Panorama of Neural Networks and Deep Learning

16-11-2018

Patrick Gallinari
Sorbonne Université – Paris 6, France
patrick.gallinari@lip6.fr
https://mlia.lip6.fr/

SORBONNE UNIVERSITÉ

MLIA Machine Learning & Deep Learning for Information Access

MLIA Machine Learning & Deep Learning for Information Access

https://mlia.lip6.fr/

- MLIA is the Machine Learning team at the computer science lab of Sorbonne
- Main research topics
  - Machine learning
    - Representation learning and Deep Learning
      - Transversal activity, models and algorithms, several application domains
    - Structured data
      - e.g. Xtreme classification, sequences, graphs, spatio-temporal data , …
  - Application domains
    - Computer Vision
      - Classification, detection, segmentation, Visual QA, …
    - Natural Language Processing and Information Retrieval
      - Information extraction, interactive IR, language grounding, language generation
    - Complex data analysis
      - Social data, mobility data, interaction traces, recommendation, etc
    - Data models for climate

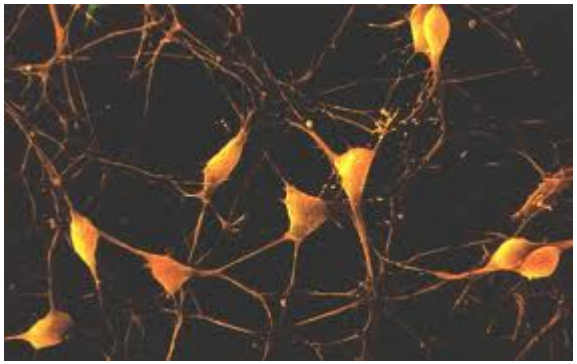
Outline

- Panorama of the evolution of the domain
  - 1960 - Early Days –Fundamendal concepts of Machine Learning
  - 1990 - Non Linear Machines – Statistical Learning Theory
  - 2010 - Deep Learning – Large Size Industrial Applications
    - NN bricks
      - Convolutional Neural Networks
      - Recurrent Neural Networks
    - Unsupervised learning
      - Generative models
- Some examples from MLIA

1960 – Early days -

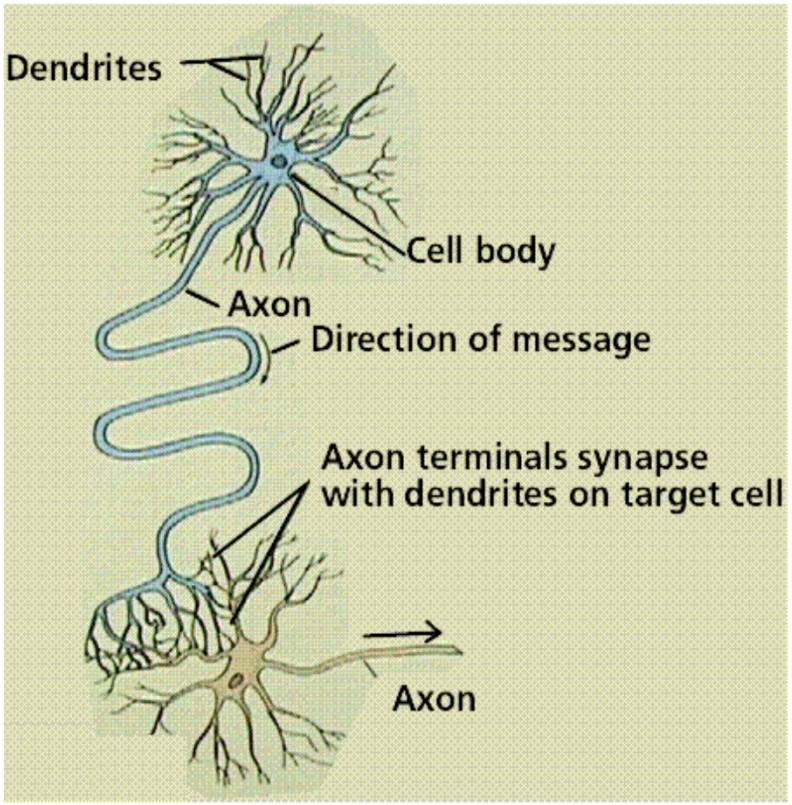
Fundamendal concepts of Machine Learning

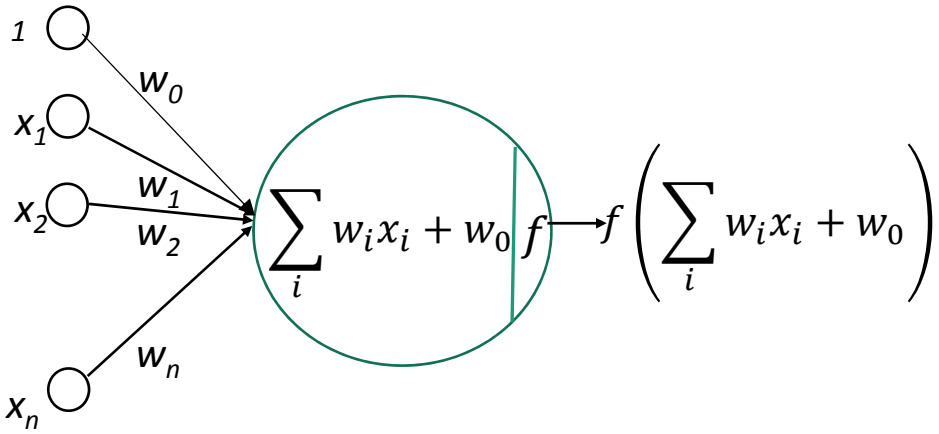
Neural Networks inspired Machine Learning



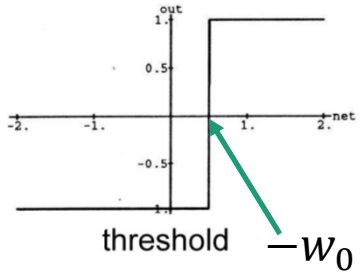
- Artificial Network Networks are an important paradigm in Statistical Machine learning and Artificial Intelligence
- Human brain is used as a source of inspiration and as a **metaphor** for developing Artificial NN
  - Human brain is a dense network 10^{11} of simple computing units, the neurons. Each neuron is connected – in mean- to 10^{4} neurons.
  - Brain as a computation model
    - Distributed computations by simple processing units
    - Information and control are distributed
    - Learning is performed by observing/ analyzing huge quantities of data and also by trials and errors

Formal Model of the Neuron
McCulloch – Pitts 1943





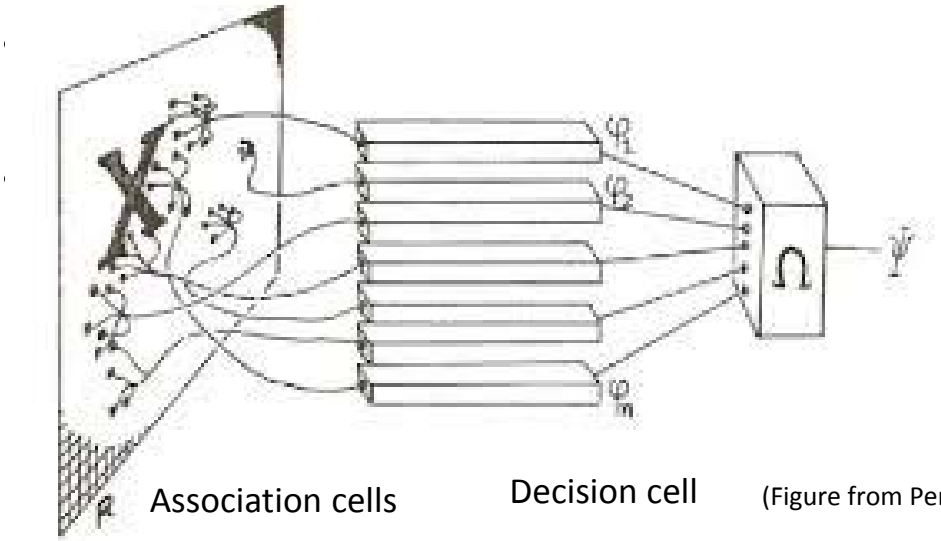
For McCulloch – Pitts neuron,
f is a threshold (sign) function



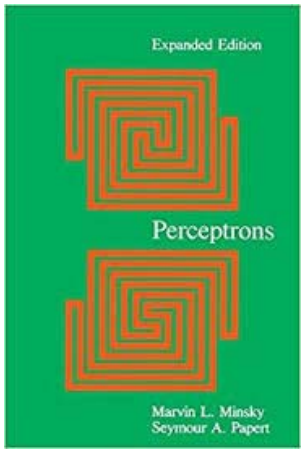
$$y = \begin{cases} 1 \; if \sum_i (w_i x_i + w_0) > 0 \\ -1 \; otherwise \end{cases}$$

A synchronous assembly of neurons is capable of universal computations (aka equivalent to a Turing machine)

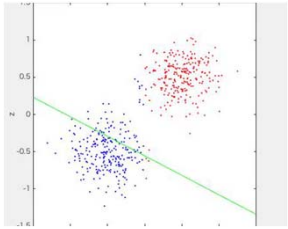
Perceptron: inspiration from perception (1958 Rosenblatt



Association cells Decision cell (Figure from Perceptrons, Minsky and Papert 1969)



- The decision cell is a threshold function (McCulloch – Pitts neuron)
  - $F(\boldsymbol{x}) = sgn(\sum_{i=1}^{n} w_i x_i + \mathrm{w}_0)$
- This 1 neuron-perceptron can perform 2 classes classification
- Training: **stochastic** (gradient) algorithm for minimizing classification error
  - Sample an example
  - If badly classified, update the neuron weights

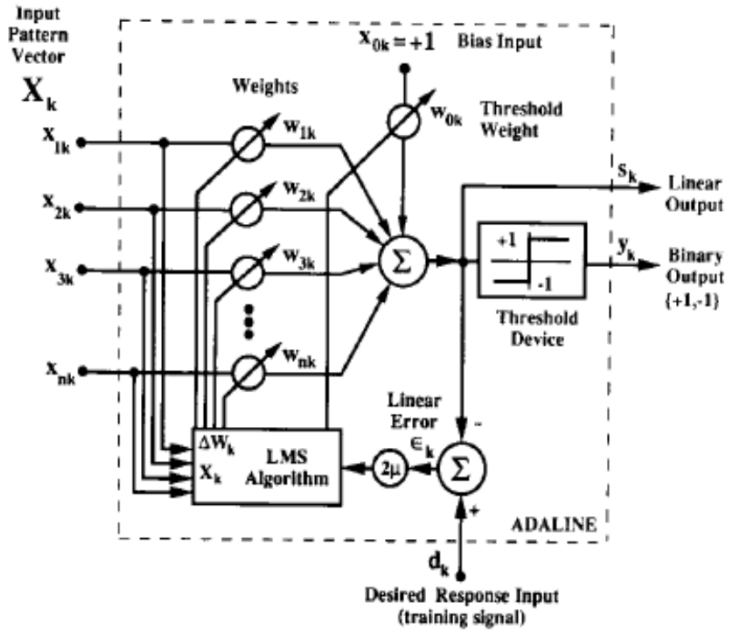


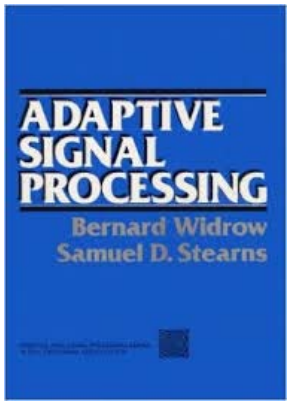
Perceptron: properties - convergence and generalization bounds

- **Convergence** theorem (Novikof, 1962)
  - Let $\mathrm{D} = \{(\boldsymbol{x}^1, y^1), \ldots, (\boldsymbol{x}^N, y^N)\}$ a data sample. If
    - $R = \max_{1 \le i \le N} \|\boldsymbol{x}^{\mathrm{i}}\|$
    - $\sup_{\boldsymbol{w}} \min_{i} d^i(\boldsymbol{w}.\boldsymbol{x}^i) > \rho$
    - The training sequence is presented a sufficient number of time
  - The algorithm will converge after at most $\left[\frac{R^2}{\rho^2}\right]$ corrections
- **Generalization** bound (Aizerman, 1964)
  - If in addition we provide the following stopping rule:
    - Perceptron stops if after correction number k, the next $m_k = \frac{1+2\ln k - \ln \eta}{-\ln(1-\epsilon)}$ data are correctly recognized
  - Then
    - the perceptron will converge in at most $\mathrm{l} \le \frac{1+4\ln R/\rho - \ln \eta}{-\ln(1-\epsilon)} [R^2/\rho^2]$ steps
    - with probability $1-\eta$, test error is less than ϵ

    Link between training and generalization performance

Adaline (Widrow - Hoff 1959)





- Conte
  - Adaptive filtering, equalization, etc.
- « Least Mean Square » LMS algorithm
  - Loss function : euclidean distance: $\|target - computed\ output\|^2$
  - Algorithm: **stochastic gradient** (Robbins – Monro (1951))
- Workhorse algorithm of adaptive signal processing
  - Simple, robust

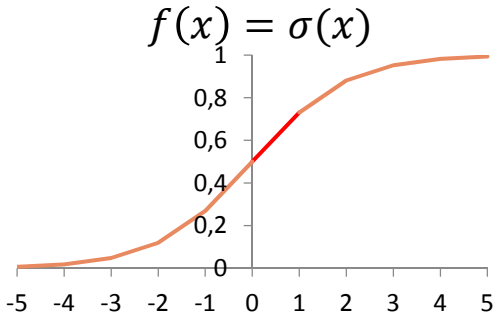
Widrow-Science in Action - YouTub

Summary

- Many of the main concepts of statistical Machine Learning are already present in the early days
  - Learning machine as alternative model of computations
    - Inspired by animal perception
  - Stochastic algorithms for optimizing loss functions
    - Stochastic Gradient Descent (**SGD**)
  - Target applications
    - Pattern recognition (speech, image, etc), control, signal processing, games, broom balancing …
  - A few performance guaranties assessed by generalization bounds

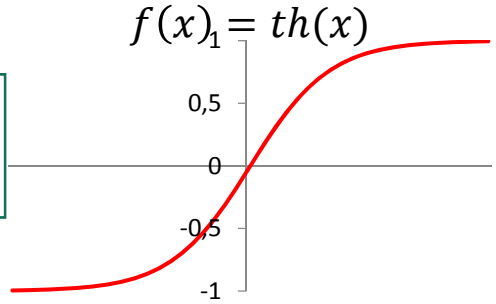
1990 - Non Linear Machines and Statistical Learning Theory

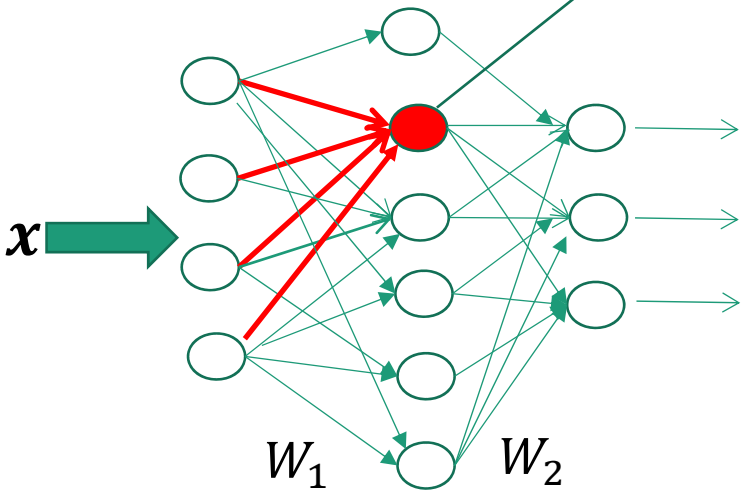
Multi-layer Perceptron (Hinton – Sejnowski – Williams 1986)



- Neurons arranged into layers
- Each neuron is a non linear unit, e.g.

$f(\boldsymbol{w}.\boldsymbol{x})$
$\boldsymbol{w}$: **cell weight vector**

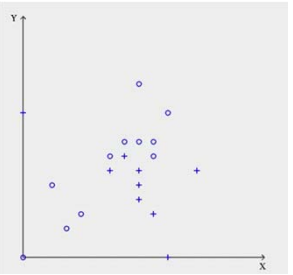




$$\hat{\boldsymbol{y}} = F_w(\boldsymbol{x}) = f\odot(W_2 f\odot(W_1\boldsymbol{x}))$$

http://playground.tensorflow.org/

Note: $\odot$ *is a pointwise operator* $f\odot(x_1, x_2) = (f(x_1), f(x_2))$



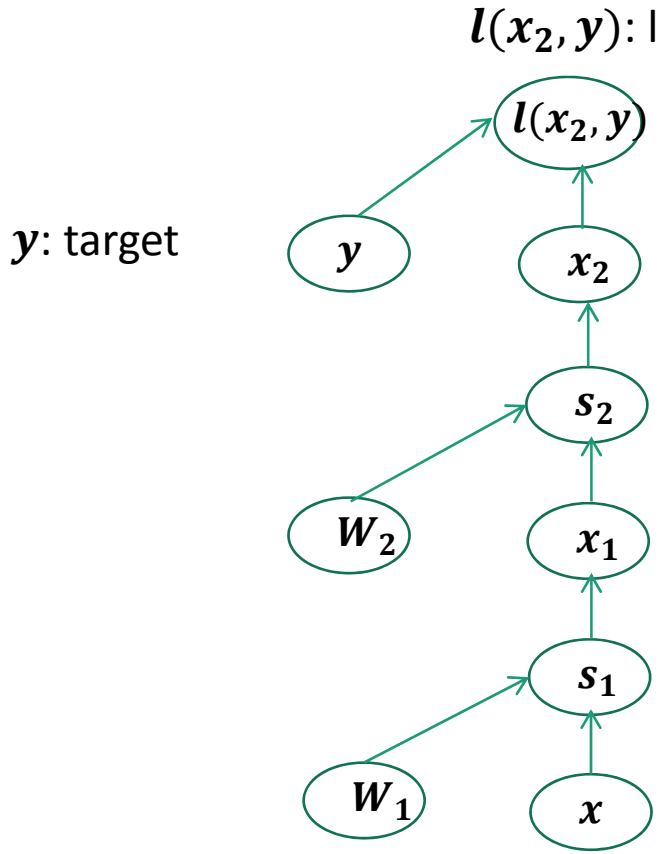
Algorithmic differentiation

- **Training algorithm**
  - Stochastic Gradient Descent
  - Same as Widdrow-Hoff –LMS- rule
  - The MLP implementation is called **Back-Propagation**

- Back-Propagation is an instance of **automatic differentiation / algorithmic differentiation - AD**
  - A mathematical expression can be written as a **computation graph**
    - i.e. graph decomposition of the expression into elementary computations
  - **AD** allows to **compute** efficiently the derivatives of every element in the graph w.r.t. any other element.
  - **AD** transform a programs computing a numerical funtion into the program for computing the derivatives

Algorithmic differentiation
Multi-layer Perceptron Training

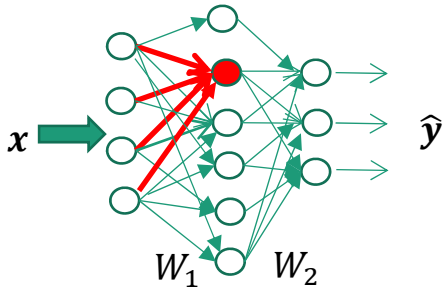
- Computation graph



Forward propagation:

$s_n = W_n x_{n-1}$

$x_n = \sigma(s_n)$

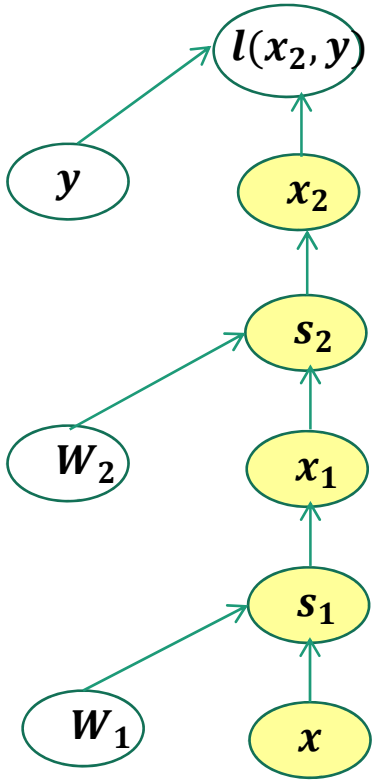


Algorithmic differentiation
Multi-layer Perceptron - Training

- Forward pass

$\boldsymbol{l}(\boldsymbol{x_2}, \boldsymbol{y})$: loss

$\boldsymbol{y}$: target



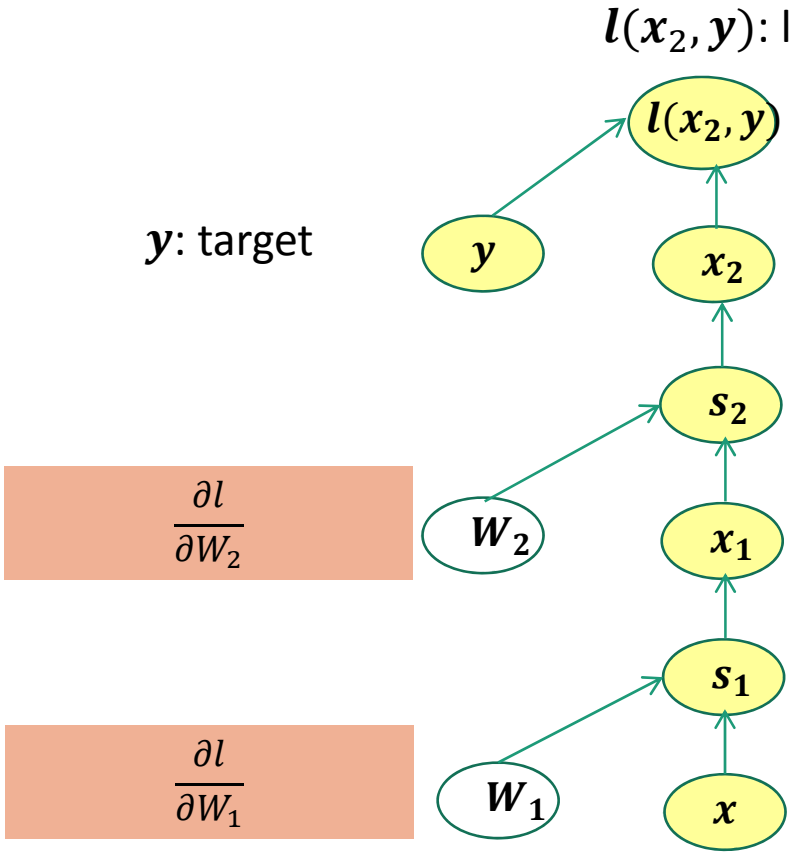
Forward propagation:

$\boldsymbol{s}_n = W_n \boldsymbol{x}_{n-1}$

$\boldsymbol{x}_n = \sigma(\boldsymbol{s}_n)$

Algorithmic differentiation
Multi-layer Perceptron - Training

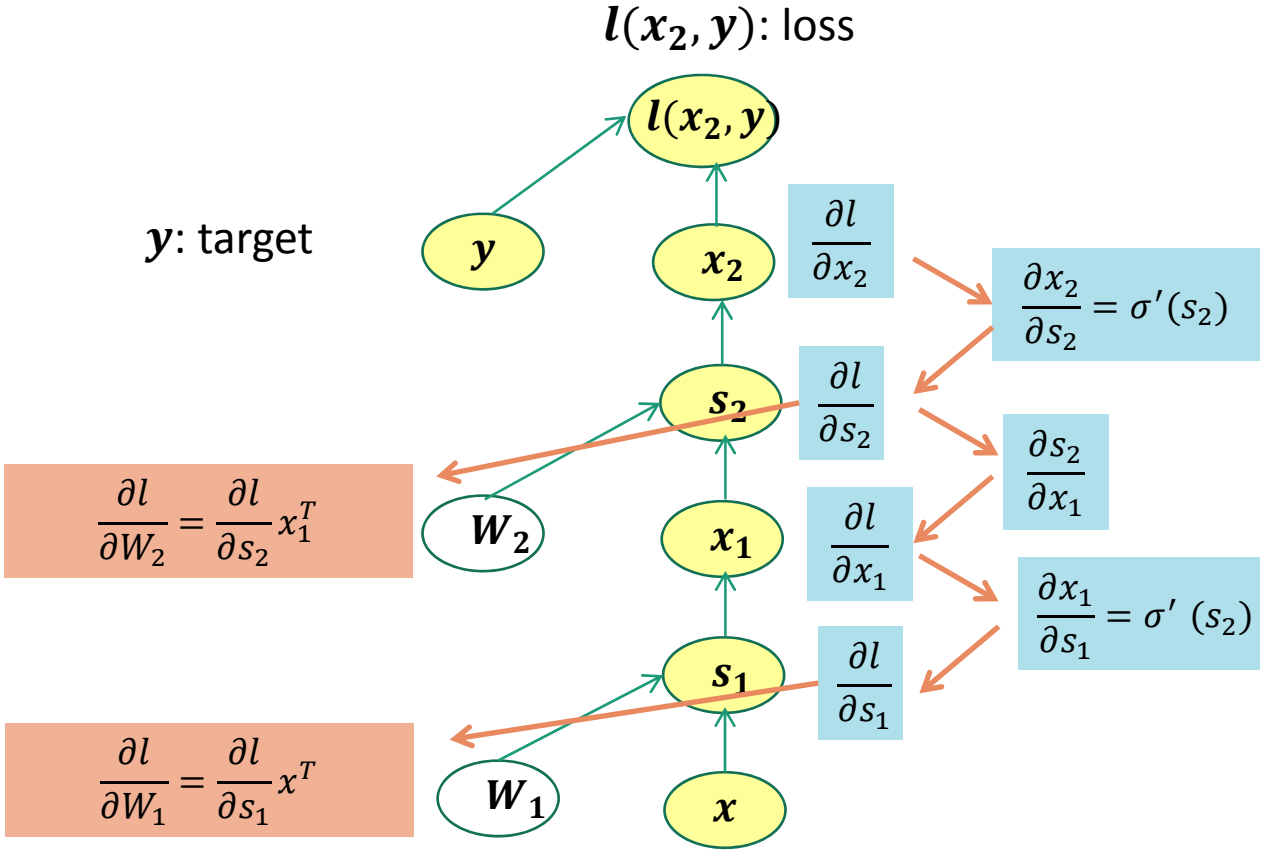
- Back Propagation: Reverse Mode Differentiation



$$W = W - \epsilon \frac{\partial l}{\partial W}$$

Algorithmic differentiation
Multi-layer Perceptron - Training

- Back propagation: Reverse Mode Differentiation



Backward propagation:

$$\frac{\partial l}{\partial s_n} = \frac{\partial l}{\partial x_n} \odot \sigma'(s_n)$$

$$\frac{\partial l}{\partial W_n} = \frac{\partial l}{\partial s_n} x_{n-1}^T$$

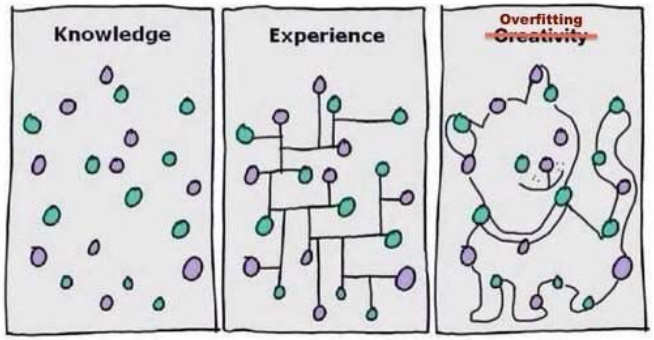
$$\frac{\partial l}{\partial x_{n-1}} = W_n^T \frac{\partial l}{\partial s_n}$$

Multi-layer Perceptron – Approximation Properties

- Universal Approximation
  - e.g. Cybenko 89: Let f be a continuous saturating function. The space of functions of the form $g(x) = \sum_{j=1}^{n} v_j f(\mathbf{w}_j . . \mathbf{x})$ is dense in the space of continuous functions on the unit cube $C(I)$. i.e. $\forall h \in C(I) et\ \forall \epsilon > 0, \exists\ g :$ $|g(\boldsymbol{x}) - h(\boldsymbol{x})| < \epsilon$ on *I*.
- Not a « constructive » result
  - e.g. number of hidden neurons or hidden layers for a given problem?

Generalization and Model Selection

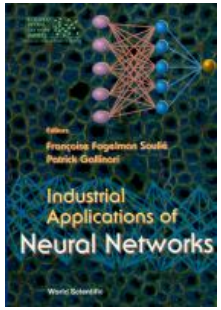
- Complex models sometimes perform worse than simple linear models
  - Overfitting/ generalization problem



- Empirical Risk Minimization is not sufficient
  - The model complexity should be adjusted both to the task and to the information brought by the examples
  - Both the model parameters and the model capacity should be learned
  - Lots of practical method and of theory has been devoted to this problem: regularization, ensemble methods, …., Vapnik ERM/SRM, PAC framework, ...

Summary

- Non linear machines
- Fundations for modern statistical machine learning
- Fundations for statistical learning theory
- Real world applications



- Also during this period
  - Convolutional Neural Networks
  - Recurrent Neural Networks
    - Extension of back propagation
  - Reinforcement Learning
    - Early work mid 80ies
    - Sutton – Barto Book 1998, including RL + NN

2010 Deep Learning

Interlude

Convolutional Neural Networks

Recurrent Neural Networks

Unsupervised learning with generative models

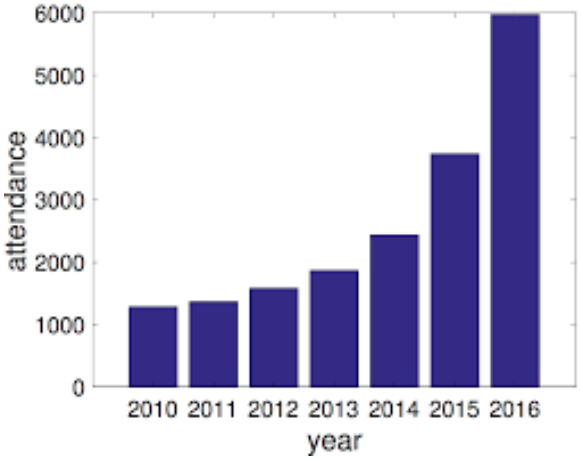
Interlude: new actors – new practices

- GAFA (Google, Apple, Facebook, Amazon) , BAT (Baidu, Tencent, Alibaba), …, Startups, are shaping the data world
- Research
  - Big Tech. actors are leading the research in DL
  - Large research groups
    - Google Brain, Google Deep Mind, Facebook FAIR, Baidu AI lab, Baidu Institute of Deep Learning, etc
  - Standard development platforms, dedicated hardware, etc
  - DL research requires access to ressources
    - sophisticated libraries
    - large computing power e.g. GPU clusters
    - large datasets, …

Facebook AI Research

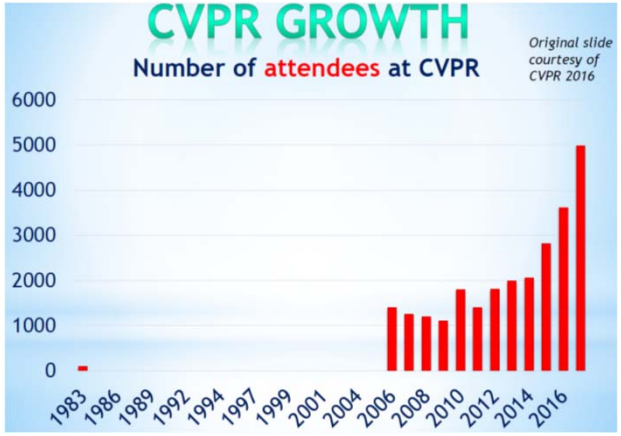
Interlude – ML conference attendance growth

- Nips attendance (T. Sejnowski)



- 2017 sold out 1 week after registration opening, 7000 participants
- 2018, 2k inscriptions sold in 11 mn!

- CVPR attendance



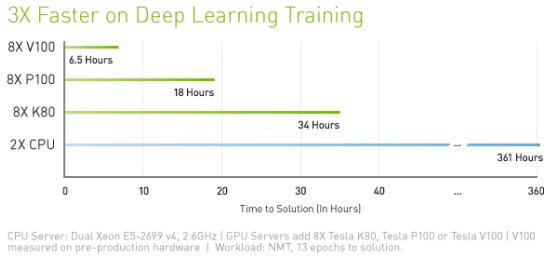
Interlude – Deep Learning platforms

- Deep Learning platforms offer
  - Classical DL models
  - Optimization algorithms
  - Automatic differentiation
  - Popular options/ tricks
  - Pretrained models
  - CUDA/ GPU/ CLOUD support
- Contributions by large open source communities: lot of code available
- Easy to build/ train sophisticated models

- Among the most populars platforms:
  - TensorFlow - Google Brain - Python, C/C++
  - PyTorch – Facebook- Python
  - Caffe – UC Berkeley / Caffe2 Facebook, Python, MATLAB
  - Higher level interfaces
    - e.g. Keras for TensorFlow
- And also:
  - PaddlePaddle (Baidu), MXNet (Amazon), Mariana (Tencent), PAI 2.0 (Alibaba), …..

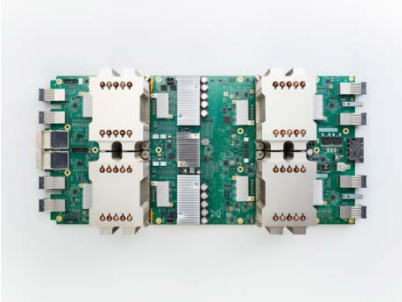
Interlude – Hardware

- 2017 - NVIDIA V100 – optimized for Deep Learning



- "With 640 Tensor Cores, Tesla V100 is the world's first GPU to break the 100 teraflops (TFLOPS) barrier of deep learning performance. The next generation of NVIDIA NVLink™ connects multiple V100 GPUs at up to 300 GB/s to create the world's most powerful computing servers."

- 2017 - Google Tensor Processor Unit



- Cloud TPU 3

Deep Learning Bricks

Convolutional Neural Networks

Recurrent Neural Networks

Convolutional nets

- ConvNet architecture (Y. Le Cun since 1988) – (inspired from Huber-Wiesel model of visual cortex – 1962 and Fukushima -Neocognitron 1980)
  - Deployed e.g. (Bell Labs -> NCR) in 1989-90 for zip code recognition
  - Character segmentation and recognition
  - Convolution with learned filters: non linear embedding in high dimension
  - Pooling: average, max

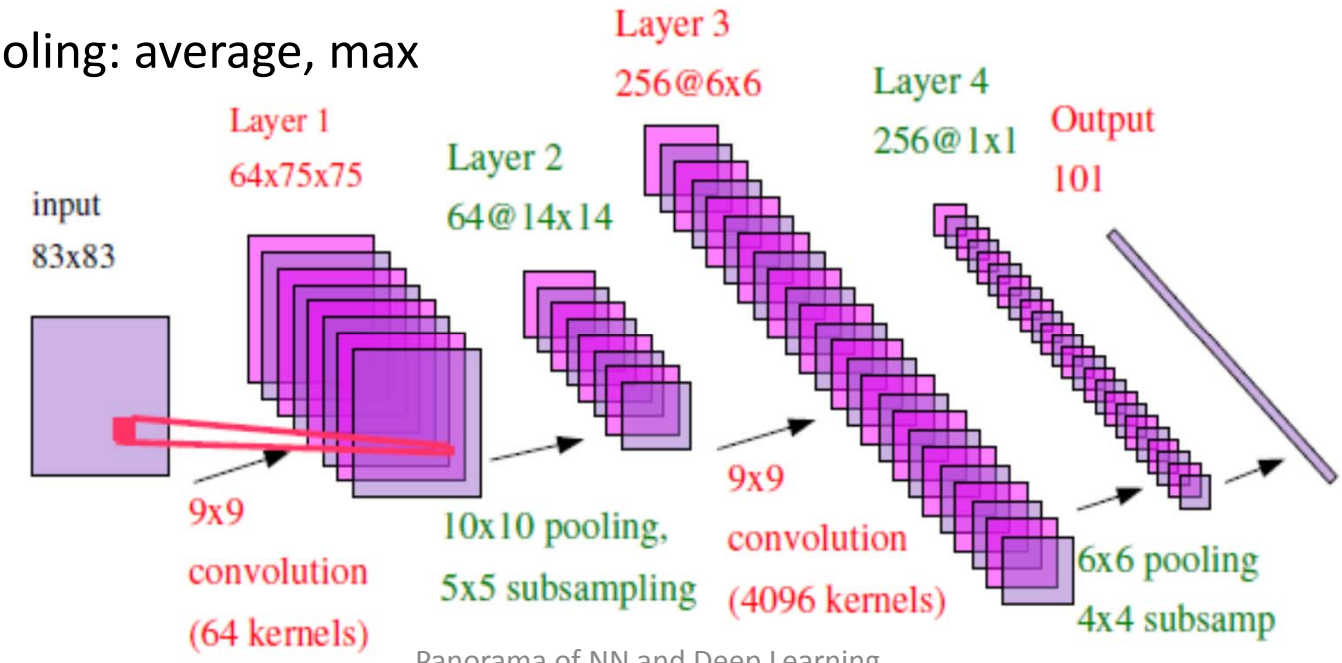
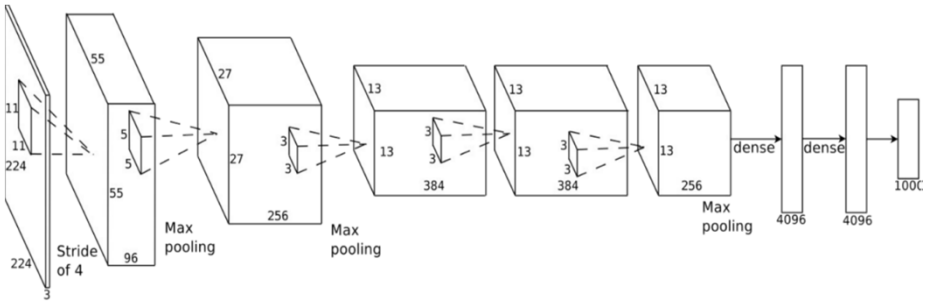
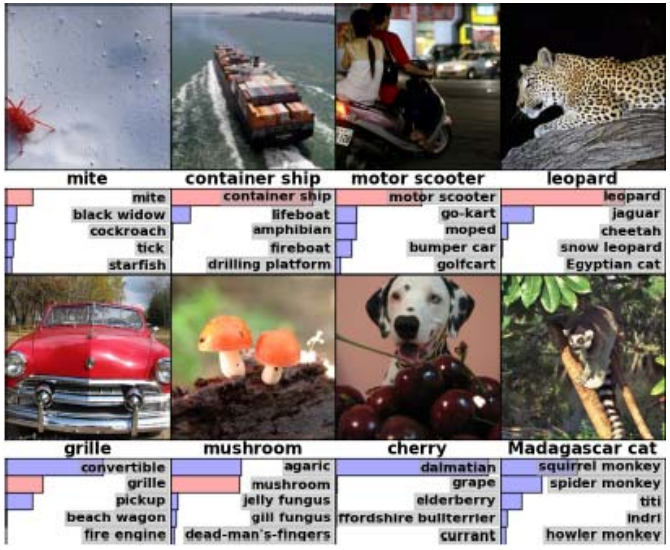


Fig. from Le Cun

Convolutional nets (Krizhevsky et al. 2012)

- A landmark in object recognition - AlexNet
- ImageNet competition
  - Large Scale Visual Recognition Challenge (ILSVRC)
  - 1000 categories, 1.5 Million labeled training samples
  - Method: large convolutional net
  - 650K neurons, 630M synapses, 60M parameters
  - Trained with SGD on GPU



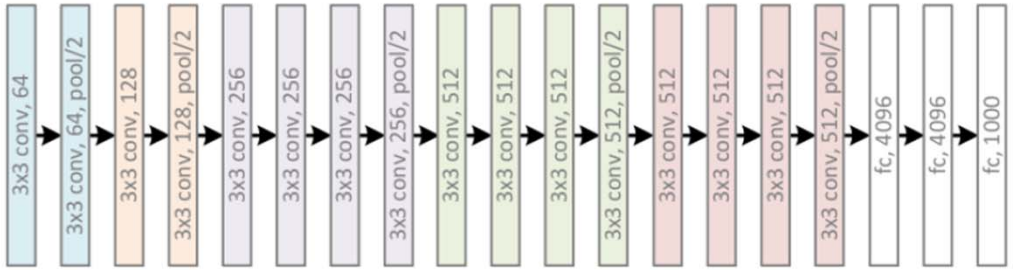


Very Deep Nets trained with GPUs

Deeper Nets with small filters – training time several days up to 1 or 2 weeks on ImageNet

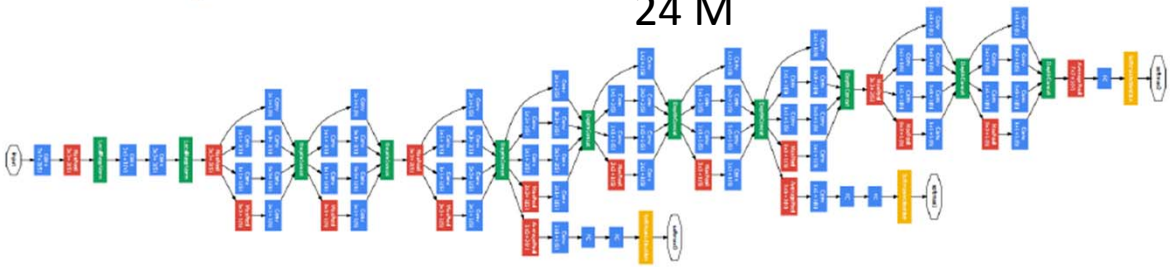
VGG, 16/19 layers, 2014

Oxford, [Simonyan 2014], Parameters 138 M



GoogleNet, 22 layers, 2014

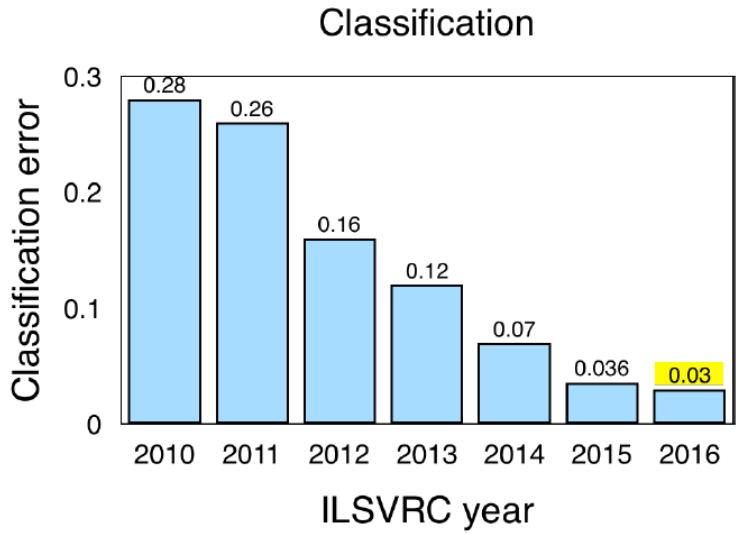
Google, [Szegedy et al. 2015], Parameters 24 M

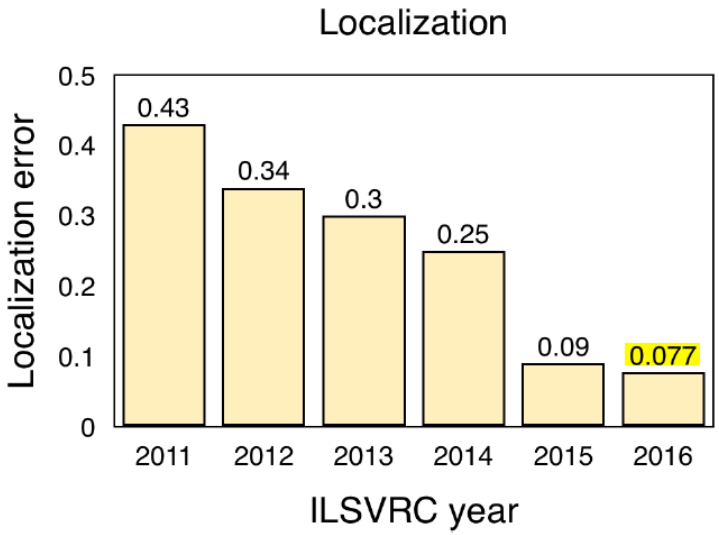


ResNet, 152 layers, 2015

MSRA, [He et al. 2016] , Parameters 60 M

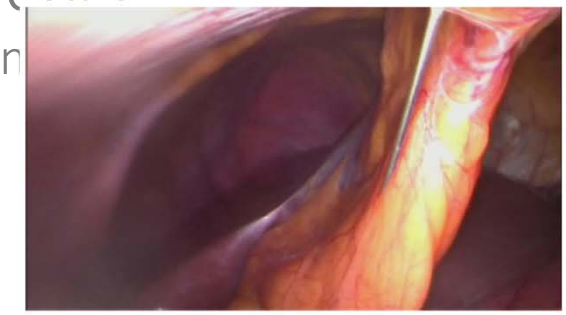
Convolutional Nets
ILSVRC performance over the years





CNNs – Transfer learning -Images from different nature,M2CAI Challenge

(R. Cadene 2016 @ Sorbonn



- Endoscopic videos (large intestine)
  - resolution of 1920 x 1080, shot at 25 frame per second at the IRCAD research center in Strasbourg, France. 27 training videos ranging from 15mn to 1hour, 15 testing videos
- Used for: monitor surgeons, Trigger automatic actions
- Objective: classification, 1 of 8 classes for each frame
  - TrocarPlacement, Preparation, CalotTriangleDissection, ClippingCutting, GallbladderDissection, GallbladderPackaging, CleaningCoagulation, GallbladderRetraction
- Resnet 200 pretrained with ImageNet -> reaches 80% correct classification

Model	Input	Param.	Depth	Implem.	Forward (ms)	Backward (ms)
Vgg16	224	138M	16	GPU	185.29	437.89
InceptionV3 [2]	399	24M	42	GPU	**102.21**	311.94
ResNet-200 [3]	224	65M	200	GPU	273.85	687.48
InceptionV3	399	24M	42	CPU	19918.82	23010.15

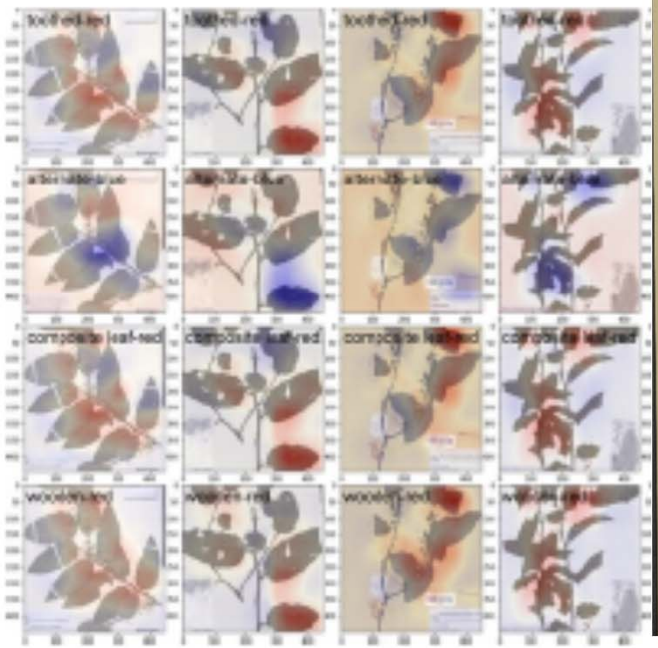
Table 1: Forward+Backward with batches of 20 images.

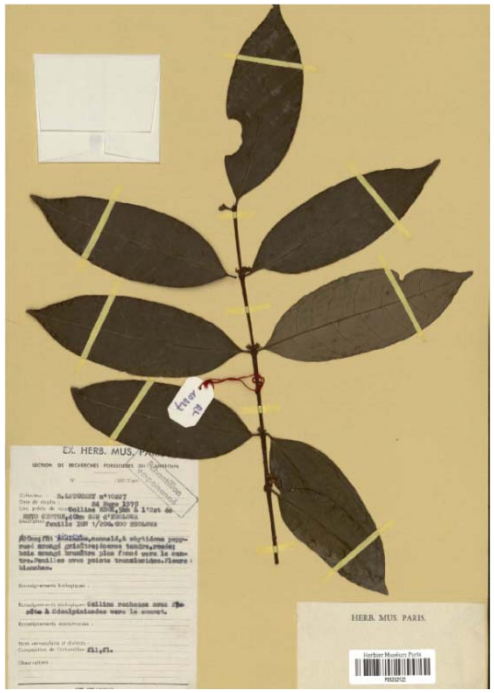
InceptionV3	Extraction (repres. of ImageNet)	60.53
InceptionV3	From Scratch (repres. of M2CAI)	69.13
InceptionV3	Fine-tuning (both representations)	79.06
ResNet200	**Fine-tuning (both representations)**	**79.24**

Table 2: Accuracy on the validation set.

CNNs – Transfer learning - Images from different nature, Plant classification (Y. Zhu- 2017 @ Sorbonne)

- Digitized plant collection from Museum of Natural History – Paris
- Largest digitized world collection (8 millions specimens)
- Goal
  - Identify plants characteristics for automatic labeling of worlwide plant collections
  - O(1000) classes, e.g. opposed/alternate leaves; simple/composed leaves; smooth/with teeth leaves, ….
- Pretrained ResNet





CNNs for Object detection
Case study: YOLO (Redmon 2015) - Design and Training

- Pretrained on ImageNet 1000 class
- Remove classification layer and replace it with 4 convolutional layers + 2 Fully Connected layers
- Activations: Linear for the last layer, leaky reLu for the others
- Requires a lot of know-how (design, training strategy, tricks, etc)
  - Not described here – see paper…
- Improved versions followed the initial paper
- Generalizes to othe

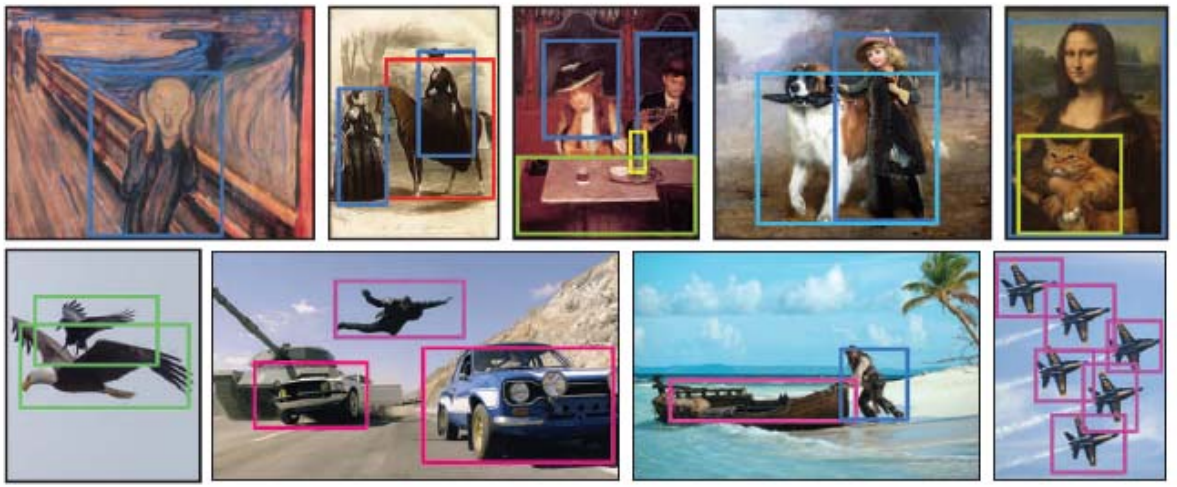
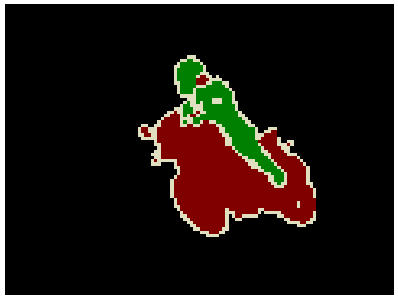


Figure 6: Qualitative Results. YOLO running on artwork and natural images. It is mostly accurate although it does think one person in an image is an airplane.

CNNs for Image Semantic Segmentation

- Objective
  - Identify the different objects in an image



- Deep learning
  - handles segmentation as pixel classification
  - re-uses network trained for image classification by making them fully convolutional
  - Currently, SOTA is Deep Learning
- Main datasets
  - Voc2012, http://host.robots.ox.ac.uk/pascal/VOC/voc2012/
  - MSCOCO, http://mscoco.org/explore/

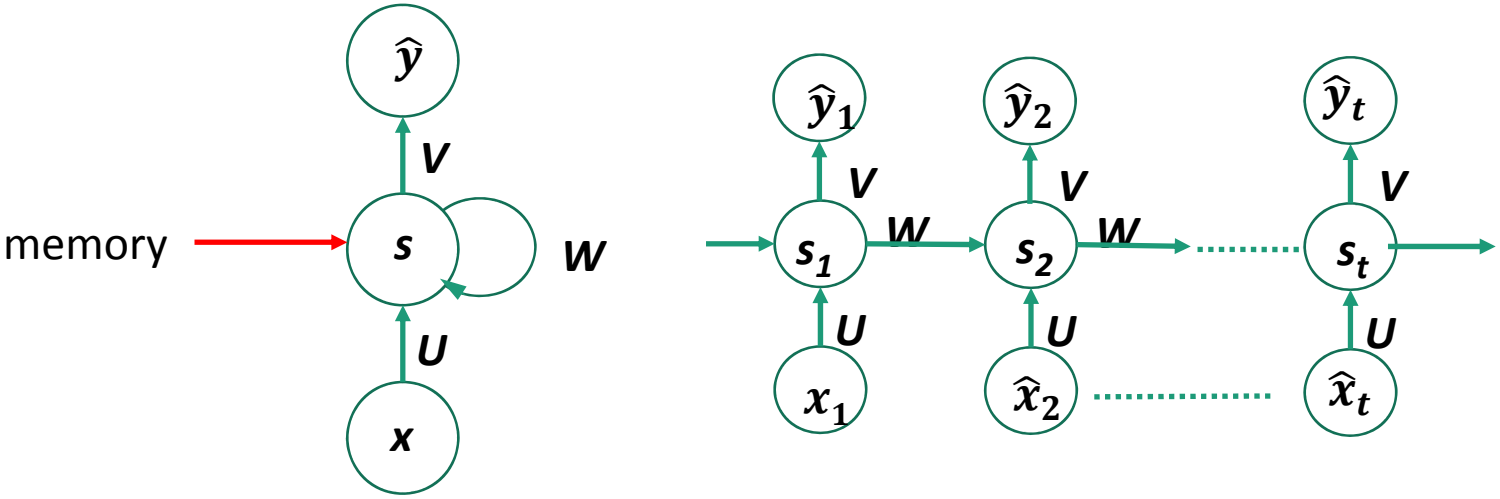
Deep Learning Bricks

Convolutional Neural Networks

Recurrent Neural Networks

Recurrent neural networks - RNNs

- Basic architecture: state space model



- Up to the 90s RNN were of no practical use, too difficult to train
- Mid 2000s successful attempts to implement RNN
  - e.g. A. Graves for speech and handwriting recognition
- Today
  - RNNs SOTA for a variety of applications e.g., speech decoding, translation, language generation, etc – today alternatives based on attention models

Recurrent neural networks
Language models

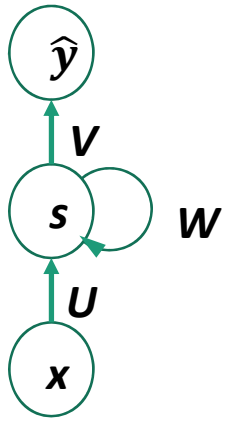
- Objective:
  - Probability models of sequences $(x^1, x^2, \dots, x^t)$
  - Items may be words or characters
  - Estimate:
    - $p(x^t | x^{t-1}, \dots, x^1)$

prediction

memory

$\widehat{y^t} = g(Vs^t)$

$s^t = f(Ws^{t-1} + Ux^t)$



- Example
  - « S'il vous plaît… dessine-moi …» what next?
  - « $x^1 x^2 x^3$ … … … … … . … . . x^{t-1} … » what is x^t?

Language models – example: text generation

(Karpathy 2015- https://karpathy.github.io/2015/05/21/rnneffectiveness/)

- Training on Tolstoy's War and Peace a character language model
  - Stacked recurrent networks (LSTM)

```
tyntd-iafhatawiaoihrdemot  lytdws  e ,tfti, astai f ogoh eoase rrranbyne 'nhthnee e
plia tklrgd t o idoe ns,smtt   h ne etie h,hregtrs nigtike,aoaenns lng
```

train more

```
"Tmont thithey" fomesscerliund
Keushey. Thom here
sheulke, anmerenith ol sivh I lalterthend Bleipile shuwy fil on aseterlome
coaniogennc Phe lism thond hon at. MeiDimorotion in ther thize."
```

train more

```
Aftair fall unsuch that the hall for Prince Velzonski's that me of
her hearly, and behs to so arwage fiving were to it beloge, pavu say falling misfort
how, and Gogition is so overelical and ofter.
```

train more

```
"Why do what that day," replied Natasha, and wishing to himself the fact the
princess, Princess Mary was easier, fed in had oftened him.
Pierre aking his soul came to the packs and drove up his father-in-law women.
```

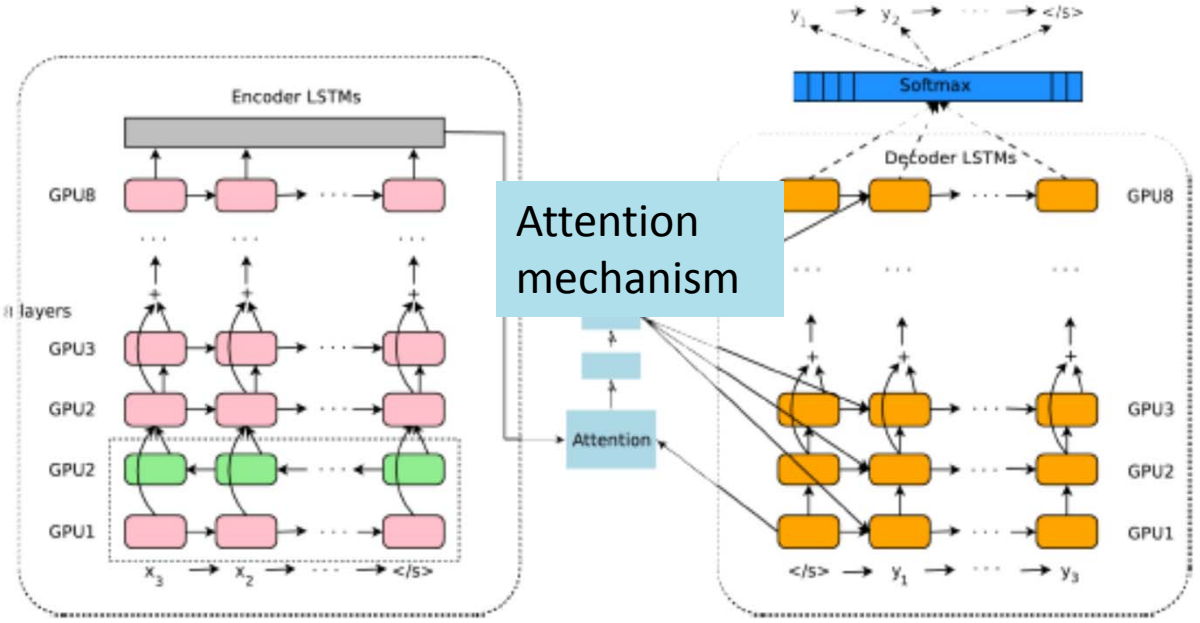
Google Neural Machine Translation System

(Wu et al 2016)
https://research.googleblog.com/2016/09/a-neural-network-for-machine.html

- General Architecture

Encoder: 8 stacked LSTM RNN + residual connections

Decoder: 8 stacked LSTM RNN + residual connections + Softmax output layer



Attention mechanism

Figure from Wu et al. 2016

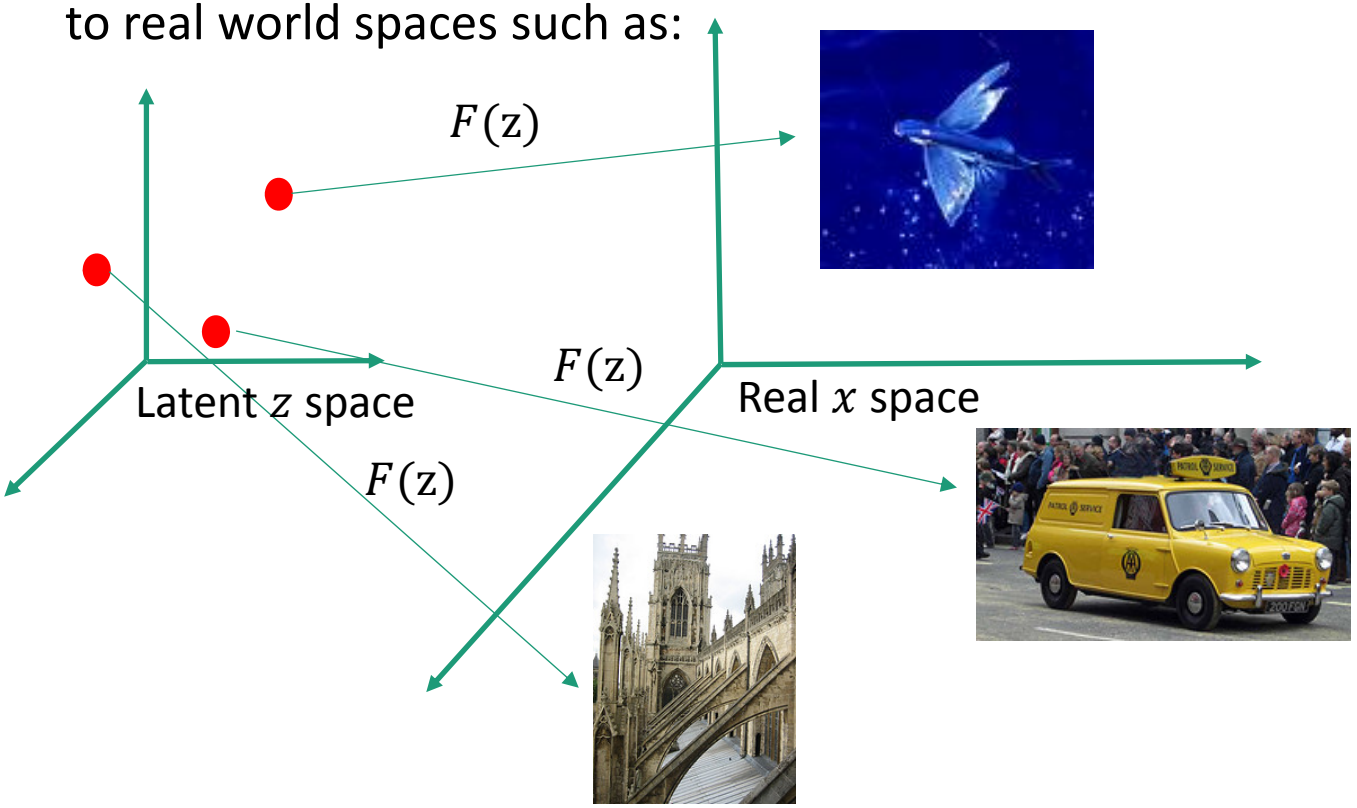
- NMT seminal papers: Cho et al. 2014, Sutskever et al. 2014
- Comparison and evaluation of NMT RNNs options (Fritz et al. 2017)
  - 250 k-hours GPU -> a 250 k$ paper !

Unsupervised learning

- Example: Generative Adversarial Networks – GANs (Goodfellows 2014)
  - 1750 GAN papers on Arxiv at 2018-11-15

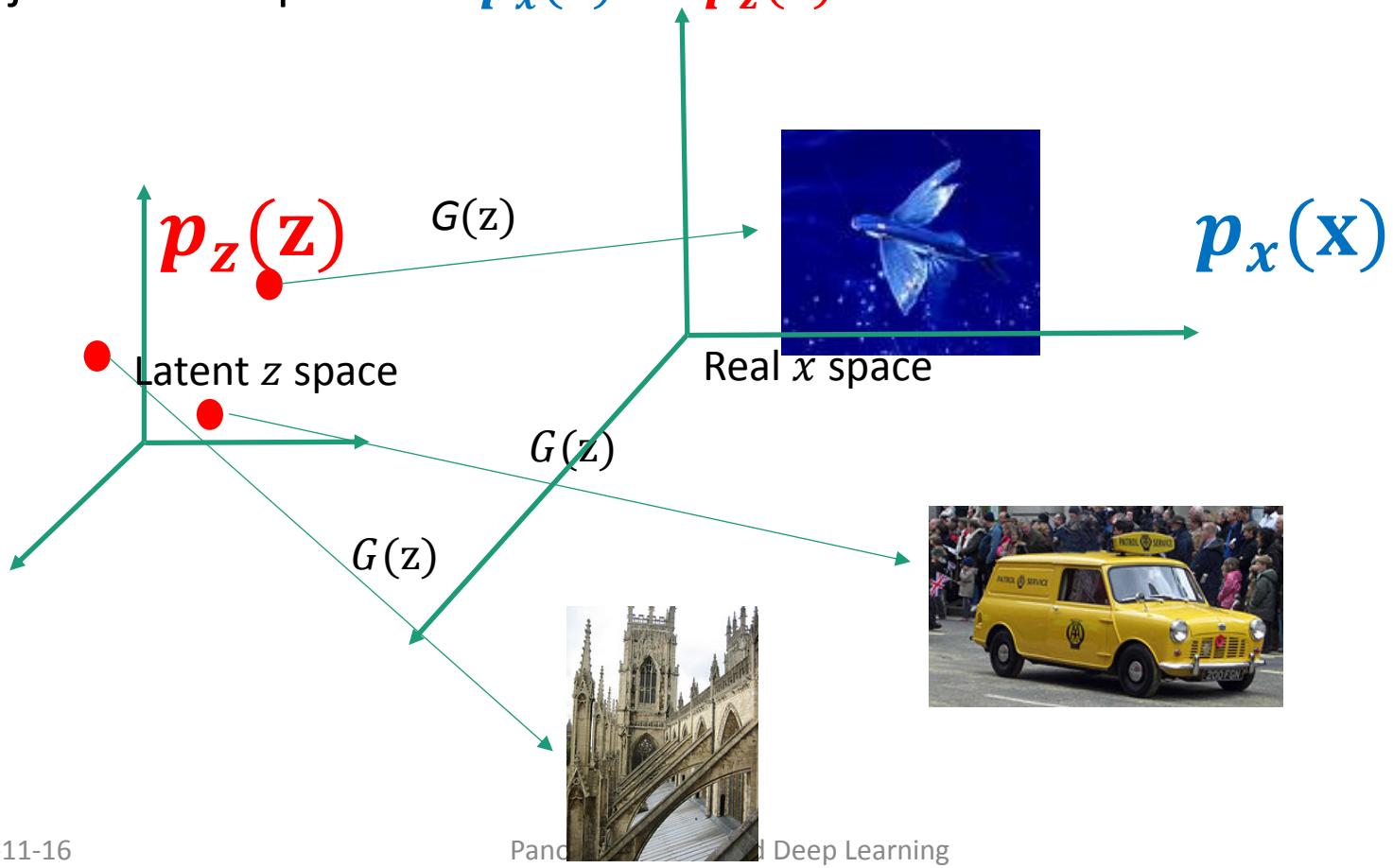
Generative models intuition

- Provided a sufficiently powerful model $F(\mathrm{z})$
  - It should be possible to learn complex mappings from latent space to real world spaces such as:



Generative models intuition

- Given a probability distribution on the latent space $p_z(\mathrm{z})$, G defines a probability distribution on the observation space
- Objective: sample from $\boldsymbol{p_x(\mathbf{x})}$ via $\boldsymbol{p_z(\mathbf{z})}$ and G



GANs examples Deep Convolutional GANs (Radford 2015) - Image generation

- LSUN bedrooms dataset - over 3 million training examples

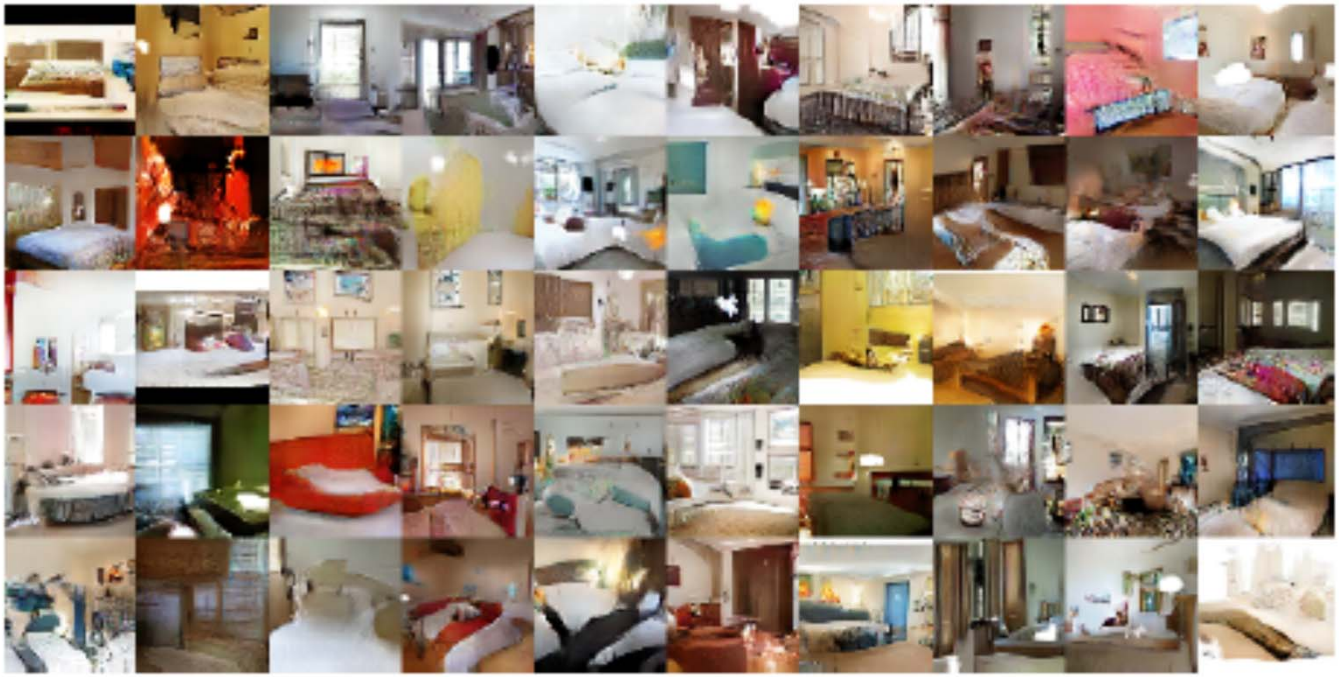


Figure 3: Generated bedrooms after five epochs of training. There appears to be evidence of visual under-fitting via repeated noise textures across multiple samples such as the base boards of some of the beds.

Fig. Radford 2015

Conditional GANs example
Generating images from text (Reed 2016)

- Objective
  - Generate images from text caption
  - Model: GAN conditionned on text input
- Compare different GAN variants on image generation
- Image size 64x64

Fig. from Reed 2016

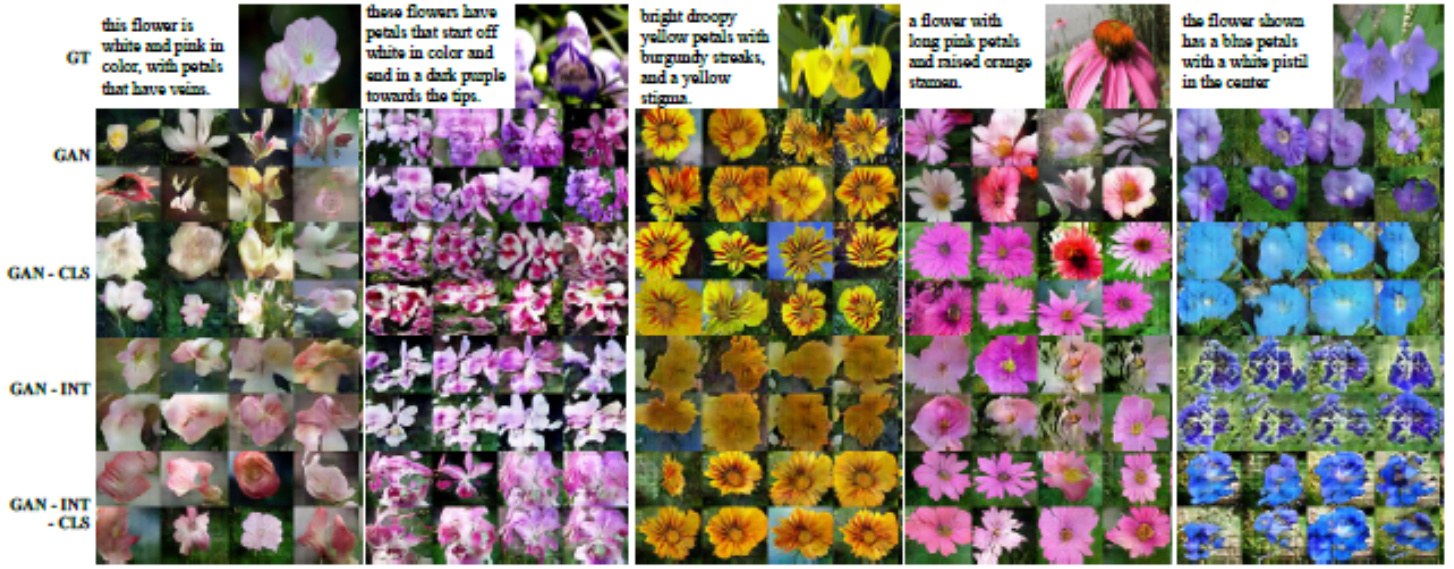
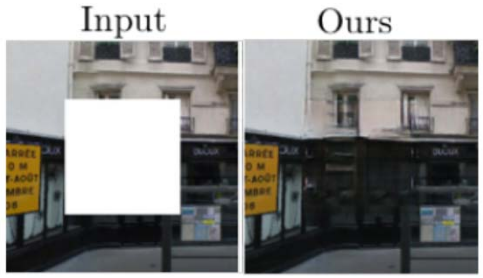


Figure 4. Zero-shot generated flower images using GAN, GAN-CLS, GAN-INT and GAN-INT-CLS. All variants generated plausible images. Although some shapes of test categories were not seen during training (e.g. columns 3 and 4), the color information is preserved.

Conditional GANs example – Pix2Pix
Image translation with cGANs (Isola 2016)

- Objective
  - Learn to « translate » images for a variety of tasks using a common framework
    - i.e. no task specific loss, but only adversarial training + conditioning
  - Tasks: semantic labels -> photos, edges -> photos, (inpainting) photo and missing pixels -> photos, etc



Cycle GANs (Zhu 2017)

- Objective
  - Learn to « translate » images without aligned corpora
    - 2 corpora available with input and output samples, but no pair alignment between images
- Examples

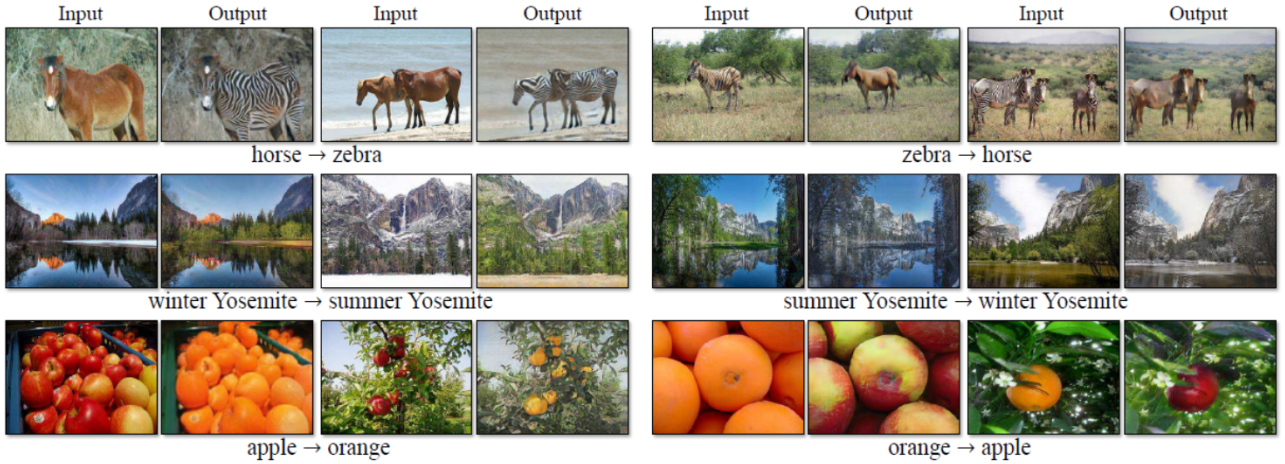


Figure 7: Results on several translation problems. These images are relatively successful results – please see our website for more comprehensive results.

Input Monet Van Gogh Cezanne Ukiyo-e

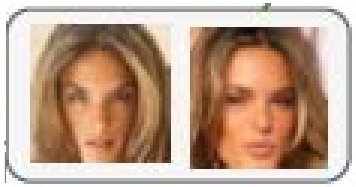
Fig (Zhu 2017)

Summary

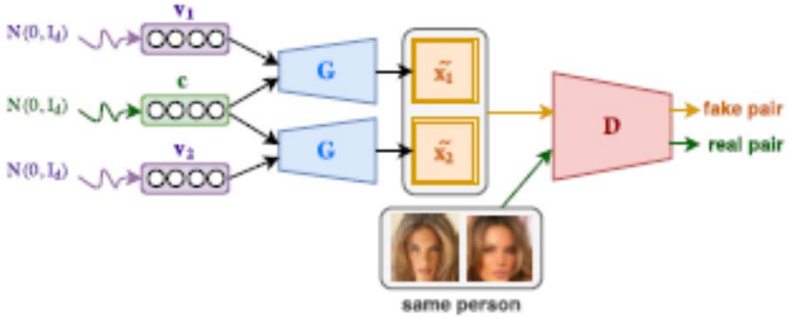
- Unprecedented developments in ML in general
  - Conjonction of several factors
    - Data deluge, Computing power, Free software ML libraries by major IT actors
    - Big players and fast « prototype to industrial deployment »
- NNs are today at the heart of this development
  - Powerful models
  - Modularity allows to build complex systems, trainable end to end
    - Gradient everywhere
  - State of the art in many domains
  - Research driven by big IT companies!
  - Theory still to be developed!

Some examples from MLIA@Sorbonne

Gan
Multi-View data generation without view supervis
(Chen 2018)

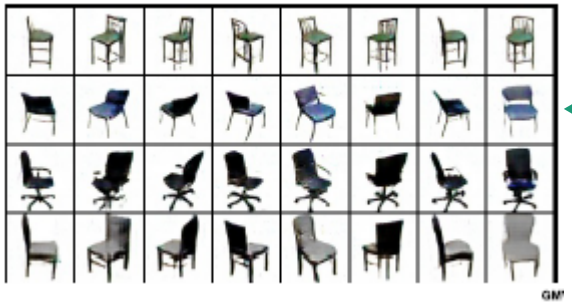


- Objective
  - Generate images by disantangling content and view
    - Eg. Content 1 person, View: position, illumination, etc
  - 2 latent spaces: view and content
    - Generate image pairs: same item with 2 different views
    - Learn to discriminate between generated and real pairs



Column = view

Column = view



1 row = 1 content

GANs

Unsupervised Adversarial Image Reconstruction
(de Bezenac & Pajot 2018 - submitted)

- Objective
  - Infer an underlying signal from incomplete/ noisy observations
- Context
  - Unsupervised learning
    - No access to (signal, lossy observation) pairs
    - No access to underlying signal samples
    - No hypothesis on the form of the signal
- Available informations
  - Lossy observations
  - Access to the corruption process distribution

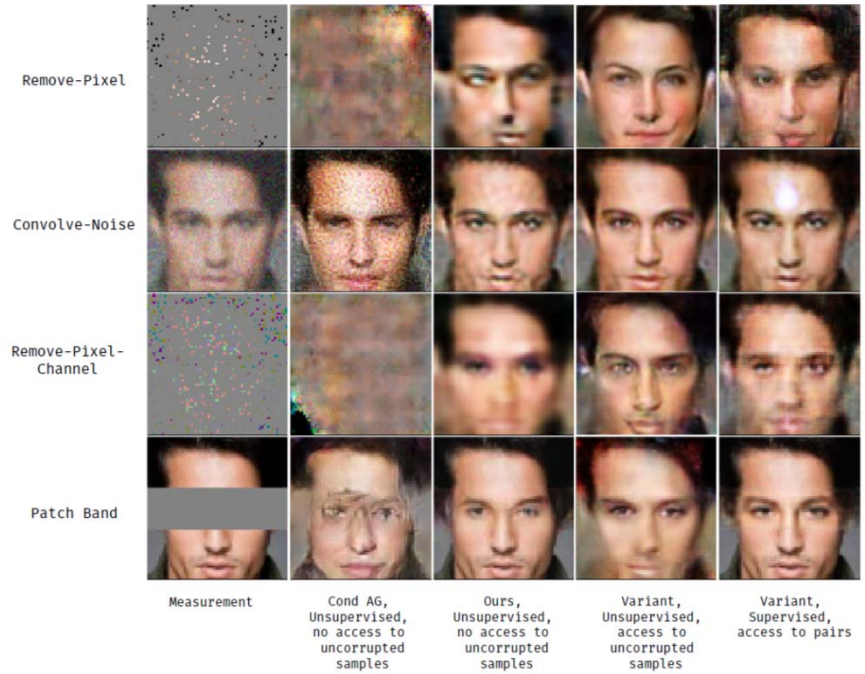


Figure 2: Reconstuction capability for each of the models. Each row correspond to a kind of measurement, and each columns to a different model

Physico-statistical systems

- Context
  - Develop the synergy between model based physically inspired models and data science paradigm
    - Model based approach developed in physics rely on an extensive knowledge of the underlying phenomenon
      - still open challenges, e.g. imperfect knowledge, specification of functional relations impossible, etc
    - Data science approaches offer a complementary/ alternative approach when data characterizing the phenomenon is available
  - Objective
    - Develop the synergy between the two paradigms (physical and data science)

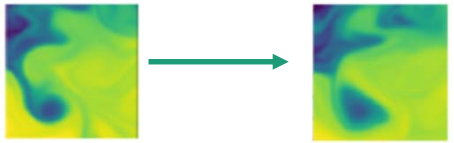
Physico-statistical systems

- Instance
  - PDE guided NN for space time dynamics
    - Can we learn the dynamics of complex phenomena via a data based approach?
    - How can we take benefit of prior physical knowledge?
- Setting
  - Spatio-temporal dynamical systems obeying:
    - $\frac{dX_t}{dt} = F(X_t, D_x X_t)$ with $D_x X_t = (\nabla X, \nabla^2 X, \dots)$
  - Questions
    - How to forecast the evolution of X, from an initial state X_0
    - If we measure many temporal paths of X, is it possible to infer the functional F?
    - How to make F physically plausible?

Incorporating prior knowledge
Physical model for fluid transport
Advection – Diffusion equation

- Describes transport of I through **advection** and **diffusion**

$$\frac{\partial I}{\partial t} + (w.\nabla)I = D\nabla^2 I$$



  - I: quantity of interest (Temperature Image)
  - $w = \frac{\Delta x}{\Delta t}$ motion vector, D diffusion coefficient

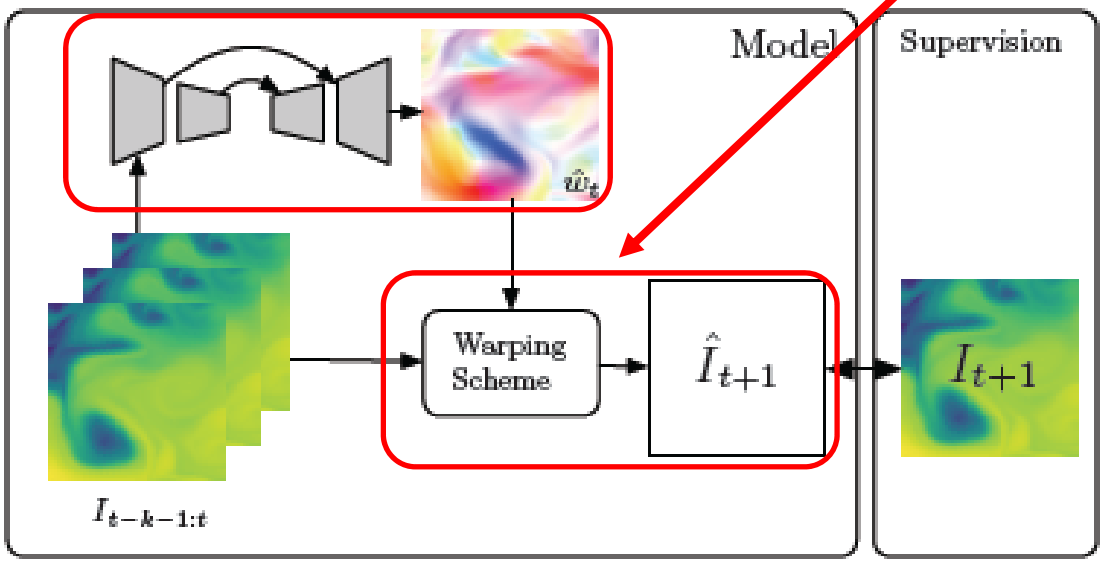
- There exists a closed form solution
  - $I_{t+\Delta t}(x) = (k * I_t)(x - w(x))$

- If we knew the motion vector w and the diffusion coefficient D we could calculate $I_{t+\Delta t}(x)$ from I_t
  - ***w* and *D* unknown**
  - **-> Learn *w* and *D***

Prediction Model
Objective: predict I_{t+1} from past I_t, I_{t-1}, …

- 2 components: Convolution- Deconvolution NN for estimating motion vector w_t

Warping Scheme Implements discretized Advection-Diffusion solution



Past Images

Target image

- End to End learning using only I_{t+1} supervision
- Stochastic gradient optimization
- Performance on par with SOTA assimilation models

Physico-statistical systems (Ayed et al. 2018 – submitted)

- General framework for learning spatio temporal dynamics characteristics of PDEs
  - Model

$$\begin{cases} \widetilde{X}_t = e_\omega(Y_{t-k+1}, \ldots, Y_t) \\ \widehat{X}_{t+1} = f_\theta(\widetilde{X}_t) \\ \widehat{Y}_{t+1} = \mathcal{H}(\widetilde{X}_{t+1}) \end{cases}$$

- Infer current state from past observations Y_{t-i}, e_ω, problem dependent NN, here U-net
- Learn the system dynamics f_θ so as to infer next state, f_θ NN implementing a multi-step finite difference approximation of a PDE
- Predict next observation $\widehat{Y}_{t+1}$, H is a predefined mapping function

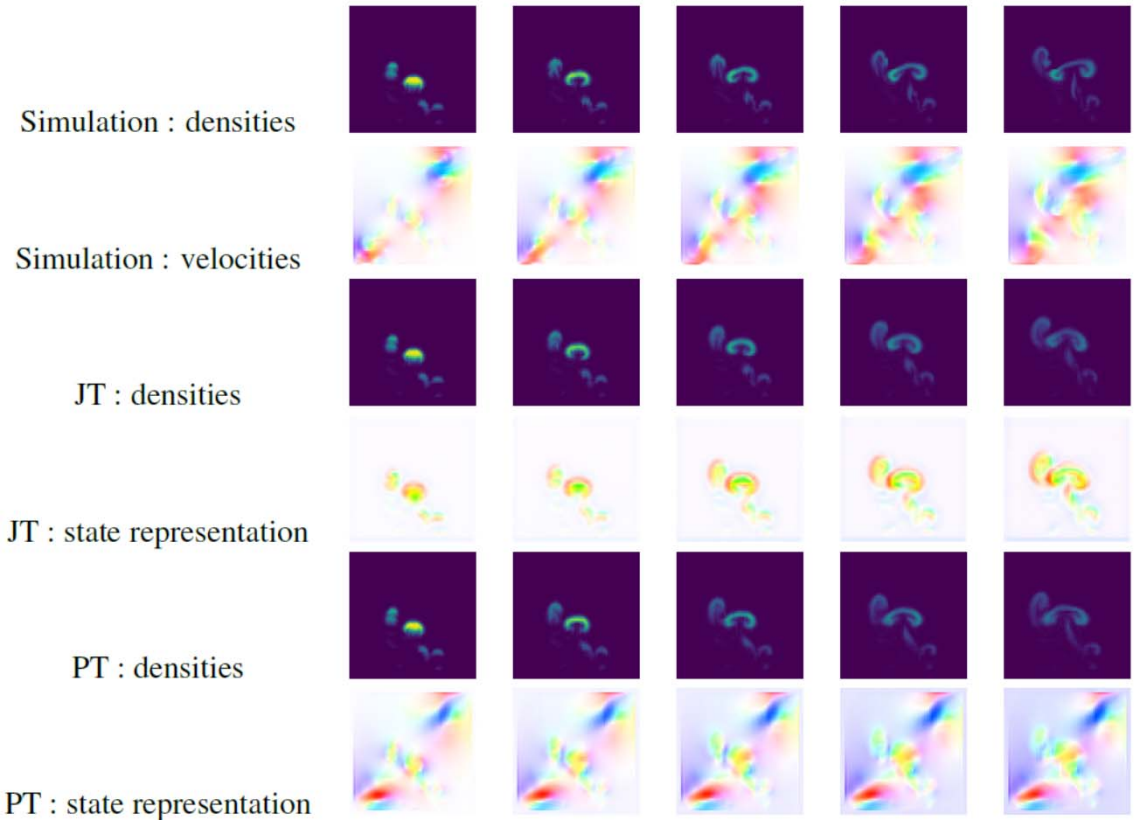
  - Training criterion

$$\min_{\theta,\omega} \mathbb{E}_{(Y_1,\ldots,Y_{k+1})\in \text{Data}}[d(\mathcal{H}(f_\theta(e_\omega(Y_1,\ldots,Y_k))), Y_{k+1})]$$

  - This framework potentially allows us to learn the dynamics of quite general dynamical systems described by PDEs
  - Different instances of the general framework

Physico-statistical systems

- Example
  - Euler Equations and Navier Stokes for incompressible fluids



- Thanks

References and links

- Videos used in the talk
  - Demo of LeNet – Early Convolutional Neural Network

    http://yann.lecun.com/exdb/lenet/index.html
  - NYU Semantic Segmentation with a Convolutional Network (33 categories)

    https://www.youtube.com/watch?v=ZJMtDRbqH40&feature=youtu.be
  - NYU Pedestrian Detection

    https://www.youtube.com/watch?v=MnZNSZGNGyc

    https://www.youtube.com/watch?v=UPVvd8WNUks
  - Hand gesture Recognition

    https://www.youtube.com/watch?v=GhqOMJIHD8A

General References

- The 1960s - Early days of Neural Networks
  - Widrow B., Stearns S.D., Adaptive signal processing, Prentice-Hall, 1985
  - Minsky M., Papert S.A., Perceptrons: An Introduction to Computational Geometry, Expanded Edition, 1987
- The 1990s – many books were published at that time:
  - Hertz J.A. , Krogh A.S., Palmer R.G. Introduction To The Theory Of Neural Computation (Santa Fe Institute Series), 1991, introduces a variety of NN models developed in the 80es
  - Bishop C.M. , Neural Networks for Pattern Recognition, Oxford University Press, 1995 (you may also have a try at: Bishop C.M, Pattern Recognition and Machine Learning, Springer 2006)
  - Introduction to Statistical Learning Theory: Vapnik V., The Nature of Statistical Learning Theory, Springer-Verlag New York, 2000
- The 2010s
  - Many courses are available on line, for a book, you may have a look at Goodfellow I., Bengio Y. Courville A. , Deep Learning ,An MIT Press book, http://www.deeplearningbook.org/

References: papers used as illustrations for the presentation

- Baydin Atilim Gunes , Barak A. Pearlmutter, Alexey Andreyevich Radul, Automatic differentiation in machine learning: a survey. CoRR abs/1502.05767 (2017)
- Cadène R., Thomas Robert, Nicolas Thome, Matthieu Cord:M2CAI Workflow Challenge: Convolutional Neural Networks with Time Smoothing and Hidden Markov Model for Video Frames Classification. CoRR abs/1610.05541 (2016)
- Chen M. Denoyer L., Artieres T. Multi-view Generative Adversarial Networks without supervision, 2017 , https://arxiv.org/abs/1711.00305.
- Cho, K., Gulcehre, B. van M.C., Bahdanau, D., Bougares, F., Schwenk, H. and Bengio, Y. 2014. Learning Phrase Representations using RNN Encoder – Decoder for Statistical Machine Translation. EMNLP 2014 (2014), 1724–1734.
- Durand T. , Thome, N. and Cord M., WELDON: Weakly Supervised Learning of Deep Convolutional Neural Networks, CVPR 2016.
- Frome, A., Corrado, G., Shlens, J., Bengio, S., Dean, J., Ranzato, M.A. and Mikolov, T. 2013. DeViSE: A Deep Visual-Semantic Embedding Model. NIPS 2013 (2013).
- Gatys, L.A., Ecker, A.S. and Bethge, M. 2015. A Neural Algorithm of Artistic Style. arXiv preprint. (2015), 3–7.
- Goodfellow I, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, Yoshua Bengio , Generative adversarial nets, NIPS 2014, 2672-2680
- Ioffe S., Szegedy C.: Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift, 1995, http://arxiv.org/abs/1502.03167
- Krizhevsky, A., Sutskever, I. and Hinton, G. 2012. Imagenet classification with deep convolutional neural networks. Advances in Neural Information. (2012), 1106–1114.

References: papers used as illustrations for the presentation

- Le, Q., Ranzato, M., Monga, R., Devin, M., Chen, K., Corrado, G., Dean, J. and Ng, A. 2012. Building high-level features using large scale unsupervised learning. Proceedings of the 29th International Conference on Machine Learning (ICML-12). (2012), 81–88.
- Pearlmutter B.A., Gradient calculations for dynamic recurrent neural networks: a survey, IEEE Trans on NN, 1995
- Radford, Luke Metz, Soumith Chintala, Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks, 2016, http://arxiv.org/abs/1511.06434
- Reed, S., Akata, Z., Yan, X., Logeswaran, L., Schiele, B. and Lee, H. 2016. Generative Adversarial Text to Image Synthesis. Icml (2016), 1060–1069.
- Ruder S. (2016). An overview of gradient descent optimization algorithms. arXiv preprint arXiv:1609.04747.
- Srivastava N., Geoffrey E. Hinton, Alex Krizhevsky, Ilya Sutskever, Ruslan Salakhutdinov: Dropout: a simple way to prevent neural networks from overfitting. Journal of Machine Learning Research 15(1): 1929-1958 (2014)
- Sutskever, I., Vinyals, O. and Le, Q. V 2014. Sequence to sequence learning with neural networks. Advances in Neural Information Processing Systems (NIPS) (2014), 3104–3112.
- Vinyals, O., Toshev, A., Bengio, S. and Erhan, D. 2015. Show and Tell: A Neural Image Caption Generator, CVPR 2015: 3156-3164
- *Wu, Yonghui, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Łukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, Jeffrey Dean,* Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation*, Technical Report, 2016.*
- Johnson, J., Hariharan, B., van der Maaten, L., Hoffman, J., Fei-Fei, L., Zitnick, C. L., & Girshick, R. (2017). Inferring and Executing Programs for Visual Reasoning. In *ICCV* (pp. 3008–3017). ttps://doi.org/10.1109/ICCV.2017.325
- Johnson, J., Hariharan, B., van der Maaten, L., Fei-Fei, L., Zitnick, C. L., & Girshick, R. (2017). CLEVR: A Diagnostic Dataset for Compositional Language and Elementary Visual Reasoning. In *CVPR* (pp. 1988–1997). https://doi.org/10.1109/CVPR.2017.215
- Lerer, A., Gross, S., & Fergus, R. (2016). Learning Physical Intuition of Block Towers by Example. In *Icml* (pp. 430–438). Retrieved from http://arxiv.org/abs/1603.01312
- Mathieu, M., Couprie, C., & LeCun, Y. (2016). Deep multi-scale video prediction beyond mean square error. In *ICLR* (pp. 1–14). Retrieved from http://arxiv.org/abs/1511.05440