Introduction to Machine Learning

CSE474/574: Lecture 8

Varun Chandola <chandola@buffalo.edu>

11 Feb 2015

Outline

Outline

Recap - Perceptrons

Training Rule for Gradient Descent

$$\vec{\mathbf{w}} = \vec{\mathbf{w}} - \eta \nabla E(\vec{\mathbf{w}})$$

For each weight component:

$$w_i = w_i - \eta \frac{\partial E}{\partial w_i}$$

Issues with Gradient Descent

- Slow convergence
- Stuck in local minima

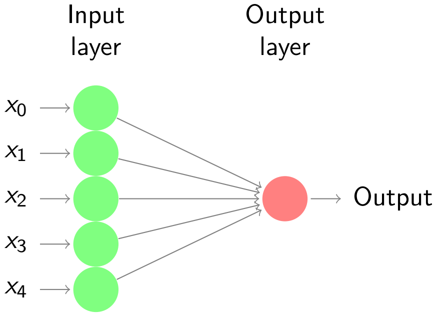
Stochastic Gradient Descent [?]

- **Update weights after every training example.**
- For sufficiently small η, closely approximates Gradient Descent.

Gradient Descent	**Stochastic Gradient Descent**
Weights updated after summing error over all examples	Weights updated after examining each example
More computations per weight update step	Significantly lesser computations
Risk of local minima	Avoids local minima

Extending Perceptrons

- Questions?
  - Why not work with thresholded perceptron?
    - Not differentiable
  - How to learn non-linear surfaces?
  - How to generalize to multiple outputs, numeric output?



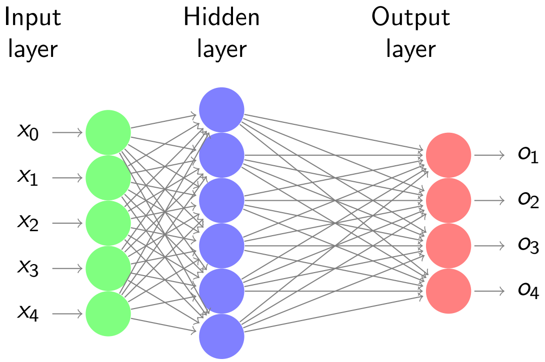
Outline

Generalizing to Multiple Labels

- Distinguishing between multiple categories
- *Solution:* Add another layer - **Multi Layer Neural Networks**

Generalizing to Multiple Labels

- Distinguishing between multiple categories
- *Solution:* Add another layer - **Multi Layer Neural Networks**



ALVINN - Autonomous Land Vehicle In a Neural Network

What Threshold Unit to Use?

- ~~Linear Unit~~
- ~~Perceptron Unit~~
- Sigmoid Unit
  - Smooth, differentiable threshold function

$$\sigma(net) = \frac{1}{1 + e^{-net}}$$

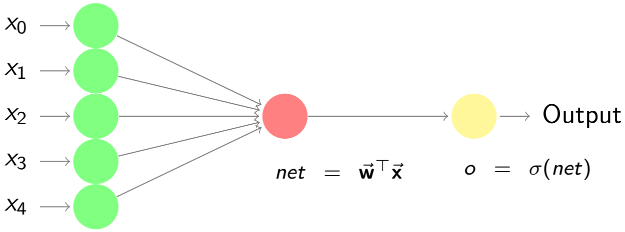
  - Non-linear output

What Threshold Unit to Use?

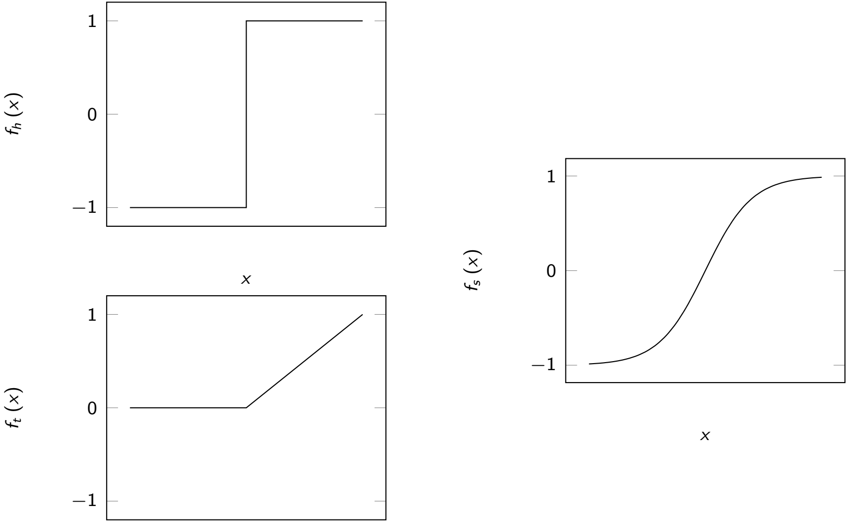
- ~~Linear Unit~~
- ~~Perceptron Unit~~
- Sigmoid Unit
  - Smooth, differentiable threshold function

$$\sigma(net) = \frac{1}{1 + e^{-net}}$$

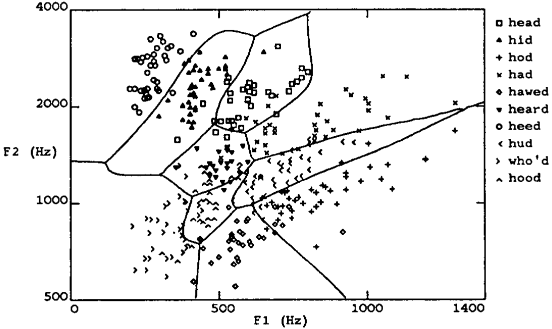
  - Non-linear output



Properties of Sigmoid Function

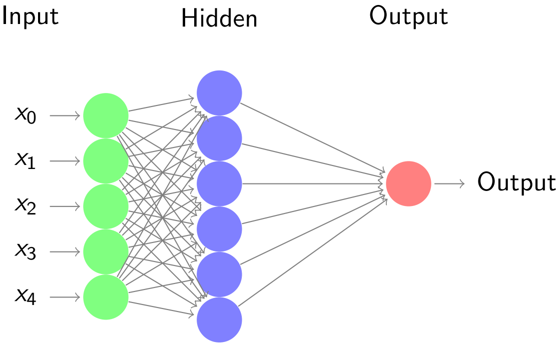


Motivation for Using Non-linear Surfaces



Outline

Feed Forward Neural Networks



- d input nodes
- m hidden nodes
- k output nodes

- At hidden nodes: $\vec{\mathbf{w}}_{\mathbf{j}}, 1 \leq j \leq m$, $\vec{\mathbf{w}}_{\mathbf{j}} \in \Re^d$
- At output nodes: $\vec{\mathbf{w}}_{\mathbf{j}}, 1 \leq j \leq k$, $\vec{\mathbf{w}}_{\mathbf{j}} \in \Re^m$

References