Deep Learning Part 2
Other networks

Slides by:

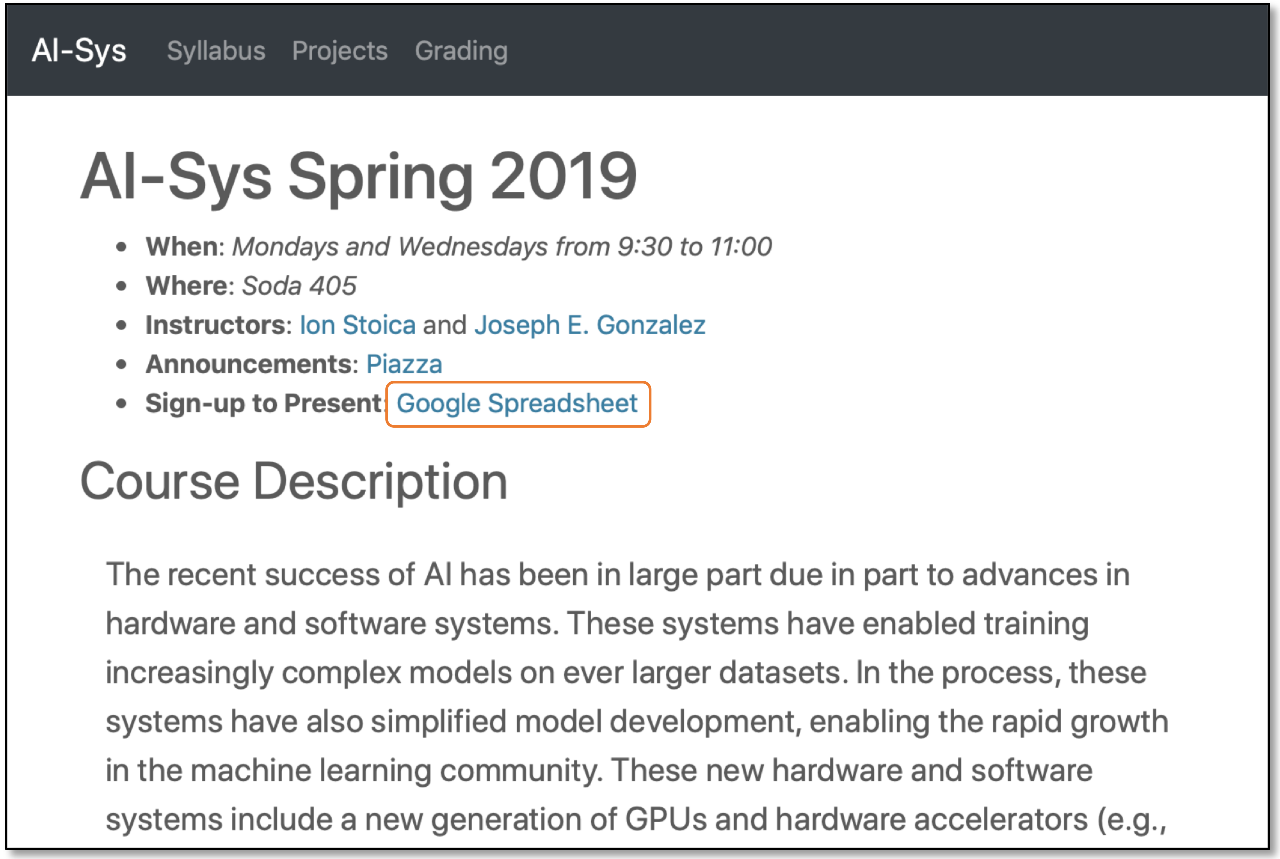
Joseph E. Gonzalez,

jegonzal@berkeley.edu

Quick Logistics

http://bit.ly/aisys-sp19

- 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?



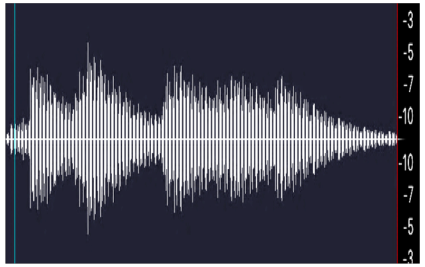
Last Time

Machine Learning ≈ Function Approximation

Object Recognition

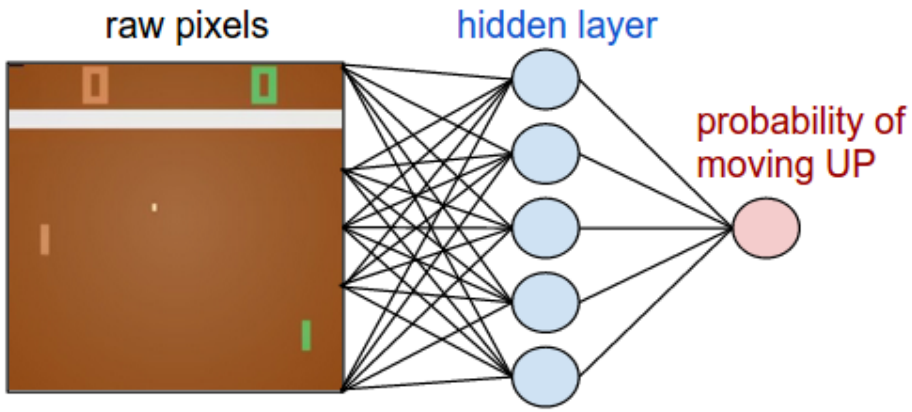
Label:*Cat*

Speech Recognition

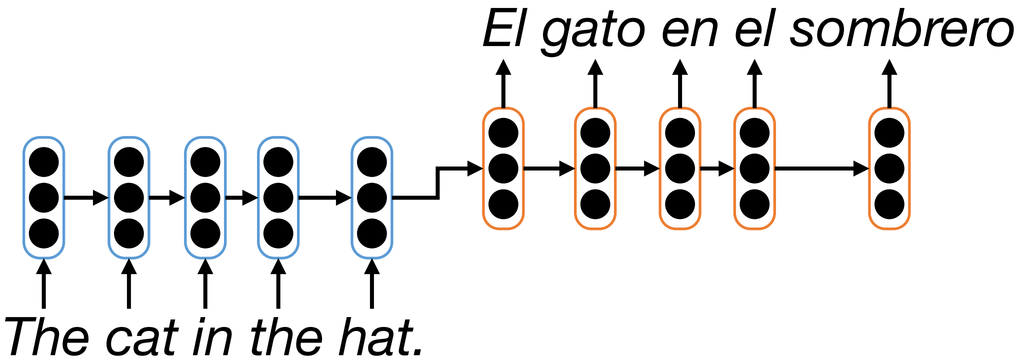


“The cat in the hat”

Robotic Control



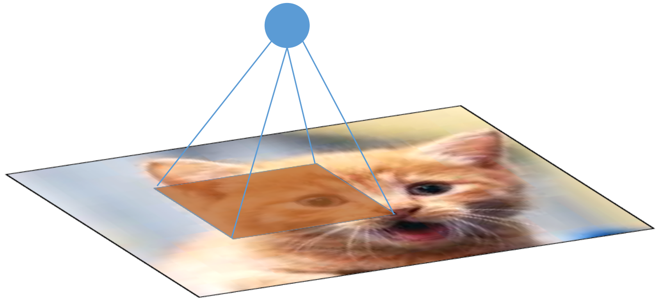
Machine Translation

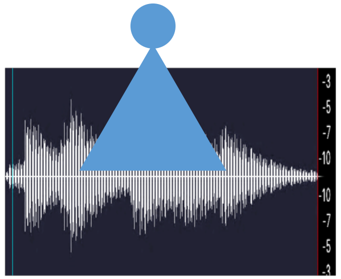


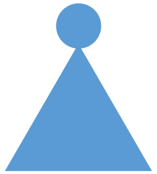
Architectures for Different kinds of inputs

Convolutional Networks

spatial reasoning tasks



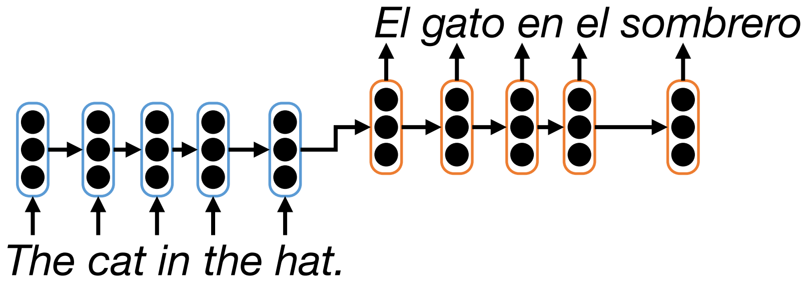




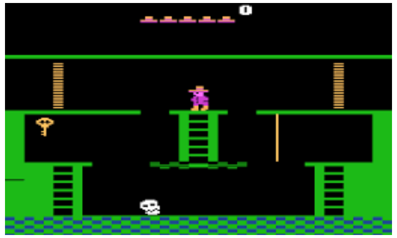
The quick brown fox…

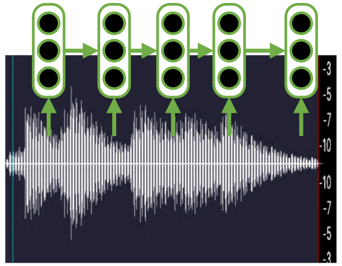
Recurrent Networks

Sequential reasoning tasks



Reinforcement
Learning



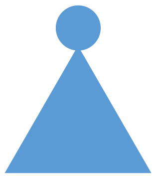


Speech
recognition

Architectures for Different kinds of inputs

al Networks

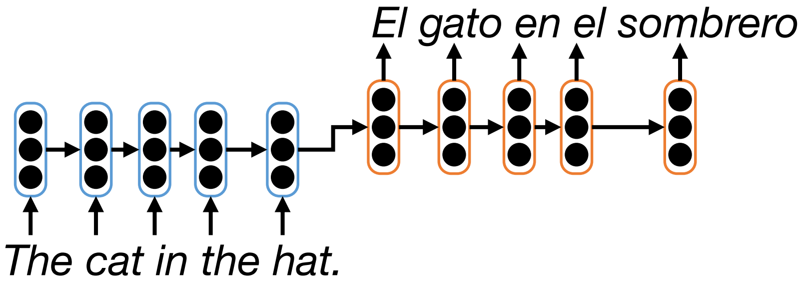
oning tasks



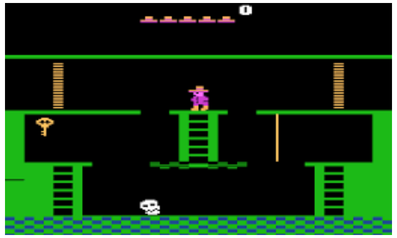
The quick brown fox…

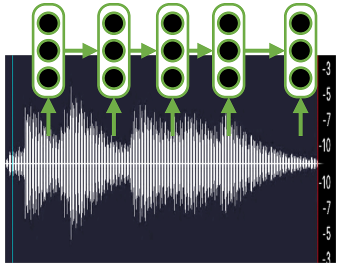
Recurrent Networks

Sequential reasoning tasks



Reinforcement Learning

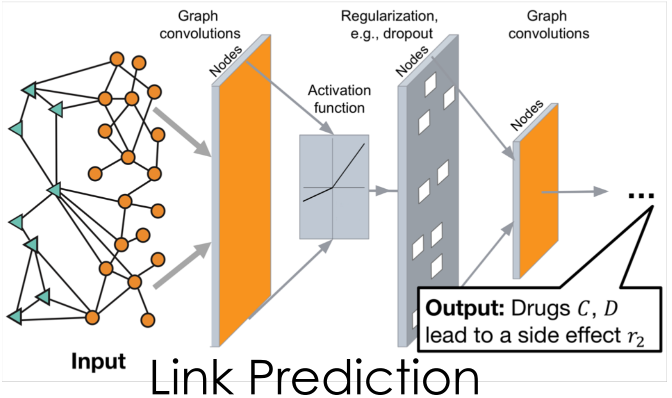




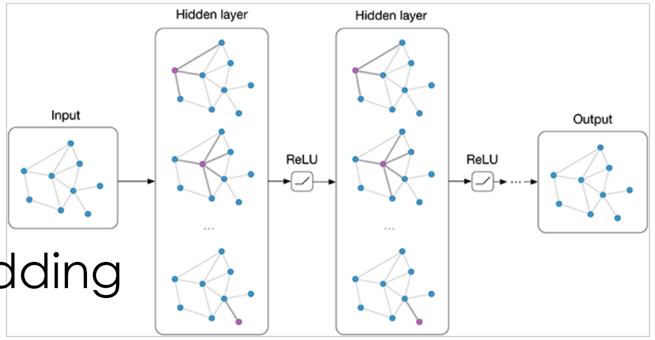
Speech recognition

Graph Networks

Operating on graph data



Link Prediction



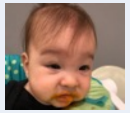
Graph Embedding

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

Loss

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

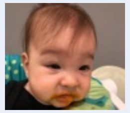
Over all future data

Training (approximates the goal over training data):

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

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:

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

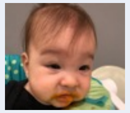
$$g(z) \rightarrow x \quad \textbf{Decoder}$$

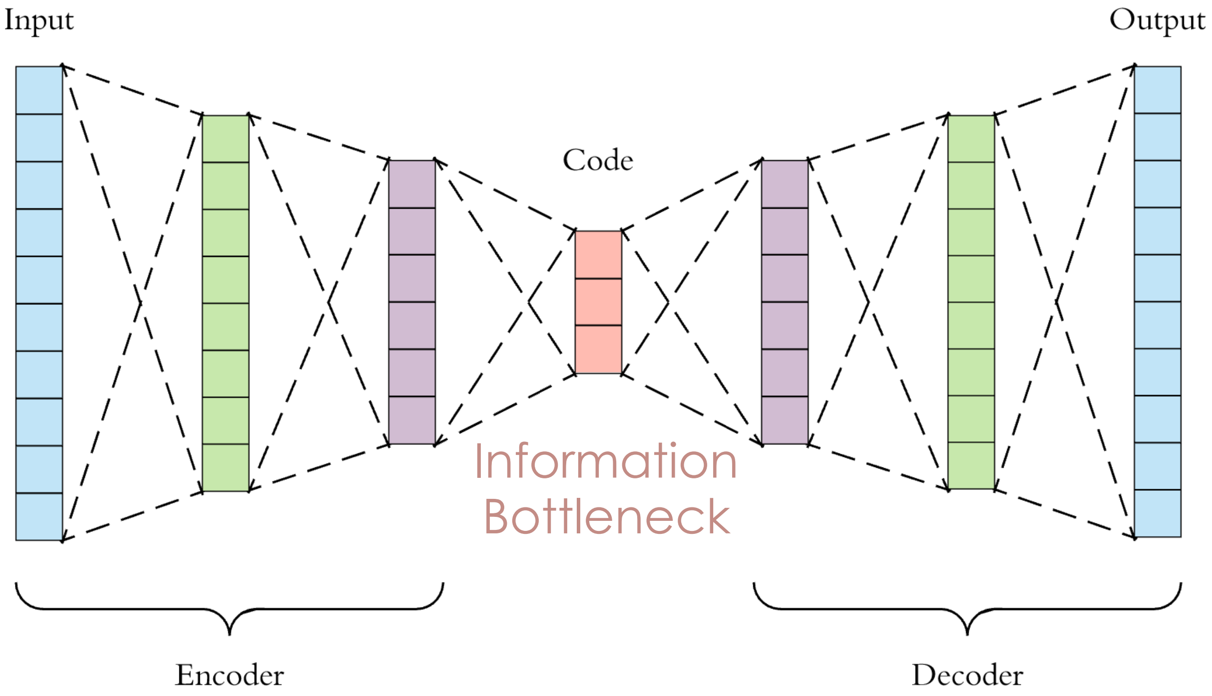
- Convert to a supervised learning problem:

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

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

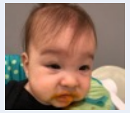
Data

Input	Output
X_1 	Y_1 cat
X_2 	Y_2 *baby*
…	…
X_n 	Y_n *baby*

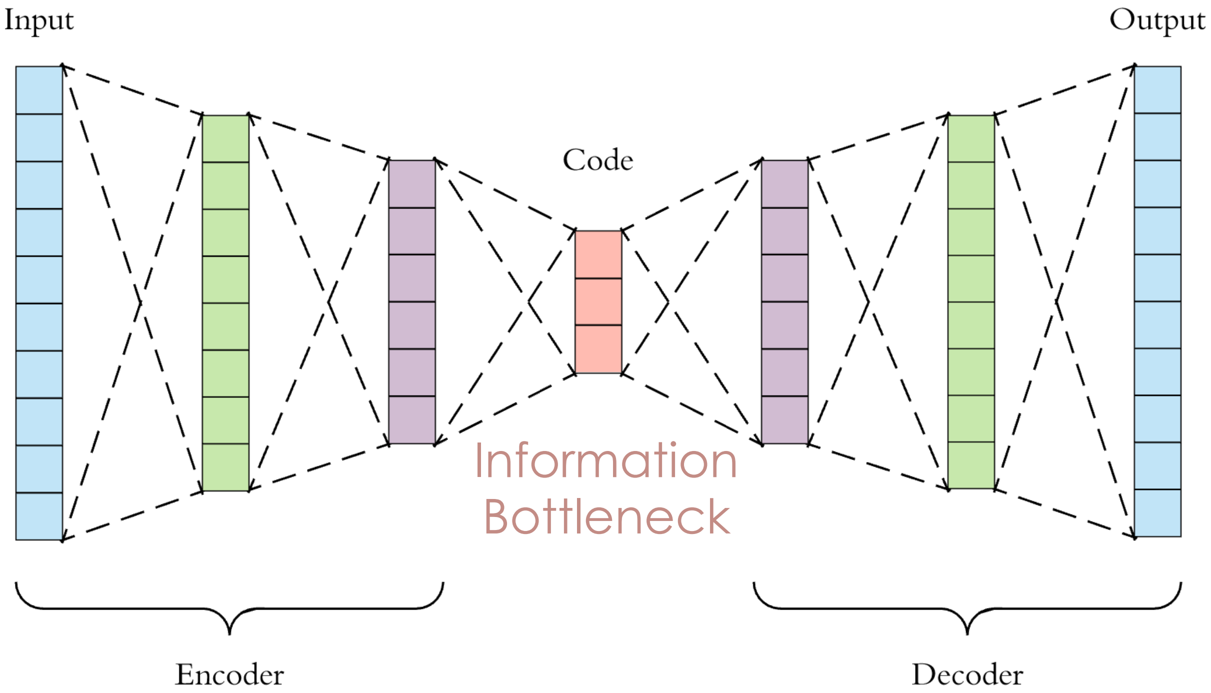


- Convert to a supervised learning problem:

Data

Input	Output
X_1 	Y_1 cat
X_2 	Y_2 *baby*
…	…
X_n 	Y_n *baby*

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



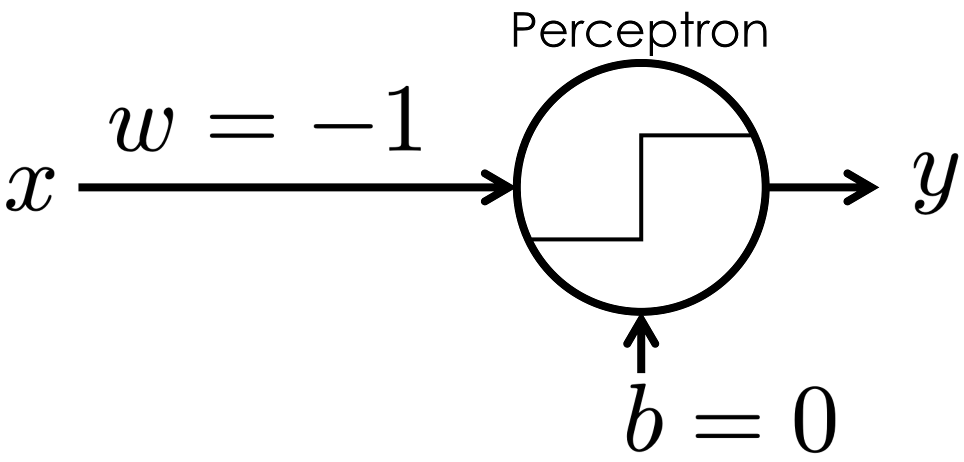
$f(x) \rightarrow z$ **Encoder**

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

Xor Perceptrons

Perceptron Not Gate

x
-1
1

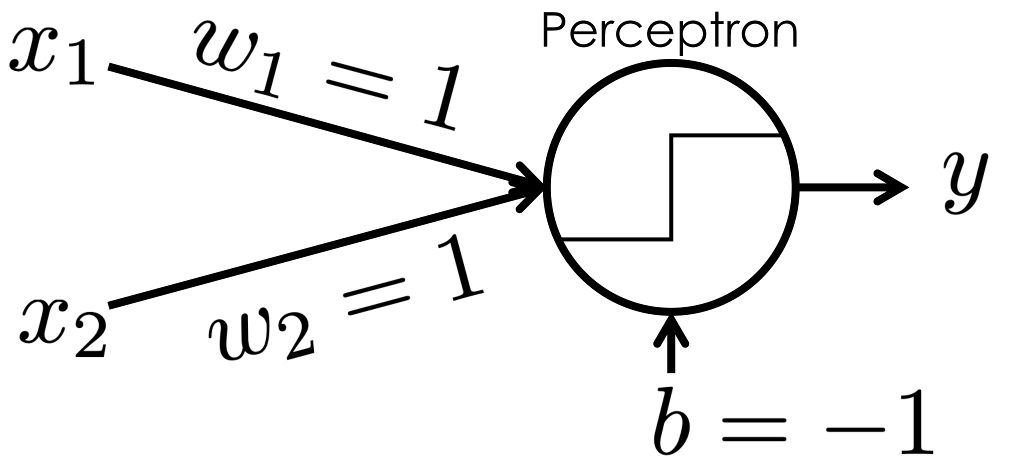


y
1
-1

$$y = \mathbf{sign}\left(wx + b\right)$$

And Gate Perceptron

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

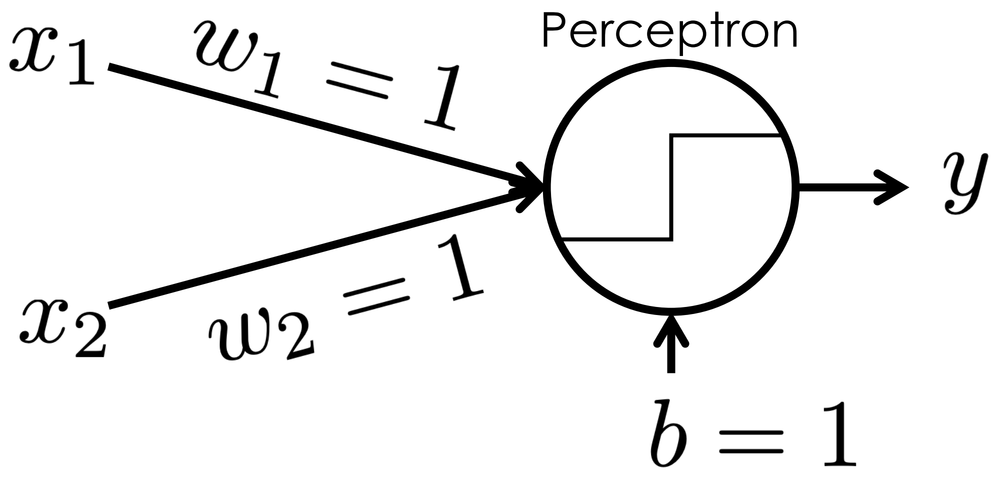


y
-1
-1
-1
1

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

Or Gate Perceptron

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

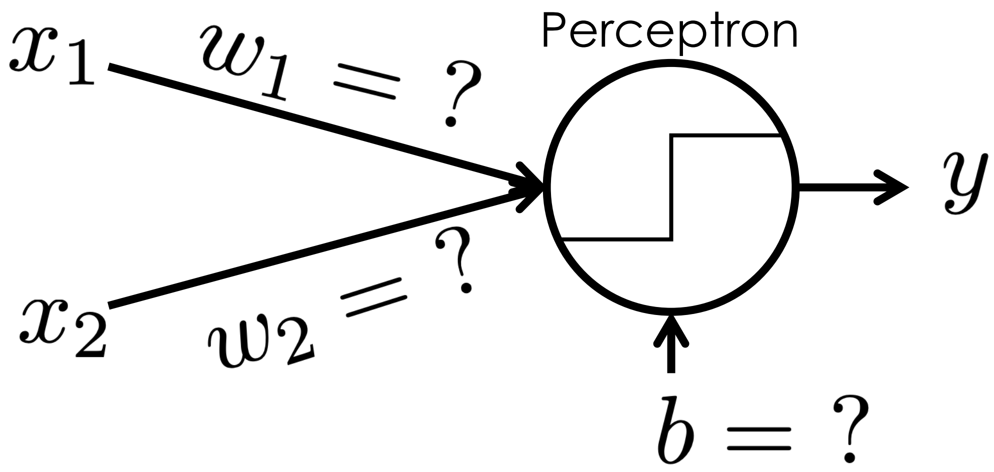


y
-1
1
1
1

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

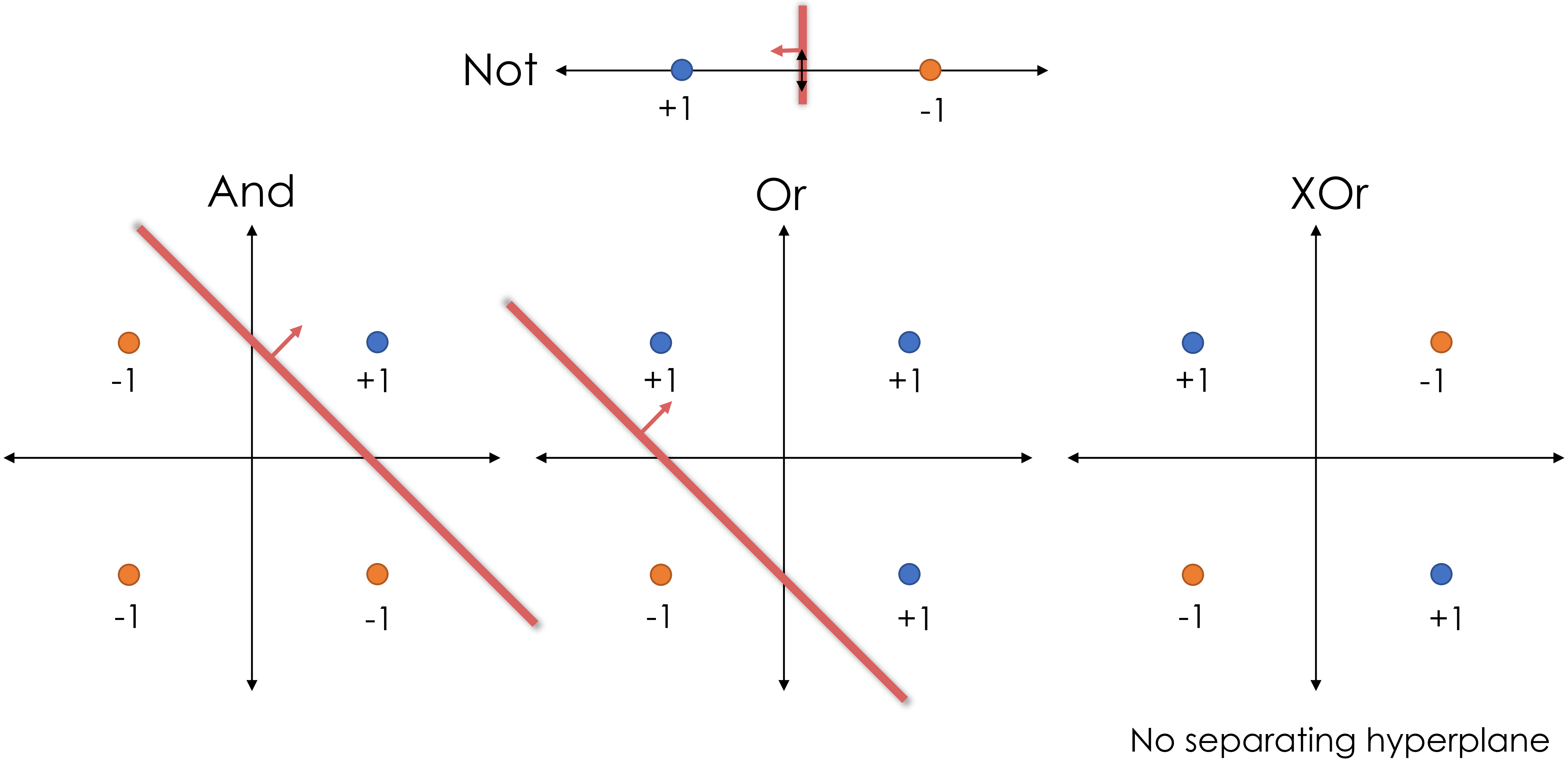
XOr Gate Perceptron

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



y
-1
1
1
-1

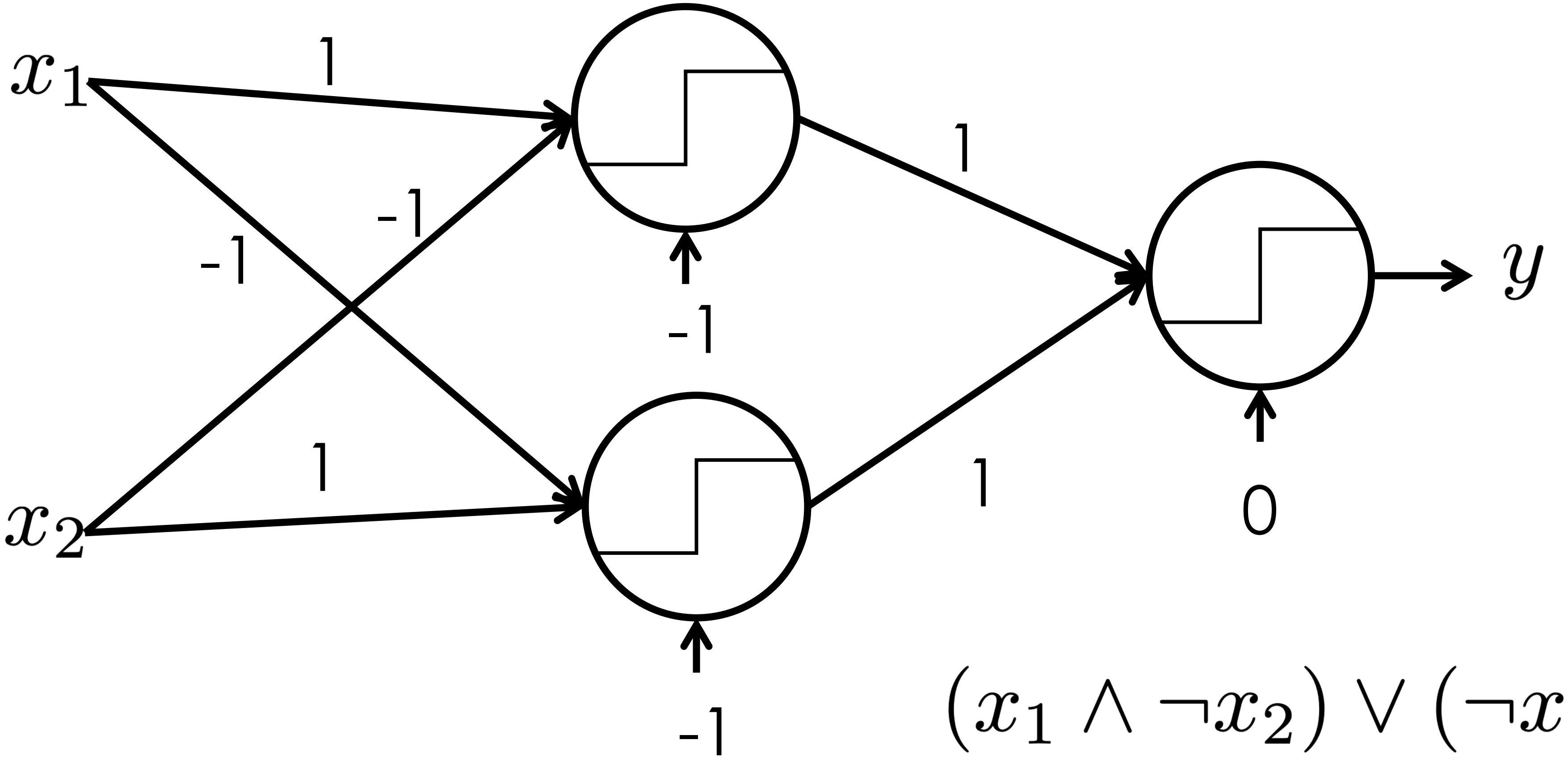
$$\mathbf{sign}\left(x_1 w_1 + x_2 w_2 + b\right) = y$$



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

Using one hidden layer

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



y
-1
1
1
-1

$$(x_1 \wedge \neg x_2) \vee (\neg x_1 \wedge x_2)$$