Signal Processing Systems

Machine learning for signal processing [5LSL0]

Ruud van Sloun, Rik Vullings

Technische Universiteit
Eindhoven
University of Technology

May 2019

Where innovation starts

Activation functions:

From optimal linear filtering to nonlinear classification

Definitions

Weight vector of n samples
$\mathbf{w} = [w_1, w_2, ..., w_n]^T$.

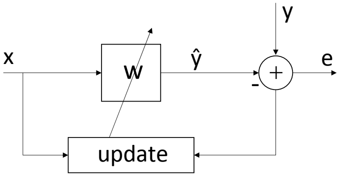
m observations consisting of data vectors $\mathbf{x}$ and outputs $\mathbf{y}$ collected in an $m \times n$ input data matrix:

$\mathbf{X} = \left[\mathbf{x}^{(0)}, \mathbf{x}^{(1)}, ..., \mathbf{x}^{(i)}, ..., \mathbf{x}^{(m)}\right]^T$.
$\mathbf{x}^{(i)} = \left[x_1^{(i)}, x_2^{(i)}, ..., x_n^{(i)}\right]^T$, with associated output $\mathbf{y}^{(i)}$ or $y^{(i)}$.

Set of input data vectors: $\mathbb{X}$

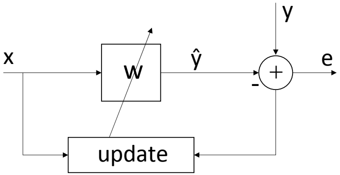
Linear models

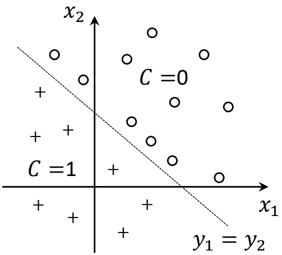
So far: optimal linear operations given some cost criterion.



Linear models

So far: optimal linear operations given some cost criterion.



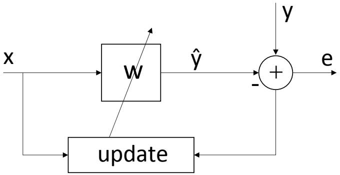


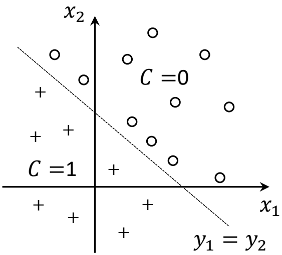
$$f(\mathbf{x}; \mathrm{W}) = \mathbf{W}\mathbf{x} = \mathbf{W}(x_1, x_2)^T$$

Same framework also enables classification:

Linear models

So far: optimal linear operations given some cost criterion.



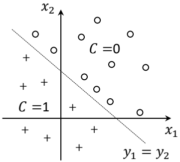


$f(\mathbf{x}; \mathrm{W}) = \mathbf{W}\mathbf{x} = \mathbf{W}(x_1, x_2)^T$

Same framework also enables classification:
$[y_1, y_2] = \mathrm{W}\mathbf{x}$. Given $\mathbf{y}$, classification through:

$$\mathcal{C}(\mathbf{x}) = \begin{cases} 1 & y_2 > y_1 \\ 0 & \text{else} \end{cases} \tag{1}$$

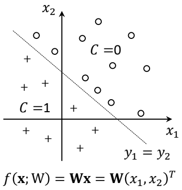
Linear models



$f(\mathbf{x}; \mathrm{W}) = \mathbf{W}\mathbf{x} = \mathbf{W}(x_1, x_2)^T$

Regression problem through MSE cost criterion:

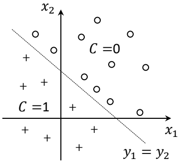
Linear models



Regression problem through MSE cost criterion:

$$J(\theta) = \frac{1}{N} \sum_{\mathbf{x} \in \mathbb{X}} \left(f^*(\mathbf{x}) - f(\mathbf{x}; \theta)\right)^2,$$

where $f^*(\mathbf{x})$ is some known/target output on $\mathbf{x} \in \mathbb{X}$, and $f(\mathbf{x}; \theta) = f(\mathbf{x}; \mathbf{W}) = \mathbf{W}^T\mathbf{x}$



$f(\mathbf{x}; \mathrm{W}) = \mathbf{Wx} = \mathbf{W}(x_1, x_2)^T$

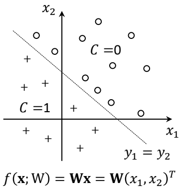
Regression problem through MSE cost criterion:

$J(\theta) = \frac{1}{N} \sum_{\mathbf{x} \in \mathbb{X}} \left(f^*(\mathbf{x}) - f(\mathbf{x}; \theta)\right)^2,$

where $f^*(\mathbf{x})$ is some known/target output on $\mathbf{x} \in \mathbb{X}$, and
$f(\mathbf{x}; \theta) = f(\mathbf{x}; \mathbf{W}) = \mathbf{W}^T \mathbf{x}$

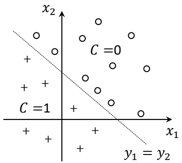
As before, we can minimize $J(\theta)$ w.r.t. the coefficients in matrix $\mathbf{W}$.

Linear models



Regression problem through MSE cost criterion:

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

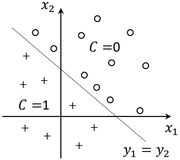


$f(\mathbf{x}; \mathrm{W}) = \mathbf{Wx} = \mathbf{W}(x_1, x_2)^T$

Minimum MSE cost criterion for vector $\mathbf{w}$:

$J(\theta) = \sum_{\mathbf{x} \in \mathbb{X}} \left(f^*(\mathbf{x}) - f(\mathbf{x}; \theta)\right)^2$ // omitting scalar

Linear models

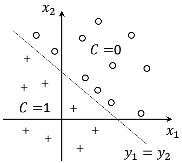


Minimum MSE cost criterion for vector **w**:

$$J(\theta) = \sum_{\mathbf{x} \in \mathbb{X}} \left(f^*(\mathbf{x}) - f(\mathbf{x}; \theta)\right)^2 \text{ // omitting scalar}$$

$$J(\mathbf{w}) = (\mathbf{w}^T\mathbf{X} - \mathbf{y})(\mathbf{w}^T\mathbf{X} - \mathbf{y})^T \text{ // rewriting in matrix form}$$

$$\mathbf{X} = \left[\mathbf{x}^{(0)}, \mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(i)}, \ldots, \mathbf{x}^{(m)}\right]$$
$$\mathbf{y} = \left[y^{(0)}, y^{(1)}, \ldots, y^{(i)}, \ldots, y^{(m)}\right]$$



$f(\mathbf{x}; \mathrm{W}) = \mathbf{Wx} = \mathbf{W}(x_1, x_2)^T$

Minimum MSE cost criterion for vector $\mathbf{w}$:

$J(\theta) = \sum_{\mathbf{x}\in\mathbb{X}} \left(f^*(\mathbf{x}) - f(\mathbf{x};\theta)\right)^2$ // omitting scalar

$J(\mathbf{w}) = (\mathbf{w}^T\mathbf{X} - \mathbf{y})(\mathbf{w}^T\mathbf{X} - \mathbf{y})^T$ // rewriting in matrix form

$$\mathbf{X} = \left[\mathbf{x}^{(0)}, \mathbf{x}^{(1)}, \dots, \mathbf{x}^{(i)}, \dots, \mathbf{x}^{(m)}\right]$$

$$\mathbf{y} = \left[y^{(0)}, y^{(1)}, \dots, y^{(i)}, \dots, y^{(m)}\right]$$

$$\Rightarrow \partial_{\mathbf{w}} J(\mathbf{w}) = 2\mathbf{X}\mathbf{X}^T\mathbf{w} - 2\mathbf{X}\mathbf{y}^T = 0$$

$$\Rightarrow \mathbf{w} = \left(\mathbf{X}\mathbf{X}^T\right)^{-1}\mathbf{X}\mathbf{y}^T$$

Example:

OR function: $\mathbb{X} = \{[0,0], [0,1], [1,0], [1,1]\}$; target function $f^*(\mathbf{x})$ gives $\{0,1,1,1\}$ on $\mathbb{X}$.

Linear models

Example:

OR function: $\mathbb{X} = \{[0,0]\,, [0,1]\,, [1,0]\,, [1,1]\}$; target function $f^*(\mathbf{x})$ gives $\{0,1,1,1\}$ on $\mathbb{X}$.

GOAL: Find an optimal (minimum MSE) function $f(\mathbf{x};\theta)$ that can fit these data points.

Linear models

Example:

OR function: $\mathbb{X} = \{[0,0], [0,1], [1,0], [1,1]\}$; target function $f^*(\mathbf{x})$ gives $\{0,1,1,1\}$ on $\mathbb{X}$.

GOAL: Find an optimal (minimum MSE) function $f(\mathbf{x};\theta)$ that can fit these data points.

Define a linear function $f(\mathbf{x};\mathbf{w}) = \mathbf{w}^T\mathbf{x}$

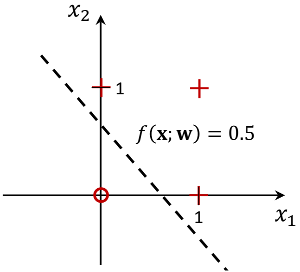
Linear models

Example:

OR function: $\mathbb{X} = \{[0,0], [0,1], [1,0], [1,1]\}$; target function $f^*(\mathbf{x})$ gives $\{0,1,1,1\}$ on $\mathbb{X}$.

GOAL: Find an optimal (minimum MSE) function $f(\mathbf{x};\theta)$ that can fit these data points.

Define a linear function $f(\mathbf{x};\mathbf{w}) = \mathbf{w}^T\mathbf{x}$



Solving the normal equations leads to: $\mathbf{w} = [0.6667, 0.6667]$

Example 2:

XOR (exclusive or): $\mathbb{X} = \{[0,0]\,, [0,1]\,, [1,0]\,, [1,1]\}$; target function $f^*(\mathbf{x})$ gives $\{0,1,1,0\}$ on $\mathbb{X}$.

Example 2:

XOR (exclusive or): $\mathbb{X} = \{[0,0], [0,1], [1,0], [1,1]\}$; target function $f^*(\mathbf{x})$ gives $\{0,1,1,0\}$ on $\mathbb{X}$.

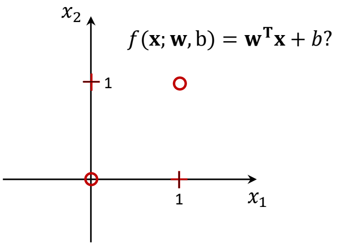
GOAL: Find an optimal (minimum MSE) function $f(\mathbf{x};\theta)$ that can fit these data points.

Example 2:

XOR (exclusive or): $\mathbb{X} = \{[0,0], [0,1], [1,0], [1,1]\}$; target function $f^*(\mathbf{x})$ gives $\{0,1,1,0\}$ on $\mathbb{X}$.

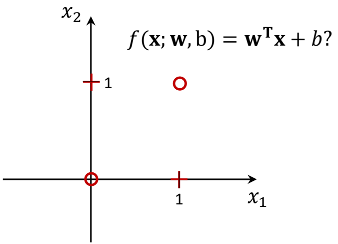
GOAL: Find an optimal (minimum MSE) function $f(\mathbf{x};\theta)$ that can fit these data points.

Define a linear function $f(\mathbf{x};\mathbf{w},b) = \mathbf{w}^T\mathbf{x} + b$



Example 2:
XOR (exclusive or): $\mathbb{X} = \{[0,0], [0,1], [1,0], [1,1]\}$; target function $f^*(\mathbf{x})$ gives $\{0,1,1,0\}$ on $\mathbb{X}$.

GOAL: Find an optimal (minimum MSE) function $f(\mathbf{x};\theta)$ that can fit these data points.
Define a linear function $f(\mathbf{x};\mathbf{w},b) = \mathbf{w}^T\mathbf{x} + b$



Solving the normal equations leads to: $\mathbf{w} = \mathbf{0}$ and $b = \frac{1}{2}$ (work this out yourselves).

Nonlinear transforms

Problem: A linear model $f(\mathbf{x}; \mathbf{w}, b) = \mathbf{w}^T\mathbf{x} + b$ is not able to represent all desirable functions (e.g XOR, and in practice, many detection/estimation signal processing functions).

Nonlinear transforms

Problem: A linear model $f(\mathbf{x}; \mathbf{w}, b) = \mathbf{w}^T \mathbf{x} + b$ is not able to represent all desirable functions (e.g XOR, and in practice, many detection/estimation signal processing functions).

Solution: Define a mapping function that (nonlinearly) transforms $\mathbf{x}$ into a space $\mathbf{h}$ in which the function can be represented with a linear model:

Nonlinear transforms

Problem: A linear model $f(\mathbf{x}; \mathbf{w}, b) = \mathbf{w}^T \mathbf{x} + b$ is not able to represent all desirable functions (e.g XOR, and in practice, many detection/estimation signal processing functions).

Solution: Define a mapping function that (nonlinearly) transforms $\mathbf{x}$ into a space $\mathbf{h}$ in which the function can be represented with a linear model:

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

Nonlinear transforms

Problem: A linear model $f(\mathbf{x}; \mathbf{w}, b) = \mathbf{w}^T\mathbf{x} + b$ is not able to represent all desirable functions (e.g XOR, and in practice, many detection/estimation signal processing functions).

Solution: Define a mapping function that (nonlinearly) transforms $\mathbf{x}$ into a space $\mathbf{h}$ in which the function can be represented with a linear model:

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

where

$$\mathbf{h} = f^{(1)}(\mathbf{x}; \mathbf{W}^{(1)}, \mathbf{b}^{(1)}). \tag{3}$$

Nonlinear transforms

Solution: Define a mapping function that (nonlinearly) transforms $\mathbf{x}$ into a space $\mathbf{h}$ in which the function can be represented with a linear model:

$$f(\mathbf{x};\theta) = f^{(2)}(\mathbf{h};\mathbf{w}^{(2)}, b^{(2)}), \tag{4}$$

where

$$\mathbf{h} = f^{(1)}(\mathbf{x};\mathbf{W}^{(1)}, b^{(1)}). \tag{5}$$

Nonlinear transforms

Solution: Define a mapping function that (nonlinearly) transforms $\mathbf{x}$ into a space $\mathbf{h}$ in which the function can be represented with a linear model:

$$f(\mathbf{x}; \theta) = f^{(2)}(\mathbf{h}; \mathbf{w}^{(2)}, b^{(2)}), \tag{4}$$

where

$$\mathbf{h} = g(\mathbf{W}^{(1)}\mathbf{x} + \mathbf{b}^{(1)}). \tag{6}$$

Nonlinear transforms

Solution: Define a mapping function that (nonlinearly) transforms $\mathbf{x}$ into a space $\mathbf{h}$ in which the function can be represented with a linear model:

$$f(\mathbf{x};\theta) = f^{(2)}(\mathbf{h};\mathbf{w}^{(2)}, b^{(2)}), \tag{4}$$

where

$$\mathbf{h} = g(\mathbf{W}^{(1)}\mathbf{x} + \mathbf{b}^{(1)}). \tag{6}$$

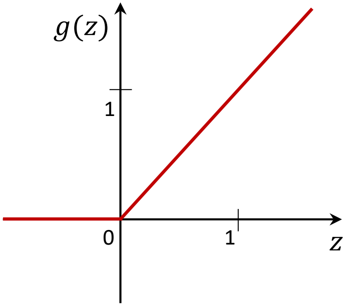
Let's consider a rectifying nonlinearity (in the machine learning community known as a rectified linear unit, or ReLU), such that

$$\mathbf{h} = \max(0, \mathbf{W}^{(1)}\mathbf{x} + \mathbf{b}^{(1)}). \tag{7}$$

Nonlinear transforms

$$f(\mathbf{x};\theta) = f^{(2)}(\mathbf{h};\mathbf{w}^{(2)}, b^{(2)}), \tag{8}$$

$$\mathbf{h} = \max(0, \mathbf{W}^{(1)}\mathbf{x} + \mathbf{b}^{(1)}). \tag{9}$$



Nonlinear transforms

$$f(\mathbf{x};\theta) = f^{(2)}(\mathbf{h};\mathbf{w}^{(2)}, b^{(2)}), \tag{10}$$

$$\mathbf{h} = \max(0, \mathbf{W}^{(1)}\mathbf{x} + \mathbf{b}^{(1)}). \tag{11}$$

The complete network is then:

$$f(\mathbf{x};\mathbf{W}^{(1)}, \mathbf{b}^{(1)}, \mathbf{w}^{(2)}, b^{(2)}) = \left(\mathbf{w}^{(2)}\right)^T \max(0, \mathbf{W}^{(1)}\mathbf{x}+\mathbf{b}^{(1)})+b^{(2)}. \tag{12}$$

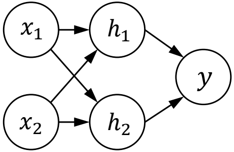
Nonlinear transforms

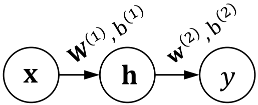
$$f(\mathbf{x};\theta) = f^{(2)}(\mathbf{h};\mathbf{w}^{(2)}, b^{(2)}), \tag{10}$$

$$\mathbf{h} = \max(0, \mathbf{W}^{(1)}\mathbf{x} + \mathbf{b}^{(1)}). \tag{11}$$

The complete network is then:

$$f(\mathbf{x};\mathbf{W}^{(1)}, \mathbf{b}^{(1)}, \mathbf{w}^{(2)}, b^{(2)}) = \left(\mathbf{w}^{(2)}\right)^T \max(0, \mathbf{W}^{(1)}\mathbf{x}+\mathbf{b}^{(1)})+b^{(2)}. \tag{12}$$





Nonlinear transforms

Given:

$$f(\mathbf{x}; \mathbf{W}^{(1)}, \mathbf{b}^{(1)}, \mathbf{w}^{(2)}, b^{(2)}) = \left(\mathbf{w}^{(2)}\right)^T \max(0, \mathbf{W}^{(1)}\mathbf{x} + \mathbf{b}^{(1)}) + b^{(2)}, \quad (13)$$

what parameter values should $\mathbf{W}^{(1)}, \mathbf{b}^{(1)}, \mathbf{w}^{(2)}$and $b^{(2)}$ have?

Nonlinear transforms

Given:

$$f(\mathbf{x}; \mathbf{W}^{(1)}, \mathbf{b}^{(1)}, \mathbf{w}^{(2)}, b^{(2)}) = \left(\mathbf{w}^{(2)}\right)^T \max(0, \mathbf{W}^{(1)}\mathbf{x}+\mathbf{b}^{(1)})+b^{(2)}, \quad (13)$$

what parameter values should $\mathbf{W}^{(1)}, \mathbf{b}^{(1)}, \mathbf{w}^{(2)}$and $b^{(2)}$ have? Consider the following values:

$$\mathbf{W}^{(1)} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}, \mathbf{b}^{(1)} = \begin{bmatrix} 0 \\ -1 \end{bmatrix}, \mathbf{w}^{(2)} = \begin{bmatrix} 1 \\ -2 \end{bmatrix} \text{ and } b^{(2)} = 0.$$

Nonlinear transforms

Given:

$$f(\mathbf{x}; \mathbf{W}^{(1)}, \mathbf{b}^{(1)}, \mathbf{w}^{(2)}, b^{(2)}) = \left(\mathbf{w}^{(2)}\right)^T \max(0, \mathbf{W}^{(1)}\mathbf{x}+\mathbf{b}^{(1)})+b^{(2)}, \quad (13)$$

what parameter values should $\mathbf{W}^{(1)}, \mathbf{b}^{(1)}, \mathbf{w}^{(2)}$and $b^{(2)}$ have? Consider the following values:

$$\mathbf{W}^{(1)} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}, \mathbf{b}^{(1)} = \begin{bmatrix} 0 \\ -1 \end{bmatrix}, \mathbf{w}^{(2)} = \begin{bmatrix} 1 \\ -2 \end{bmatrix} \text{ and } b^{(2)} = 0.$$

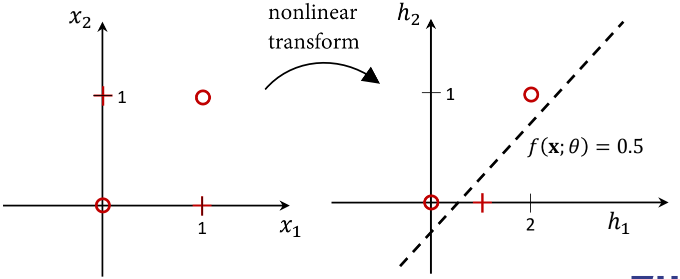
Then $\mathbf{X} = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \end{bmatrix}$, maps to outputs: $y = \begin{bmatrix} 0 & 1 & 1 & 0 \end{bmatrix}$.

Nonlinear transforms

$$f(\mathbf{x}; \mathbf{W}^{(1)}, \mathbf{b}^{(1)}, \mathbf{w}^{(2)}, b^{(2)}) = \left(\mathbf{w}^{(2)}\right)^T \max(0, \mathbf{W}^{(1)}\mathbf{x} + \mathbf{b}^{(1)}) + b^{(2)}, \quad (14)$$

where

$$\mathbf{W}^{(1)} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}, \mathbf{b}^{(1)} = \begin{bmatrix} 0 \\ -1 \end{bmatrix}, \mathbf{w}^{(2)} = \begin{bmatrix} 1 \\ -2 \end{bmatrix} \text{ and } b^{(2)} = 0.$$



Cost function - maximum likelihood

Optimal parameter values?

Cost function - maximum likelihood

Optimal parameter values?

⇒ Maximize the likelihood of the observations ($\mathbf{y}$) given the model (θ) and the input data ($\mathbf{x}$).

Negative log-likelihood cost function:

$$J(\theta) = -\mathbb{E}_{(\mathbf{x},\mathbf{y})\sim\hat{p}_{\text{data}}} \log p_{\text{model}}(\mathbf{y}|\mathbf{x},\theta), \tag{15}$$

Cost function - maximum likelihood

Optimal parameter values?
$\Rightarrow$ Maximize the likelihood of the observations ($\mathbf{y}$) given the model (θ) and the input data ($\mathbf{x}$).

Negative log-likelihood cost function:

$$J(\theta) = -\mathbb{E}_{(\mathbf{x},\mathbf{y})\sim\hat{p}_{\text{data}}} \log p_{\text{model}}(\mathbf{y}|\mathbf{x},\theta), \tag{15}$$

$\hat{p}_{\text{data}}$: Empirical data distribution defined by the training set $\mathbb{X}$
$\hat{p}_{\text{model}}(\mathbf{y}|\mathbf{x};\theta)$: Error model for the predictions.

Cost function - maximum likelihood

Optimal parameter values?
$\Rightarrow$ Maximize the likelihood of the observations ($\mathbf{y}$) given the model (θ) and the input data ($\mathbf{x}$).

Negative log-likelihood cost function:

$$J(\theta) = -\mathbb{E}_{(\mathbf{x},\mathbf{y})\sim\hat{p}_{\text{data}}} \log p_{\text{model}}(\mathbf{y}|\mathbf{x},\theta), \tag{15}$$

$\hat{p}_{\text{data}}$: Empirical data distribution defined by the training set $\mathbb{X}$
$\hat{p}_{\text{model}}(\mathbf{y}|\mathbf{x};\theta)$: Error model for the predictions.

$$J(\theta) = -\log \prod_{i=0}^{m-1} \hat{p}_{\text{model}}(\mathbf{y}^{(i)}|\mathbf{x}^{(i)},\theta), \tag{16}$$

Cost function - maximum likelihood

Optimal parameter values?
$\Rightarrow$ Maximize the likelihood of the observations ($\mathbf{y}$) given the model (θ) and the input data ($\mathbf{x}$).

Negative log-likelihood cost function:

$$J(\theta) = -\mathbb{E}_{(\mathbf{x},\mathbf{y})\sim\hat{p}_{\text{data}}} \log p_{\text{model}}(\mathbf{y}|\mathbf{x},\theta), \tag{15}$$

$\hat{p}_{\text{data}}$: Empirical data distribution defined by the training set $\mathbb{X}$
$\hat{p}_{\text{model}}(\mathbf{y}|\mathbf{x};\theta)$: Error model for the predictions.

$$J(\theta) = -\log \prod_{i=0}^{m-1} \hat{p}_{\text{model}}(\mathbf{y}^{(i)}|\mathbf{x}^{(i)},\theta), \tag{16}$$

$$J(\theta) = -\sum_{i=0}^{m-1} \log \hat{p}_{\text{model}}(\mathbf{y}^{(i)}|\mathbf{x}^{(i)},\theta), \tag{17}$$

Cost function - maximum likelihood

Optimal parameter values?

$\Rightarrow$ Maximize the likelihood of the observations ($\mathbf{y}$) given the model (θ) and the input data ($\mathbf{x}$).

Negative log-likelihood cost function:

$$J(\theta) = -\sum_{i=0}^{m-1} \log \hat{p}_{\text{model}}(\mathbf{y}^{(i)}|\mathbf{x}^{(i)}, \theta), \tag{18}$$

Cost function - maximum likelihood

Optimal parameter values?

$\Rightarrow$ Maximize the likelihood of the observations ($\mathbf{y}$) given the model (θ) and the input data ($\mathbf{x}$).

Negative log-likelihood cost function:

$$J(\theta) = -\sum_{i=0}^{m-1} \log \hat{p}_{\text{model}}(\mathbf{y}^{(i)}|\mathbf{x}^{(i)}, \theta), \tag{18}$$

$$\hat{\theta} = \underset{\theta}{\operatorname{argmin}}\, J(\theta) = \underset{\theta}{\operatorname{argmin}} \left(-\sum_{i=0}^{m-1} \log \hat{p}_{\text{model}}(\mathbf{y}^{(i)}|\mathbf{x}^{(i)}, \theta) \right), \tag{19}$$

Cost function - maximum likelihood

Optimal parameter values:

$$\hat{\theta} = \underset{\theta}{\operatorname{argmin}} \left(- \sum_{i=0}^{m-1} \log \hat{p}_{\text{model}}(\mathbf{y}^{(i)} | \mathbf{x}^{(i)}, \theta) \right), \tag{20}$$

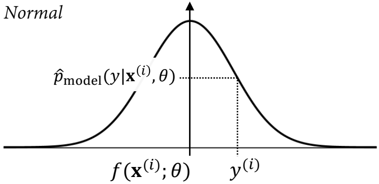
Choice of $\hat{p}_{\text{model}}$ depends on the error distribution, e.g.:

Cost function - maximum likelihood

Optimal parameter values:

$$\hat{\theta} = \underset{\theta}{\operatorname{argmin}} \left(- \sum_{i=0}^{m-1} \log \hat{p}_{\text{model}}(\mathbf{y}^{(i)} | \mathbf{x}^{(i)}, \theta) \right), \tag{20}$$

Choice of $\hat{p}_{\text{model}}$ depends on the error distribution, e.g.:



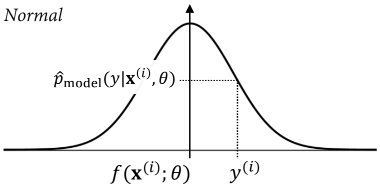
- Normal (Gaussian) distribution
    - Regression of continuous variables

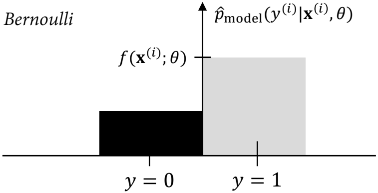
Cost function - maximum likelihood

Optimal parameter values:

$$\hat{\theta} = \underset{\theta}{\operatorname{argmin}} \left(- \sum_{i=0}^{m-1} \log \hat{p}_{\text{model}}(\mathbf{y}^{(i)} | \mathbf{x}^{(i)}, \theta) \right), \tag{20}$$

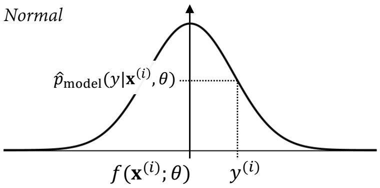
Choice of $\hat{p}_{\text{model}}$ depends on the error distribution, e.g.:





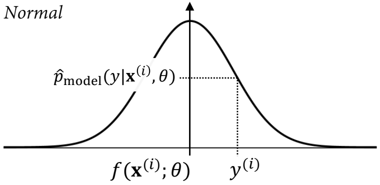
- Normal (Gaussian) distribution
  - Regression of continuous variables
- Bernoulli/Categorical distribution
  - Binary/multi-class classification

Cost function - maximum likelihood



Optimal parameter values for Gaussian error distribution,
$\hat{p}_{\text{model}} \sim \mathcal{N}\left\{\mathbf{y}; f(\mathbf{x}; \theta), \sigma\mathbf{I}\right\}$:

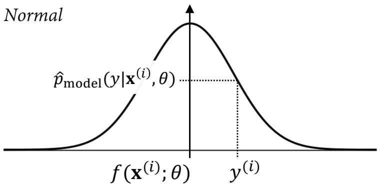
Cost function - maximum likelihood



Optimal parameter values for Gaussian error distribution, $\hat{p}_{\text{model}} \sim \mathcal{N}\left\{\mathbf{y}; f(\mathbf{x};\theta), \sigma\mathbf{I}\right\}$:

"The probability of observing $y^{(i)}$ follows a normal distribution with the model prediction $f(\mathbf{x}^{(i)};\theta)$ as its mean."

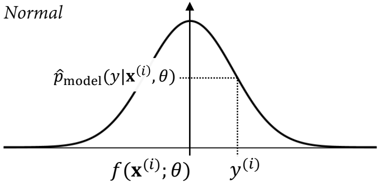
Cost function - maximum likelihood



Optimal parameter values for Gaussian error distribution, $\hat{p}_{\text{model}} \sim \mathcal{N}\{\mathbf{y}; f(\mathbf{x}; \theta), \sigma \mathbf{I}\}$:

$$\hat{\theta} = \underset{\theta}{\operatorname{argmin}} \left(-\frac{1}{m} \sum_{i=0}^{m-1} \log \hat{p}_{\text{model}}(\mathbf{y}^{(i)} | \mathbf{x}^{(i)}, \theta) \right). \tag{21}$$

Cost function - maximum likelihood

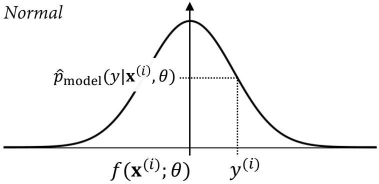


Optimal parameter values for Gaussian error distribution,
$\hat{p}_{\text{model}} \sim \mathcal{N}\left\{\mathbf{y}; f(\mathbf{x};\theta), \sigma\mathbf{I}\right\}$:

$$\hat{\theta} = \underset{\theta}{\operatorname{argmin}} \left(-\frac{1}{m} \sum_{i=0}^{m-1} \log \hat{p}_{\text{model}}\left(\mathbf{y}^{(i)} | \mathbf{x}^{(i)}, \theta\right) \right), \tag{22}$$

$$\hat{\theta} = \underset{\theta}{\operatorname{argmin}} \left(-\frac{1}{m} \sum_{i=0}^{m-1} \log e^{-\frac{1}{2\sigma^2}\left[\mathbf{y}^{(i)} - f(\mathbf{x}^{(i)};\theta)\right]\left[\mathbf{y}^{(i)} - f(\mathbf{x}^{(i)};\theta)\right]^T} \right), \tag{23}$$

Cost function - maximum likelihood

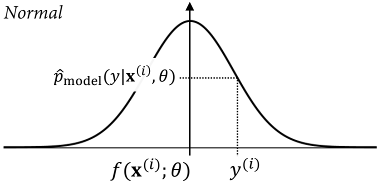


Optimal parameter values for Gaussian error distribution,
$\hat{p}_{\text{model}} \sim \mathcal{N}\left\{\mathbf{y}; f(\mathbf{x}; \theta), \sigma\mathbf{I}\right\}$:

$$\hat{\theta} = \underset{\theta}{\operatorname{argmin}} \left(-\frac{1}{m} \sum_{i=0}^{m-1} \log e^{-\frac{1}{2\sigma^2}\left[\mathbf{y}^{(i)} - f(\mathbf{x}^{(i)};\theta)\right]\left[\mathbf{y}^{(i)} - f(\mathbf{x}^{(i)};\theta)\right]^T} \right), \tag{24}$$

$$\hat{\theta} = \underset{\theta}{\operatorname{argmin}} \left(\frac{1}{m} \sum_{i=0}^{m-1} \left[\mathbf{y}^{(i)} - f(\mathbf{x}^{(i)}; \theta)\right] \left[\mathbf{y}^{(i)} - f(\mathbf{x}^{(i)}; \theta)\right]^T \right), \tag{25}$$

Cost function - maximum likelihood

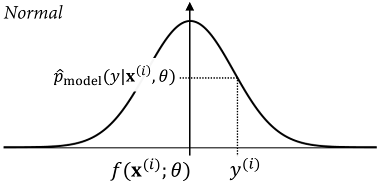


Optimal parameter values for Gaussian error distribution, $\hat{p}_{\text{model}} \sim \mathcal{N}\left\{\mathbf{y}; f(\mathbf{x};\theta), \sigma\mathbf{I}\right\}$:

$$\hat{\theta} = \underset{\theta}{\operatorname{argmin}} \left(\frac{1}{m} \sum_{i=0}^{m-1} \left[\mathbf{y}^{(i)} - f(\mathbf{x}^{(i)};\theta)\right] \left[\mathbf{y}^{(i)} - f(\mathbf{x}^{(i)};\theta)\right]^T \right), \tag{26}$$

$$\hat{\theta} = \underset{\theta}{\operatorname{argmin}} \frac{1}{m} \sum_{i=0}^{m-1} \left\| \mathbf{y}^{(i)} - f(\mathbf{x}^{(i)};\theta) \right\|_2^2, \tag{27}$$

Cost function - maximum likelihood

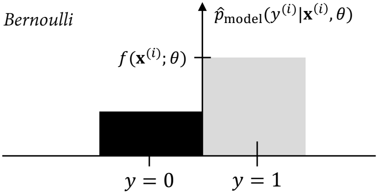


Optimal parameter values for Gaussian error distribution, $\hat{p}_{\text{model}} \sim \mathcal{N}\{\mathbf{y}; f(\mathbf{x};\theta), \sigma\mathbf{I}\}$:

$$\hat{\theta} = \underset{\theta}{\operatorname{argmin}} \frac{1}{m}\sum_{i=0}^{m-1} \left\|\mathbf{y}^{(i)} - f(\mathbf{x}^{(i)};\theta)\right\|_2^2, \tag{28}$$

$$y^{(i)} \in \mathbb{R}^1 \Rightarrow \hat{\theta} = \underset{\theta}{\operatorname{argmin}} \underbrace{\frac{1}{m}\sum_{i=0}^{m-1} \left(y^{(i)} - f(\mathbf{x}^{(i)};\theta)\right)^2}_{J(\theta)=\text{mean squared error}}, \tag{29}$$

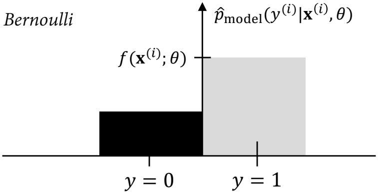
Cost function - maximum likelihood



Optimal parameter values for Bernoulli error distribution,

$$\hat{p}_{\text{model}}(y^{(i)}|\mathbf{x}^{(i)},\theta) = \begin{cases} p & \text{for } y^{(i)} = 1 \\ 1-p & \text{for } y^{(i)} = 0 \end{cases}$$

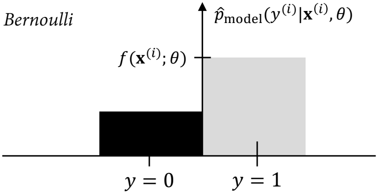
Cost function - maximum likelihood



Optimal parameter values for Bernoulli error distribution,

$$\hat{p}_{\text{model}}(y^{(i)}|\mathbf{x}^{(i)}, \theta) = p^{y^{(i)}}(1-p)^{\left(1-y^{(i)}\right)} \text{ for } y^{(i)} \in \{0, 1\}$$

Cost function - maximum likelihood

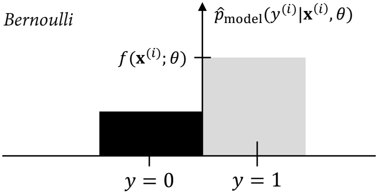


Optimal parameter values for Bernoulli error distribution,

$$\hat{p}_{\text{model}}(y^{(i)}|\mathbf{x}^{(i)},\theta) = p^{y^{(i)}}(1-p)^{\left(1-y^{(i)}\right)} \text{ for } y^{(i)} \in \{0,1\}$$

"The probability of observing $y^{(i)}=1$ follows a Bernoulli distribution with the model prediction $f(\mathbf{x}^{(i)};\theta)$ determining its probability p."

Cost function - maximum likelihood

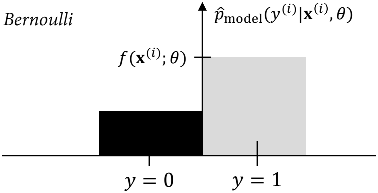


Optimal parameter values for Bernoulli error distribution,

$$\hat{p}_{\text{model}}(y^{(i)}|\mathbf{x}^{(i)},\theta) = p^{y^{(i)}}(1-p)^{\left(1-y^{(i)}\right)} \text{ for } y^{(i)} \in \{0,1\}$$

$$\hat{\theta} = \underset{\theta}{\operatorname{argmin}} \left(-\frac{1}{m} \sum_{i=0}^{m-1} \log \hat{p}_{\text{model}}(y^{(i)}|\mathbf{x}^{(i)},\theta) \right). \tag{30}$$

Cost function - maximum likelihood



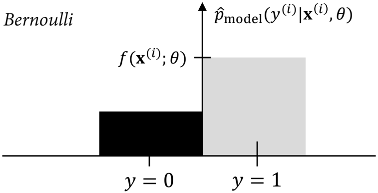
Optimal parameter values for Bernoulli error distribution,

$$\hat{p}_{\text{model}}(y^{(i)}|\mathbf{x}^{(i)},\theta) = p^{y^{(i)}}(1-p)^{\left(1-y^{(i)}\right)} \text{ for } y^{(i)} \in \{0,1\}$$

$$\hat{\theta} = \underset{\theta}{\operatorname{argmin}} \left(- \sum_{i=0}^{m-1} \log \hat{p}_{\text{model}}(y^{(i)}|\mathbf{x}^{(i)},\theta) \right). \tag{31}$$

$$\hat{\theta} = \underset{\theta}{\operatorname{argmin}} \left(- \sum_{i=0}^{m-1} \log \left(\left(p^{(i)}\right)^{y^{(i)}} \left(1-p^{(i)}\right)^{\left(1-y^{(i)}\right)} \right) \right), \tag{32}$$

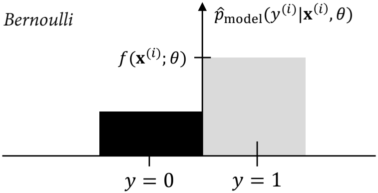
Cost function - maximum likelihood



Optimal parameter values for Bernoulli error distribution,

$$\hat{p}_{\text{model}}(y^{(i)}|\mathbf{x}^{(i)},\theta) = p^{y^{(i)}}(1-p)^{\left(1-y^{(i)}\right)} \text{ for } y^{(i)} \in \{0,1\}$$

$$\Rightarrow \hat{\theta} = \underset{\theta}{\operatorname{argmin}} \left(-\sum_{i=0}^{m-1} y^{(i)} \log(p^{(i)}) + \left(1-y^{(i)}\right) \log\left(1-p^{(i)}\right) \right), \quad (33)$$



Optimal parameter values for Bernoulli error distribution,

$$\hat{p}_{\text{model}}(y^{(i)}|\mathbf{x}^{(i)},\theta) = p^{y^{(i)}}(1-p)^{\left(1-y^{(i)}\right)} \text{ for } y^{(i)} \in \{0,1\}$$

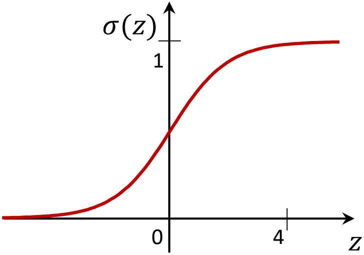
$$\Rightarrow \hat{\theta} = \underset{\theta}{\text{argmin}} \underbrace{\left(-\sum_{i=0}^{m-1} y^{(i)}\log(p^{(i)}) + \left(1-y^{(i)}\right)\log\left(1-p^{(i)}\right)\right)}_{J(\theta)=\text{Binary cross entropy}}, \quad (34)$$

Model predictions of probabilities

We require model that maps inputs to probabilities p.

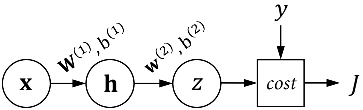
Consider a nonlinearity that squeezes all model output values between 0 and 1:

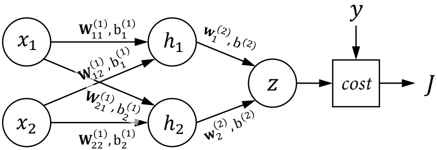
$$p = \sigma\left(f(z)\right) = \frac{1}{1+e^{-z}}. \tag{35}$$

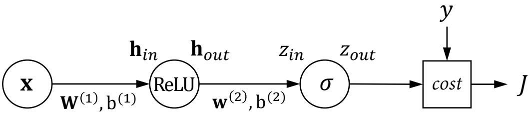


Nonlinear binary classification model

The full binary classification model becomes:







binary cross-entropy:

$$J = \frac{1}{m}\sum_{i=0}^{m-1} y_i \log z_{out} + (1 - y_i)\log(1 - z_{out})$$

Parameter optimization

Finding optimal parameter values given the cost

⇒ Gradient-based learning (e.g. Gradient-Descent algorithm):

- $\mathbf{W}^{(1)}_{n+1} = \mathbf{W}^{(1)}_{n} - \mu\partial_{\mathbf{W}^{(1)}} J(\theta_n)$
- $\mathbf{w}^{(2)}_{n+1} = \mathbf{w}^{(2)}_{n} - \mu\partial_{\mathbf{w}^{(2)}} J(\theta_n)$
- $\mathbf{b}^{(1)}_{n+1} = \mathbf{b}^{(1)}_{n} - \mu\partial_{\mathbf{b}^{(1)}} J(\theta_n)$
- $b^{(2)}_{n+1} = b^{(2)}_{n} - \mu\partial_{b^{(2)}} J(\theta)$

With learning rate μ.

Parameter optimization

<u>Finding optimal parameter values given the cost</u>

⇒ Gradient-based learning (e.g. Gradient-Descent algorithm):

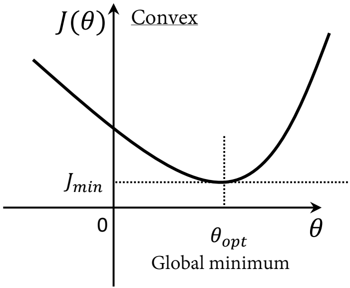
- Linear models; Convex → guaranteed global convergence.
- Nonlinear models; Often non-convex → no global convergence guarantees.

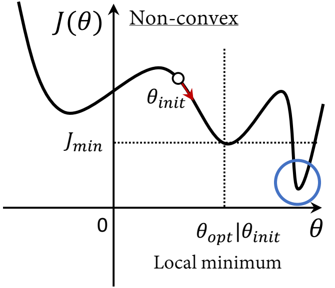
Parameter optimization

Finding optimal parameter values given the cost

$\Rightarrow$ Gradient-based learning (e.g. Gradient-Descent algorithm):

- Linear models; Convex $\rightarrow$ guaranteed global convergence.
- Nonlinear models; Often non-convex $\rightarrow$ no global convergence guarantees.





TU/e Technische Universiteit Eindhoven University of Technology