MECA653

Introduction to Machine Learning for Mechanical Engineers

Pierre Nagorny, Spring 2020

Learning outcomes

Student will know

- Machine Learning basis and limits
- Classical Machine Learning problems
- Real-world applications to Dimensional Reduction
- The overfitting trade-off and basic methods to avoid it

Student will be able to

- Use Support Vector Machines for classification
- Use Convolutional Neural Networks for classification & regression
- Evaluate model performances using cross-validation

Today's lecture

- Part 0 – Introduction
- Part 1 – Classical Machine Learning
- Part 2 – Deep Learning
- Part 3 – Reinforcement Learning

Next practical works

- One Tutorial (1.5h) - Machine Learning on the MNIST dataset
- One Lab (4h) - Transfer Learning with CNN on a fun dataset
- *Bonus: Reinforcement Learning on OpenAI Gym*

0- Introduction

1– Machine Learning

2– Deep Learning

3– Reinforcement Learning

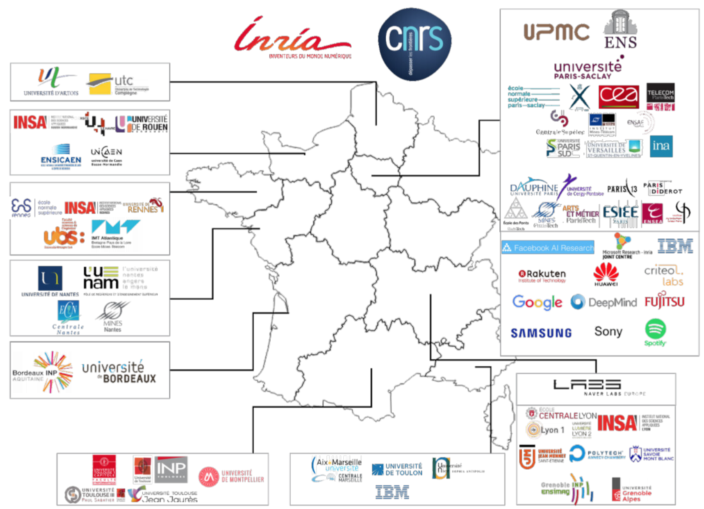
Machine Learning in France?

Artificial Intelligence **in France**

- 305+ startups
- 16 privates labs, 100+ publics
  - PRAIRIE (Paris Artificial Intelligence Research InstitutE): 1st dedicated institute.
  - [2018/11/22] final "Instituts Interdisciplinaires d'Intelligence Artificielle (3IA)" announced with 100M€ budget.

" France will invest **€1.5B** over the next five years in four related parts:

- Building a Data Focused Economic Policy
- Promoting Agile and Enabling Research
- Assessing the Effects of AI, AI working for a more Ecological Economy
- Ethical Considerations of AI and Inclusive and Diverse AI. "



Why Machine Learning as Mechanical Engineers?

- Extract meaningful informations from Big Data
- Statistically proved
- Need to anticipate outliners (robust models)

It's an optimization problem.

Danger in Statistics: Beware bias

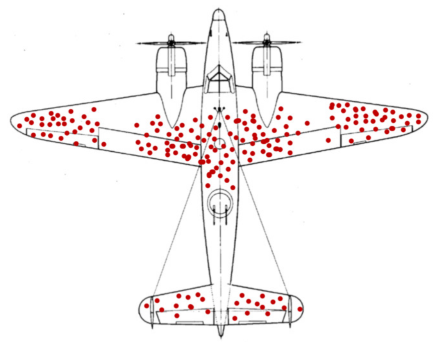


Image from Wikipedia's Survivorship bias article, based on the work of Abraham Wald

Part 1 - Classical Statistical Learning (ML)

1. Linear Regression
2. Support Vector Machine
3. Neural Networks

We will not talk about Random Forest, k-Nearest-Neighbors and many others important technics 😭

Linear Model

The basic tool we should deeply understand, and use the most.

" *Everything is a linear regression* " -> aka the **BLUE (Best Linear Unbiased Estimator)**.

0- Introduction

1– Machine Learning

2– Deep Learning

3– Reinforcement Learning

Linear Model

$$y = w_0 x_0 + w_1 x_1 + ... + w_m x_m$$

$$y = \sum_{j=0}^{m} w_m x_m$$

$$y = \mathbf{w}^T \mathbf{x}$$

To find X the Closed-form solution, we use:

$$\mathbf{w} = (X^T X)^{-1} X^T y$$

More discussion in Sebastian Raschka's Closed-form vs Gradient Descent

Robust Linear regression: RANSAC (RANdom SAmple Consensus).

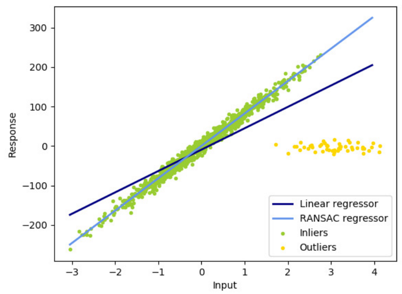


Image from SciKit documentation: Robust linear model estimation using RANSAC

Also Ridge, Lasso, Elastic-Net, LARS, LARS Lasso, Theil-Sen regressions.

Dimension Reduction

- Principal Component Analysis (PCA)
- Linear discriminant analysis (LDA by Sir Ronald Fisher in 1936)

More in Matthew Conlen & Fred Hohman great Beginner's Guide to Dimensionality Reduction.

Principal Component Analysis

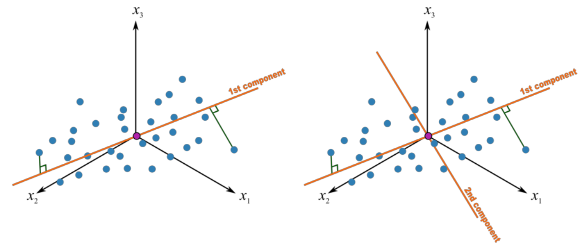


Image from Kevin Dunn's Process Improvement Using Data: 6.5.2. Geometric explanation of PCA

Principal Component Analysis

[1987] PCA on faces = eigenface, by Sirovich and Kirby

```
def PCA(X):
      M = np.dot(X, X.T)  # Covariance matrix
      e, EV = np.linalg.eigh(M)  # Eigenvalues and Eigenvectors
      V = np.dot(X.T,EV).T  # This is the 'compact' trick
      S = np.sqrt(e)
      return V / S
```

Demo using the Labeled Faces in the Wild dataset.

Support Vector Machines

- **Found the Maximum Margin**
- **Kernel trick**

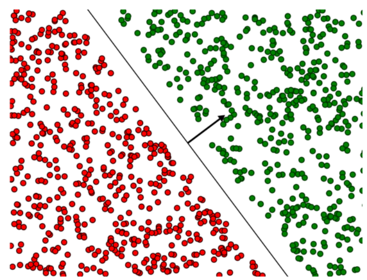
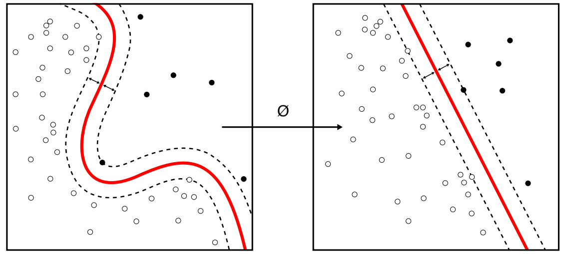


Image by Jeremy Kun's Formulating the Support Vector Machine Optimization Problem

Demo by [Jeremy Kun](#)

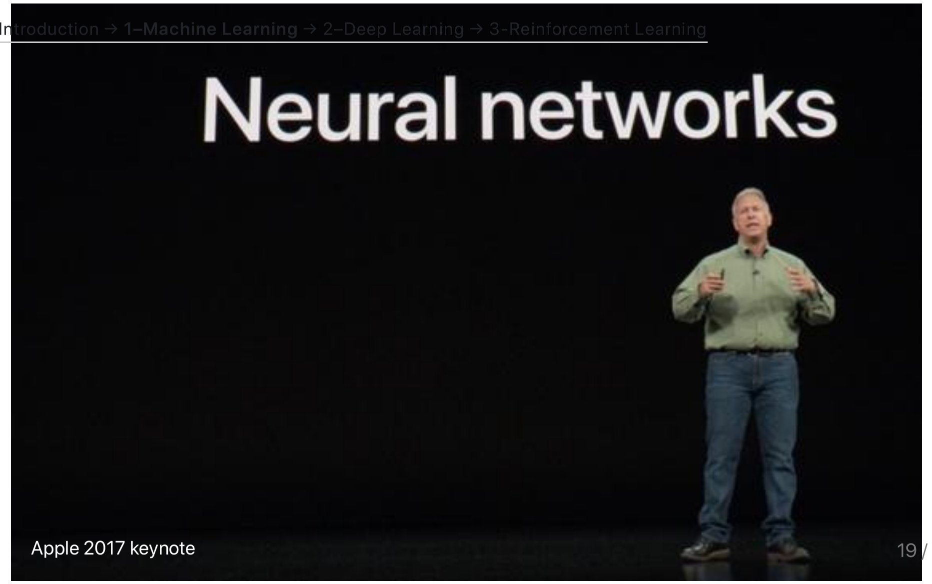
Support Vector Machines

The Kernel trick



[1992] The Kernel trick by Bernhard Boser, Isabelle Guyon, Vladimir Vapnik.

Image from Wikipedia, by Alisneaky & Zirguezi



Apple 2017 keynote

1957 Perceptron by Frank Rosenblatt

Neural Networks

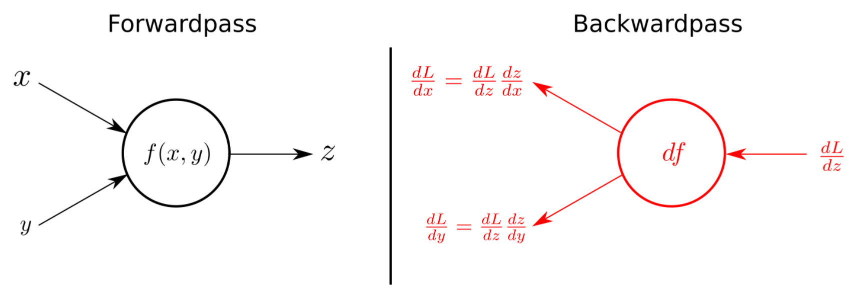


Image from Frederik Kratzert's Understanding the backward pass through Batch Norm Layer, 2016

What we have not talk about

Clustering

- K-Nearest Neighbor, K-Means

Ensemble methods

- Bagging, Boosting

Bayesian algorithms

- Random Forest, Monte Carlo

Evolutionary algorithms

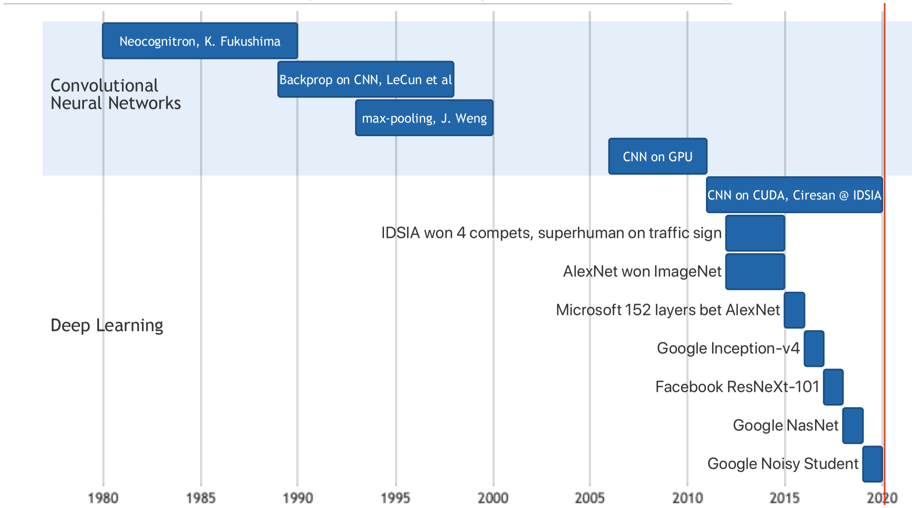
- Genetic algorithms

The Scikit-Learn documentation is an excellent resource to dig the field.

2 - Deep Learning

Why Deep Learning?

- Neural networks are **non-linear** models with monstruous **Degrees of Freedom**
- Many times, NN performs better than classical Machine Learning
- Solve problems on complex data (images, sounds, videos) with **Deep Convolutional Neural Networks**



AlexNet, when it *re*-began

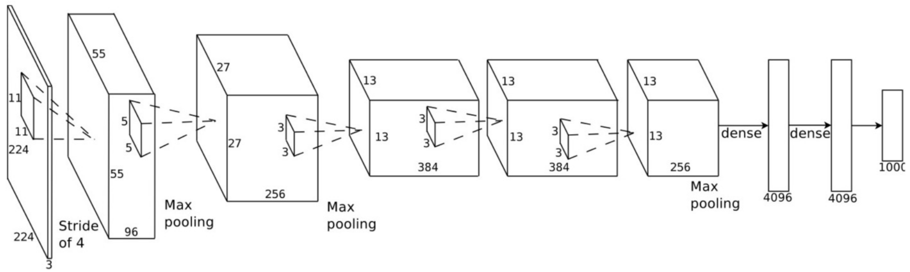
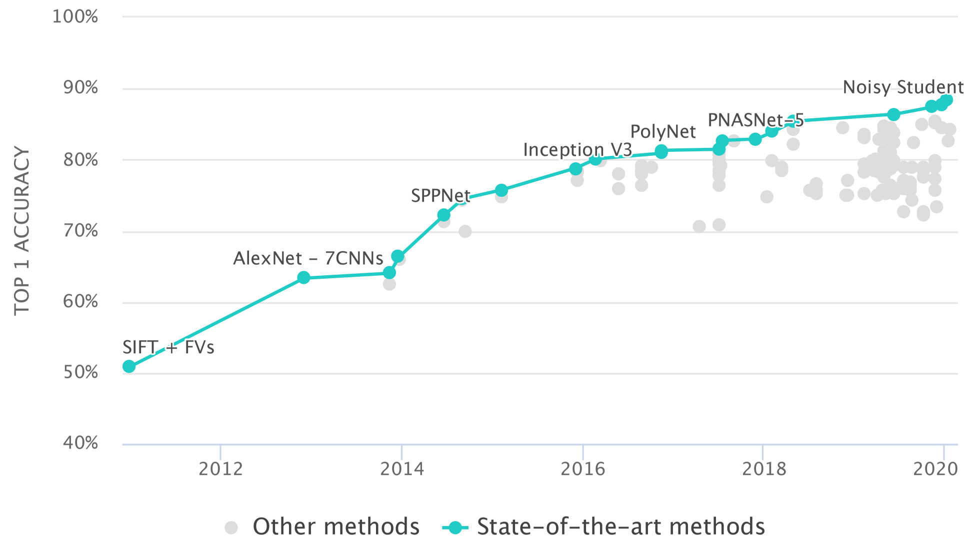


Image from *ImageNet Classification with Deep Convolutional Neural Networks*, by Alex Krizhevsky, Ilya Sutskever Geoffrey Hinton, Univ. Toronto, NIPS 2012

ImageNet 20000 categories - Andrej Karpathy's try-it-yourself



paperswithcode.com/sota/image-classification-on-imagenet

Applications of Deep Learning in Mechanics

Physics analysis and simulations

- 2017: The NIPS Deep Learning for Physical Science Workshop
- 2016: Accelerating Eulerian Fluid Simulation With Convolutional Networks: aka Google FluidNet.

3D generation (2D-to-3D)

- 2016: **Deep3D**: First Deep Learning approach on this problem
- 2018: **Pixel2Mesh**, **Neural Renderer**
- **2019: State of the Art**
  - **SMPL-X**: Expressive body capture: 3D hands, face, and body from a single image
  - **LiveCap**: Real-time Human Performance Capture from Monocular Video

How to Deep Learn?

- Big dataset
- Neural Networks
- Stack many layers, thus *Deep*
- Use specific layers like convolutions, residual memories, Long-Short-Term-Memories
- Buy GPUs

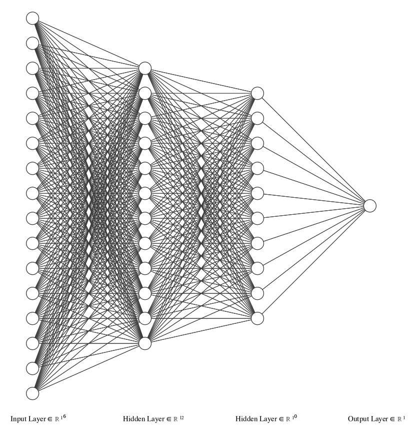


Image generated by Alexander Lenail's NN-SVG

A fully Connected Neural Networks by Adam Harley

Convolutional Neural Networks

Convolution (Forward pass)

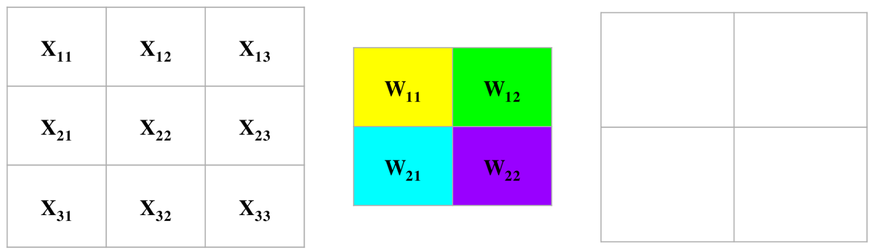


Image from Mayank Agarwal's Back Propagation in Convolutional Neural Networks—Intuition and Code

A Convolutional Neural Networks by Adam Harley

How to train a model?

- Linear Regression: find the **closed-form** solution (inverse or pseudo-inverse)
- Support Vector Machine: use the **Kernel trick**, then solve the linear problem
- Neural Networks: compute the **Gradient** then **Descent** and **repeat many times**

How does Gradient Descent work?

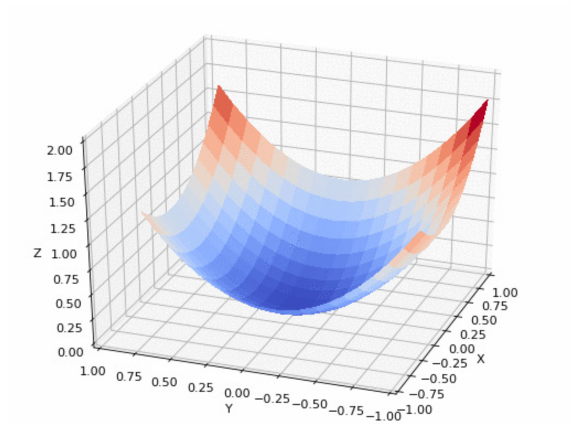


Image from Ayoosh Kathuria's Intro to Optimization in Deep Learning - Gradient Descent on Paperspace.com

How does Gradient Descent work?

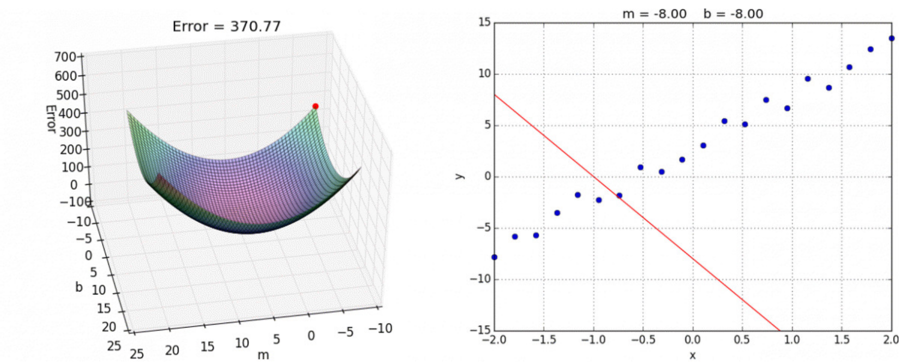


Image from Alykhan Tejani's Brief Introduction To Gradient Descent

Optimizing Gradient Descent

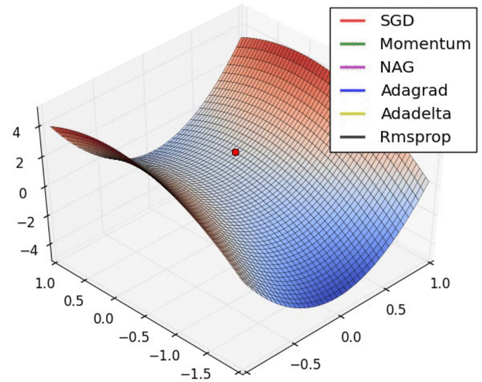
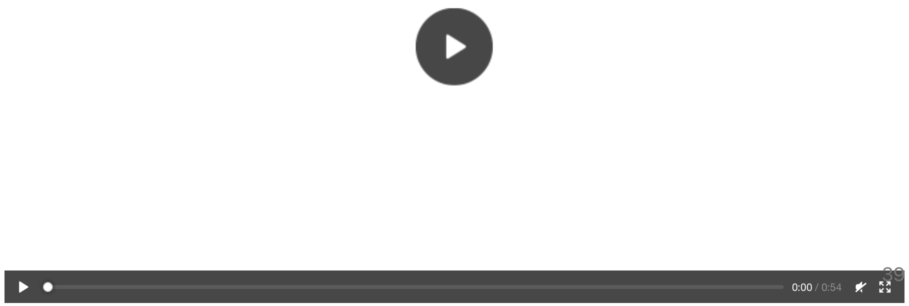
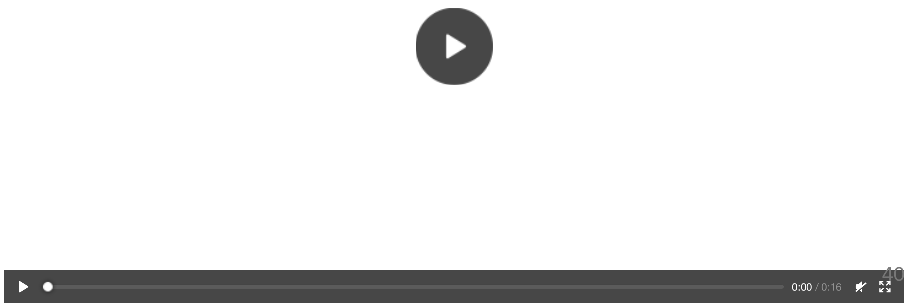


Image from Sebastian Ruder's Overview of gradient descent optimization algorithms

Gradient Descent on Neural Networks



Gradient Descent on Fully Connected Neural Nets



Gradient Descent on Convolutional Neural Networks

Derivative pass (Backward pass)

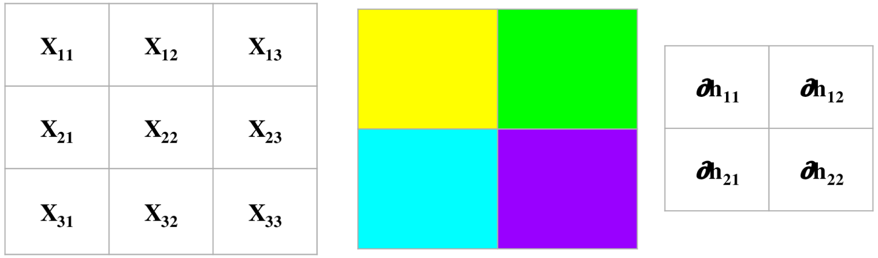
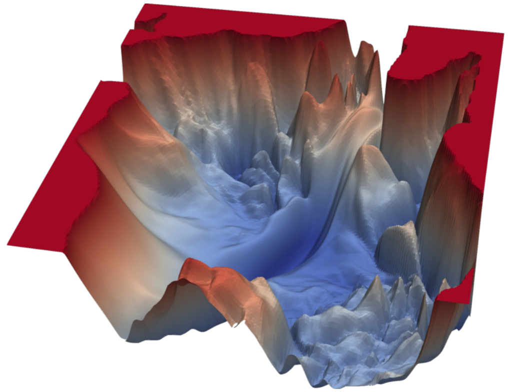


Image from Mayank Agarwal's Back Propagation in Convolutional Neural Networks—Intuition and Code

Gradient Descent on Deep Networks



Let's avoid local minima...

Overfitting is bad

- **What** is overfitting?
- **Why** is it bad?

- How could we **avoid** it?

Overfitting

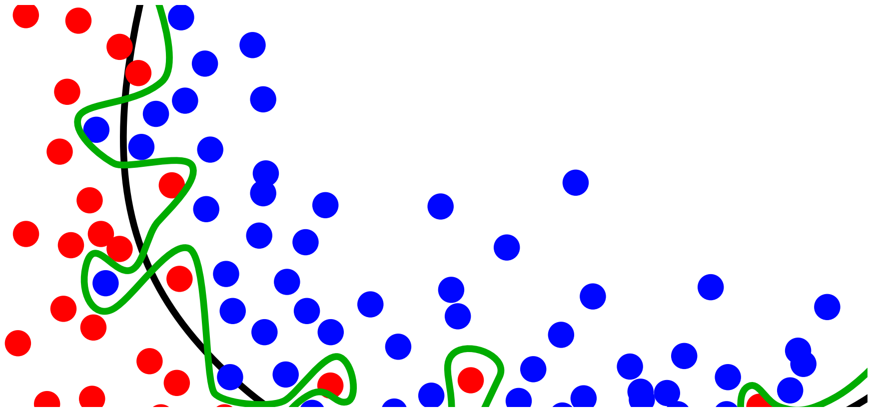
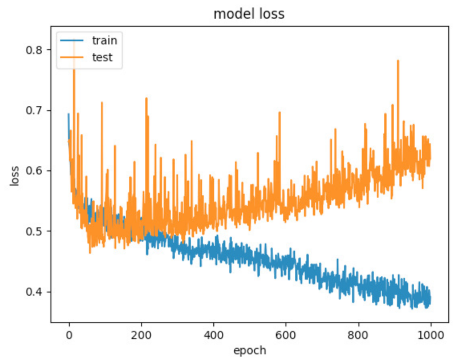


Image from Wikipedia By Chabacano (Ignacio Icke) - Own work, CC BY-SA 4.0

Overfitting

In practice



Transfer Learning

- Re-train/specialized a pre-trained networks on your alternive specific task
- Use this heavily in your mini-project if your dataset is small

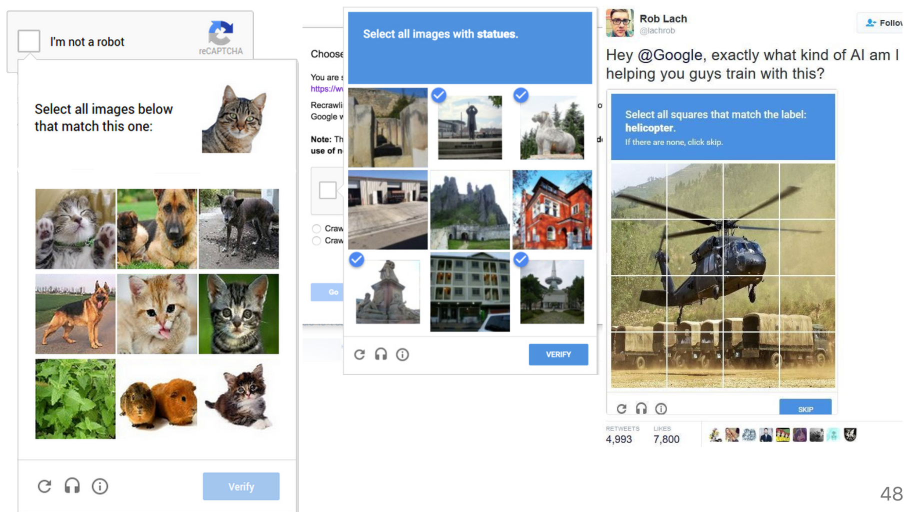
Deep Human Supervised Learning

A Deep Model needs a Deep Dataset

We are already training Software 2.0

- Google's ReCaptcha
- Facebook's Timeline
- Financial optimization (advertisement, social forcast)
- Facial recognition

A Deep Model needs a Deep Dataset



Toward Artificial General Intelligence (AGI)?

About the human brain:

- 100M of **Purkinje cells** with 100,000 synapses each
- Human brain: **1.075 × 10^21 FLOPS**
- **20 Watts**

About computers:

- Nov. 2018 Fastest computer: **1,4 x 10^17 FLOPS** (10^20 peak) (IBM Summit at Oak Ridge)
- **13 Mega Watts** (1 Nvidia GeForce Titan X: **250 watts**)

From Wikipedia's Animals' Number of Neurons
And Tim Dettmers's The Brain vs Deep Learning Part I

Next slides papers references:
Google's MoE-131072-h Shazeer & al., Outrageously Large Neural Networks
Facebook's ResNeXt-101–32x48d Mahajan & al., Exploring the Limits of Weakly Supervised Pretraining

Animal	Neurons	Synapses	Cerebral Cortex/Pallium
Sponge	0	0	No
Fruit fly	250k	$< 10^7$	No
2018 ResNeXt-101	**470k**	**829M**	?
Honey bee	960k	10^9	No
Frog	16M	?	?
2017 MoE-131072	**78M**	$\mathbf{10^{11}}$	?
Mouse	71M	10^{12}	14M
Cat	760M	10^{13}	250M
Dog	2,2B	?	530M
Human	80B	$\mathbf{1.5{\times}10^{14}}$	16B

0- Introduction

1– Machine Learning

2– Deep Learning

3– Reinforcement Learning

Mesuring AGI performance

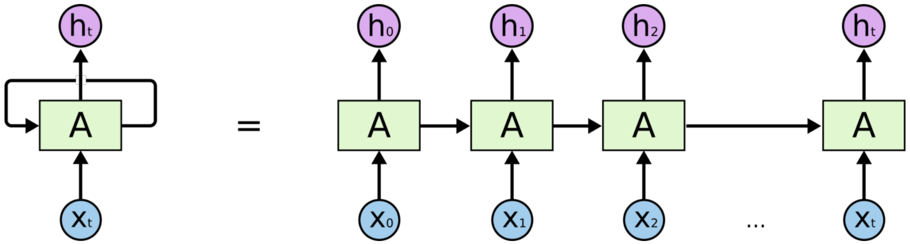
The AI Box Experiment

" A transhuman AI would just convince you to let it out " by Eliezer Yudkowsky.

Recurrent Neural Networks: a key to AGI

Long Short Term Memory (LSTM)

[1997] by Sepp Hochreiter & Jürgen Schmidhuber then Felix Gers, Fred Cummins, then many...



Images and great explanations in Christopher Olah's Understanding LSTM Networks

Reinforcement Learning milestones

- [2015/10] DeepMind's AlphaGo on Go from human dataset
- [2017/04] DeepMind's AlphaGo Zero on Go **without human dataset**
- [2017/12] DeepMind's AlphaZero on Go, Chess, Shogi **without human dataset**
- [2018/06] OpenAI's Five on Dota2
  - Plays **180 years** worth of games against itself **every day**
  - 256 GPUs and 128,000 CPU cores
  - **A separate LSTM for each 5 heros**
  - Learned recognizable strategies.
- [2019/01] DeepMind AlphaStar on StarCraft2
- [2019/02] OpenAI GPT-2 transformer-based language model
  - Not released due to malicious concerns
  - Sometimes writes about fires happening underwater

Learning to dress

Learning To Dress: Synthesizing Human Dressing Motion via Deep Reinforcement Learning, Clegg & al., 2018

Beware your reward function

OpenAI's Faulty Reward Functions in the Wild

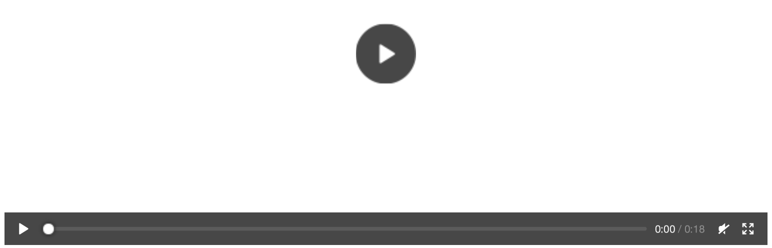
Learning strategy example: Evolution

Going Meta: Learning to Learn

World Models

Ha and Schmidhuber, "Recurrent World Models Facilitate Policy Evolution", 2018.

World Models: let's model the World with Deep Nets



Ha and Schmidhuber, "Recurrent World Models Facilitate Policy Evolution", 2018.

World Models: Model the World with Deep Nets

World Models: Model the World with Deep Nets

World Models: Can agents learn inside of their own dreams?

https://worldmodels.github.io/demo/carrnn_demo.html

https://worldmodels.github.io/doomrnn/

Conclusion: Is Machine Learning real better?

About AI failure

- Specification gaming examples in AI, Spreadsheet list.

Necessity of Interpretability for Machine Learning models

- Grad-CAM for Convolutionnal Networks

Today's summary

Machine Learning

1. Linear Model
2. Dimension Reduction
3. Classifier SVM & Neural networks

Deep Learning

1. Architectures
2. Learning with Gradient Descent
3. Overfitting

Reinforcement Learning

1. Applications
2. Meta Learning

Enjoy Machine Learning! 👍

Slides made with Marpit.