Data Science: Principles and Practice

Lecture 4: Deep Learning, Part I

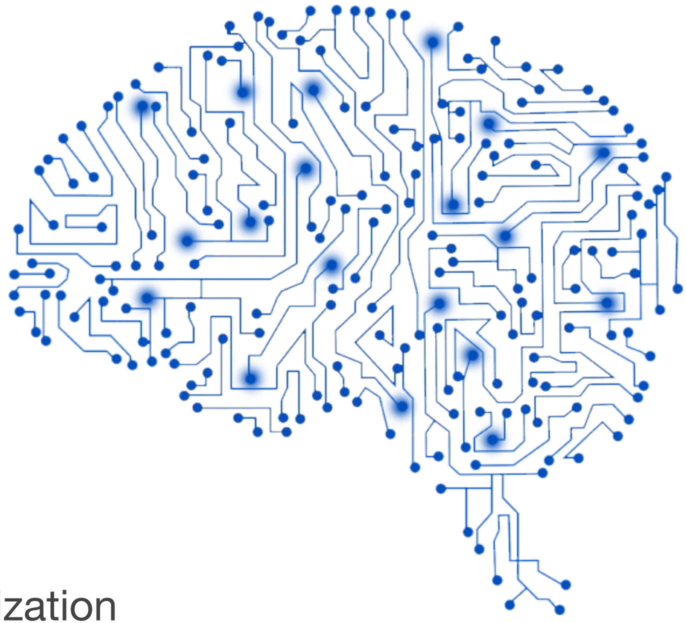
Marek Rei

UNIVERSITY OF CAMBRIDGE

What is Deep Learning?

Deep learning is a class of machine learning algorithms.

Neural network models with multiple hidden layers.

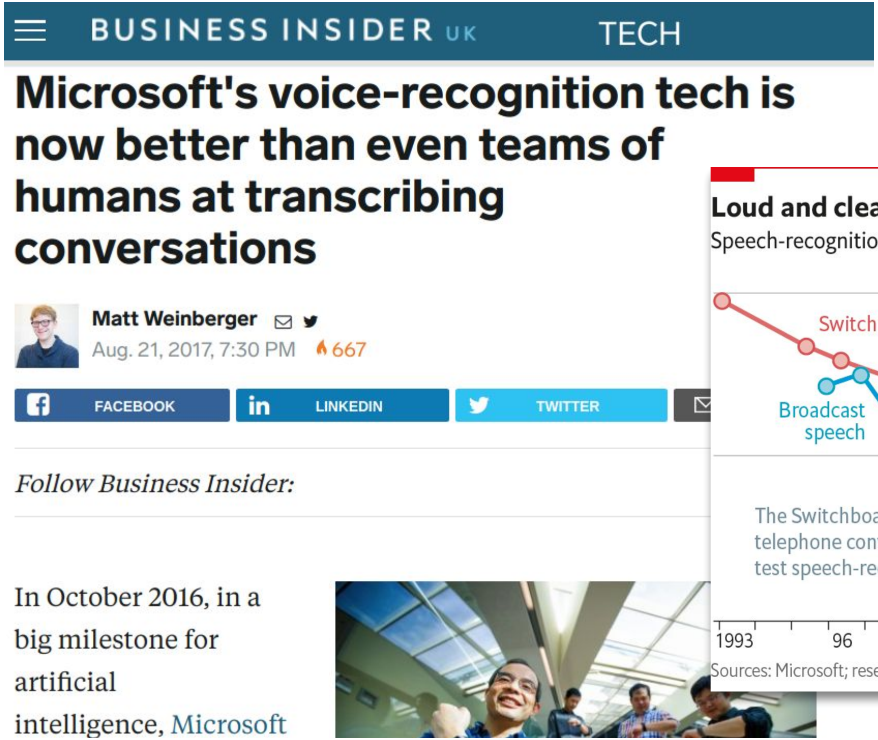


Today: The basics of neural network models, optimization

Next lecture: Implementing models with Tensorflow, network components, practical tips

https://trends.google.com

The Rise of Deep Learning



BUSINESS INSIDER UK TECH

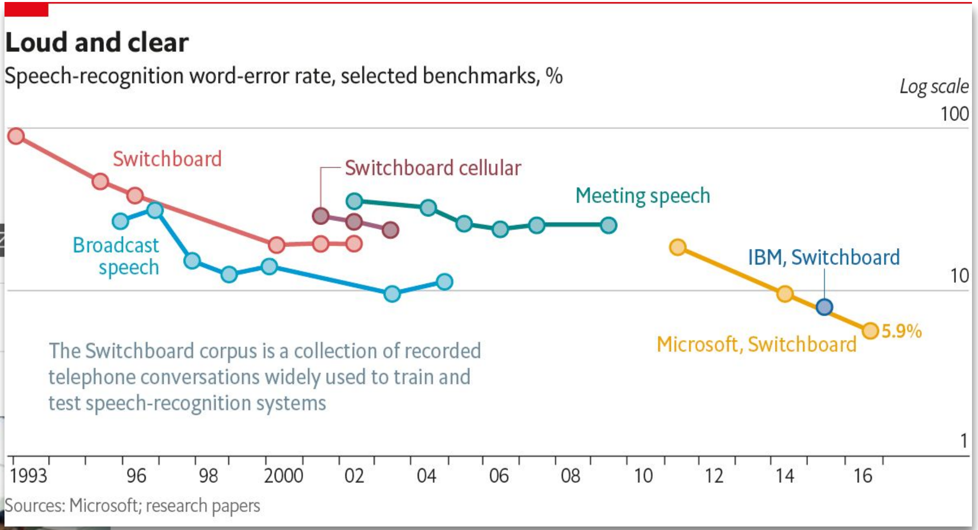
Microsoft's voice-recognition tech is now better than even teams of humans at transcribing conversations

Matt Weinberger
Aug. 21, 2017, 7:30 PM 667

FACEBOOK LINKEDIN TWITTER

Follow Business Insider:

In October 2016, in a big milestone for artificial intelligence, Microsoft



http://uk.businessinsider.com/microsoft-research-beats-humans-at-speech-transcription-2017-8
https://www.economist.com/technology-quarterly/2017-05-01/language

The Rise of Deep Learning

AI

Google taps neural nets for better offline translation in 59 languages

KHARI JOHNSON @KHARIJOHNSON JUNE 12, 2018 10:16 AM

Above: Google Translate for iOS.
Image Credit: Jordan Novet / VentureBeat

Google's online translations have been p
the company is rolling out its neural net-
for Google Translate iOS and Android ap

Offline NMT was made by the Translate
Google product manager Julie Cattiau to
95 percent of Google Translate's user ba
Indonesia, Cattiau said.

https://venturebeat.com/2018/06/12/google-taps-neural-nets-for-better-offline-translation-in-59-languages/

The Rise of Deep Learning

ars TECHNICA SUBSCRIPTIONS SIGN IN

TECH —

Google's AlphaGo AI beats world's best human Go player

STR/AFP/Getty Images

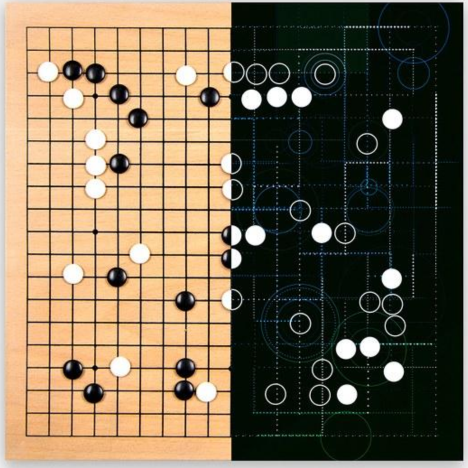
Ke Jie tried to use AlphaGo's own moves and lost.

SEBASTIAN ANTHONY - 5/23/2017, 2:20 PM

Enlarge / China's 19-year-old Go player Ke Jie (L) prepares to make a move during the first match against Google's artific
AlphaGo in Wuzhen, east China's Zhejiang province on May 23, 2017.

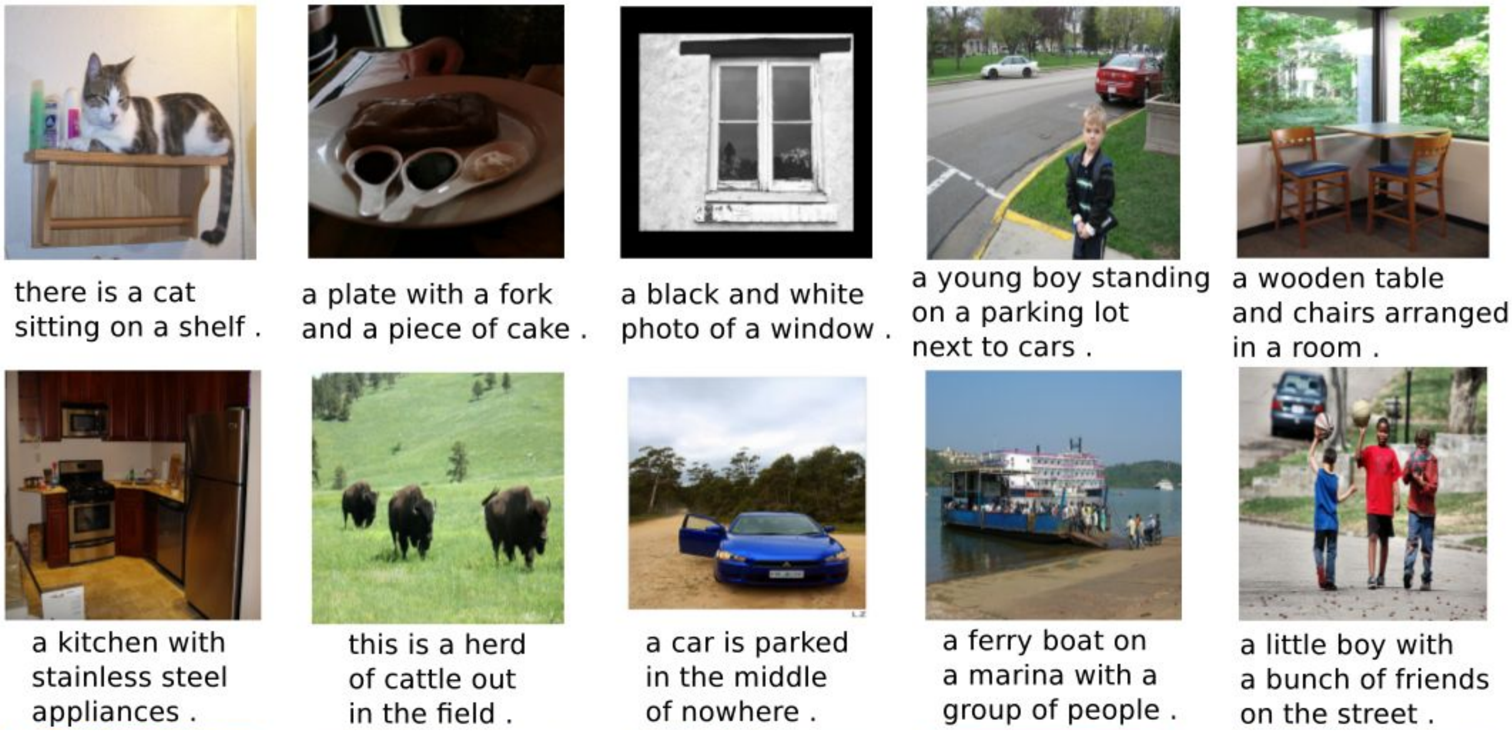
DeepMind's AlphaGo AI has defeated Ke Jie in the first round of a best-of-three Go match in China. A video
embedded below. Ke Jie was defeated by just a half a point—the closest margin possible—but scoring versu
disingenuous: DeepMind's AI doesn't try to win by a large margin; it just plots the surest route to victory, ev
point.

Ke Jie is generally considered to be the world's best human Go player, but he wasn't expected to win; Alpha
Chinese 19-year-old earlier in the year during an unbeaten online 60-match victory streak.



https://arstechnica.com/gadgets/2017/05/googles-alphago-ai-beats-worlds-best-human-go-player/

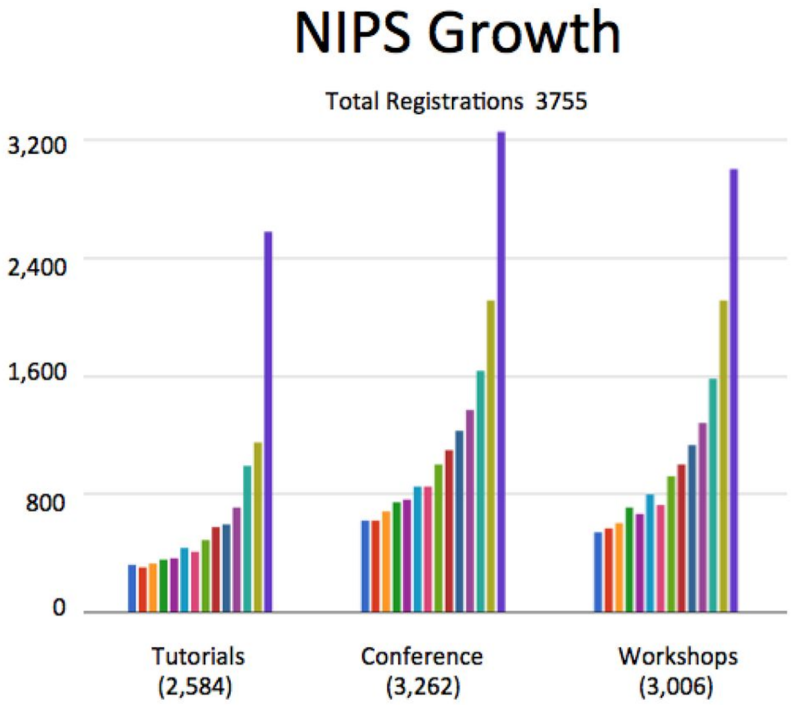
The Rise of Deep Learning

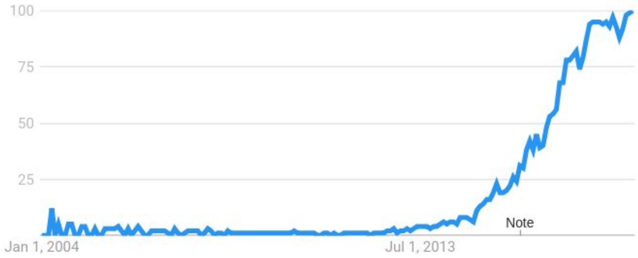


Kiros et al. (2014) Unifying Visual-Semantic Embeddings with Multimodal Neural Language Models

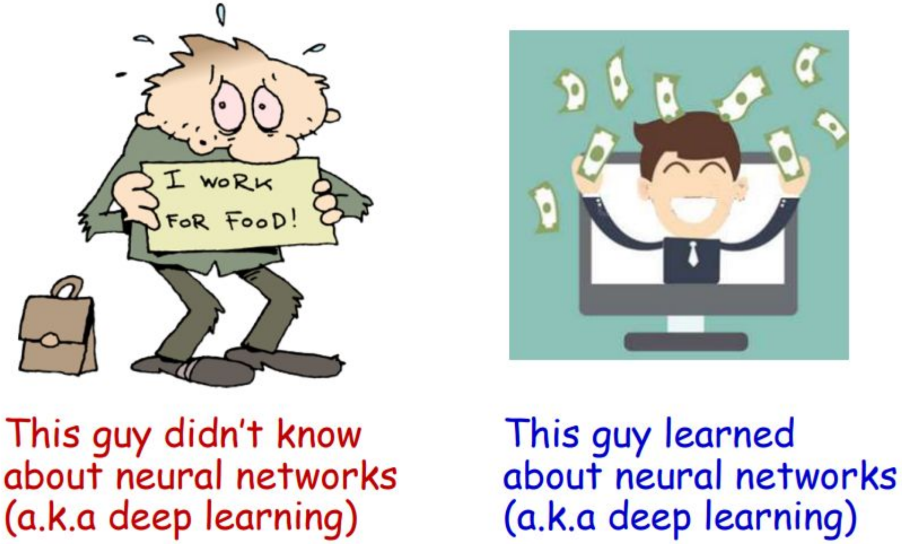
The Rise of Deep Learning

Conference on Neural Information Processing Systems (NIPS) - one of the main conferences on deep learning and machine learning.





The Hype Train of Deep Learning



http://deeplearning.cs.cmu.edu

- “Deep learning” is often used as a buzzword, even without understanding it.
- Be mindful - it’s a powerful class of machine learning algorithms, but not a magic solution to every problem.

But Why Now?

- 2012 - AlexNet wins ImageNet, Krizhevsky
- 2006 - Restricted Boltzmann Machine, Hinton
- 1998 - ConvNets for OCR, LeCun
- 1997 - LSTM, Hochreiter & Schmidhuber
- 1974 - backpropagation, Werbos
- 1958 - perceptrons, Rosenblatt

The theory was there before, but the conditions are now better for putting it into action.

1. Big Data

- Large datasets for training
- Better methods for storing and managing data

WIKIPEDIA
The Free Encyclopedia

2. Faster Hardware

- Graphics Processing Units (GPUs)
- Faster CPUs
- More affordable

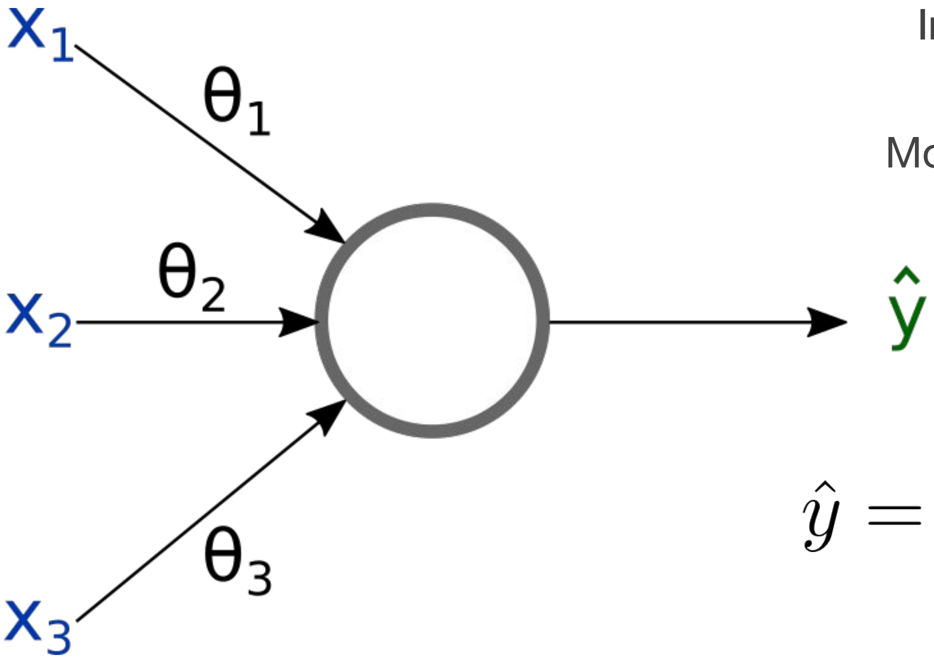
3. Better Software

- Better Optimization Algorithms
- Automatic Differentiation Libraries

TensorFlow

Fundamentals of Neural Networks

Remember Linear Regression



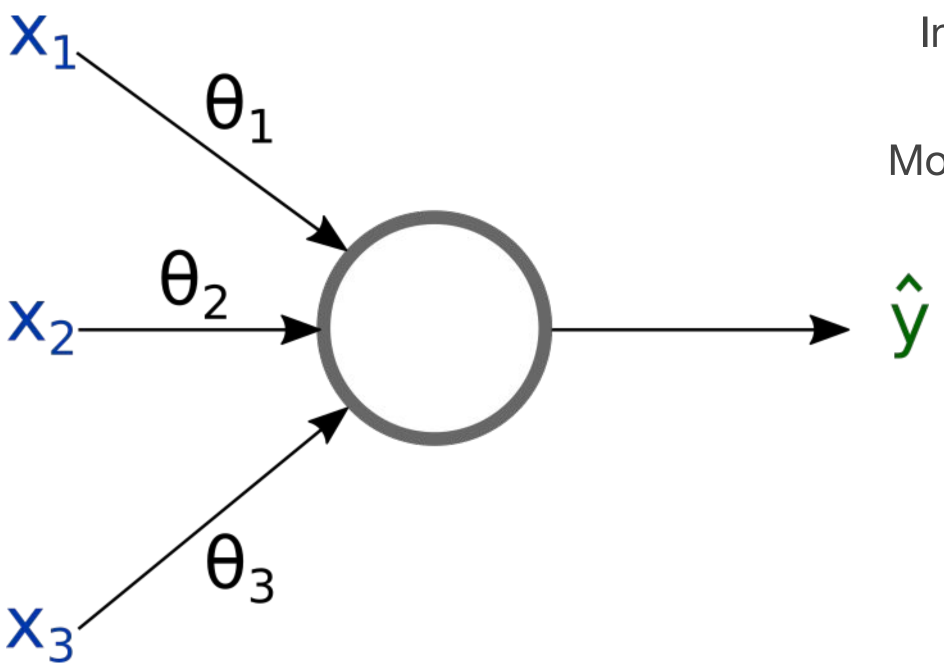
Input features: $[x_1, x_2, x_3]$

Model parameters: $[\theta_1, \theta_2, \theta_3]$

$$\hat{y} = \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 + b$$

Often implicit

Remember Linear Regression



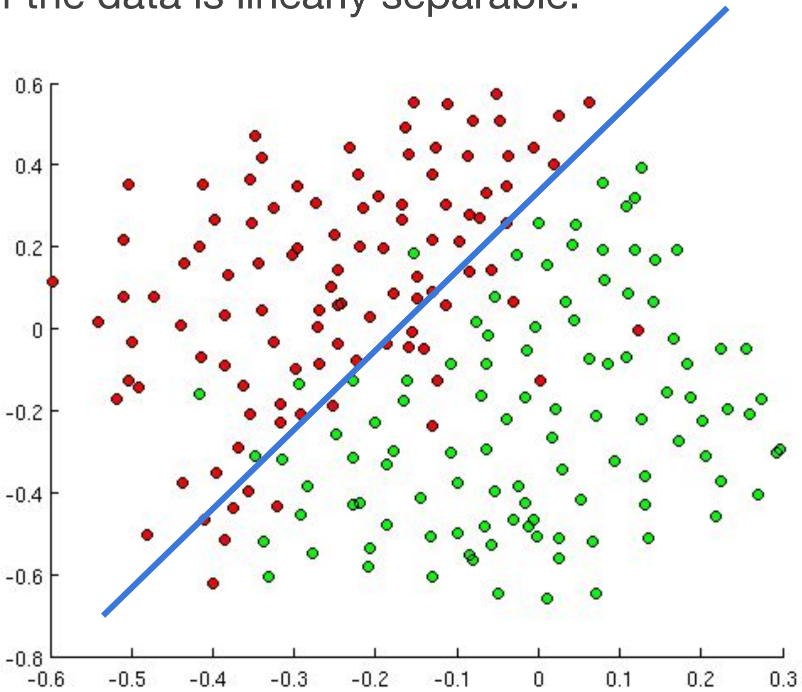
Input features: $[x_1, x_2, x_3]$

Model parameters: $[\theta_1, \theta_2, \theta_3]$

$$\hat{y} = \sum_k \theta_k x_k$$

Linear Separability of Classes

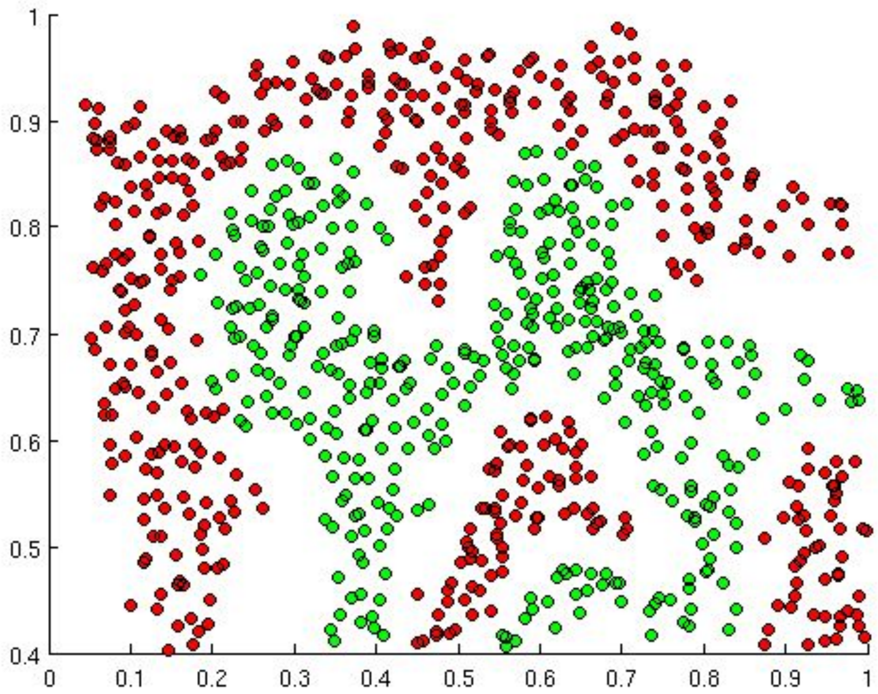
Linear models are great if the data is linearly separable.



http://introtodeeplearning.com/

Linear Separability of Classes

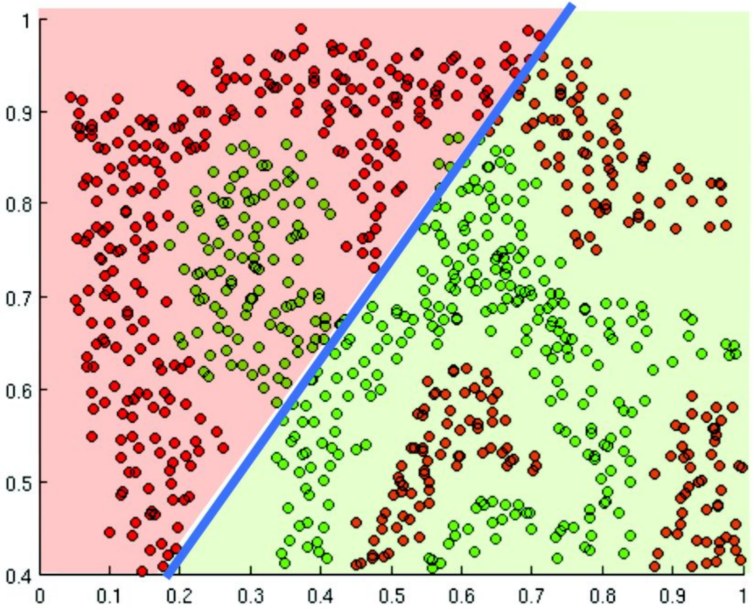
… but often that is not the case.

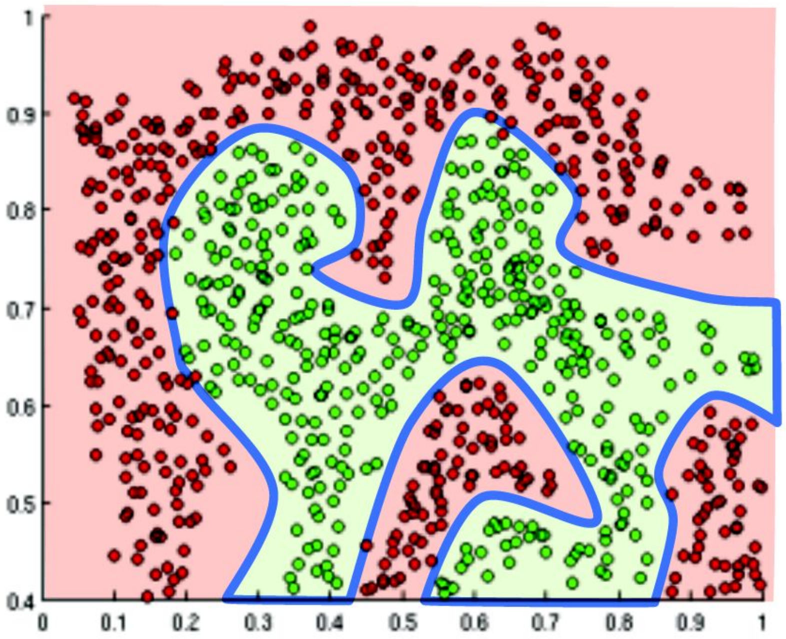


http://introtodeeplearning.com/

Linear Separability of Classes

Linear models are not able to capture complex patterns in the data.

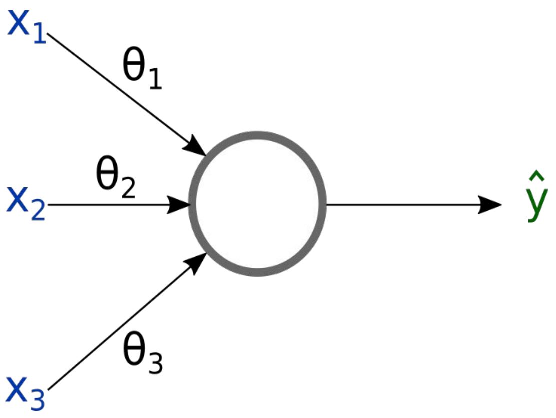




http://introtodeeplearning.com/

Non-linear Activation Functions

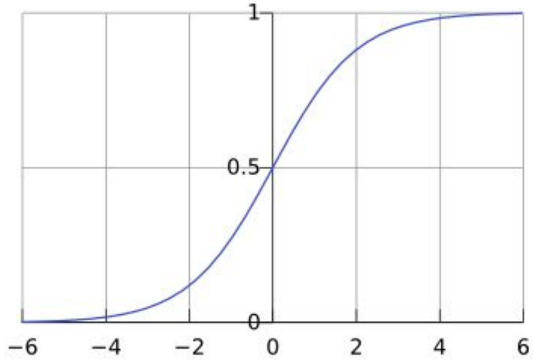
$$\hat{y} = g(\sum_k \theta_k x_k)$$



The logistic function, aka the sigmoid function.

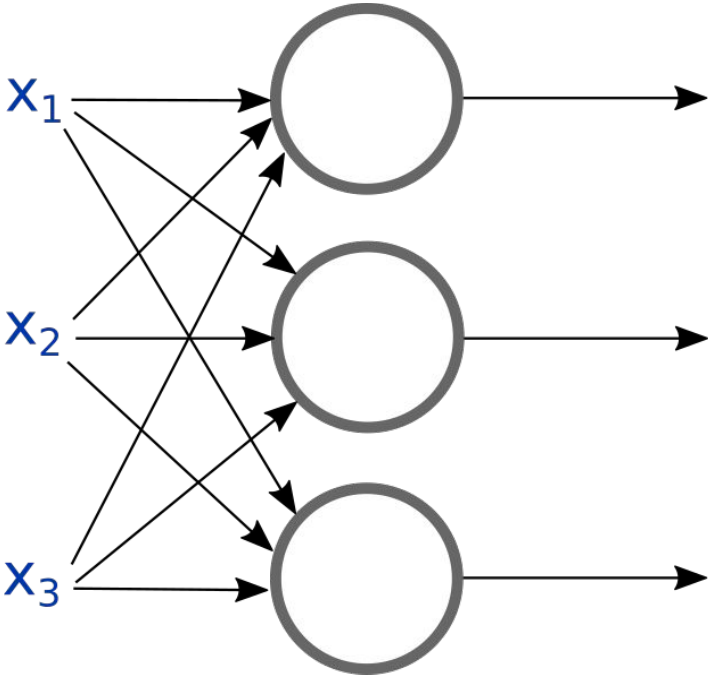
$$g(z) = \frac{1}{1 + e^{-z}}$$

$$z \in [-\infty, \infty] \qquad \hat{y} \in [0, 1]$$



Connecting the Neurons

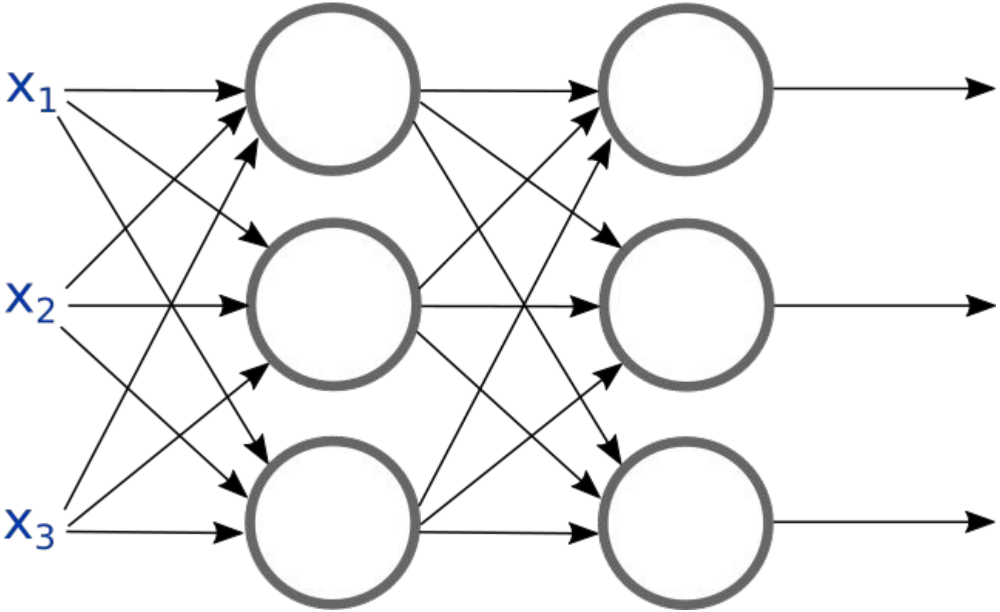
We can connect multiple neurons in parallel - each one will learn to detect something different.



Multilayer Perceptron

Not actually a perceptron

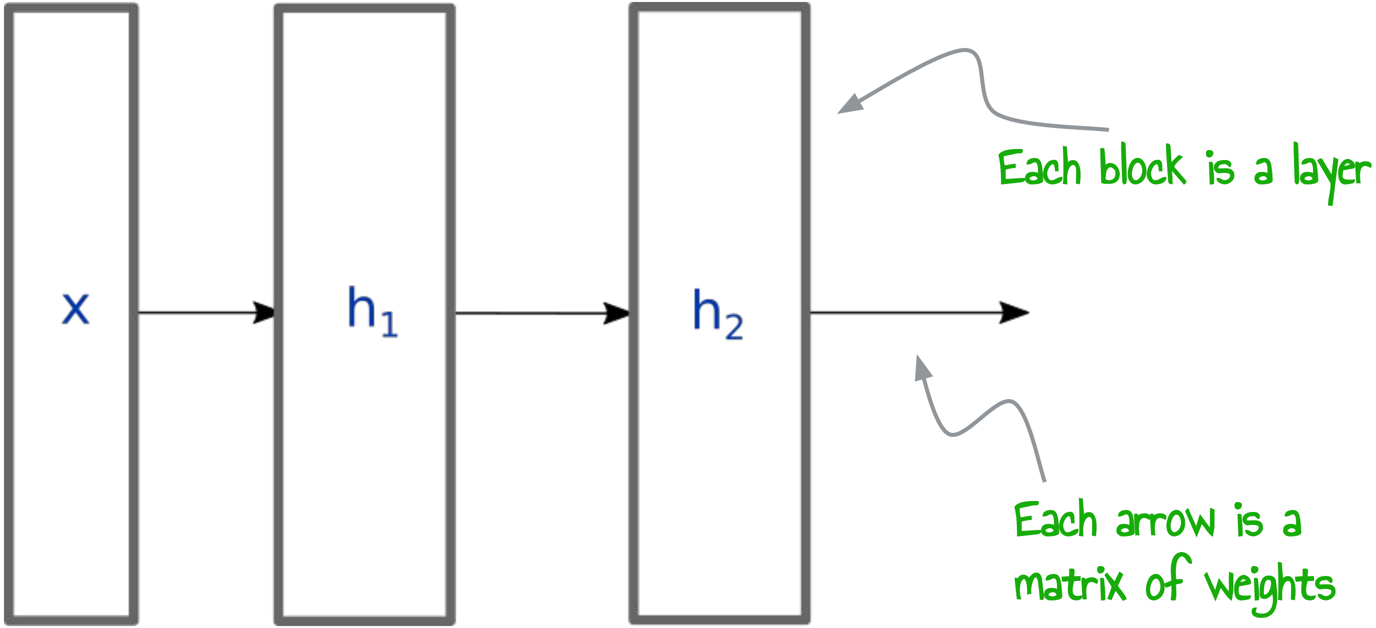
We can connect neurons in sequence in order to learn from higher-order features.



An MLP with sufficient number of neurons can theoretically model an arbitrary function over an input.

Multilayer Perceptron

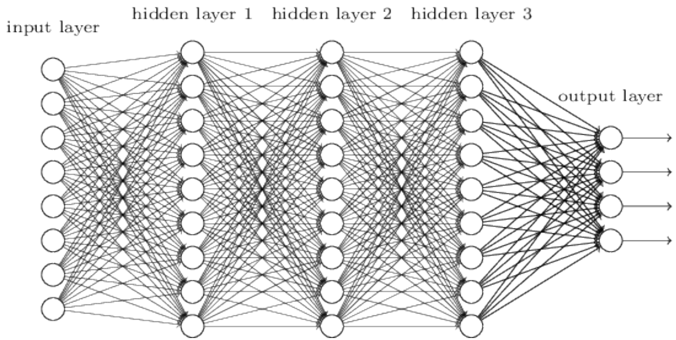
We can connect neurons in sequence in order to learn from higher-order features.

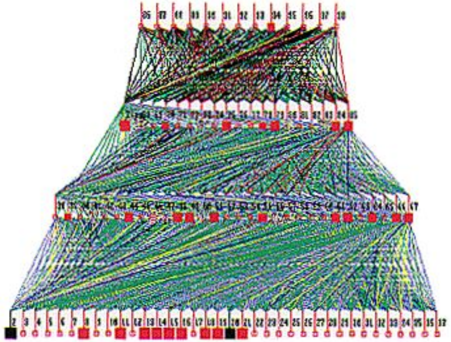


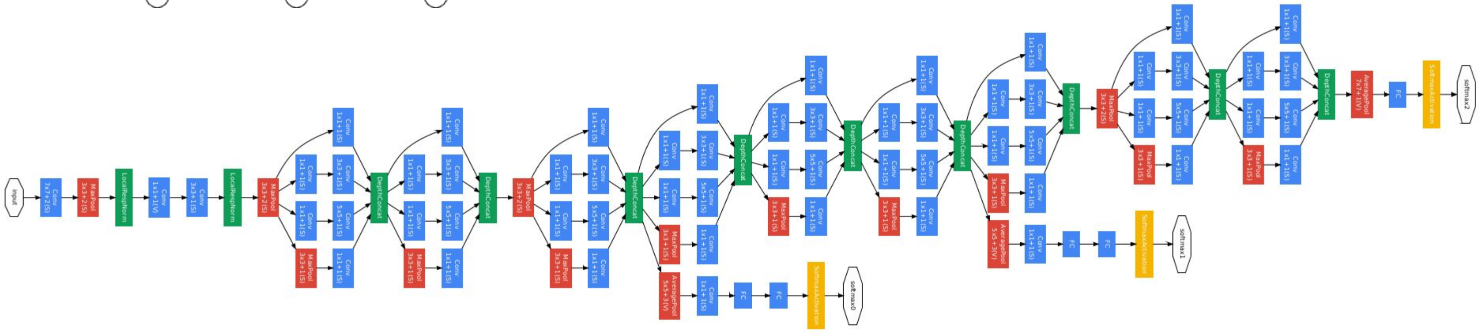
An MLP with sufficient number of neurons can theoretically model an arbitrary function over an input.

Neep Neural Networks

In practice we train neural neural networks with thousands of neurons and millions (or billions) of trainable weights.

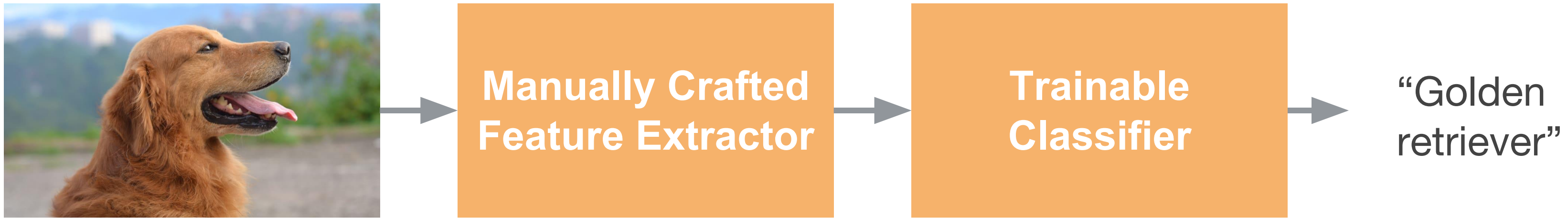




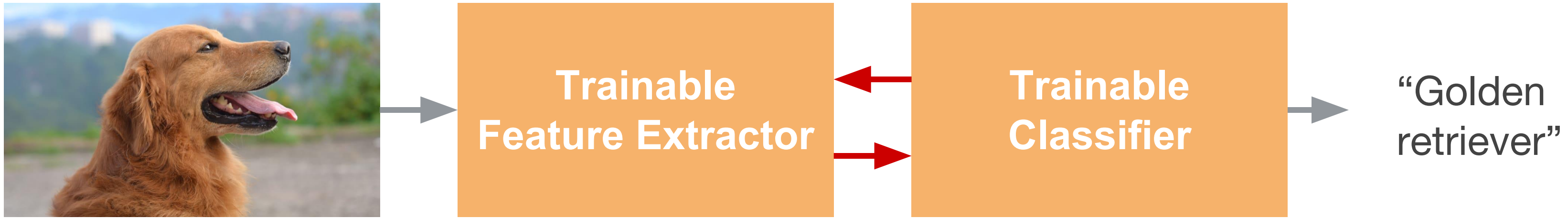


Learning Representations & Features

Traditional pattern recognition

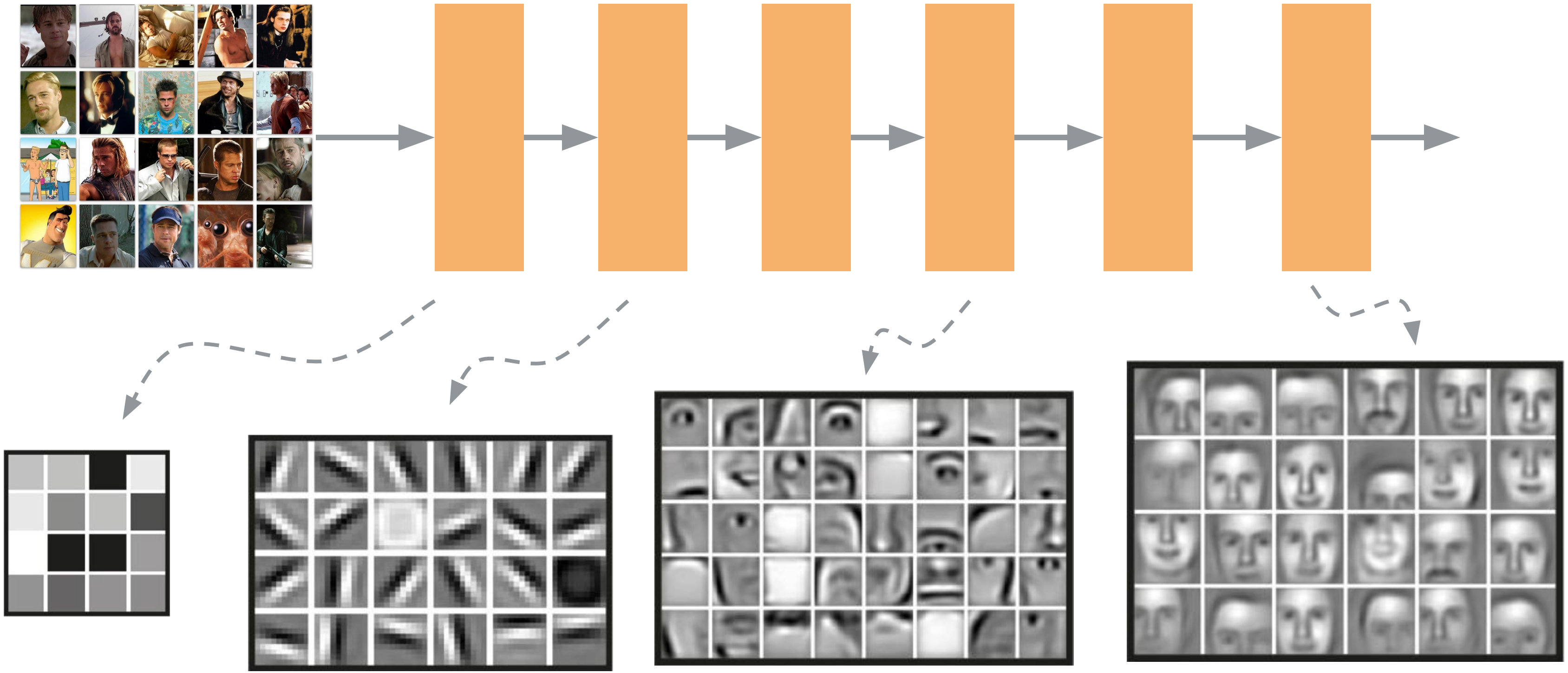


End-to-end training: Learn useful features also from the data



Learning Representations & Features

Automatically learning increasingly more complex feature detectors from the data.



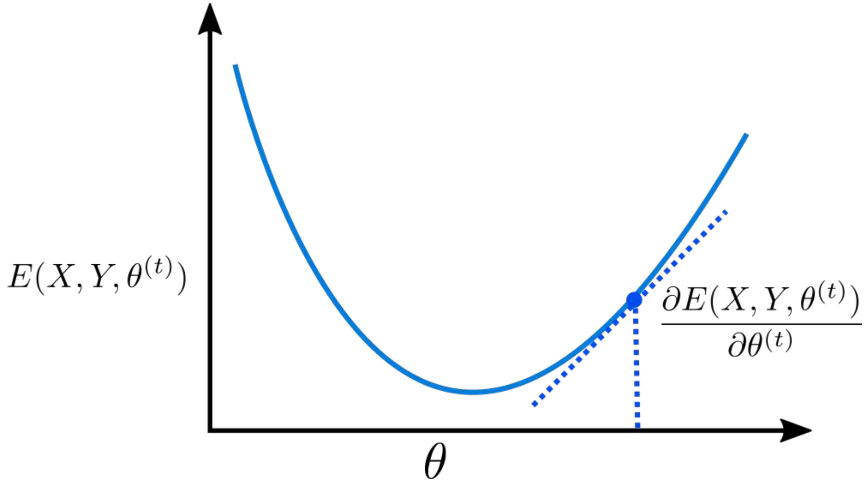
Neural Network Optimization

Optimizing Neural Networks

Define a **loss function** that we want to minimize

Update the parameters using **gradient descent**, taking small steps in the direction of the gradient (going downhill on the slope).

All the operations in the network need to be **differentiable**.

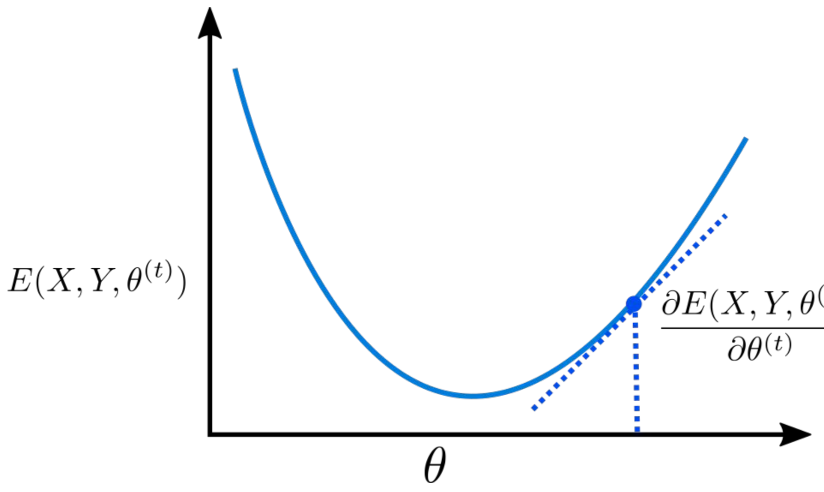


$$\theta_i^{(t+1)} = \theta_i^{(t)} - \alpha \frac{\partial E}{\partial \theta_i^{(t)}}$$

Gradient Descent

Algorithm

1. Initialize weights randomly
2. Loop until convergence:
3. Compute gradient based on the whole dataset
4. Update weights
5. Return weights

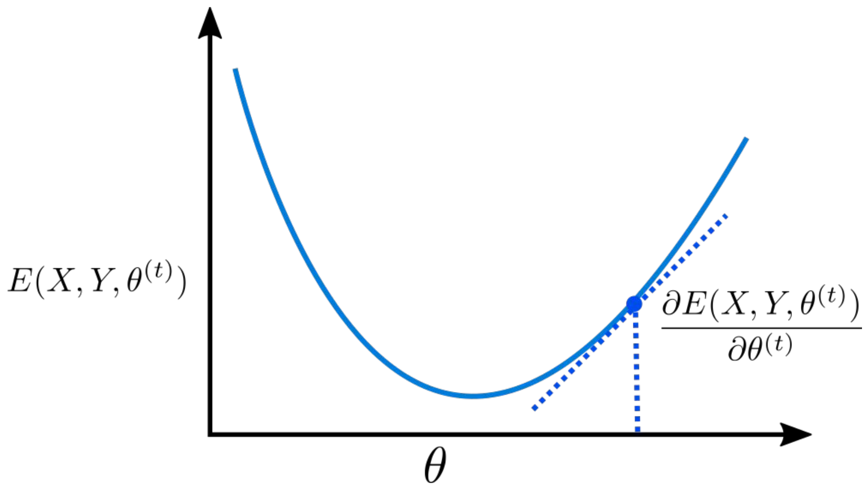


In practice, datasets are often too big for this

Stochastic Gradient Descent

Algorithm

1. Initialize weights randomly
2. Loop until convergence:
3. Loop over **each datapoint**:
4. Compute gradient based on the datapoint
5. Update weights
6. Return weights

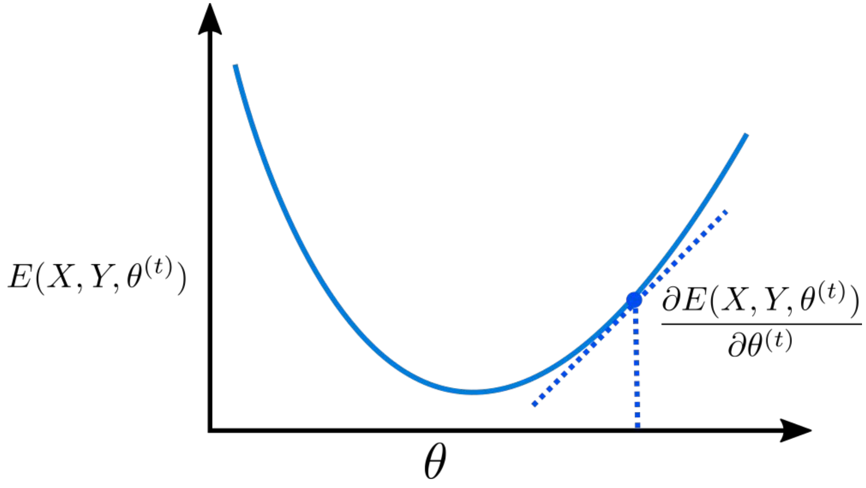


Very noisy to take steps based only on a single datapoint

Mini-batched Gradient Descent

Algorithm

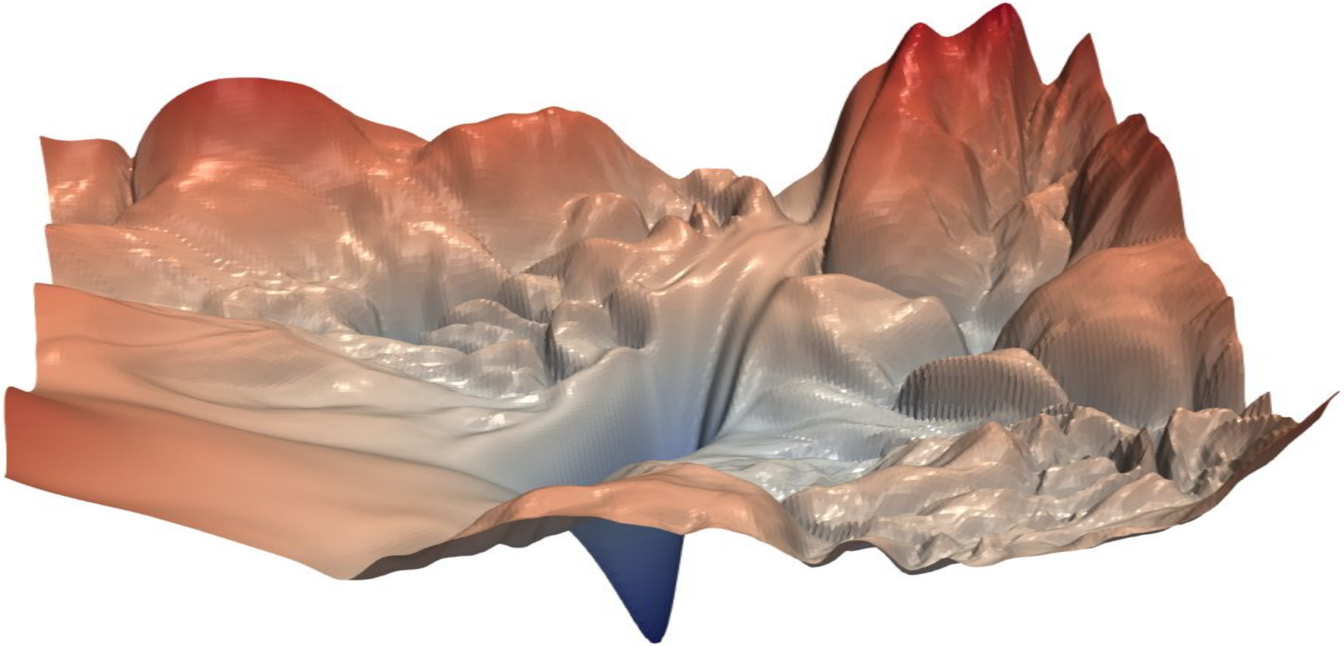
1. Initialize weights randomly
2. Loop until convergence:
3. Loop over **batches of datapoints**:
4. Compute gradient based on the batch
5. Update weights
6. Return weights



This is what we mostly use in practice

Optimizing Neural Networks

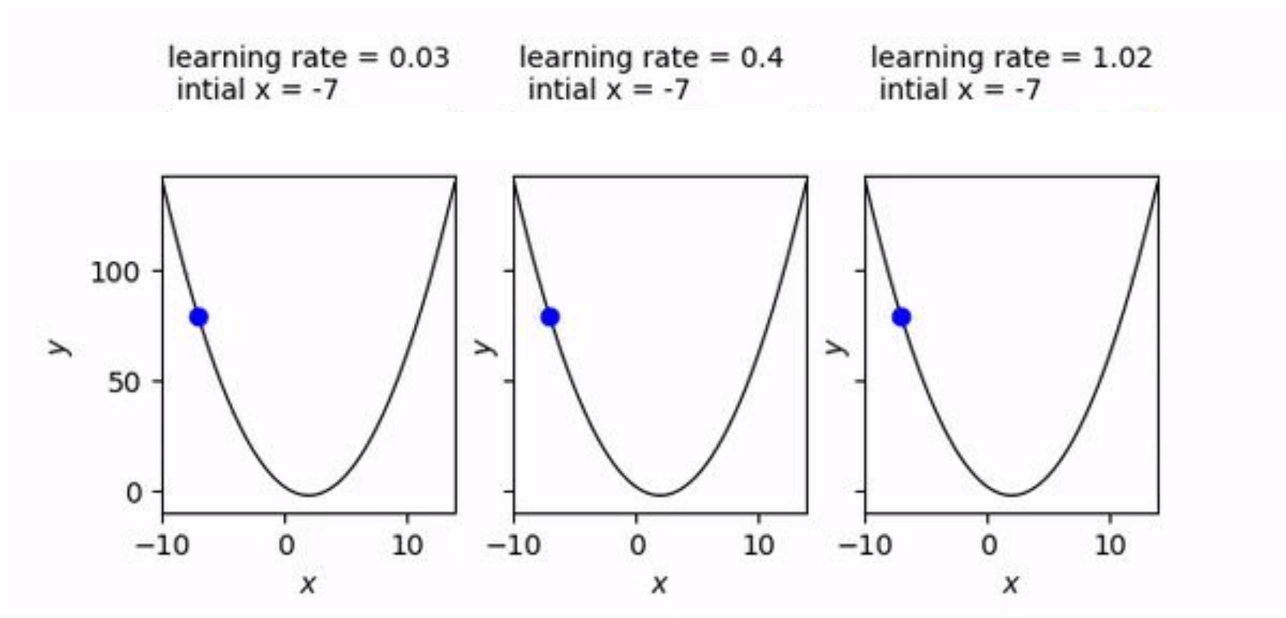
Neural networks have very complex loss surfaces and finding the optimum is difficult.



Li et al., 2018. "Visualizing the Loss Landscape of Neural Nets"

The Importance of the Learning Rate

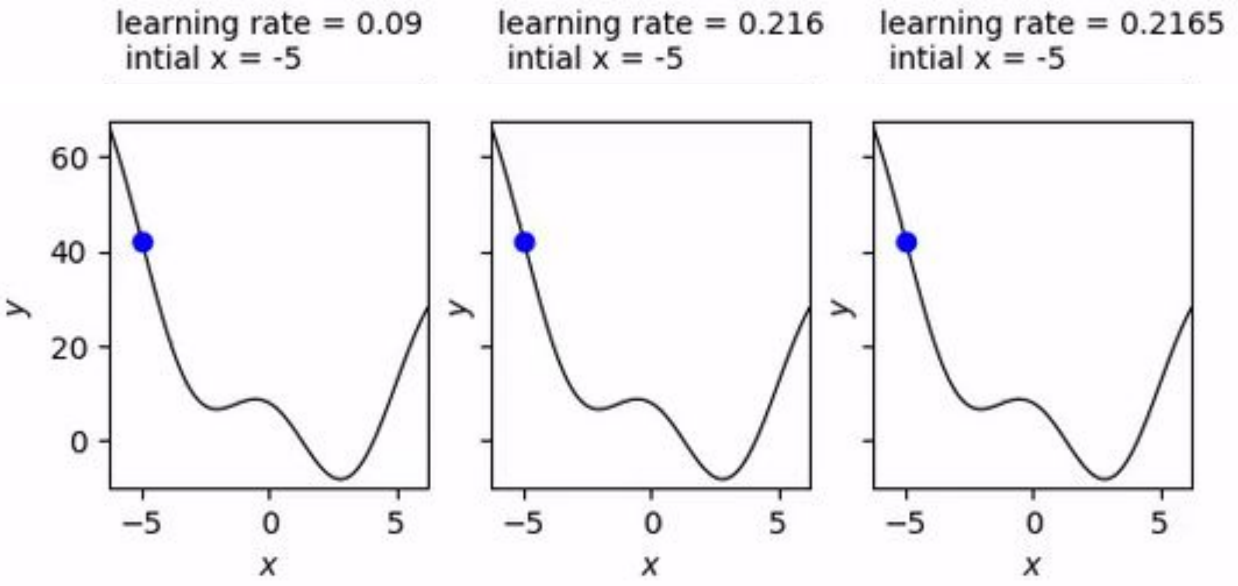
If the learning rate is too low, the model will take forever to converge.
If the learning rate is too high, we will just keep stepping over the optimum values.



https://jed-ai.github.io/opt2_gradient_descent_1/

The Importance of the Learning Rate

A small learning rate can get the model stuck in local minima.
A bigger learning rate can help the model converge better (if it doesn't overshoot).



https://jed-ai.github.io/opt2_gradient_descent_1/

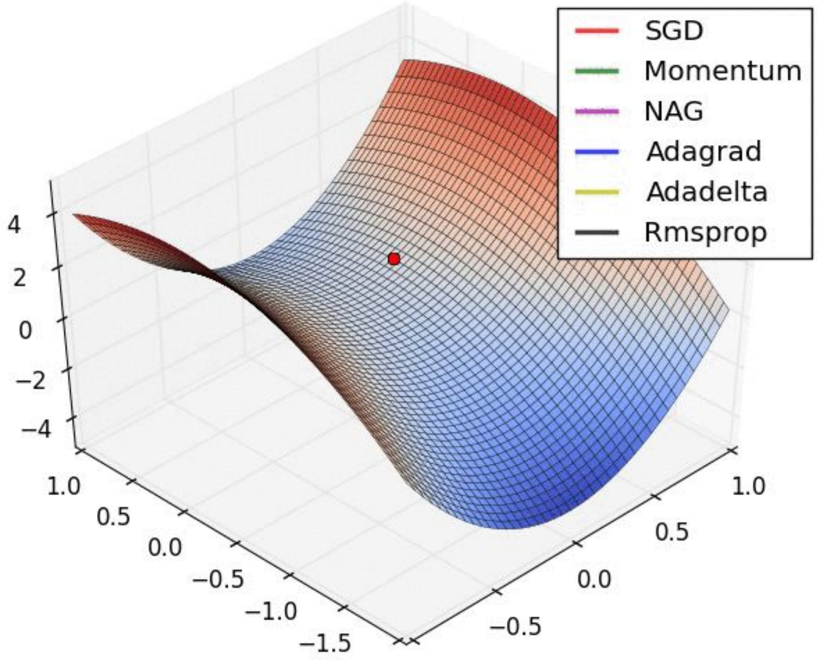
Adaptive Learning Rates

Intuition:

Have a different learning rate for each parameter.

Take bigger steps if a parameter has not been updated much recently.

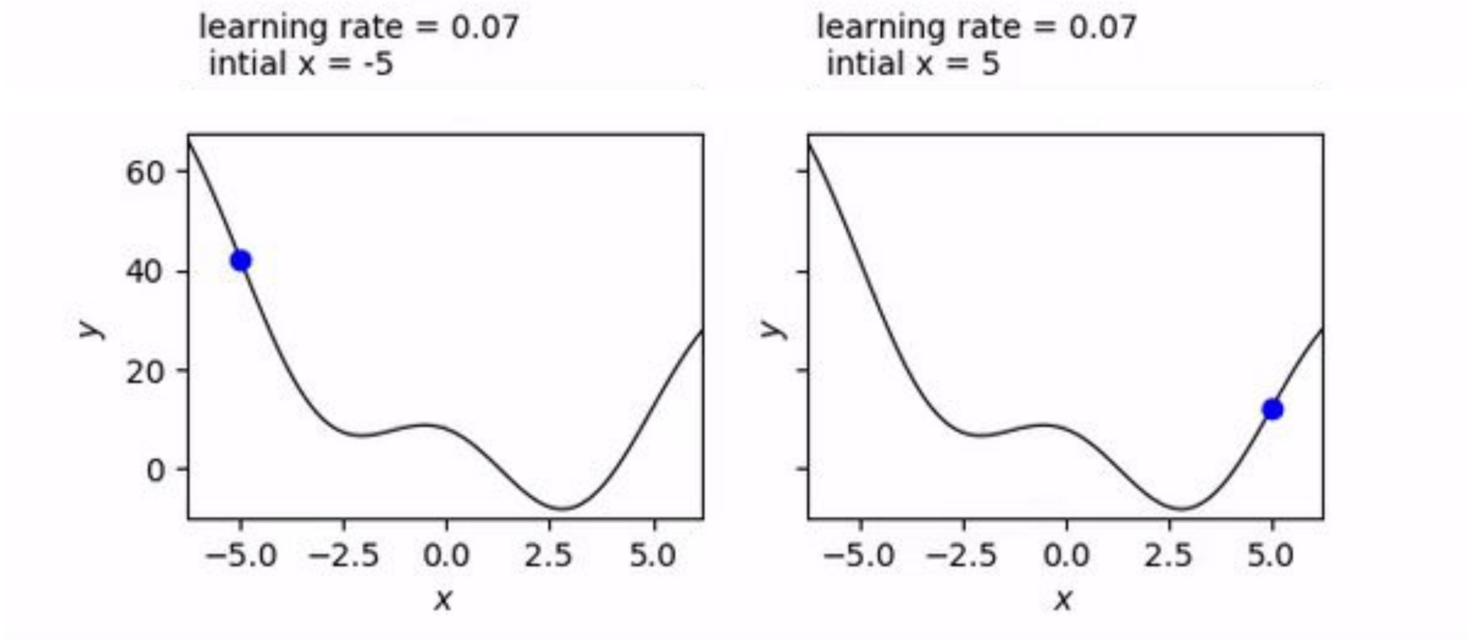
Take smaller steps if a parameter has been getting many big updates.



Alex Radford

Random initialization Matters

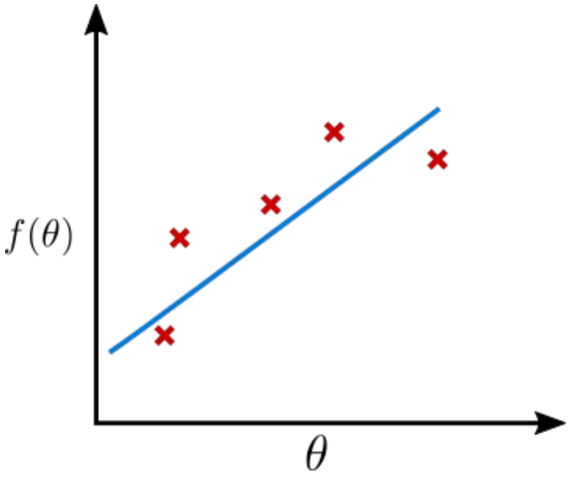
All other things being equal, just starting from a different location can lead to a different result.



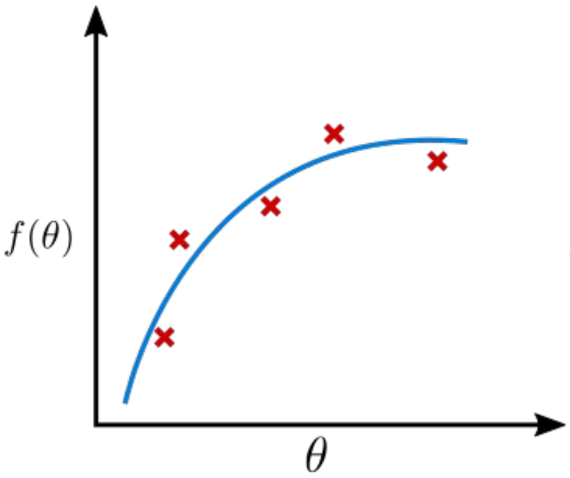
https://jed-ai.github.io/opt2_gradient_descent_1/

Fitting the Data

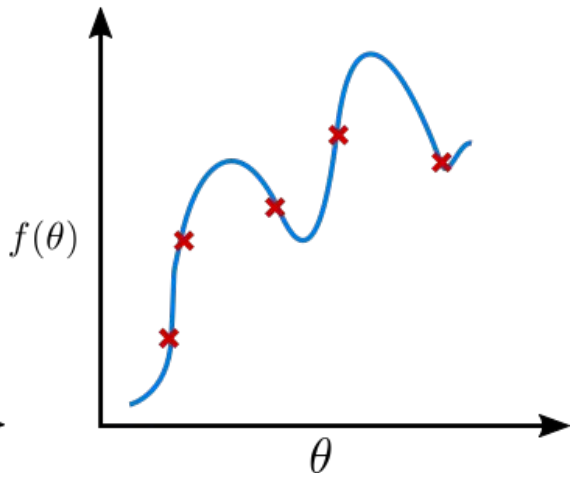
Underfitting
The model does not have the capacity to properly model the data.



Ideal fit



Overfitting
Too complex, the model memorizes the data, does not generalize.



Splitting the Dataset

In order to get realistic results for our experiments, we need to evaluate on a held-out test set.
Also using a separate development set for choosing hyperparameters is even better.

For training your models, fitting the parameters

For continuous evaluation and hyperparameter selection

For realistic evaluation once the training and tuning is done

Early Stopping

A sufficiently powerful model will keep improving on the training data until it overfits.
We can use the development data to choose when to stop.