CS 461: Machine Learning
Lecture 4

Dr. Kiri Wagstaff
wkiri@wkiri.com

Plan for Today

- Solution to HW 2
- Support Vector Machines
- Neural Networks
  - Perceptrons
  - Multilayer Perceptrons

Review from Lecture 3

- Decision trees
  - Regression trees, pruning, extracting rules
- Evaluation
  - Comparing two classifiers: McNemar's test
- Support Vector Machines
  - Classification
    - Linear discriminants, maximum margin
    - Learning (optimization): gradient descent, QP

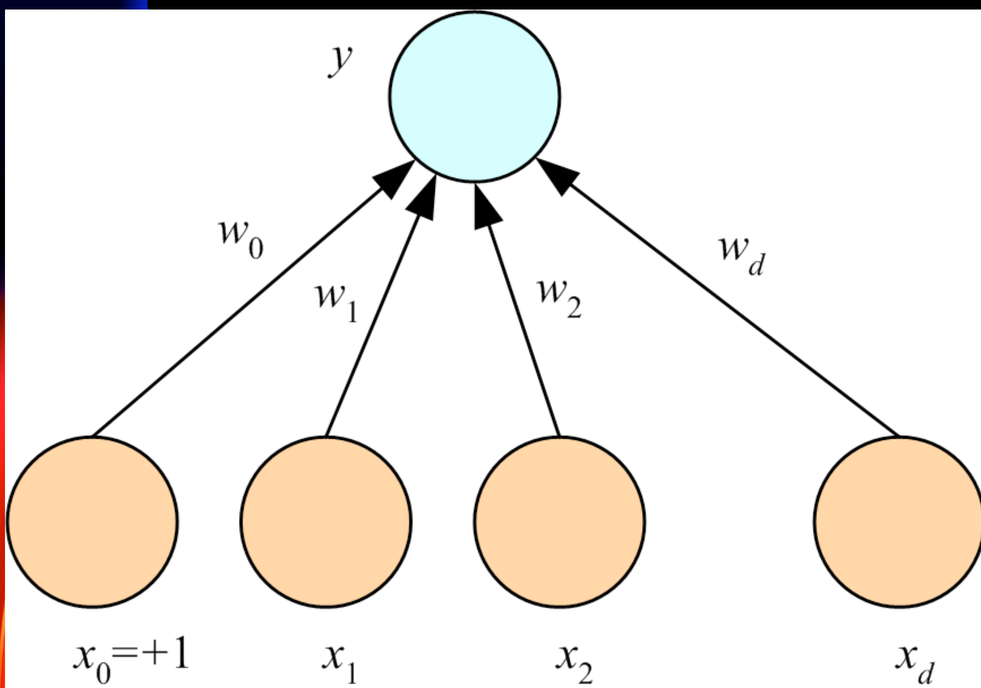
Neural Networks

Chapter 11

It Is Pitch Dark

Perceptron

Graphical



Math

$$y = \sum_{j=1}^{d} w_j x_j + w_0 = \boldsymbol{w}^T \boldsymbol{x}$$

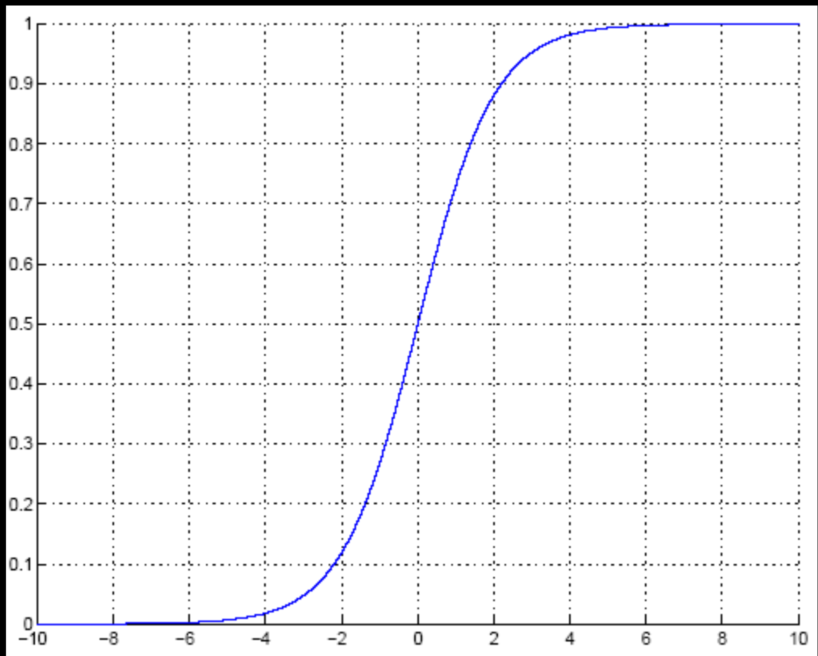
$$\boldsymbol{w} = [w_0, w_1, \ldots, w_d]^T$$

$$\boldsymbol{x} = [1, x_1, \ldots, x_d]^T$$

"Smooth" Output: Sigmoid Function

1. Calculate $g(\mathbf{x}) = \mathbf{w}^T\mathbf{x}$ and choose C_1 if $g(\mathbf{x}) > 0$, or

2. Calculate $y = \text{sigmoid}(\mathbf{w}^T\mathbf{x})$ and choose C_1 if $y > 0.5$

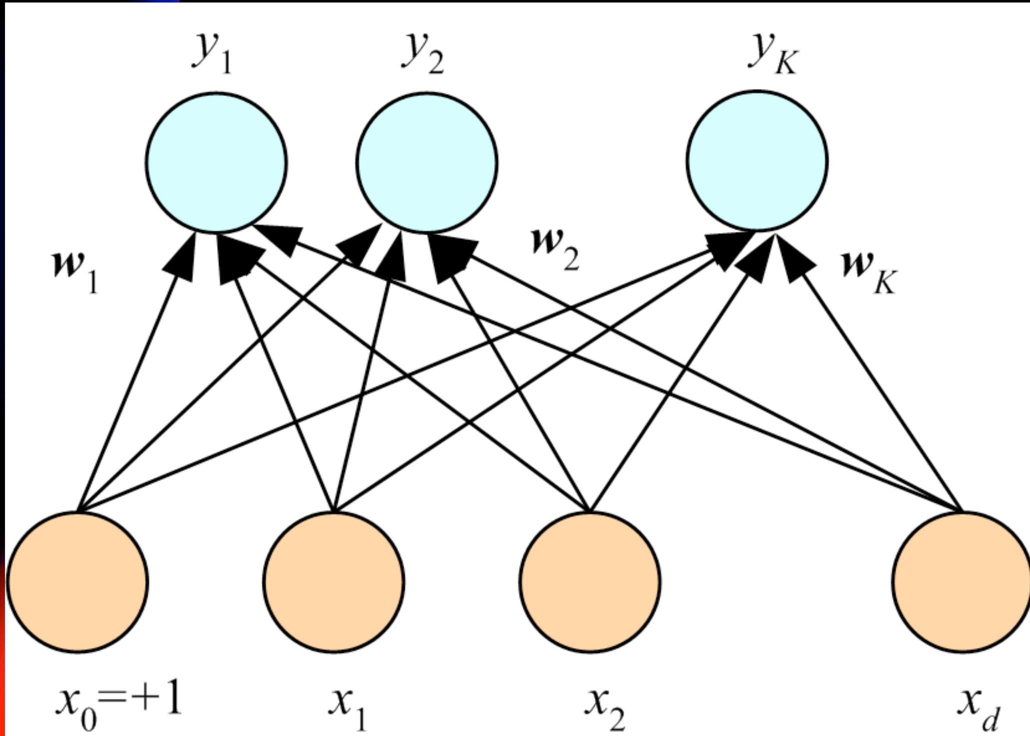


$$y = \text{sigmoid}(\mathbf{w}^T\mathbf{x}) = \frac{1}{1 + \exp\left[-\mathbf{w}^T\mathbf{x}\right]}$$

Why?

- Converts output to probability!
- Less "brittle" boundary

K outputs



Regression:

$$y_i = \sum_{j=1}^{d} w_{ij} x_j + w_{i0} = \boldsymbol{w}_i^T \boldsymbol{x}$$

$$\boldsymbol{y} = \mathbf{W}\boldsymbol{x}$$

Classification:

$$o_i = \boldsymbol{w}_i^T \boldsymbol{x}$$

$$y_i = \frac{\exp o_i}{\sum_k \exp o_k}$$

choose C_i

if $y_i = \max_k y_k$

Softmax

Training a Neural Network

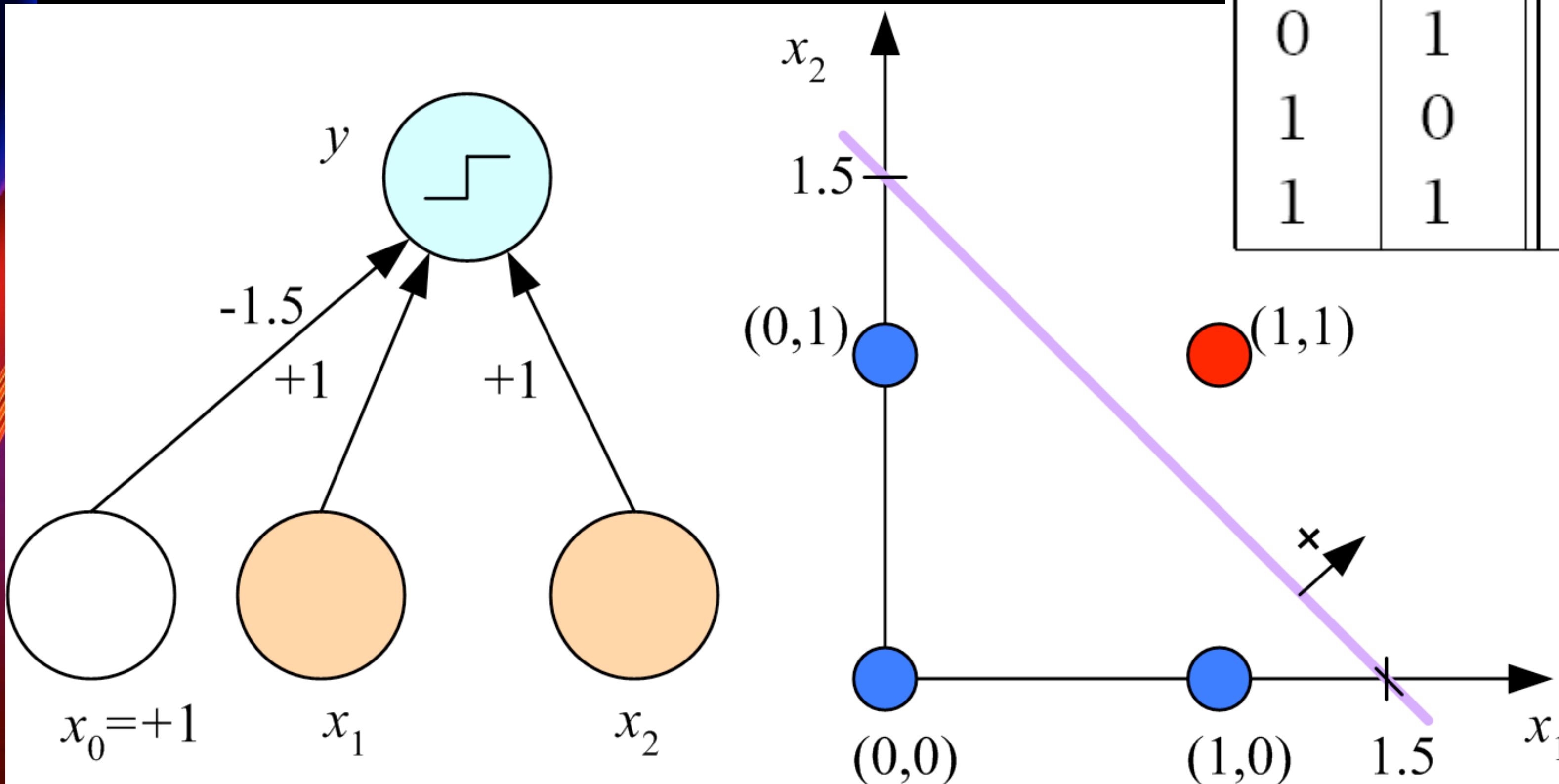
1. Randomly initialize weights
2. Update =
   Learning rate * (Desired - Actual) * Input

$$\Delta w_j^t = \eta\left(y^t - \hat{y}^t\right)x_j^t$$

Learning Boolean AND

$$\Delta w_j^t = \eta\left(y^t - \hat{y}^t\right)x_j^t$$

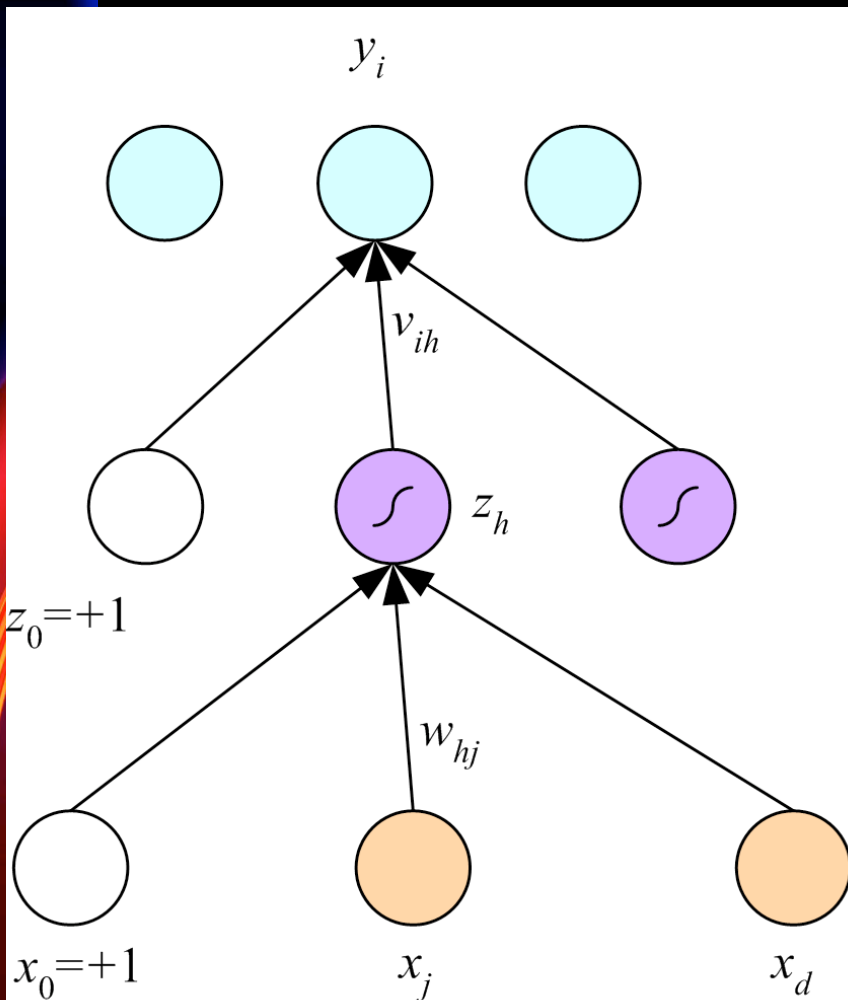
x_1	x_2	r
0	0	0
0	1	0
1	0	0
1	1	1



Perceptron demo

[Alpaydin 2004 © The MIT Press]

Multilayer Perceptrons = MLP = ANN



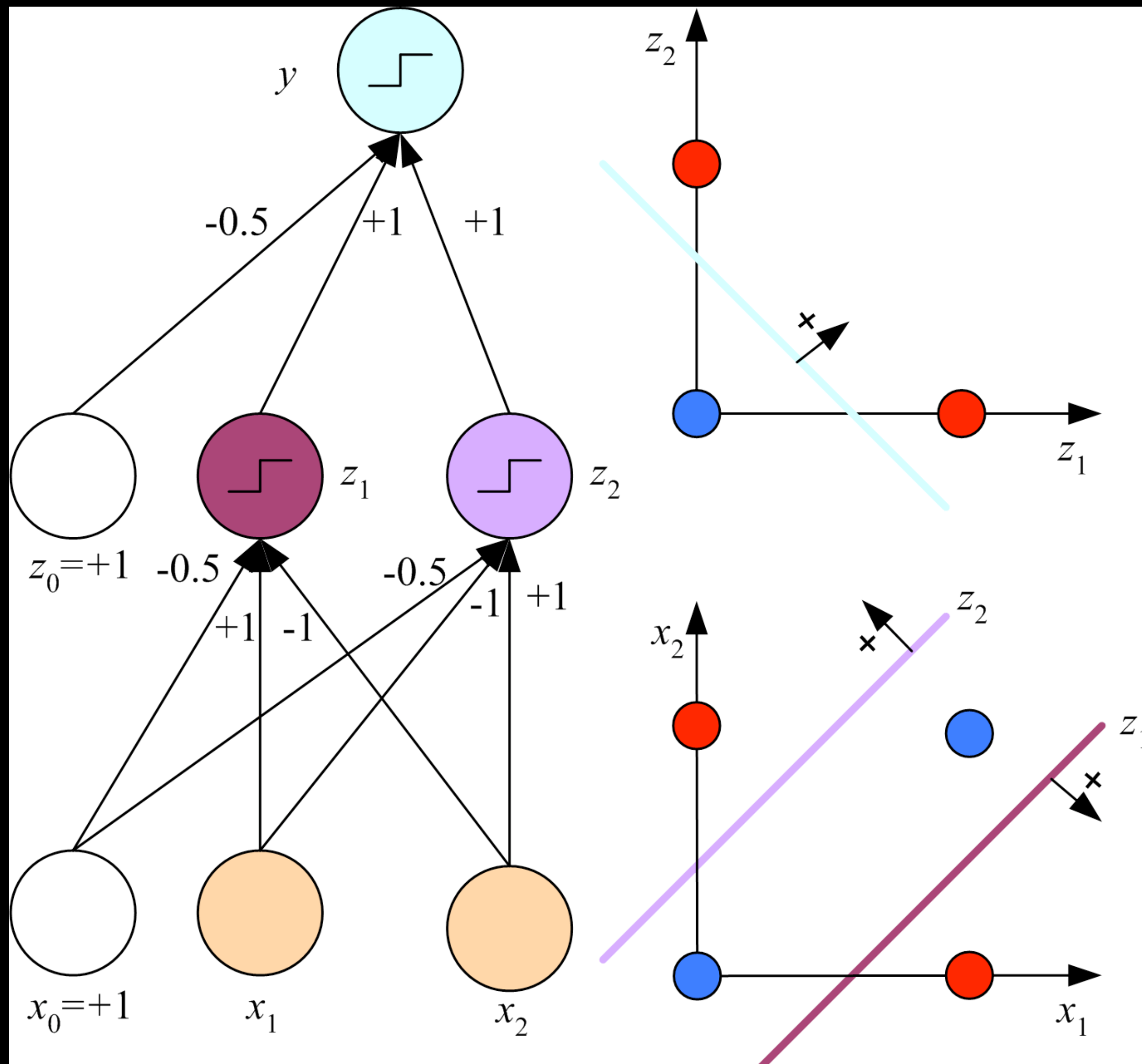
$$y_i = \mathbf{v}_i^T \mathbf{z} = \sum_{h=1}^{H} v_{ih} z_h + v_{i0}$$

$$z_h = \text{sigmoid}\left(\mathbf{w}_h^T \mathbf{x}\right)$$

$$= \frac{1}{1 + \exp\left[-\left(\sum_{j=1}^{d} w_{hj} x_j + w_{h0}\right)\right]}$$

[Alpaydin 2004 © The MIT Press]

x_1 XOR x_2 = (x_1 AND ~x_2) OR (~x_1 AND x_2)



Examples

- Digit Recognition
- Ball Balancing

ANN vs. SVM

- SVM with sigmoid kernel = 2-layer MLP
- Parameters
  - ANN: # hidden layers, # nodes
  - SVM: kernel, kernel params, C
- Optimization
  - ANN: local minimum (gradient descent)
  - SVM: global minimum (QP)
- Interpretability? About the same…
- So why SVMs?
  - Sparse solution, geometric interpretation, less likely to overfit data

Summary: Key Points for Today

- Support Vector Machines
- Neural Networks
  - Perceptrons
    - Sigmoid
    - Training by gradient descent
  - Multilayer Perceptrons
- ANN vs. SVM

Next Time

- Midterm Exam!
  - 9:10 – 10:40 a.m.
  - Open book, open notes (no computer)
  - Covers all material through today
- Neural Networks
  (read Ch. 11.1-11.8)
- Questions to answer from the reading
  - Posted on the website (calendar)
  - Three volunteers?