Machine Learning I
MATH80629A

Apprentissage Automatique I
MATH80629

“Mid-term-ish" summary

- Brief summary of what we have seen so far
- Explain concepts within a single framework
- Focus on a few more advanced concepts

Supervised Machine Learning

Datum $\longrightarrow$ $\boxed{\text{Model}}$ $\longrightarrow$ Prediction

$$\mathbf{x} \longrightarrow \boxed{\mathbf{f}, \theta} \longrightarrow \mathbf{f}(\mathbf{x}; \theta) \equiv \hat{\mathbf{y}}$$

Loss function

$$\mathbf{x} \longrightarrow \boxed{\mathbf{f}, \theta} \longrightarrow \mathbf{f}(\mathbf{x}; \theta) \equiv \hat{\mathbf{y}}$$

Loss function

Loss

$$\mathbf{x} \longrightarrow \boxed{\mathbf{f}, \theta} \longrightarrow \mathbf{f}(\mathbf{x}; \theta) \equiv \hat{\mathbf{y}}$$

Loss function

Loss

$\mathbf{x} \longrightarrow \boxed{\mathbf{f}, \theta} \longrightarrow \mathbf{f}(\mathbf{x}; \theta) \equiv \hat{\mathbf{y}}$

$\mathbf{L}(\hat{\mathbf{y}}, \mathbf{y})$

Loss function

Loss

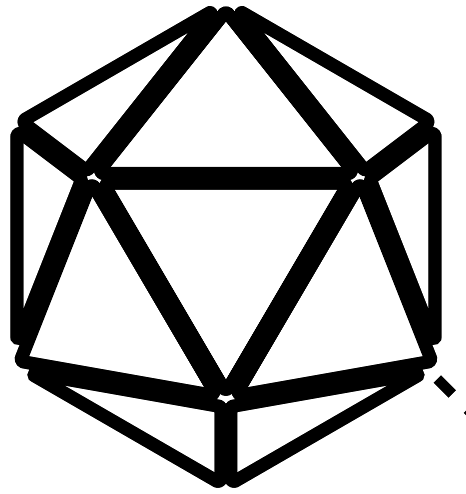
$$\mathbf{x} \longrightarrow \boxed{\mathbf{f}, \theta} \longrightarrow \mathbf{f}(\mathbf{x}; \theta) \equiv \hat{\mathbf{y}} \qquad \mathbf{L}(\hat{\mathbf{y}}, \mathbf{y})$$

Different losses for different types of y's

$\mathbf{y} \in \mathcal{R}$	**Regression**	$(\hat{\mathbf{y}} - \mathbf{y})^2$
y categorical e.g., $\{\text{cat}, \text{dog}, \text{bird}\}$	**Classification**	**accuracy**
$\mathbf{y} \in \{0, 1\}$	**Binary Classification**	**AUC**

Learning Process

Distribution over (x,y): P(x,y)



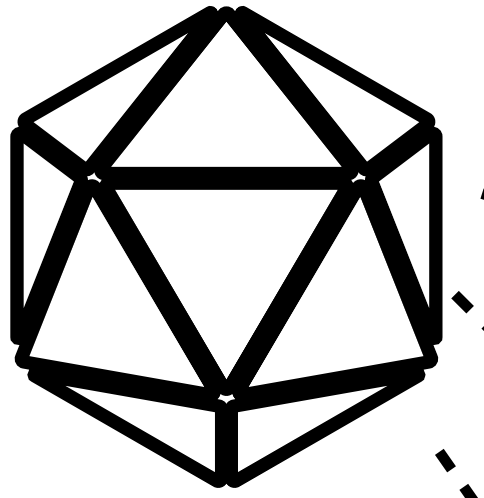
Loss

$\mathbf{x}_{\text{train}} \longrightarrow \boxed{\mathbf{f}, \theta} \longrightarrow \hat{\mathbf{y}}_{\text{train}} \qquad \mathbf{L}(\hat{\mathbf{y}}_{\text{train}}, \mathbf{y}_{\text{train}})$

$\mathbf{x}_{\text{test}} \longrightarrow \boxed{\mathbf{f}, \hat{\theta}} \longrightarrow \hat{\mathbf{y}}_{\text{test}} \qquad \mathbf{L}(\hat{\mathbf{y}}_{\text{test}}, \mathbf{y}_{\text{test}})$

Learning Process
In practice

Distribution over (x,y): P(x,y)



Loss

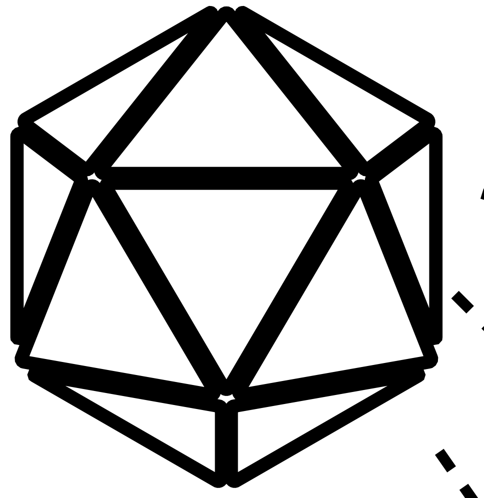
$x_{train} \longrightarrow \boxed{f, \theta} \longrightarrow \hat{y}_{train}$ $\quad L(\hat{y}_{train}, y_{train})$

$x_{valid} \longrightarrow \boxed{f, \hat{\theta}} \longrightarrow \hat{y}_{valid}$

$x_{test} \longrightarrow \boxed{f, \hat{\theta}} \longrightarrow \hat{y}_{test}$ $\quad L(\hat{y}_{test}, y_{test})$

Learning Process
In practice

Distribution over (x,y): P(x,y)



Loss

x_{train} ⟶ $\boxed{f, \theta}$ ⟶ $\hat{y}_{train}$ $L(\hat{y}_{train}, y_{train})$

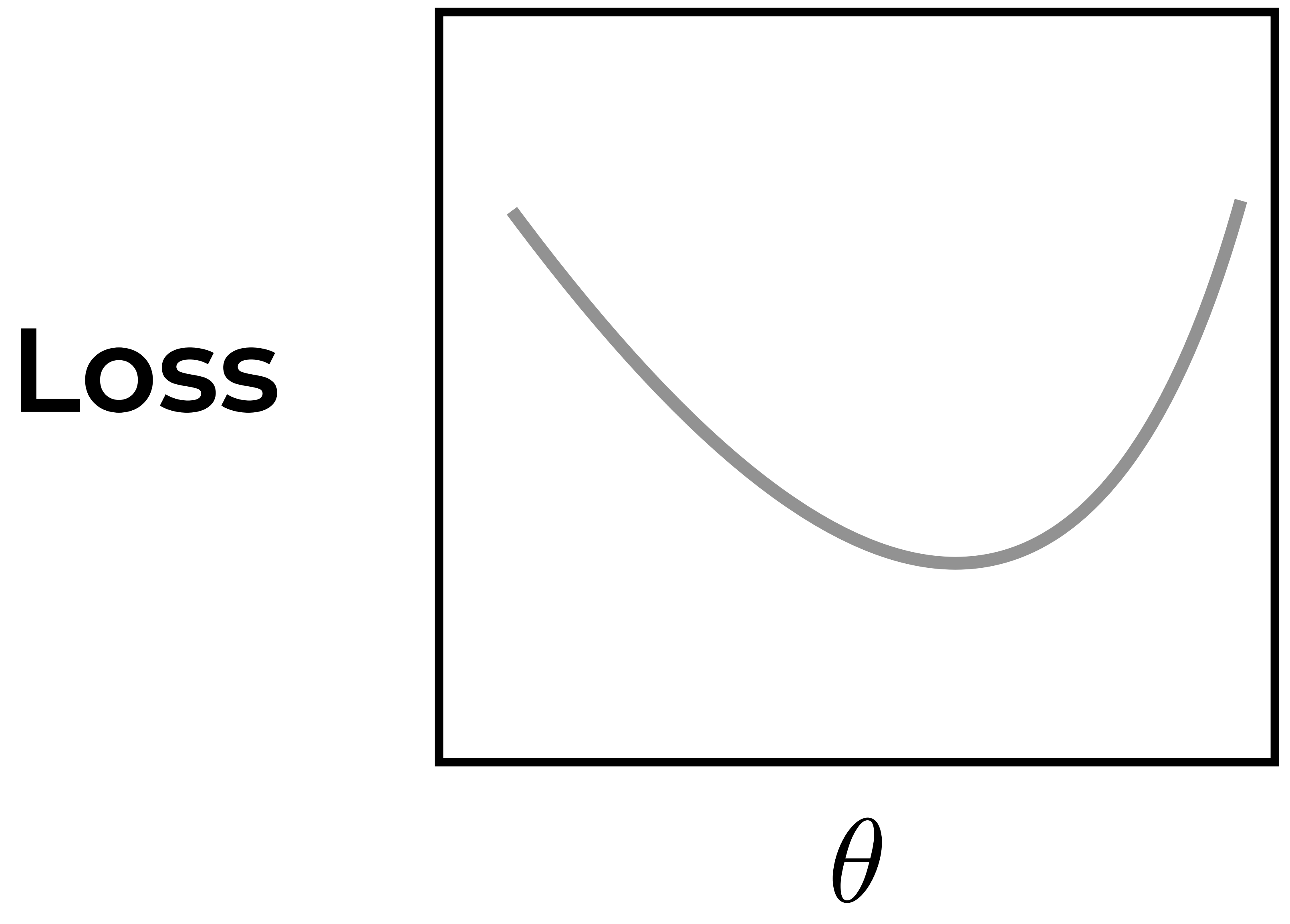
x_{valid} ⟶ $\boxed{f, \hat{\theta}}$ ⟶ $\hat{y}_{valid}$

Useful:
- to select hyper-parameters
- To pick the best model

x_{test} ⟶ $\boxed{f, \hat{\theta}}$ ⟶ $\hat{y}_{test}$ $L(\hat{y}_{test}, y_{test})$

Learning

- Learn: Change the parameters to obtain better predictions



- In other words: change the parameters to minimize the loss
  - Take the derivative of the loss wrt the parameter: $\frac{\text{d Loss}}{\text{d}\theta}$

Different models

- f: linear regression, θ has a closed-form solution

- f: neural network, θ does not have a closed-forum solution. Gradient descent is used

- Given a training set: $\{(\mathrm{x}_{\mathrm{train}}, \mathrm{y}_{\mathrm{train}})\}$

- Initialize $\hat{\theta}_1$ randomly

for t $= 1, 2, \ldots$ (epochs) **do**
    for i $= 1, 2, \ldots$ (datum) **do**

        - Obtain the predictions $\{\mathrm{f}(\mathrm{x}_{\mathrm{train}}; \hat{\theta}_{\mathrm{t}})\}$ (Forward propagation)

        - Compute the Loss: $\mathrm{Loss}_{\mathrm{ti}} := \mathrm{L}(\mathrm{f}(\mathrm{x}_{\mathrm{i}}; \hat{\theta}_{\mathrm{t}}), \mathrm{y}_{\mathrm{i}})$

        - Find the derivative of the loss: $\frac{\mathrm{d}\ \mathrm{Loss}_{\mathrm{ti}}}{\mathrm{d}\ \hat{\theta}_{\mathrm{t}}}$
        - Update parameters: $\hat{\theta}_{\mathrm{t}+1} = \hat{\theta}_{\mathrm{t}} - \alpha \frac{\mathrm{d}\ \mathrm{Loss}_{\mathrm{ti}}}{\mathrm{d}\ \hat{\theta}_{\mathrm{t}}}$
        - If $||\hat{\theta}_{\mathrm{t}+1} - \hat{\theta}_{\mathrm{t}}||_2^2 < \epsilon$ then stop
    end for
end for

- Given a training set: $\{(\mathbf{x}_{\mathbf{train}}, \mathbf{y}_{\mathbf{train}})\}$

- Initialize $\hat{\theta}_1$ randomly

Stochastic Gradient Descent

for t $= 1, 2, \ldots$ (epochs) **do**
    for i $= 1, 2, \ldots$ (datum) **do**

- Obtain the predictions $\{\mathbf{f}(\mathbf{x}_{\mathbf{train}}; \hat{\theta}_{\mathbf{t}})\}$ (Forward propagation)

- Compute the Loss: $\mathbf{Loss}_{\mathbf{ti}} := \mathbf{L}(\mathbf{f}(\mathbf{x}_{\mathbf{i}}; \hat{\theta}_{\mathbf{t}}), \mathbf{y}_{\mathbf{i}})$

- Find the derivative of the loss: $\frac{\mathbf{d}\ \mathbf{Loss}_{\mathbf{ti}}}{\mathbf{d}\ \hat{\theta}_{\mathbf{t}}}$
- Update parameters: $\hat{\theta}_{\mathbf{t}+1} = \hat{\theta}_{\mathbf{t}} - \alpha \frac{\mathbf{d}\ \mathbf{Loss}_{\mathbf{ti}}}{\mathbf{d}\ \hat{\theta}_{\mathbf{t}}}$
- If $||\hat{\theta}_{\mathbf{t}+1} - \hat{\theta}_{\mathbf{t}}||_2^2 < \epsilon$ then stop

    end for
end for

Probabilistic Models

Loss

$$\mathbf{x} \longrightarrow \boxed{\mathbf{f}, \theta} \longrightarrow \mathbf{f}(\mathbf{x}; \theta) \equiv \hat{\mathbf{y}} \qquad \mathbf{L}(\hat{\mathbf{y}}, \mathbf{y})$$

Example

Data: 952 1064 965 1037 871 1029 1138 (unsupervised problem)

Example

Data: 952 1064 965 1037 871 1029 1138 (unsupervised problem)

Model: $P(x \mid \theta) := \mathcal{N}(\mu, 1)$

Example

Data: $\boxed{\textbf{952}}$ **1064 965 1037 871 1029 1138 (unsupervised problem)**

Model: $\mathbf{P}(\mathbf{x} \mid \theta) := \mathcal{N}(\boldsymbol{\mu}, \mathbf{1})$

Likelihood for a single datum:

Example

Data: $\boxed{952}$ **1064 965 1037 871 1029 1138 (unsupervised problem)**

Model: $\mathsf{P}(\mathsf{x} \mid \theta) := \mathcal{N}(\mu, 1)$

$\boxed{\text{Likelihood for a single datum:}}$

$$\text{Likelihood}(\mathsf{x} \mid \mu, 1) = \frac{1}{\sqrt{2\pi}} \exp - \frac{(\mathsf{x} - \mu)^2}{2}$$

Example

Data: $\boxed{952}$ **1064 965 1037 871 1029 1138 (unsupervised problem)**

Model: $P(x \mid \theta) := \mathcal{N}(\mu, 1)$

$\boxed{\text{Likelihood for a single datum:}}$

$$\text{Likelihood}(x \mid \mu, 1) = \frac{1}{\sqrt{2\pi}} \exp - \frac{(x-\mu)^2}{2}$$

Log-Likelihood

$$= \log \frac{1}{\sqrt{2\pi}} \exp - \frac{(x-\mu)^2}{2}$$

$$= \log 1 - \frac{1}{2} \log 2\pi - \frac{(x-\mu)^2}{2}$$

Example

Data: $\boxed{952}$ **1064 965 1037 871 1029 1138 (unsupervised problem)**

Model: $\mathbf{P}(\mathbf{x} \mid \theta) := \mathcal{N}(\mu, 1)$

Likelihood for a single datum:

$$\mathbf{Likelihood}(\mathbf{x} \mid \mu, 1) = \frac{1}{\sqrt{2\pi}} \exp - \frac{(\mathbf{x} - \mu)^2}{2}$$

Log-Likelihood

$$= \log \frac{1}{\sqrt{2\pi}} \exp - \frac{(\mathbf{x} - \mu)^2}{2}$$

$$= \log 1 - \frac{1}{2} \log 2\pi - \frac{(\mathbf{x} - \mu)^2}{2}$$

What value of μ maximizes it?

Example

Data: $\boxed{952}$ 1064 965 1037 871 1029 1138 (unsupervised problem)

Model: $P(x \mid \theta) := \mathcal{N}(\mu, 1)$

$\boxed{\text{Likelihood for a single datum:}}$

$$\text{Likelihood}(x \mid \mu, 1) = \frac{1}{\sqrt{2\pi}} \exp - \frac{(x-\mu)^2}{2}$$

Log-Likelihood

$$= \log \frac{1}{\sqrt{2\pi}} \exp - \frac{(x-\mu)^2}{2}$$

$$= \log 1 - \frac{1}{2} \log 2\pi - \frac{(x-\mu)^2}{2}$$

What value of μ maximizes it?

$$\frac{\text{d Log-Likelihood}}{\text{d } \mu}$$

$$= \frac{\text{d} \frac{(x-\mu)^2}{2}}{\text{d } \mu}$$

$$= (x - \mu)$$

set to 0

$$\mu = x$$

Example

Data: $\boxed{952}$ 1064 965 1037 871 1029 1138 (unsupervised problem)

Model: $P(x \mid \theta) := \mathcal{N}(\mu, 1)$

Likelihood for a single datum:

$$\text{Likelihood}(x \mid \mu, 1) = \frac{1}{\sqrt{2\pi}} \exp -\frac{(x-\mu)^2}{2}$$

Log-Likelihood

$$= \log \frac{1}{\sqrt{2\pi}} \exp -\frac{(x-\mu)^2}{2}$$

$$= \log 1 - \frac{1}{2}\log 2\pi - \frac{(x-\mu)^2}{2}$$

What value of μ maximizes it?

$$\frac{d \text{ Log-Likelihood}}{d\mu}$$

$$= \frac{d \frac{(x-\mu)^2}{2}}{d\mu}$$

$$= (x - \mu)$$

set to 0

$\mu = x$ = 952

Data: (x,y)
Naive Bayes

Data: (x,y)
Naive Bayes

$$\textbf{Model}: \mathbf{P}(\mathbf{x}, \mathbf{y} \mid \theta) = \mathbf{P}(\mathbf{x} \mid \mathbf{y}, \theta)\mathbf{P}(\mathbf{y} \mid \theta)$$

Data: (x,y)
Naive Bayes

$$\textbf{Model}: \mathbf{P}(\mathbf{x}, \mathbf{y} \mid \theta) = \mathbf{P}(\mathbf{x} \mid \mathbf{y}, \theta)\mathbf{P}(\mathbf{y} \mid \theta)$$

Data: (x,y)
Naive Bayes

$$\textbf{Model}: \mathbf{P}(\mathbf{x}, \mathbf{y} \mid \theta) = \mathbf{P}(\mathbf{x} \mid \mathbf{y}, \theta)\mathbf{P}(\mathbf{y} \mid \theta)$$

Data: (x,y)
Naive Bayes

$\textbf{Model}: \mathbf{P}(\mathbf{x}, \mathbf{y} \mid \theta) = \mathbf{P}(\mathbf{x} \mid \mathbf{y}, \theta)\mathbf{P}(\mathbf{y} \mid \theta)$

Data: x
Gaussian Mixture Models

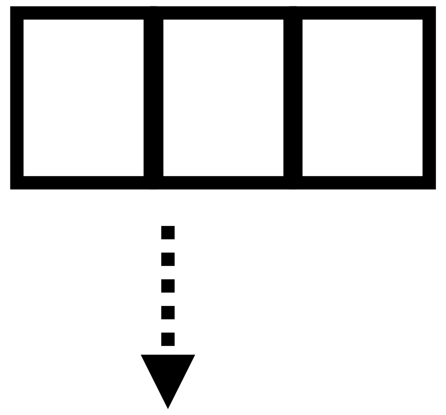
Data: (x,y)
Naive Bayes

$$\textbf{Model} : \mathbf{P}(\mathbf{x}, \mathbf{y} \mid \theta) = \mathbf{P}(\mathbf{x} \mid \mathbf{y}, \theta)\mathbf{P}(\mathbf{y} \mid \theta)$$

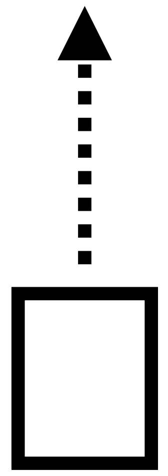
Data: x
Gaussian Mixture Models

$$\textbf{Model} : \mathbf{P}(\mathbf{x} \mid \theta) = \sum_{\mathbf{k}=1}^{\mathbf{K}} \mathbf{P}(\theta_{\mathbf{x}} = \mathbf{k}) \underbrace{\mathbf{P}(\mathbf{x} \mid \theta_{\mathbf{k}})}_{\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_{\mathbf{k}}, \Sigma_{\mathbf{k}})} \quad \textbf{(K components)}$$

Data: (x,y)
Naive Bayes

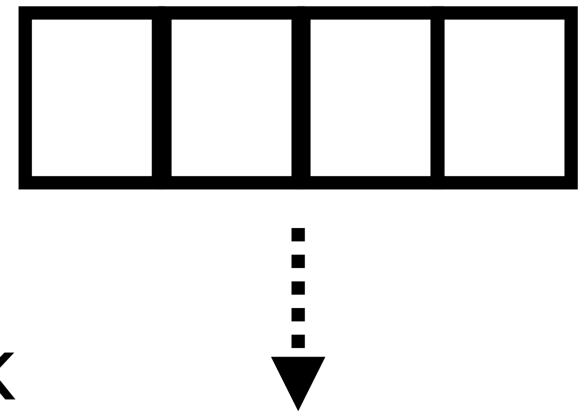


$$\textbf{Model}: \mathbf{P}(\mathbf{x}, \mathbf{y} \mid \theta) = \mathbf{P}(\mathbf{x} \mid \mathbf{y}, \theta)\mathbf{P}(\mathbf{y} \mid \theta)$$



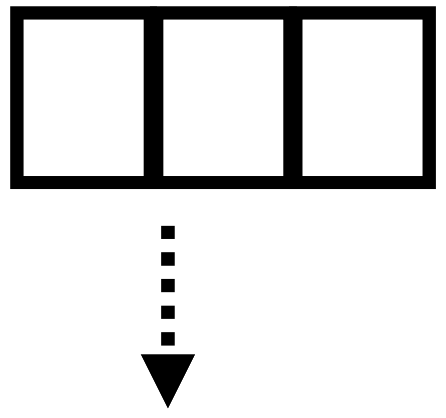
Data: x
Gaussian Mixture Models

K=4

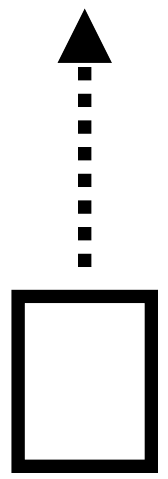


$$\textbf{Model}: \mathbf{P}(\mathbf{x} \mid \theta) = \sum_{\mathbf{k}=1}^{\mathbf{K}} \mathbf{P}(\theta_{\mathbf{x}} = \mathbf{k}) \underbrace{\mathbf{P}(\mathbf{x} \mid \theta_{\mathbf{k}})}_{\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_{\mathbf{k}}, \Sigma_{\mathbf{k}})} \quad \textbf{(K components)}$$

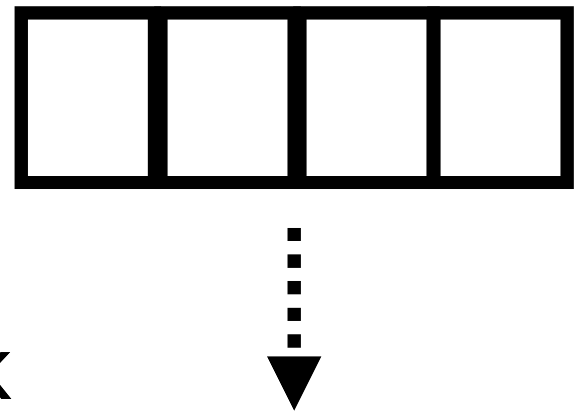
Data: (x,y)
Naive Bayes



$$\textbf{Model}: \mathbf{P}(\mathbf{x}, \mathbf{y} \mid \theta) = \mathbf{P}(\mathbf{x} \mid \mathbf{y}, \theta)\mathbf{P}(\mathbf{y} \mid \theta)$$



Data: x
Gaussian Mixture Models

K=4 

$$\textbf{Model}: \mathbf{P}(\mathbf{x} \mid \theta) = \sum_{\mathbf{k}=1}^{\mathbf{K}} \mathbf{P}(\theta_{\mathbf{x}} = \mathbf{k}) \underbrace{\mathbf{P}(\mathbf{x} \mid \theta_{\mathbf{k}})}_{\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_{\mathbf{k}}, \Sigma_{\mathbf{k}})} \text{ (K components)}$$

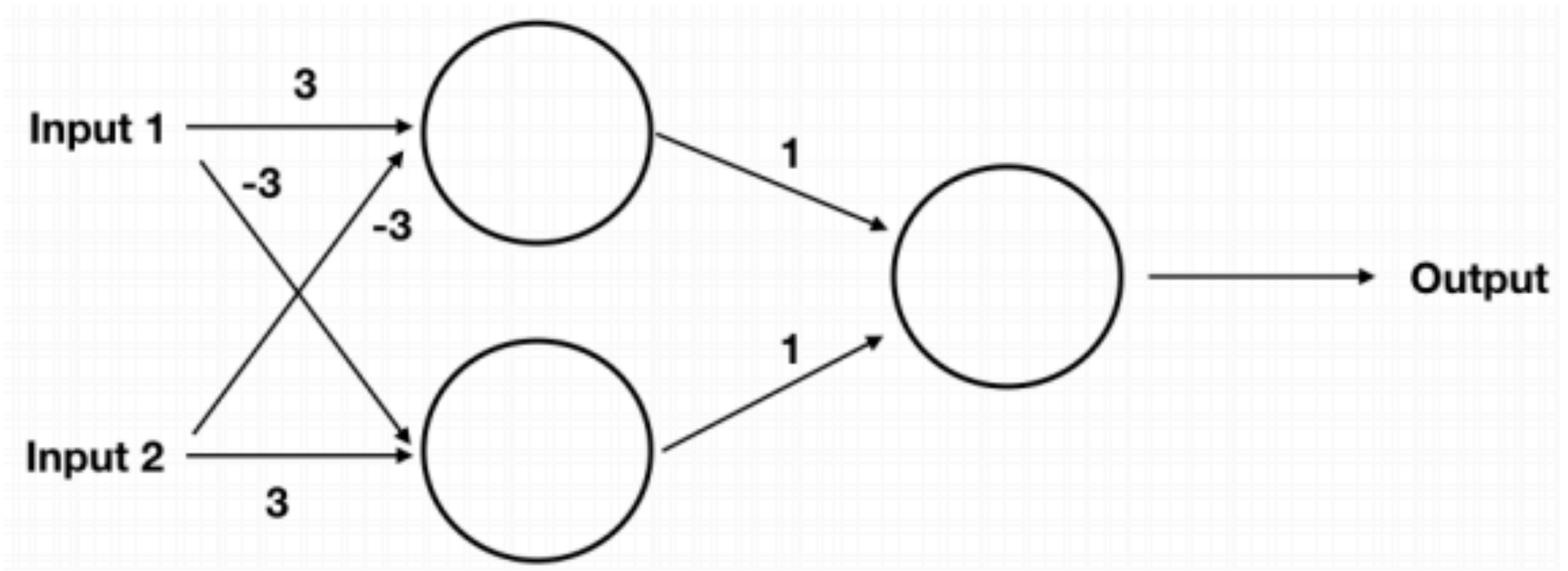
$$\textbf{Max. likelihood (MLE)}: \hat{\theta}_{\mathbf{MLE}} = \arg\max_{\theta} \mathbf{P}(\mathbf{x} \mid \theta)$$

MLPs / RNNs / CNNs

- MLPs: layers are fully-connected to the next layer
- RNNs: inputs at each layer
  - Typical application: time-series modelling
- CNNs: replace matrix multiplications by convolutions (sparse connections, weight sharing) + pooling
  - Typical application: object recognition in images

MLPs

(e) (4 points) Consider the neural network below. We have estimated its parameters (shown next to their corresponding arrows).



The activation function of each unit in the network is a simple thresholding function:

$$\text{threshold}(x) = \begin{cases} 0 & \text{if } x \leq 0, \\ 1 & \text{if } x > 0. \end{cases} \tag{1}$$

For each of these four sets of inputs write down the network's output (i.e., its prediction) in the "Output" column of the table below.

Input 1	Input 2	Output
0	0	
1	1	
0	1	
1	0	