Introduction to Deep Learning

Charles Ollion - Olivier Grisel

Goal of the class

Overview

- When and where to use DL
- "How" it works
- Frontiers of DL

Goal of the class

Overview

- When and where to use DL
- "How" it works
- Frontiers of DL

Arcanes of DL

- Implement using `Numpy`, and `Tensorflow` (`Keras`)
- Engineering knowledge for building and training DL

What is Deep Learning

Good old Neural Networks, with more layers/modules

What is Deep Learning

Good old Neural Networks, with more layers/modules

Non-linear, hierarchical, abstract representations of data

What is Deep Learning

Good old Neural Networks, with more layers/modules

Non-linear, hierarchical, abstract representations of data

Flexible models with any input/output type and size

What is Deep Learning

Good old Neural Networks, with more layers/modules

Non-linear, hierarchical, abstract representations of data

Flexible models with any input/output type and size

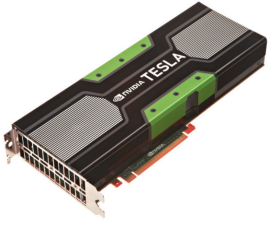
Differentiable Functional Programming

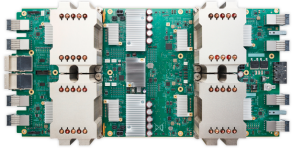
Why Deep Learning Now?

- Better algorithms & understanding
- Computing power (GPUs, TPUs, ...)
- Data with labels
- Open source tools and models

Why Deep Learning Now?

- Better algorithms & understanding
- Computing power (GPUs, TPUs, ...)
- Data with labels
- Open source tools and models

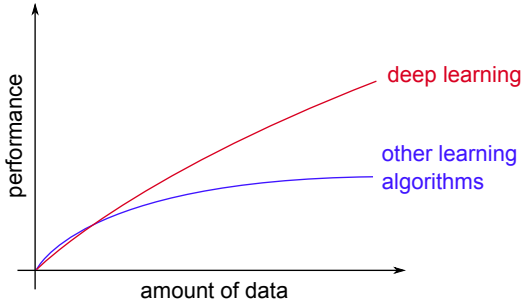




GPU and TPU

Why Deep Learning Now?

- Better algorithms & understanding
- Computing power (GPUs, TPUs, ...)
- Data with labels
- Open source tools and models

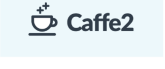


Why Deep Learning Now?

- Better algorithms & understanding
- Computing power (GPUs, TPUs, ...)
- Data with labels
- Open source tools and models

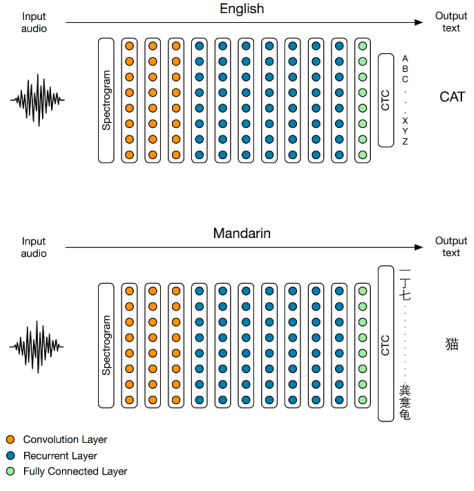






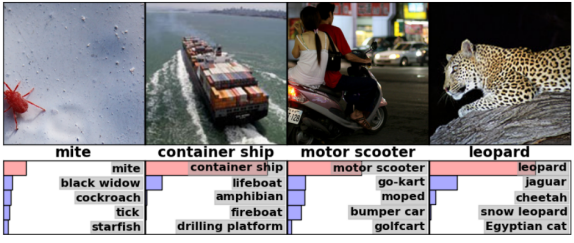


DL Today: Speech-to-Text



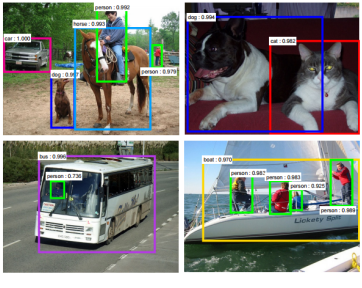
[Baidu 2014]

DL Today: Vision



[Krizhevsky 2012]

[Ciresan et al. 2013]

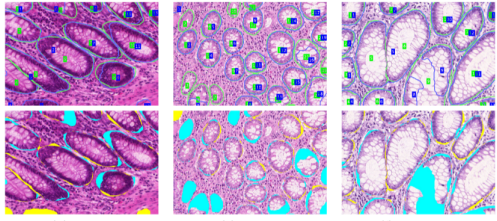


[Faster R-CNN - Ren 2015]

[NVIDIA dev blog]

DL Today: Vision

[Stanford 2017]



[Nvidia Dev Blog 2017]

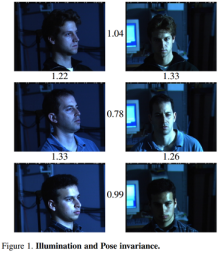
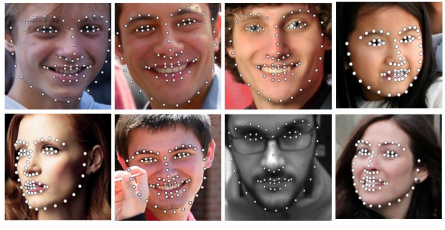


Figure 1. **Illumination and Pose invariance.**

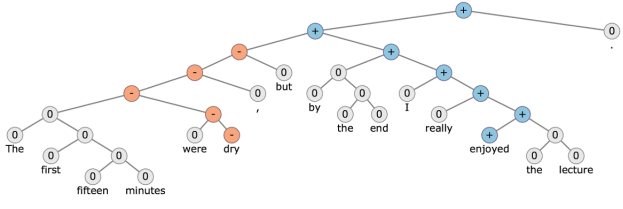
[FaceNet - Google 2015]



[Facial landmark detection CUHK 2014]

DL Today: NLP

[Google Translate System - 2016]



[Socher 2015]

DL Today: NLP

[Google Inbox Smart Reply]

[Amazon Echo / Alexa]

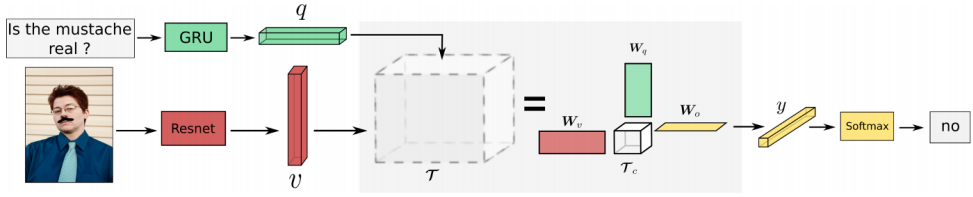
DL Today: NLP

[Google Inbox Smart Reply]

[Amazon Echo / Alexa]

Most of chatbots claiming "AI" do not use Deep Learning (yet?)

DL Today: Vision + NLP



[VQA - Mutan 2017]

"man in black shirt is playing guitar."

"construction worker in orange safety vest is working on road."

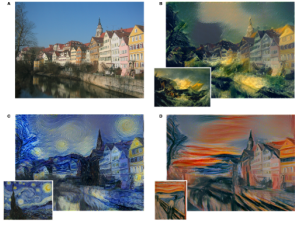
"two young girls are playing with lego toy."

"boy is doing backflip on wakeboard."

[Karpathy 2015]

DL Today: Image translation

[DeepDream 2015]



[Gatys 2015]

[Ledig 2016]

DL Today: Generative models

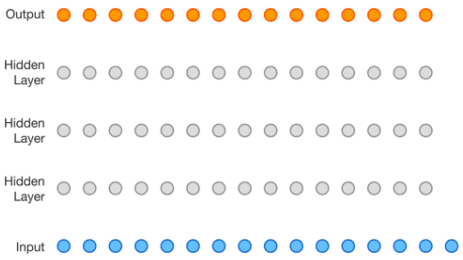
Sampled celebrities [Nvidia 2017]

DL Today: Generative models

Sampled celebrities [Nvidia 2017]

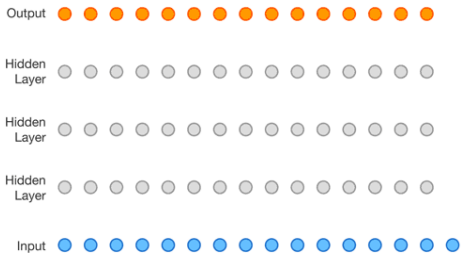
StackGAN v2 [Zhang 2017]

DL Today: Generative models



Sound generation with WaveNet [DeepMind 2017]

DL Today: Generative models



Sound generation with WaveNet [DeepMind 2017]

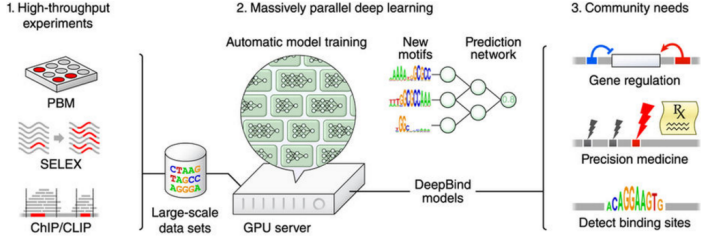
Guess which one is generated?

Tacotron 2 Natural TTS Synthesis by Conditioning WaveNet on Mel Spectrogram

Language / Image models

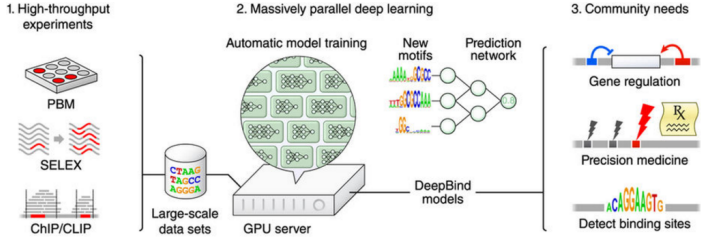
Open-AI GPT-3, or DALL-E: https://openai.com/blog/dall-e/

DL in Science: Genomics

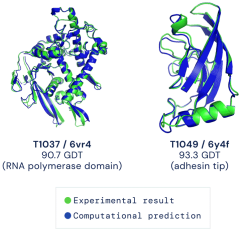


[Deep Genomics 2017]

DL in Science: Genomics

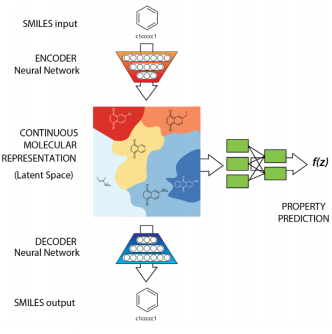


[Deep Genomics 2017]



AlphaFold by DeepMind

DL in Science: Chemistry, Physics



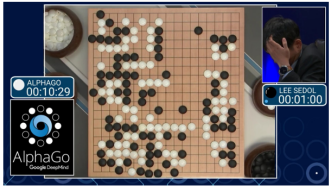
[Gómez-Bombarelli 2016]

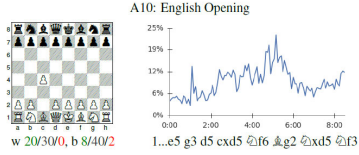
[Tompson 2016]

DL in Science: Chemistry, Physics

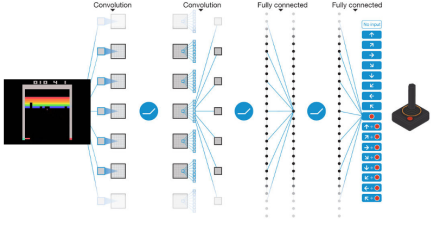
- Finite element simulator accelerated (~100 fold) by a 3D convolutional network

DL for AI in games

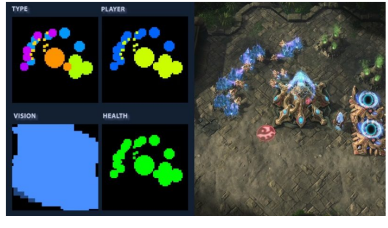




[Deepmind AlphaGo / Zero 2017]

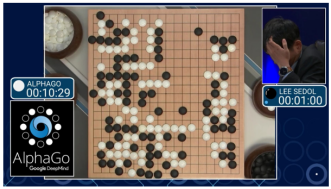


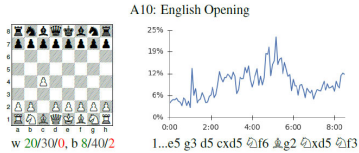
[Atari Games - DeepMind 2016]



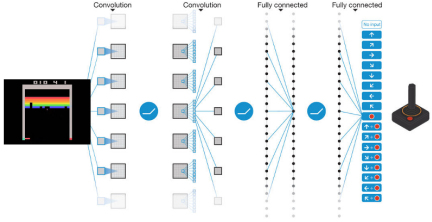
[Starcraft 2 for AI research]

DL for AI in games

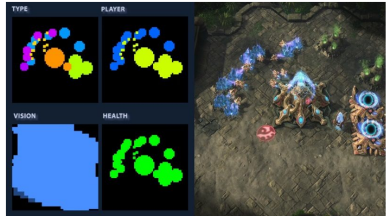




[Deepmind AlphaGo / Zero 2017]



[Atari Games - DeepMind 2016]



[Starcraft 2 for AI research]

AlphaGo/Zero: Monte Carlo Tree Search, Deep Reinforcement Learning, self-play

Outline of the class

Backpropagation

Outline of the class

Backpropagation

Computer Vision (2)

Outline of the class

Backpropagation

Computer Vision (2)

Recommender Systems

Outline of the class

Backpropagation

Computer Vision (2)

Recommender Systems

Natural Language Processing (2)

Outline of the class

Backpropagation

Computer Vision (2)

Recommender Systems

Natural Language Processing (2)

Optimization: theory, methods and tricks (2)

Outline of the class

Backpropagation

Computer Vision (2)

Recommender Systems

Natural Language Processing (2)

Optimization: theory, methods and tricks (2)

Generative models & unsupervised learning

How this unit works

Lectures 1h-1h30

- Can include a Quiz on Moodle (from time to time)
- Small part of the final grade

How this unit works

Lectures 1h-1h30

- Can include a Quiz on Moodle (from time to time)
- Small part of the final grade

Coding sessions 2h-2h30

- BYO laptop, you can work in pairs
- Homework 3h per session (finish notebooks, read solutions)

How this unit works

Lectures 1h-1h30

- Can include a Quiz on Moodle (from time to time)
- Small part of the final grade

Coding sessions 2h-2h30

- BYO laptop, you can work in pairs
- Homework 3h per session (finish notebooks, read solutions)

Final Project

- Project of your choice in teams of 2-4 people.
- Create a private channel on slack to decide on a topic with instructors as early as possible.

Recommended reading

- deeplearningbook.org: Math and main concepts
- Francois Chollet's book: Keras programming
- Aurélien Géron's book: Generic Machine Learning with Scikit-learn and Deep Learning with TF/Keras

Frameworks and Computation Graphs

Libraries & Frameworks

PYTORCH

Microsoft CNTK

Caffe2

dmlc mxnet

gensim

spaCy

theano

This lecture is using **Keras**: high level frontend for **TensorFlow** (and MXnet, Theano, CNTK)

Libraries & Frameworks

PYTORCH

Microsoft CNTK

Caffe2

dmlc mxnet

gensim

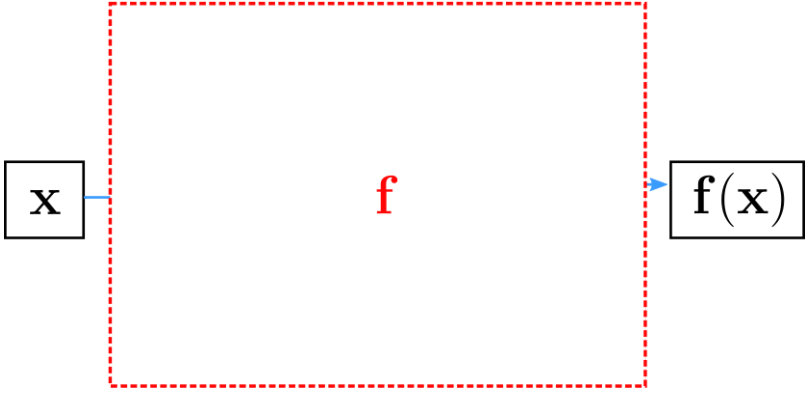
spaCy

theano

This lecture is using **Keras**: high level frontend for **TensorFlow** (and MXnet, Theano, CNTK)

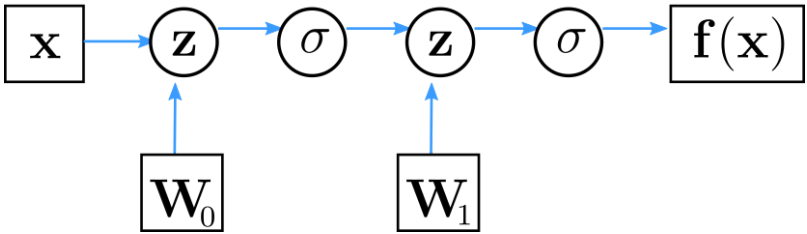
One lab will be dedicated to a short **Pytorch** introduction.

Computation Graph



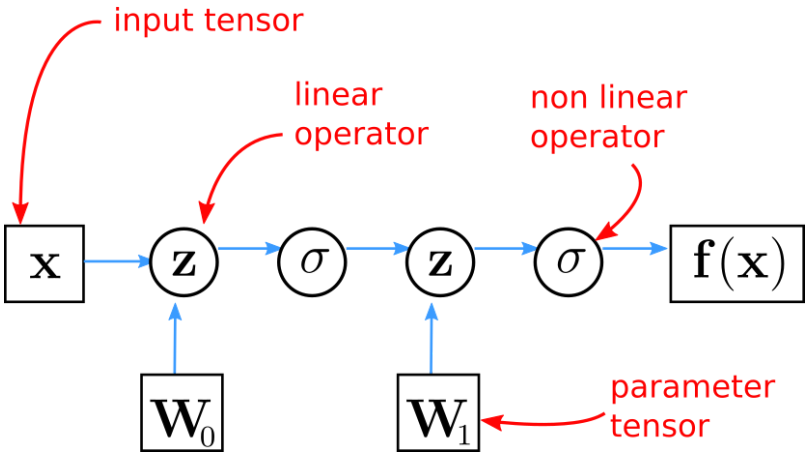
Neural network = parametrized, non-linear function

Computation Graph



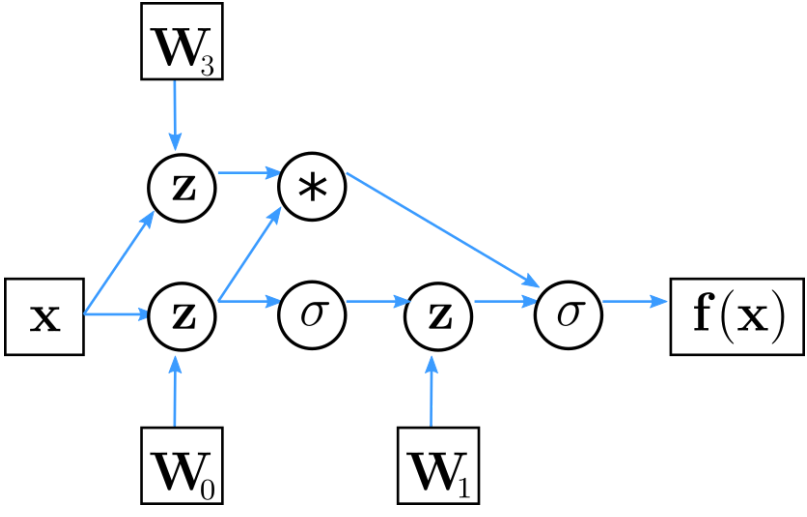
Computation graph: Directed graph of functions, depending on parameters (neuron weights)

Computation Graph



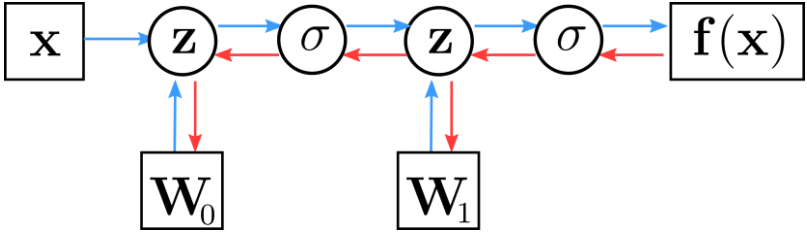
Combination of linear (parametrized) and non-linear functions

Computation Graph



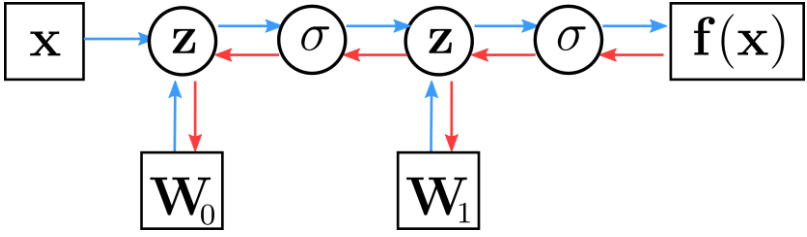
Not only sequential application of functions

Computation Graph



Automatic computation of gradients: all modules are **differentiable**!

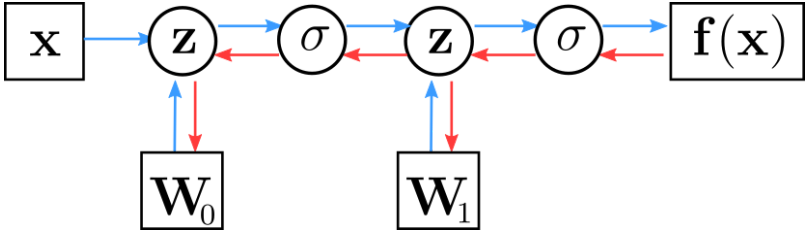
Computation Graph



Automatic computation of gradients: all modules are **differentiable**!

Theano (now Aesara), **Tensorflow 1**, etc. build a static computation graph via static declarations.

Computation Graph

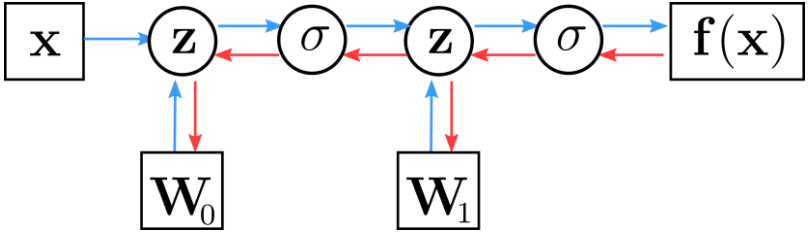


Automatic computation of gradients: all modules are **differentiable**!

Theano (now Aesara), **Tensorflow 1**, etc. build a static computation graph via static declarations.

Tensorflow 2, **PyTorch**, **JAX**, etc. rely on dynamic differentiable modules: "define-by-run".

Computation Graph



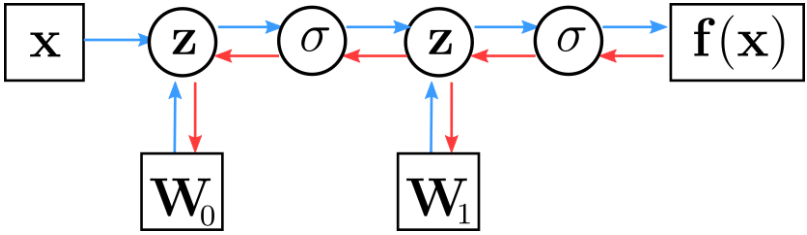
Automatic computation of gradients: all modules are **differentiable**!

Theano (now Aesara), **Tensorflow 1**, etc. build a static computation graph via static declarations.

Tensorflow 2, **PyTorch**, **JAX**, etc. rely on dynamic differentiable modules: "define-by-run".

Vector computation on **CPU** and accelerators (**GPU** and **TPU**).

Computation Graph



Simple keras implementation

```
model = Sequential()
model.add(Dense(H, input_dim=N))  # defines W0
model.add(Activation("tanh"))
model.add(Dense(K))               # defines W1
model.add(Activation("softmax"))
```

Lab 1: here in 15min!