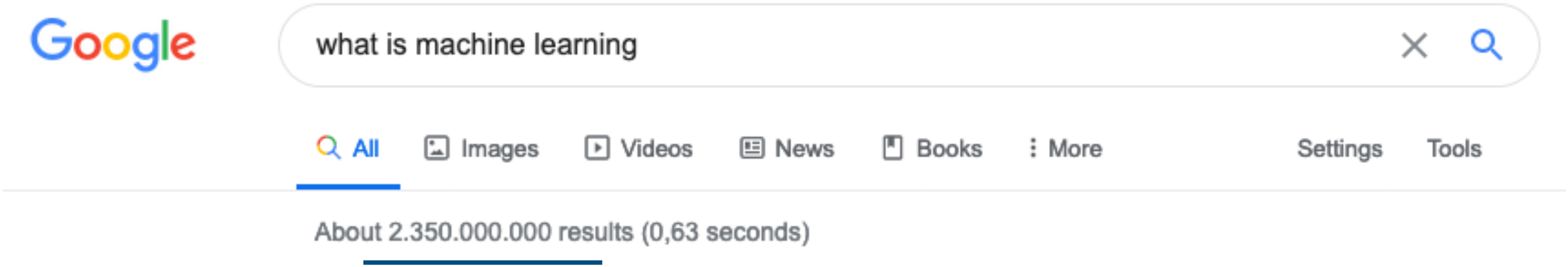
# Neural Networks

*Maxwell Cai, PhD*

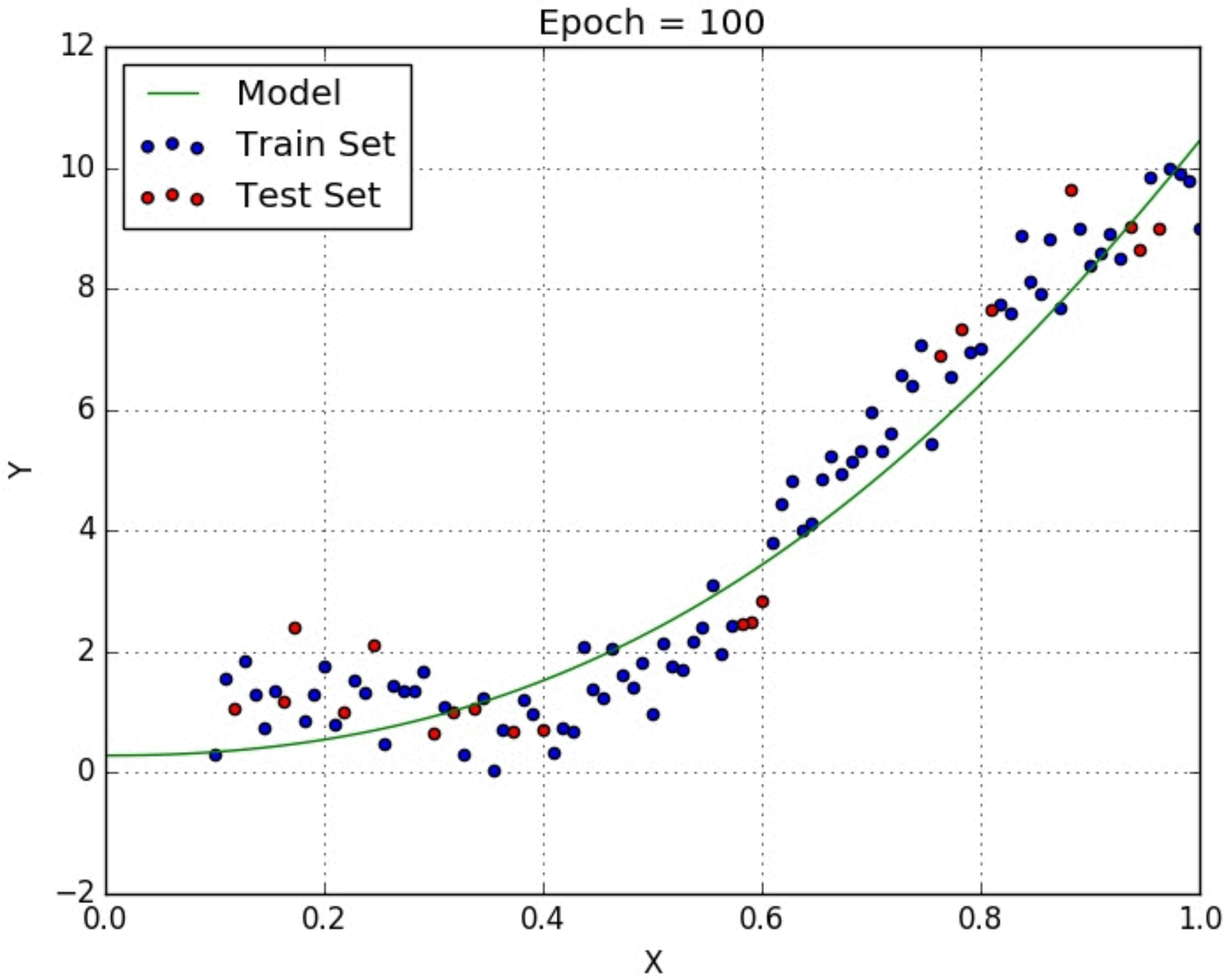
# Let’s get started!

# What *is* machine learning?

**Search: “what is machine learning”**



# What *is* machine learning?



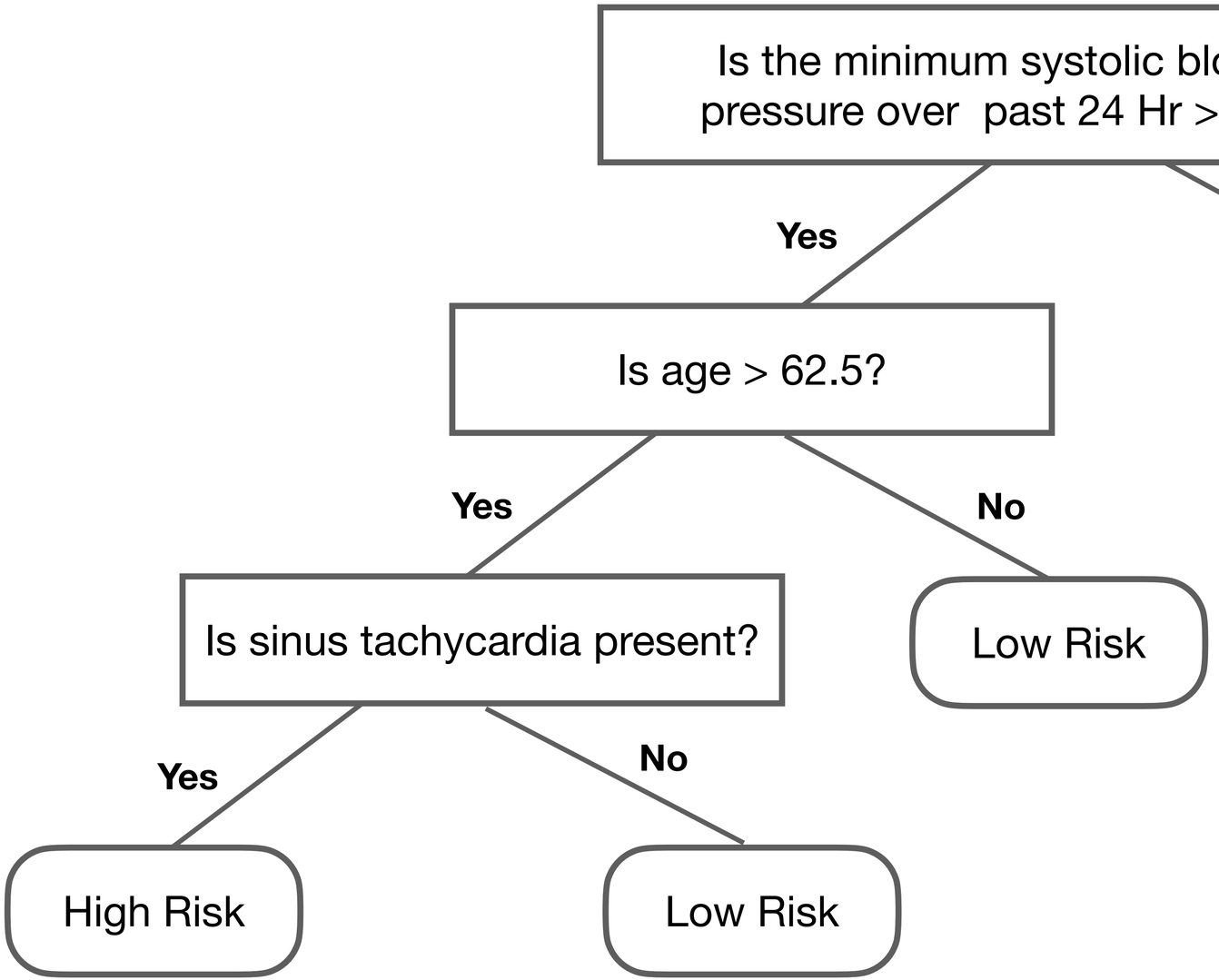

**Could be as simple as curve fitting!**

$$f(x) = ax^2 + bx + c$$

The machine learns to figure out these parameters.

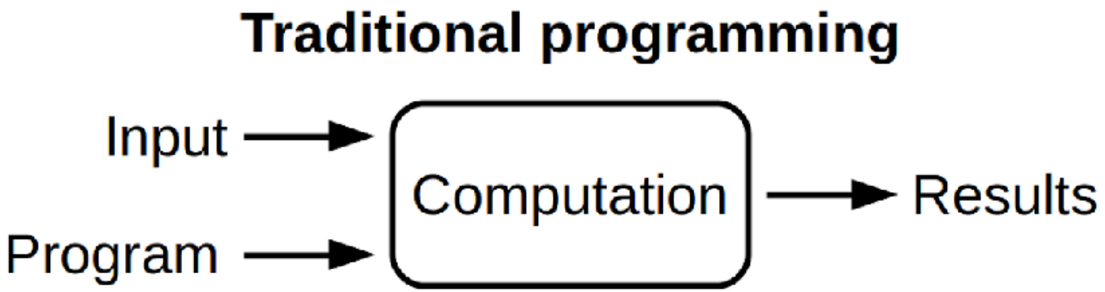
# Why machine learning?

Think about a simple **decision tree**:



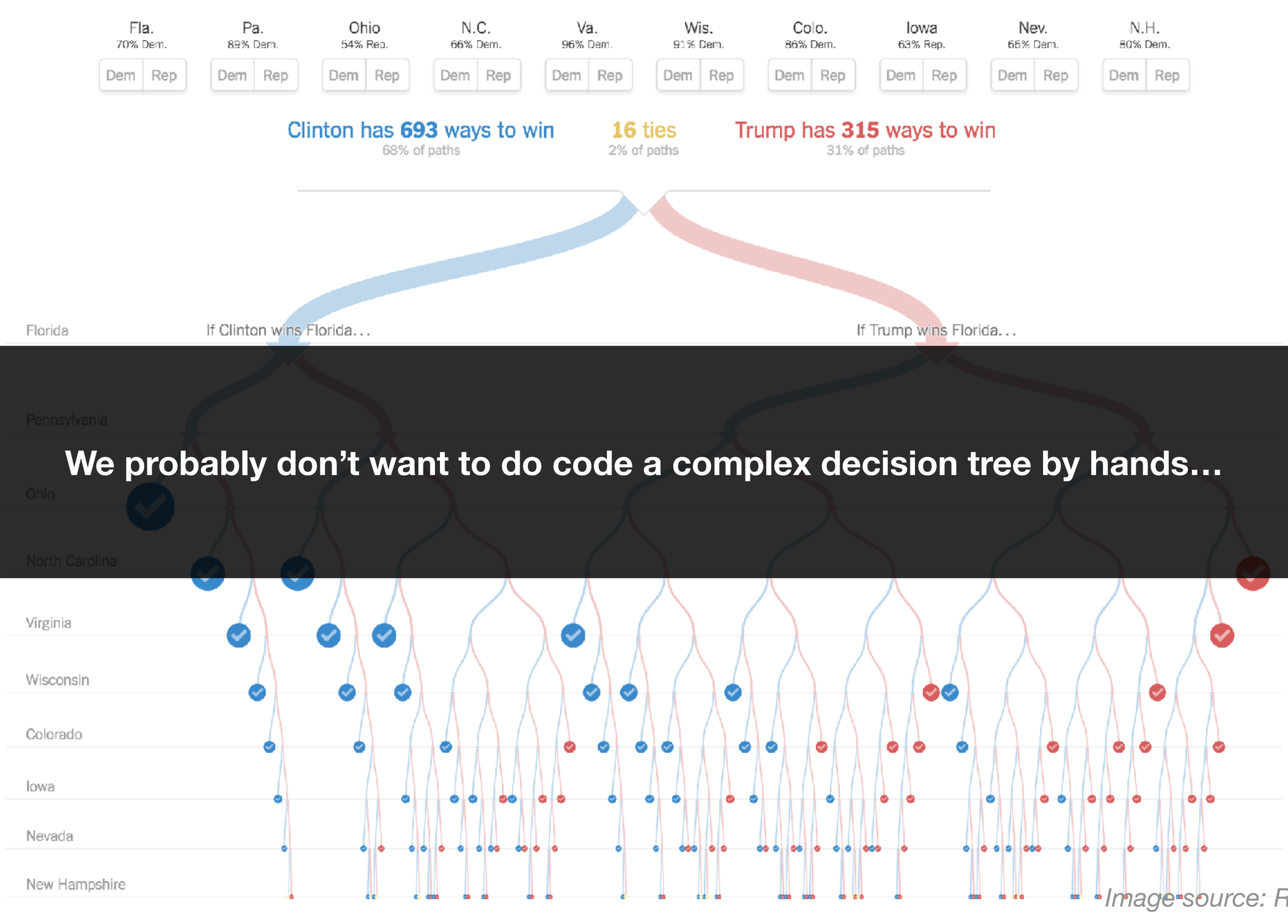

```
if blood_pressure > 91:
    if age > 62.5:
        if sinus_tach:
            ...
        else:
            ...
else:
    ...
```

# A traditional way of programming




Knowledge base & rules
Expert systems
**Human input**

*Source: Machine Learning for an Expert System to Predict Preterm Birth Risk , Woolery et al. (1994)*

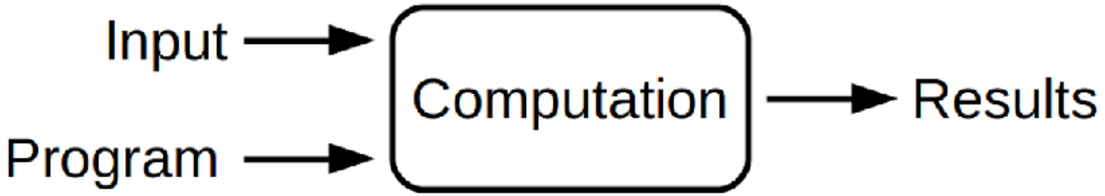



**We probably don’t want to do code a complex decision tree by hands…**

*Image source: Rahul/Medium*

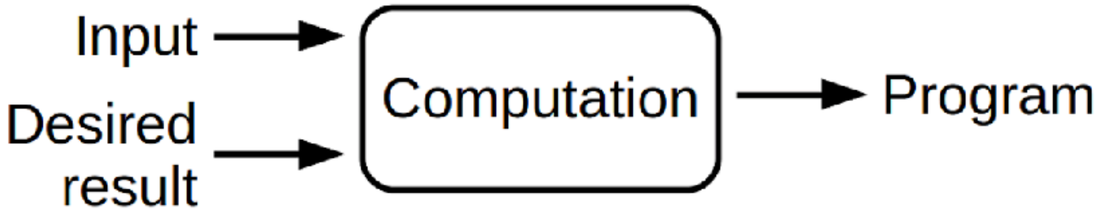
# Machine learning: A new way of programming

**Traditional programming** **Driven by experience**




Knowledge base & rules
Expert systems
**Human input**

**Machine learning** **Driven by data**




Features and result
'Decision' system
**Limited human input**

*Source: Machine Learning for an Expert System to Predict Preterm Birth Risk , Woolery et al. (1994)*

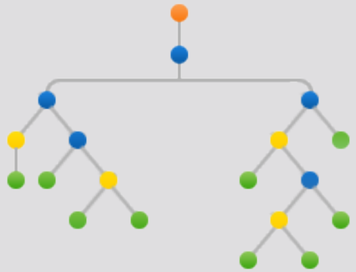
# AI vs ML vs DL

## Artificial Intelligence

*Having computers to mimic human behaviours*

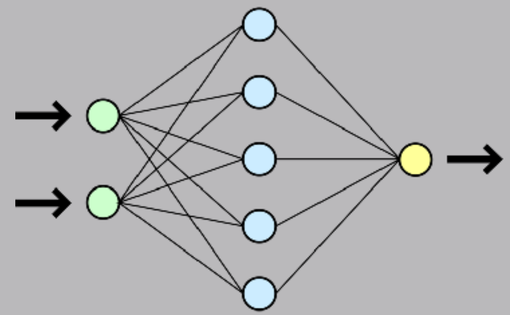
## Machine learning

*Perform tasks without explicitly programmed*



## Deep learning

*Use (deep) neural networks to preform tasks without explicitly programmed*

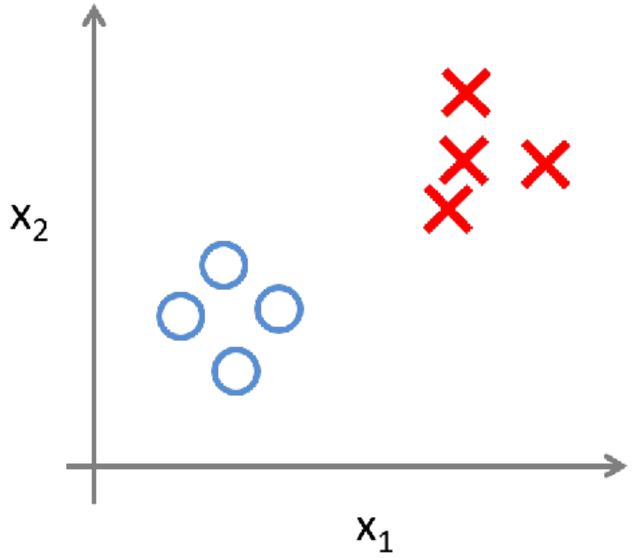


# Categories of machine learning

## Supervised

**Learn from the labels**

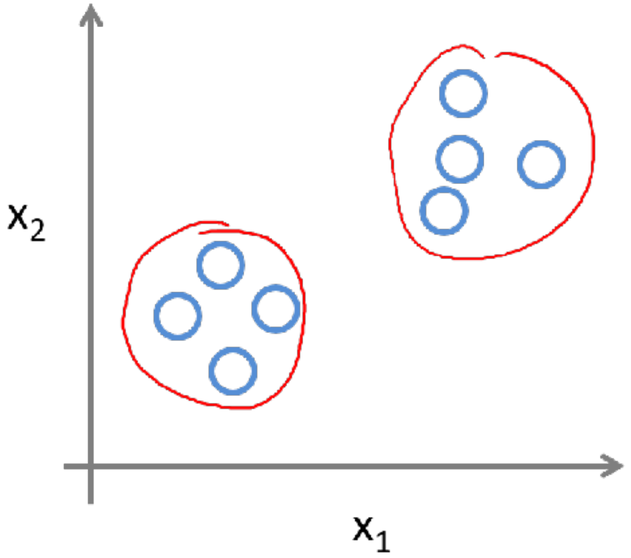
Regression, Classification




## Unsupervised

**Detect patterns in the data**

Clustering, dimensionality reduction

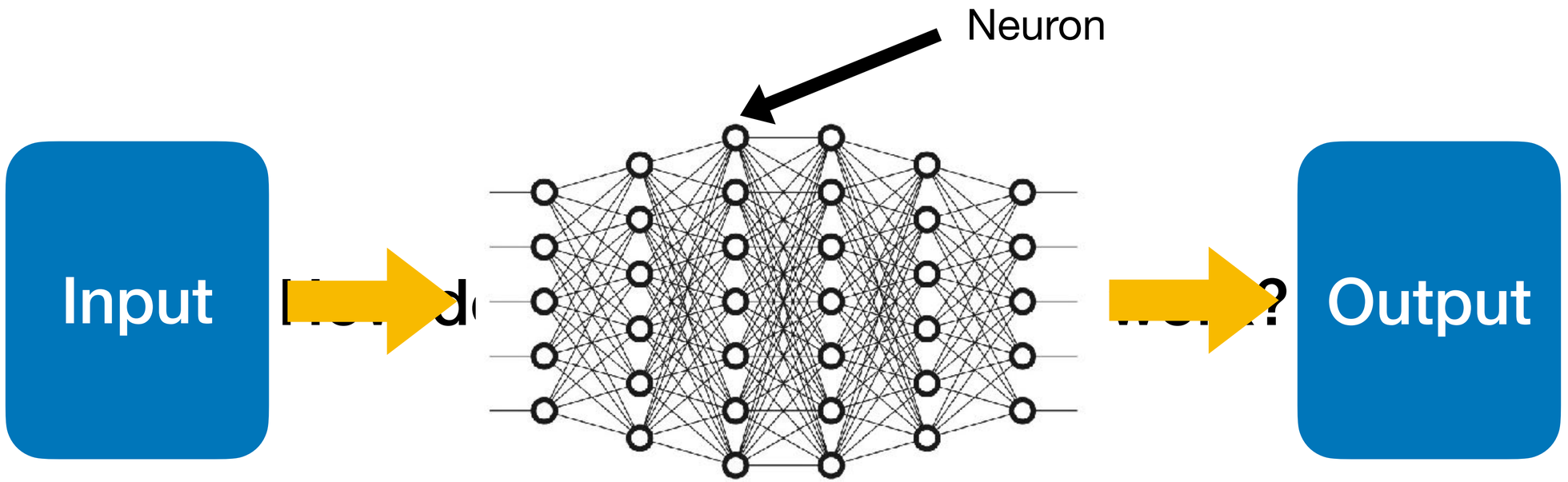



## Reinforcement

**Learn from mistakes**

Control, gaming

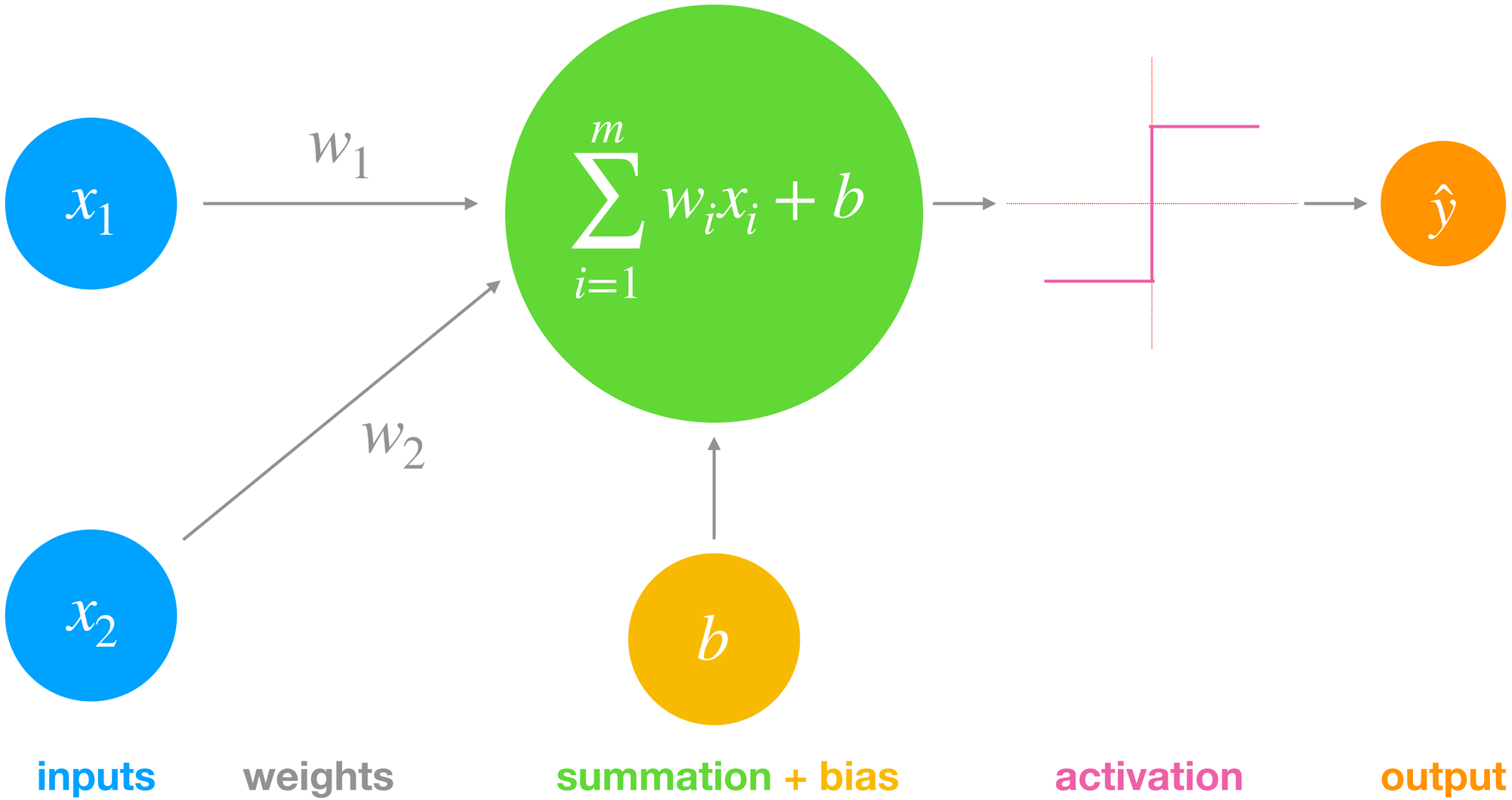
# How does a neural network work?




$$\hat{y} = f_{\text{NN}}(x_1, x_2, \ldots, x_n)$$

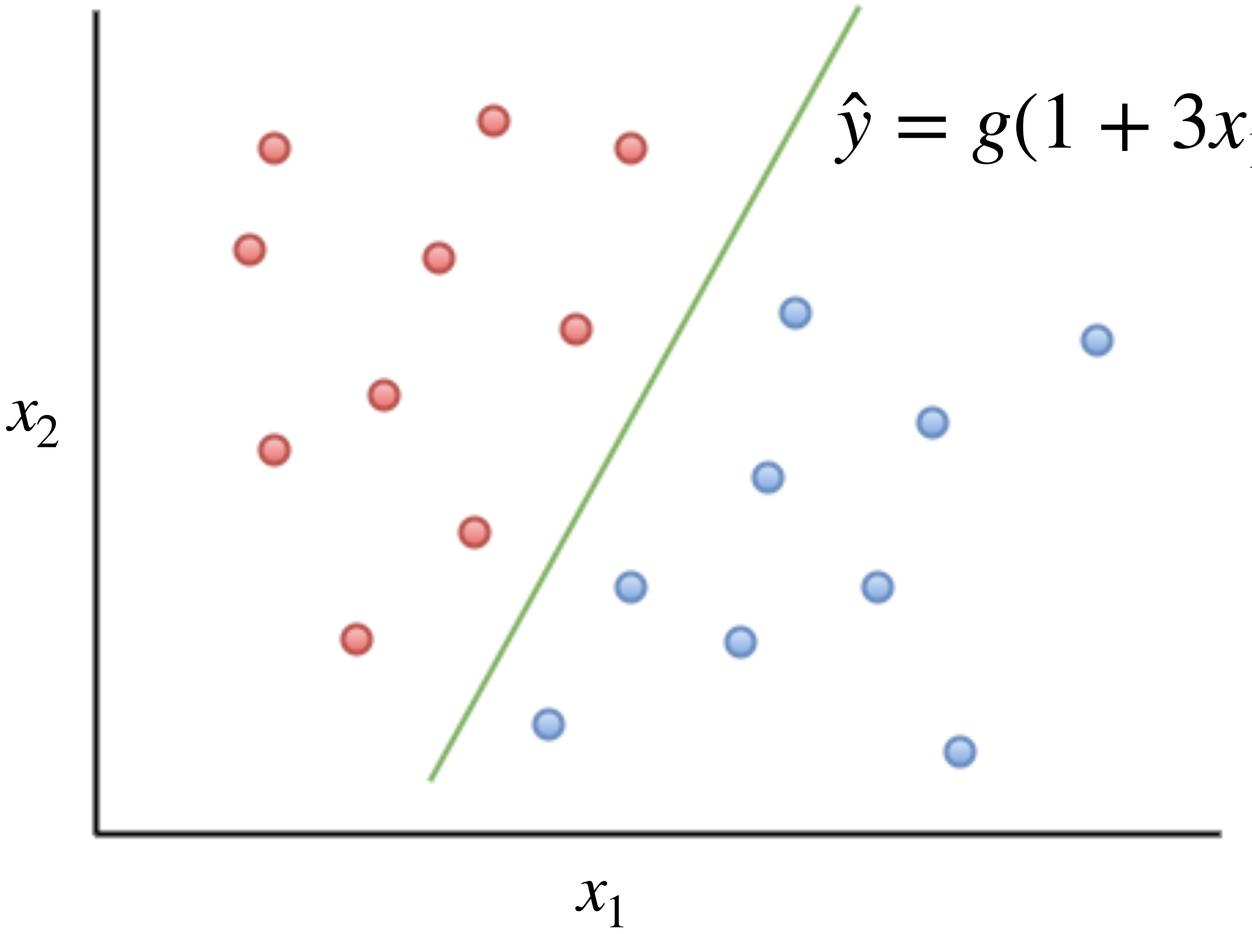
**Basic idea:** use a (deep) neural network to **approximate** an unknown **function**.

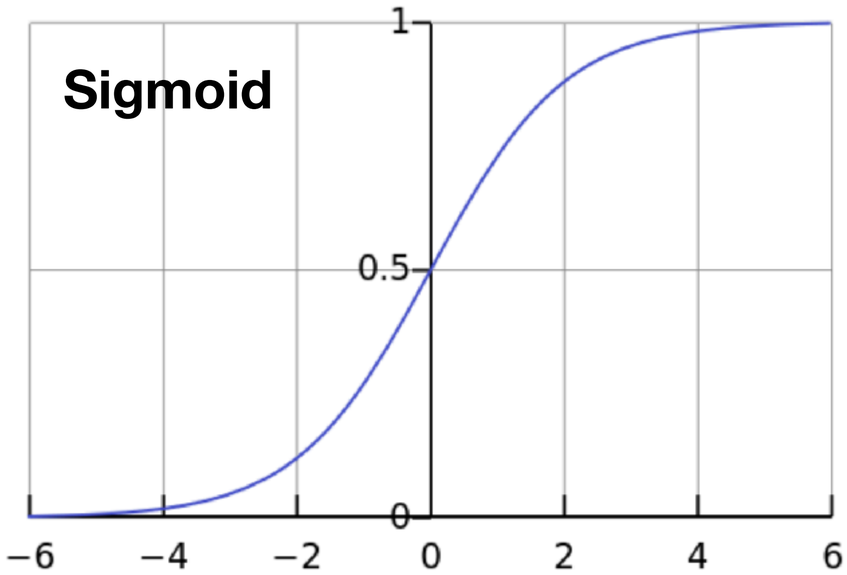
# Anatomy of a Neuron

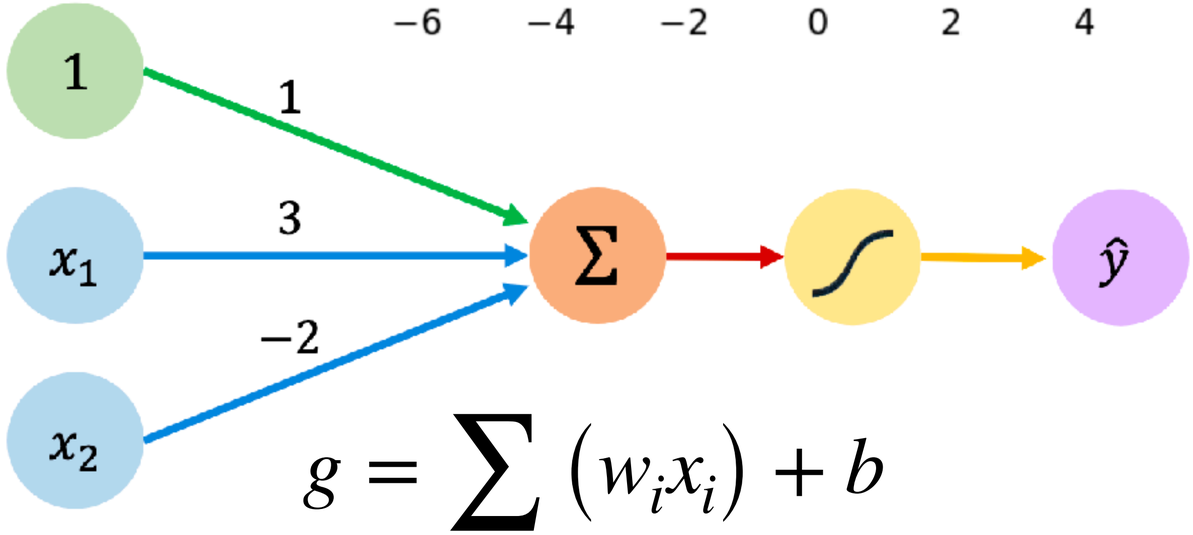


# Neurons/Perceptrons & Activation Functions

Example: Consider a simple **binary classification** task:

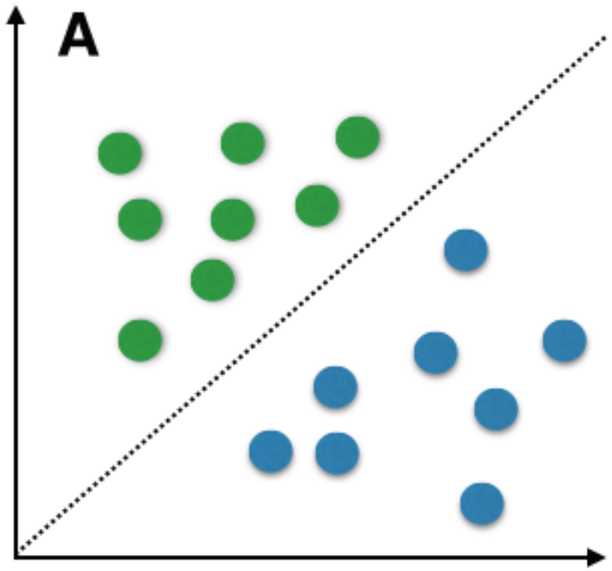


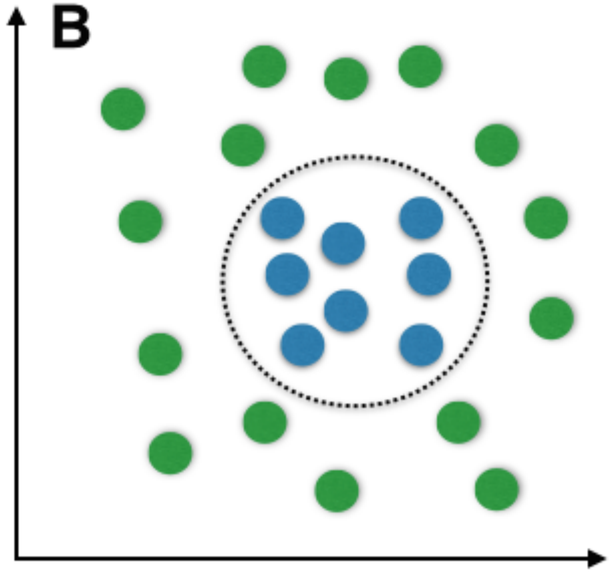



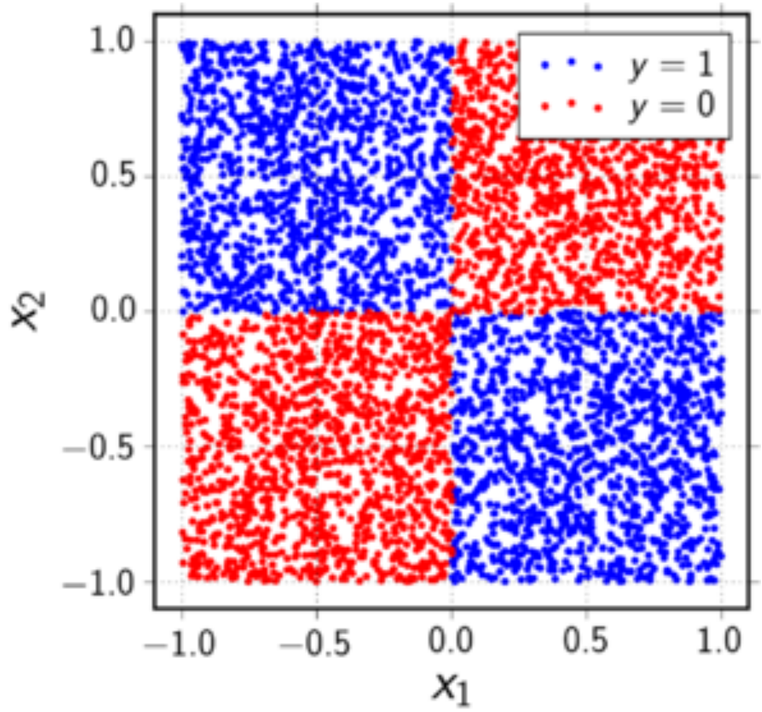




$$g = \sum (w_i x_i) + b$$

# Limitation of a Linear Single-layer Classifier



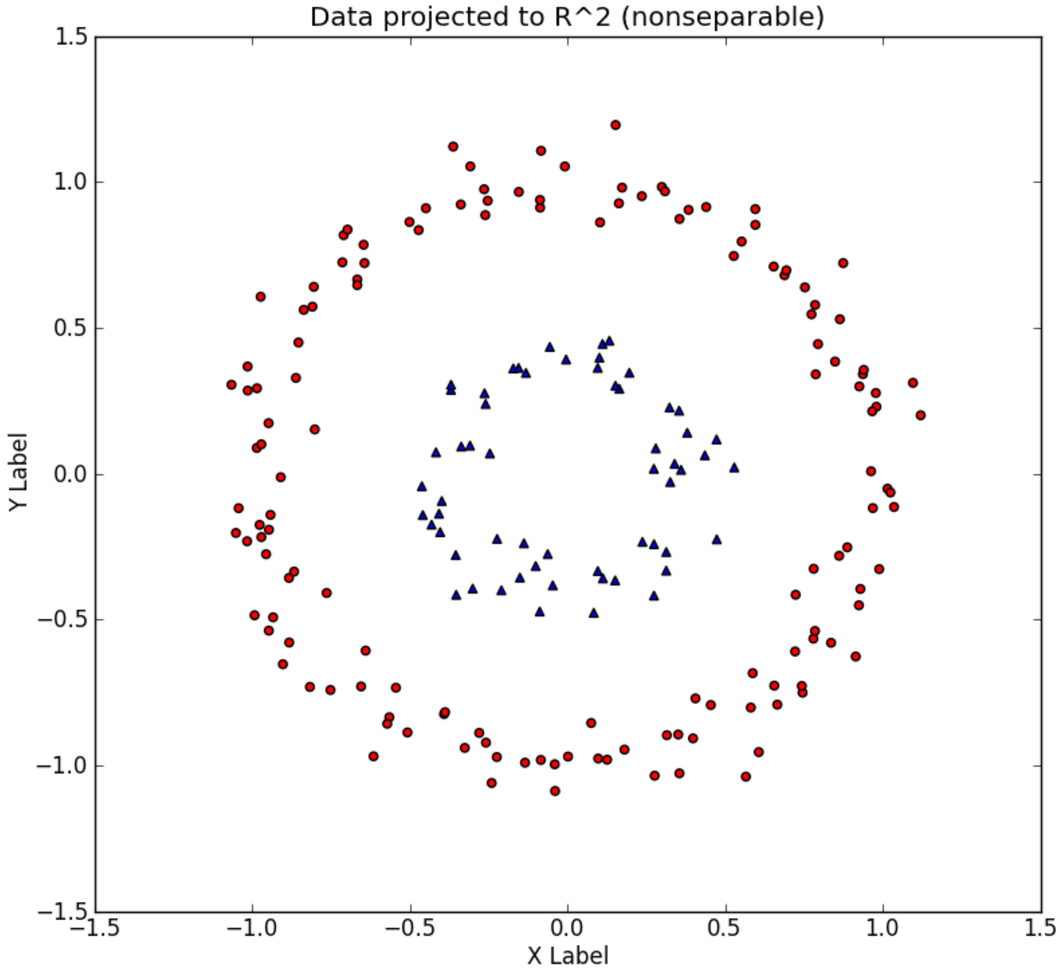



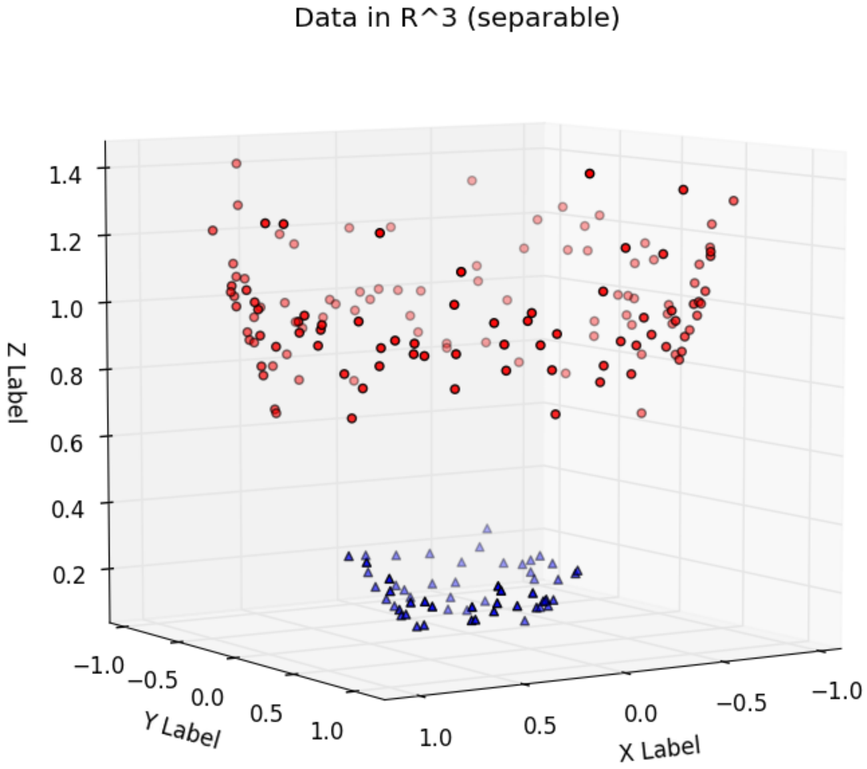

**XOR Problem**

**Possible solutions:**

- map into another space (kernel trick)
- Add more layers (deep learning)

# Kernel Trick

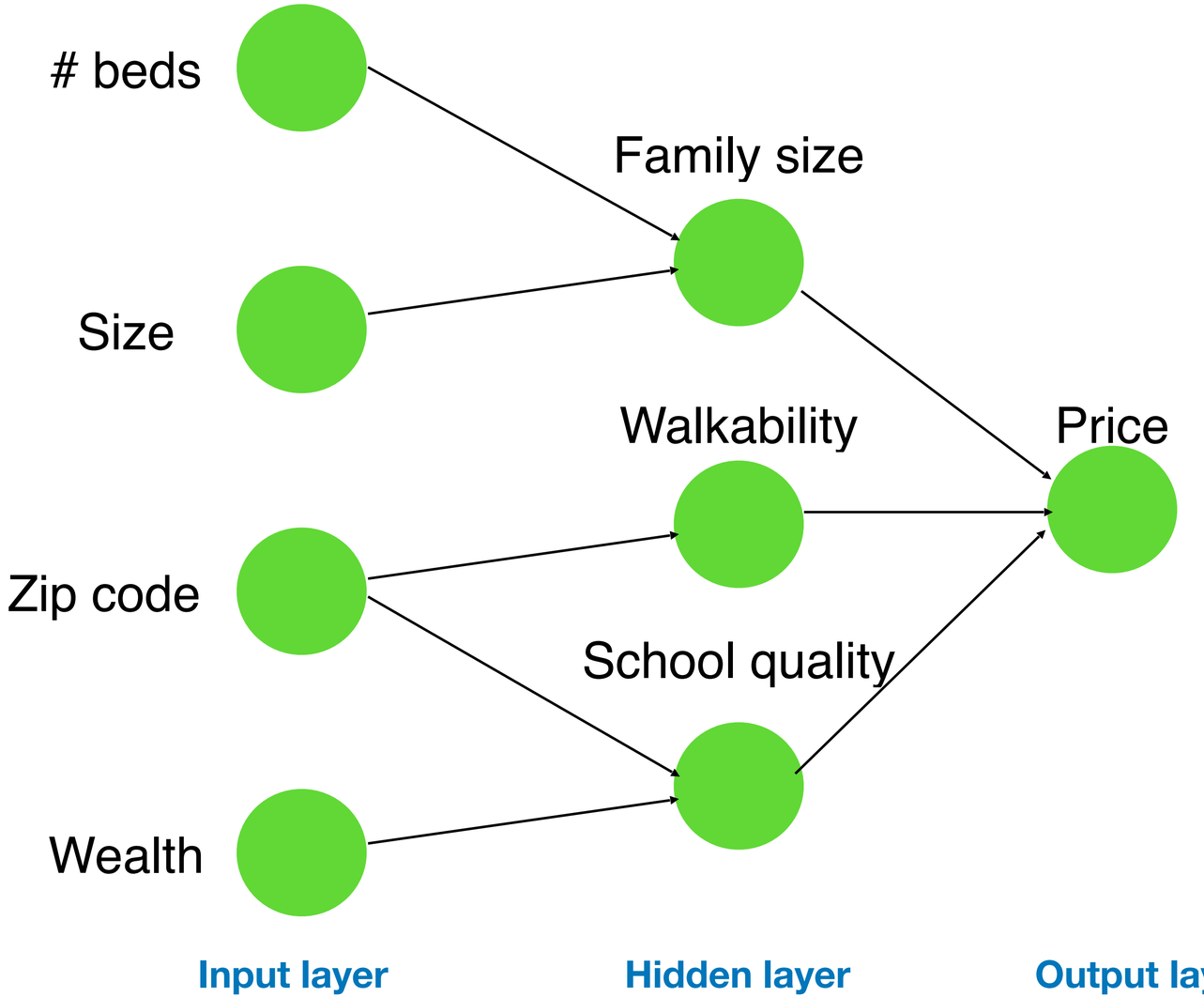


Or, add one more layer?

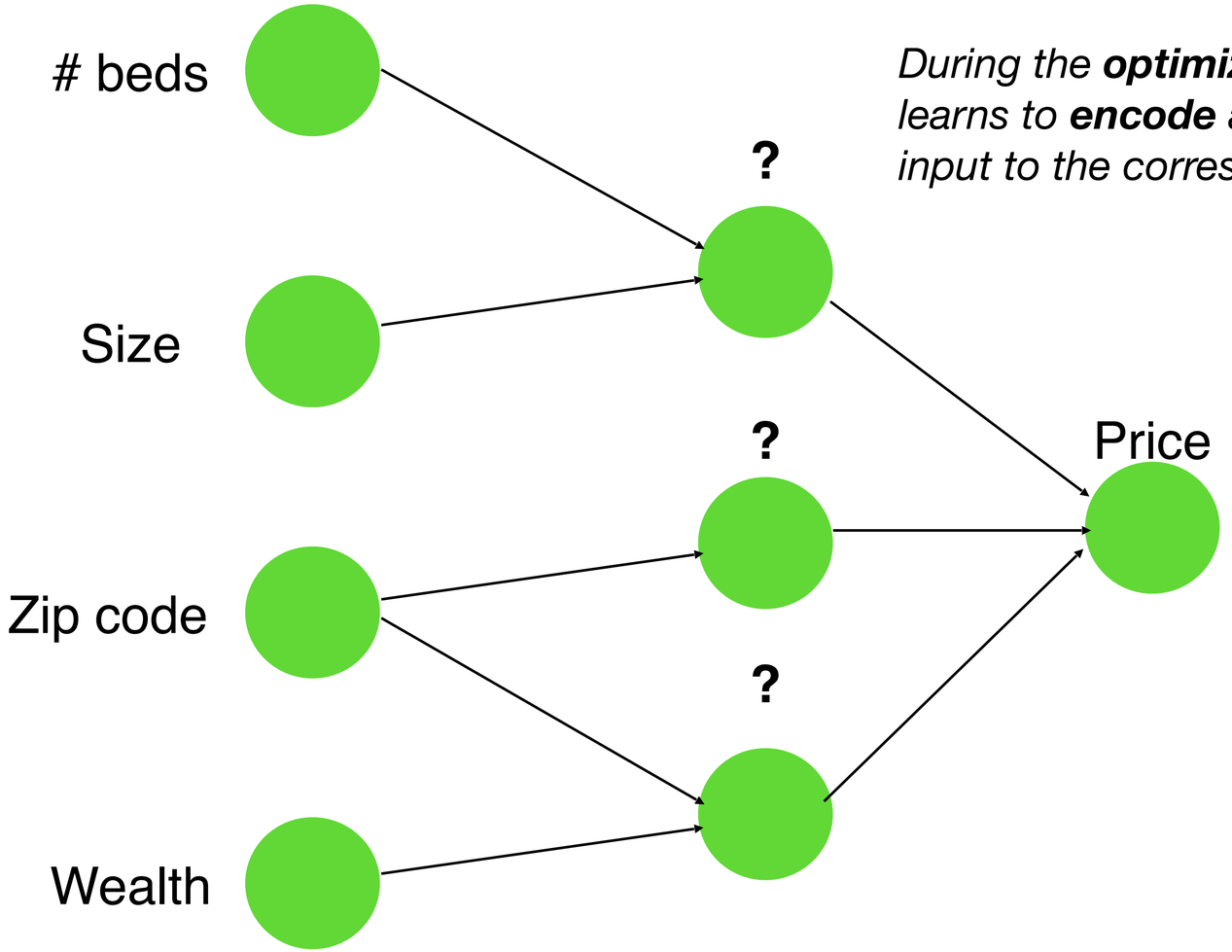
# Intuition of a multi-layer neural network

**Example**: house price prediction model (designed by humans)



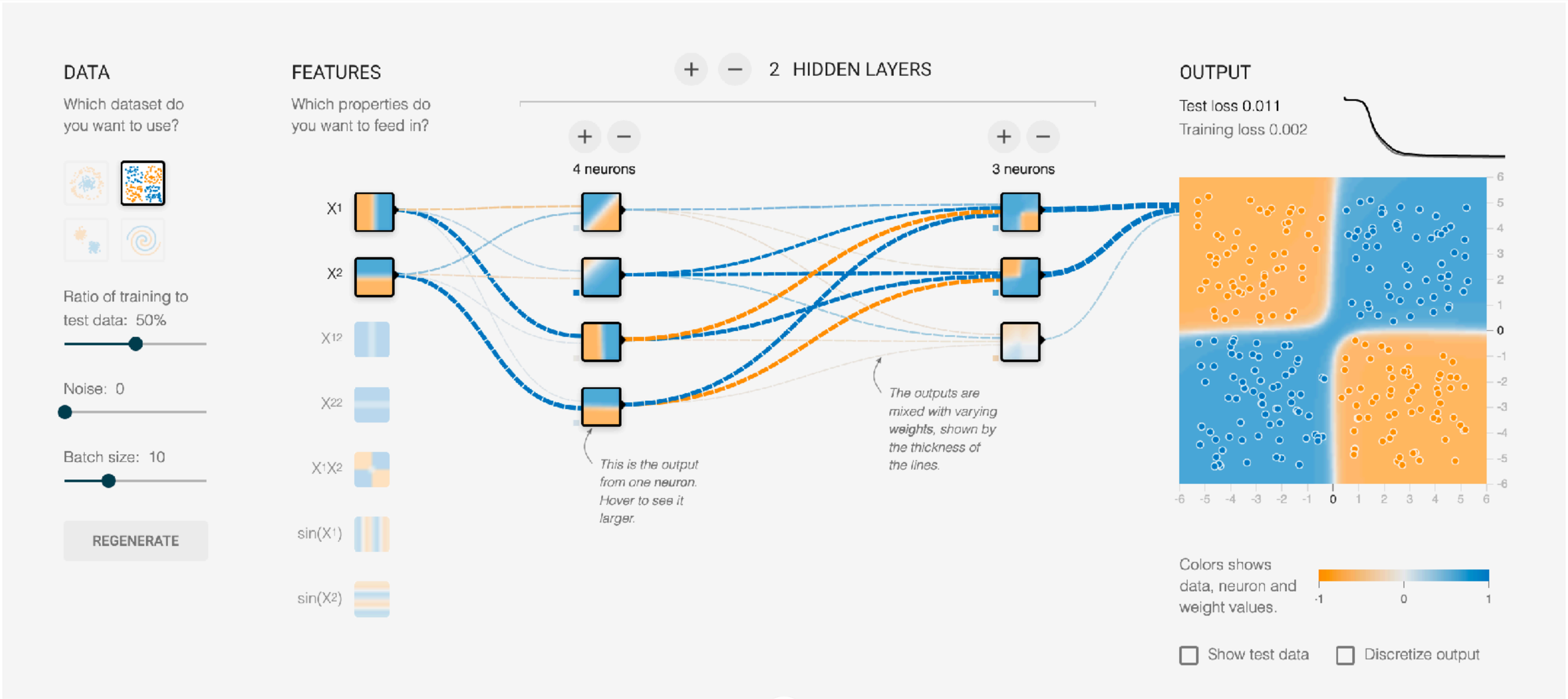
# Intuition of a multi-layer neural network

**Example**: house price prediction model (designed by machines)




*During the **optimization** process, the machine learns to **encode** a presentation that **maps** the input to the corresponding output.*

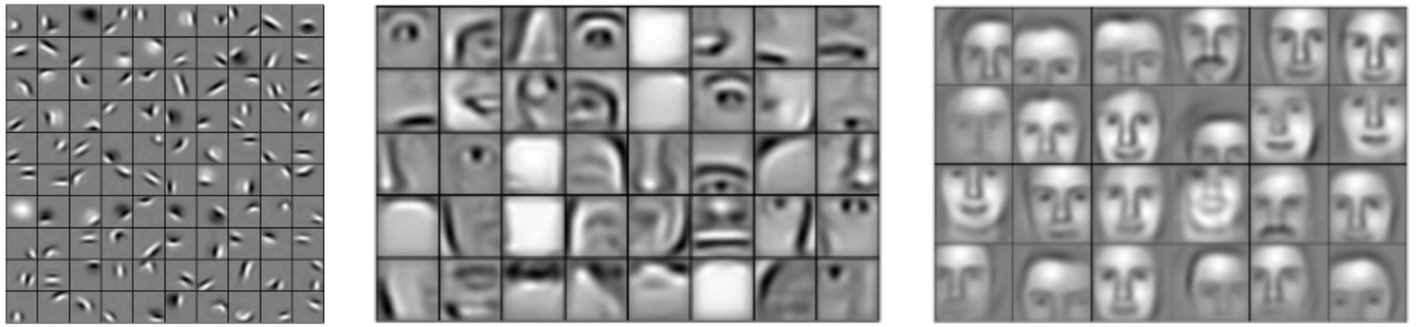
# Live demo: Multi-layer Perceptrons

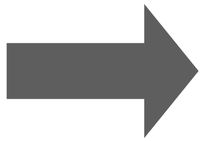


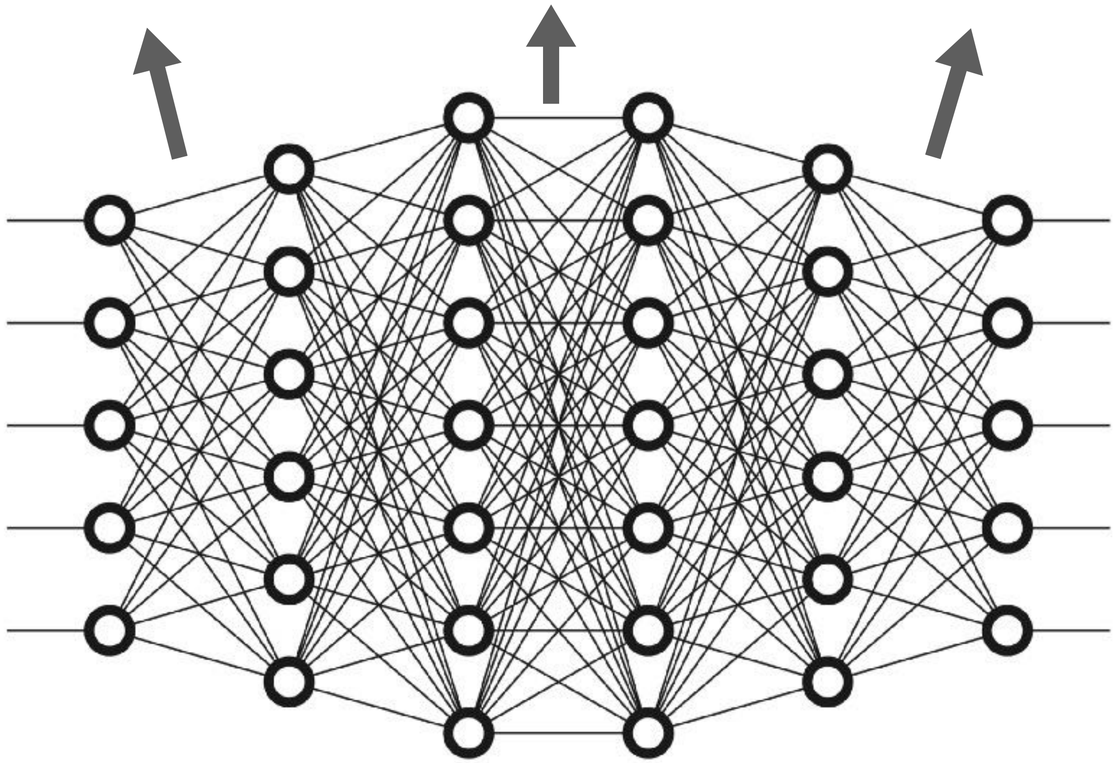

https://playground.tensorflow.org

# A DNN encodes the representation hierarchically

A deep neural network **encodes** the **representation** in an increasingly abstract way.







# What is a machine actually *learning*?

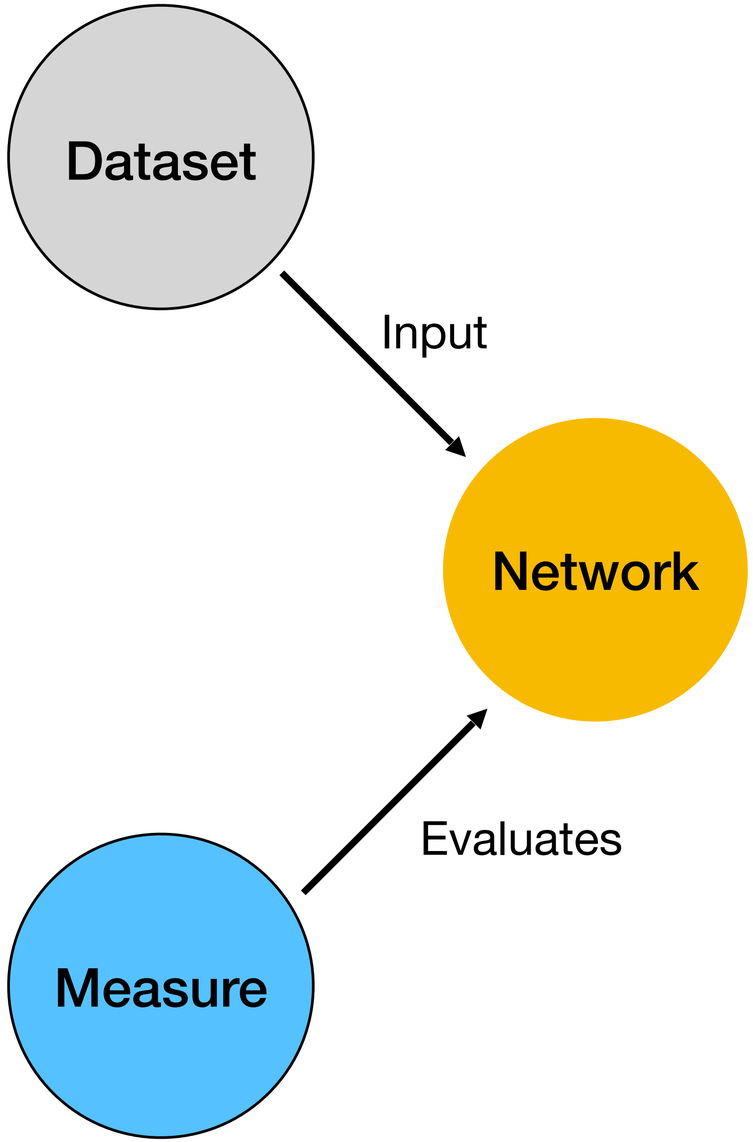
Machine learning is an **optimization** process.

# The loss function (for supervised learning)

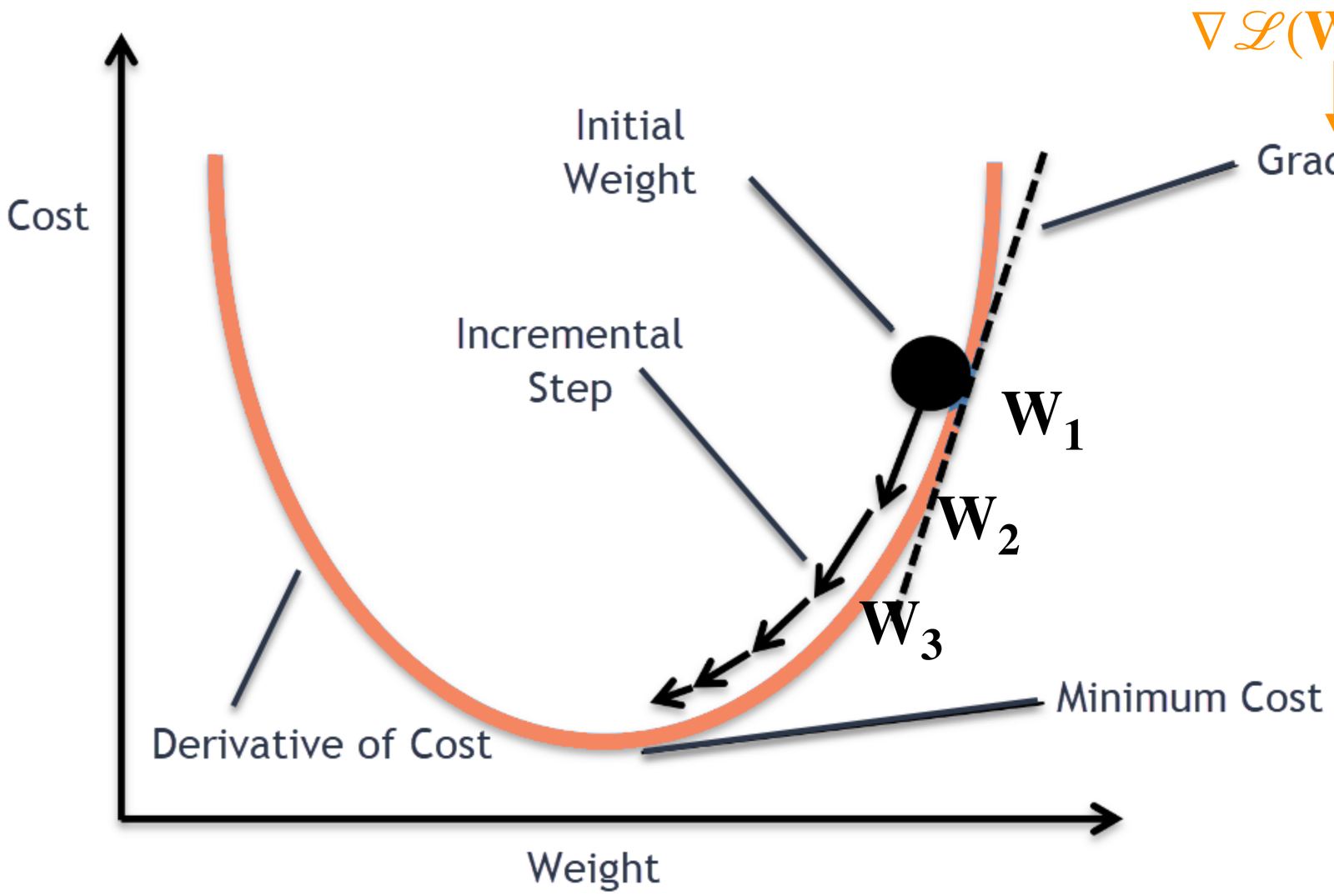
**Estimate:** $$\hat{y} = f_{\text{NN}}(x_1, x_2, \ldots, x_n)$$

**Loss:** $$\mathscr{L}(y, \hat{y}) = L(\mathbf{W}, b) = \frac{1}{N} \sum_{i=1}^{N} \left( y_i - \hat{y}_i \right)^2$$

**Ground truth:** $$\mathbf{x} = (x_1, x_2, \ldots, x_n), y$$



# Optimization fo the loss function

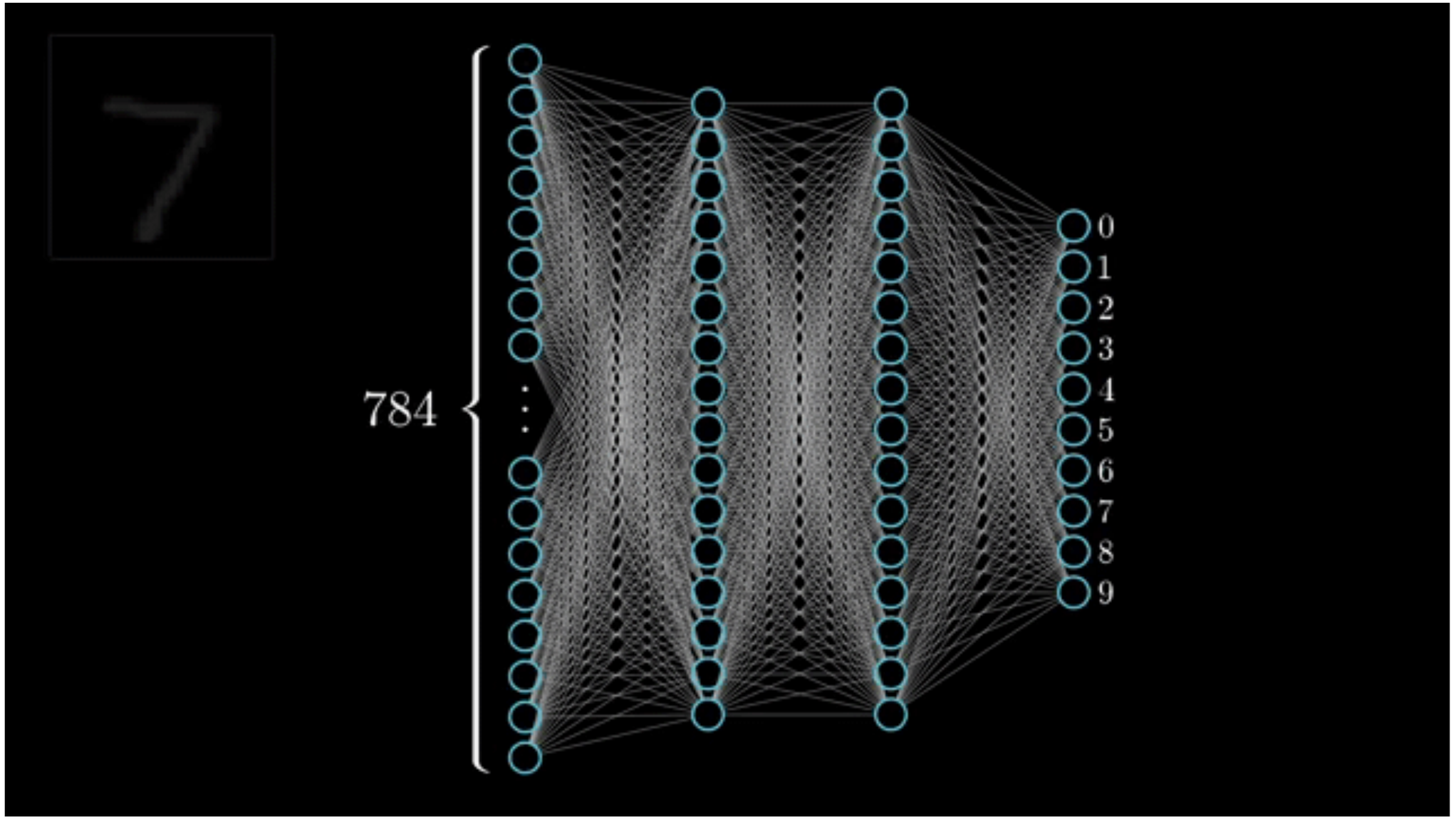



$$\mathscr{L}(\mathbf{W}, b) = \frac{1}{N}\sum_{i=1}^{N}\left(y_i - \hat{y}_i\right)^2$$

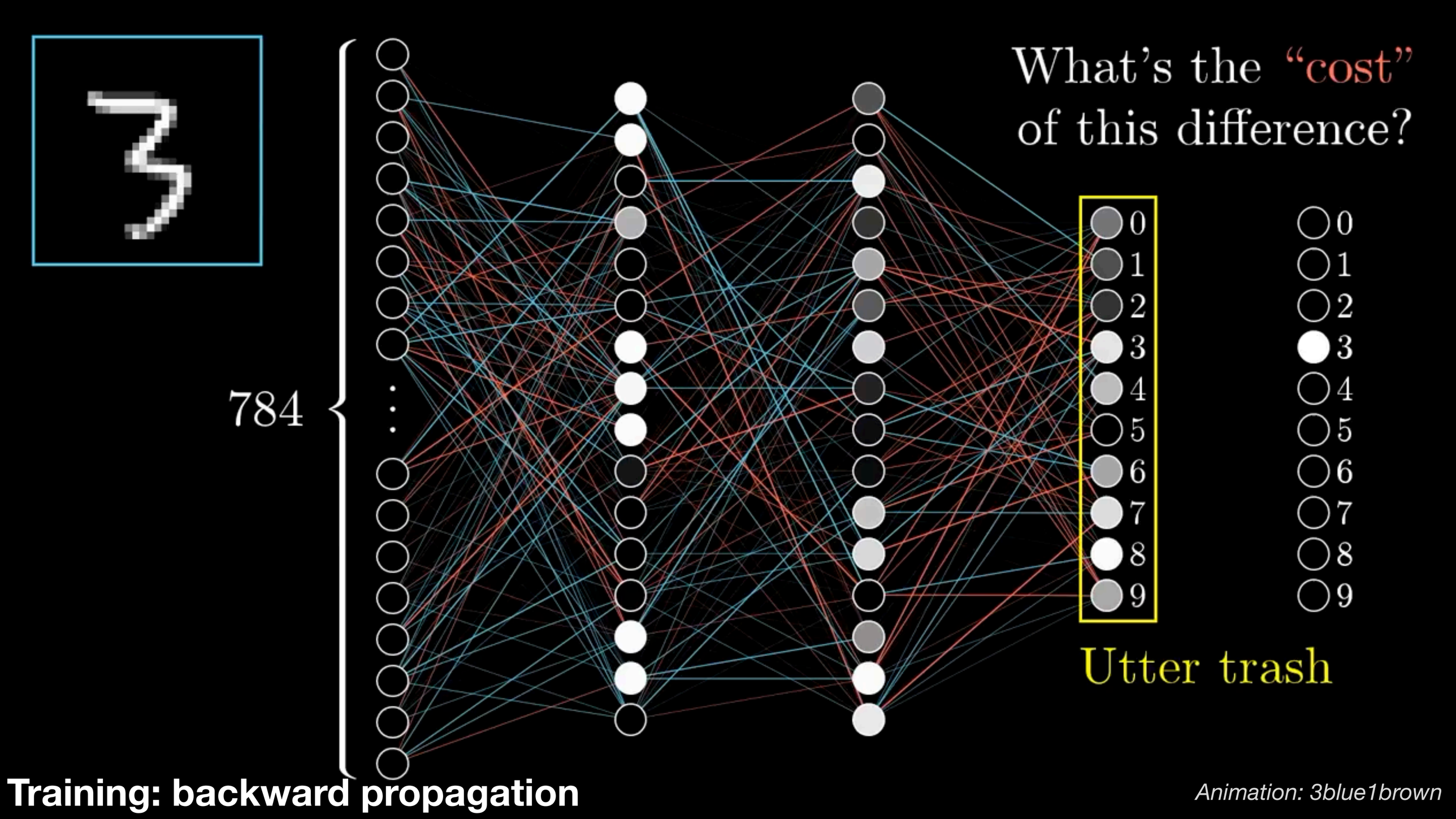
$$\mathbf{W}_{j+1} = W_j - \alpha \nabla \mathscr{L}(\mathbf{W}_j, b)$$

Learning rate

# Prediction: forward propagation

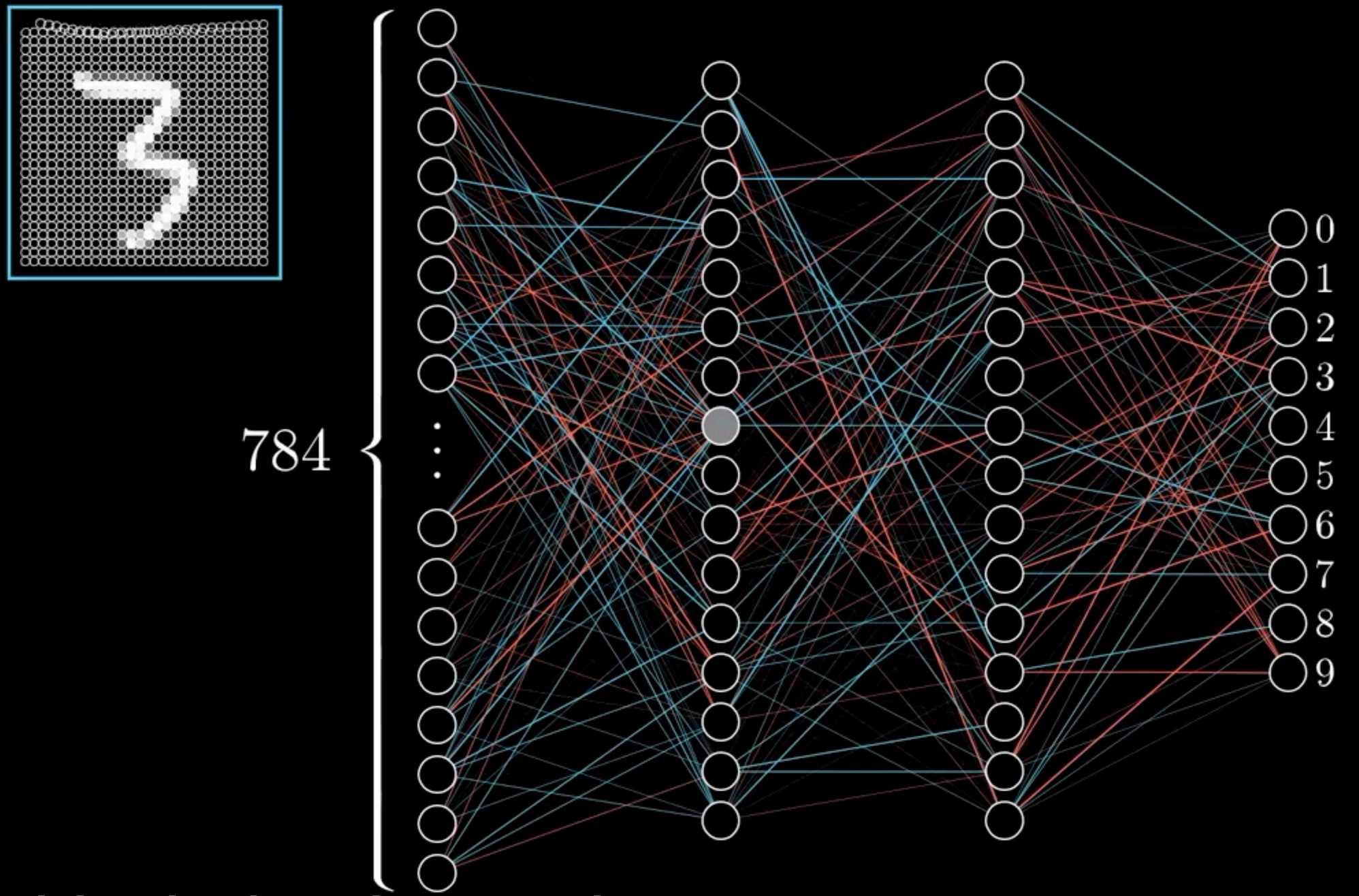



*Source: 3blue1brown*




**Training: backward propagation**

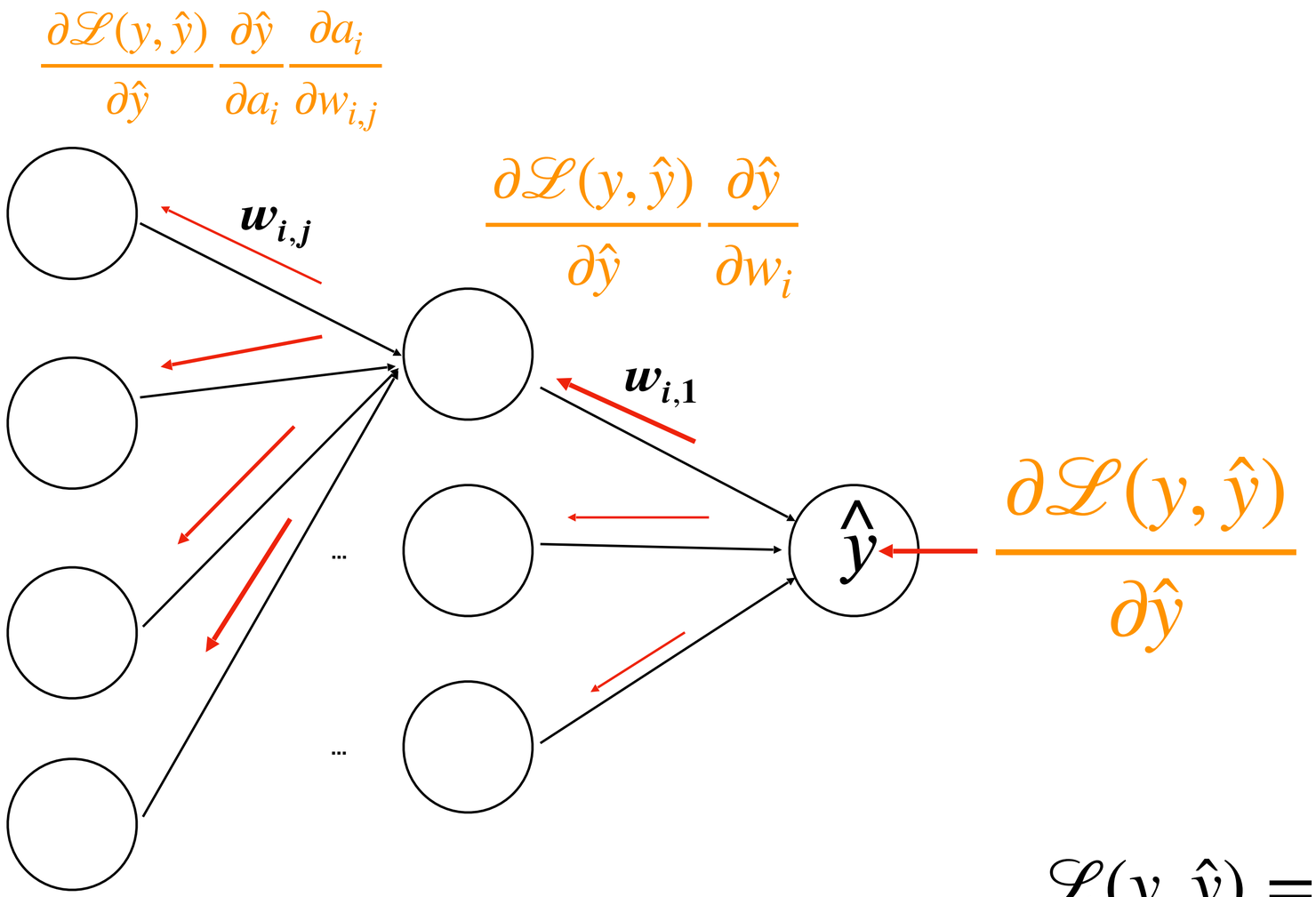
*Animation: 3blue1brown*




**Training: backward propagation**

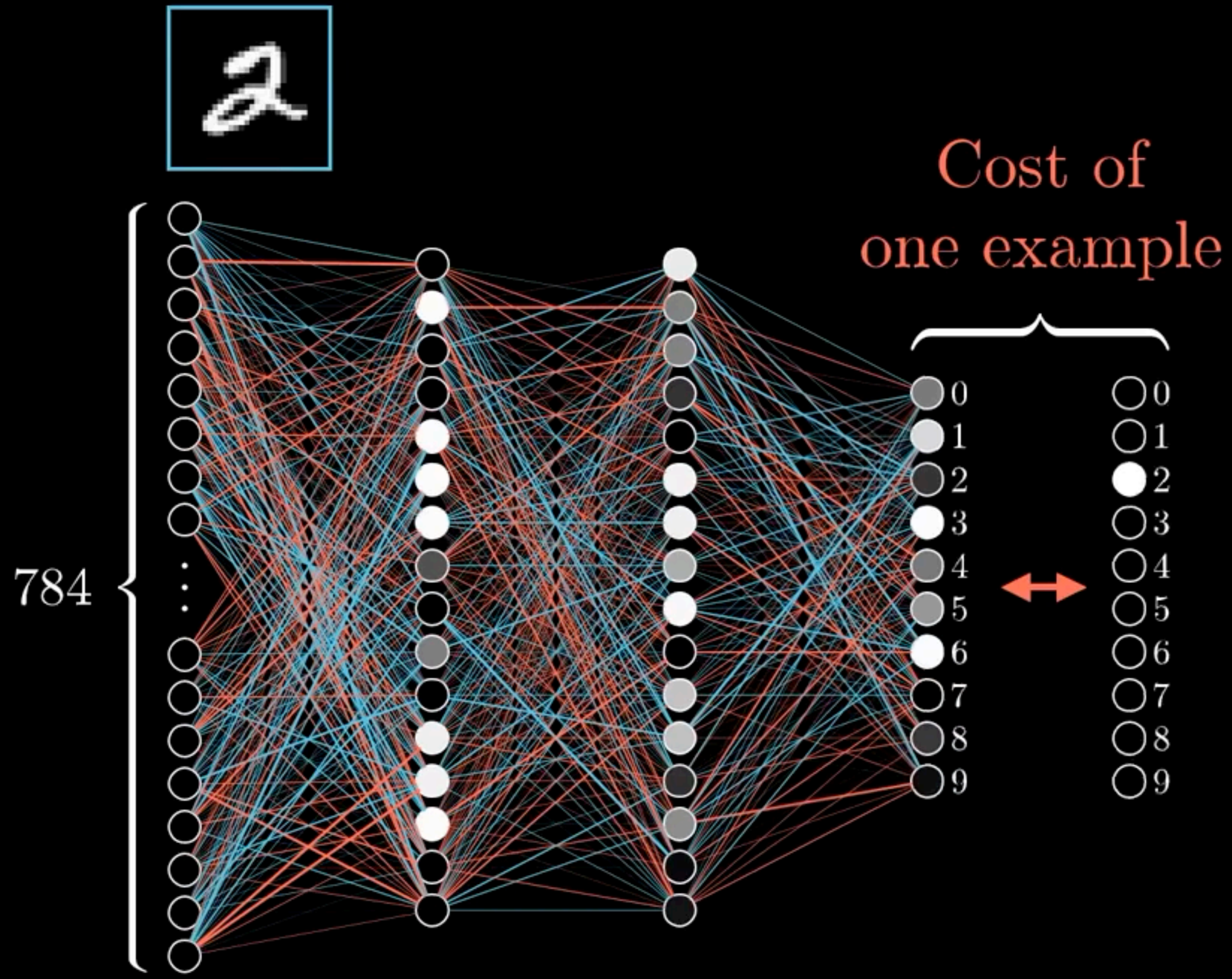
*Animation: 3blue1brown*

# Training: backward propagation



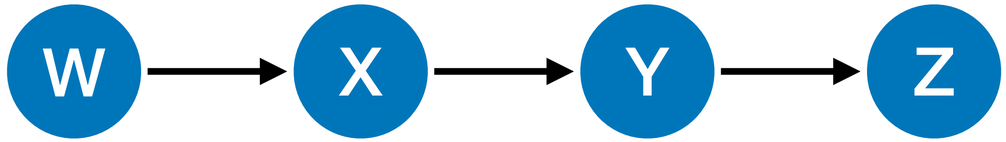

- In the context of DL we need to compute the gradient for each layer.
- We do this by applying the **chain rule** of derivatives.
- This algorithm is known as **backpropagation**.

$$\mathscr{L}(y,\hat{y}) = L(\mathbf{W},b) = \frac{1}{N}\sum_{i=1}^{N}\left(y_i - \hat{y}_i\right)^2$$



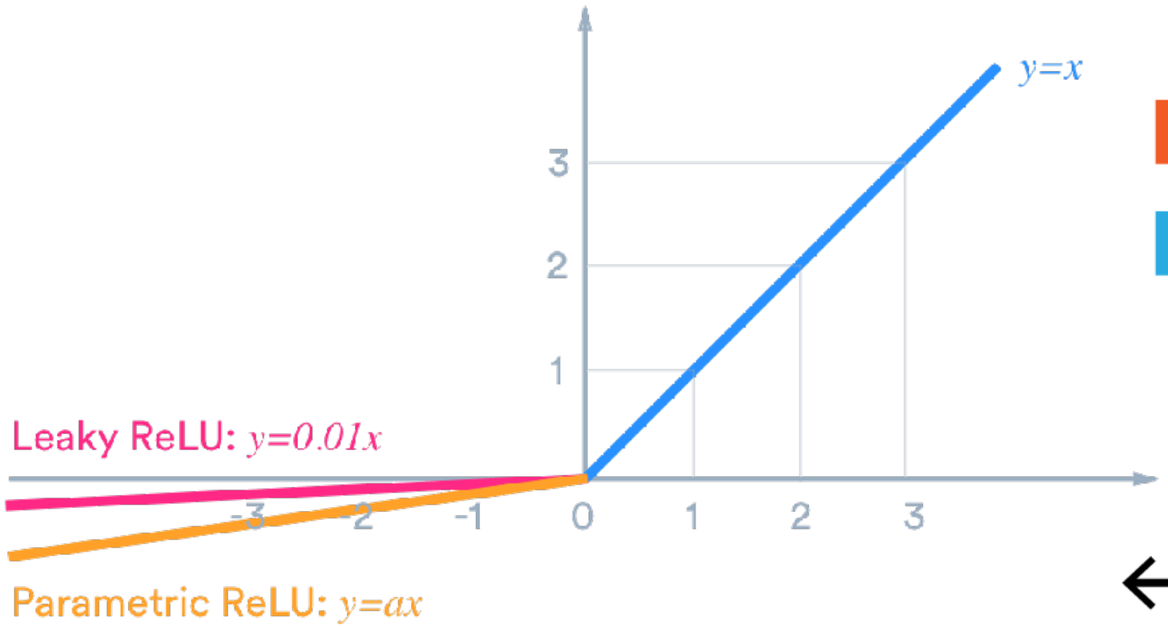

**Training: backward propagation**

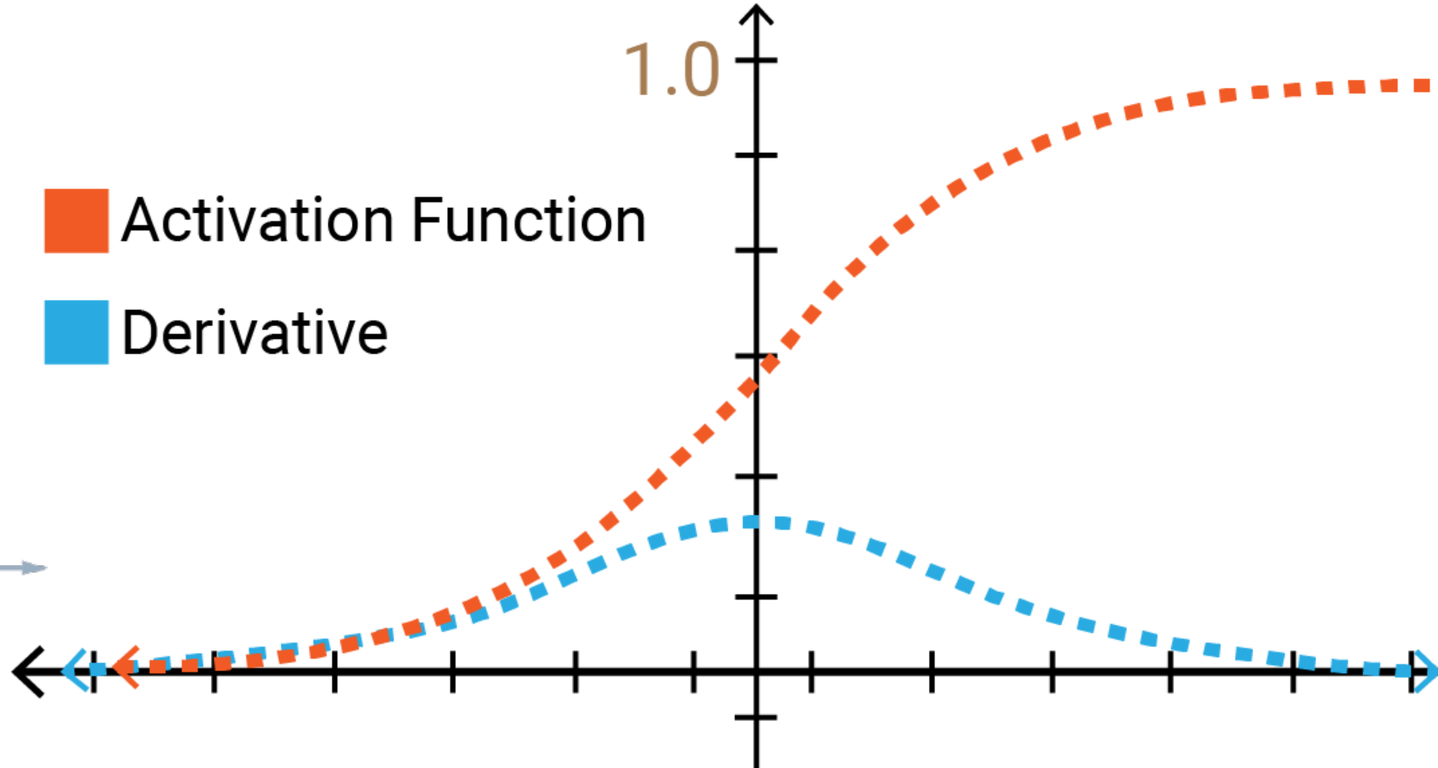
# The vanishing gradient problem




**Chain rule**

$$\frac{\partial z}{\partial w} = \frac{\partial z}{\partial y}\frac{\partial y}{\partial x}\frac{\partial x}{\partial w}$$

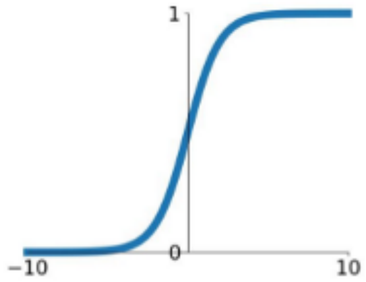


# Activation Functions

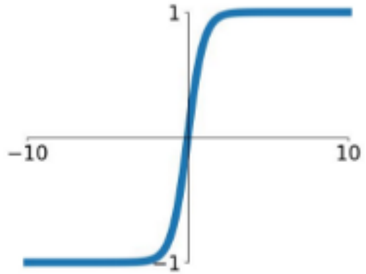
**Sigmoid**

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



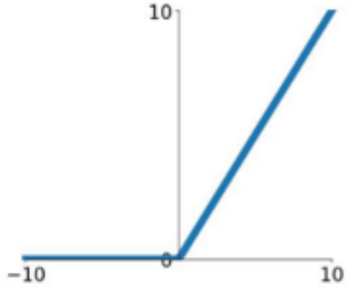

**tanh**

$$\tanh(x)$$



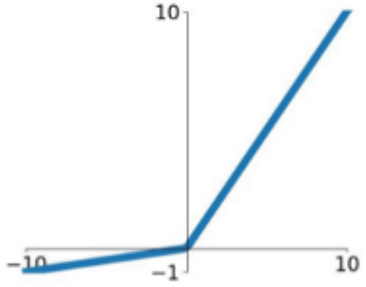

**ReLU**

$$\max(0, x)$$




**Leaky ReLU**

$$\max(0.1x, x)$$

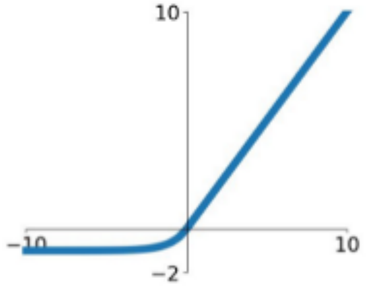



**Maxout**

$$\max(w_1^T x + b_1, w_2^T x + b_2)$$

**ELU**

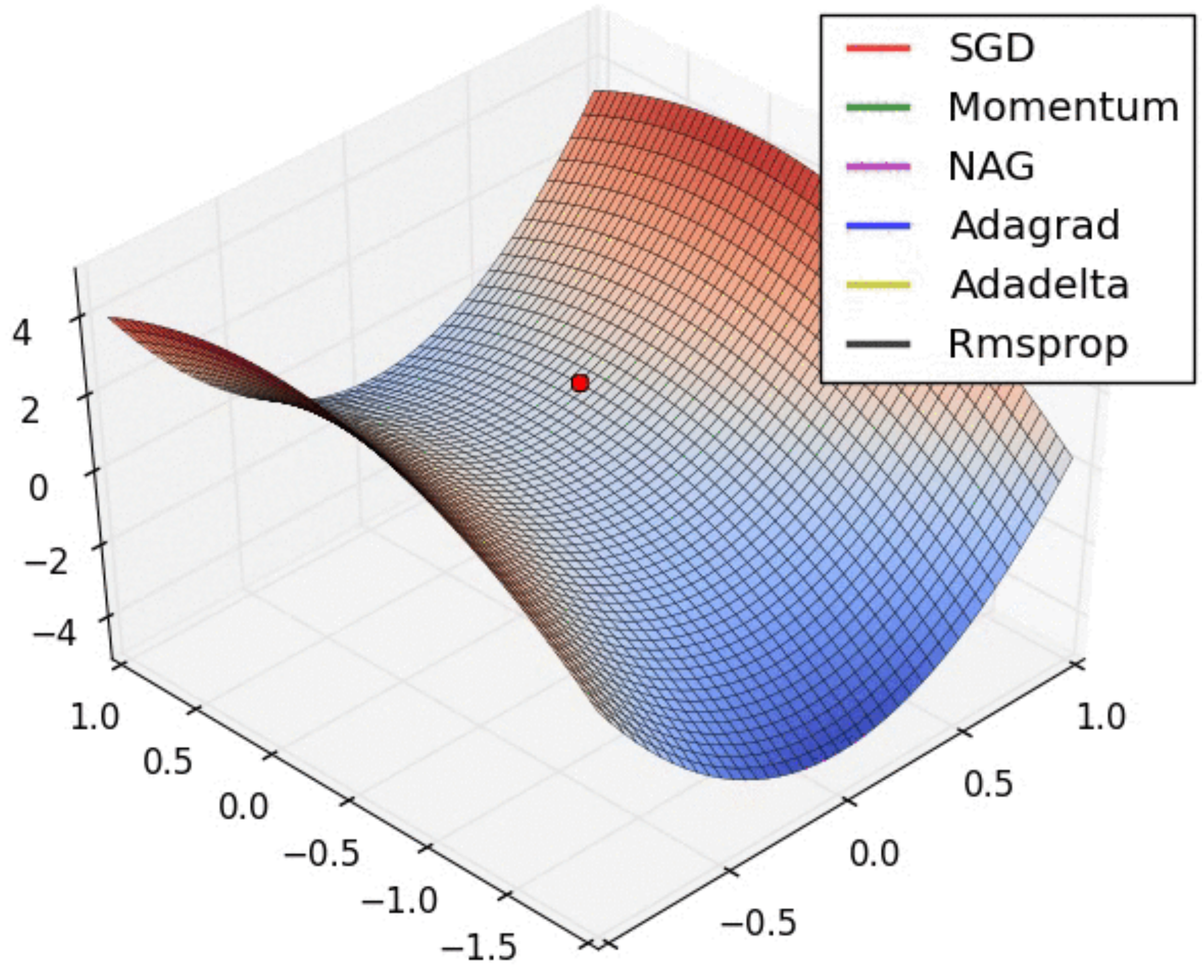
$$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$$

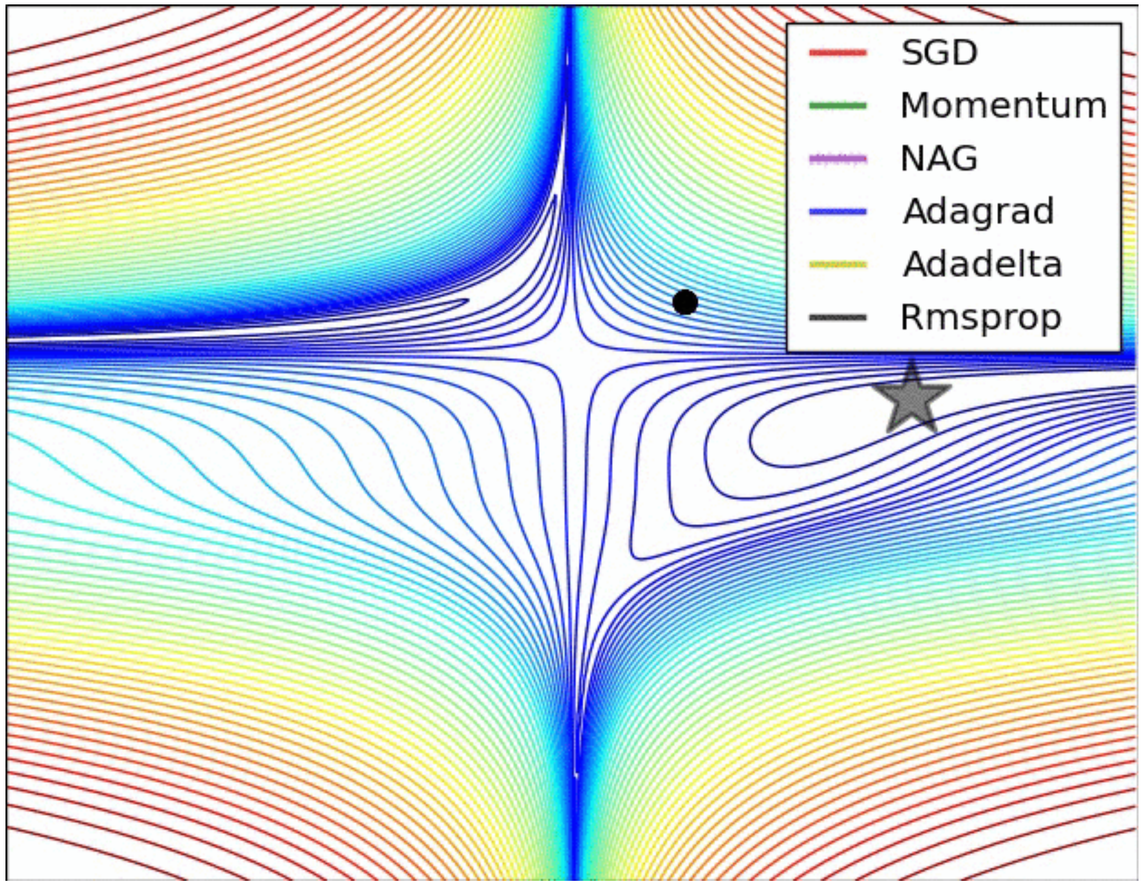



**But one can design their own activation functions!**

# Stochastic gradient descent (SGD)

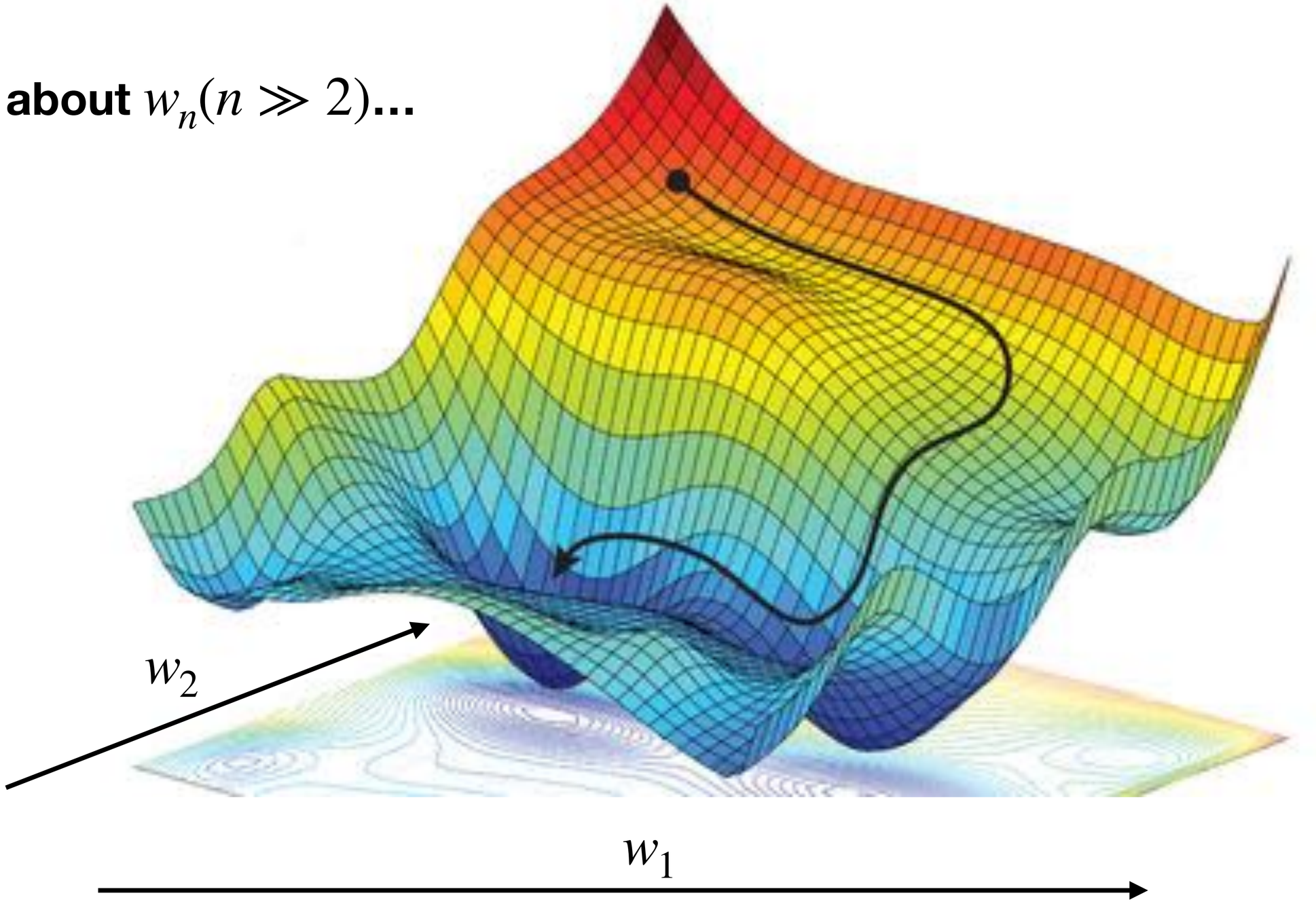
# Optimizers



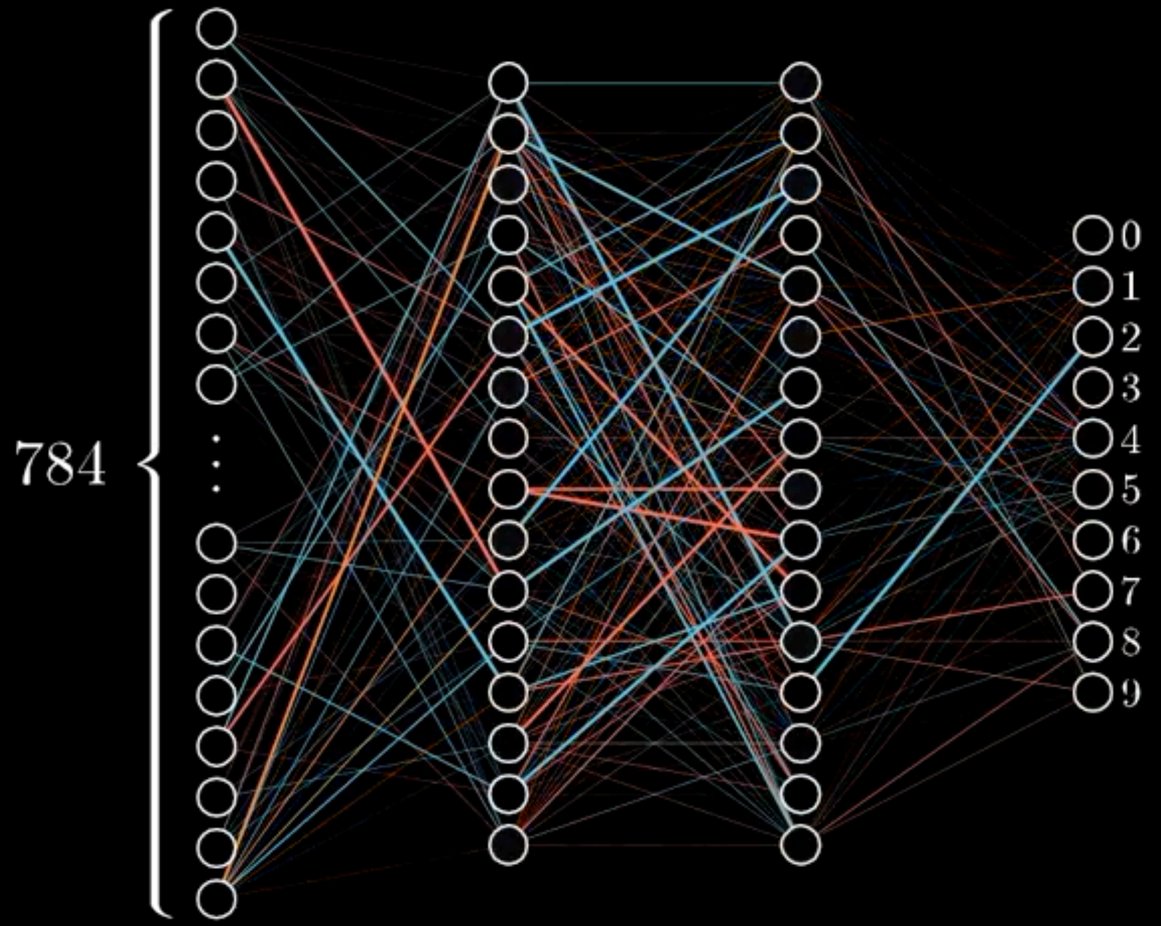




*Source: medium.com/analytics-vidhya*

**Think about $w_n (n \gg 2)$...**



# Training in progress. . .

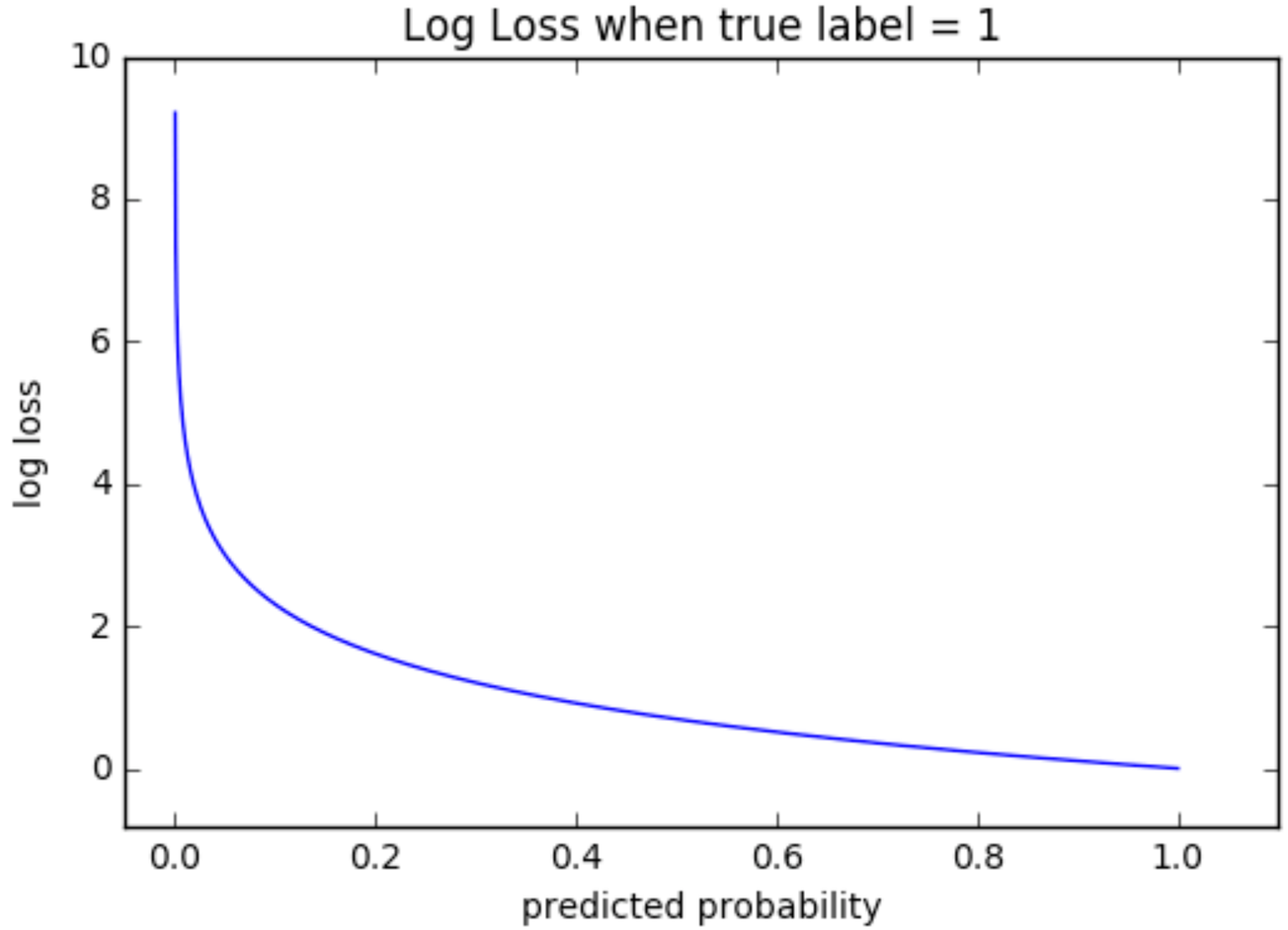


# Some commonly used loss functions

**Regression**
- Mean squared error
- Mean squared logarithmic error
- Mean absolute error

**Binary Classification**
- Binary cross-entropy
- Hinge loss
- Squared hinge loss

**Multi-Class Classification**
- Multi-class cross-entropy
- Sparse multi-class cross-entropy
- Kullback-Leibler divergence



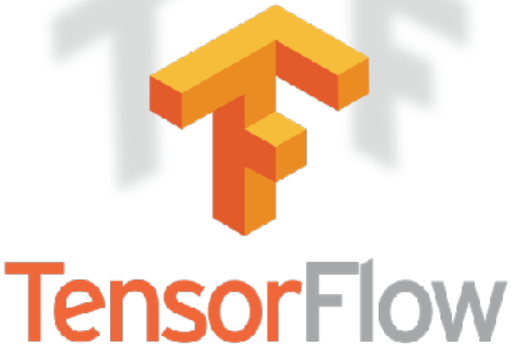

**Cross-entropy loss outputs a log probability**

# DL frameworks

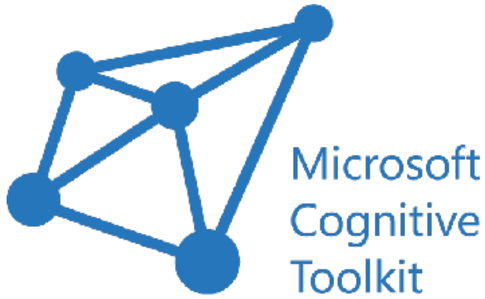
**In DL, you need to**

- Define neurons and layers
- Define loss function
- Calculate losses
- Calculate gradient (multivariate calculus)
- Backward propagation
- Update weights






- Many frameworks exist; **TensorFlow**, **CNTK**, **Torch**, **Keras**, **Theano**, **Caffe**, ...
- We will use **Keras** (https://keras.io/)
- Keras used to call TensorFlow as a *backend*, but is now fully integrated in TensorFlow.




# Open Datasets

## Datasets

Find and use datasets or complete tasks. Learn more.

**Processed, balanced, well-behaved, labeled datasets to benchmark your networks!**

Help the community by creating and solving Tasks on datasets!

Search 29,853 datasets | Feedback | Filter

PUBLIC | Sort by: Hottest

**Hotel booking demand** — 270
Jesse Mostipak
19 days · 1 MB · 10.0 · 1 File (CSV) · 1 Task

**Big Five Personality Test** — 134
Bojan Tunguz
14 days · 159 MB · 9.7 · 3 Files (CSV, other)

**StartUp Investments (Crunchbase)** — 92
Andy_M
14 days · 3 MB · 8.8 · 1 File (CSV)

Open Tasks

Can we predict the possibility of a bo...
0 Submissions · In Hotel booking demand

Visualize US Accidents Dataset
12 Submissions · In US Accidents (3.0 million...

What to watch on Netflix ?
4 Submissions · In Netflix Movies and TV Sh...

https://www.tensorflow.org/datasets
https://www.kaggle.com/datasets
http://topepo.github.io/caret/data-sets.html
https://github.com/awesomedata/awesome-public-datasets

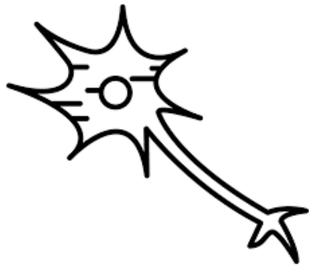
# Take home messages

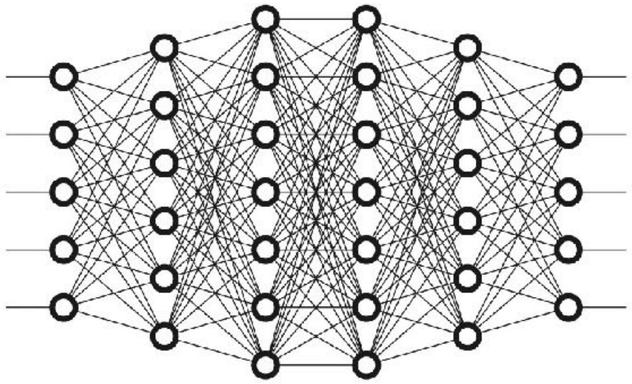
**Machine learning**
… can be as simple as curve fitting
… is a new paradigm of programming, driven by data
… is an optimization process

**Deep learning**
… is a subfield of machine learning
… relies on deep neural networks
… learns to encode the input data using many layers of concept hierarchies

# Take-home messages

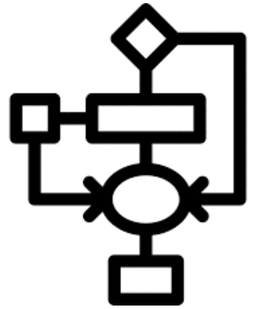
**In a neuron:**
… the main job is to calculate a weighted average
… the decision is made through the activation function





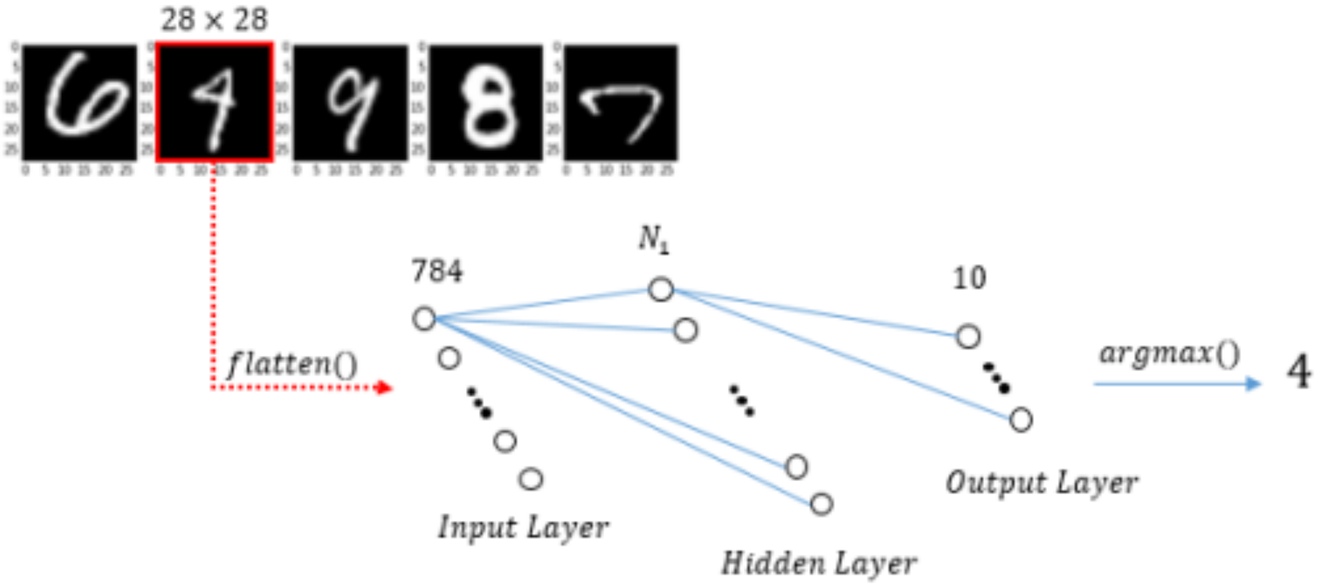
**In a neural network:**
… losses are calculated using the loss function
… losses are calculated by comparing the labels and the prediction
… predictions are made through forward propagation
… weights are updated through the backward propagation process
… optimizers are used to decide the weights updating strategies

**In a deep learning workflow:**
… the heavy lifting is mostly done by DL frameworks
… open datasets are crucial for benchmarking and bootstrapping DNNs




jupyter

https://jupyter.lisa.surfsara.nl/jhlsrf008




**MNIST dataset**

(**M**odified **N**ational **I**nstitute for **S**tandards and **T**echnology)