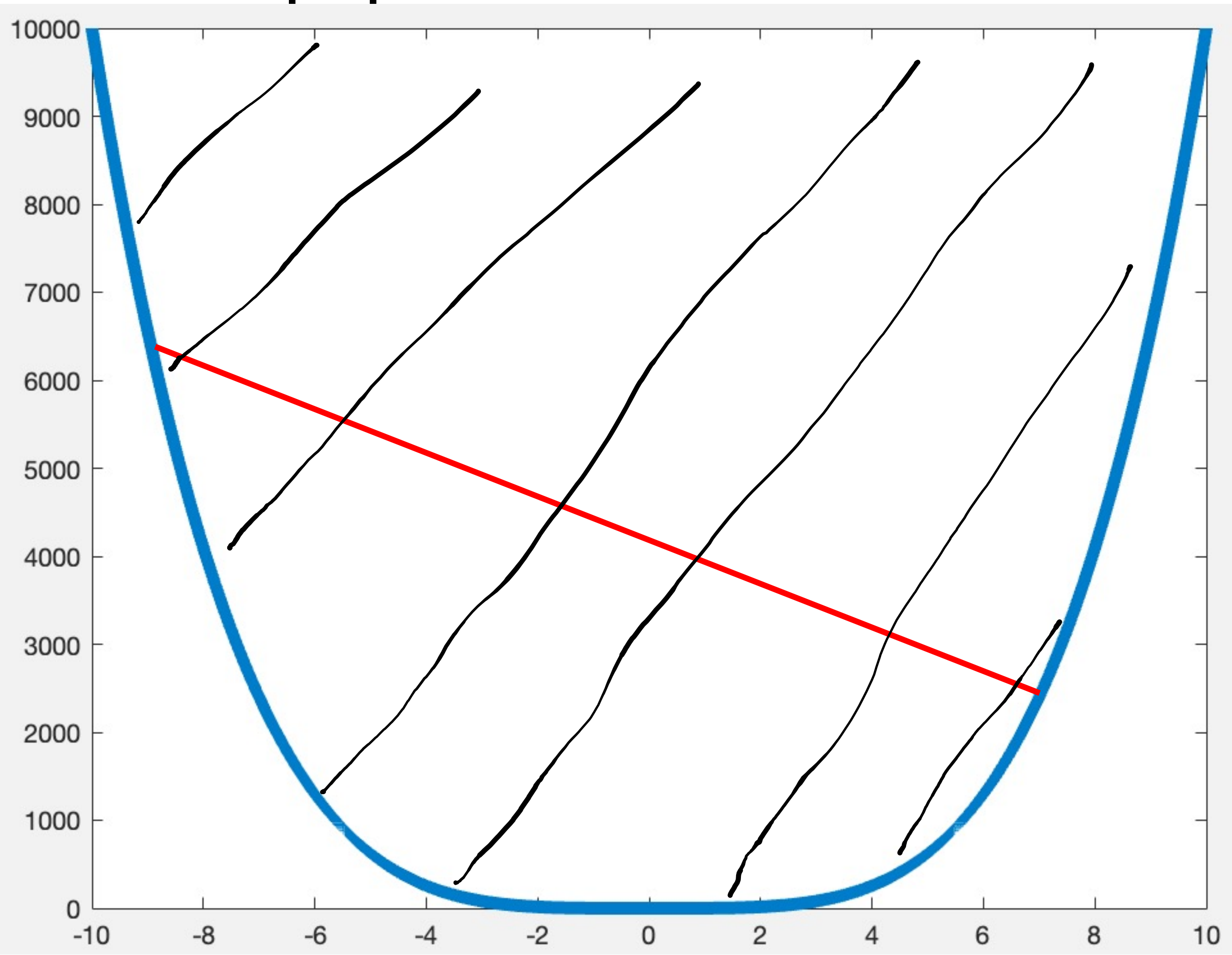
What makes feedforward network different from linea

- Nonlinear fur

$f_3(f$

- 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

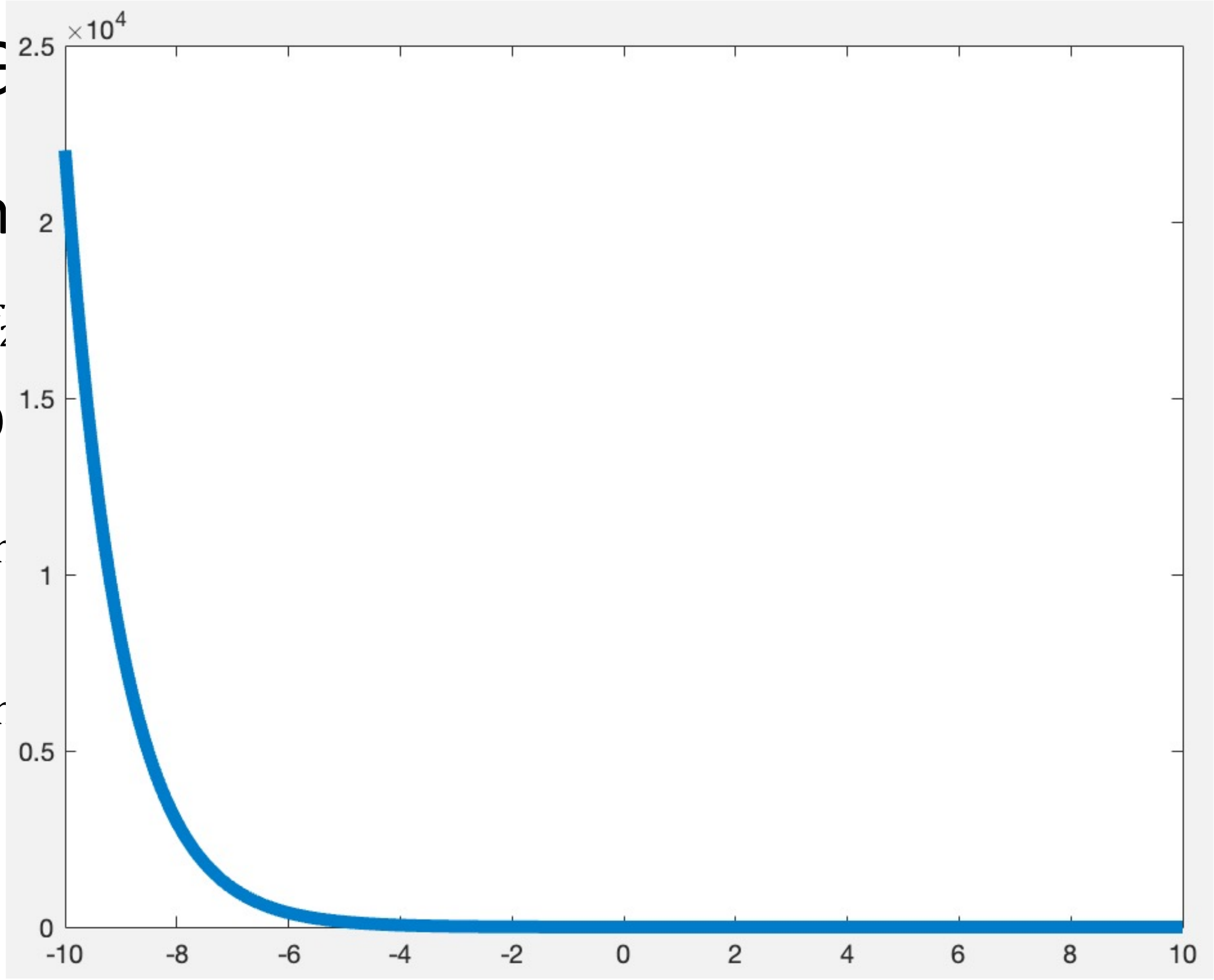
- Nonlinear fun

$f_3 (f_2$

- Q: Why comp

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

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



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

What makes feedforward network different from linear

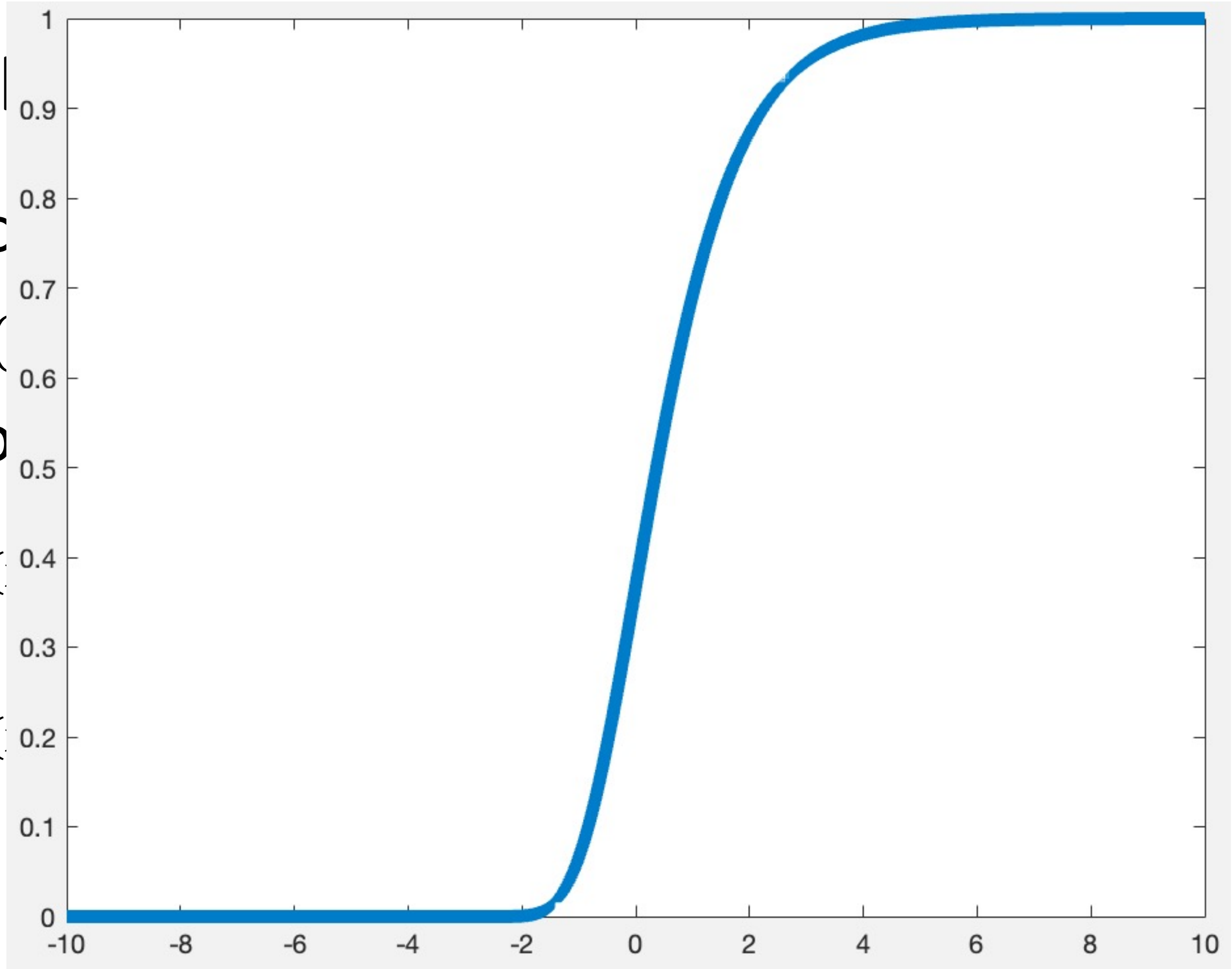
- Nonlinear func

$$f_3\left(f_2(\right.$$

- 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

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

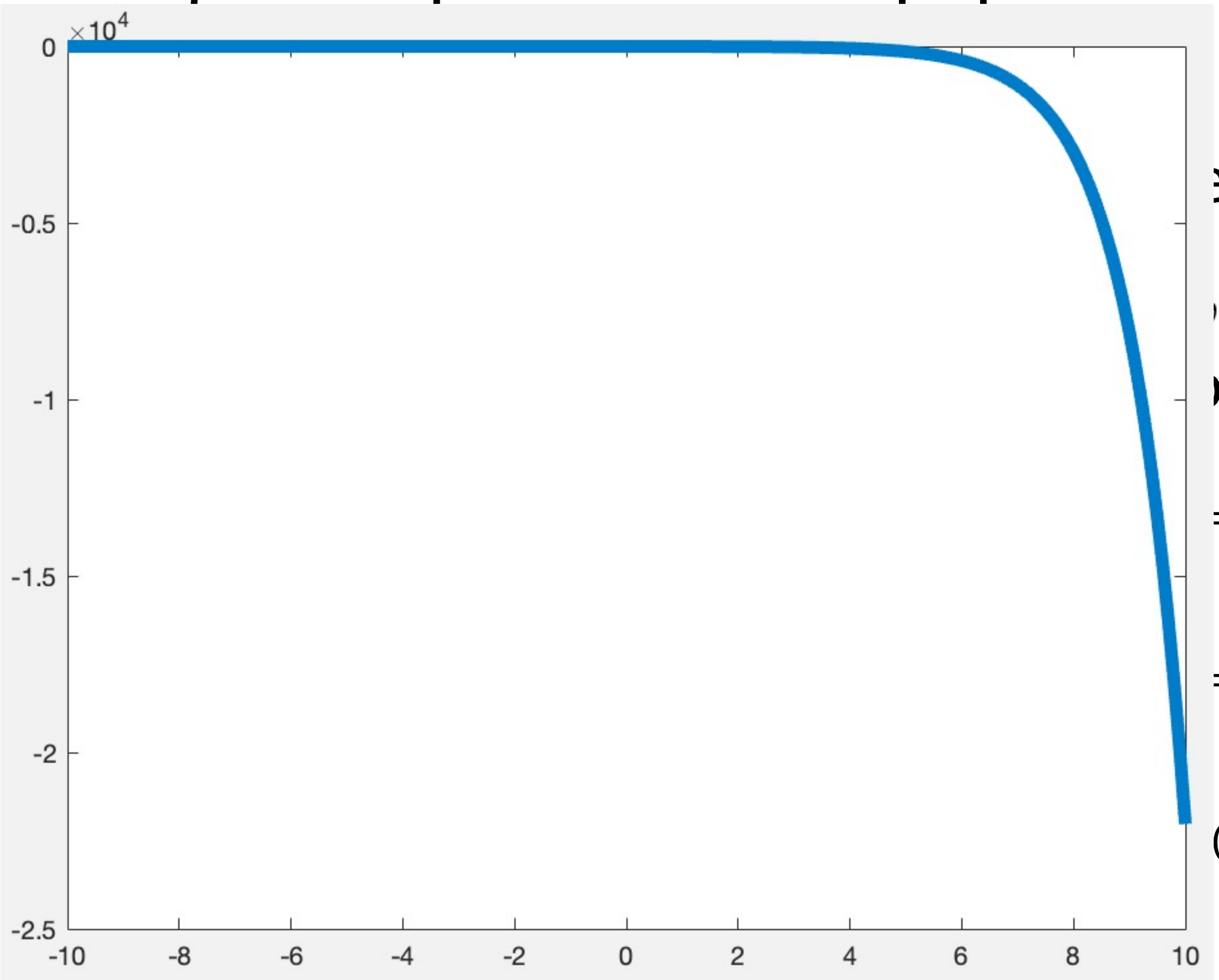
- Q: Why composition makes nonconvexity?

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

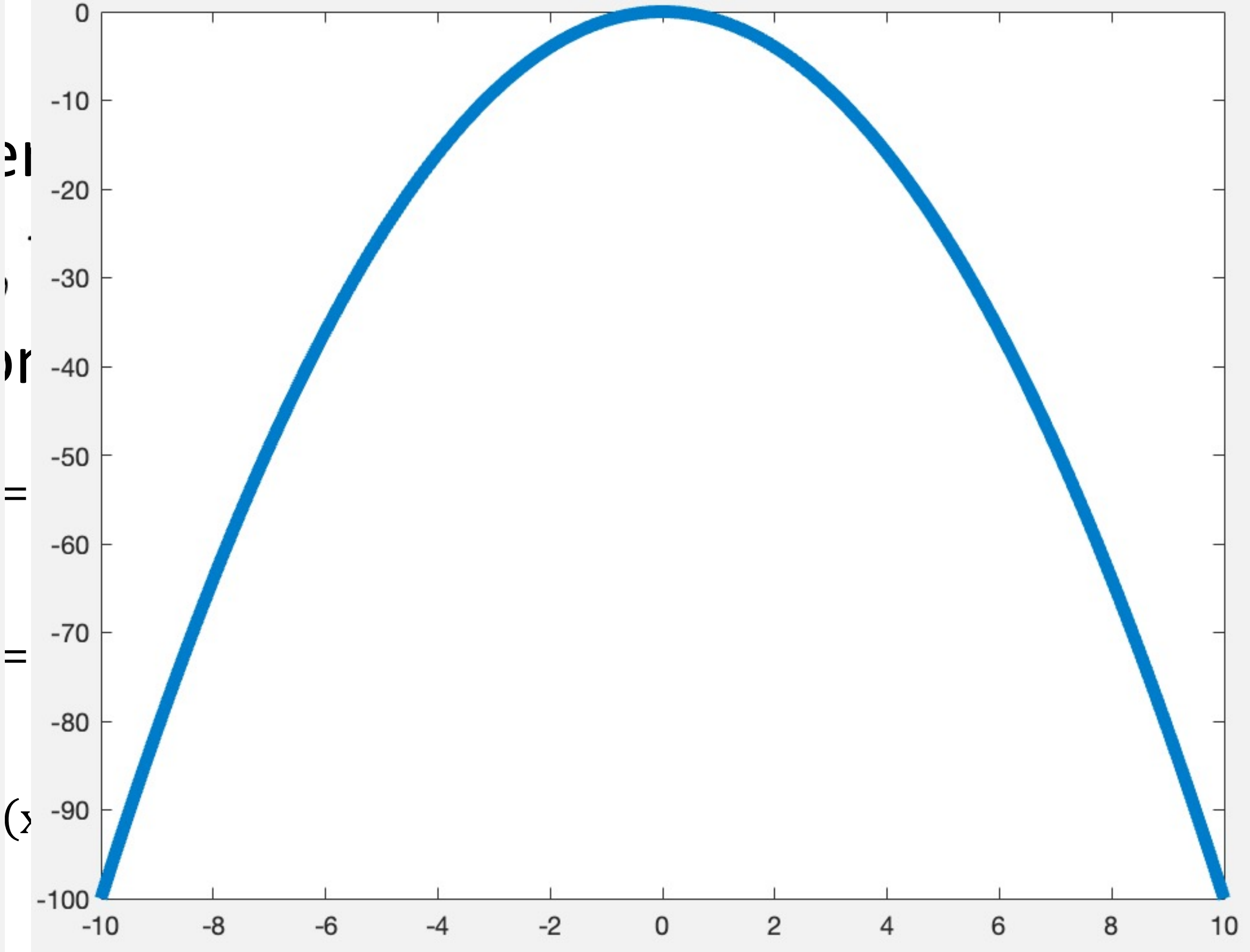
$f(x) = \mathrm{g}(\mathrm{h}(\mathrm{x}))$ where $\mathrm{g}(\mathrm{x}) = \mathrm{h}(\mathrm{x}) = \exp(-\mathrm{x})$

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

What makes feedforward network different from linear models



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



$h(x) = -x^2$

What makes feedforward network different from linear model

- Nonlinear fur

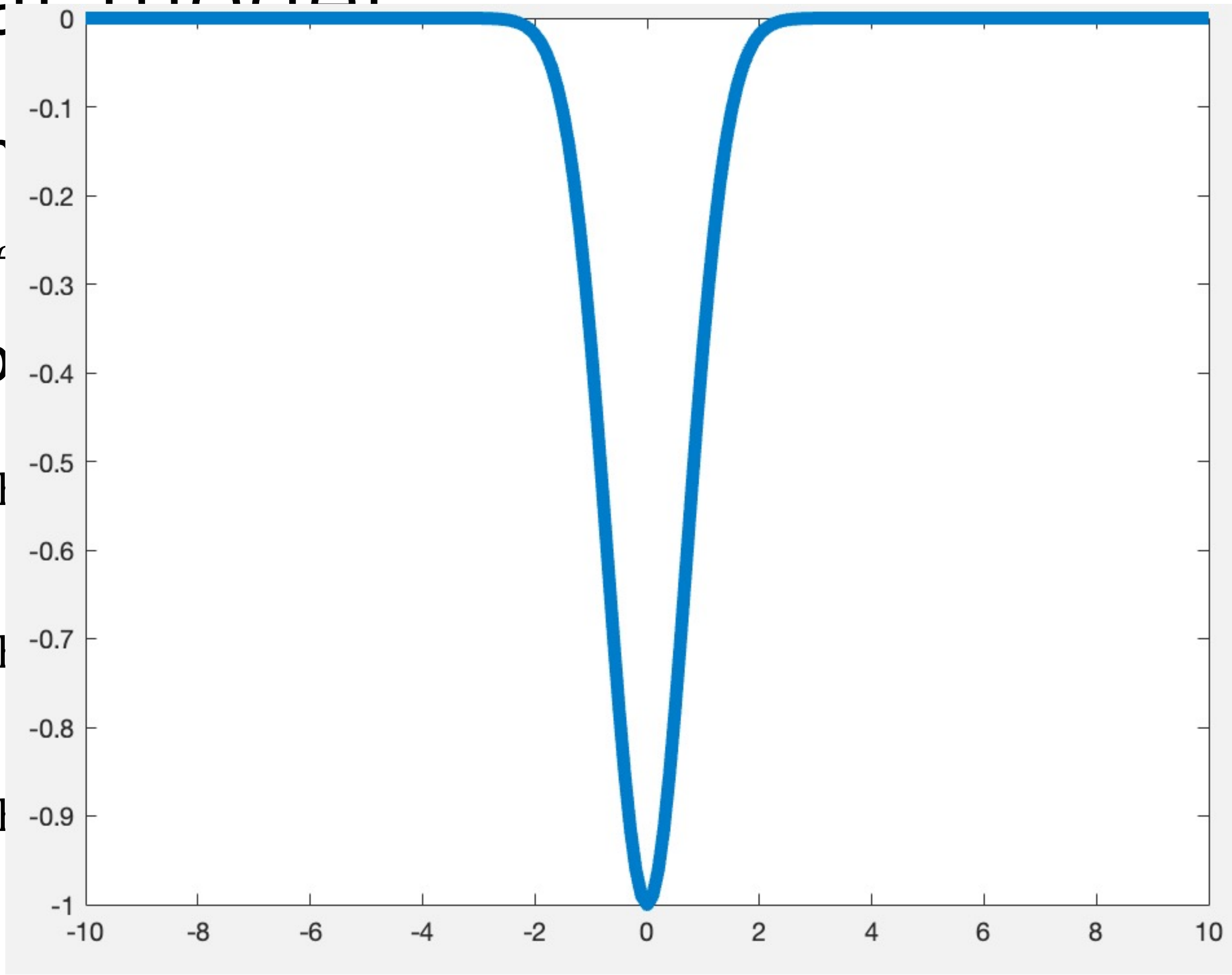
$f_3(f$

- Q: Why comp

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

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

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



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

What makes feedforward network different from linear model

- Nonlinear fur

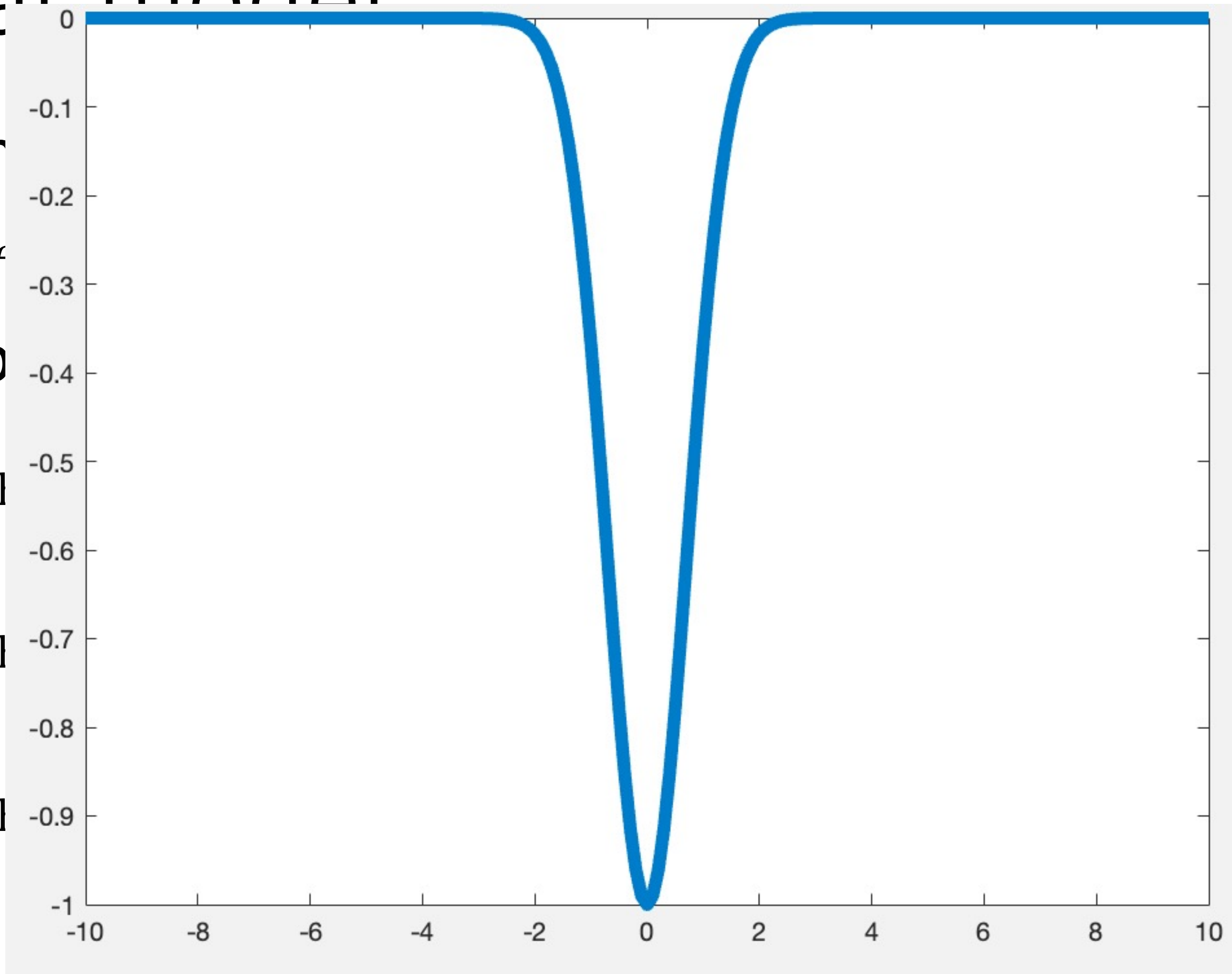
$f_3\big(f$

- 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

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

- Q: Why composition makes nonconvexity?

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

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

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

- Q: Why composition makes nonconvexity?

$$\boxed{\begin{array}{l} f \text{ is convex if } h \text{ is convex and nondecreasing, and } g \text{ is convex,} \\ f \text{ is convex if } h \text{ is convex and nonincreasing, and } g \text{ is concave,} \\ f \text{ is concave if } h \text{ is concave and nondecreasing, and } g \text{ is concave,} \\ f \text{ is concave if } h \text{ is concave and nonincreasing, and } g \text{ is convex.} \end{array}}$$

- 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