Neural Networks

Jitendra Malik

CS 189
Key facts

• The human brain has ~ 100 billion neurons (cells)
• Most neurons input signals via the **dendrites** and output signals via the **axon**
• At the tip of an axon’s branches, there are **axon terminals**, where the neuron can transmit a signal across the **synapse** to another cell
• Typically the signal flow along an axon is in the form of voltage spikes, with more spikes per second indicating a stronger output
How Neurons Communicate
Mathematical Abstraction

\[ y_i = g \left( \sum_j w_{ij} x_j \right) \]
Single layer neural network

\[ g(z) = \frac{1}{1 + e^{-z}} \]

\[ V_1 \]

\[ V_2 = g \left( \sum_{k} w_{jk} x_k \right) \]
Two layer neural network
Two layer neural network

\[ X_1 \rightarrow X_2 \rightarrow V_1 \rightarrow V_2 \rightarrow V_3 \rightarrow O_1 \]

\[ W_{11} \quad W_{12} \]

\[ X_3 \rightarrow X_4 \rightarrow X_5 \rightarrow V_j \rightarrow O_i \]

\[ W_{23} \quad W_{35} \quad W_{jk} \]
\[ V_j = g \left( \sum_k w_{jk} x_k \right); \quad O_i = g \left( \sum w_{ij} V_j \right) \]
\[ O_i = g \left( \sum_j W_{ij} g \left( \sum_k W_{jk} x_k \right) \right) \]
Recall from multivariable calculus that the *gradient* of $f$ is the vector

$$\nabla f(x) = \begin{bmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \\ \vdots \\ \frac{\partial f}{\partial x_d} \end{bmatrix}.$$ 

The gradient is the direction which leads to a maximal increase of $f$. Similarly, the negative gradient is the direction of *steepest descent*. Gradient descent uses this fact to construct an algorithm: at every step, compute the gradient and follow that direction to minimize $f$. 
Training a neural network

Goal: Find \( W \) such that \( O_i \) is as close as possible to \( y_i \) (desired output)

Approach: Define loss function \( L(W) \)

1. Compute \( \nabla W L \)
2. \( W_{\text{new}} \leftarrow W_{\text{old}} - \eta \nabla W L \)
Training a single layer neural network

- For binary classification, a good choice of loss function is the cross entropy. For regression, use squared error

\[ L = - \sum_{\text{input data}} \left( y_i \ln O_i + (1-y_i) \ln(1-O_i) \right) \]

- We model the activation function \( g \) as a sigmoid

\[ g(z) = \frac{1}{1 + \exp(-z)} \]

- Finding \( w \) reduces to logistic regression!

We can use STOCHASTIC GRADIENT DESCENT.