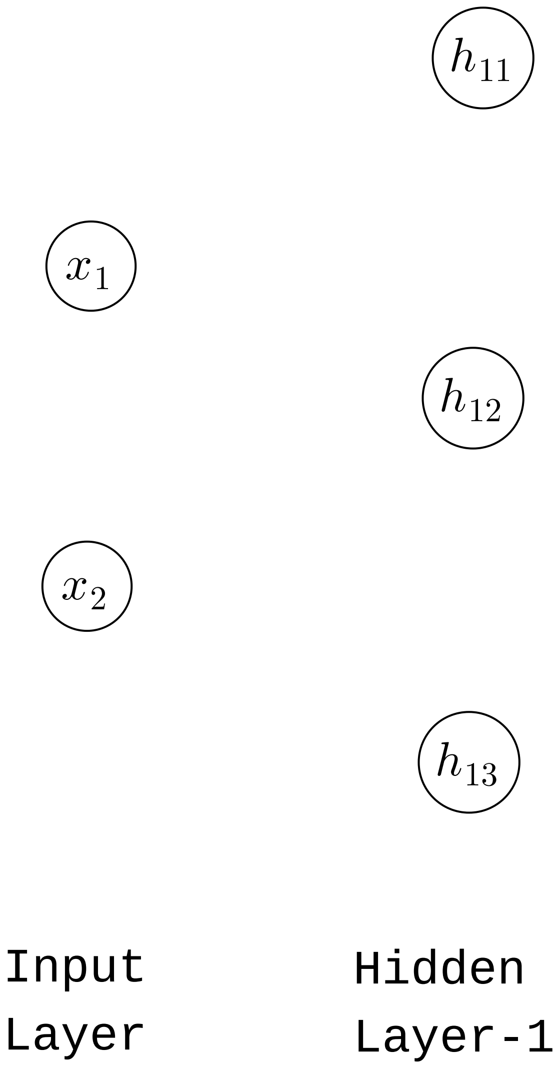
Neural Networks

Machine Learning Techniques

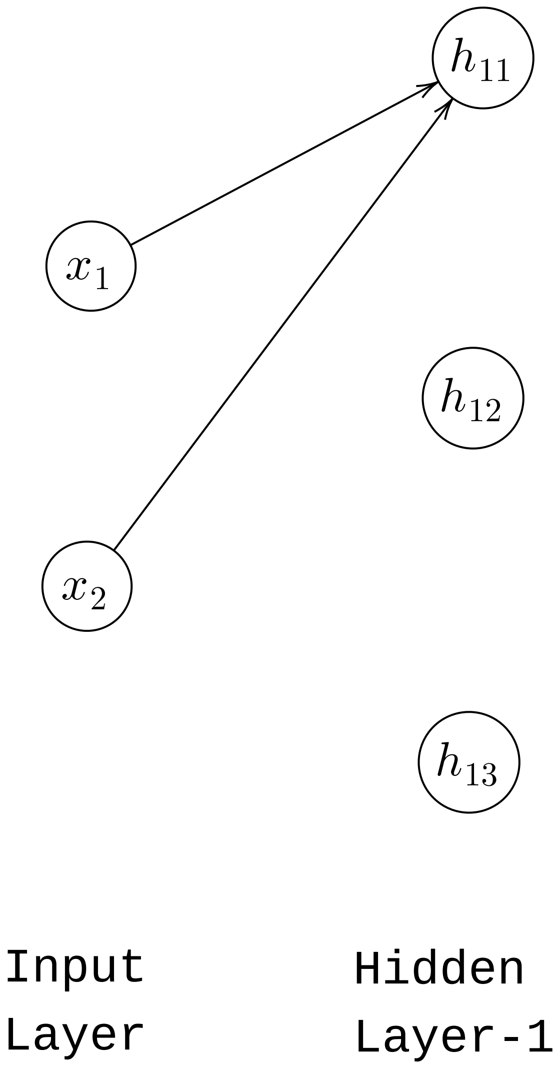
Network Architecture

Input
Layer

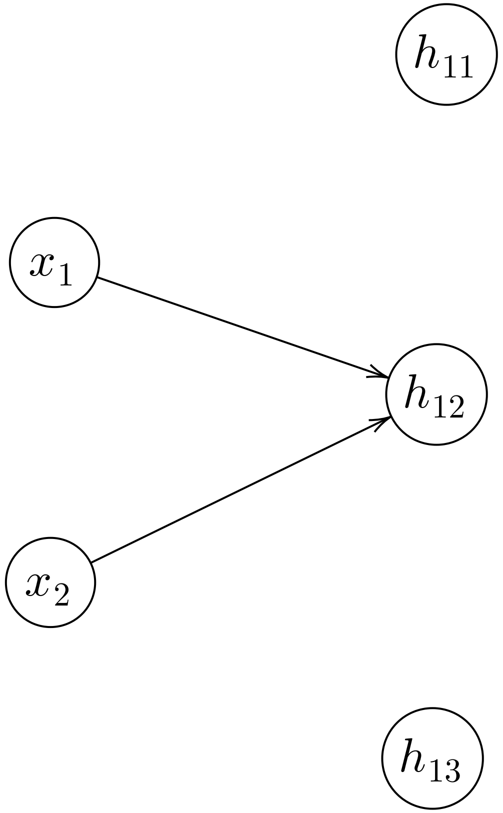
Network Architecture



Network Architecture



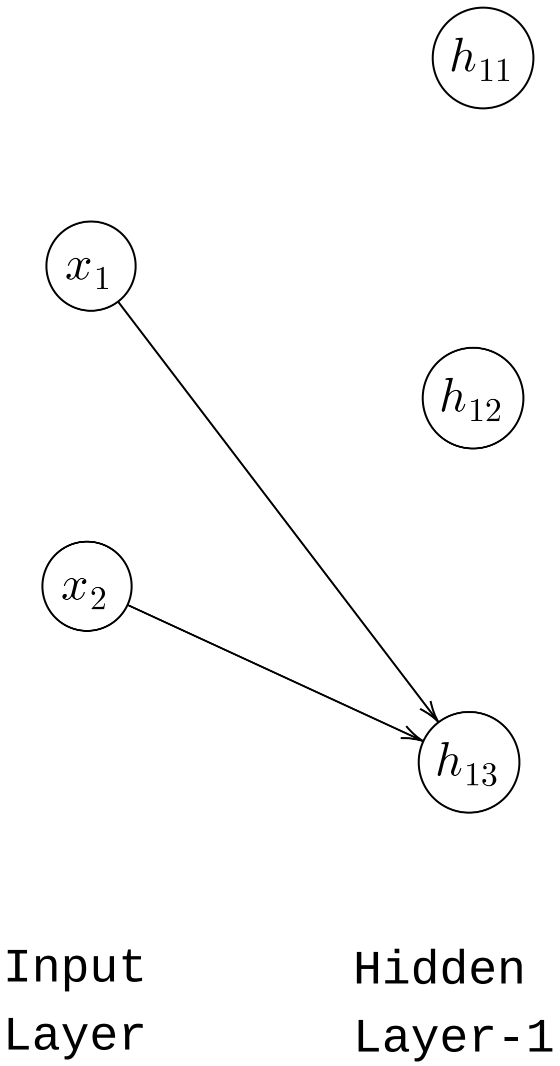
Network Architecture



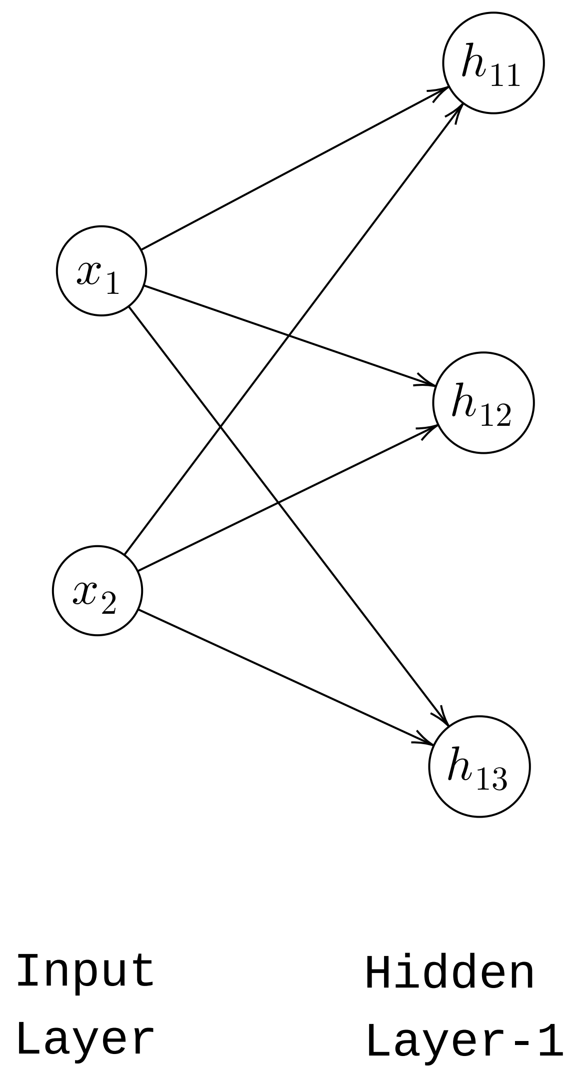
Input
Layer

Hidden
Layer-1

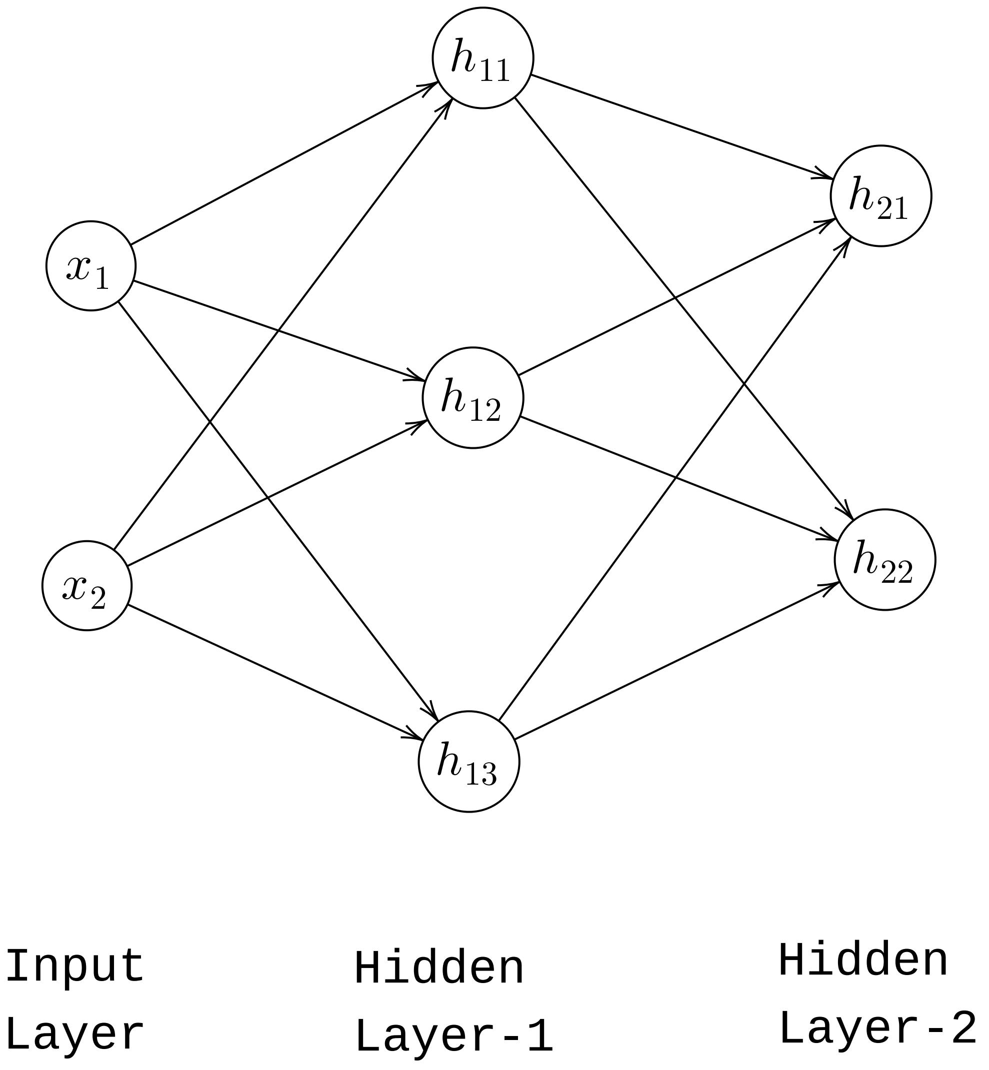
Network Architecture



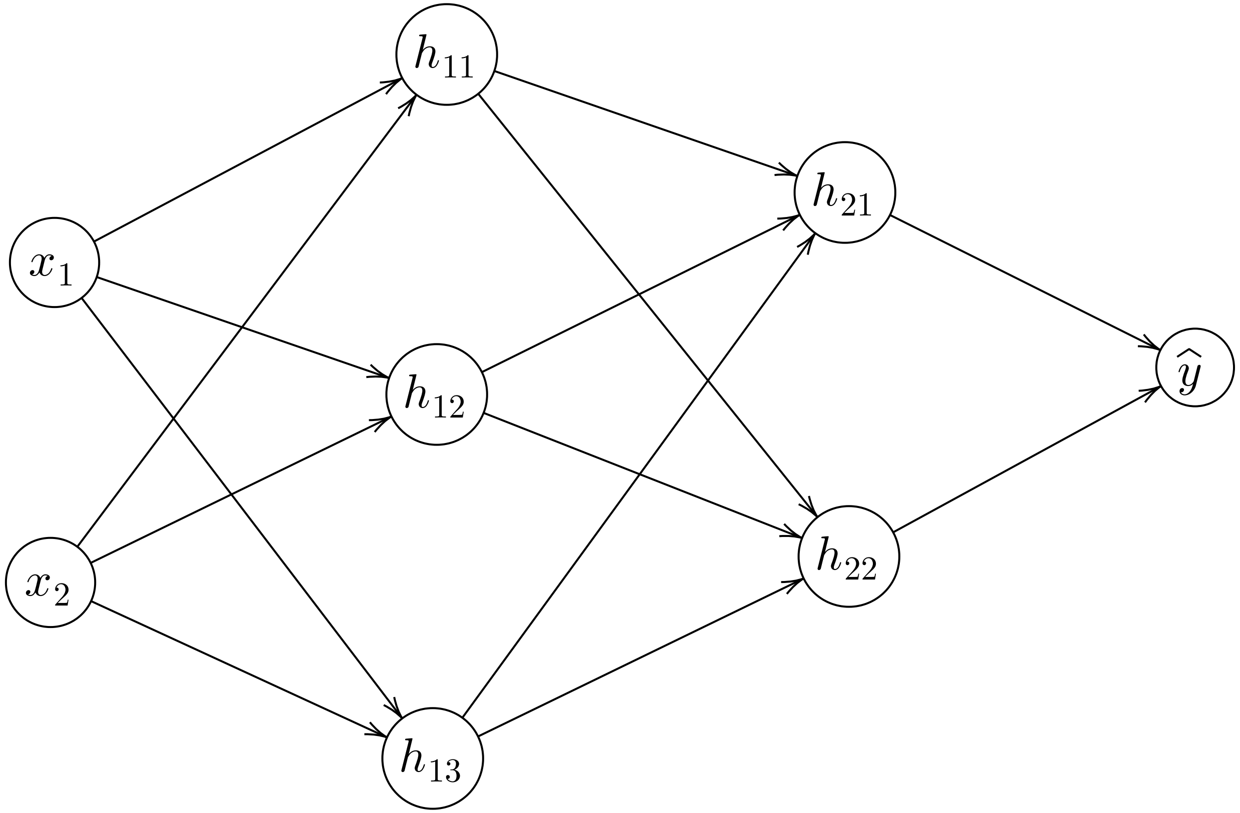
Network Architecture



Network Architecture



Network Architecture



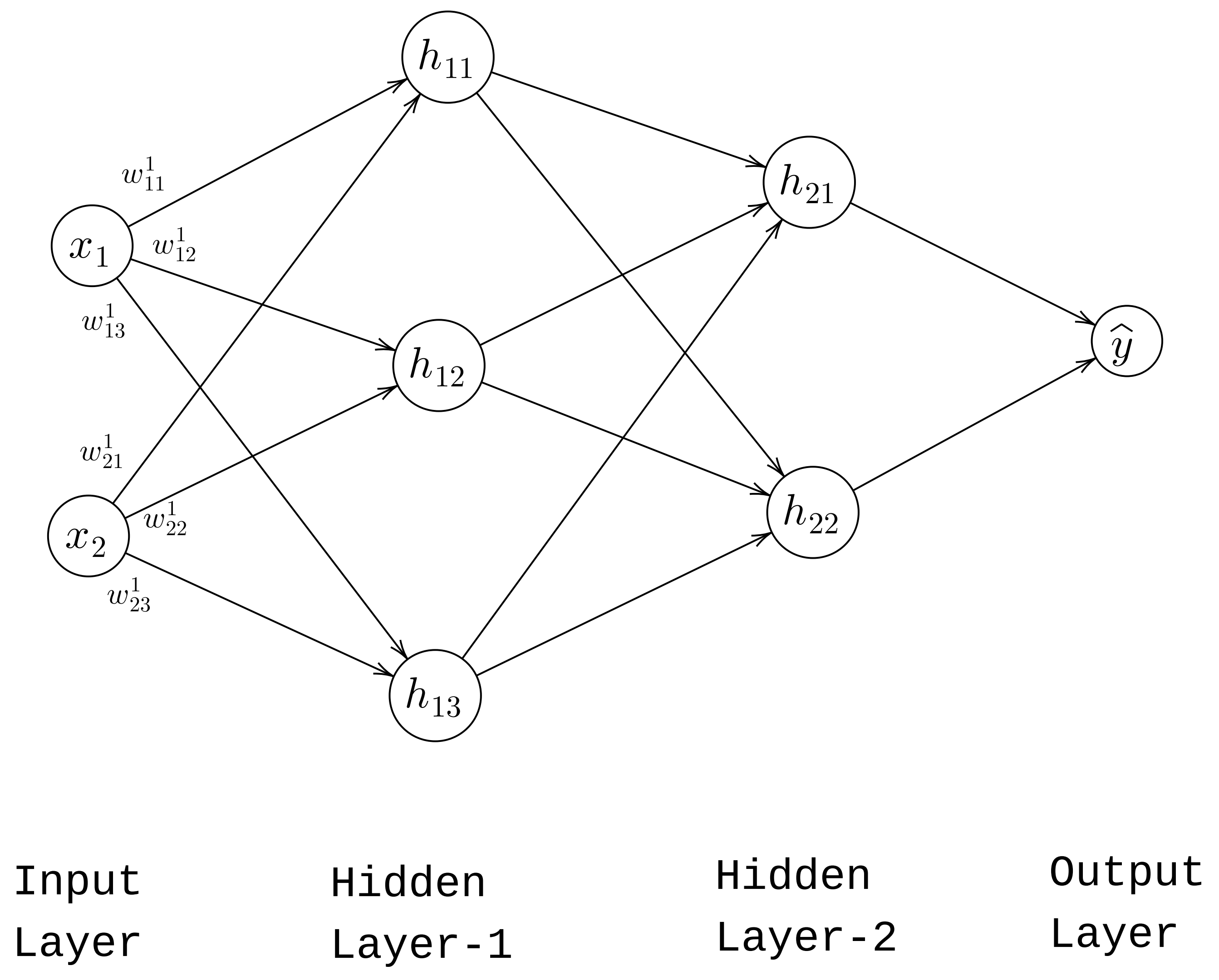
Input Layer

Hidden Layer-1

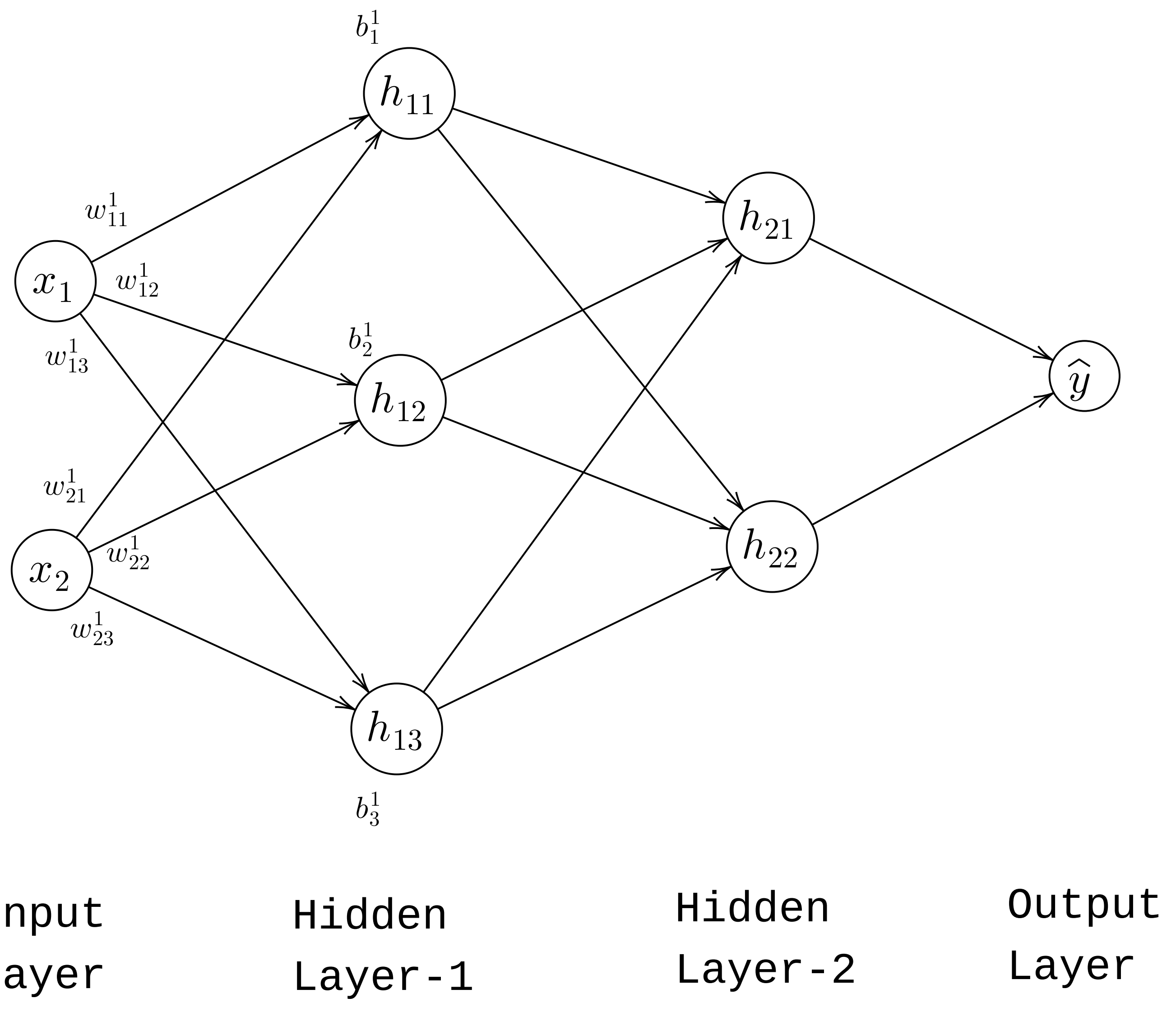
Hidden Layer-2

Output Layer

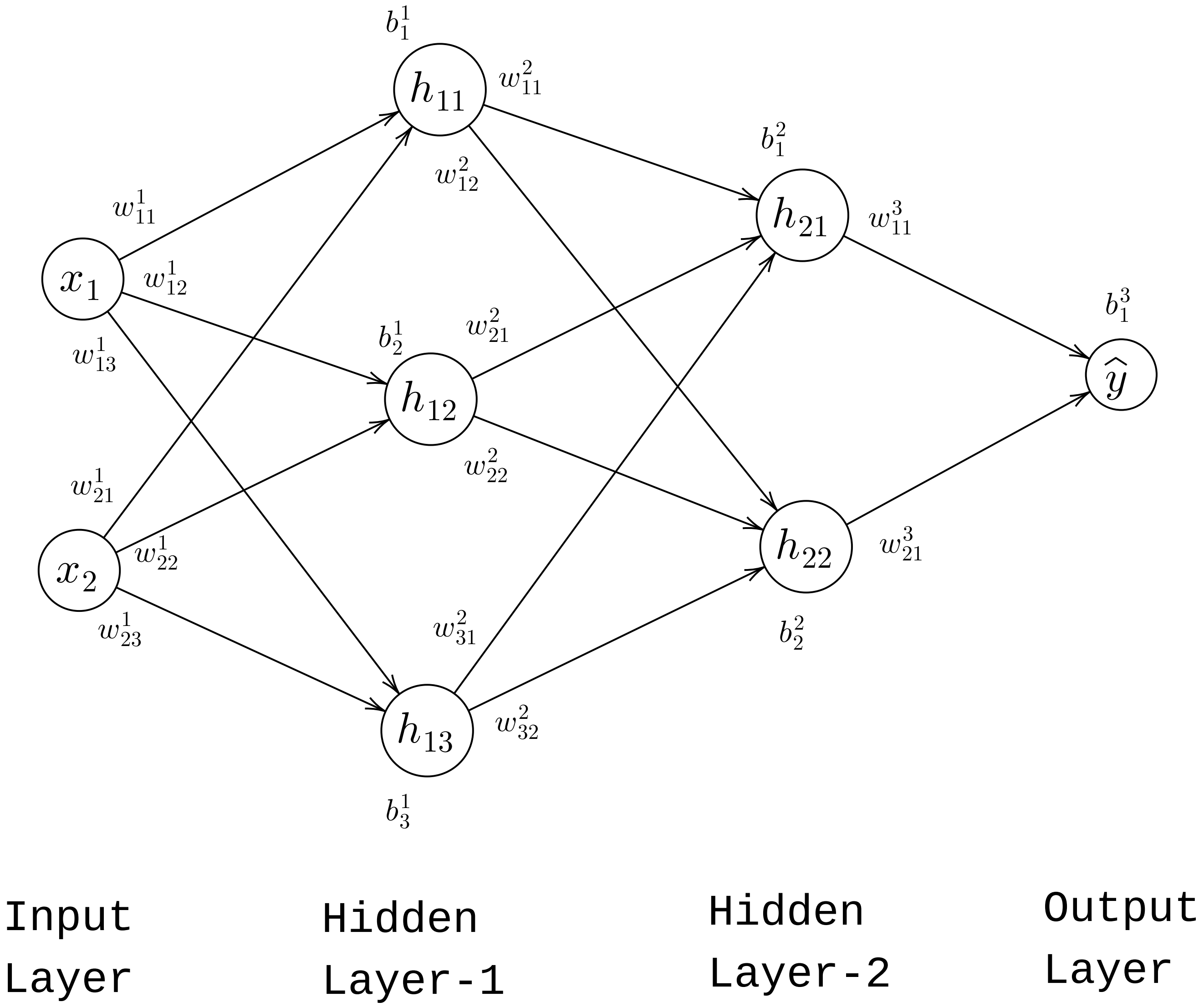
Network Architecture



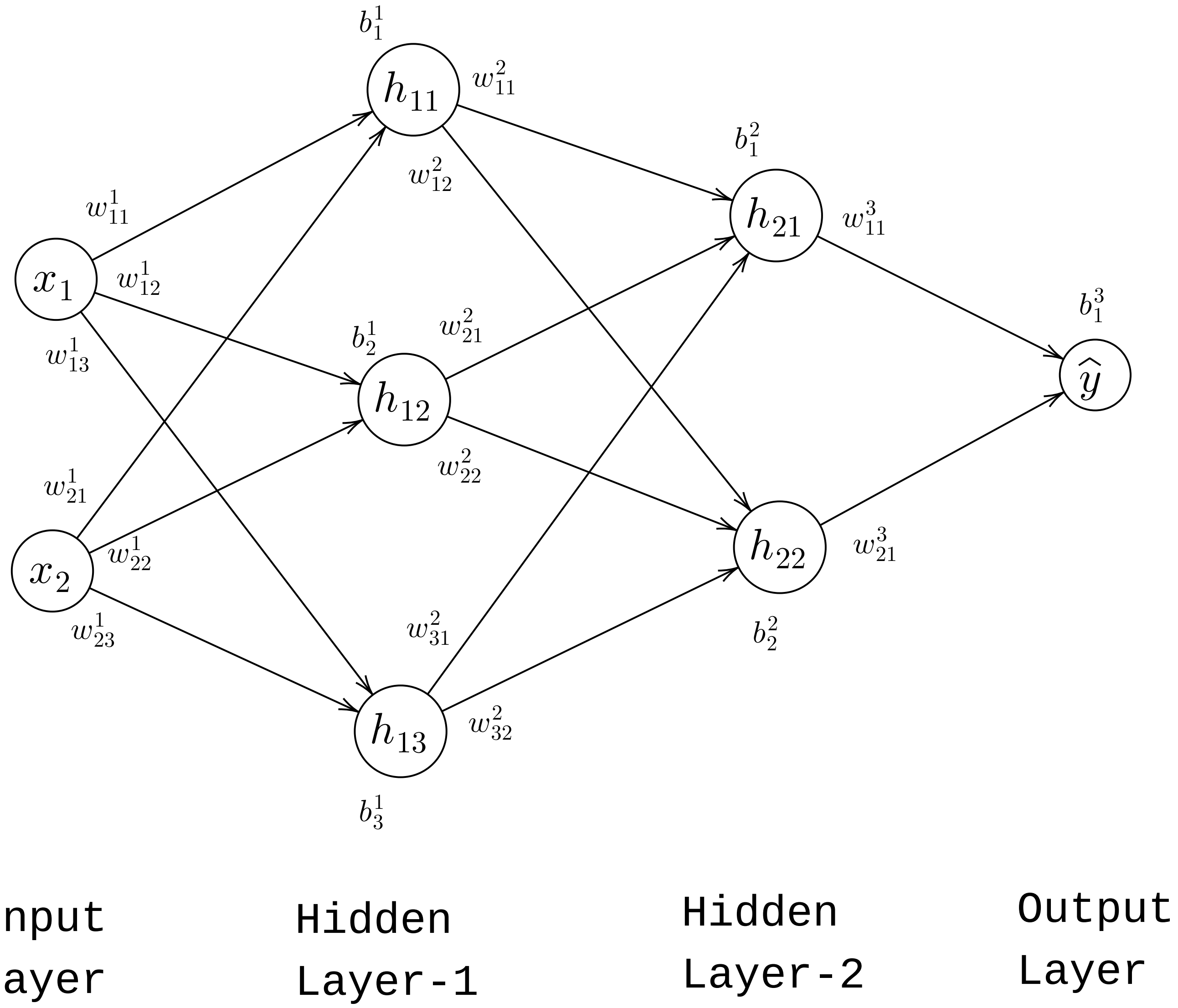
Network Architecture



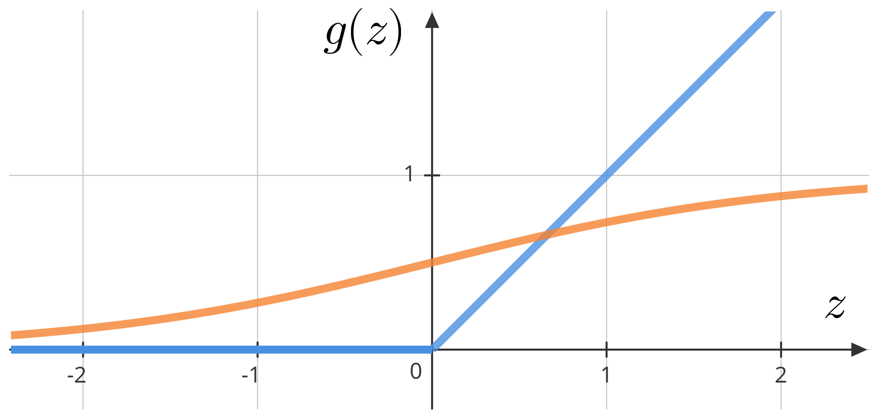
Network Architecture



Network Architecture



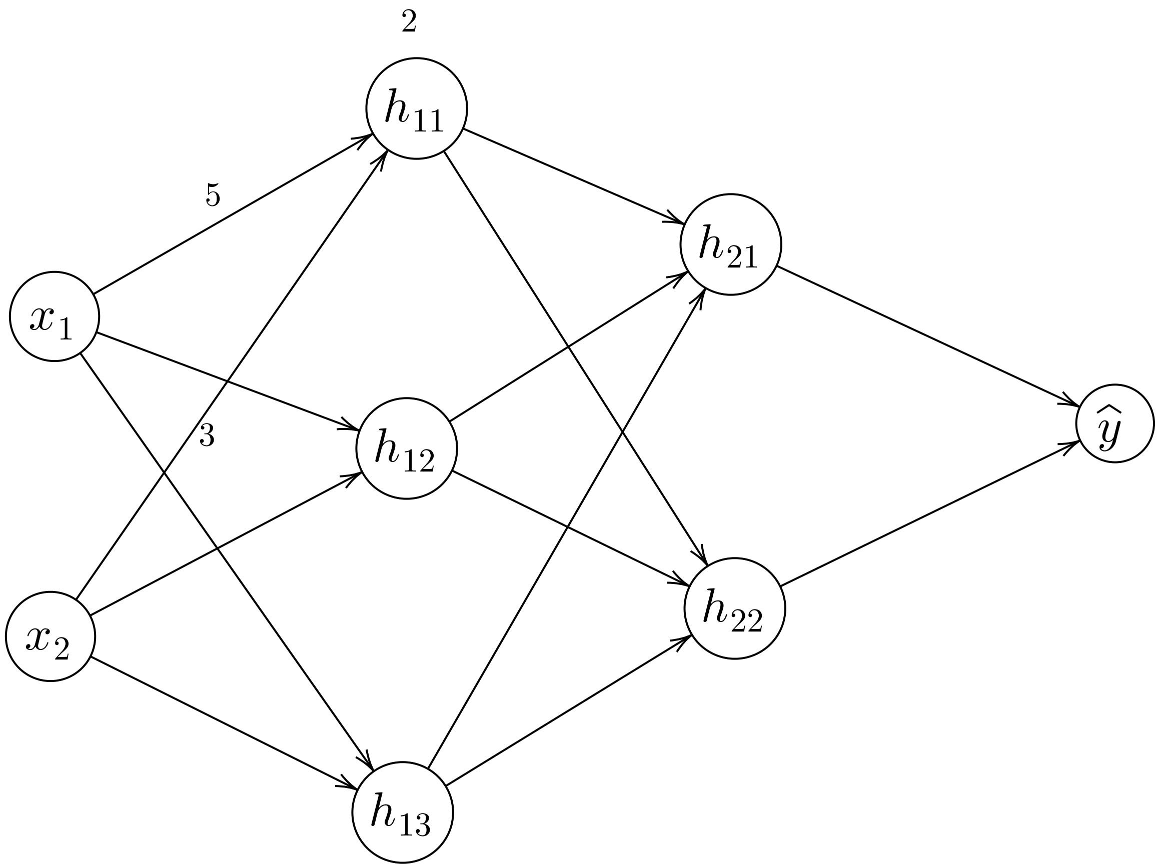
Activation Functions



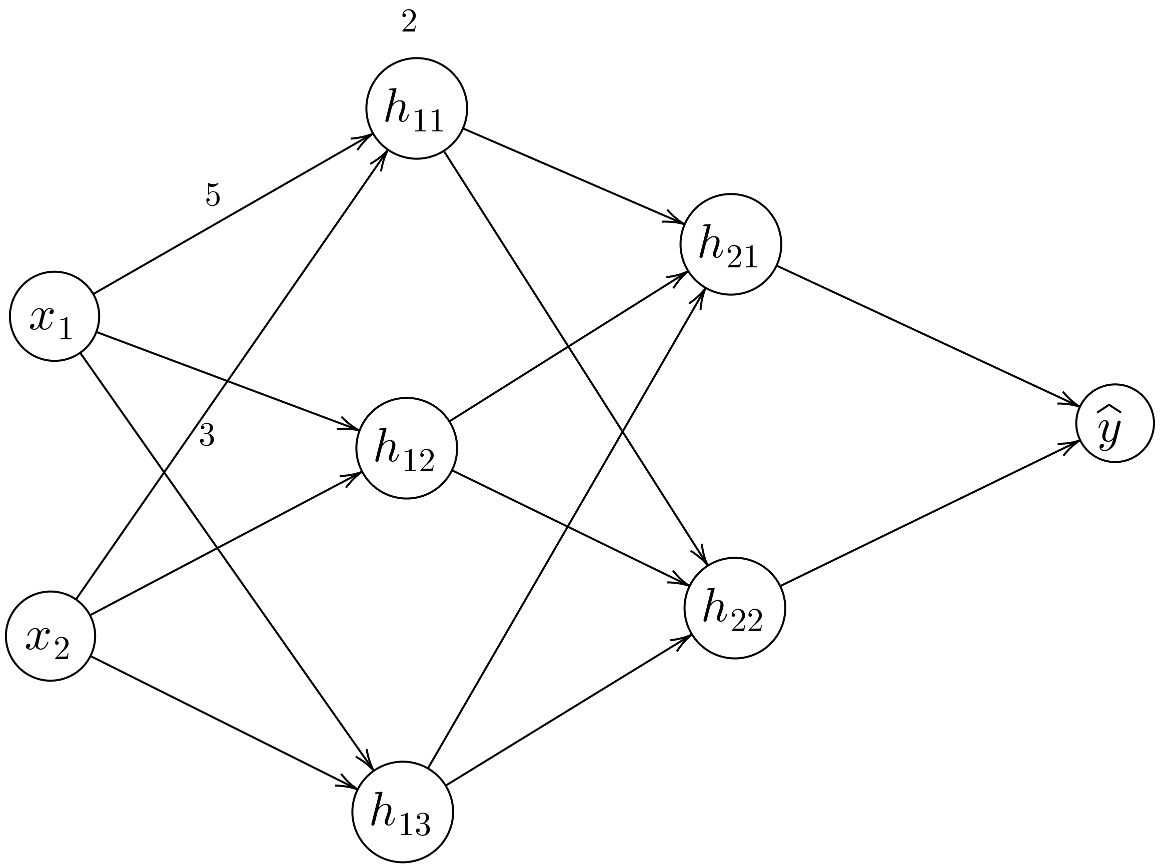
$$\texttt{ReLU}(z) = \max(0, z)$$

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

Forward Pass

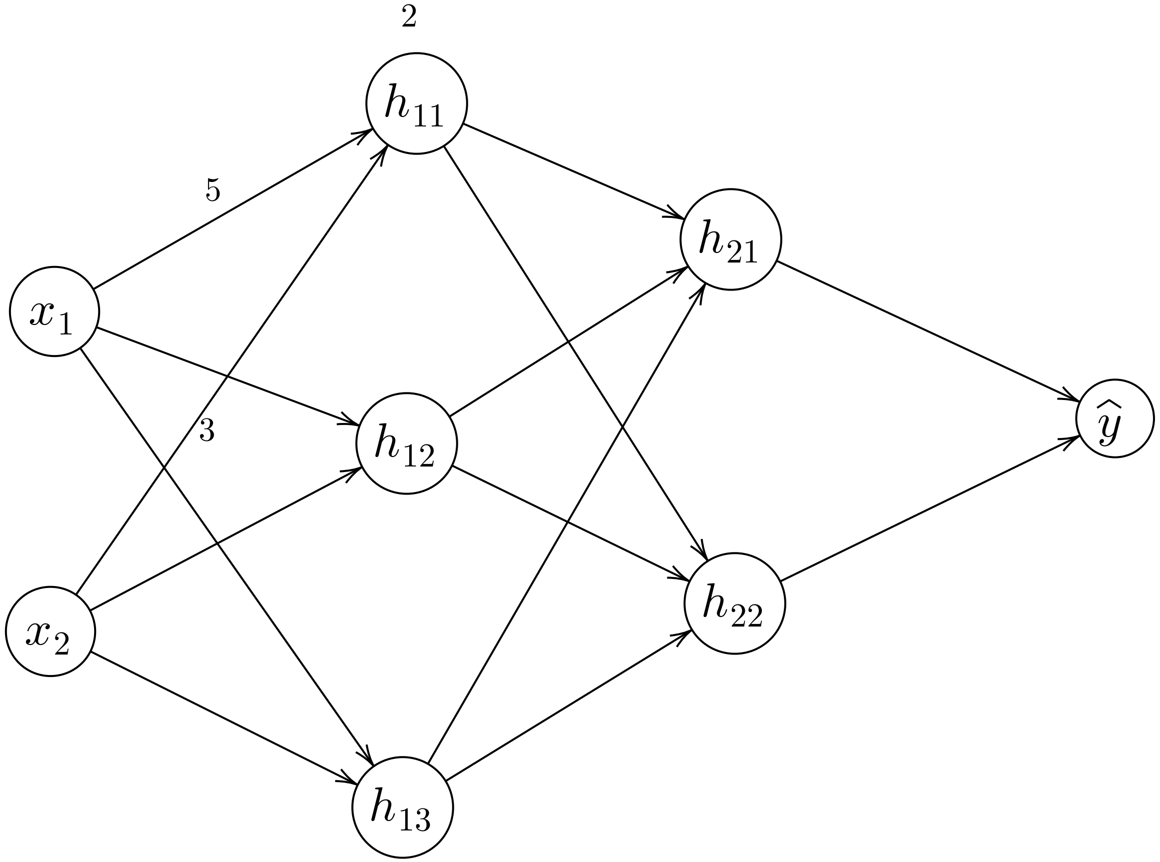


Forward Pass



$$h_{11} = g(5x_1 + 3x_2 + 2)$$

Forward Pass

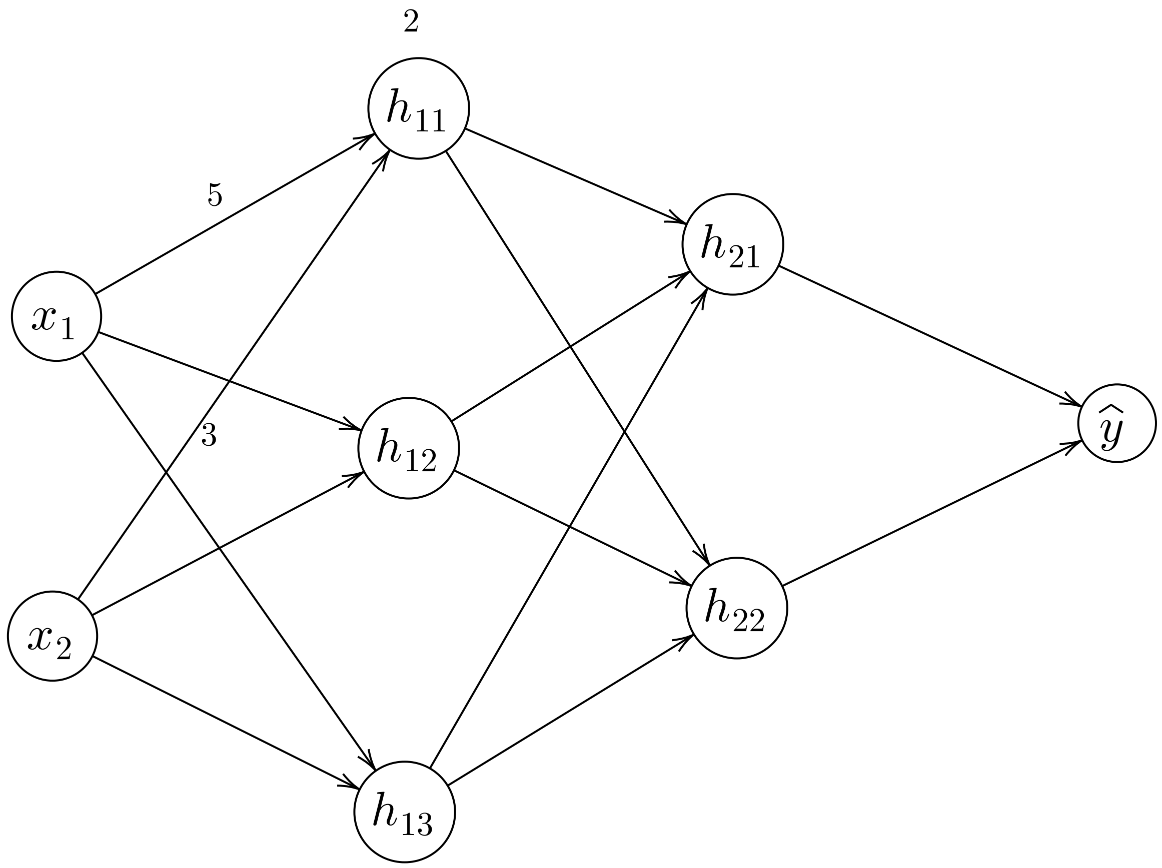


$$h_{11} = g(5x_1 + 3x_2 + 2)$$

$$\Big\downarrow \text{ReLU}$$

$$h_{11} = \max(5x_1 + 3x_2 + 2, 0)$$

Forward Pass



$$h_{11} = g(5x_1 + 3x_2 + 2)$$

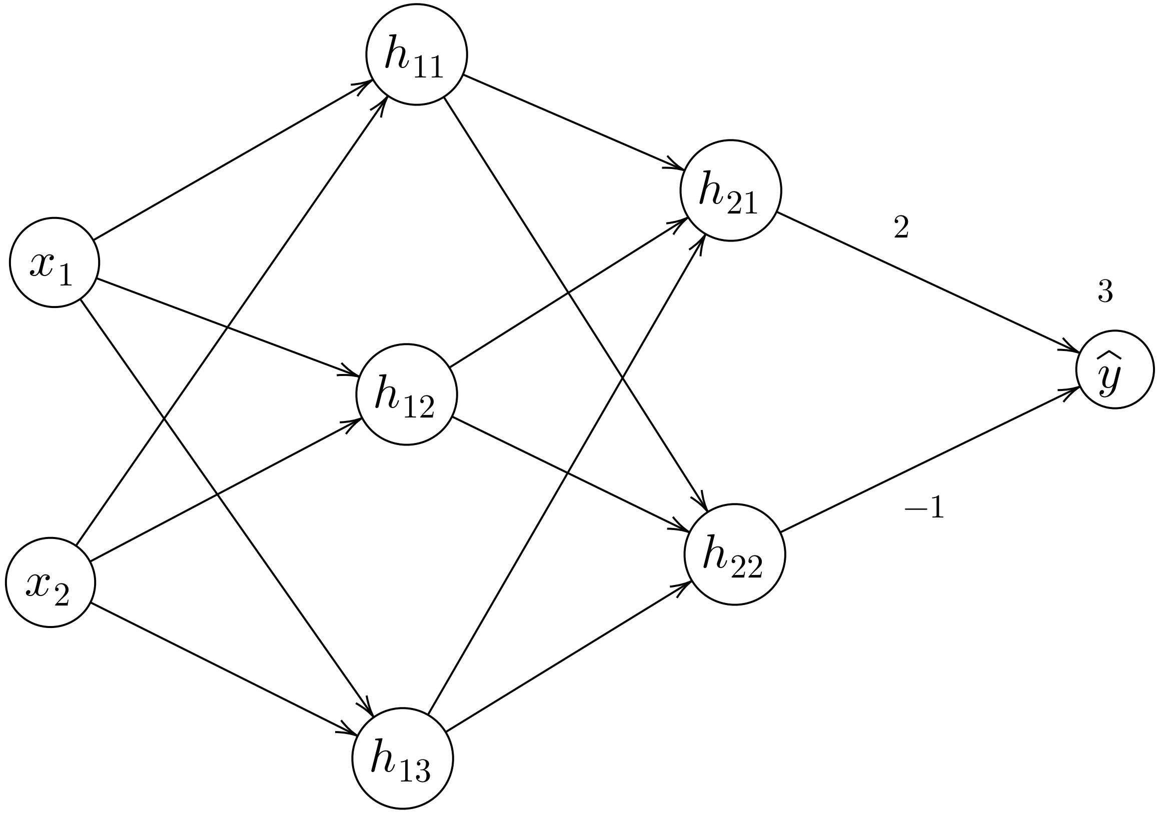
ReLU

$$h_{11} = \max(5x_1 + 3x_2 + 2, 0)$$

Sigmoid

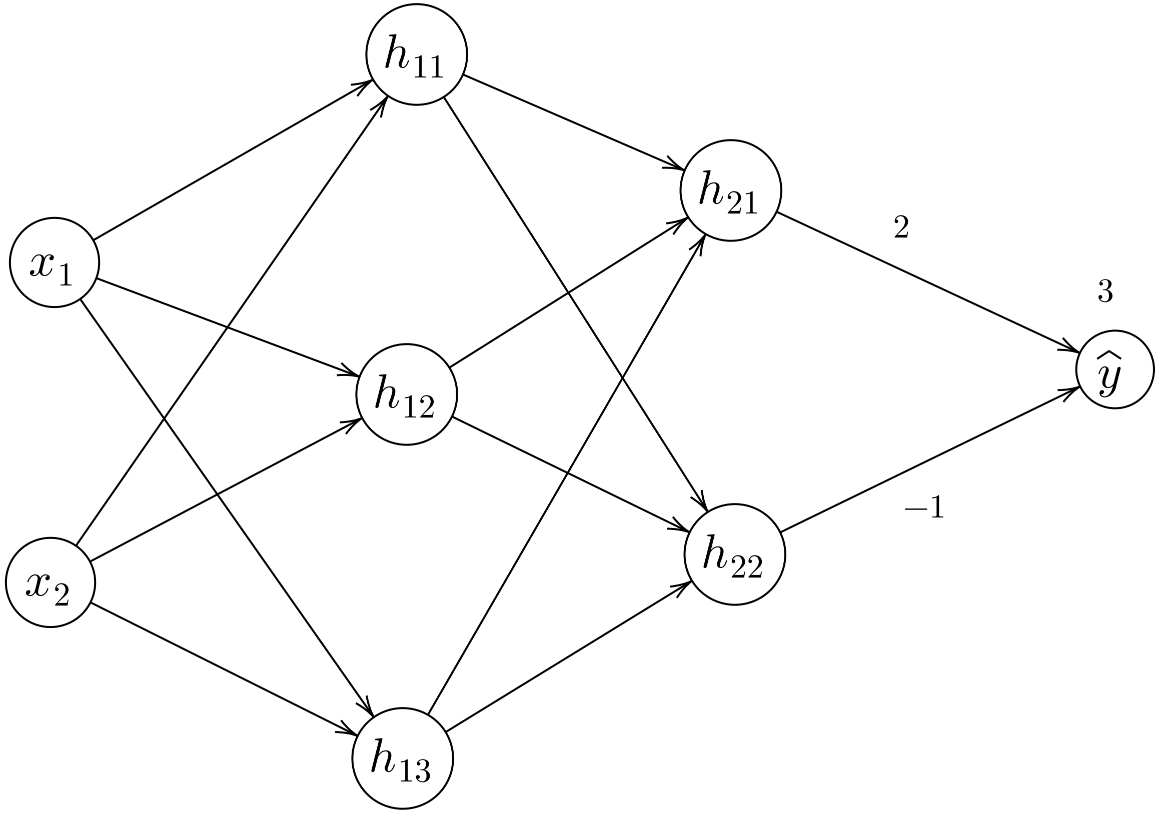
$$h_{11} = \sigma(5x_1 + 3x_2 + 2)$$

Forward Pass



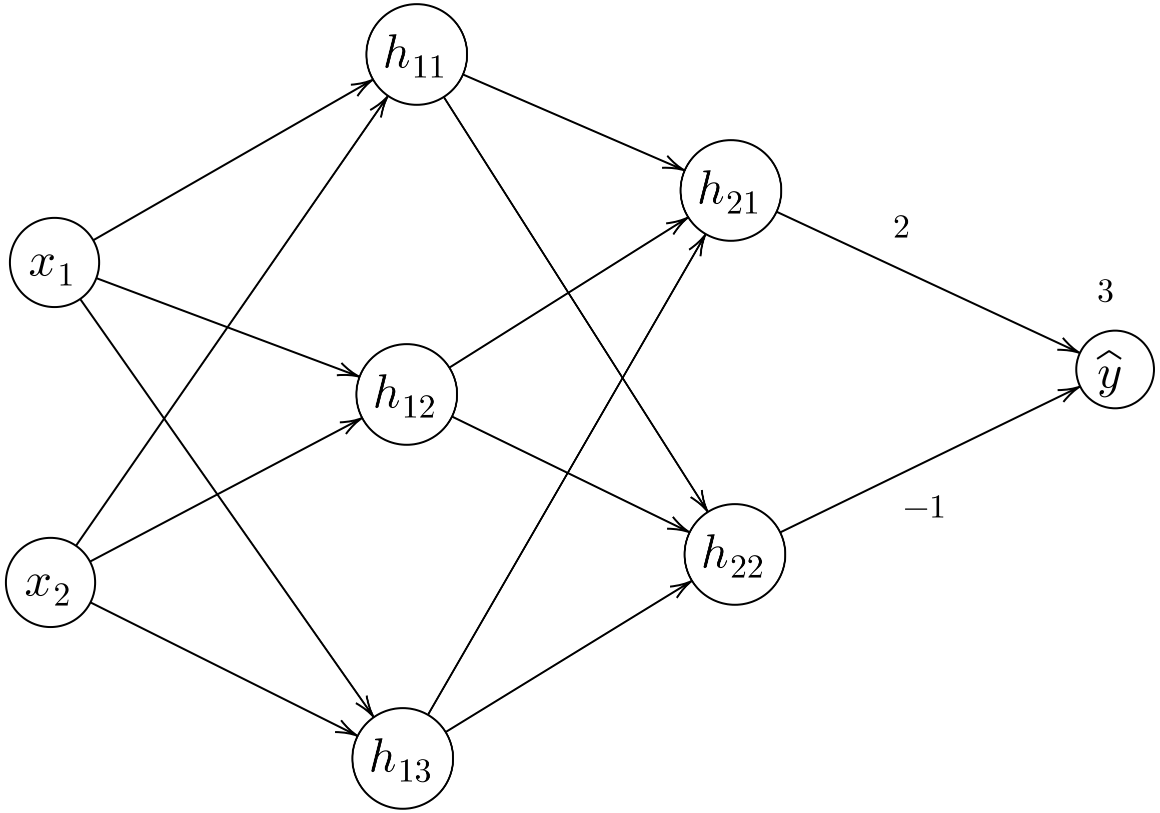
Forward Pass

$$\widehat{y} = g(2h_{21} - h_{22} + 3)$$



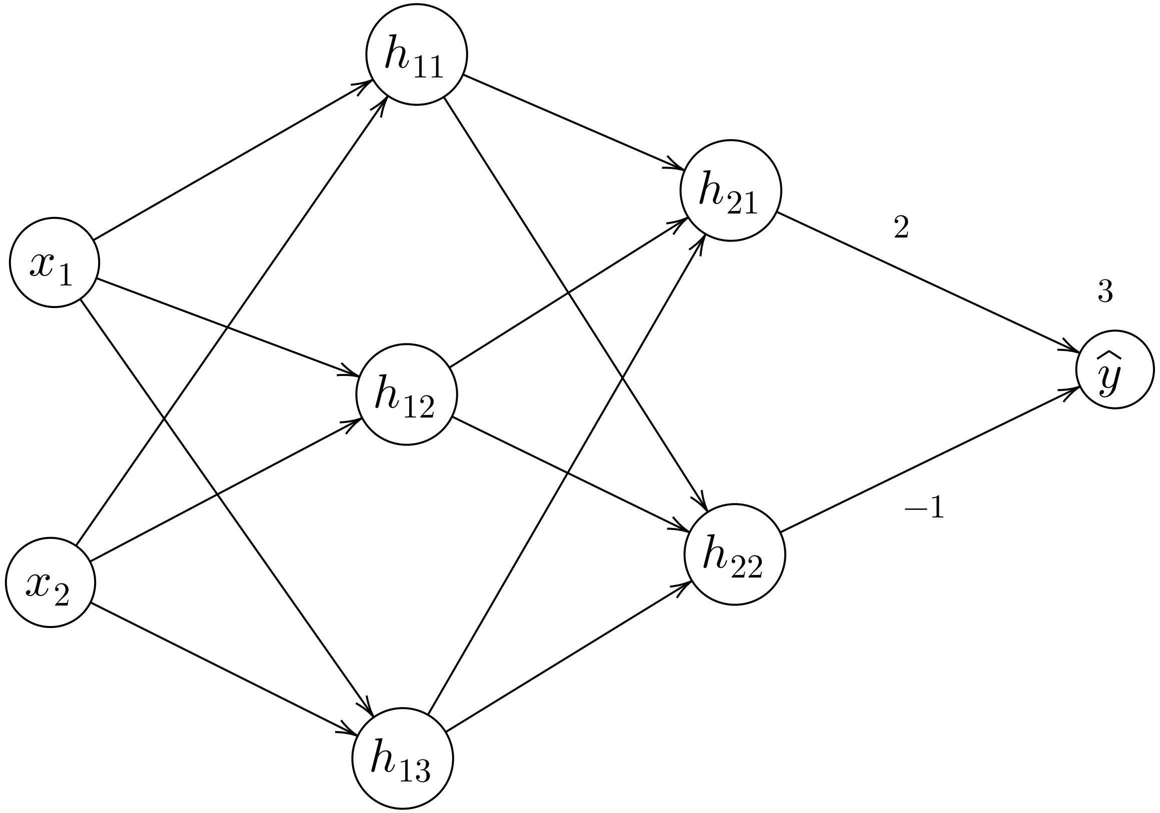
Forward Pass

$$\widehat{y} = 2h_{21} - h_{22} + 3 \xleftarrow{\text{Linear}} \widehat{y} = g(2h_{21} - h_{22} + 3)$$

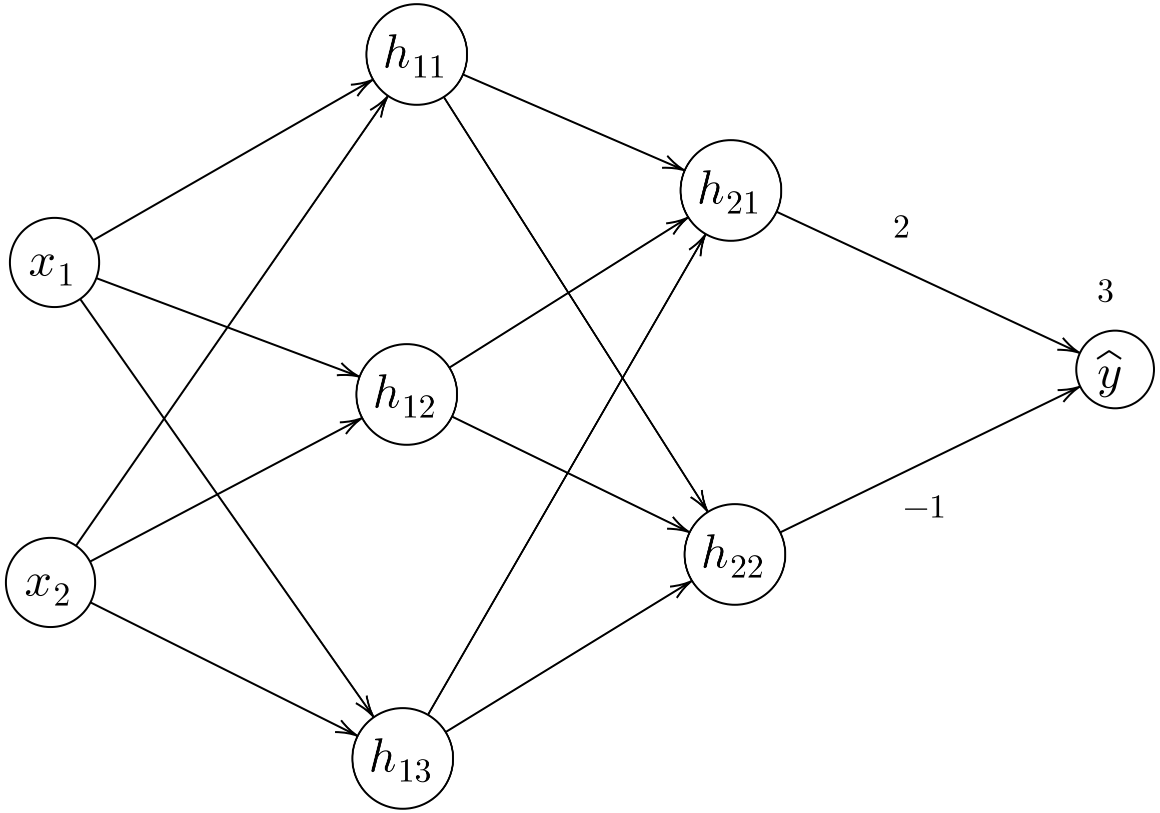


Forward Pass

$$\begin{tikzcd} \widehat{y} = 2h_{21} - h_{22} + 3 & \widehat{y} = g(2h_{21} - h_{22} + 3) \arrow[l, "\text{Linear}"'] \arrow[d, "\text{Sigmoid}"'] \\ & \widehat{y} = \sigma(2h_{21} - h_{22} + 3) \end{tikzcd}$$



Forward Pass



$\widehat{y} = g(2h_{21} - h_{22} + 3)$

Linear

$\widehat{y} = 2h_{21} - h_{22} + 3$

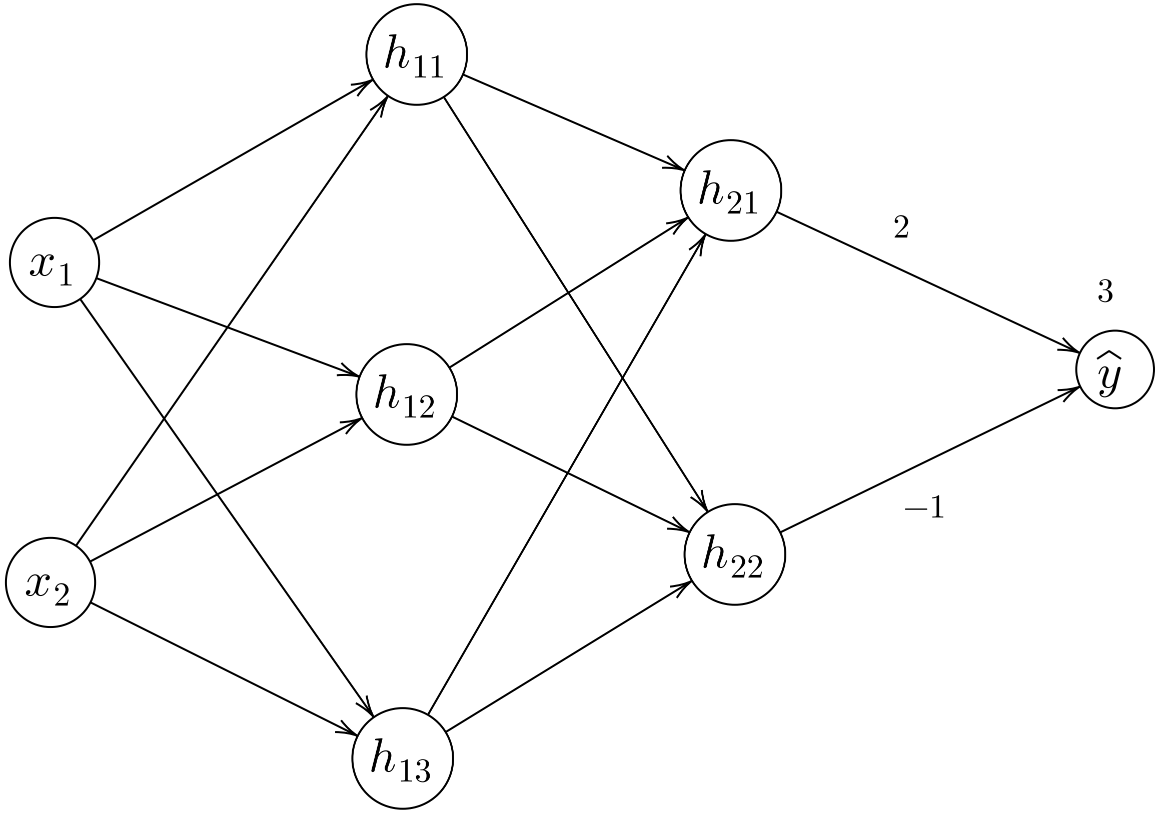
Sigmoid

$\widehat{y} = \sigma(2h_{21} - h_{22} + 3)$

Regression

$\widehat{y} = h(\mathbf{x})$

Forward Pass



$$\widehat{y} = 2h_{21} - h_{22} + 3 \xleftarrow{\text{Linear}} \widehat{y} = g(2h_{21} - h_{22} + 3)$$

$$\widehat{y} = g(2h_{21} - h_{22} + 3) \xrightarrow{\text{Sigmoid}} \widehat{y} = \sigma(2h_{21} - h_{22} + 3)$$

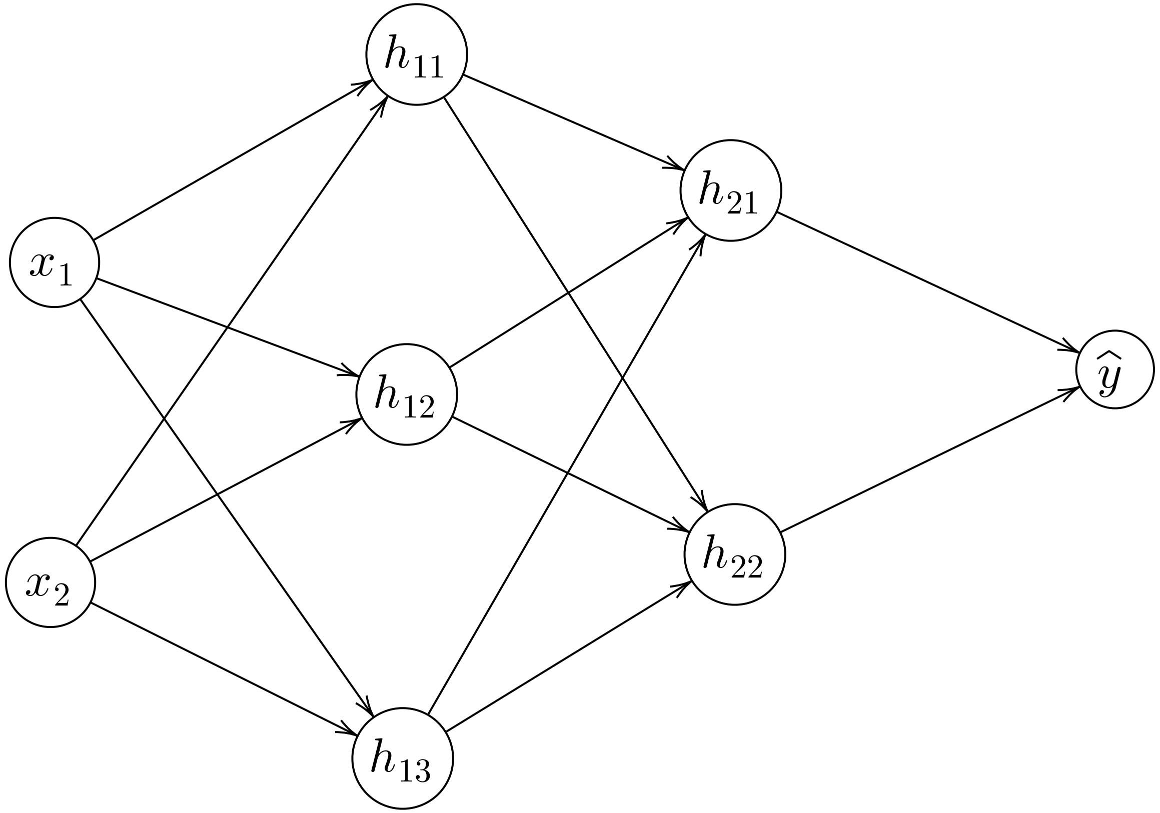
Regression

$$\widehat{y} = h(\mathbf{x})$$

Binary Classification

$$\widehat{y} = h(\mathbf{x}) = P(y = 1 \mid \mathbf{x})$$

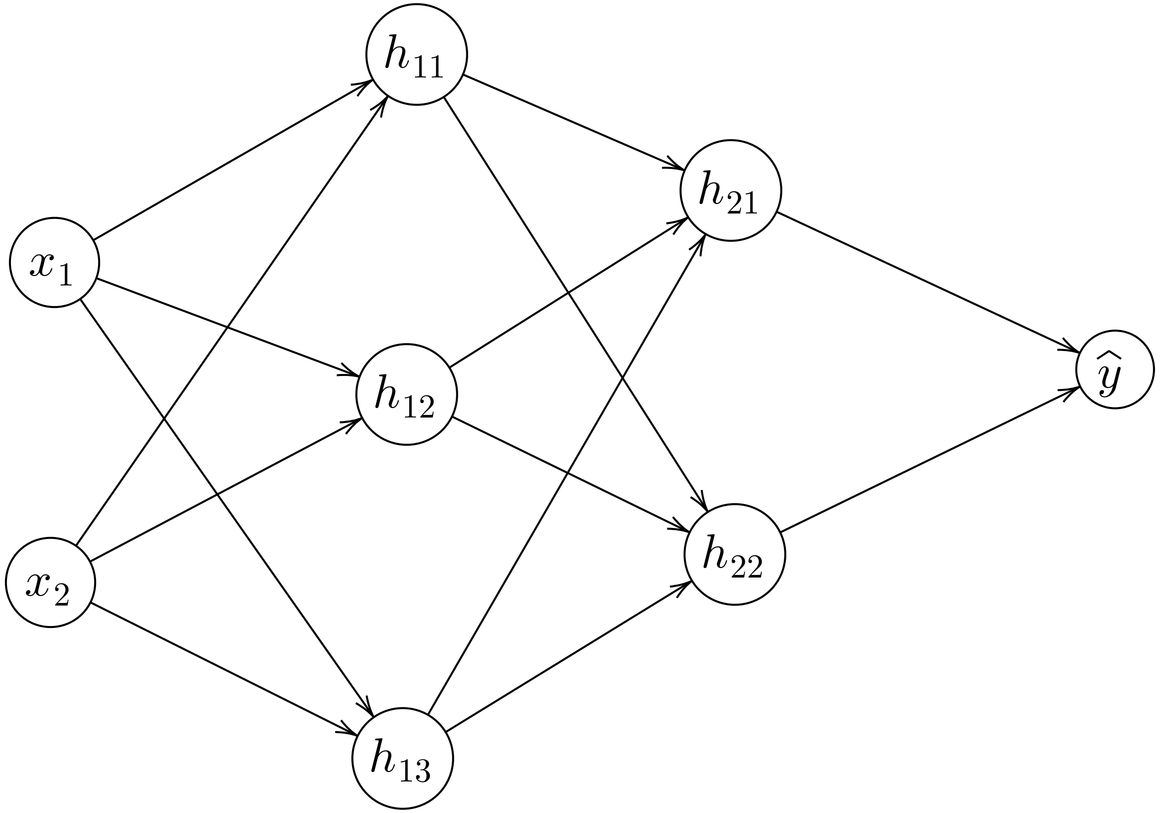
Forward Pass



Regression

$$L(y, \widehat{y}) = (y - \widehat{y})^2$$

Forward Pass



Regression

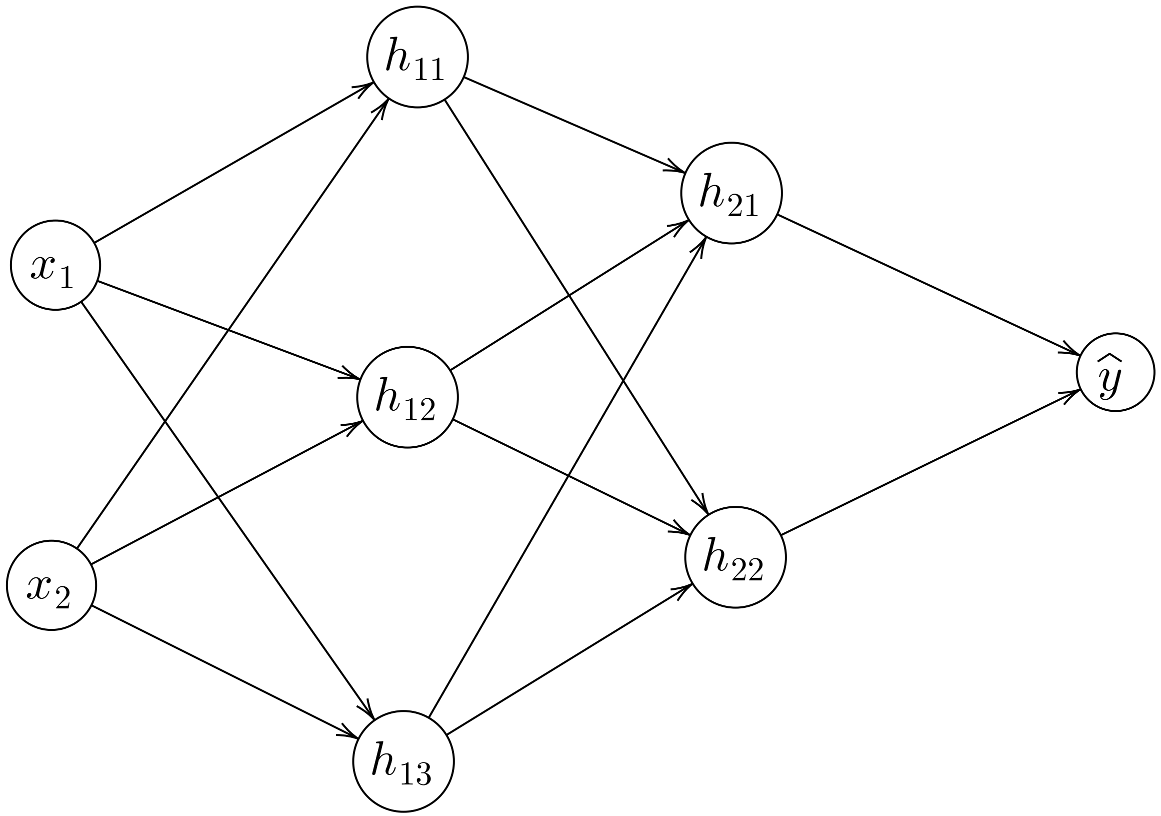
$$L(y, \widehat{y}) = (y - \widehat{y})^2$$

Binary Classification

$$L(y, \widehat{y}) = -y \log \widehat{y} - (1-y)\log(1-\widehat{y})$$

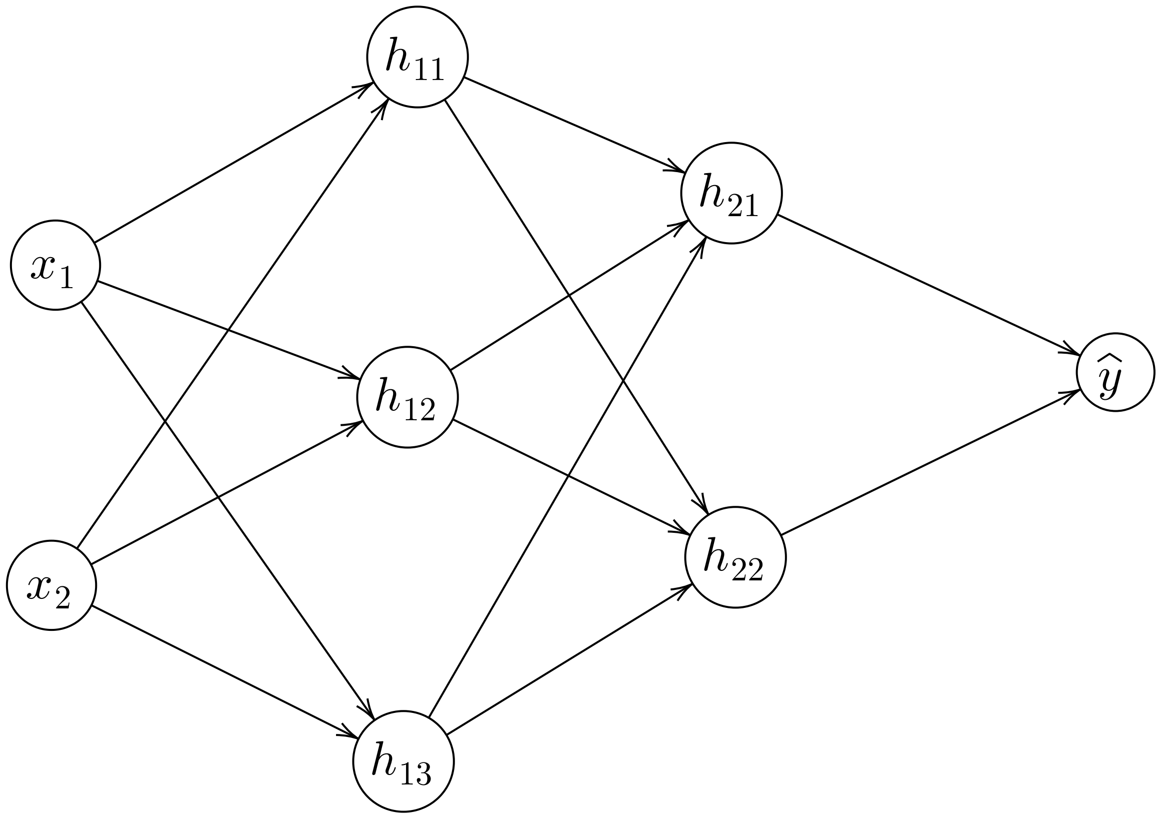
Binary Cross Entropy Loss

Backward Pass



$$\boldsymbol{\theta} = \left[w_{11}^1,\ \cdots, w_{11}^2,\ \cdots, w_{11}^3,\ \cdots, b_1^1,\ \cdots, b_1^3\right]^T$$

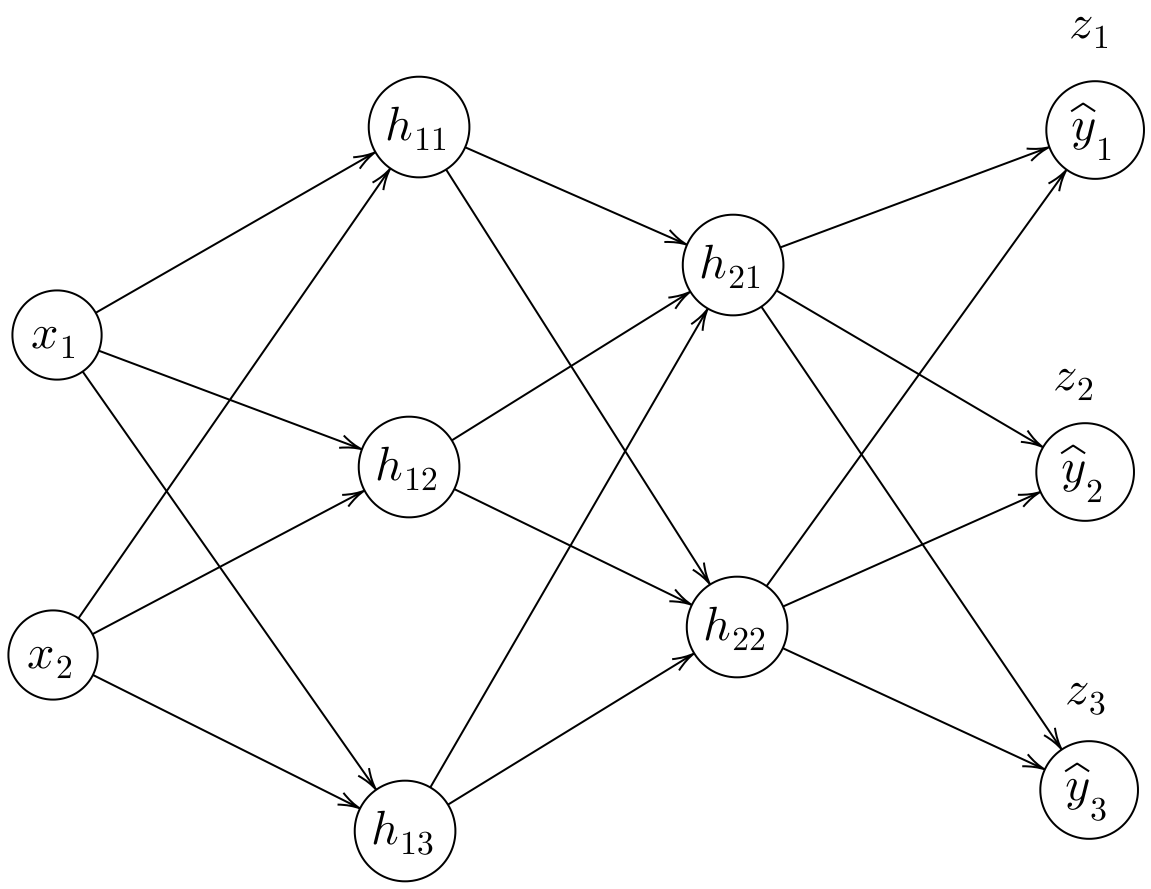
Backward Pass



$$\boldsymbol{\theta} = \left[w_{11}^1,\ \cdots, w_{11}^2,\ \cdots, w_{11}^3,\ \cdots, b_1^1,\ \cdots, b_1^3 \right]^T$$

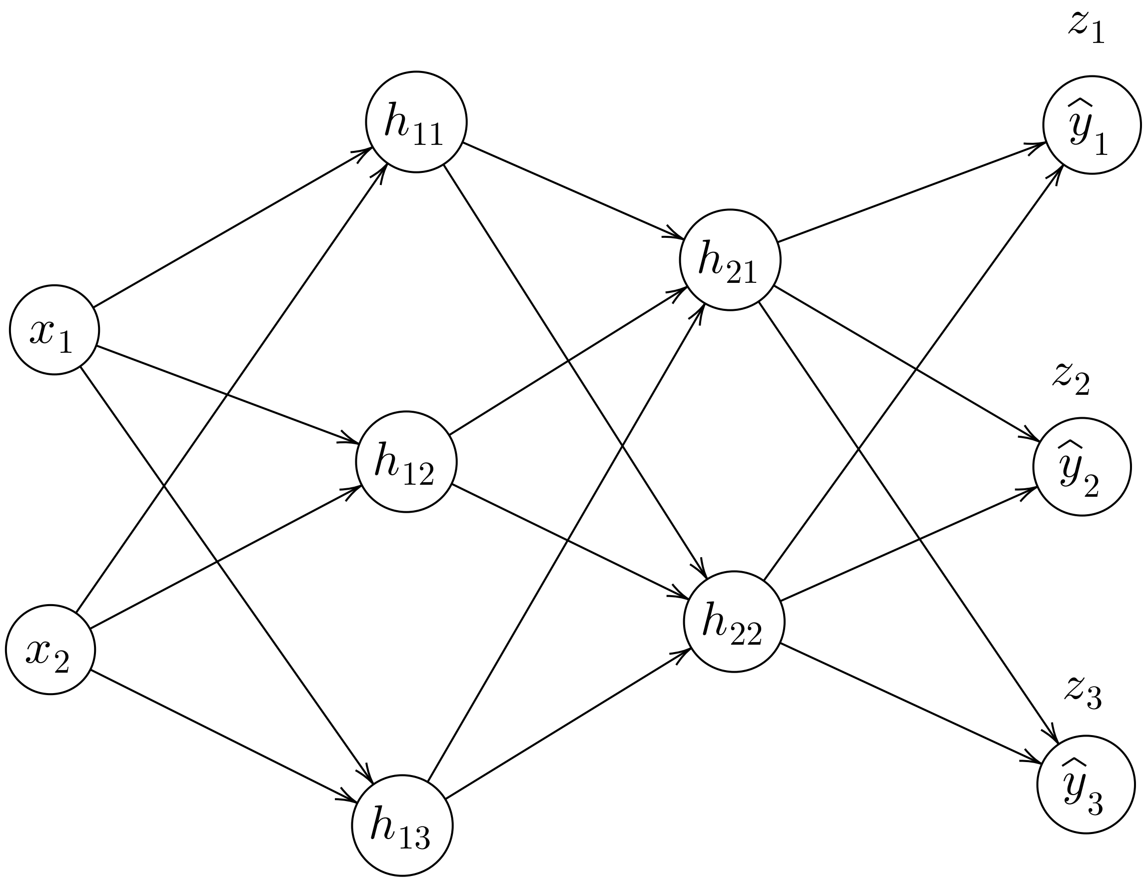
$$\boldsymbol{\theta}^{t+1} = \boldsymbol{\theta}^t - \alpha \cdot \underbrace{\nabla \left[\sum_{i=1}^{n} L(y_i, \widehat{y}_i) \right]}_{\texttt{Backpropagation}}$$

Multi-Class Classification



Multi-Class Classification

Softmax ACtivation



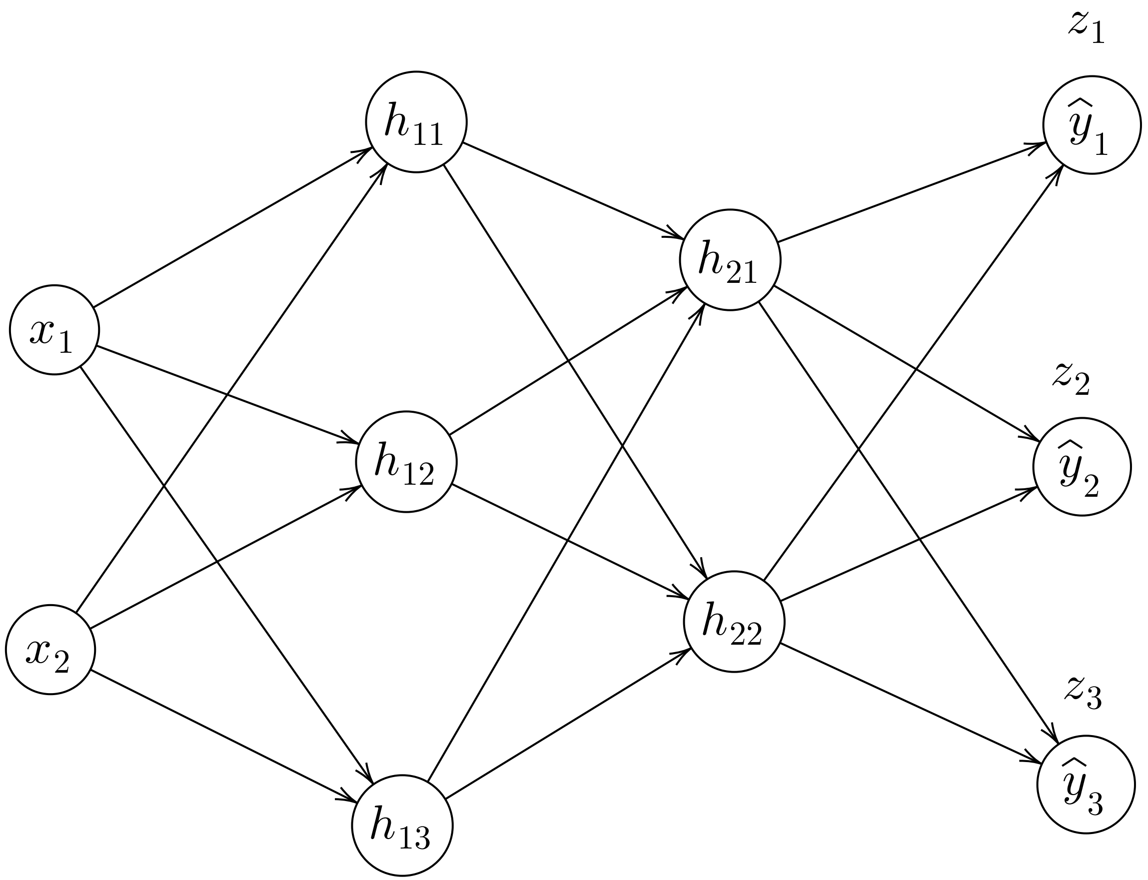
$$\hat{y}_1 = \frac{e^{z_1}}{e^{z_1} + e^{z_2} + e^{z_3}}$$

$$\hat{y}_2 = \frac{e^{z_2}}{e^{z_1} + e^{z_2} + e^{z_3}}$$

$$\hat{y}_3 = \frac{e^{z_2}}{e^{z_1} + e^{z_2} + e^{z_3}}$$

Multi-Class Classification

Softmax Activation



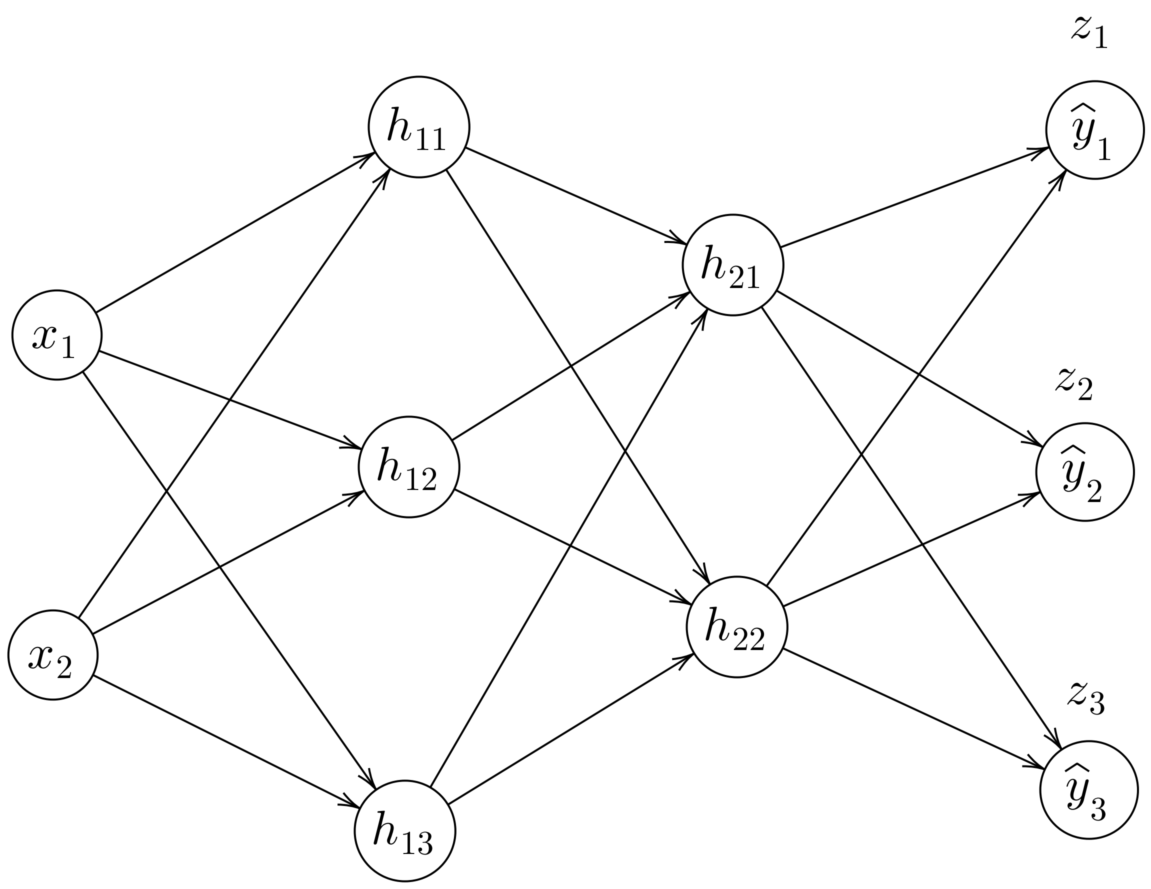
$$\widehat{y}_1 = \frac{e^{z_1}}{e^{z_1} + e^{z_2} + e^{z_3}} = P\big(\mathbf{y} = [1 \;\; 0 \;\; 0]^T \mid \mathbf{x}\big)$$

$$\widehat{y}_2 = \frac{e^{z_2}}{e^{z_1} + e^{z_2} + e^{z_3}} = P\big(\mathbf{y} = [0 \;\; 1 \;\; 0]^T \mid \mathbf{x}\big)$$

$$\widehat{y}_3 = \frac{e^{z_2}}{e^{z_1} + e^{z_2} + e^{z_3}} = P\big(\mathbf{y} = [0 \;\; 0 \;\; 1]^T \mid \mathbf{x}\big)$$

Multi-Class Classification

Softmax Activation



one-hot-encoding

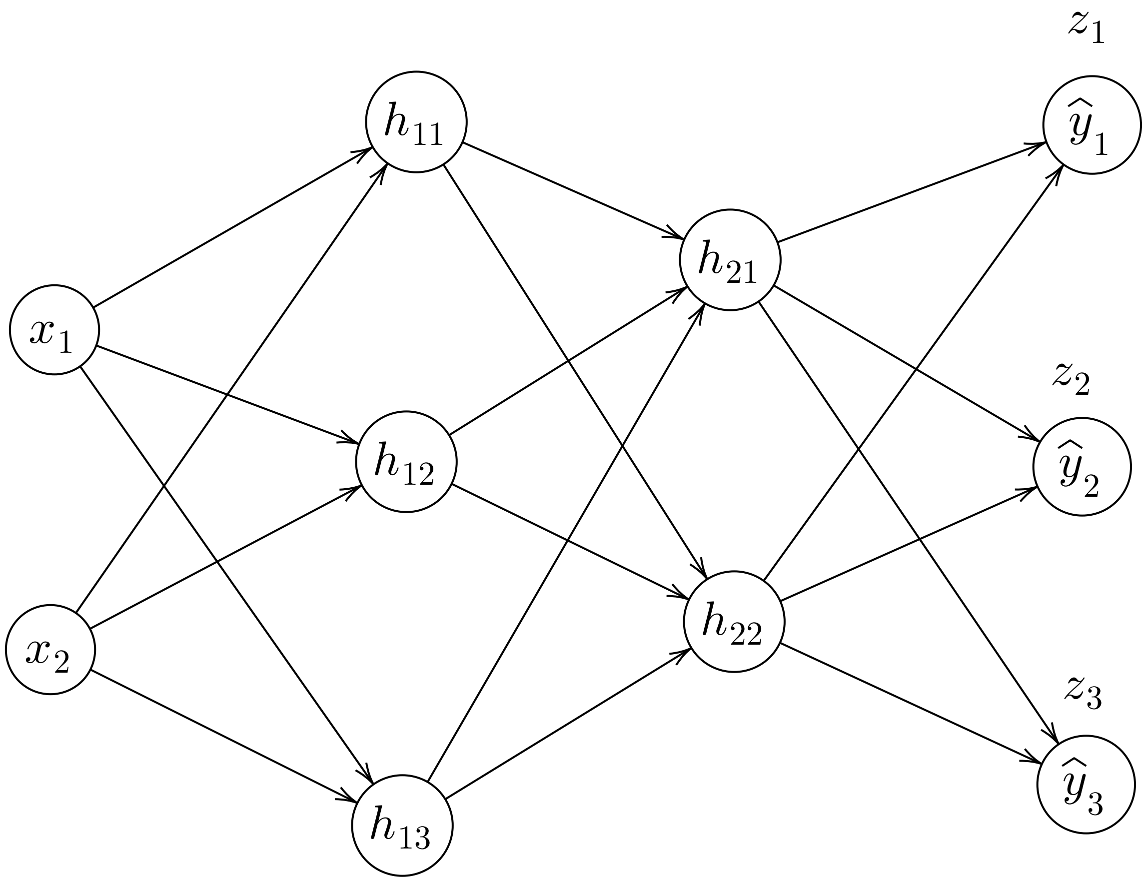
$$\widehat{y}_1 = \frac{e^{z_1}}{e^{z_1} + e^{z_2} + e^{z_3}} = P\big(\mathbf{y} = [1 \;\; 0 \;\; 0]^T \mid \mathbf{x}\big)$$

$$\widehat{y}_2 = \frac{e^{z_2}}{e^{z_1} + e^{z_2} + e^{z_3}} = P\big(\mathbf{y} = [0 \;\; 1 \;\; 0]^T \mid \mathbf{x}\big)$$

$$\widehat{y}_3 = \frac{e^{z_2}}{e^{z_1} + e^{z_2} + e^{z_3}} = P\big(\mathbf{y} = [0 \;\; 0 \;\; 1]^T \mid \mathbf{x}\big)$$

Multi-Class Classification

Softmax Activation



one-hot-encoding

$$\widehat{y}_1 = \frac{e^{z_1}}{e^{z_1} + e^{z_2} + e^{z_3}} = P\big(\mathbf{y} = [1 \;\; 0 \;\; 0]^T \mid \mathbf{x}\big)$$

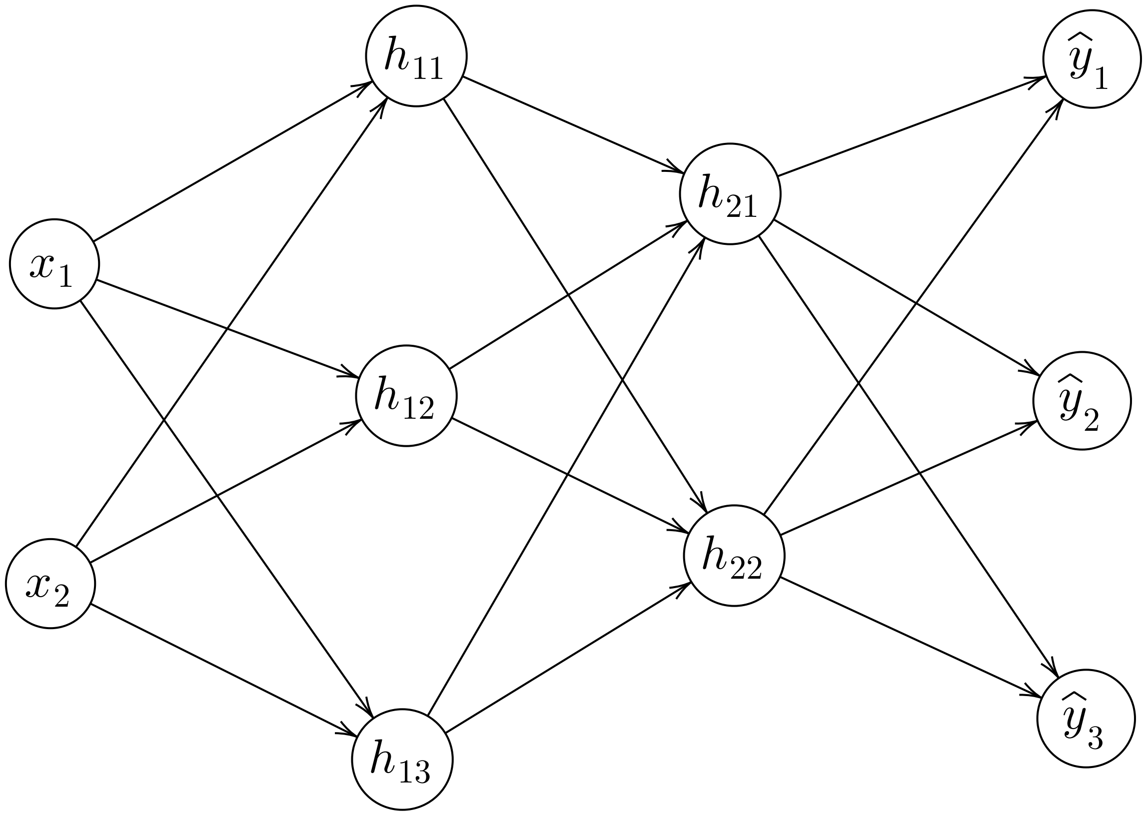
$$\widehat{y}_2 = \frac{e^{z_2}}{e^{z_1} + e^{z_2} + e^{z_3}} = P\big(\mathbf{y} = [0 \;\; 1 \;\; 0]^T \mid \mathbf{x}\big)$$

$$\widehat{y}_3 = \frac{e^{z_2}}{e^{z_1} + e^{z_2} + e^{z_3}} = P\big(\mathbf{y} = [0 \;\; 0 \;\; 1]^T \mid \mathbf{x}\big)$$

Categorical Distribution
(tossing a 3-sided dice)

Multi-Class Classification

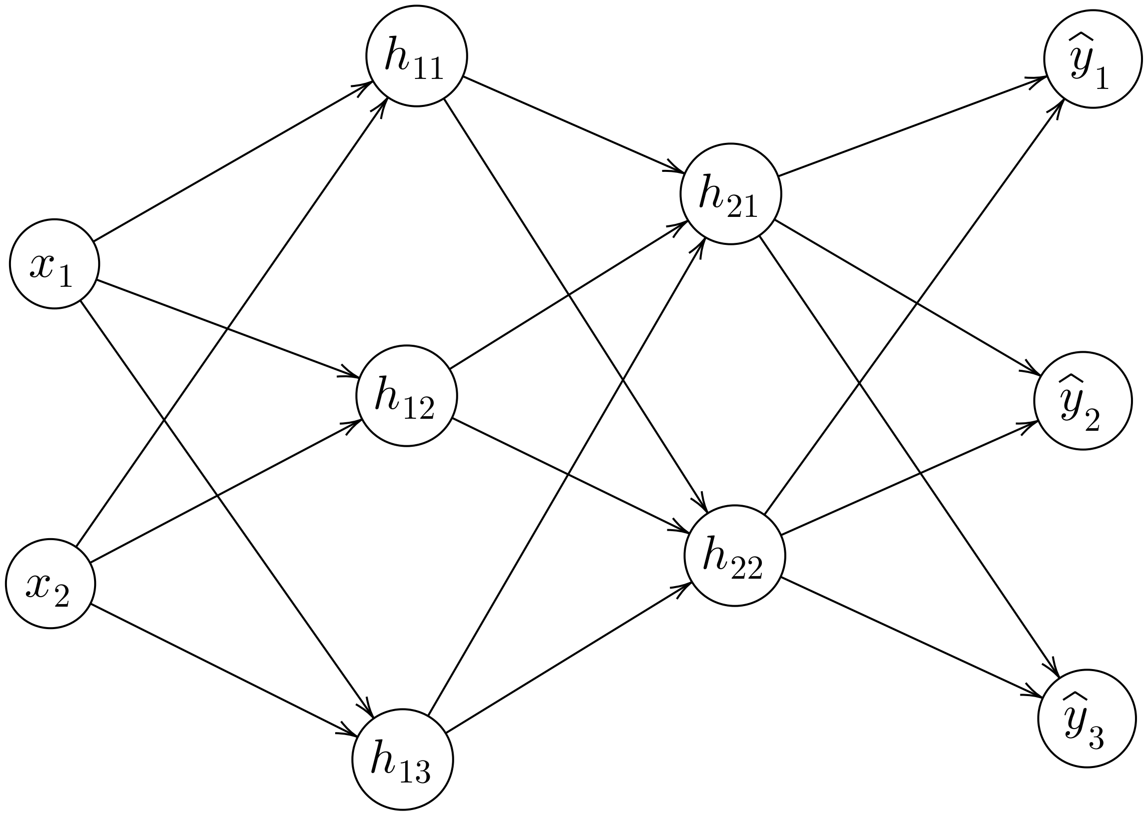
Categorical Cross Entropy



$$L(\widehat{\mathbf{y}}, \mathbf{y}) = -y_1 \log_2(\widehat{y}_1) - y_2 \log_2(\widehat{y}_2) - y_3 \log(\widehat{y}_3)$$

Multi-Class Classification

Categorical Cross Entropy

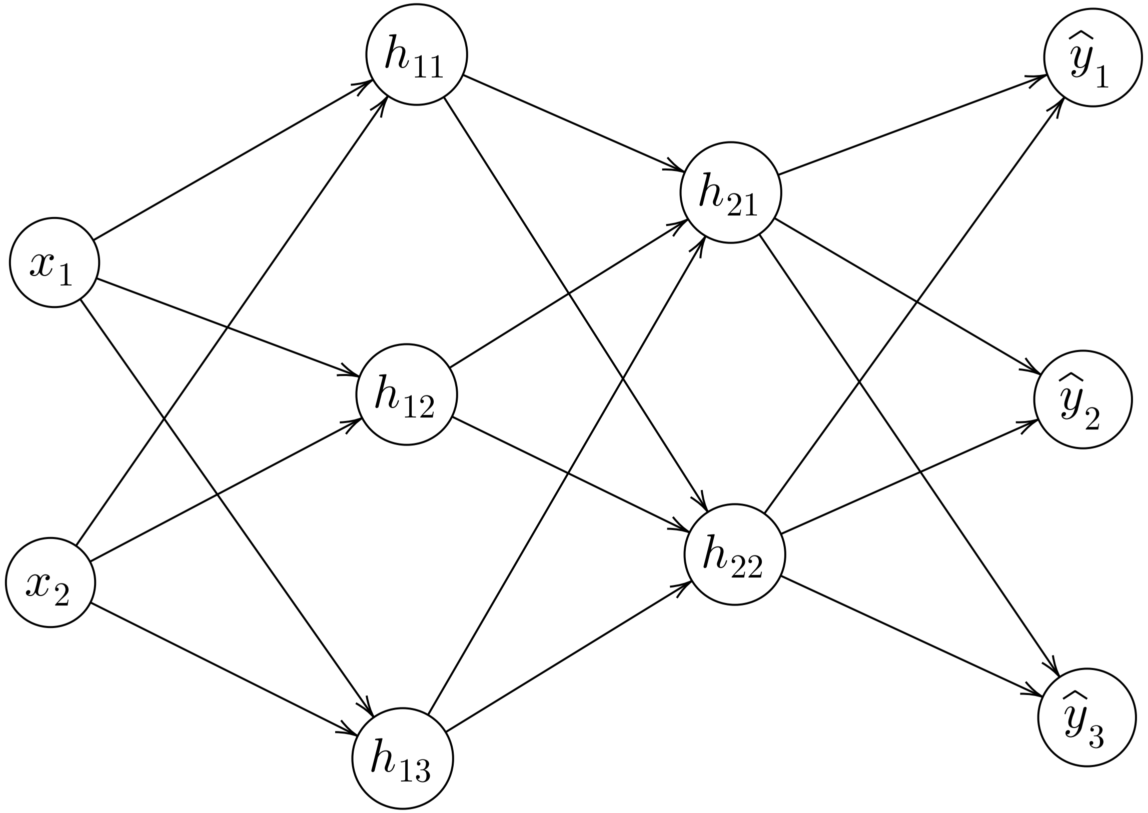


$$L(\widehat{\mathbf{y}}, \mathbf{y}) = -y_1 \log_2(\widehat{y}_1) - y_2 \log_2(\widehat{y}_2) - y_3 \log(\widehat{y}_3)$$

$$\mathbf{y} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \qquad \widehat{\mathbf{y}} = \begin{bmatrix} 0.2 \\ 0.3 \\ 0.5 \end{bmatrix}$$

Multi-Class Classification

Categorical Cross Entropy



$$L(\widehat{\mathbf{y}}, \mathbf{y}) = -y_1 \log_2(\widehat{y}_1) - y_2 \log_2(\widehat{y}_2) - y_3 \log(\widehat{y}_3)$$

$$\mathbf{y} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \qquad \widehat{\mathbf{y}} = \begin{bmatrix} 0.2 \\ 0.3 \\ 0.5 \end{bmatrix}$$

$$\begin{aligned} L(\widehat{\mathbf{y}}, \mathbf{y}) &= -0 \times \log_2(0.2) - 1 \times \log_2(0.3) - 0 \log(0.5) \\ &= -\log_2(0.3) \\ &= 1.73 \end{aligned}$$

Data, Algorithm, Computation

Data, Algorithm, Computation

Large (Labelled) Datasets

ImageNet
14 million images

Data, Algorithm, Computation

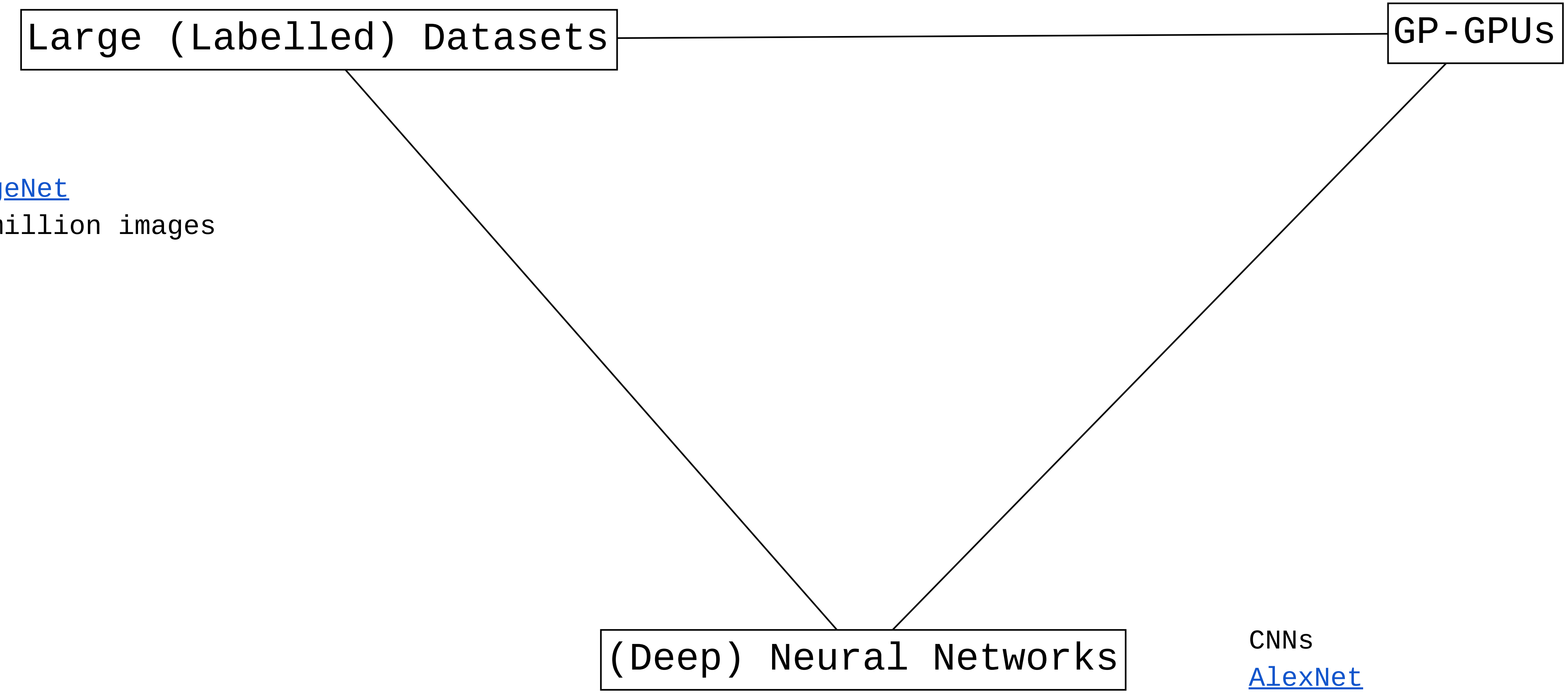
Large (Labelled) Datasets

ImageNet
14 million images

(Deep) Neural Networks

CNNs
AlexNet

Data, Algorithm, Computation



ImageNet
14 million images

CNNs
AlexNet

Model building:

- **Given** a feature matrix → $\mathbf{X}$, train:
  - Linear Regression
  - Logistic Regression
  - SVM
  - Random Forest
  - AdaBoost
- Choose the best model by cross validation

MLT

Model building:

- **Given** a feature matrix → $\mathbf{X}$, train:
  - Linear Regression
  - Logistic Regression
  - SVM
  - Random Forest
  - AdaBoost
- Choose the best model by cross validation

$$\boxed{\mathbf{X}} \longrightarrow \boxed{\texttt{MLT}} \longrightarrow \boxed{h:\ \mathbb{R}^d \to \{-1, 1\}}$$

Machine Learning Lifecycle

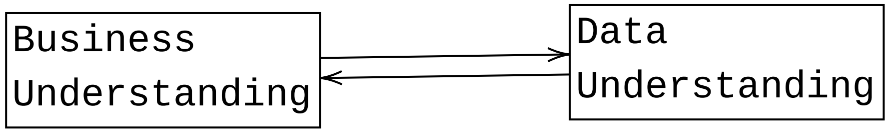
CRISP-DM: **CR**oss-**I**ndustry **S**tandard **P**rocess for **D**ata-**M**ining

Business Understanding

Source: Chapter-2, Trustworthy Machine Learning

Machine Learning Lifecycle

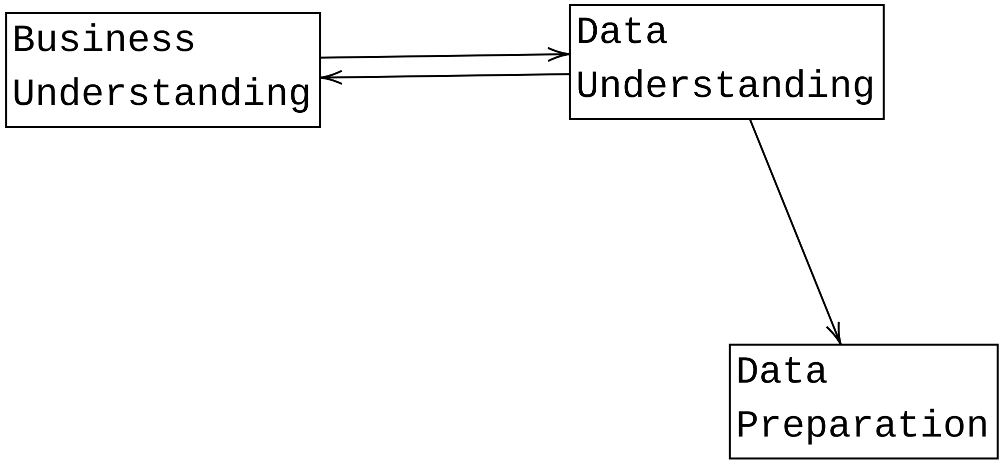
CRISP-DM: **CR**oss-**I**ndustry **S**tandard **P**rocess for **D**ata-**M**ining



Source: Chapter-2, Trustworthy Machine Learning

Machine Learning Lifecycle

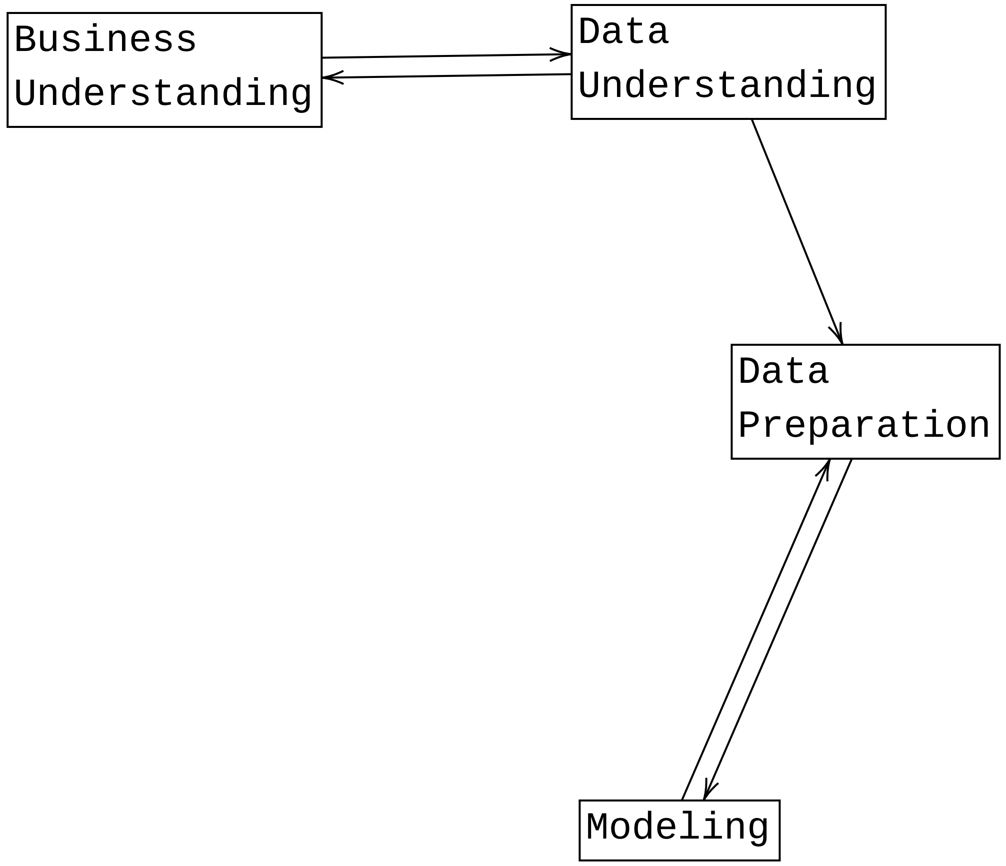
CRISP-DM: **CR**oss-**I**ndustry **S**tandard **P**rocess for **D**ata-**M**ining



Source: Chapter-2, Trustworthy Machine Learning

Machine Learning Lifecycle

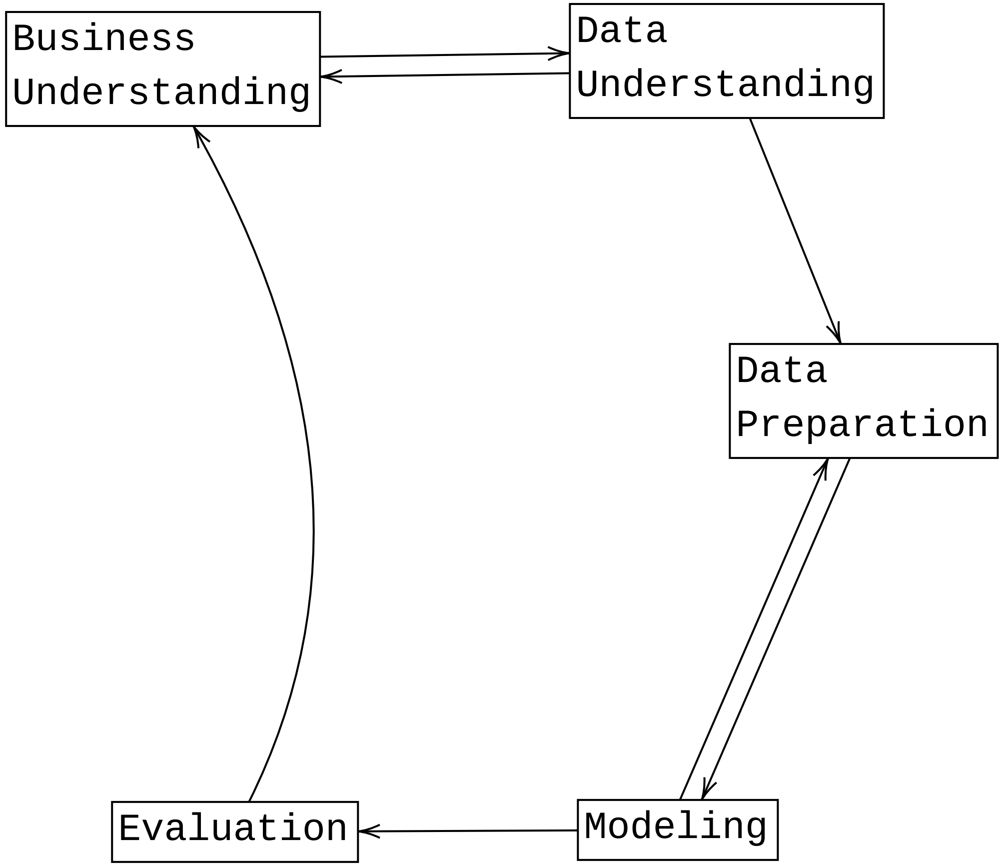
CRISP-DM: **CR**oss-**I**ndustry **S**tandard **P**rocess for **D**ata-**M**ining



Source: Chapter-2, Trustworthy Machine Learning

Machine Learning Lifecycle

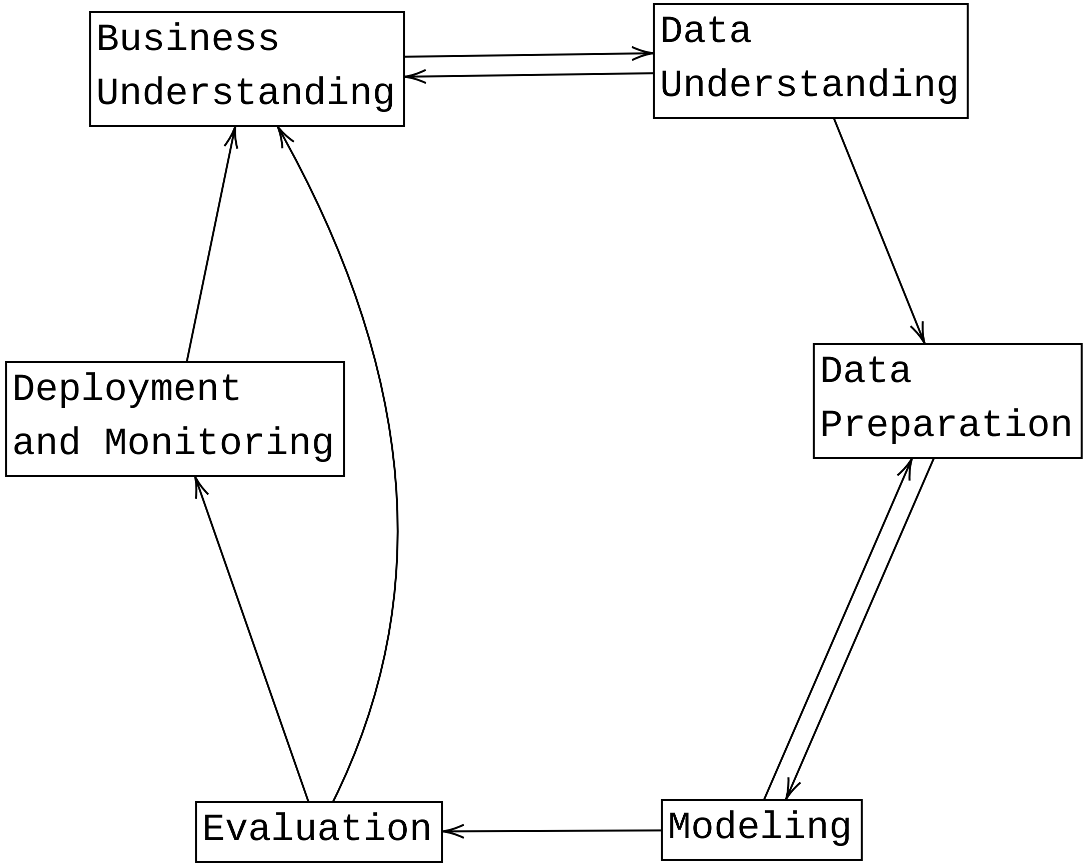
[CRISP-DM]: **CR**oss-**I**ndustry **S**tandard **P**rocess for **D**ata-**M**ining



[Source]: Chapter-2, Trustworthy Machine Learning

Machine Learning Lifecycle

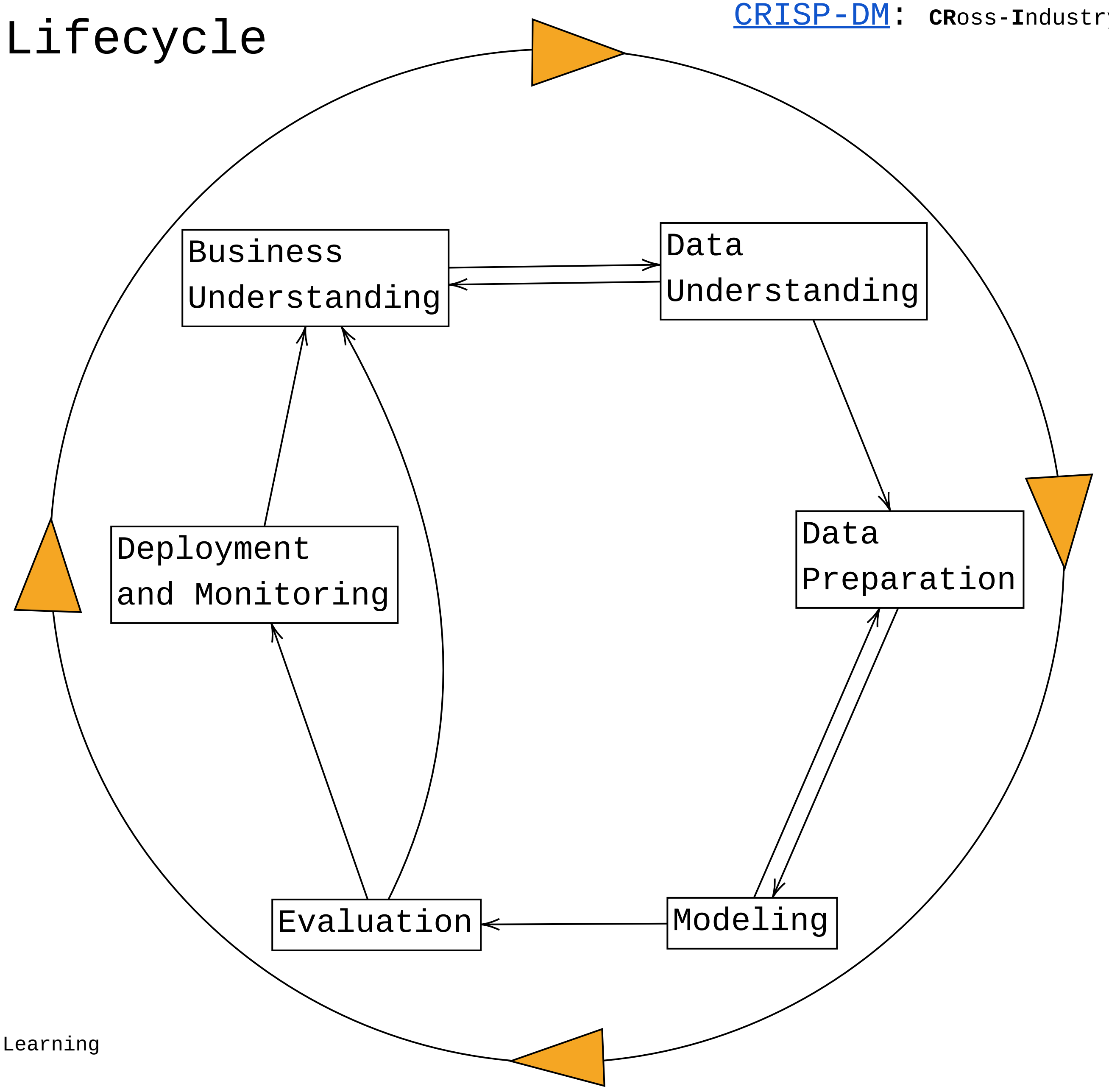
[CRISP-DM]: **CR**oss-**I**ndustry **S**tandard **P**rocess for **D**ata-**M**ining



Source: Chapter-2, Trustworthy Machine Learning

Machine Learning Lifecycle

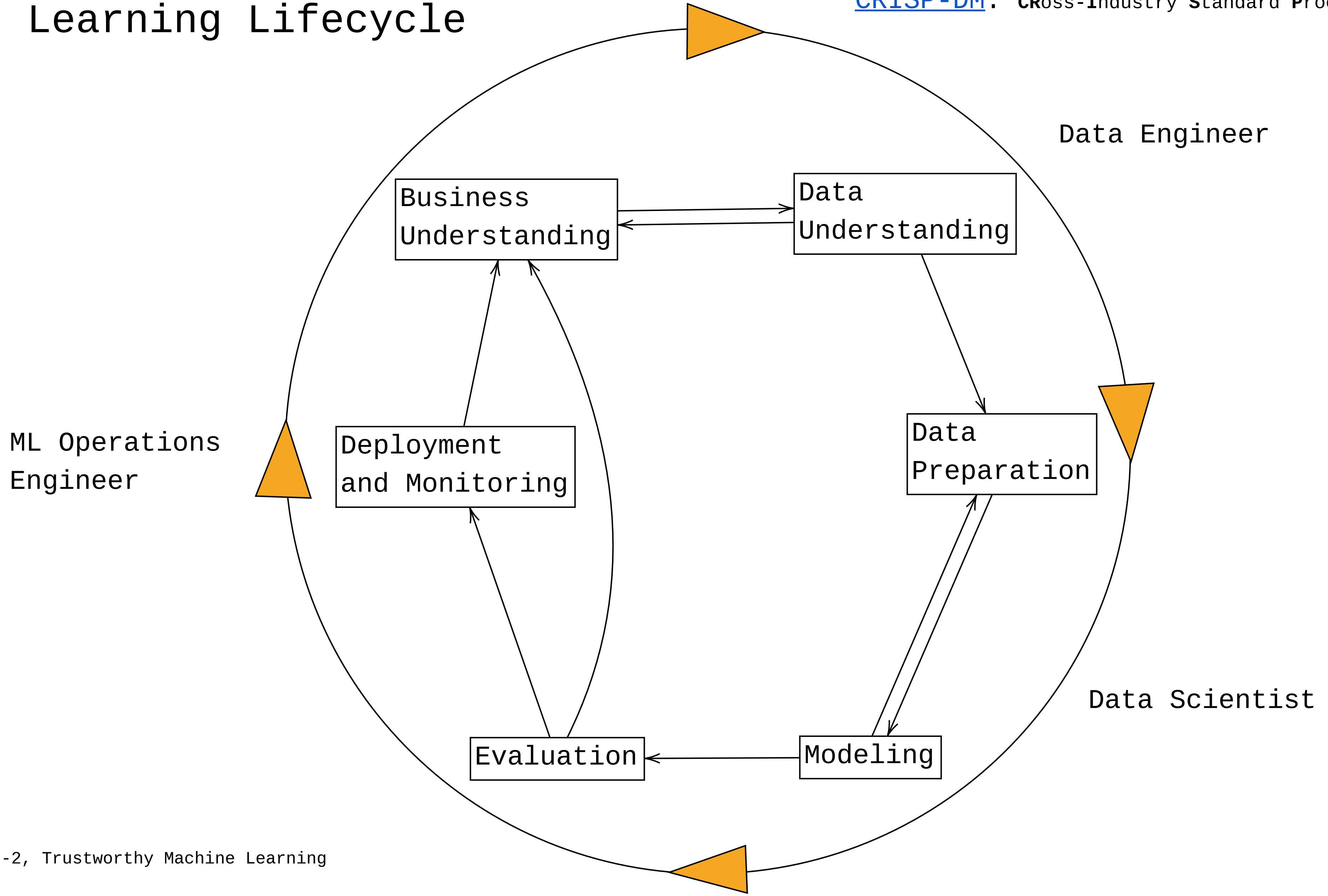
[CRISP-DM]: **CR**oss-**I**ndustry **S**tandard **P**rocess for **D**ata-**M**ining



[Source]: Chapter-2, Trustworthy Machine Learning

Machine Learning Lifecycle

CRISP-DM: **CR**oss-**I**ndustry **S**tandard **P**rocess for **D**ata-**M**ining



Source: Chapter-2, Trustworthy Machine Learning