Neural networks

Feedforward neural network - multilayer neural network
ARTIFICIAL NEURON

**Topics:** capacity of single neuron

- Can't solve non linearly separable problems...

\[
\text{XOR}(x_1, x_2)
\]

\[
\text{AND}(x_1, \overline{x_2})
\]

\[
\text{AND}(\overline{x_1}, x_2)
\]

- ... unless the input is transformed in a better representation
**Topics:** single hidden layer neural network

- **Hidden layer pre-activation:**
  \[
  a(x) = b^{(1)} + W^{(1)}x \\
  (a(x)_i = b^{(1)}_i + \sum_j W^{(1)}_{i,j} x_j)
  \]

- **Hidden layer activation:**
  \[
  h(x) = g(a(x))
  \]

- **Output layer activation:**
  \[
  f(x) = o\left(b^{(2)} + w^{(2)^T}h^{(1)}x\right)
  \]
Topics: softmax activation function

• For multi-class classification:
  ‣ we need multiple outputs (1 output per class)
  ‣ we would like to estimate the conditional probability \( p(y = c | x) \)

• We use the softmax activation function at the output:

\[
o(a) = \text{softmax}(a) = \left[ \frac{\exp(a_1)}{\sum_c \exp(a_c)} \ldots \frac{\exp(a_C)}{\sum_c \exp(a_c)} \right]^T
\]

  ‣ strictly positive
  ‣ sums to one

• Predicted class is the one with highest estimated probability
Topics: multilayer neural network

- Could have $L$ hidden layers:
  - layer pre-activation for $k > 0$: $h^{(0)}(x) = x$
    \[
    a^{(k)}(x) = b^{(k)} + W^{(k)}h^{(k-1)}(x)
    \]
  - hidden layer activation ($k$ from 1 to $L$):
    \[
    h^{(k)}(x) = g(a^{(k)}(x))
    \]
  - output layer activation ($k = L+1$):
    \[
    h^{(L+1)}(x) = o(a^{(L+1)}(x)) = f(x)
    \]