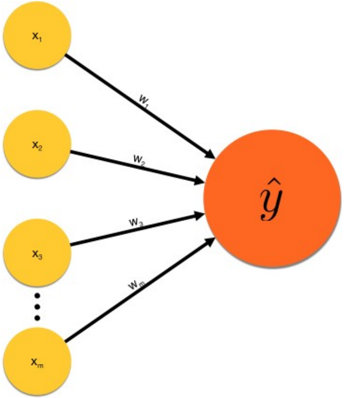
Neural Networks and Deep Learning: The Formal Neuron

Nicolas Thome

Conservatoire National des Arts et Métiers (Cnam)
Département Informatique

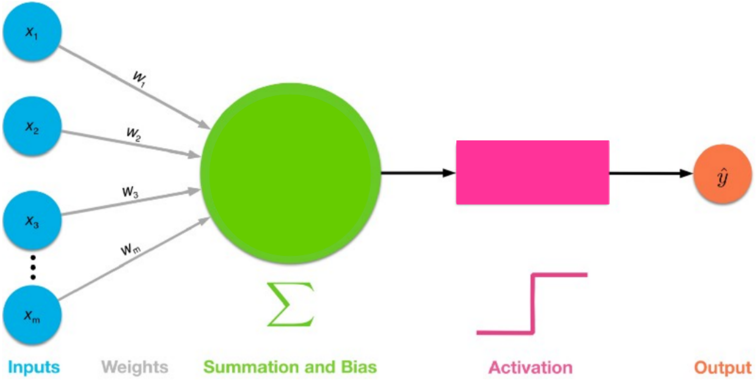
The Formal Neuron: 1943 [McCulloch and Pitts, 1943]

- Basis of Neural Networks
- Input: vector $\mathbf{x} \in \mathbb{R}^m$, *i.e.* $\mathbf{x} = \{x_i\}_{i \in \{1,2,...,m\}}$
- Neuron output $\hat{y} \in \mathbb{R}$: scalar



The Formal Neuron: 1943 [McCulloch and Pitts, 1943]

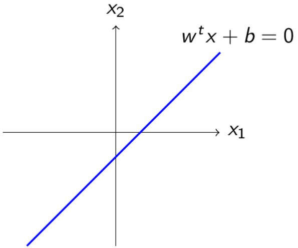
- Mapping from $\mathbf{x}$ to $\hat{y}$:
  1. Linear (affine) mapping: $s = \mathbf{w}^\top \mathbf{x} + b$
  2. Non-linear activation function: f: $\hat{y} = f(s)$



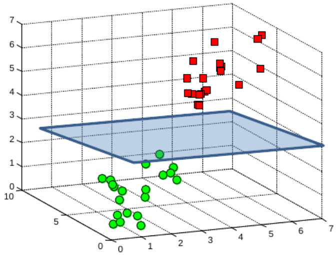
The Formal Neuron: Linear Mapping

- Linear (affine) mapping:
  $$s = \mathbf{w}^\top \mathbf{x} + b = \sum_{i=1}^{m} w_i x_i + b$$
  - $\mathbf{w}$: normal vector to an hyperplane in $\mathbb{R}^m \Rightarrow$ **linear boundary**
  - b bias, shift the hyperplane position

2D hyperplane
line

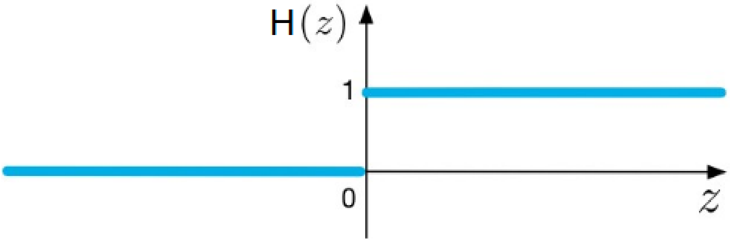


3D hyperplane:
plane

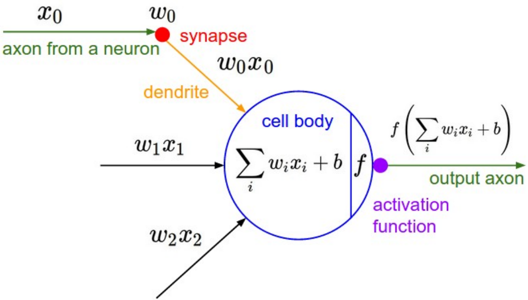


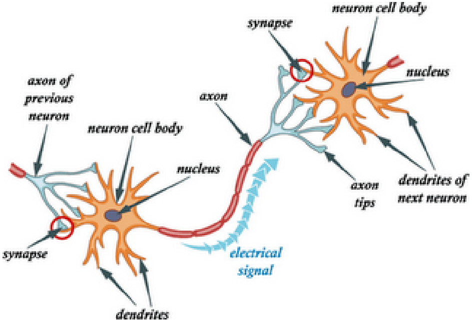
The Formal Neuron: Activation Function

- $\hat{y} = f(\mathbf{w}^\top \mathbf{x} + b)$, f activation function
  - Popular f choices: step, sigmoid, tanh
- Step (Heaviside) function: $H(z) = \begin{cases} 1 & \text{if } z \geq 0 \\ 0 & \text{otherwise} \end{cases}$



Step function: Connection to Biological Neurons

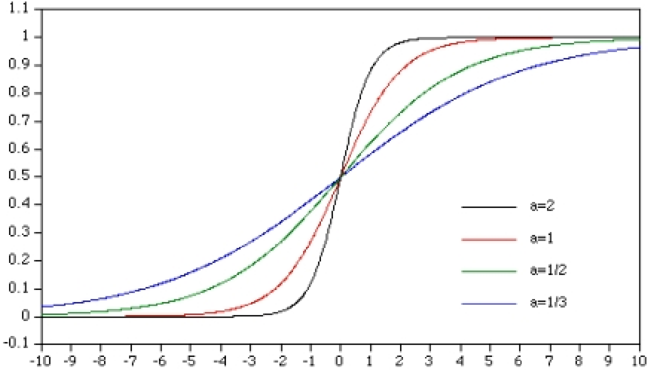




- Formal neuron, step activation H: $\hat{y} = H(\mathbf{w}^\top \mathbf{x} + b)$
  - $\hat{y} = 1$ (activated) $\Leftrightarrow \mathbf{w}^\top \mathbf{x} \geq -b$
  - $\hat{y} = 0$ (unactivated) $\Leftrightarrow \mathbf{w}^\top \mathbf{x} < -b$
- **Biological Neurons:** output activated
  $\Leftrightarrow$ input weighted by synaptic weight $\geq$ threshold

Sigmoid Activation Function

- Neuron output $\hat{y} = f(\mathbf{w}^\top \mathbf{x} + b)$, f activation function
- Sigmoid: $\sigma(z) = (1 + e^{-az})^{-1}$

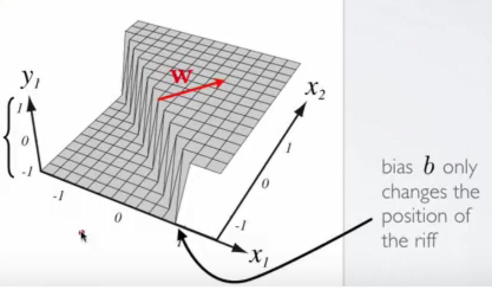


- $a \uparrow$: more similar to step function (step: $a \to \infty$)
- Sigmoid: linear and saturating regimes

The Formal neuron: Application to Binary Classification

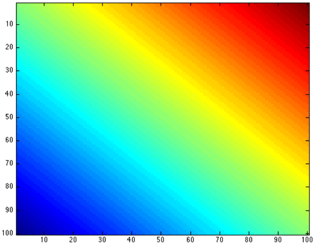
- Binary Classification: label input $\mathbf{x}$ as belonging to class 1 or 0
- Neuron output with sigmoid: $\hat{y} = \frac{1}{1+e^{-a(\mathbf{w}^\top \mathbf{x}+b)}}$
- Sigmoid: probabilistic interpretation $\Rightarrow \hat{y} \sim P(1/\mathbf{x})$ ($\sim$ with tanh)
  - Input $\mathbf{x}$ classified as 1 if $P(1/\mathbf{x}) > 0.5 \Leftrightarrow \mathbf{w}^\top \mathbf{x} + b > 0$
  - Input $\mathbf{x}$ classified as 0 if $P(1/\mathbf{x}) < 0.5 \Leftrightarrow \mathbf{w}^\top \mathbf{x} + b < 0$

  $\Rightarrow$ sign$(\mathbf{w}^\top \mathbf{x} + b)$: linear boundary decision in input space !



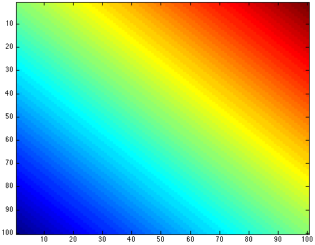
The Formal neuron: Toy Example for Binary Classification

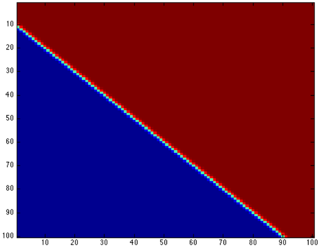
- 2d example: $m = 2$,
  $\mathbf{x} = \{x_1, x_2\} \in [-5; 5] \times [-5; 5]$
- Linear mapping: $\mathbf{w} = [1; 1]$ and $b = -2$
- Result of linear mapping : $s = \mathbf{w}^\top \mathbf{x} + b$

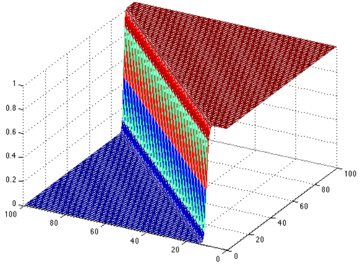


The Formal neuron: Toy Example for Binary Classification

- 2d example: $m = 2$,
  $\mathbf{x} = \{x_1, x_2\} \in [-5; 5] \times [-5; 5]$
- Linear mapping: $\mathbf{w} = [1; 1]$ and $b = -2$
- Result of linear mapping : $s = \mathbf{w}^\top \mathbf{x} + b$
- Sigmoid activation function:
  $\hat{y} = \left(1 + e^{-a(\mathbf{w}^\top \mathbf{x} + b)}\right)^{-1}$, $a = 10$

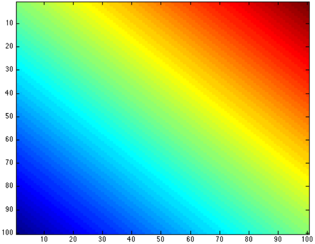


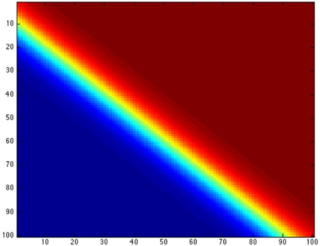


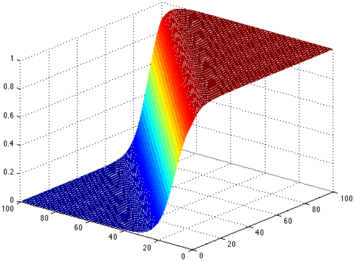


The Formal neuron: Toy Example for Binary Classification

- 2d example: $m = 2$,
  $\mathbf{x} = \{x_1, x_2\} \in [-5; 5] \times [-5; 5]$
- Linear mapping: $\mathbf{w} = [1; 1]$ and $b = -2$
- Result of linear mapping : $s = \mathbf{w}^\top \mathbf{x} + b$
- Sigmoid activation function:
  $\hat{y} = \left(1 + e^{-a(\mathbf{w}^\top \mathbf{x} + b)}\right)^{-1}$, $a = 1$

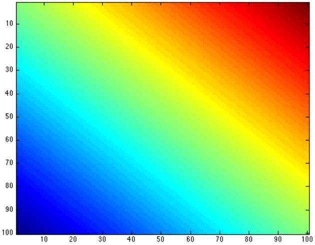


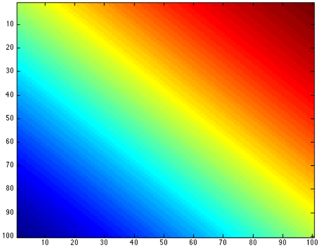


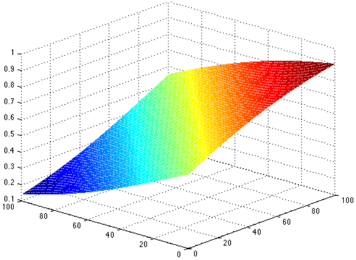


The Formal neuron: Toy Example for Binary Classification

- 2d example: $m = 2$,
  $\mathbf{x} = \{x_1, x_2\} \in [-5; 5] \times [-5; 5]$
- Linear mapping: $\mathbf{w} = [1; 1]$ and $b = -2$
- Result of linear mapping : $s = \mathbf{w}^\top \mathbf{x} + b$
- Sigmoid activation function:
  $\hat{y} = \left(1 + e^{-a(\mathbf{w}^\top \mathbf{x} + b)}\right)^{-1}$, $a = 0.1$

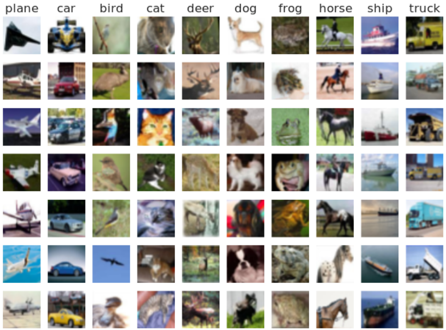


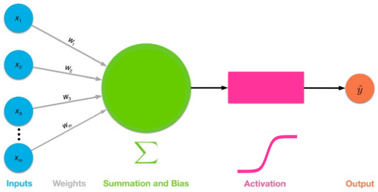




From Formal Neuron to Neural Networks

- Formal Neuron:
  1. A single scalar output
  2. Linear decision boundary for binary classification
- Single scalar output: limited for several tasks
  - Ex: multi-class classification, *e.g.* MNIST or CIFAR





References I

McCulloch, W. S. and Pitts, W. (1943).
A logical calculus of the ideas immanent in nervous activity.
The bulletin of mathematical biophysics, 5(4):115–133.