

Deep Learning Part 2

Other networks

Slides by:

Joseph E. Gonzalez,

jegonzal@berkeley.edu

Quick Logistics

- Please sign up to present
- Complete reading questions before lecture
- Skim the reading for the lecture you are presenting (later in semester).
 - Is there a better paper?

<http://bit.ly/aisys-sp19>

AI-Sys Syllabus Projects Grading

AI-Sys Spring 2019

- **When:** Mondays and Wednesdays from 9:30 to 11:00
- **Where:** Soda 405
- **Instructors:** Ion Stoica and Joseph E. Gonzalez
- **Announcements:** [Piazza](#)
- **Sign-up to Present:** [Google Spreadsheet](#)

Course Description

The recent success of AI has been in large part due in part to advances in hardware and software systems. These systems have enabled training increasingly complex models on ever larger datasets. In the process, these systems have also simplified model development, enabling the rapid growth in the machine learning community. These new hardware and software systems include a new generation of GPUs and hardware accelerators (e.g.,

Last Time

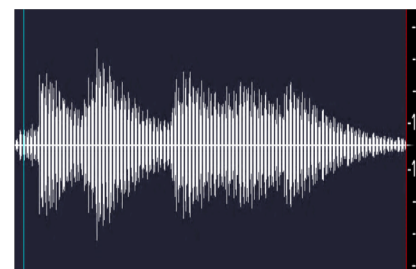
Machine Learning \approx Function Approximation

Object Recognition



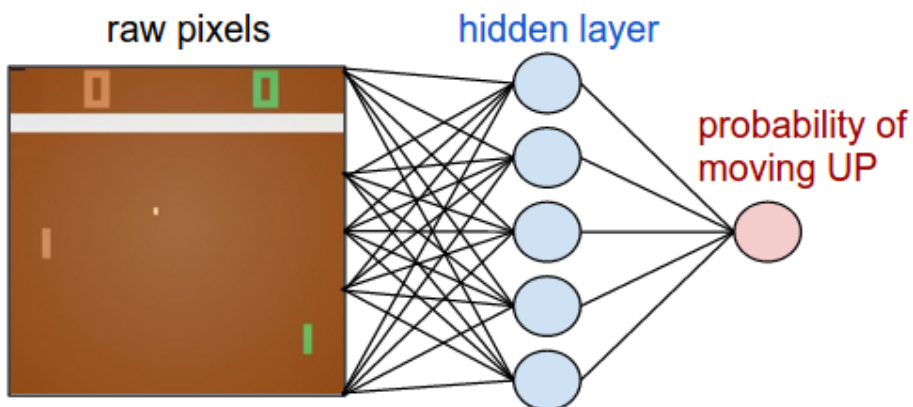
Label:*Cat*

Speech Recognition

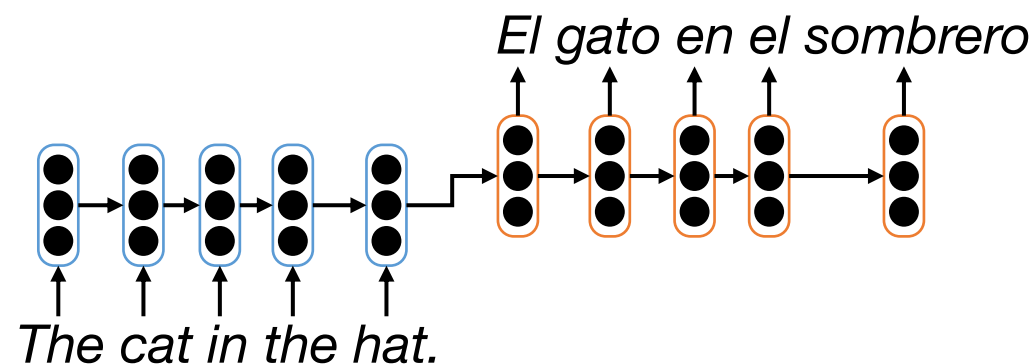


"The cat in the hat"

Robotic Control



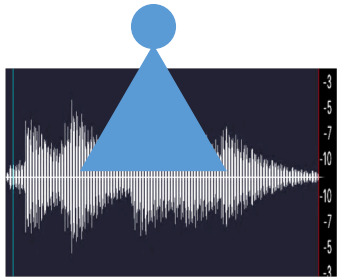
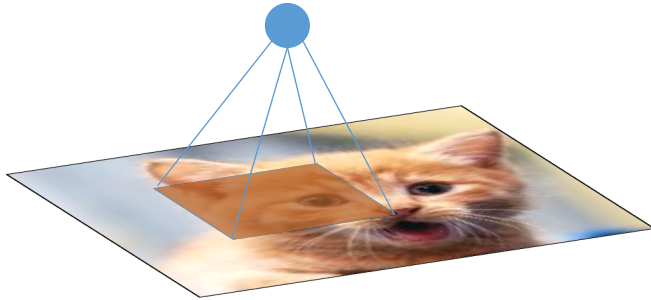
Machine Translation



Architectures for Different kinds of inputs

Convolutional Networks

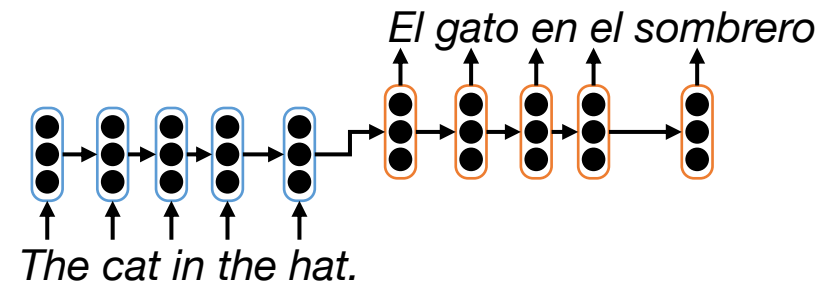
spatial reasoning tasks



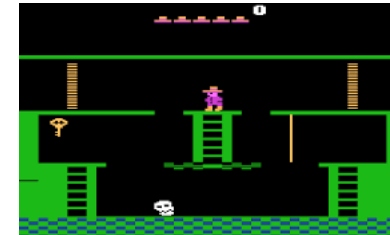
The quick brown fox...

Recurrent Networks

Sequential reasoning tasks



Reinforcement Learning

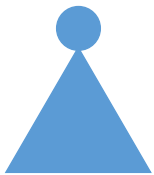


Speech recognition

Architectures for Different kinds of inputs

al Networks

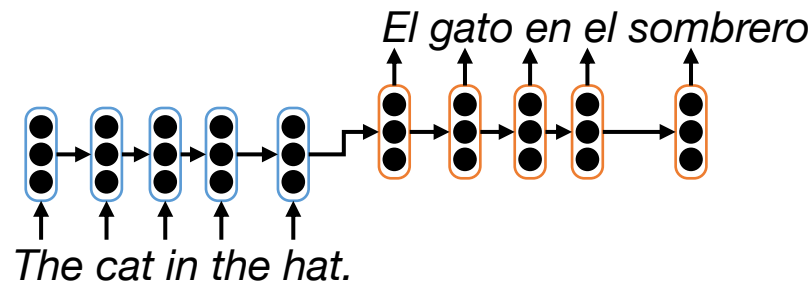
oning tasks



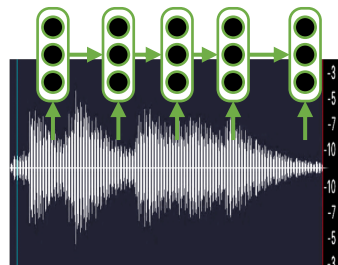
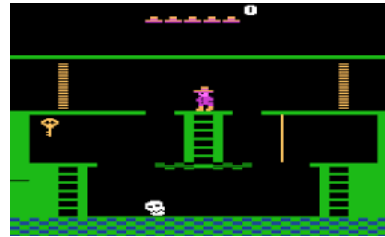
The quick brown fox...

Recurrent Networks

Sequential reasoning tasks



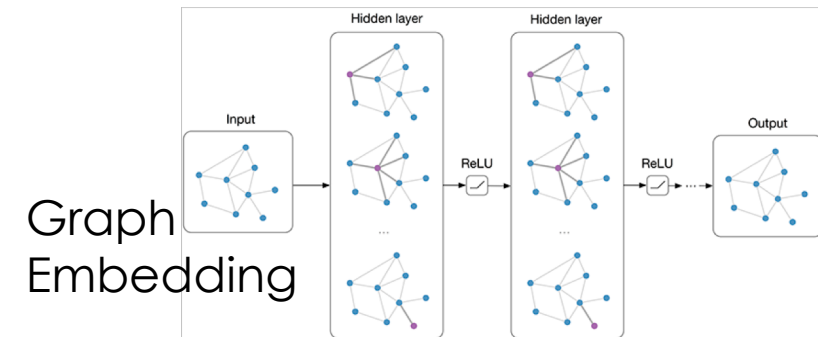
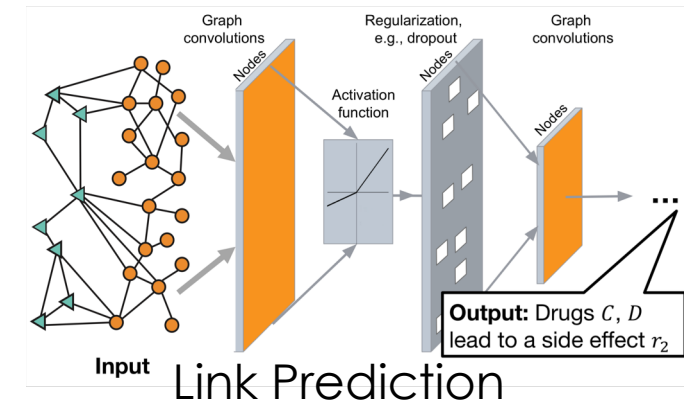
Reinforcement Learning



Speech recognition

Graph Networks

Operating on graph data


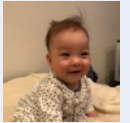
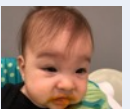


Last Time

Supervised Machine Learning

- Given data containing the function **inputs** and **outputs**

Data

Input	Output
X_1 	Y_1 cat
X_2 	Y_2 baby
...	...
X_n 	Y_n baby

Model

$$f_{\theta}(x) \rightarrow y$$

Parameters

Goal

$$\theta^* = \arg \min_{\theta} \mathbb{E}_D [L(f_{\theta}(x), y)]$$

Loss



Over all future data

Training (approximates the goal over training data):

$$\hat{\theta} = \arg \min_{\theta} \frac{1}{n} \sum_{i=1}^n L(f_{\theta}(x_i), y_i)$$

Learning without Labels

Data

Input	Output
X_1 	Y_1 cat
X_2 	Y_2 baby
...	...
X_n 	Y_n baby

- Can we learn what inputs look like?
 - Useful inductive bias when training for a later supervised task.
 - Often done when labeled data is available but limited
- Convert to a supervised learning problem:




$$f(x) \rightarrow z \quad \textbf{Encoder}$$

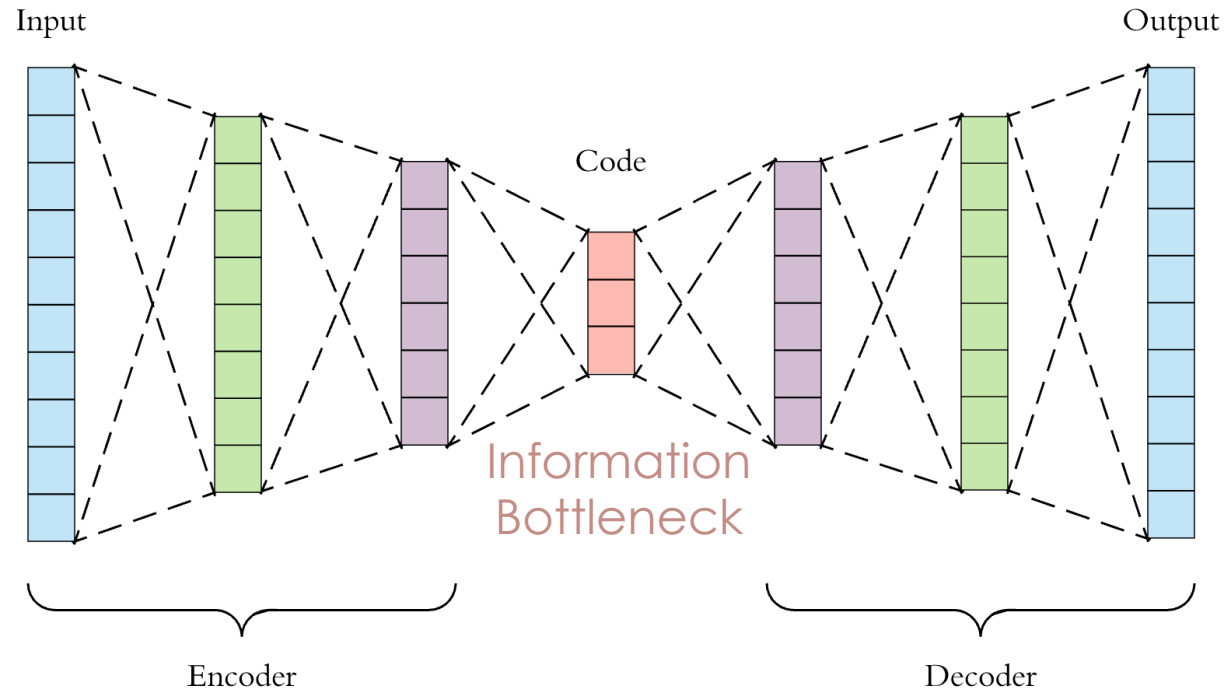
$$g(z) \rightarrow x \quad \textbf{Decoder}$$


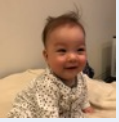

- Convert to a supervised learning problem:

$$f(x) \rightarrow z \quad \textbf{Encoder}$$

$$g(z) \rightarrow x \quad \textbf{Decoder}$$

Data	
Input	Output
X_1 	Y_1 cat
X_2 	Y_2 baby
...	...
X_n 	Y_n baby

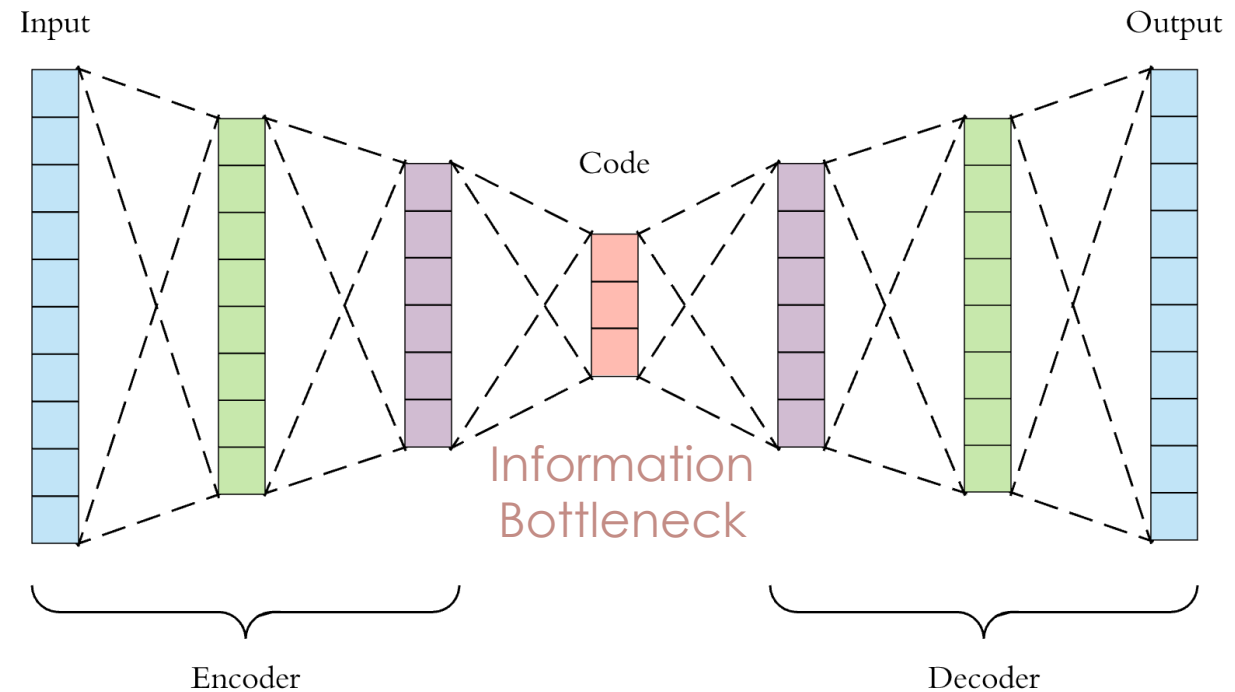


Data	
Input	Output
X_1 	Y_1 cat
X_2 	Y_2 baby
...	...
X_n 	Y_n baby

➤ Convert to a supervised learning problem:

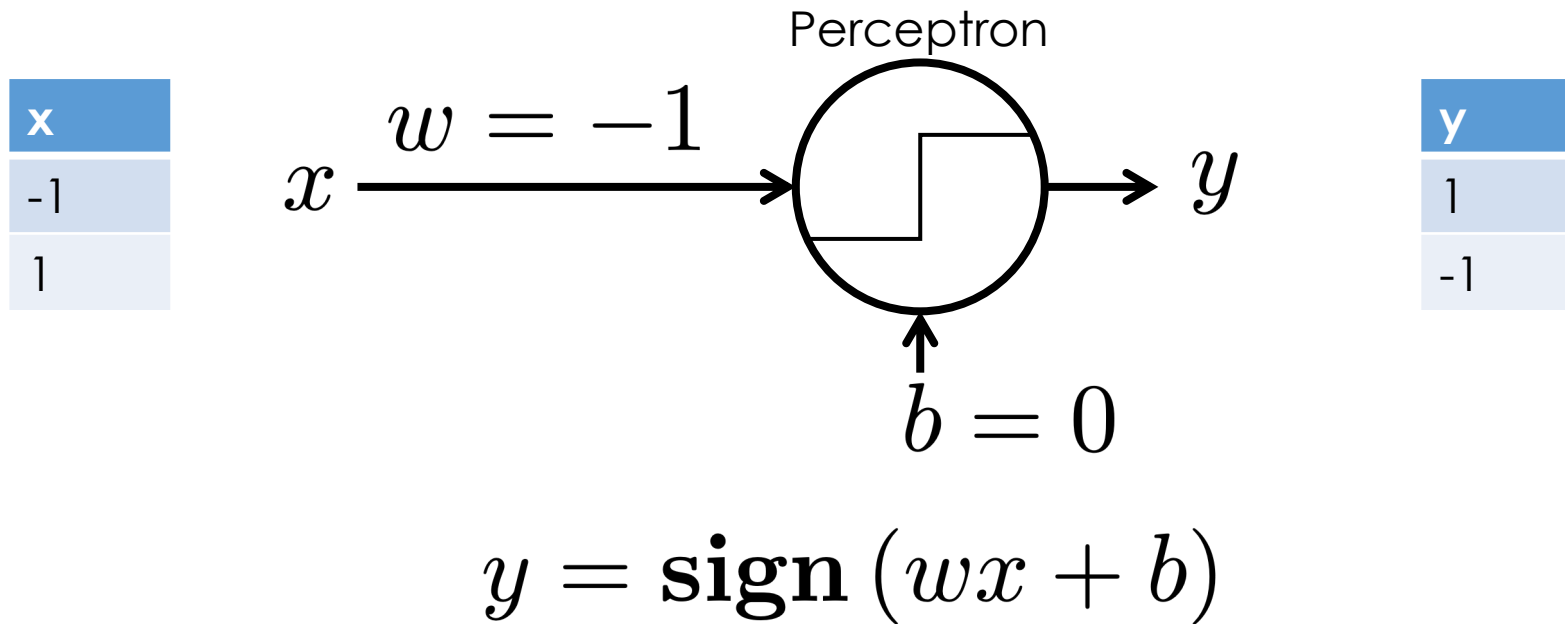
$$\hat{\theta} = \arg \min_{\theta} \frac{1}{n} \sum_{i=1}^n L(g_{\theta_2}(f_{\theta_1}(x_i)), x_i)$$

$f(x) \rightarrow z$ **Encoder**
 $g(z) \rightarrow x$ **Decoder**



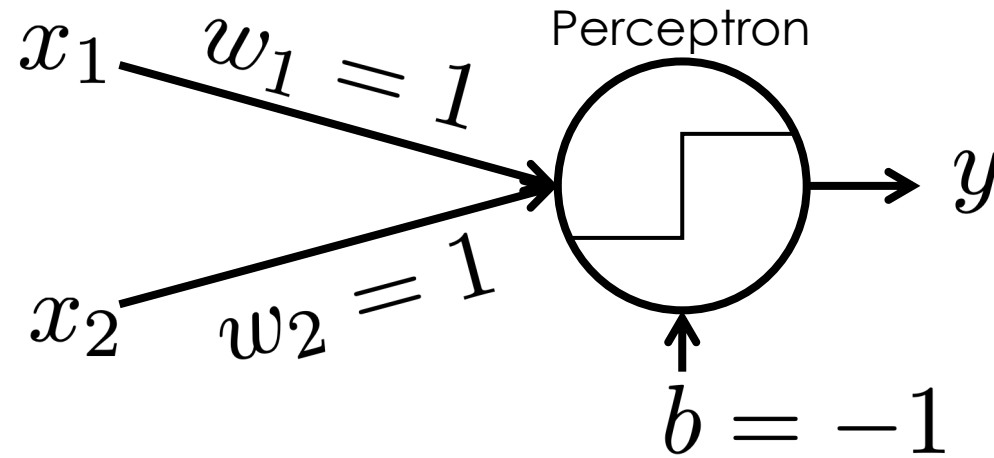
Xor Perceptrons

Perceptron Not Gate



And Gate Perceptron

x_1	x_2
-1	-1
-1	1
1	-1
1	1

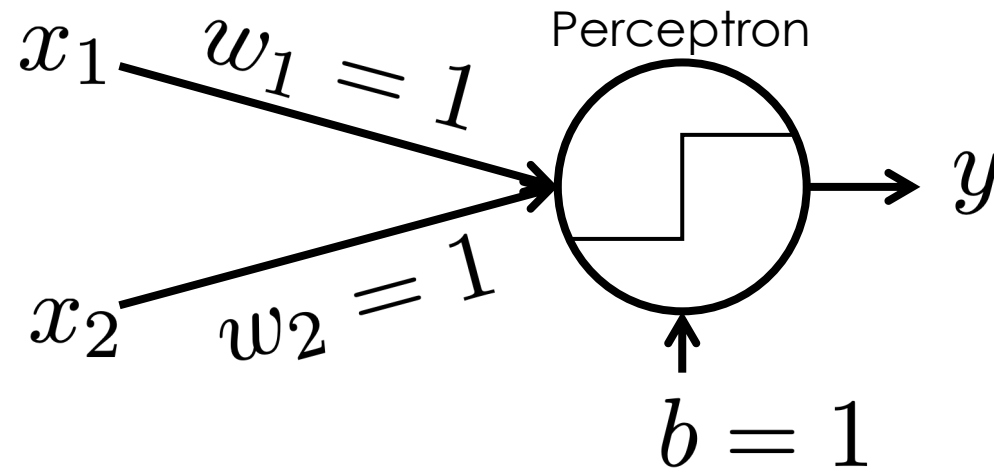


y
-1
-1
-1
1

$$\mathbf{sign}(x_1 w_1 + x_2 w_2 + b) = y$$

Or Gate Perceptron

x_1	x_2
-1	-1
-1	1
1	-1
1	1

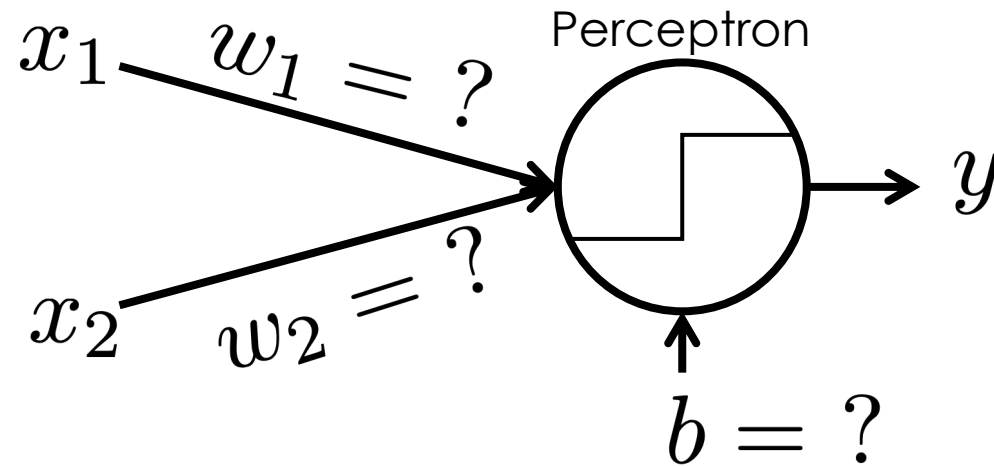


y
-1
1
1
1

$$\mathbf{sign}(x_1 w_1 + x_2 w_2 + b) = y$$

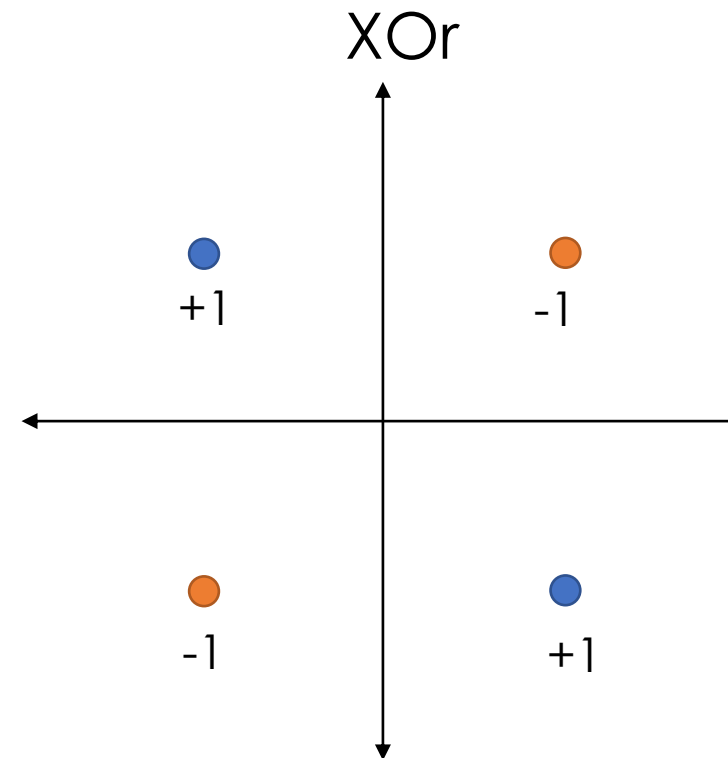
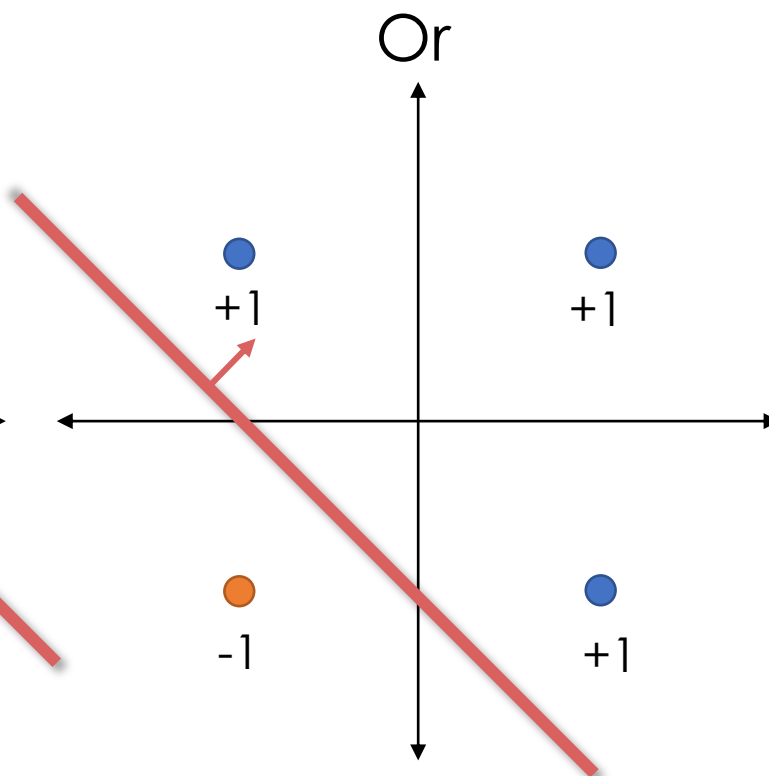
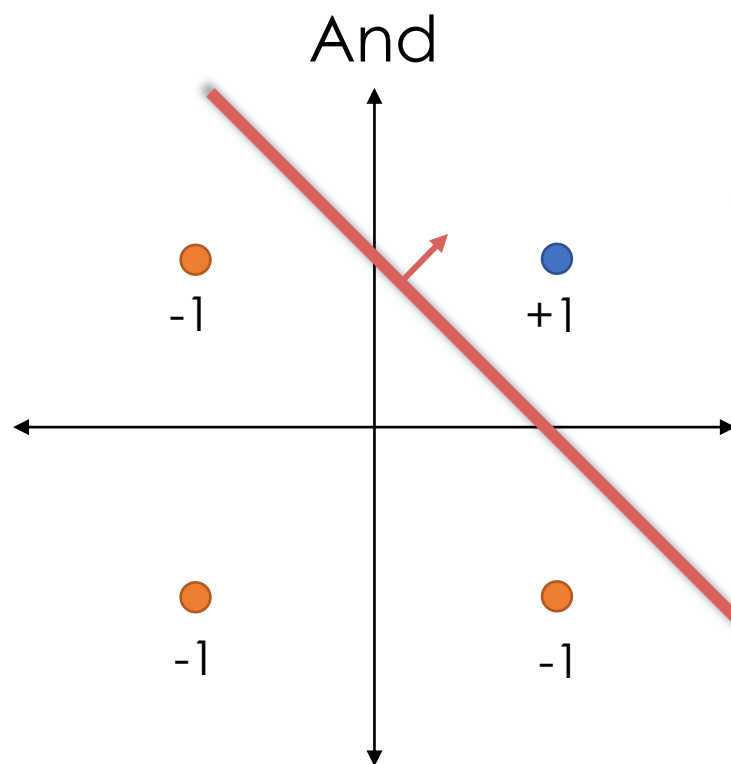
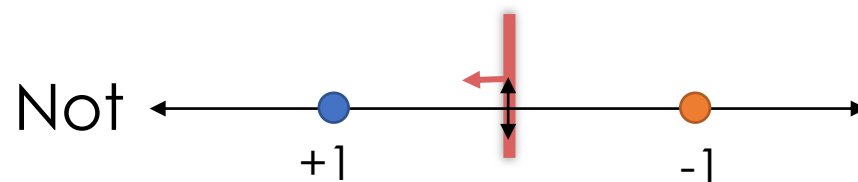
XOr Gate Perceptron

x_1	x_2
-1	-1
-1	1
1	-1
1	1



y
-1
1
1
-1

$$\text{sign}(x_1 w_1 + x_2 w_2 + b) = y$$



No separating hyperplane

$$\text{sign}(x_1w_1 + x_2w_2 + b) = y$$

Using one hidden layer

