Introduction to Neural Networks and Keras

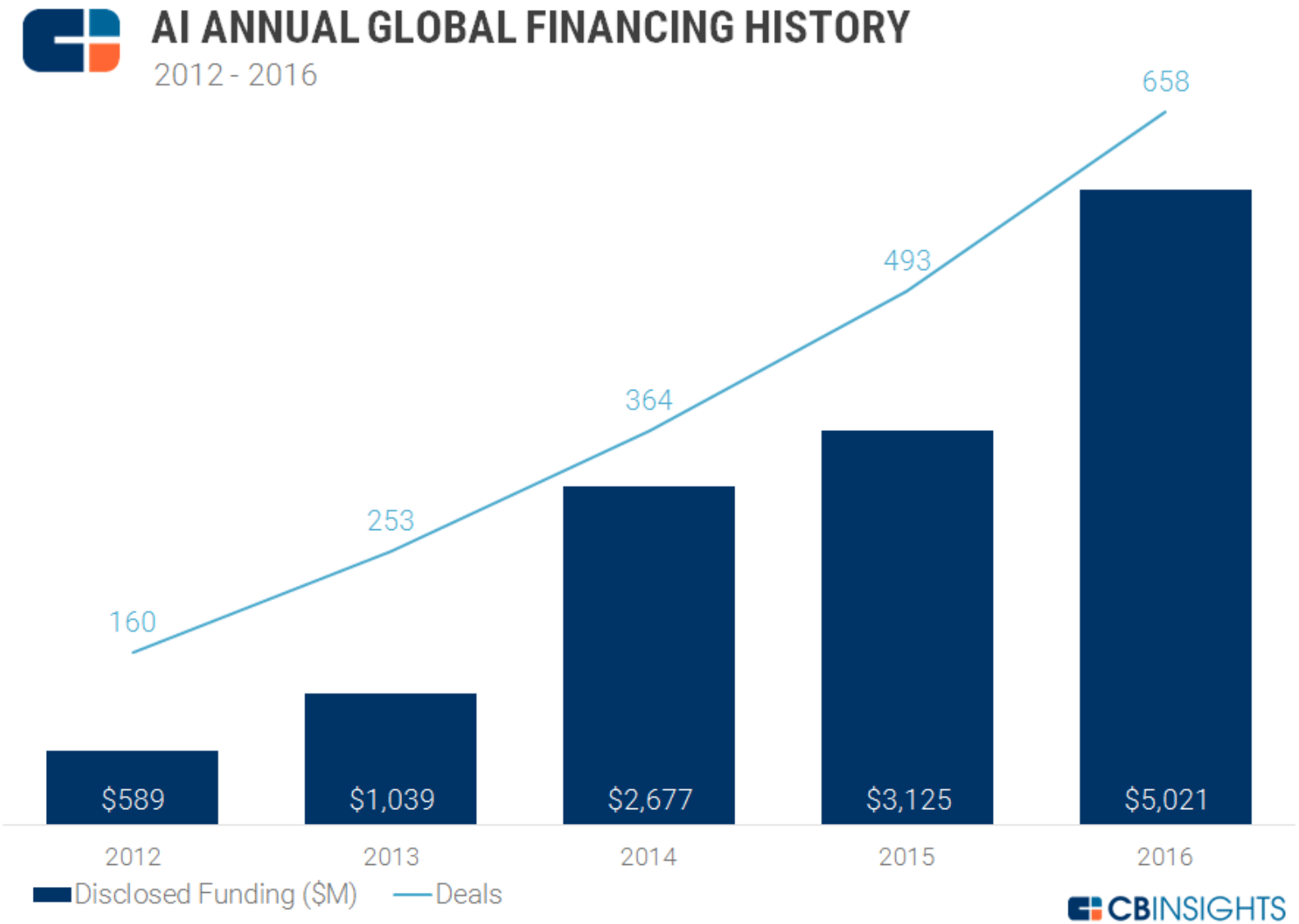
Meetup - Big data - Montpellier

Jie He

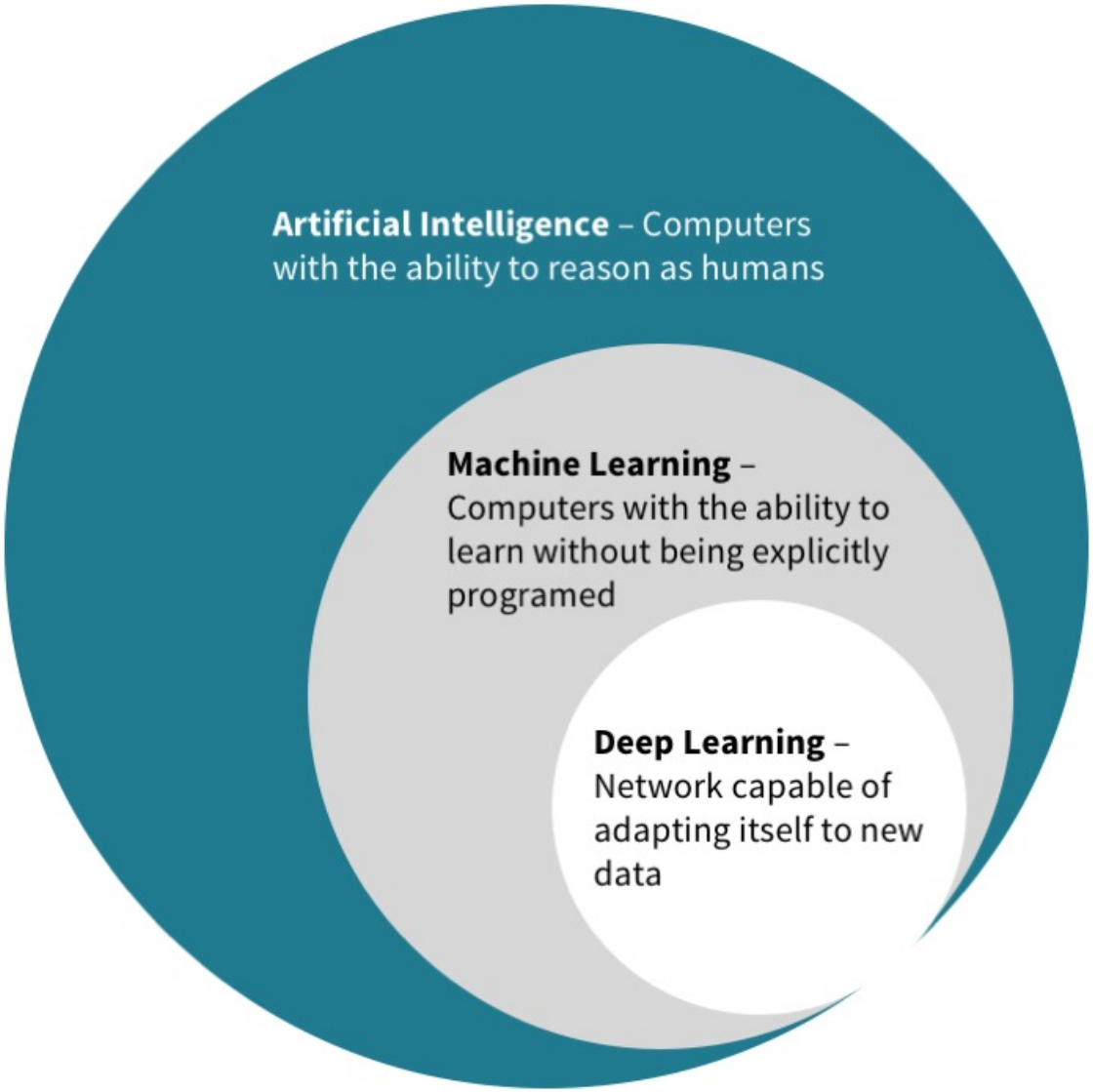
TabMo

June 29, 2017

Investment trending of Artificial Intelligence (AI)

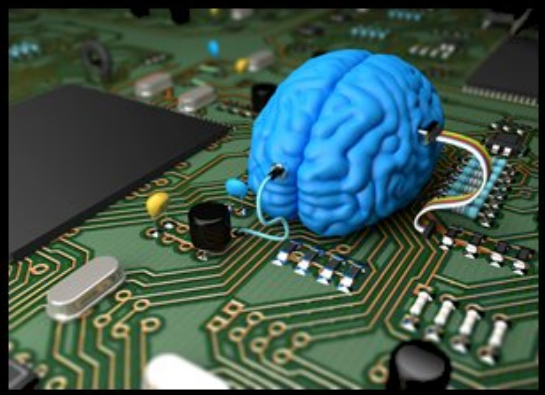


Deep Learning and Artificial Intelligence



Deep Learning

What society thinks I do



What my friends think I do

What other computer scientists think I do

What mathematicians think I do

What I think I do

```
In [1]:
import keras
Using TensorFlow backend.
```

What I actually do

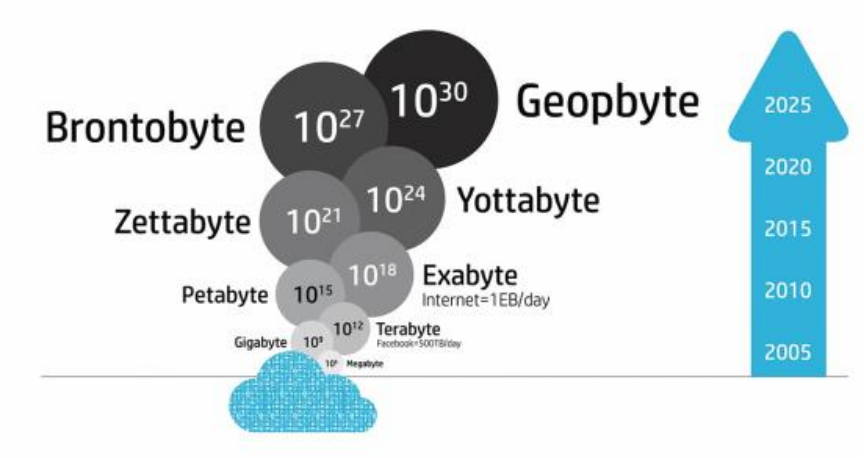
Outline

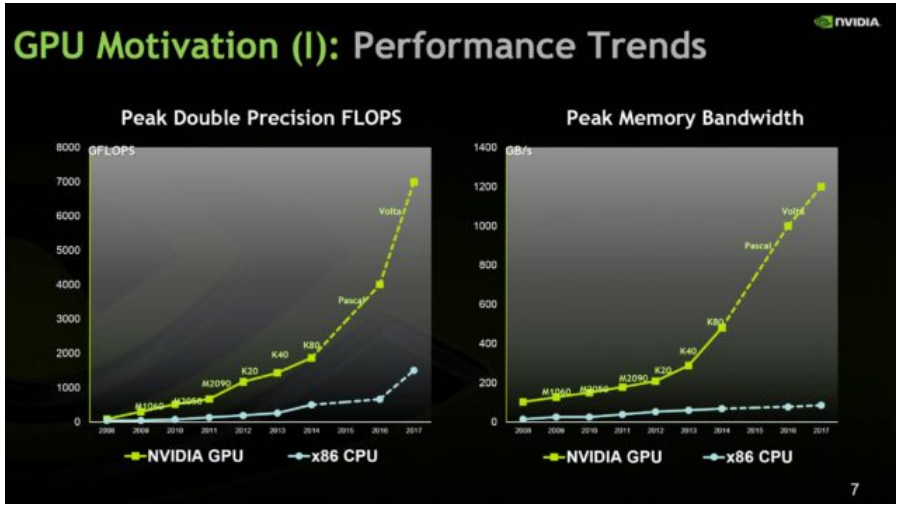
Outline

- **1943** - Walter Pitts and Warren McCulloch proposed the first **mathematical** model of a neural network.
- **1965** - Ivakhnenko and Lapa developed the earliest deep-learning-like algorithms which had **multiple** layers. But the training process was not optimized.
- **1970s** - **AI winter** came as a result of reduced funding and interest in AI research (which was due to the chain reaction of the AI hype).
- **1989** - Yann LeCun provided the first practical demonstration of **backpropagation** at Bell Labs. He also combined convolutional neural networks with backpropagation to read handwritten digits.
- **1985-90s** - the second **AI winter** kicked in. Nevertheless, some significant advances were made, e.g., support vector machine, long short-term memory.
- **And now** - AI regains its popularity. 62% of organizations will be using AI Technologies by 2018, according to the survey from Narrative Science.

Outline

Background

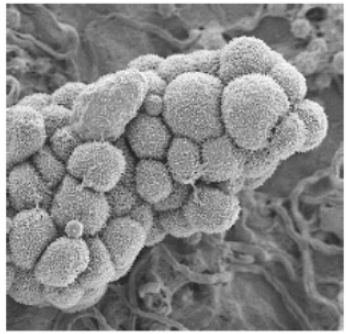
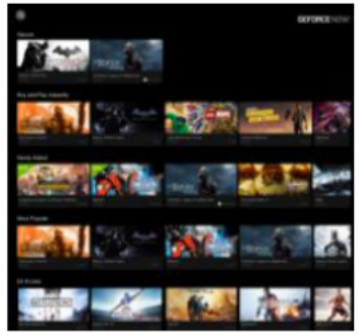




Availability of big data **Increase of computational power**

Improved algorithms/architectures

DEEP LEARNING EVERYWHERE

INTERNET & CLOUD	MEDICINE & BIOLOGY	MEDIA & ENTERTAINMENT	SECURITY & DEFENSE	AUTONOMOUS MACHINES
Image Classification	Cancer Cell Detection	Video Captioning	Face Detection	Pedestrian Detection
Speech Recognition	Diabetic Grading	Video Search	Video Surveillance	Lane Tracking
Language Translation	Drug Discovery	Real Time Translation	Satellite Imagery	Recognize Traffic Sign
Language Processing				
Sentiment Analysis				
Recommendation				

source:developer.nvidia.com/deep-learning-courses

Outline

Three kinds of variables

Categorical variables

A **finite number** of categories or distinct groups. Categorical data might not have a logical order. E.g., gender - male/female

Discrete variables

Numeric variables that have a **countable number** of values between any two values. E.g., the number of children in a family - 3.

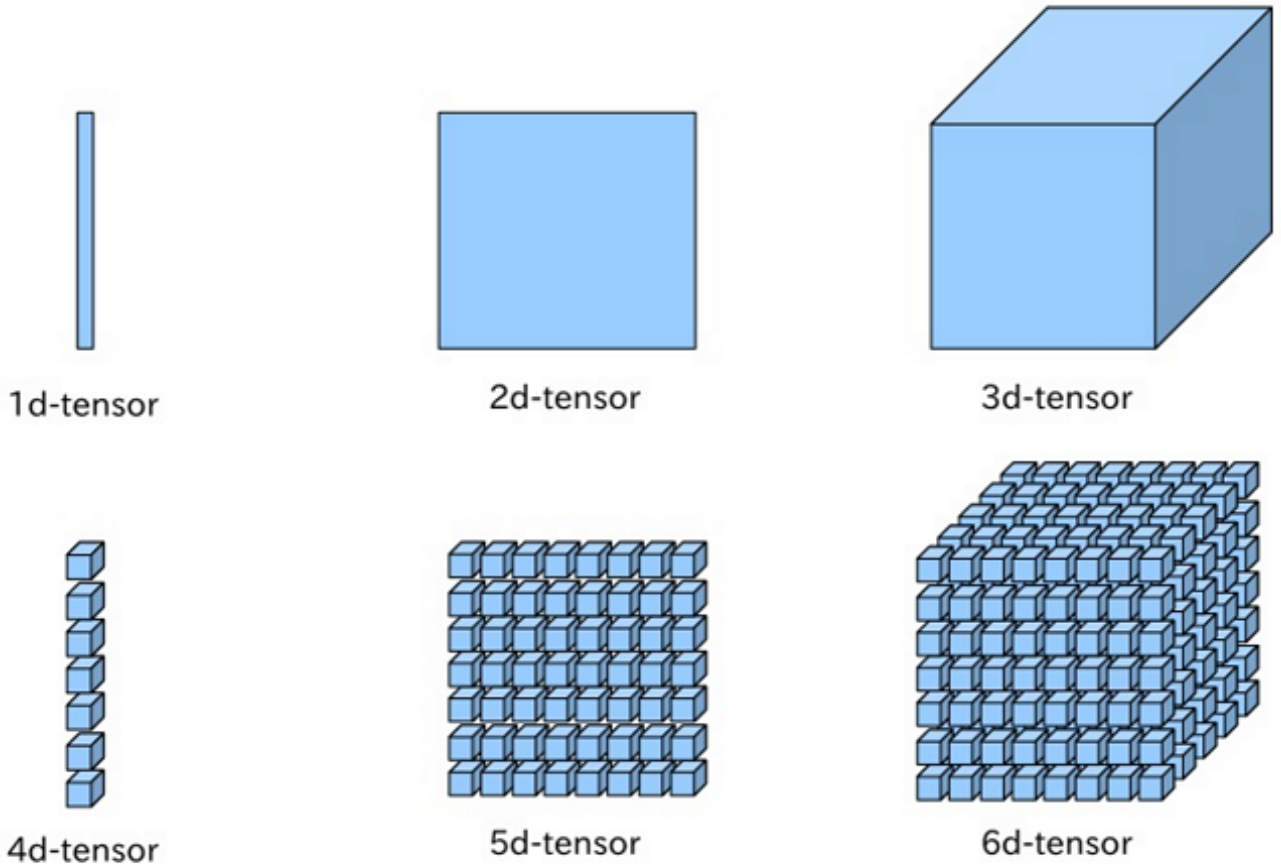
Continuous variables

Numeric variables that have an **infinite number** of values between any two values. E.g., temperature - 25.9

Tensor representation of data

What is tensor?

Tensor is a general name of multidimensional array numeric data.



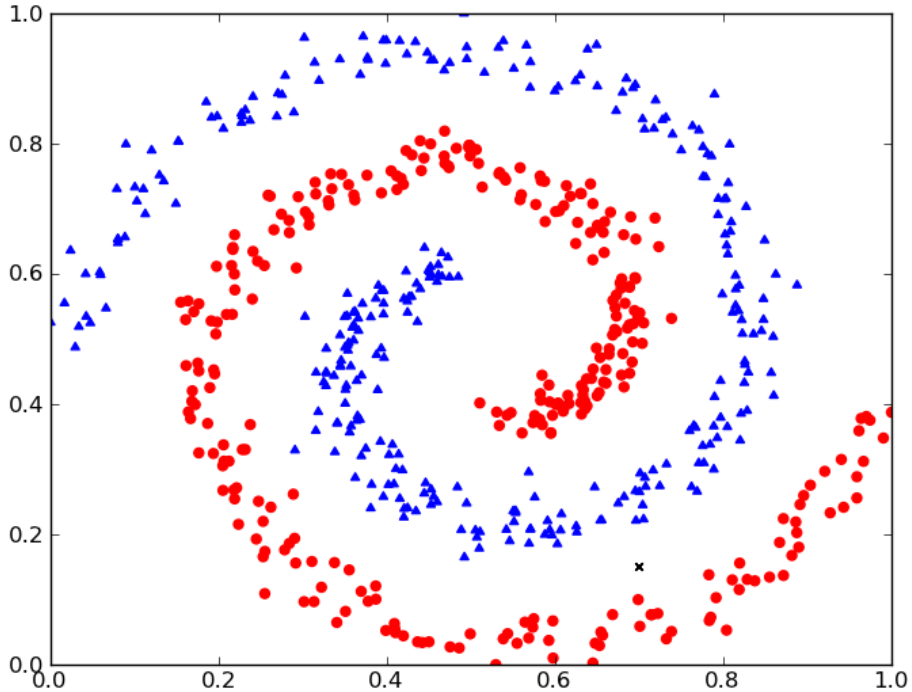
Outline

Problem definition

Binary classification

Table: Data examples

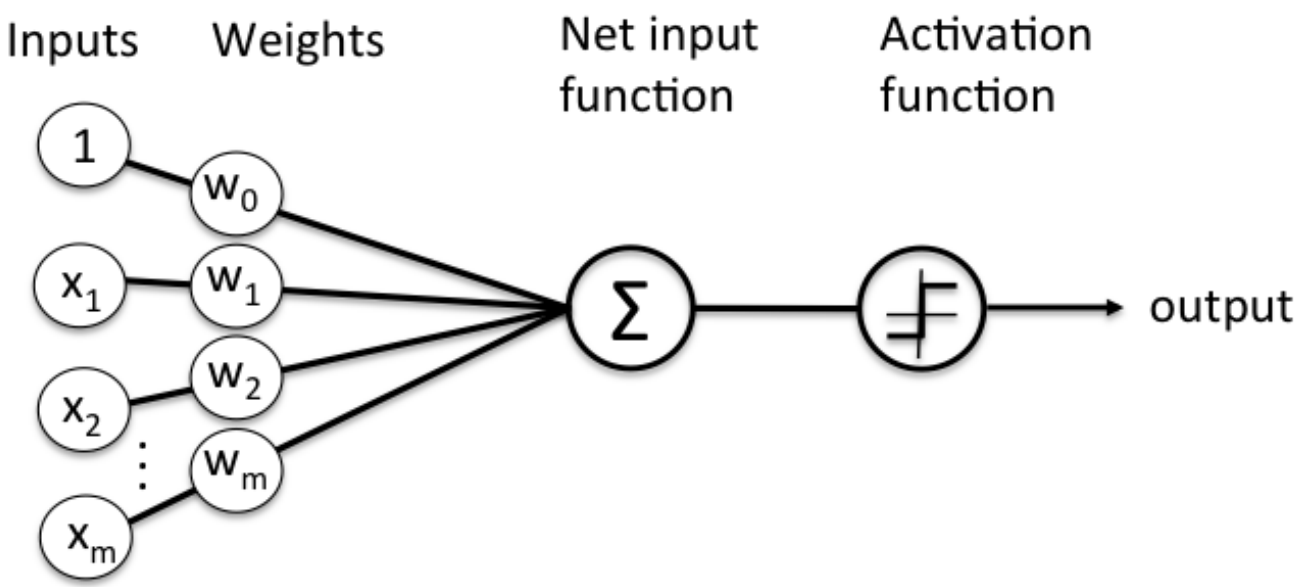
x1	x2	label
1.4	2.7	0
3.8	3.4	0
1.5	0.7	1
2.5	3.1	0
0.5	0.3	1
1.2	2.8	0
...	...	...



▸ Demo on data transformation

The magic behind the transformation

Perceptron

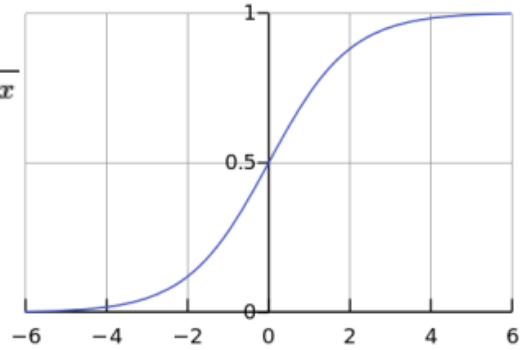


$$y = f(\sum x_i w_i)$$

The magic behind the transformation

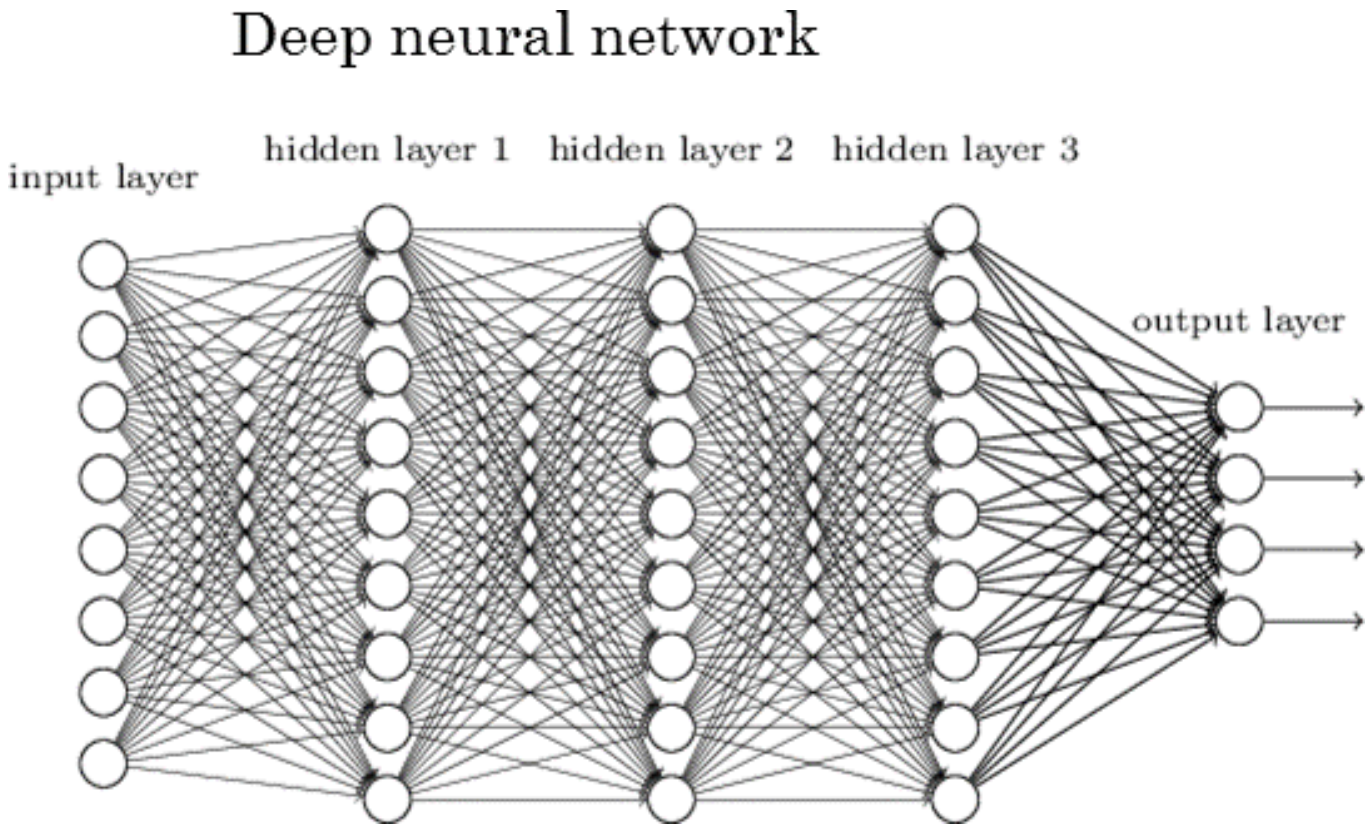
Perceptron

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



The magic behind the transformation

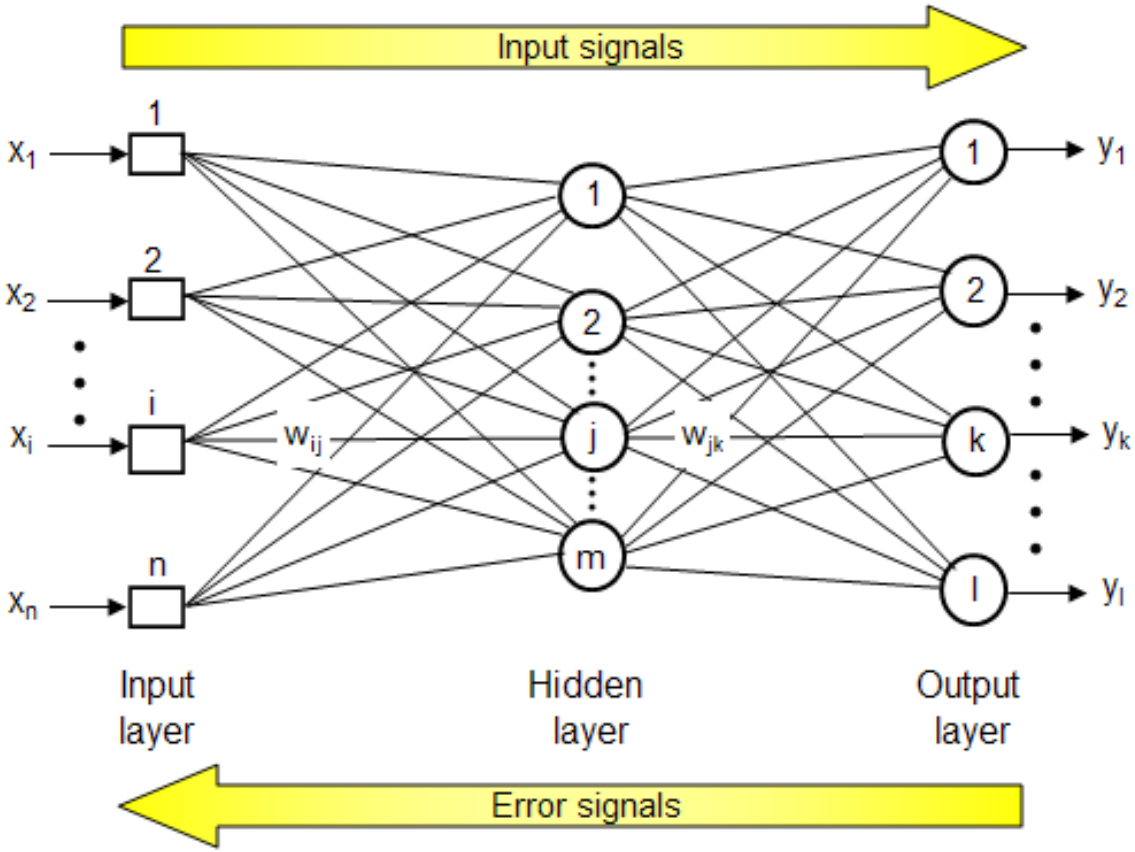
Stack layers of perceptron cells



Outline

Training a DNN means **finding the optimal weights**

Forward Propagation and Back Propagation



Outline

Which platform and language to use

Low-level: Google's **Tensorflow**, Microsoft's **CNTK**, Apple's **core ML**, Amazon's **mxnet**, and University of Montreal's **theano**, PyTorch...
High-level: Keras, **TFLearn**, **TensorLayer**

About Keras

Keras is a model-level library, providing high-level building blocks for developing deep learning models, and has plugged several different backend engines:**TensorFlow** backend, **Theano** backend and **CNTK** backend.

If you're a beginner and interested in quickly implementing your ideas

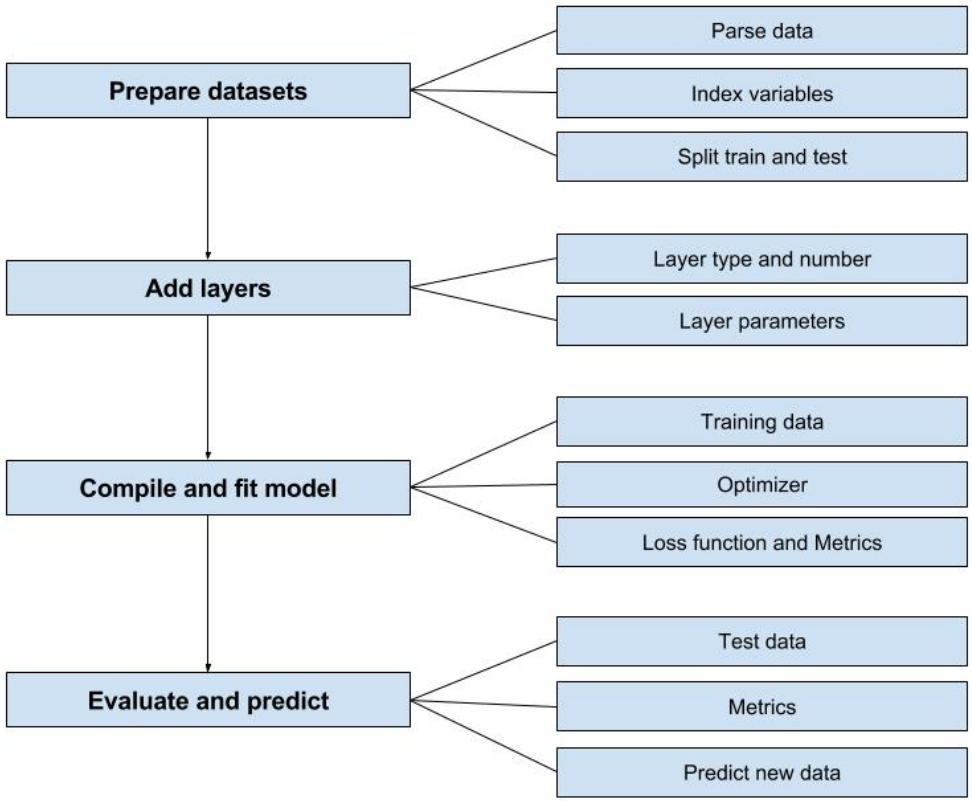
Python + Keras: Super fast implementation, good extensibility

If you want to do fundamental research in Deep Learning

Python + Tensorflow or PyTorch: Excellent extensibility

Outline

4 steps to apply a DNN model



Outline

1 Binary classification model

Table: Synthetic data

	f0	**f1**	**f2**	**f3**	**f4**	**f5**	**f6**	**f7**	**f8**	**f9**	**label**
0	16	41	3	95	45	8	19	56	95	40	0
1	26	79	51	85	55	40	35	7	54	70	1
2	17	79	48	14	17	37	3	84	66	22	0
3	85	42	12	70	51	9	51	51	0	37	0
4	88	28	61	22	80	52	10	74	7	27	1
...					...						...

Synthetic pattern

The label is **1** if:

$f0 > 67, f1 < 32, f2 \bmod 7 < 3, f3 \bmod 30 > 12$

▸ Demo on Sublime

2 Decimal addition model

How to do decimal addition on computer? Like $\begin{array}{r} 58 \\ +\ 26 \\ \hline \end{array}$?

Use a calculator, how simple that would be!

But... a Deep Neural Network can do that too,

and in a **human-like manner**!

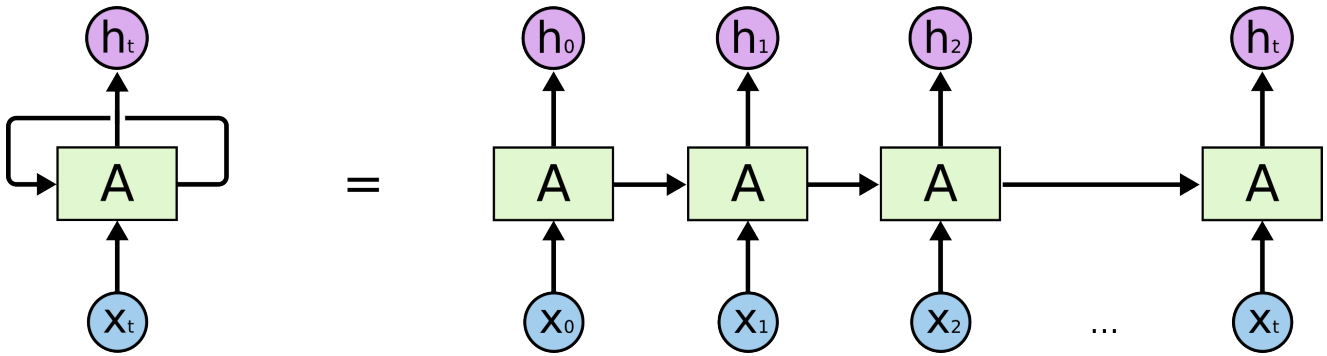


Figure: Recurrent Neural Network (from *here*)

▸ Demo on Sublime

Pros and cons about DNNs

Pros

- No need for feature engineering
- No limit for data volume
- Fast training on huge data
- Sustainability - incremental learning

Cons

- Too many hyperparameters and variants
- Easy to train a good model but difficult to get an excellent one
- Black box - how the model works

For Further Reading

Christopher Olah.
Neural Networks, Manifolds, and Topology

Jason Brownlee.
Time Series Prediction with LSTM Recurrent Neural Networks in Python with Keras

Andrew Trask.
A Neural Network in 11 lines of Python