Neural Networks Part I

The Multilayer Perceptron

Neural Networks History Lesson

1962: Rosenblatt, *Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms*

- First neuron-based learning algorithm
- Allegedly "could learn anything that you could program"

Neural Networks History Lesson

1962: Rosenblatt, *Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms*

- First neuron-based learning algorithm
- Allegedly "could learn anything that you could program"

1969: Minsky & Papert, Perceptron: *An Introduction to Computational Geometry*

- First real complexity analysis
- Showed, in principle, many things that perceptrons can't learn to do
- Shut down any interest in neural networks

Neural Networks History Lesson

1986: Rumelhart, **Hinton** & Williams, *Back Propagation*

- Overcame many difficulties raised by Minsky, et al
- Neural Networks wildly popular again (for a while)

Neural Networks History Lesson

1999-2005:

Shift to **Bayesian Methods**:

- Best ideas from neural networks
- Direct statistical computing

Support Vector Machines:

- Nice mathematical properties
- Global optima vs local optima (like NNs)

Neural Networks History Lesson

1999-2005:

Shift to **Bayesian Methods**:

- Best ideas from neural networks
- Direct statistical computing

Support Vector Machines:

- Nice mathematical properties
- Global optima vs local optima (like NNs)

A few people still playing with NNs:

- Hinton (Univ. Toronto and Google)
- LeCun (NYU and Facebook)
- Bengio (Univ. Montreal)

Neural Networks History Lesson

2005-2010:

- Core group of people continue to make improvements
- Developed various tricks to make NN learning practical

2010-Present:

- Went back to the NN methods of the 1980s
- Have been wildly successful

Neural Networks History Lesson

2005-2010:

- Core group of people continue to make improvements
- Developed various tricks to make NN learning practical

2010-Present:

- Went back to the NN methods of the 1980s
- Have been wildly successful

Question for You: Why? What made a 1980s algorithm suddenly amazing 30 years later?

Neural Networks History Lesson

Question for You: Why? What made a 1980s algorithm suddenly amazing 30 years later?

Efficient algorithms (Back Propagation)

Raw computing power

Massive amounts of training data:

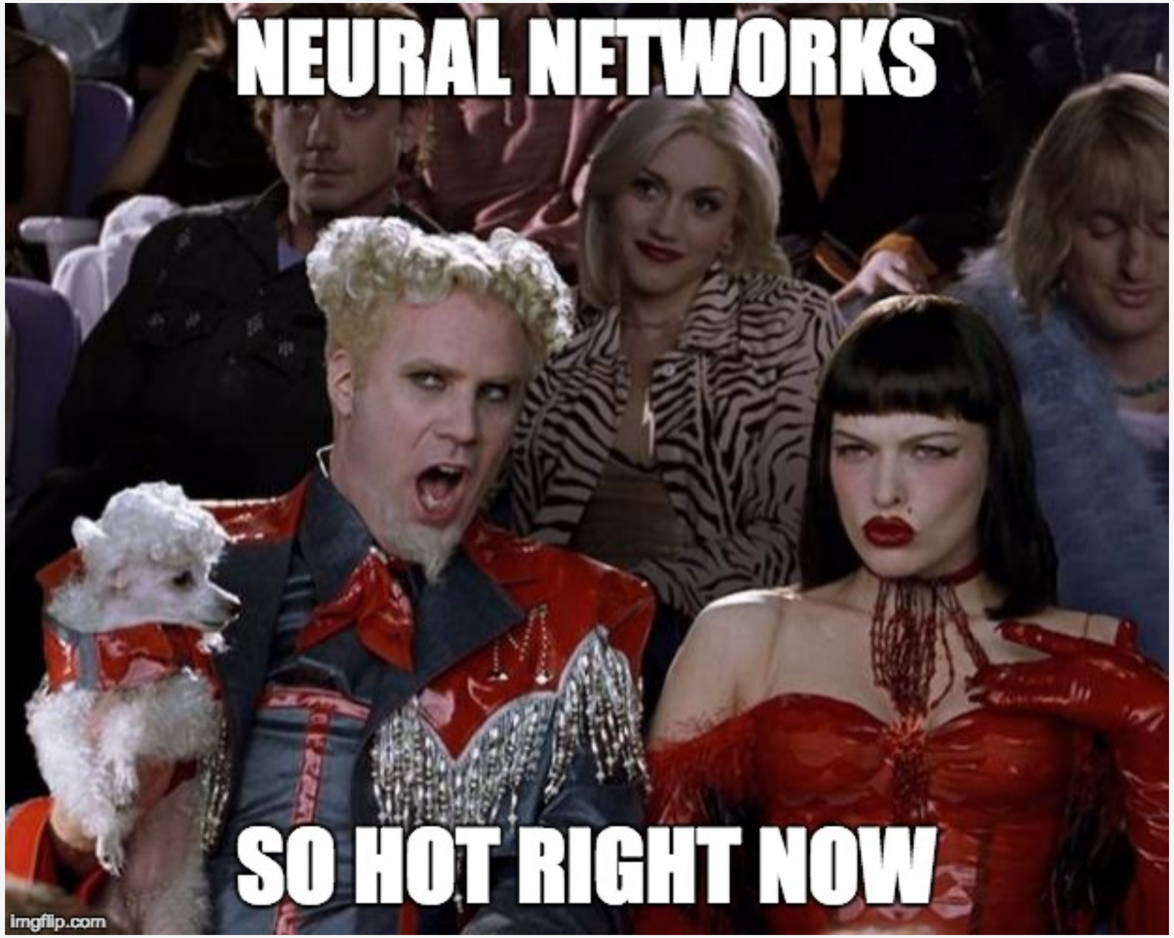
- Deep Neural Nets in particular have a massive amount of parameters
- Need tons of data to effectively train accurate models

Neural Networks History Lesson

The history of ML has been very **cyclic**

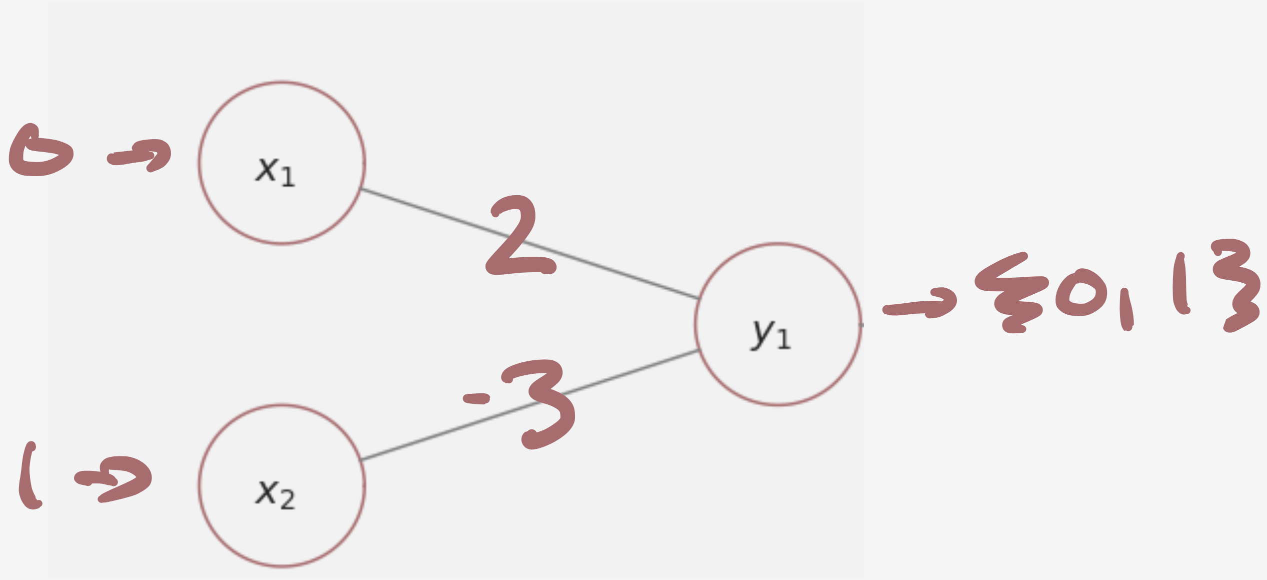
But will they be 20 years from now?

¯_(ツ)_/¯



The Perceptron

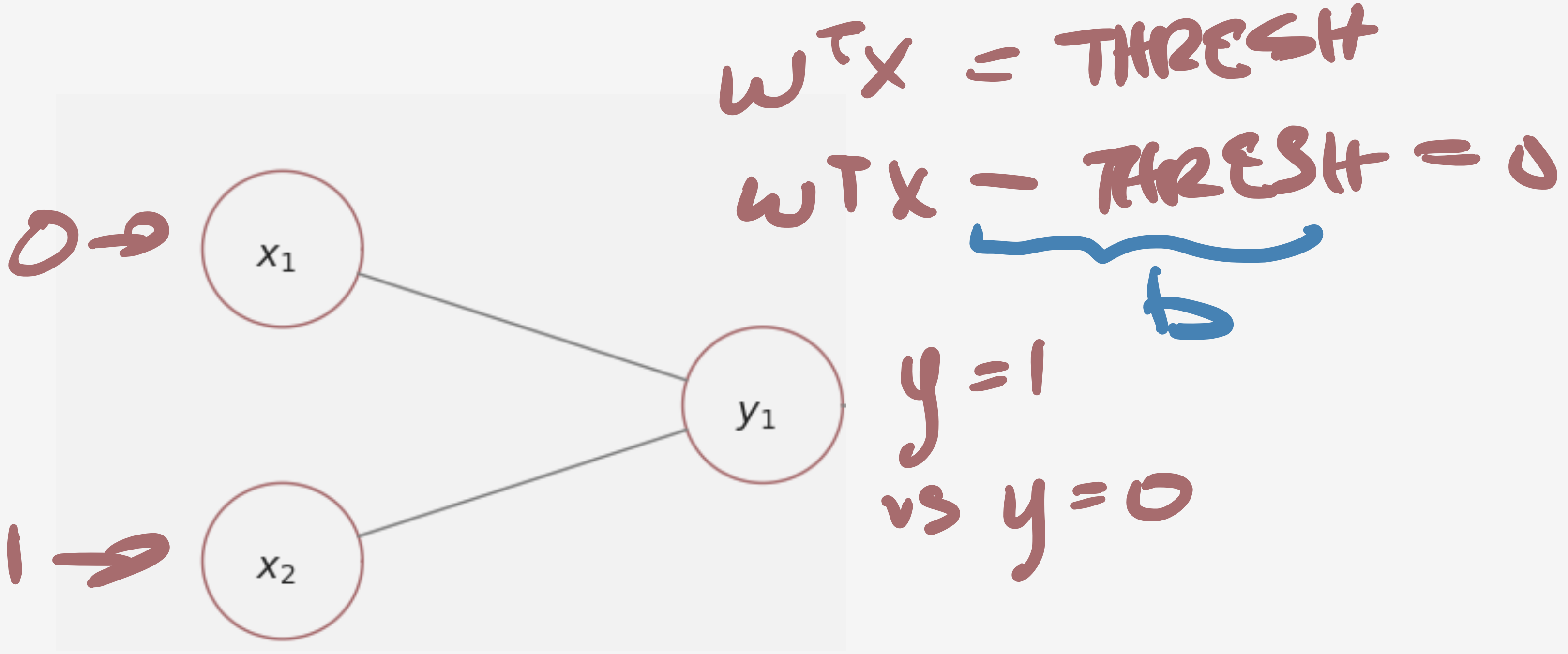
In its simplest form, a perceptron takes some binary inputs, and predicts a binary output



Given some training data, learn weights associated with edges that make predictions accurate

The Perceptron

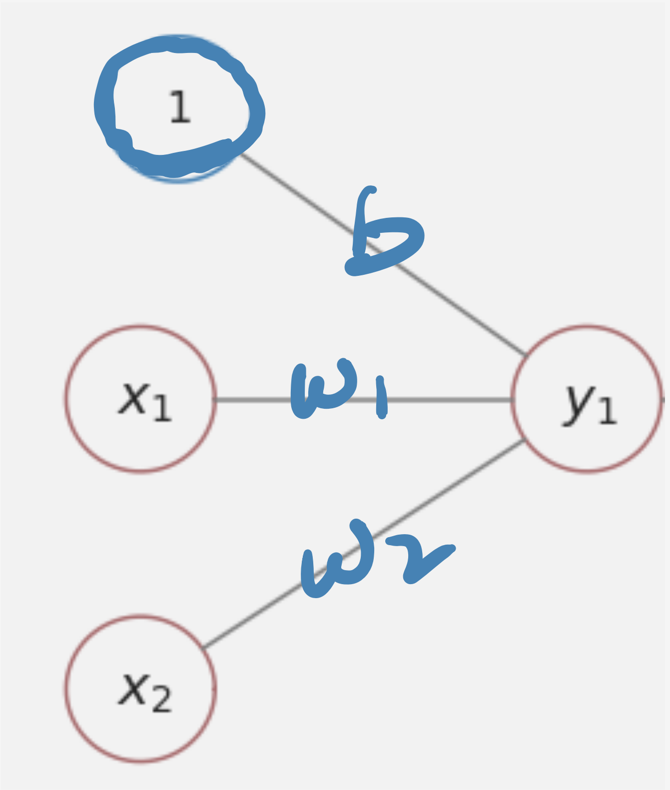
In its simplest form, a perceptron takes some binary inputs, and predicts a binary output



$$\hat{y} = \begin{cases} 0 & \text{if } \mathbf{w}^T\mathbf{x} \leq \text{ threshold} \\ 1 & \text{if } \mathbf{w}^T\mathbf{x} > \text{ threshold} \end{cases}$$

The Perceptron

In its simplest form, a perceptron takes some binary inputs, and predicts a binary output

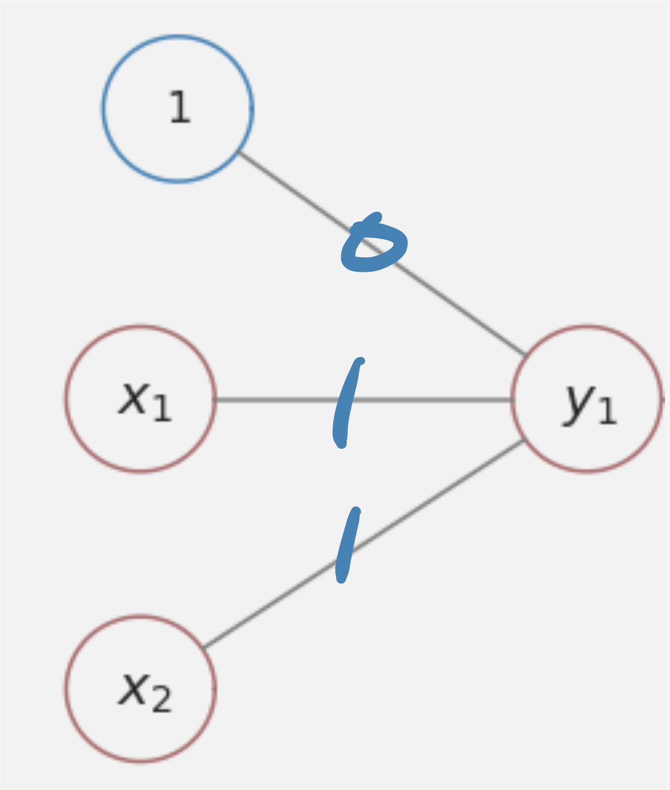


$$\hat{y} = \begin{cases} 0 & \text{if } \mathbf{w}^T\mathbf{x} + b \leq 0 \\ 1 & \text{if } \mathbf{w}^T\mathbf{x} + b > 0 \end{cases}$$

The Perceptron

Simple Example: Learning OR

$$\hat{y} = \begin{cases} 0 & \text{if } \mathbf{w}^T\mathbf{x} + b \leq 0 \\ 1 & \text{if } \mathbf{w}^T\mathbf{x} + b > 0 \end{cases}$$



$w = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$ $b = 0$

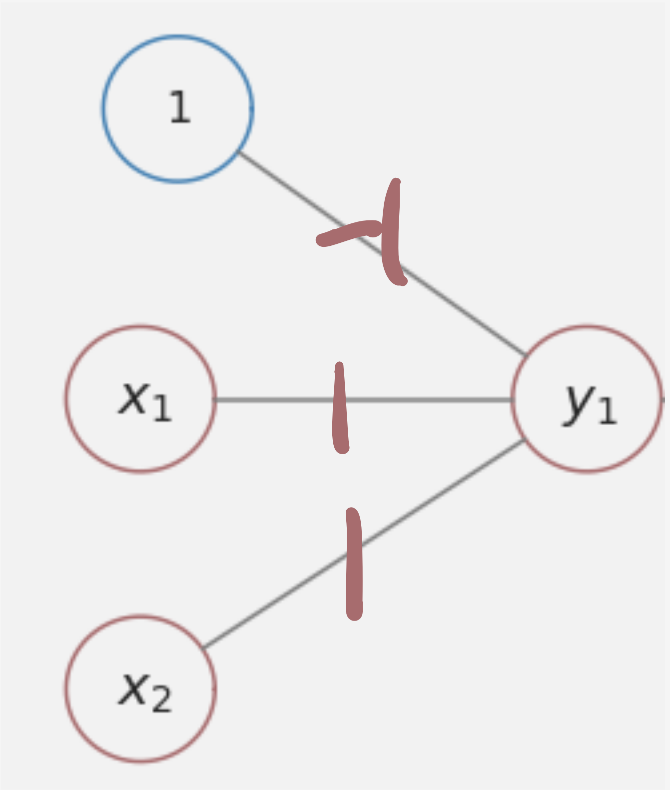
x_1	0	1	0	1
x_2	0	0	1	1
x_1 OR x_2	0	1	1	1

$w^T \begin{bmatrix} 1 \\ 1 \end{bmatrix} + b = [1 \;\; 1]^T \begin{bmatrix} 1 \\ 1 \end{bmatrix} + 0 = 2 > 0$

$\Rightarrow y = 1$

The Perceptron

Simple Example: Learning AND



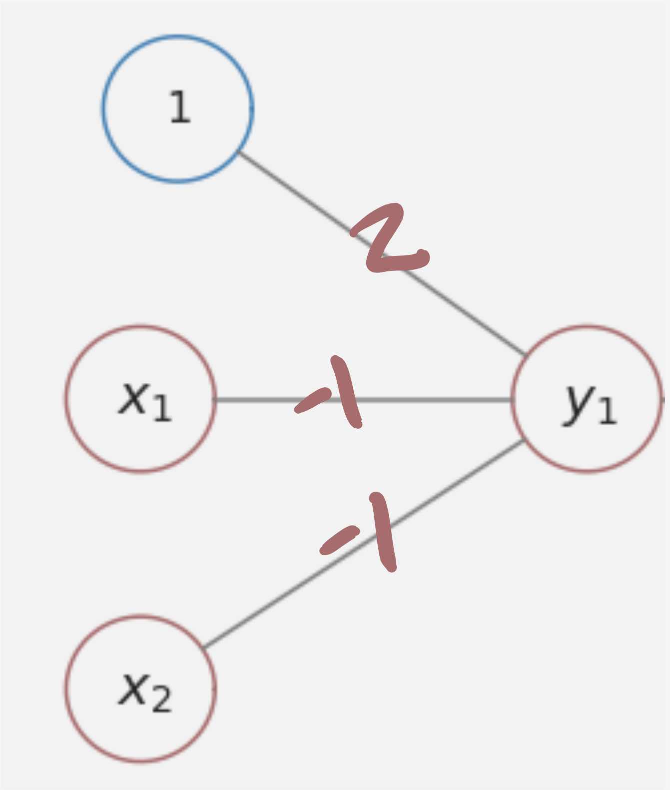
$$\hat{y} = \begin{cases} 0 & \text{if } \mathbf{w}^T\mathbf{x} + b \leq 0 \\ 1 & \text{if } \mathbf{w}^T\mathbf{x} + b > 0 \end{cases}$$

x_1	0	1	0	1
x_2	0	0	1	1
x_1 AND x_2	0	0	0	1

$w = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$ & $b = -1$

The Perceptron

Simple Example: Learning NAND



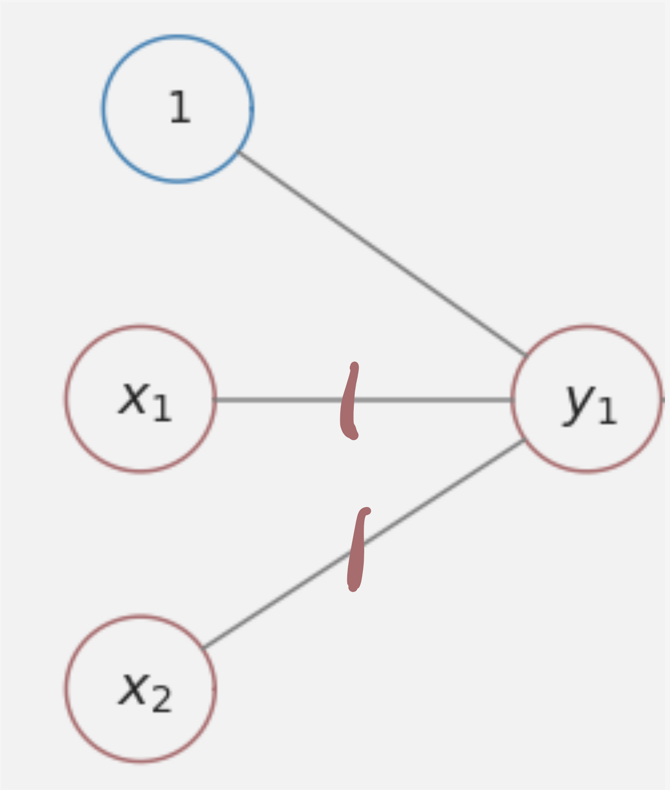
$$\hat{y} = \begin{cases} 0 & \text{if } \mathbf{w}^T\mathbf{x} + b \leq 0 \\ 1 & \text{if } \mathbf{w}^T\mathbf{x} + b > 0 \end{cases}$$

x_1	0	1	0	1
x_2	0	0	1	1
x_1 NAND x_2	1	1	1	0

$$w^T x + b = [-1 \ -1]^T \begin{bmatrix} 1 \\ 1 \end{bmatrix} + 1.5$$

The Perceptron

Simple Example: Can we learn XOR?



$$\hat{y} = \begin{cases} 0 & \text{if } \mathbf{w}^T\mathbf{x} + b \leq 0 \\ 1 & \text{if } \mathbf{w}^T\mathbf{x} + b > 0 \end{cases}$$

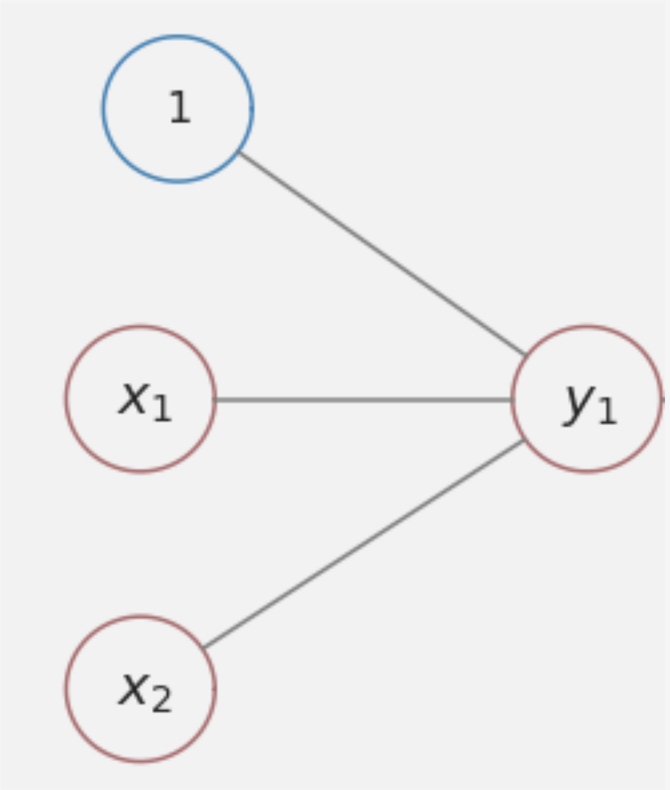
x_1	0	1	0	1
x_2	0	0	1	1
x_1 XOR x_2	0	1	1	0

$w^T x = +1$

$w^T x + b \leq 0$

The Perceptron

Simple Example: Can we learn XOR?



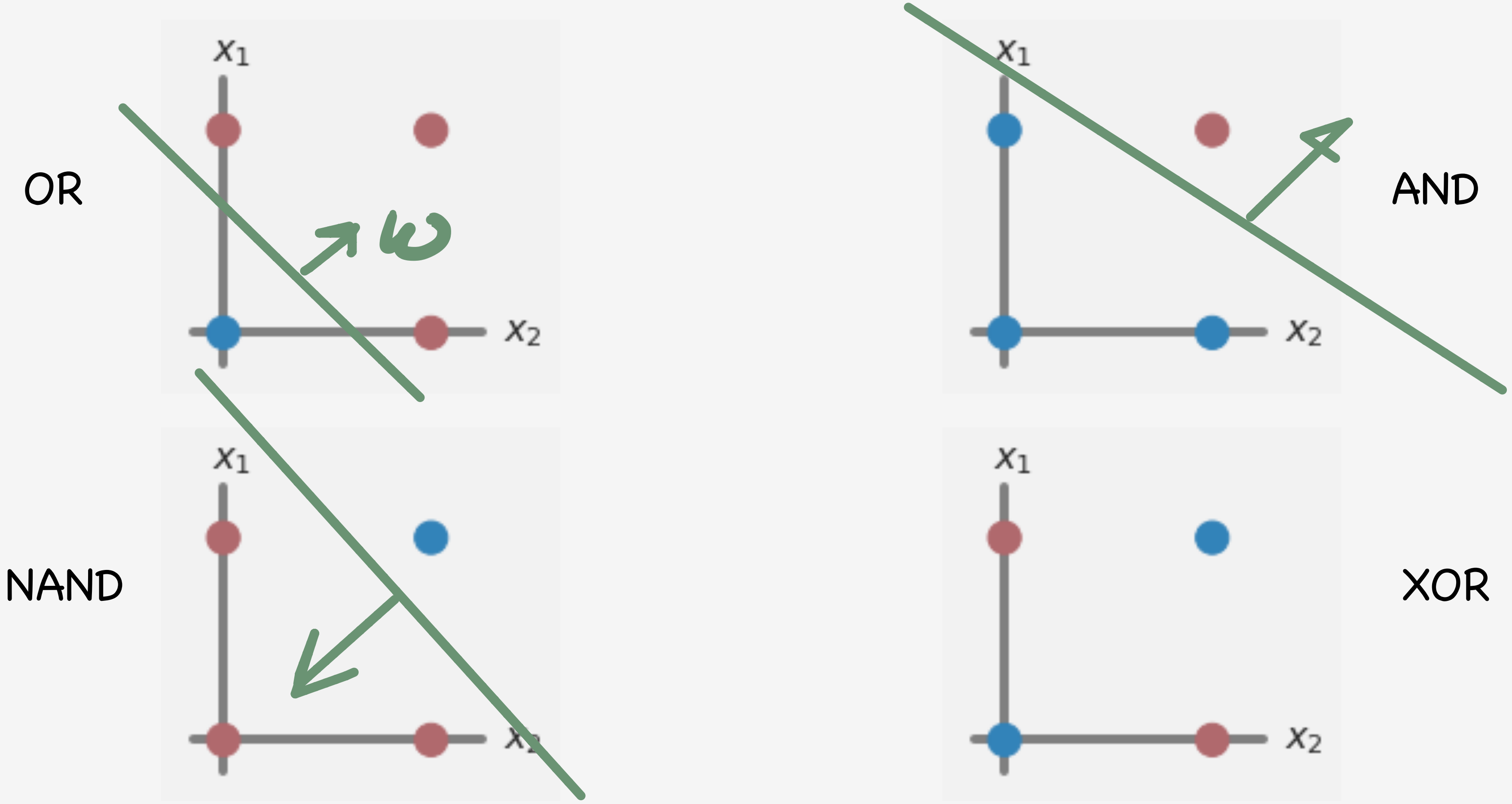
$$\hat{y} = \begin{cases} 0 & \text{if } \mathbf{w}^T\mathbf{x} + b \leq 0 \\ 1 & \text{if } \mathbf{w}^T\mathbf{x} + b > 0 \end{cases}$$

x_1	0	1	0	1
x_2	0	0	1	1
x_1 XOR x_2	0	1	1	0

NOPE!

The Perceptron

Question: Why can't we learn XOR?



The Perceptron

Question: Why can't we learn XOR?

Answer: The perceptron is a linear classifier. Can only learn things that are linearly separable

Question: How can we fix this?

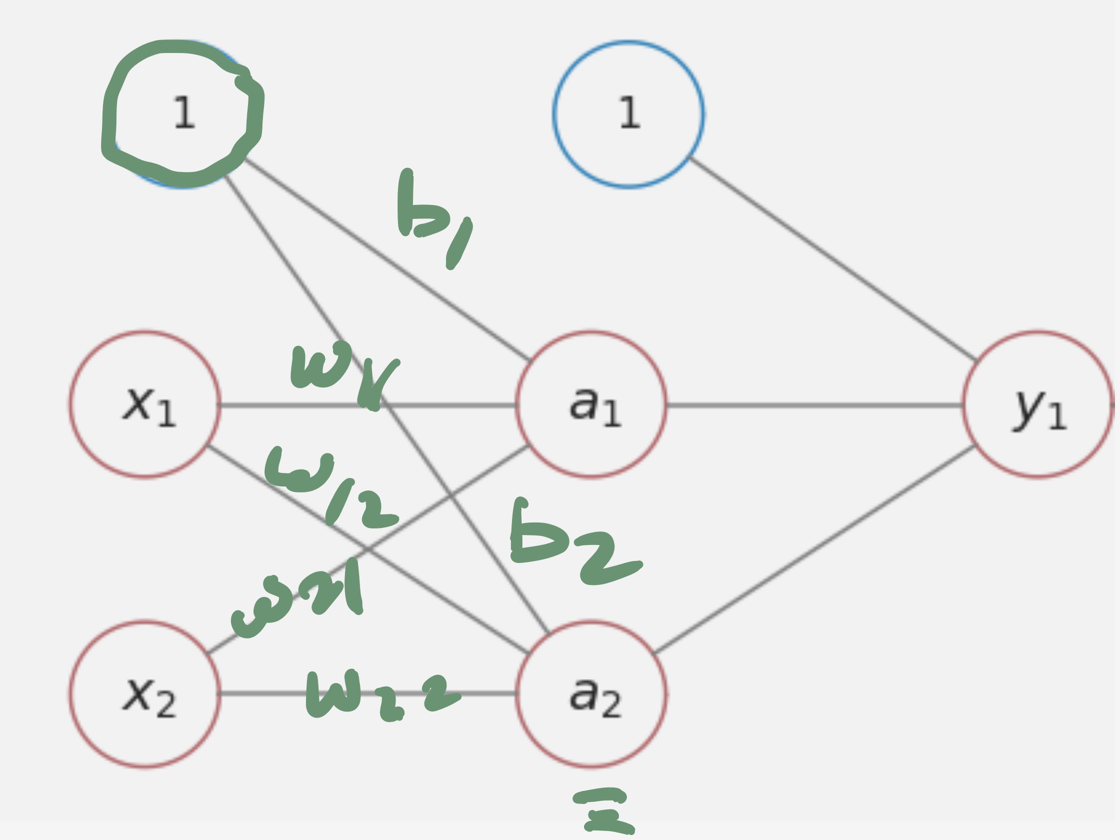
The Multilayer Perceptron

Question: How can we fix this?

Answer: The Multilayer Perceptron

Anatomy:

- Input Layer
- Hidden Layer
- Output Layer



The Multilayer Perceptron

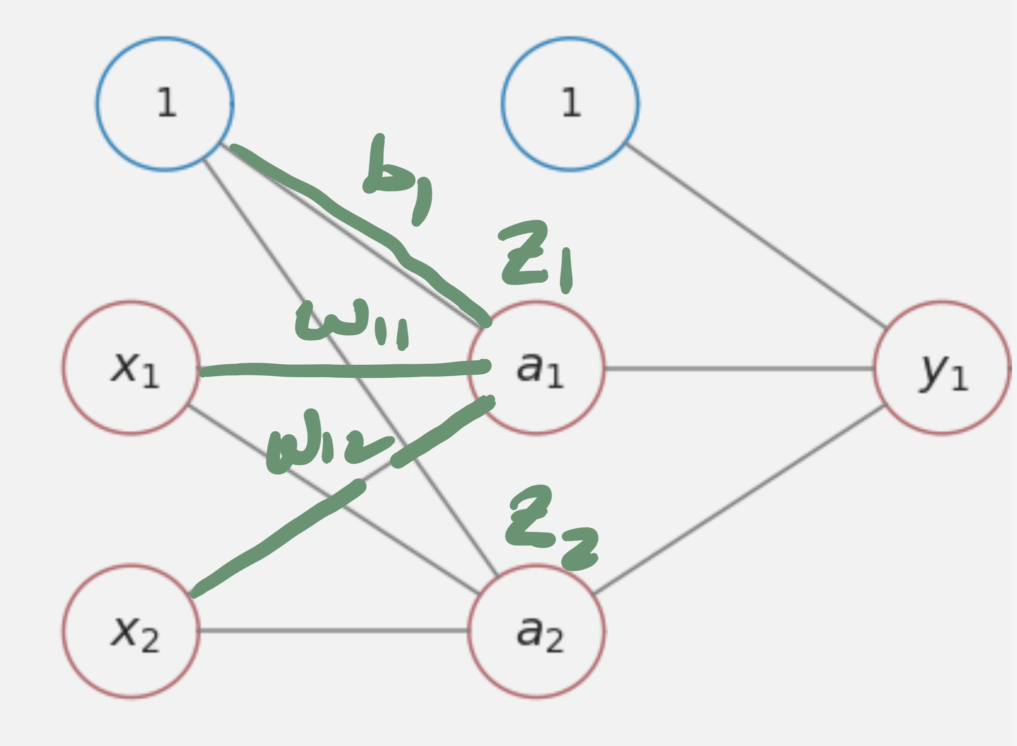
Anatomy: Input Layer to Hidden Layer

- Define intermediate **activities z**
- Define **activations a**

The indicator I is the **activation function**

$$z_1 = w^1_{11}x_1 + w^1_{12}x_2 + b^1_1, \quad a_1 = \mathrm{I}(z_1 > 0)$$

$$z_2 = w^1_{21}x_1 + w^1_{22}x_2 + b^1_2, \quad a_2 = \mathrm{I}(z_2 > 0)$$



The Multilayer Perceptron

Vectorized Anatomy: Input Layer to Hidden Layer

$$z_1 = w^1_{11}x_1 + w^1_{12}x_2 + b^1_1, \quad a_1 = \mathrm{I}(z_1 > 0)$$
$$z_2 = w^1_{21}x_1 + w^1_{22}x_2 + b^1_2, \quad a_2 = \mathrm{I}(z_2 > 0)$$

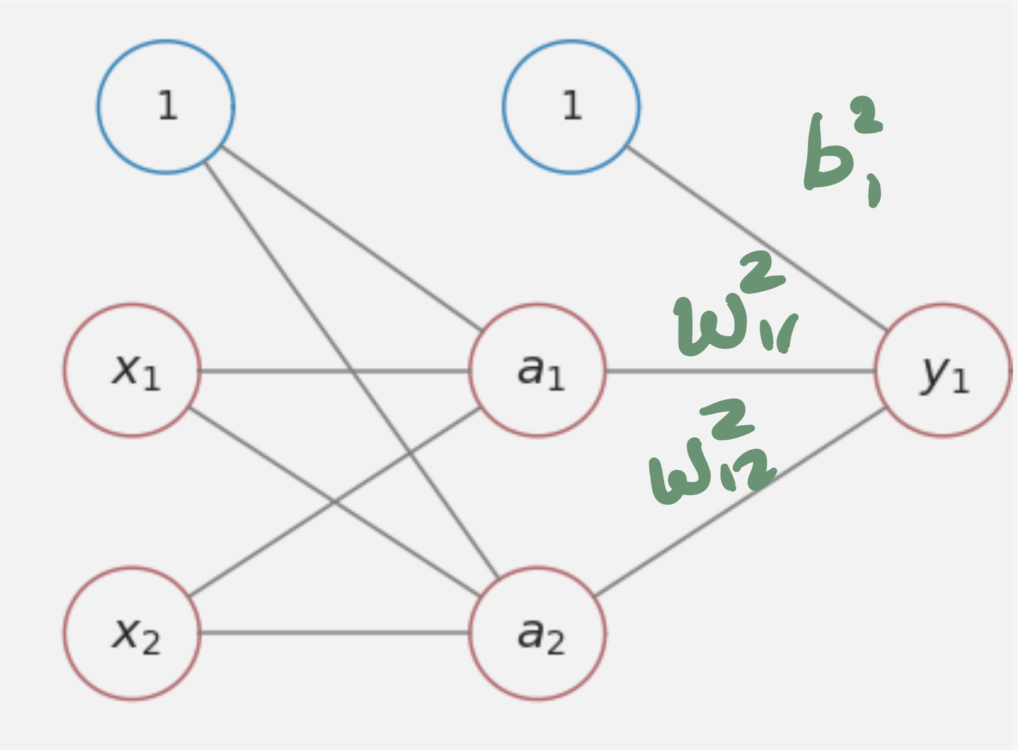
$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$

becomes

$$\begin{aligned} \mathbf{z}^2 &= W^1\mathbf{x}^1 + \mathbf{b}^1 \\ \mathbf{a}^2 &= \mathrm{I}(\mathbf{z}^2 > 0) = \mathrm{I}(W^1\mathbf{x}^1 + \mathbf{b}^1 > 0) \end{aligned}$$

where

$$W^1 = \begin{bmatrix} w^1_{11} & w^1_{12} \\ w^1_{21} & w^1_{22} \end{bmatrix}, \quad \mathbf{b}^1 = \begin{bmatrix} b^1_1 \\ b^1_2 \end{bmatrix}$$



$[w^2_{11} \; w^2_{12}]$

$b^2 = [b^2_1]$

The Multilayer Perceptron

Vectorized Anatomy: Hidden Layer to Output Layer

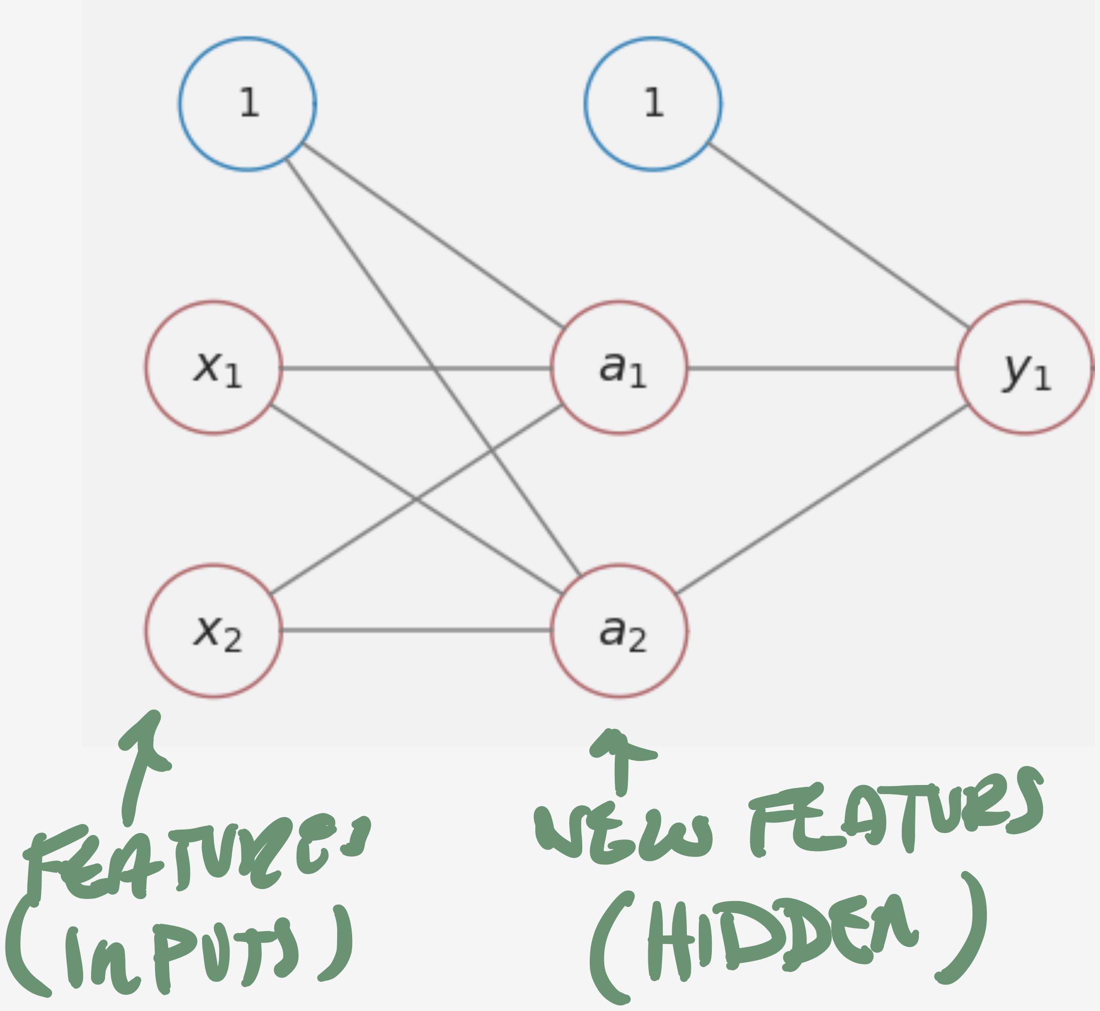
$$
\begin{aligned}
\mathbf{z}^3 &= W^2\mathbf{a}^2 + \mathbf{b}^2 \\
\hat{y} &= \mathrm{I}(\mathbf{z}^3 > 0) = \mathrm{I}(W^2\mathbf{a}^2 + \mathbf{b}^2 > 0)
\end{aligned}
$$

where

$$
W^2 = \begin{bmatrix} w^2_{11} & w^2_{12} \end{bmatrix}, \quad \mathbf{b}^2 = \begin{bmatrix} b^2_1 \end{bmatrix}
$$

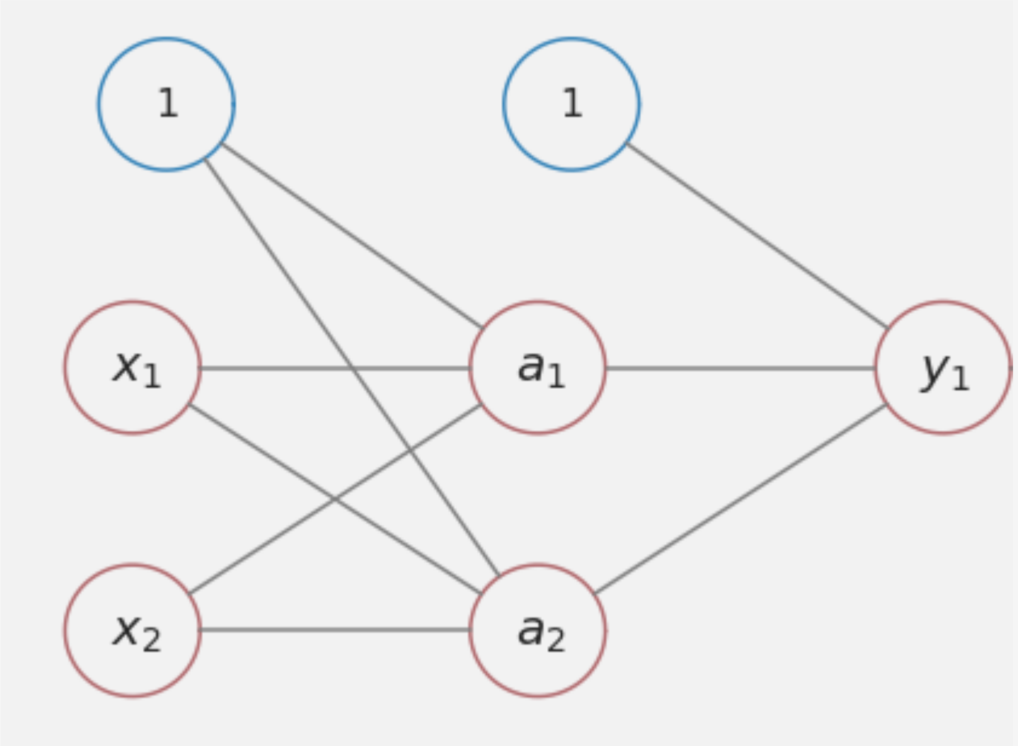
$$z^3 = W^2 a^2 + b^2$$

$$a^3 = \hat{y} = I(z^3 > 0)$$



The Multilayer Perceptron

Example: Can we learn XOR?

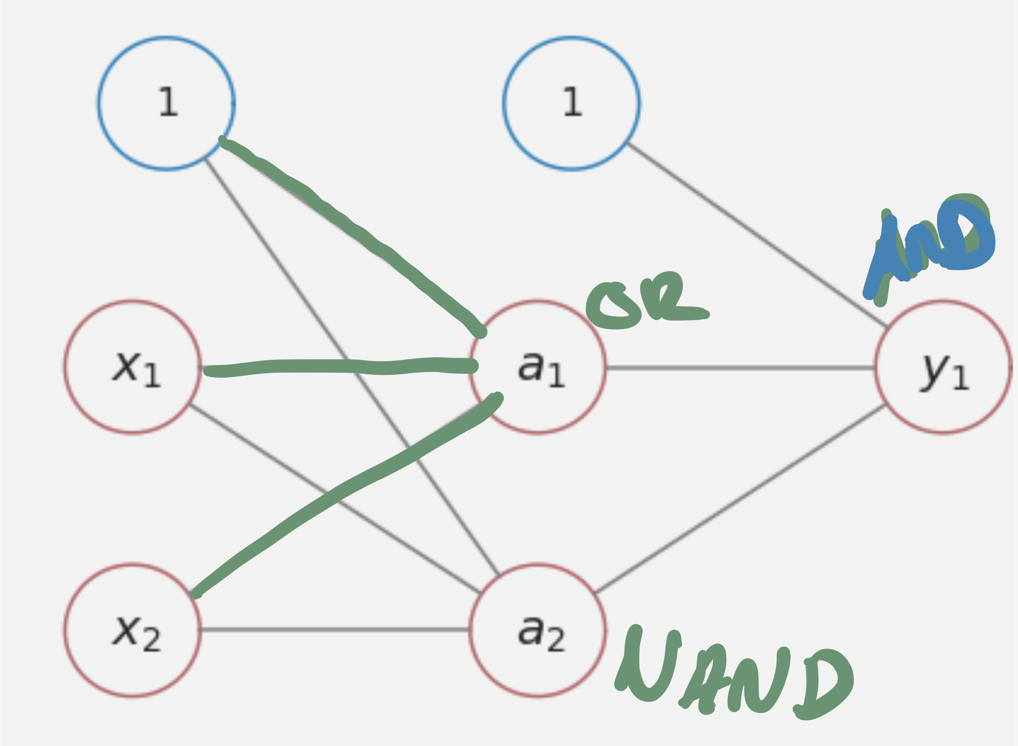


x_1	0	1	0	1
x_2	0	0	1	1
x_1 XOR x_2	0	1	1	0

The Multilayer Perceptron

Example: Can we learn XOR?

x_1	0	1	0	1
x_2	0	0	1	1
$(x_1 \texttt{ OR } x_2) \texttt{ AND } (x_1 \texttt{ NAND } x_2)$	0	1	1	0



OR: NAND: $\begin{bmatrix} 1 & 1 \\ -1 & -1 \end{bmatrix} \begin{bmatrix} 0 \\ 1.5 \end{bmatrix} = W^1, b^1$

AND: $\begin{bmatrix} 1 & 1 \end{bmatrix} \begin{bmatrix} -1 \end{bmatrix}$ W^2 b^2

The Multilayer Perceptron

Example: XOR Check

$$W^1 = \begin{bmatrix} 1 & 1 \\ -1 & -1 \end{bmatrix}, \quad \mathbf{b}^1 = \begin{bmatrix} 0 \\ 1.5 \end{bmatrix} \qquad W^2 = \begin{bmatrix} 1 & 1 \end{bmatrix}, \quad \mathbf{b}^2 = \begin{bmatrix} -1 \end{bmatrix}$$

x_1	0	1	0	1
x_2	0	0	1	1
$(x_1 \ \texttt{OR} \ x_2) \ \texttt{AND} \ (x_1 \ \texttt{NAND} \ x_2)$	0	1	1	0

$I(z^2 > 0)$

$$\mathbf{z}^2 = W^1\mathbf{x}^1 + \mathbf{b}^1 = \begin{bmatrix} 1 & 1 \\ -1 & -1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 1.5 \end{bmatrix} = \begin{bmatrix} 0 \\ 1.5 \end{bmatrix}, \quad \mathbf{a}^2 = \mathrm{I}(\mathbf{z}^2 > 0) = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$\mathbf{z}^3 = W^2\mathbf{a}^2 + \mathbf{b}^2 = \begin{bmatrix} 1 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} + \begin{bmatrix} -1 \end{bmatrix} = \begin{bmatrix} 0 \end{bmatrix}, \quad \hat{y} = \mathrm{I}(\mathbf{z}^3 > 0) = \begin{bmatrix} 0 \end{bmatrix}$$

The Multilayer Perceptron

Example: XOR Check

$$W^1 = \begin{bmatrix} 1 & 1 \\ -1 & -1 \end{bmatrix}, \quad \mathbf{b}^1 = \begin{bmatrix} 0 \\ 1.5 \end{bmatrix} \qquad W^2 = \begin{bmatrix} 1 & 1 \end{bmatrix}, \quad \mathbf{b}^2 = \begin{bmatrix} -1 \end{bmatrix}$$

x_1	0	1	0	1
x_2	0	0	1	1
$(x_1 \texttt{ OR } x_2) \texttt{ AND } (x_1 \texttt{ NAND } x_2)$	0	1	1	0

$$\mathbf{z}^2 = W^1\mathbf{x}^1 + \mathbf{b}^1 = \begin{bmatrix} 1 & 1 \\ -1 & -1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 1.5 \end{bmatrix} = \begin{bmatrix} 1 \\ .5 \end{bmatrix}, \quad \mathbf{a}^2 = \mathrm{I}(\mathbf{z}^2 > 0) = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

$$\mathbf{z}^3 = W^2\mathbf{a}^2 + \mathbf{b}^2 = \begin{bmatrix} 1 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} + \begin{bmatrix} -1 \end{bmatrix} = \begin{bmatrix} 1 \end{bmatrix}, \quad \hat{y} = \mathrm{I}(\mathbf{z}^3 > 0) = \begin{bmatrix} 1 \end{bmatrix}$$

The Multilayer Perceptron

Example: XOR Check

$$W^1 = \begin{bmatrix} 1 & 1 \\ -1 & -1 \end{bmatrix}, \quad \mathbf{b}^1 = \begin{bmatrix} 0 \\ 1.5 \end{bmatrix} \qquad W^2 = \begin{bmatrix} 1 & 1 \end{bmatrix}, \quad \mathbf{b}^2 = \begin{bmatrix} -1 \end{bmatrix}$$

x_1	0	1	0	1
x_2	0	0	1	1
$(x_1 \texttt{ OR } x_2) \texttt{ AND } (x_1 \texttt{ NAND } x_2)$	0	1	1	0

$$\mathbf{z}^2 = W^1\mathbf{x}^1 + \mathbf{b}^1 = \begin{bmatrix} 1 & 1 \\ -1 & -1 \end{bmatrix}\begin{bmatrix} \quad \end{bmatrix} + \begin{bmatrix} 0 \\ 1.5 \end{bmatrix} = \begin{bmatrix} \quad \end{bmatrix}, \quad \mathbf{a}^2 = \mathrm{I}(\mathbf{z}^2 > 0) = \begin{bmatrix} \quad \end{bmatrix}$$

$$\mathbf{z}^3 = W^2\mathbf{a}^2 + \mathbf{b}^2 = \begin{bmatrix} 1 & 1 \end{bmatrix}\begin{bmatrix} \quad \end{bmatrix} + \begin{bmatrix} -1 \end{bmatrix} = \begin{bmatrix} \quad \end{bmatrix}, \quad \hat{y} = \mathrm{I}(\mathbf{z}^3 > 0) = \begin{bmatrix} \quad \end{bmatrix}$$

The Multilayer Perceptron

Example: XOR Check

$$W^1 = \begin{bmatrix} 1 & 1 \\ -1 & -1 \end{bmatrix}, \quad \mathbf{b}^1 = \begin{bmatrix} 0 \\ 1.5 \end{bmatrix} \qquad W^2 = \begin{bmatrix} 1 & 1 \end{bmatrix}, \quad \mathbf{b}^2 = \begin{bmatrix} -1 \end{bmatrix}$$

x_1	0	1	0	1
x_2	0	0	1	1
$(x_1 \ \texttt{OR} \ x_2) \ \texttt{AND} \ (x_1 \ \texttt{NAND} \ x_2)$	0	1	1	0

$$\mathbf{z}^2 = W^1\mathbf{x}^1 + \mathbf{b}^1 = \begin{bmatrix} 1 & 1 \\ -1 & -1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} + \begin{bmatrix} 0 \\ 1.5 \end{bmatrix} = \begin{bmatrix} 2 \\ -.5 \end{bmatrix}, \quad \mathbf{a}^2 = \mathrm{I}(\mathbf{z}^2 > 0) = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$$\mathbf{z}^3 = W^2\mathbf{a}^2 + \mathbf{b}^2 = \begin{bmatrix} 1 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} + \begin{bmatrix} -1 \end{bmatrix} = \begin{bmatrix} 0 \end{bmatrix}, \quad \hat{y} = \mathrm{I}(\mathbf{z}^3 > 0) = \begin{bmatrix} 0 \end{bmatrix}$$

The Multilayer Perceptron

OK, so the multilayer perceptron is cool and all, but when do we get to REAL neural networks??

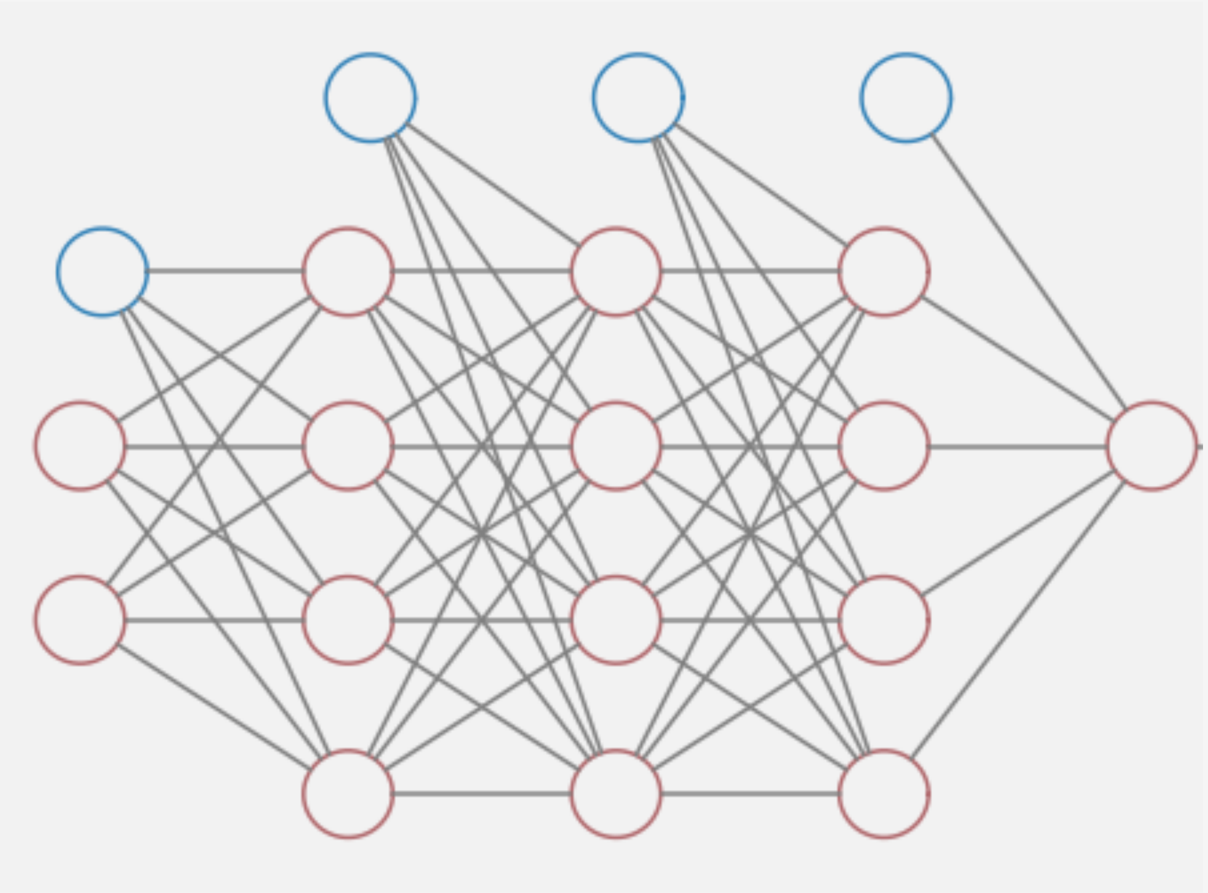
The Multilayer Perceptron

OK, so the multilayer perceptron is cool and all, but when do we get to REAL neural networks??

We're pretty much already there.

Things we'll explore moving forward:

- Non-Binary Features
- Better activation functions
- How to choose architectures
- How to train these things
- Regression or Classification (Answer: Yes)



Neural Networks I Wrap-Up

- Simplest network is just a regular old perceptron (a linear classifier)
- We can learn non-linear decision boundaries by chaining perceptrons
- We can represent these complicated interactions using linear algebra

Next Time:

- The Feed-Forward Neural Network