Multilayer Networks

Machine Learning: Jordan Boyd-Graber
University of Maryland

INTRODUCTION
Deep Learning was once known as “Neural Networks”
But it came back . . .

- More data
- Better tricks (regularization)
- Faster computers
And companies are investing ...
And companies are investing …

‘Chinese Google’ Opens Artificial-Intelligence Lab in Silicon Valley

BY DANIELA HERNANDEZ  04.12.13  |  6:30 AM  |  PERMALINK
And companies are investing …

Facebook’s ‘Deep Learning’ Guru Reveals the Future of AI

BY CADE METZ 12.12.13 | 6:30 AM | PERMALINK
Map inputs to output

\[
\begin{align*}
\text{Input} & \quad \mathbf{x} = (x_1, \ldots, x_d) \\
\text{Output} & \quad f = \sum_i W_i x_i + b \\
\end{align*}
\]

Activation: \[ f(z) = \frac{1}{1 + \exp(-z)} \]
Map inputs to output

Input

Vector $x_1 \ldots x_d$

inputs encoded as real numbers
Map inputs to output

Input
Vector $x_1 \ldots x_d$

Output

$$f \left( \sum_i W_i x_i + b \right)$$

multiply inputs by
Map inputs to output

Input
Vector $x_1 \ldots x_d$

Output

$$f \left( \sum_i W_i x_i + b \right)$$

add bias
Map inputs to output

Input
Vector \( x_1 \ldots x_d \)

Output
\[ f\left( \sum_i W_i x_i + b \right) \]

Activation
\[ f(z) \equiv \frac{1}{1 + \exp(-z)} \]

pass through nonlinear sigmoid
Why is it called activation?
In the shallow end

- This is still logistic regression
- Engineering features $x$ is difficult (and requires expertise)
- Can we learn how to represent inputs into final decision?
Better name: non-linearity

- **Logistic / Sigmoid**
  \[ f(x) = \frac{1}{1 + e^{-x}} \]  

- **tanh**
  \[ f(x) = \tanh(x) = \frac{2}{1 + e^{-2x}} - 1 \]  

- **ReLU**
  \[ f(x) = \begin{cases} 
  0 & \text{for } x < 0 \\
  x & \text{for } x \geq 0 
\end{cases} \]  

- **SoftPlus**: \[ f(x) = \ln(1 + e^x) \]