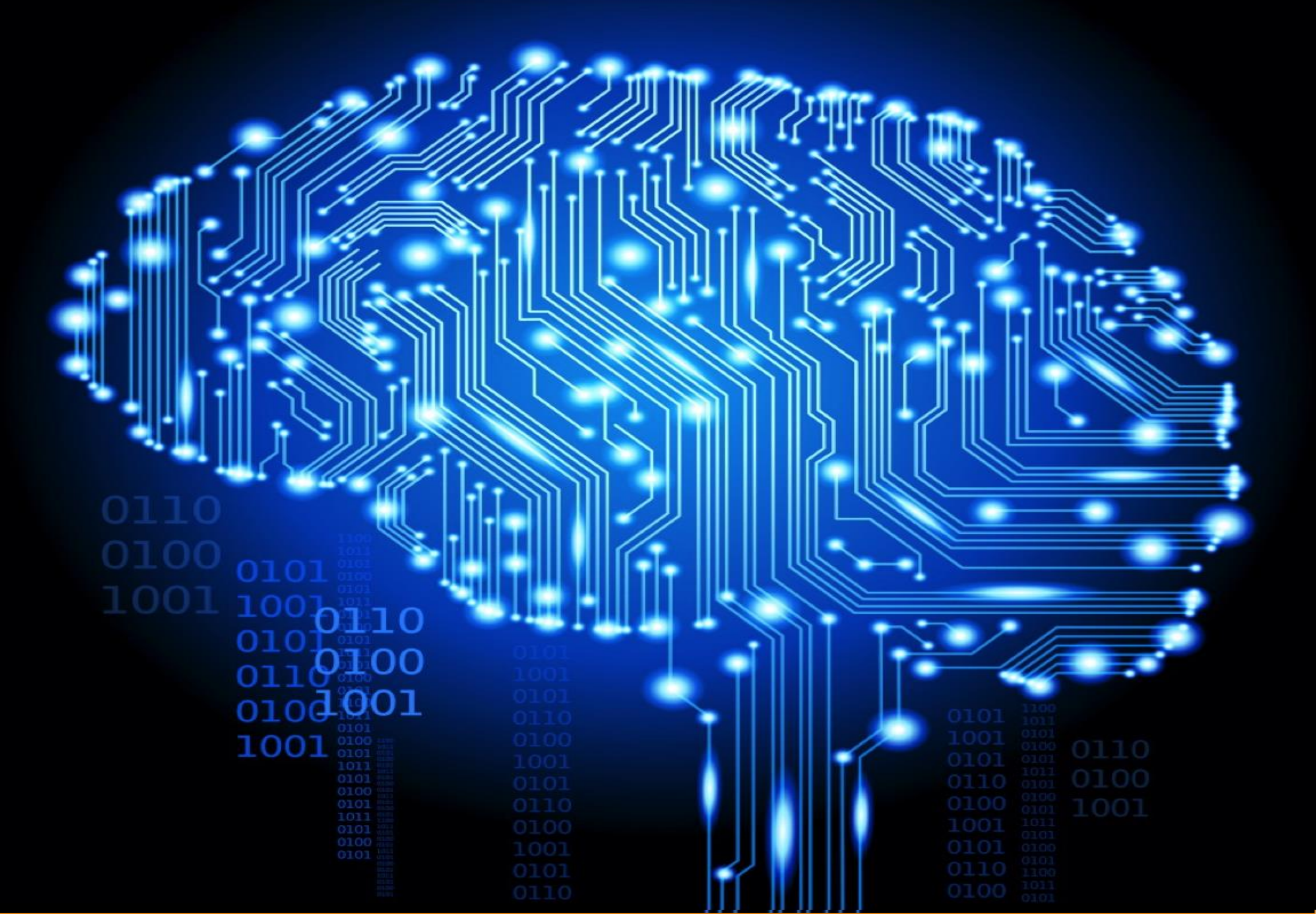


Lecture 1: Introduction to Deep Learning and Neural Networks

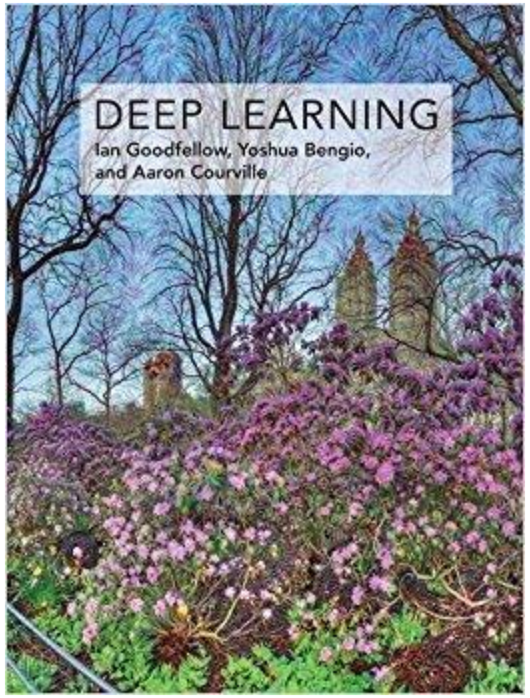
Deep Learning @ UvA

Prerequisites

- Machine Learning 1
- Calculus, Linear Algebra
  - Derivatives, integrals
  - Matrix operations
  - Computing lower bounds, limits
- Probability and Statistics
- Advanced programming
- Time, patience and drive

Course overview

- Course: Theory (4 hours per week) + Labs (4 hours per week)
- Book: *Deep Learning*, (available online)
  by I. Goodfellow, Y. Bengio, A. Courville
- World-class experts give some of the lectures
- All material on
  http://deeplearningamsterdam.github.io
- We organize the course through Piazza.
  Please, subscribe today!
  - **Link:** http://piazza.com/university_of_amsterdam/fall2017/uvadlc/home
  - I will send an email with the link after the course also
- Final grade = 50% from lab assignments + 50% from final exam



Learning goals

- Design in theory and program in practice basic neural networks, such as multi-layer perceptrons
- Solve in pen and paper the backpropagation algorithm for deriving the gradients of deep learning models
- Describe, analyze and implement optimization methods for deep learning models, including SGD, Nestorov's Momentum, RMSprop, Adam.
- Describe, analyze and implement regularization methods for deep learning models, including weight decay, early stopping, dropout
- Describe the most popular, state-of-the-art convolutional and recurrent neural network architectures and their advantages
- Analyze the reasons behind the success of recent convolutional neural network architectures
- Analyze the behavior of recurrent neural networks and the difficulties of training them

Learning goals (2)

- State the differences between Supervised and Unsupervised learning of Deep Learning models and describe the most popular Unsupervised Learning method
- Perform transfer learning from pretrained networks to novel inference tasks, such as image classification and regression.
- State the differences between various Generative Models, including Variational Autoencoders and Generative Adversarial Networks
- Describe the connection between Autoencoders, Variational Autoencoders and Unsupervised Learning
- Derive the variational inference principle for deriving optimization lower bounds on neural networks
- Describe and implement applications of Deep Learning models in Computer Vision (object classification, detection, segmentation), Natural Language Processing (machine translation, question answering), Reinforcement Learning, currently popularized in software products like Google Photos, Google Translate, Self-driving Autonomous cars, DeepMind AlphaGO

Practicals

- 3 lab equally weighted assignments, <u>individually</u> (Max grade: 3x20)
  - However, you are more than encouraged to cooperate and help each other. In fact, the top 3 contributors in Piazza (questions, answers, participation) will get +1 grade
  - Python + Tensorflow
- Paper presentation, <u>in groups of 3</u> (Max grade: 15)
  - About 7 min per presentation, 3 min for questions, we will give you template, dates to be announced.
- 1 poster presentation in the end organized as a workshop, <u>in groups of 3</u>. I will try to invite researchers and companies. (Max grade: 25)
  - What to present: Find a paper you really like and for which existing code is available. Run experiments that you believe is relevant, make an adaptation, present it!
- <u>By next Monday make your team:</u> we will prepare a Google Spreadsheet

Prepare to vote

Internet

1 Go to shakeq.com

2 Log in with uva507

This presentation has been loaded without the Shakespeak add-in.
Want to download the add-in for free? Go to http://shakespeak.com/en/free-download/.

TXT

1 Text to 06 4250 0030

2 Type uva507 <space> your choice (*e.g. uva507 b*)

No additional charge per message

shakespeak

In this edition we will try for a more interactive course. Would you like to try this out?

A. Yes, why not?
B. Nope!
C. Yes, under conditions.

The question will open when you start your session and slideshow.

shakespeak

Time: 60s

This presentation has been loaded without the Shakespeak add-in.
Want to download the add-in for free? Go to http://shakespeak.com/en/free-download/.

Internet Go to shakeq.com

TXT

Votes: 65

In this edition we will try for a more interactive course. Would you like to try this out?

A. Yes, why not? — 81.5%

B. Nope! — 0.0%

C. Yes, under conditions. — 18.5%

Closed

shakespeak

Who we are and how to reach us

- Efstratios Gavves, Assistant Professor at QUVA Deep Vision Lab (C3.229)
  - Focus on Continuous Learning, with applications to Computer Vision and Video Analysis
- TAs: Kirill Gavrilyuk, Berkay Kicanaoglu, Peter O'Connor, Tom Runia



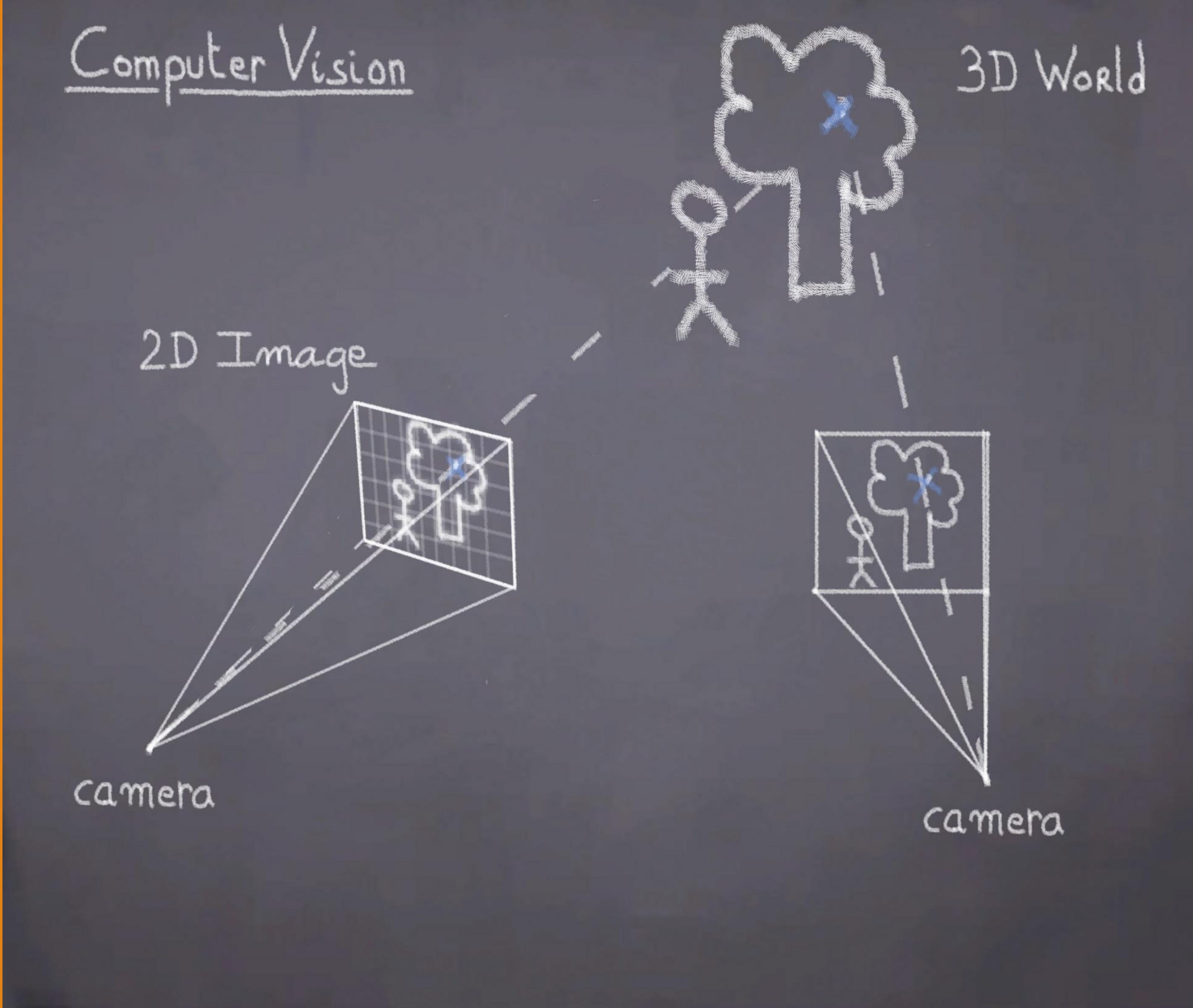


- Course website: http://deeplearningamsterdam.github.io
  - Lecture sides & notes, practicals
- Virtual classroom
  - **Piazza**: http://piazza.com/university_of_amsterdam/fall2017/uvadlc/home **Access Code:** deeplearningamsterdam
  - **Datanose:** https://datanose.nl/#course[61784]
  - **Blackboard** is available, but most of the activities through Piazza and the course website
- A couple of volunteers would be great to record student questions so that we can answer them in the course website. Come and find me in the break if interested.

Lecture Overview

- Deep Learning in
  - Computer Vision
  - Natural Language Processing (NLP)
  - Speech
  - Robotics and AI
  - Music and the arts!
- A brief history of Neural Networks and Deep Learning
- Basics of Neural Networks

Deep Learning in Computer Vision



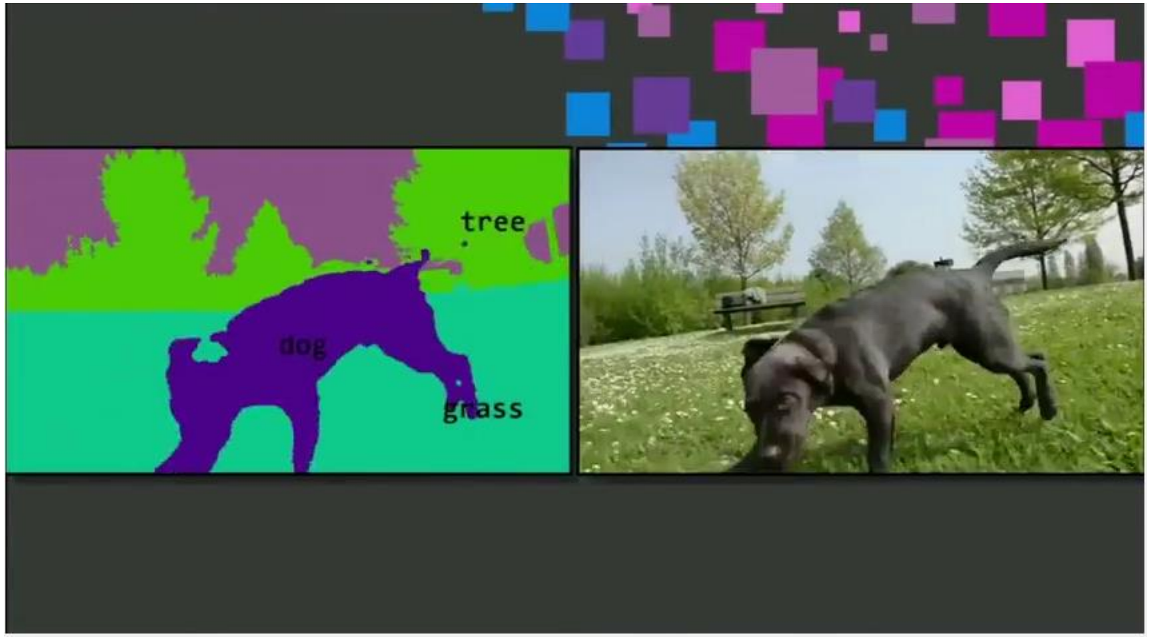
Object and activity recognition

Click to go to the video in Youtube

Large-scale Video Classification with Convolutional Neural Networks, CVPR 2014

Object detection, segmentation, pose estimation

Click to go to the video in Youtube



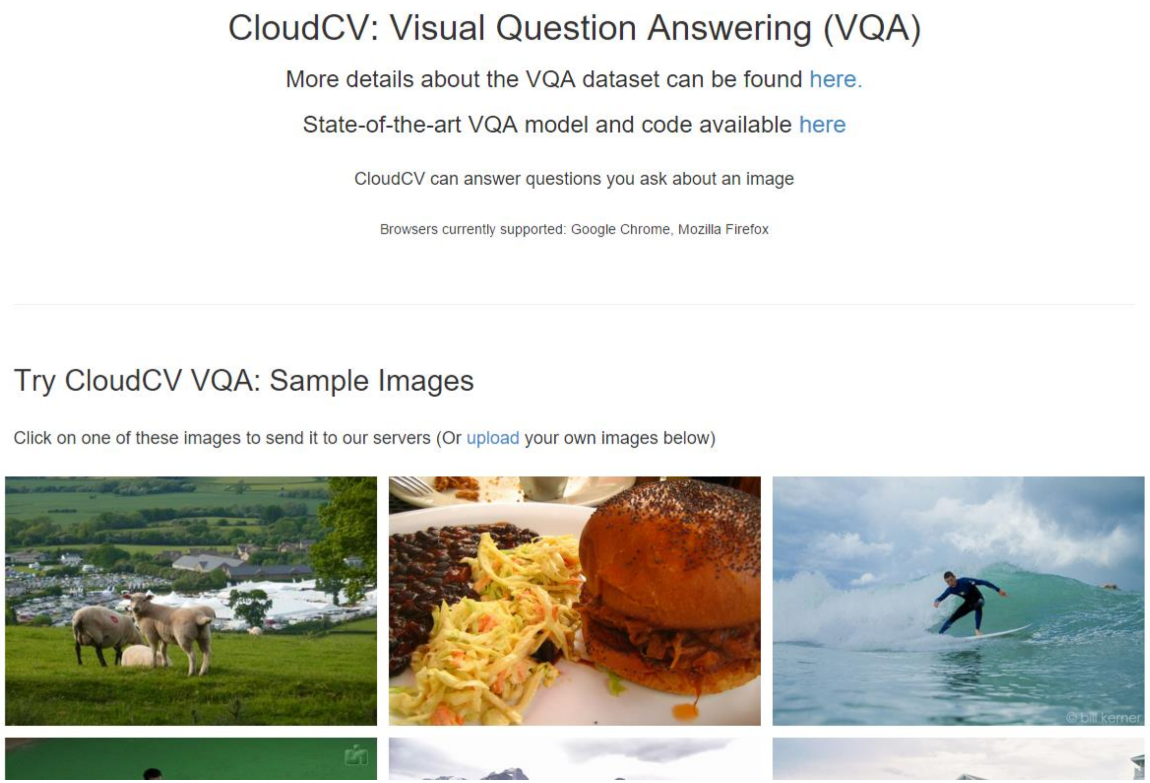
Microsoft Deep Learning Semantic Image Segmentation

Image captioning and Q&A

Click to go to the video in Youtube

NeuralTalk and Walk, recognition, text description of the image while walking

Click to go to the website



Why should we be impressed?

- Vision is ultra challenging!
  - For a small 256x256 resolution and for 256 pixel values
  - a total $2^{524,288}$ of possible images
  - In comparison there are about 10^{24} stars in the universe
- Visual object variations
  - Different viewpoints, scales, deformations, occlusions
- Semantic object variations
  - Intra-class variation
  - Inter-class overlaps

Deep Learning in Robotics

Self-driving cars

Click to go to the video in Youtube

Drones and robots

Click to go to the video in Youtube

Deep Sensorimotor Learning

Game AI

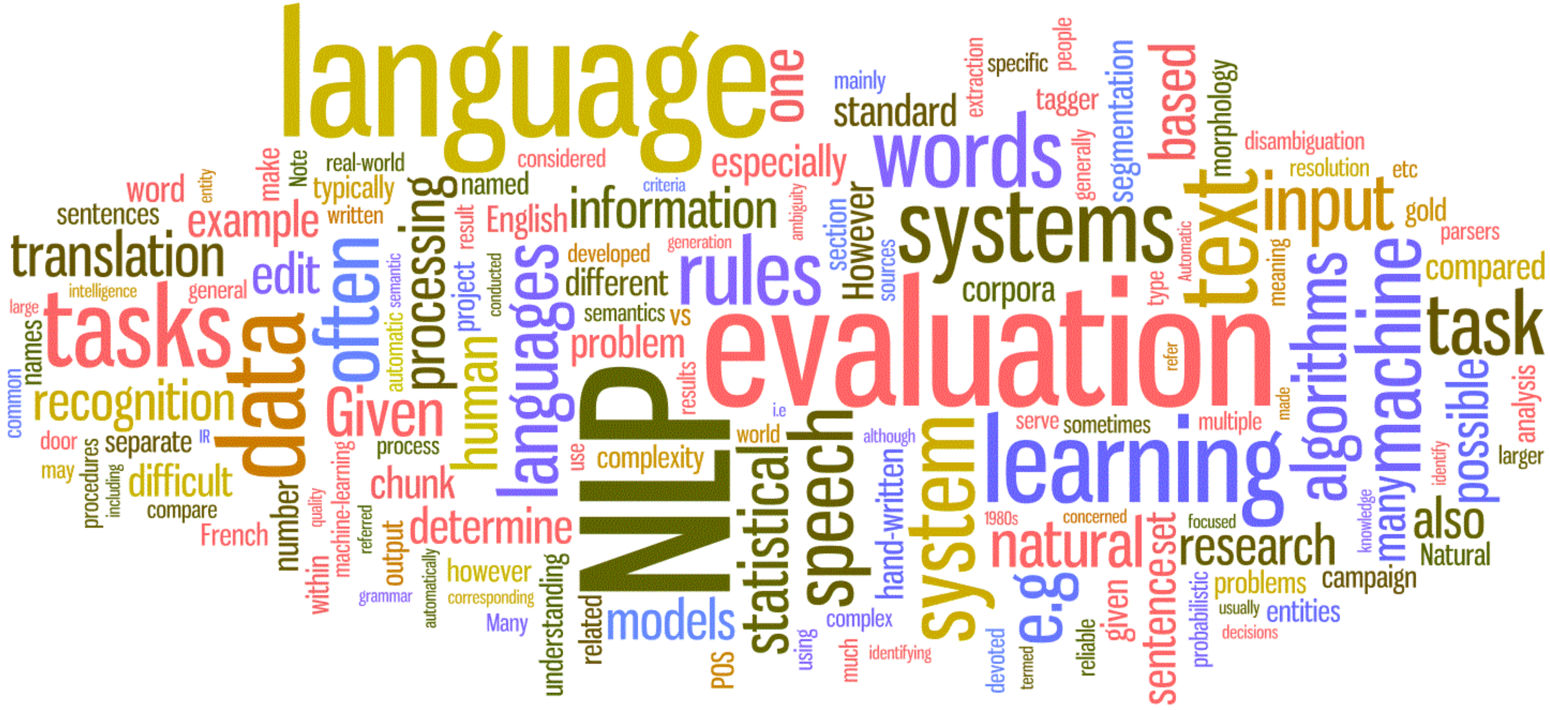
Click to go to the video in Youtube

Google DeepMind's Deep Q-learning playing Atari Breakout

Why should we be impressed?

- Typically robotics are considered in controlled environments
  - Laboratory settings, Predictable positions, Standardized tasks (like in factory robots)
- What about real life situations?
  - Environments constantly change, new tasks need to be learnt without guidance, unexpected factors must be dealt with
- Game AI
  - At least $10^{10^{48}}$ possible GO games. Where do we even start?

Deep Learning in NLP and Speech

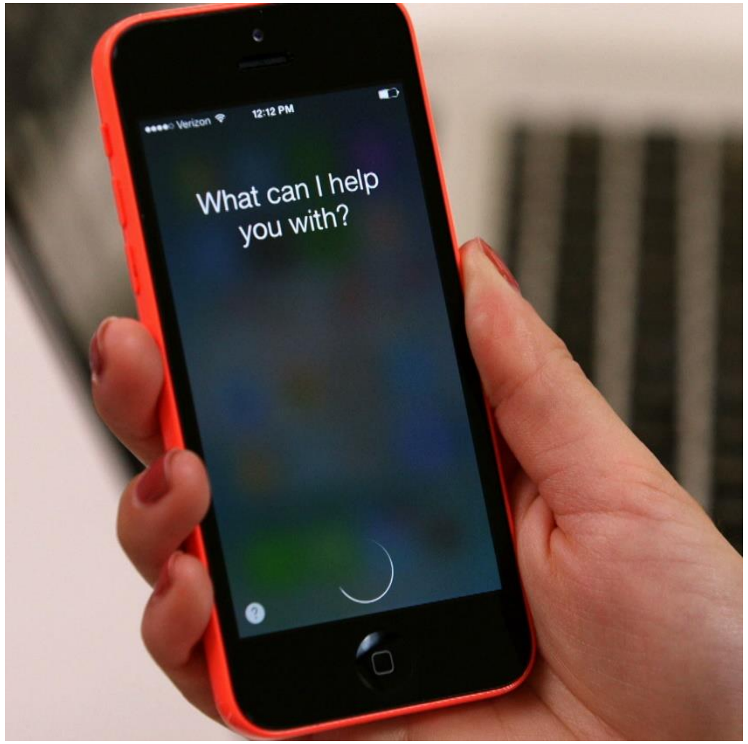


Word and sentence representations

Newspapers			
New York San Jose	New York Times San Jose Mercury News	Baltimore Cincinnati	Baltimore Sun Cincinnati Enquirer
NHL Teams			
Boston Phoenix	Boston Bruins Phoenix Coyotes	Montreal Nashville	Montreal Canadiens Nashville Predators
NBA Teams			
Detroit Oakland	Detroit Pistons Golden State Warriors	Toronto Memphis	Toronto Raptors Memphis Grizzlies
Airlines			
Austria Belgium	Austrian Airlines Brussels Airlines	Spain Greece	Spainair Aegean Airlines
Company executives			
Steve Ballmer Samuel J. Palmisano	Microsoft IBM	Larry Page Werner Vogels	Google Amazon

Table 2: Examples of the analogical reasoning task for phrases (the full test set has 3218 examples). The goal is to compute the fourth phrase using the first three. Our best model achieved an accuracy of 72% on this dataset.

Speech recognition and Machine translation

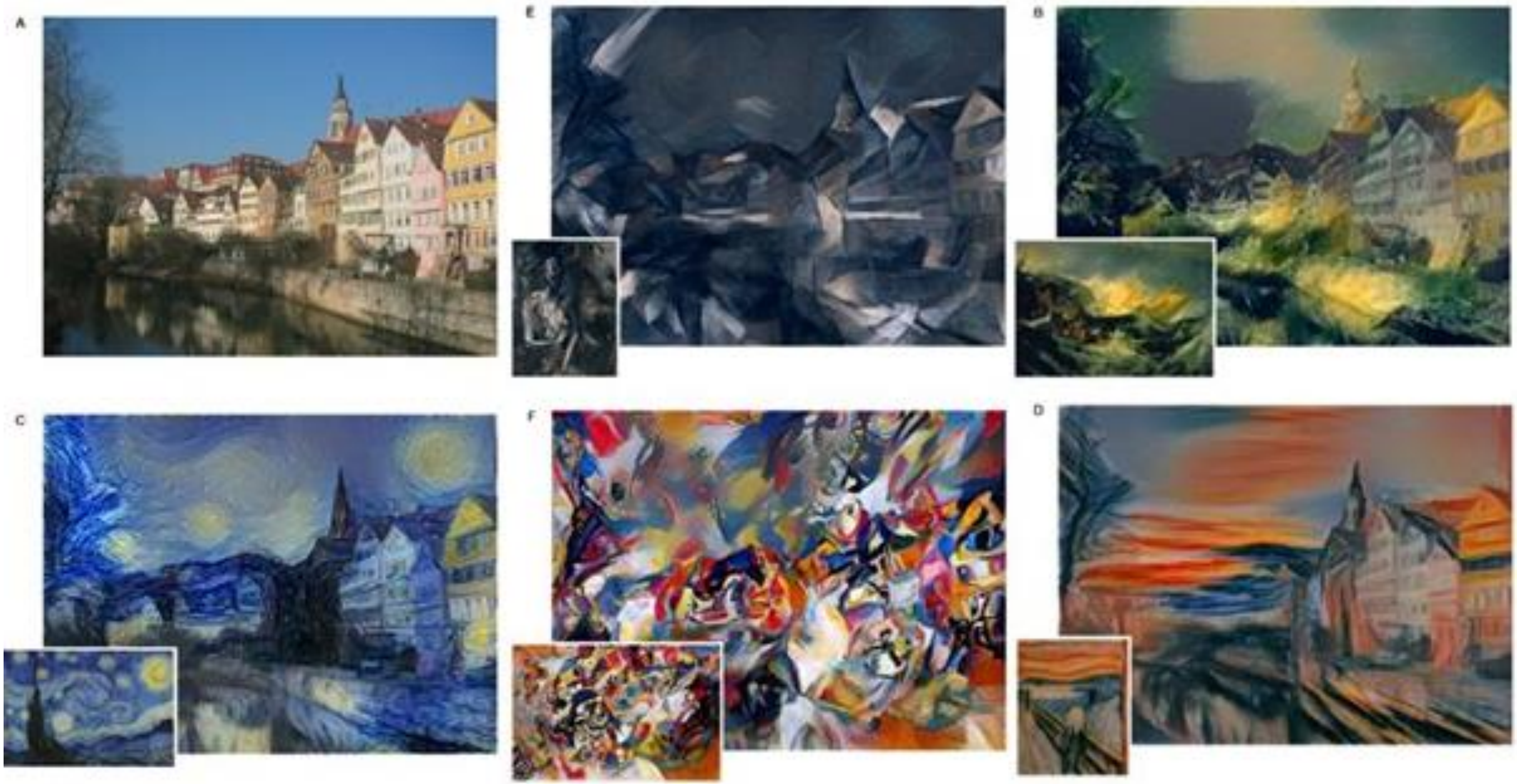


Why should we be impressed?

- NLP is an extremely complex task
  - synonymy (“chair”, “stool” or “beautiful”, “handsome”)
  - ambiguity (“I made her duck”, “Cut to the chase”)
- NLP is very high dimensional
  - assuming 150K english words, we need to learn 150K classifiers
  - with quite sparse data for most of them
- Beating NLP was considered the crown jewel of AI
- However, true AI goes beyond NLP, Vision, Robotics, … alone

Deep Learning in the arts

Imitating famous painters

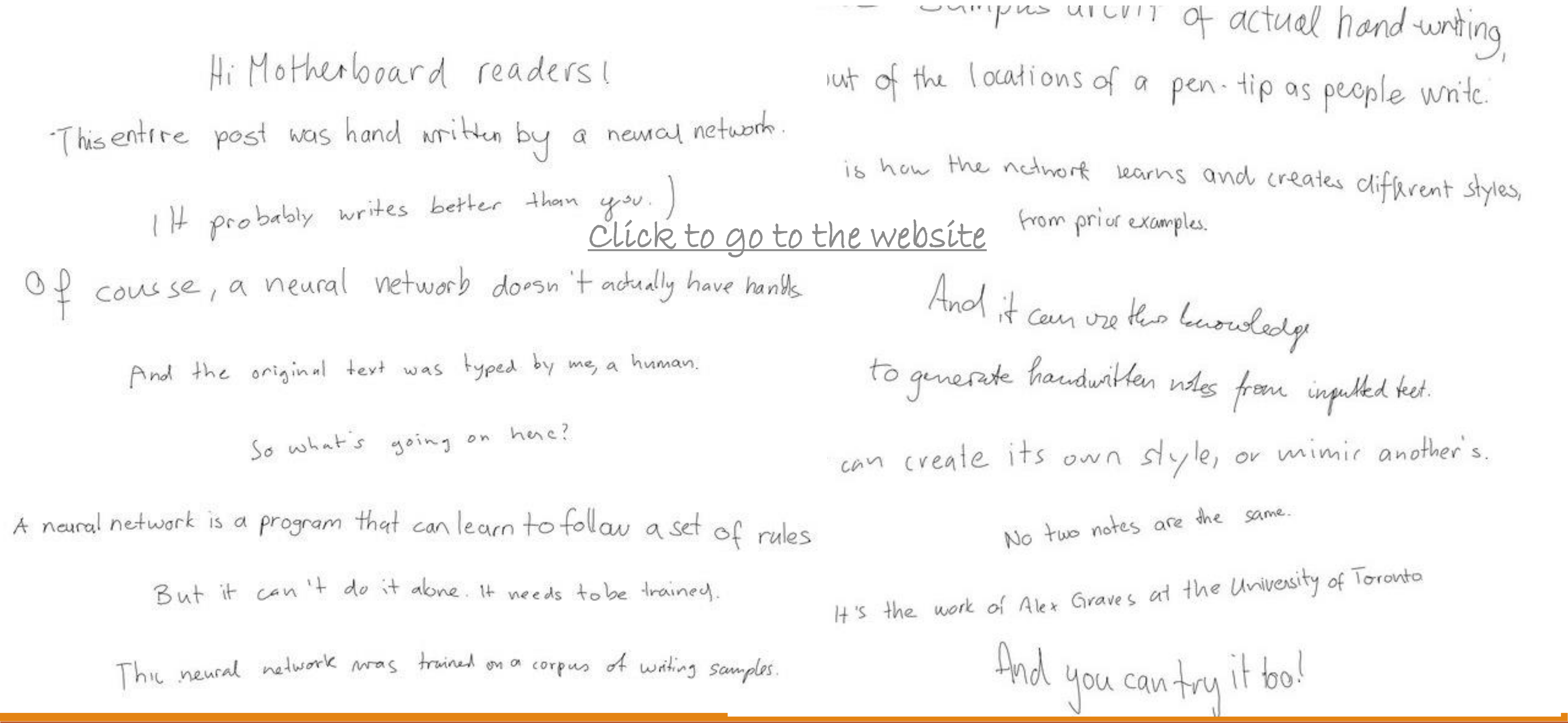


Or dreaming …

Click to go to the video in Youtube

Journey Through the Layers of the Mind

Handwriting

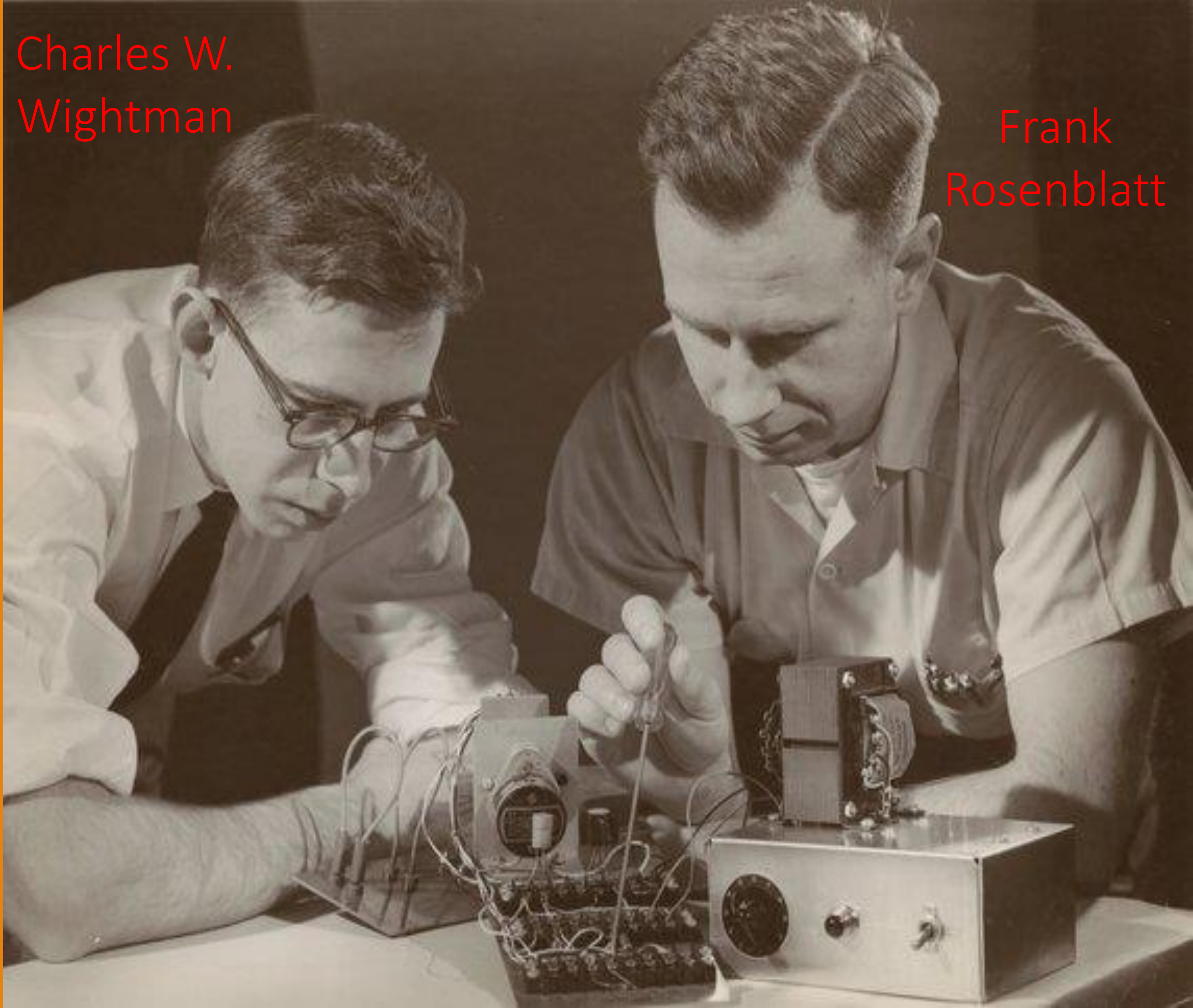


Click to go to the website

Why should we be impressed?

- Music, painting, etc. are tasks that are uniquely human
  - Difficult to model
  - Even more difficult to evaluate (if not impossible)
- If machines can generate novel pieces even remotely resembling art, they must have understood something about “beauty”, “harmony”, etc.
- Have they really learned to generate new art, however?
  - Or do they just fool us with their tricks?

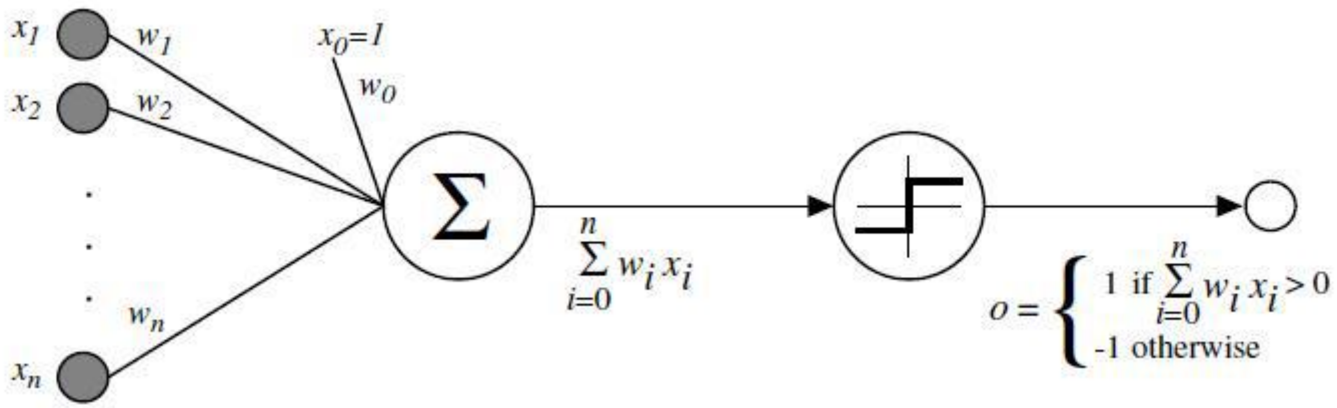
A brief history of Neural Networks & Deep Learning



First appearance (roughly)

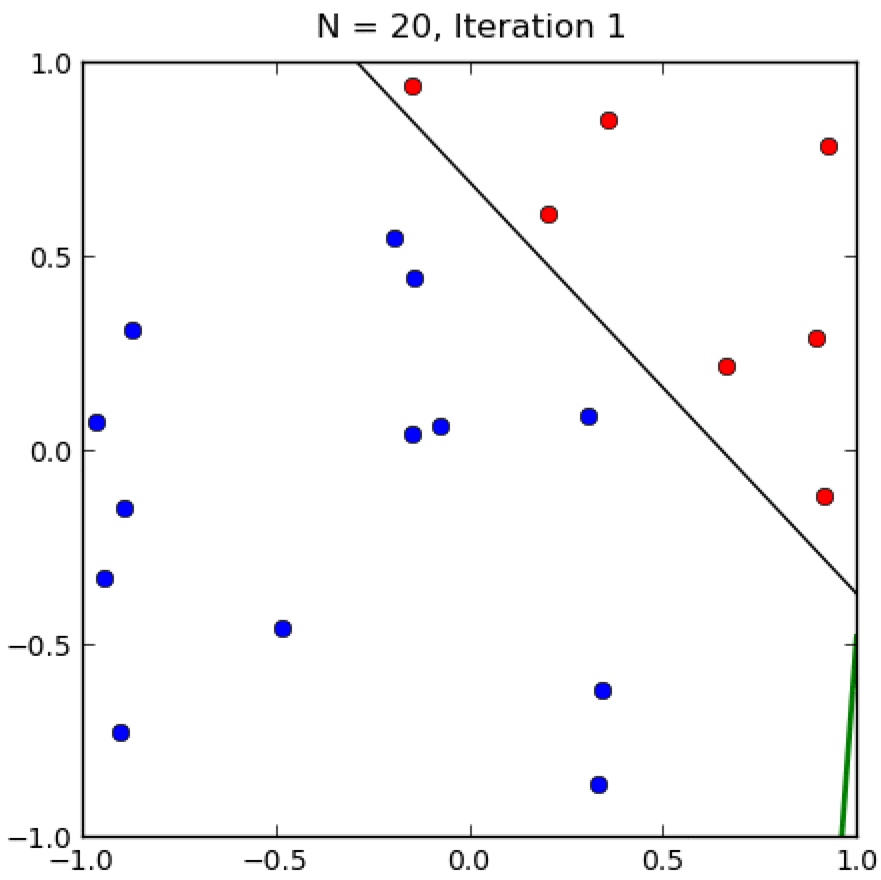
Perceptrons

- Rosenblatt proposed a machine for binary classifications
- Main idea
  - One weight w_i per input x_i
  - Multiply weights with respective inputs and add bias x_0 =+1
  - If result larger than threshold return 1, otherwise 0



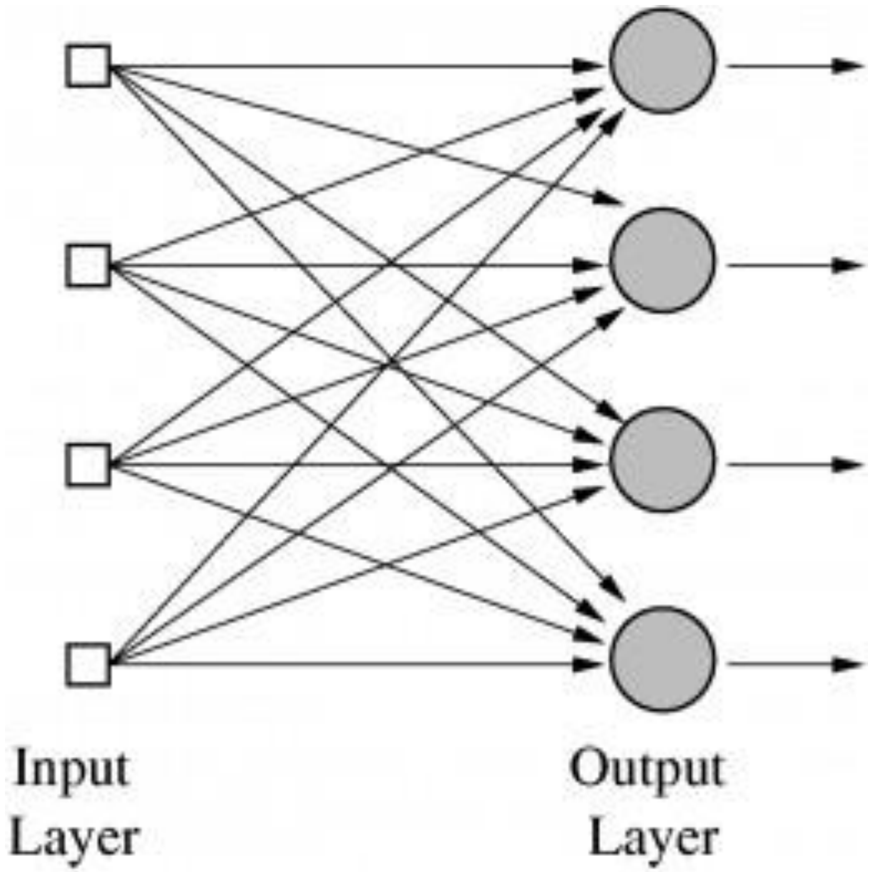
Training a perceptron

- Rosenblatt's innovation was mainly the learning algorithm for perceptrons
- Learning algorithm
  - Initialize weights randomly
  - Take one sample x_i and predict y_i
  - For erroneous predictions update weights
    - If the output was $\widehat{y_i} = 0$ and $y_i = 1$, increase weights
    - If the output was $\widehat{y_i} = 1$ and $y_i = 0$, decrease weights
  - Repeat until no errors are made



From a perceptron to a neural network

- One perceptron = one decision
- What about multiple decisions?
  - E.g. digit classification
- Stack as many outputs as the possible outcomes into a layer
  - Neural network
- Use one layer as input to the next layer
  - Multi-layer perceptron (MLP)



What could be a problem with perceptrons?

A. They can only return one output, so only work for binary problems
B. They are linear machines, so can only solve linear problems
C. They can only work for vector inputs
D. They are too complex to train, so they can work with big computers only

The question will open when you start your session and slideshow.

Time: 60s

shakespeak

This presentation has been loaded without the Shakespeak add-in.
Want to download the add-in for free? Go to http://shakespeak.com/en/free-download/.

Votes: 59

What could be a problem with perceptrons?

A.	They can only return one output, so only work for binary problems	15.3%
B.	They are linear machines, so can only solve linear problems	74.6%
C.	They can only work for vector inputs	10.2%
D.	They are too complex to train, so they can work with big computers only	0.0%

shakespeak

Closed

XOR & Multi-layer Perceptrons

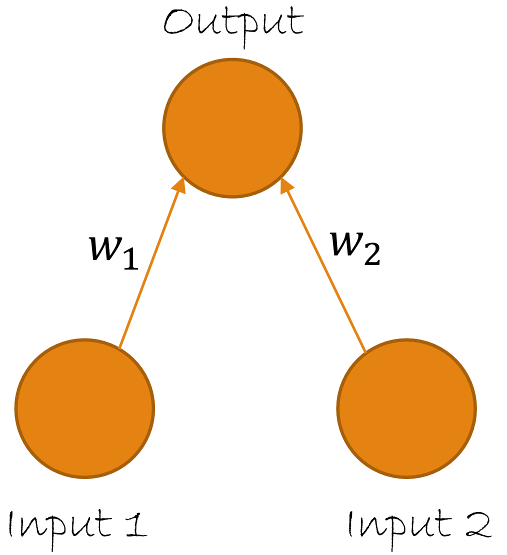
- However, the exclusive or (XOR) cannot be solved by perceptrons
  - [Minsky and Papert, “Perceptrons”, 1969]

Input 1	Input 2	Output
1	1	0
1	0	1
0	1	1
0	0	0

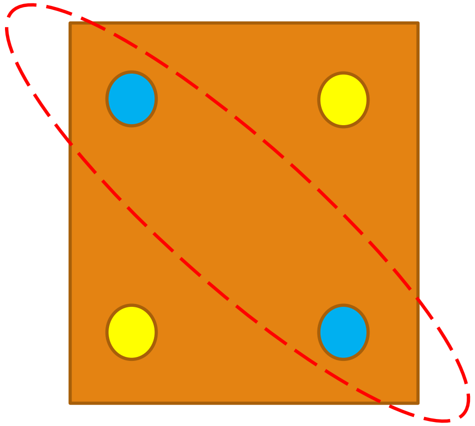
  - $0\, w_1 + 0 w_2 < \theta \rightarrow 0 < \theta$
  - $0\, w_1 + 1 w_2 > \theta \rightarrow w_2 > \theta$
  - $1\, w_1 + 0 w_2 > \theta \rightarrow w_1 > \theta$
  - $1\, w_1 + 1 w_2 < \theta \rightarrow w_1 + w_2 < \theta$

Inconsistent!!

Graphically

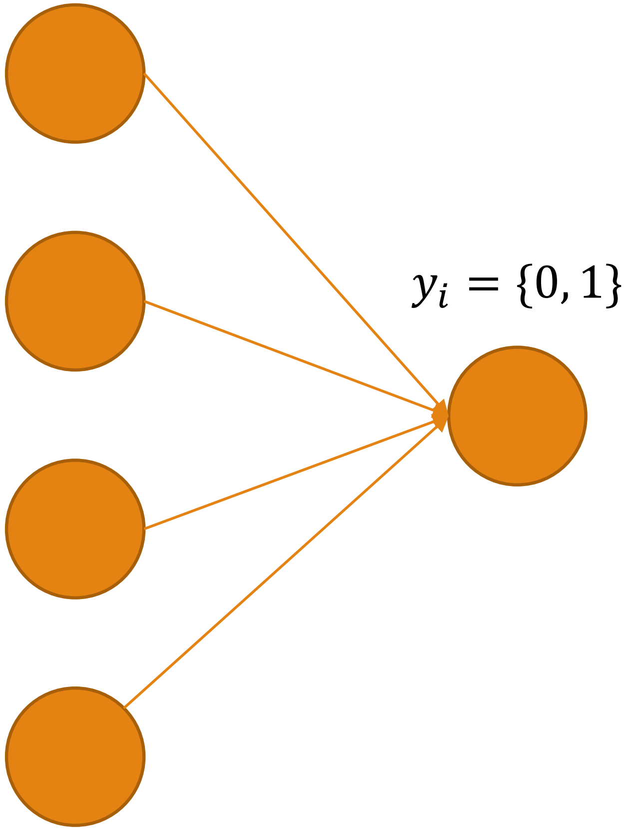


The classification boundary to solve XOR is not a line!!



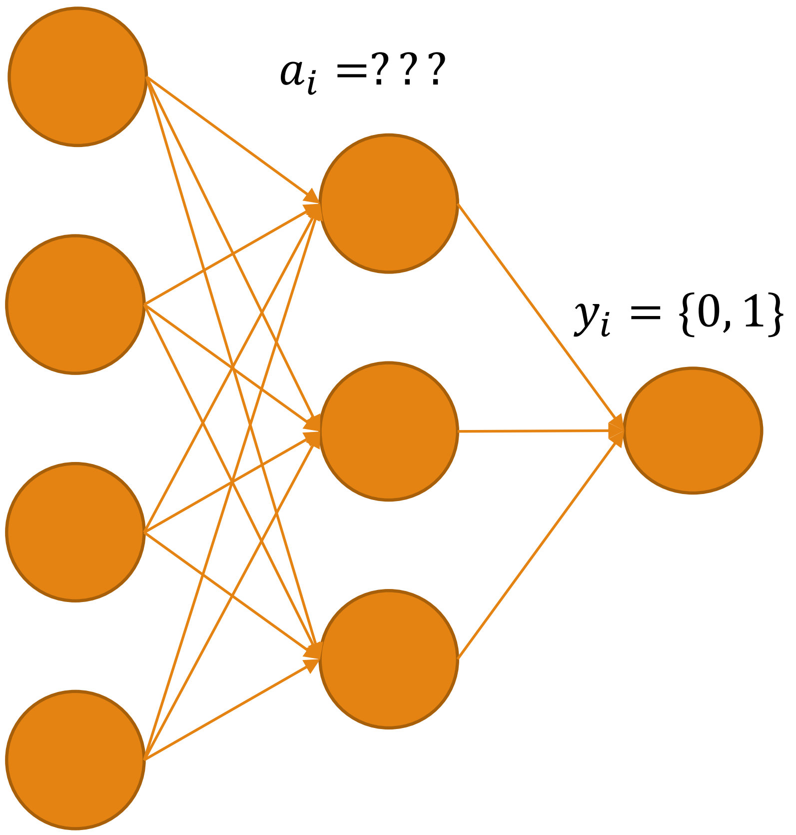
Minsky & Multi-layer perceptrons

- Interestingly, Minksy **never said** XOR cannot be solved by neural networks
  - Only that XOR cannot be solved with **1 layer** perceptrons
- Multi-layer perceptrons can solve XOR
  - 9 years earlier Minsky built such a multi-layer perceptron
- However, how to train a multi-layer perceptron?
- Rosenblatt's algorithm not applicable
  - It expects to know the desired target

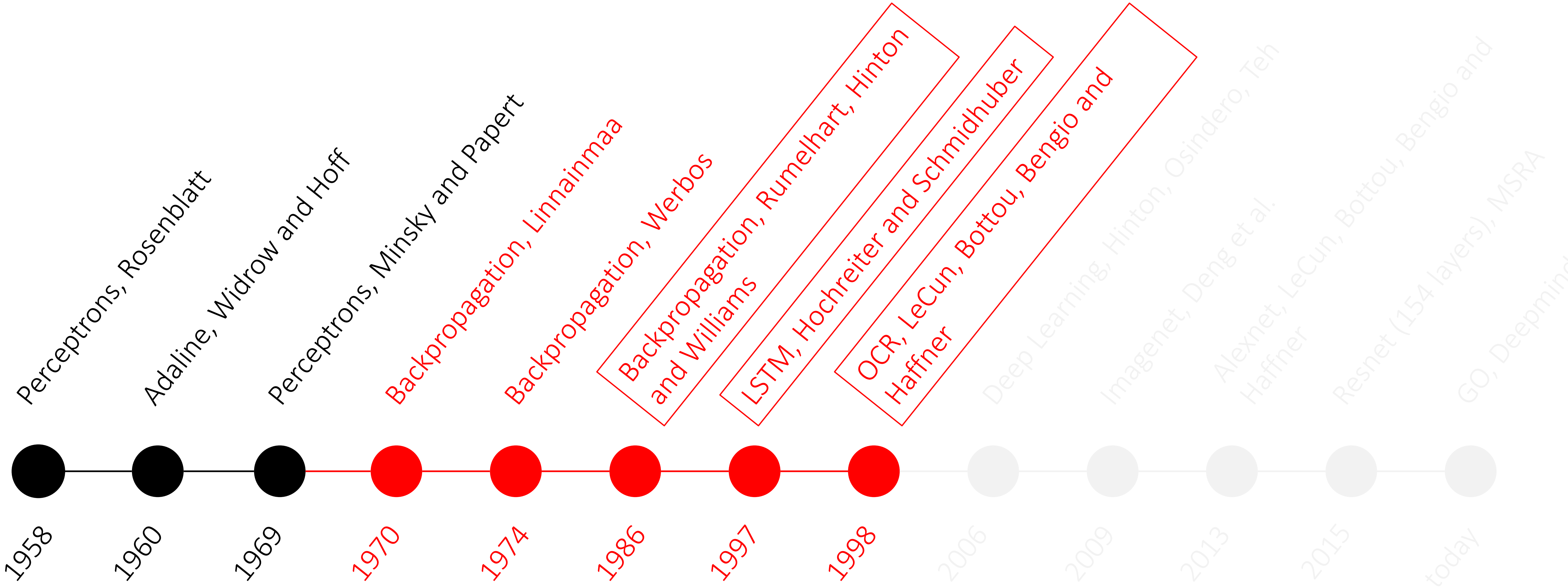


Minsky & Multi-layer perceptrons

- Interestingly, Minksy **never said** XOR cannot be solved by neural networks
  - Only that XOR cannot be solved with **1 layer** perceptrons
- Multi-layer perceptrons can solve XOR
  - 9 years earlier Minsky built such a multi-layer perceptron
- However, how to train a multi-layer perceptron?
- Rosenblatt's algorithm not applicable
  - It expects to know the desired target
  - For hidden layers we cannot know the desired target a_i, which must be learned



The “AI winter” despite notable successes



The first "AI winter"

- What everybody thought: "If a perceptron cannot even solve XOR, why bother?
  - Also, the exaggeration did not help (walking, talking robots were promised in the 60s)
- As results were never delivered, further funding was slushed, neural networks were damned and AI in general got discredited
- "The **AI winter** is coming"
- Still, a few people persisted
- Significant discoveries were made, that laid down the road for today's achievements

Backpropagation

- Learning multi-layer perceptrons now possible
  - XOR and more complicated functions can be solved
- Efficient algorithm
  - Process hundreds of example without a sweat
  - Allowed for complicated neural network architectures
- Backpropagation still is the backbone of neural network training today
- Digit recognition in cheques (OCR) solved before the 2000

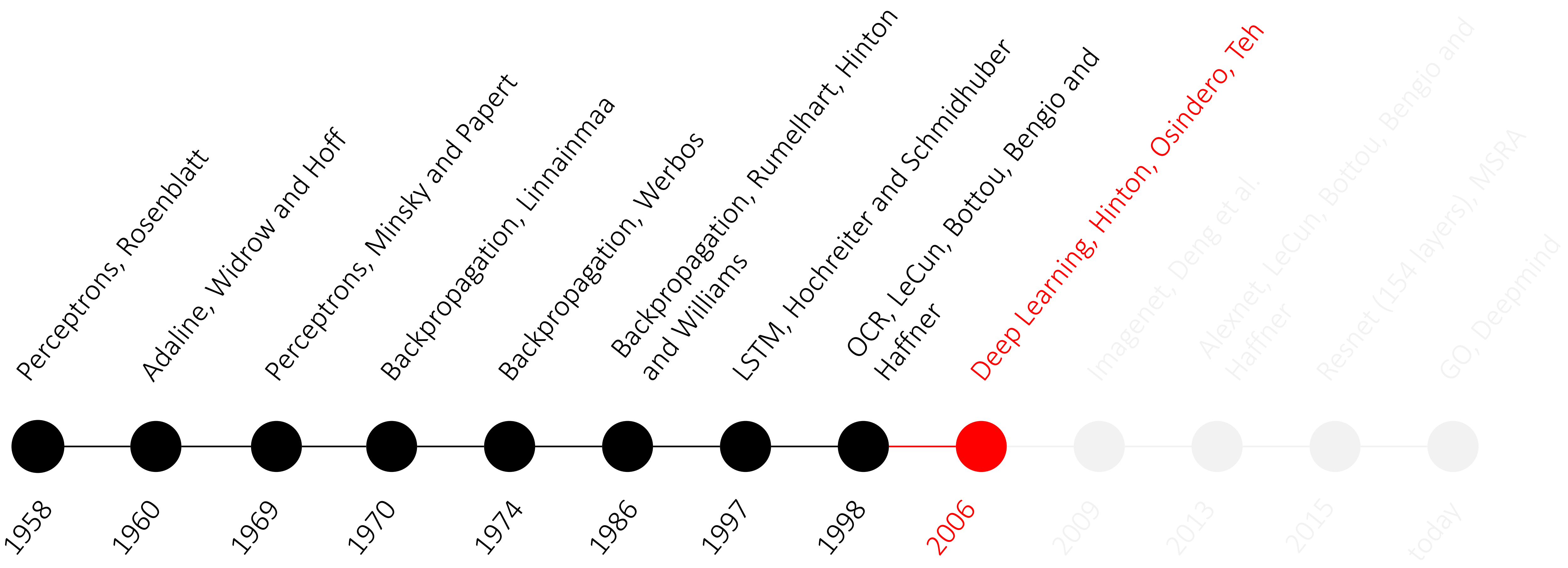
Recurrent networks

- Traditional networks are “too plain”
  - Static Input → Processing → Static Output
- What about sequences?
  - Temporal data, Language, Sequences
- What about feedback connections to the model?
- Memory is needed to “remember” state changes
  - Recurrent feedback connections
- What kind of memory
  - Long, Short?
  - Both! Long-short term memory networks (LSTM), Schmidhuber 1997

The second "AI winter"

- Until 1998 some nice algorithms and methods were proposed
  - Backpropagation
  - Recurrent Long-Short Term Memory Networks
  - OCR with Convolutional Neural Networks
- However, at the same time Kernel Machines (SVM etc.) and Graphical Models suddenly become very popular
  - Similar accuracies in the same tasks
  - Neural networks could not improve beyond a few layers
  - Kernel Machines included much fewer heuristics & nice proofs on generalization
- As a result, once again the AI community turns away from Neural Networks

The thaw of the “AI winter”



Neural Network and Deep Learning problems

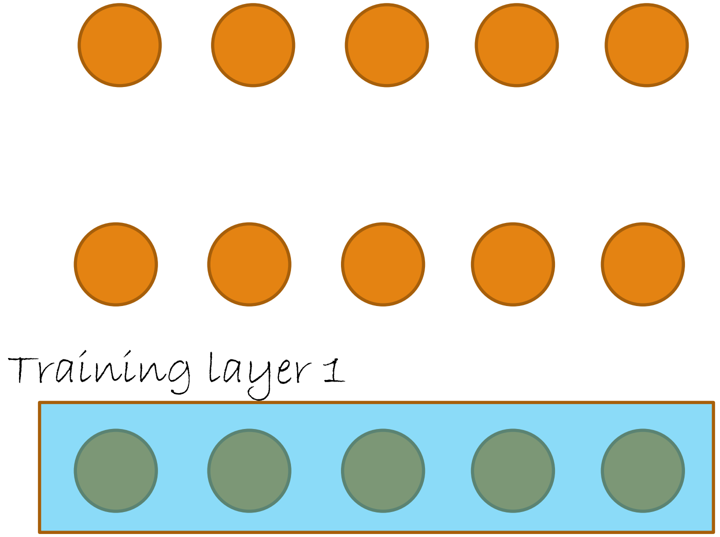
- Lack of processing power
  - No GPUs at the time
- Lack of data
  - No big, annotated datasets at the time
- Overfitting
  - Because of the above, models could not generalize all that well
- Vanishing gradient
  - While learning with NN, you need to multiply several numbers $a_1 \cdot a_2 \cdot \dots \cdot a_n$.
  - If all are equal to 0.1, for $n = 10$ the result is 0.0000000001, too small for any learning

Despite Backpropagation …

- Experimentally, training multi-layer perceptrons was not that useful
  - Accuracy didn't improve with more layers
- The inevitable question
  - Are 1-2 hidden layers the best neural networks can do?
  - Or is it that the learning algorithm is not really mature yet

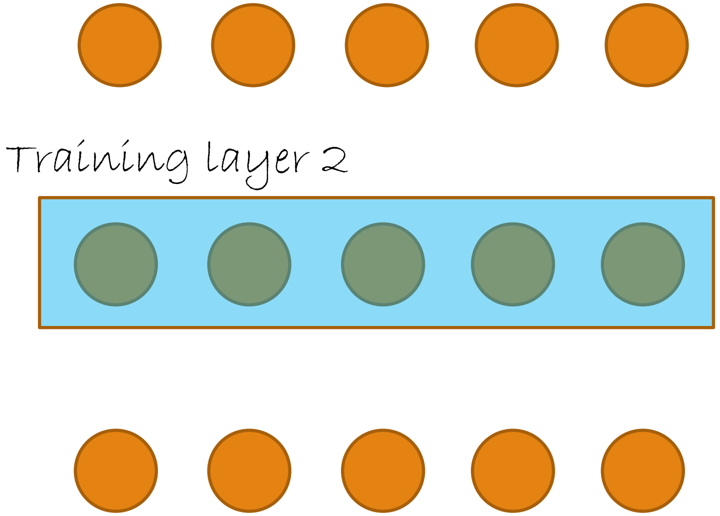
Deep Learning arrives

- Layer-by-layer training
  - The training of each layer individually is an easier undertaking
- Training multi-layered neural networks became easier
- Per-layer trained parameters initialize further training using contrastive divergence



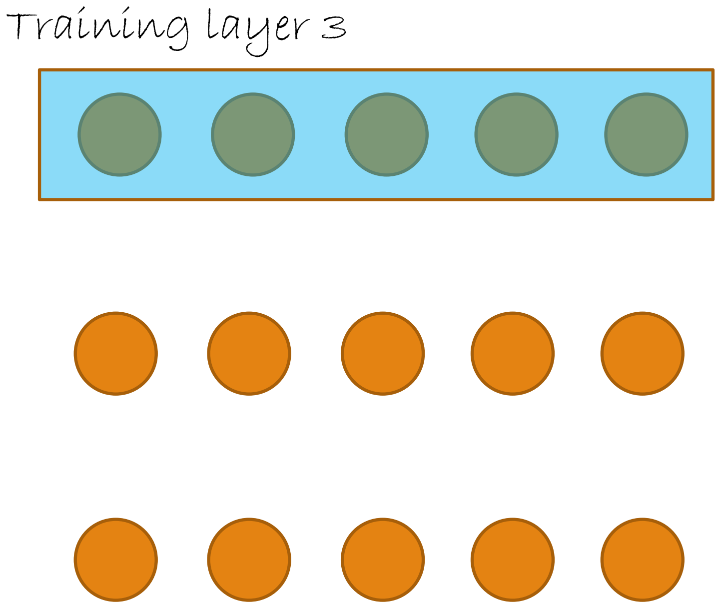
Deep Learning arrives

- Layer-by-layer training
  - The training of each layer individually is an easier undertaking
- Training multi-layered neural networks became easier
- Per-layer trained parameters initialize further training using contrastive divergence

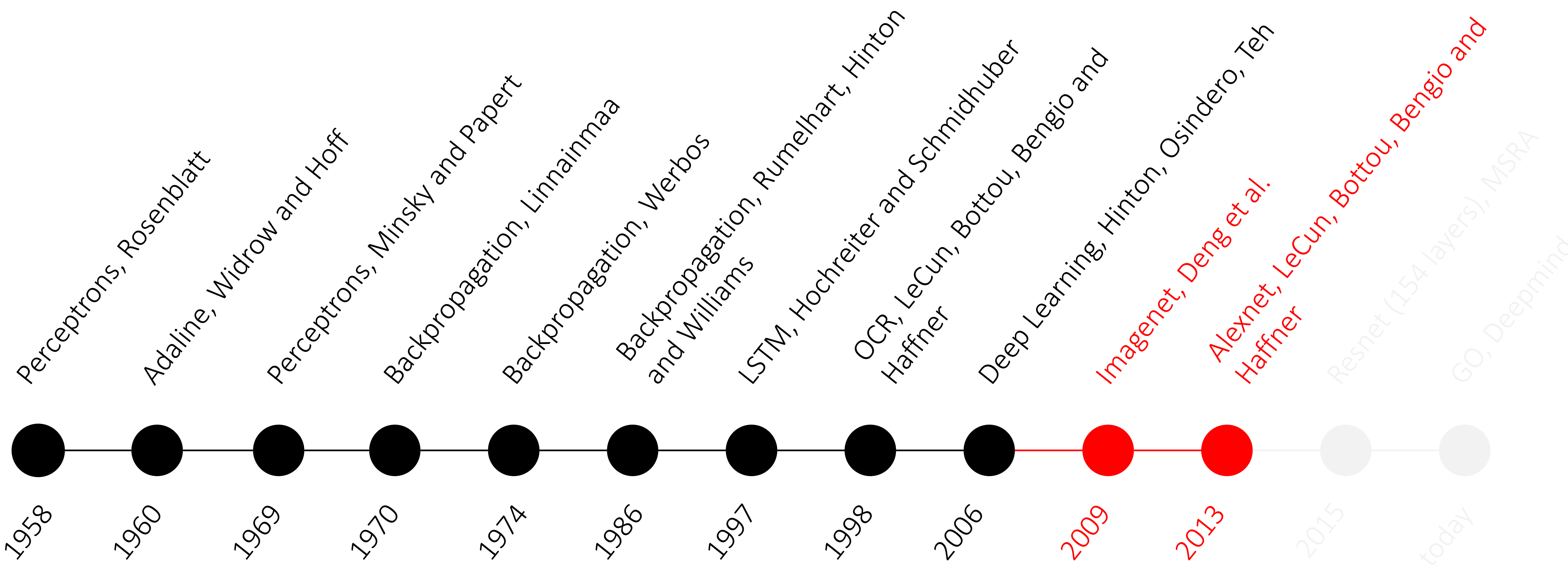


Deep Learning arrives

- Layer-by-layer training
  - The training of each layer individually is an easier undertaking
- Training multi-layered neural networks became easier
- Per-layer trained parameters initialize further training using contrastive divergence



Deep Learning Renaissance



More data, more …

- In 2009 the Imagenet dataset was published [Deng et al., 2009]
  - Collected images for each term of Wordnet (100,000 classes)
  - Tree of concepts organized hierarchically
    - “Ambulance”, “Dalmatian dog”, “Egyptian cat”, …
  - About 16 million images annotated by humans
- Imagenet Large Scale Visual Recognition Challenge (ILSVRC)
  - 1 million images
  - 1,000 classes
  - Top-5 and top-1 error measured

Alexnet

- In 2013 Krizhevsky, Sutskever and Hinton re-implemented [Krizhevsky2013] a convolutional neural network [LeCun1998]
  - Trained on Imagenet, Two GPUs were used for the implementation
- Further theoretical improvements
  - Rectified Linear Units (ReLU) instead of sigmoid or tanh
  - Dropout
  - Data augmentation
- In the 2013 Imagenet Workshop a legendary turmoil
  - Blasted competitors by an impressive 16% top-5 error, Second best around 26%
  - Most didn't even think of NN as remotely competitive
- At the same time similar results in the speech recognition community
  - One of G. Hinton students collaboration with Microsoft Research, improving state-of-the-art by an impressive amount after years of incremental improvements [Hinton2012]

Alexnet architecture

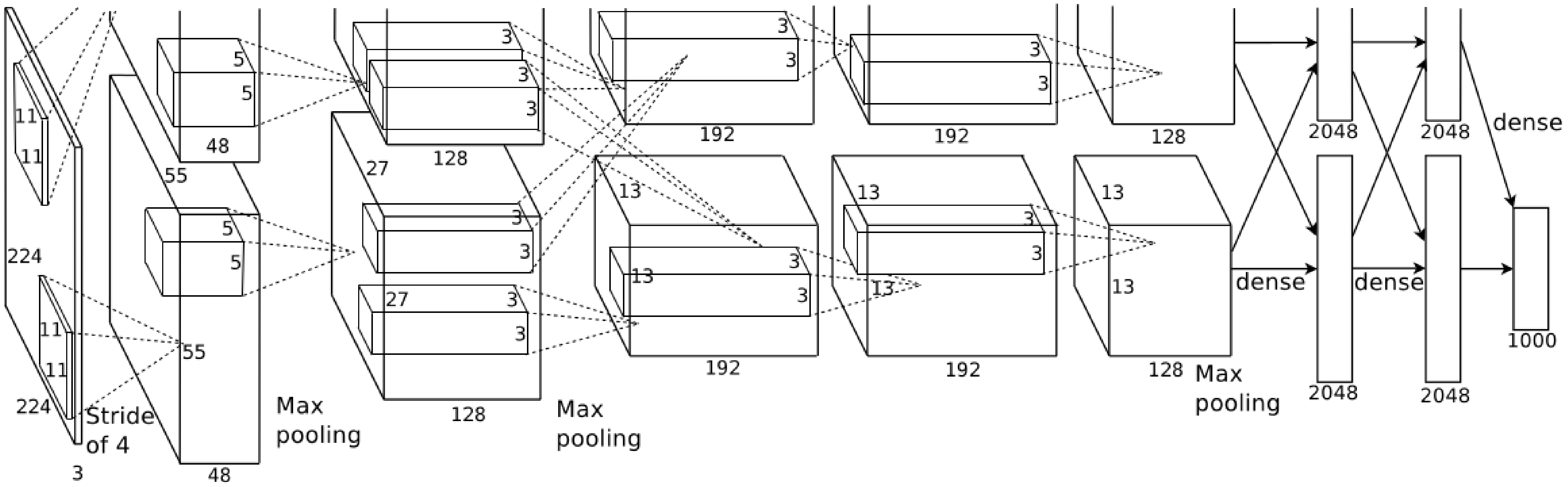


Figure 2: An illustration of the architecture of our CNN, explicitly showing the delineation of responsibilities between the two GPUs. One GPU runs the layer-parts at the top of the figure while the other runs the layer-parts at the bottom. The GPUs communicate only at certain layers. The network's input is 150,528-dimensional, and the number of neurons in the network's remaining layers is given by 253,440–186,624–64,896–64,896–43,264–4096–4096–1000.

Deep Learning Golden Era

- 1958 — Perceptrons, Rosenblatt
- 1960 — Adaline, Widrow and Hoff
- 1969 — Perceptrons, Minsky and Papert
- 1970 — Backpropagation, Linnainmaa
- 1974 — Backpropagation, Werbos
- 1986 — Backpropagation, Rumelhart, Hinton and Williams
- 1997 — LSTM, Hochreiter and Schmidhuber
- 1998 — OCR, LeCun, Bottou, Bengio and Haffner
- 2006 — Deep Learning, Hinton, Osindero, Teh
- 2009 — Imagenet, Deng et al.
- 2013 — Alexnet, LeCun, Bottou, Bengio and Haffner
- 2015 — Resnet (154 layers), MSRA
- today — GO, Deepmind

The today: applications

- Deep Learning is almost everywhere
  - Object classification, object detection, segmentation, pose estimation, image captioning, visual question answering, robotics
  - Machine translation, question answering, analogical reasoning
  - Speech recognition, speech synthesis
- Some strongholds
  - Theoretical underpinnings, causality, generalization, integrating external knowledge
  - Continuous learning (non static inputs/outputs), learning from few data
  - Reinforcement learning
  - Video analysis, action classification, compositionality

The ILSVC Challenge over the last three years

CNN based, non-CNN based

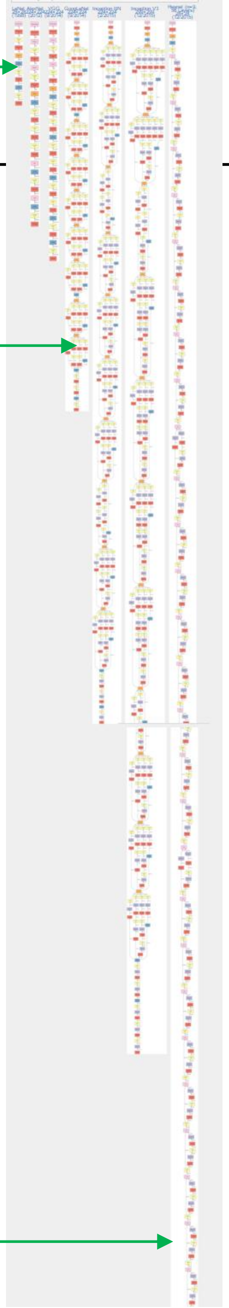
2012 Teams	%error
Supervision (Toronto)	15.3
ISI (Tokyo)	26.1
VGG (Oxford)	26.9
XRCE/INRIA	27.0
UvA (Amsterdam)	29.6
INRIA/LEAR	33.4

2013 Teams	%error
Clarifai (NYU spinoff)	11.7
NUS (singapore)	12.9
Zeiler-Fergus (NYU)	13.5
A. Howard	13.5
OverFeat (NYU)	14.1
UvA (Amsterdam)	14.2
Adobe	15.2
VGG (Oxford)	15.2
VGG (Oxford)	23.0

2014 Teams	%error
GoogLeNet	6.6
VGG (Oxford)	7.3
MSRA	8.0
A. Howard	8.1
DeeperVision	9.5
NUS-BST	9.7
TTIC-ECP	10.2
XYZ	11.2
UvA	12.1

Figures taken from Y. LeCun's CVPR 2015 plenary talk

2015 ILSVRC Challenge

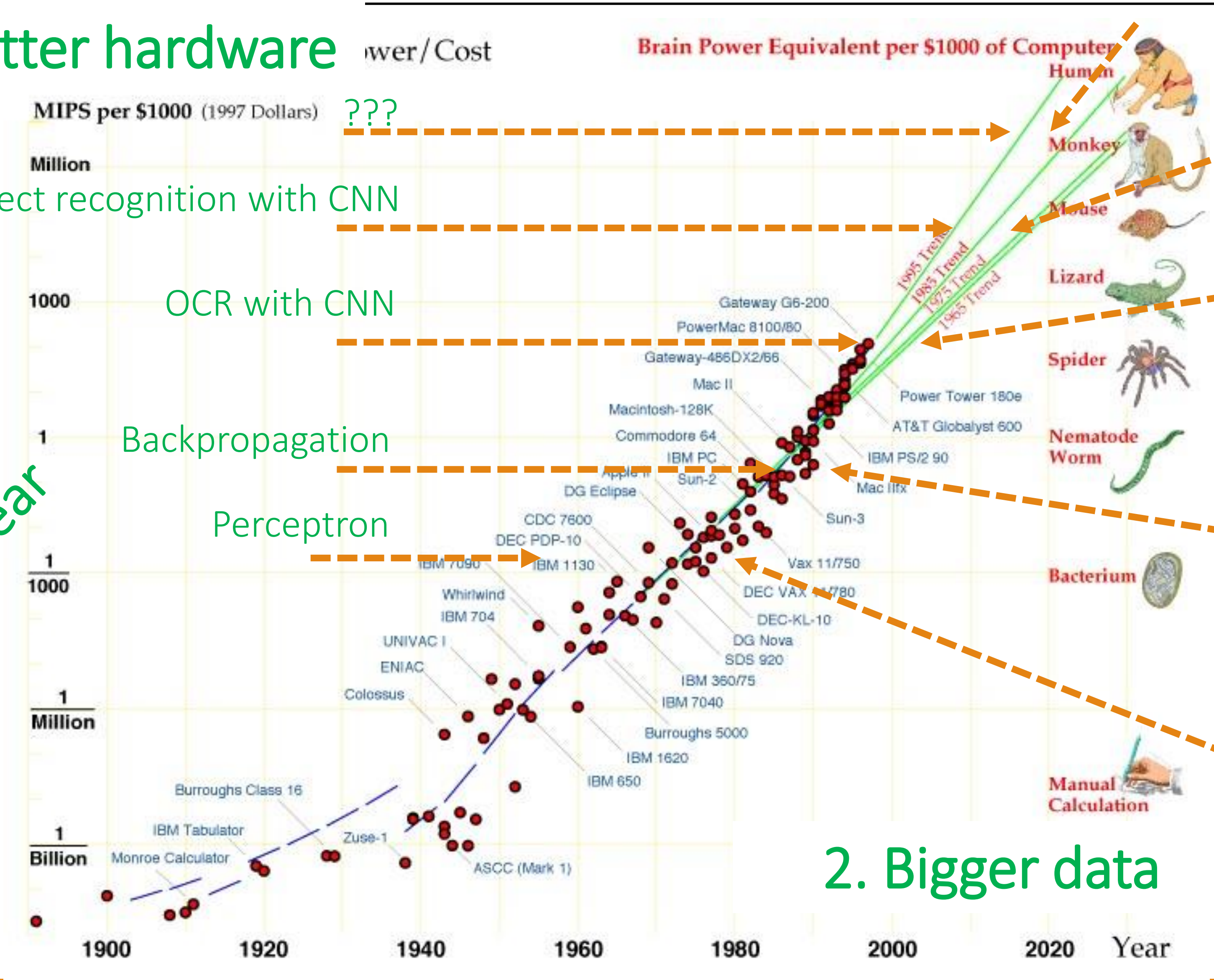


- Microsoft Research Asia won the competition with a legendary 150-layered network
- Almost superhuman accuracy: 3.5% error
  - In 2016 <3% error
- In comparison in 2014 GoogLeNet had 22 layers

So, why now?

Datasets of everything (captions, question-answering, …), reinforcement learning, ???

1. Better hardware



???

Object recognition with CNN

OCR with CNN

Backpropagation

Perceptron

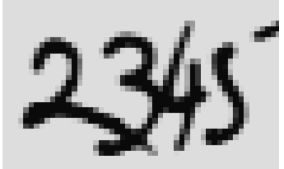
Loglinear

Results:

- Persian cat: 0.35211
- Egyptian cat: 0.23635
- hamster: 0.20282
- tiger cat: 0.05896
- lynx: 0.05759

Imagenet: 1,000 classes from real images, 1,000,000 images

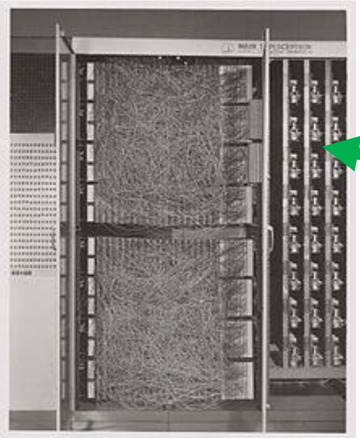
Bank cheques



Parity, negation problems

D1	D2	D3	Even-Parity
0	0	0	True
0	0	1	False
0	1	0	False
0	1	1	True
1	0	0	False
1	0	1	True
1	1	0	True
1	1	1	False

Mark I Perceptron



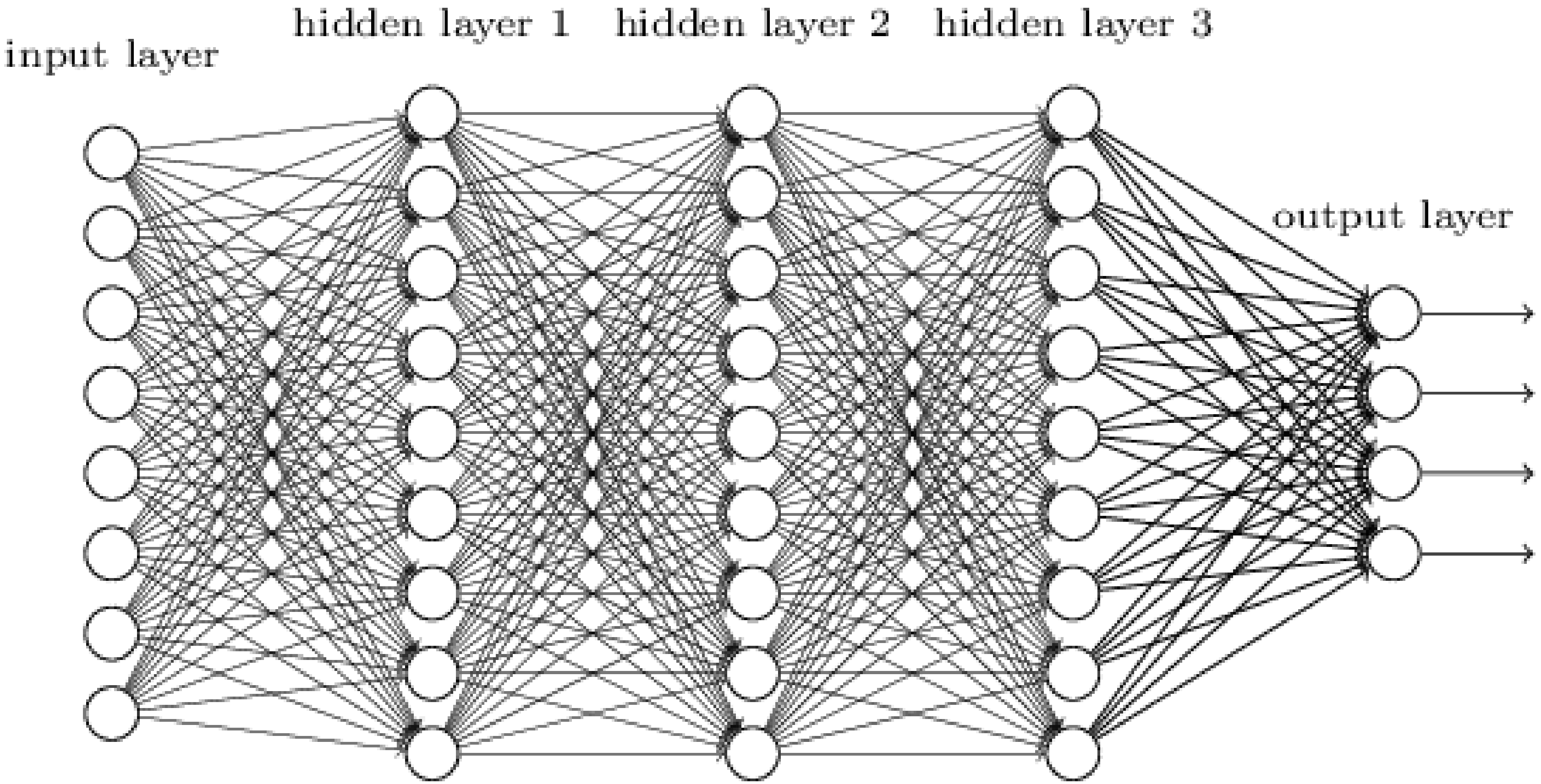
Potentiometers implement perceptron weights

2. Bigger data

So, why now? (2)

1. Better hardware
2. Bigger data
3. Better regularization methods, such as dropout
4. Better optimization methods, such as Adam, batch normalization

Deep Learning: The *What* and *Why*



Long story short

- A family of parametric, non-linear and hierarchical representation learning functions, which are massively optimized with stochastic gradient descent to encode domain knowledge, i.e. domain invariances, stationarity.
- $a_L(x; \theta_{1,\ldots,L}) = h_L\,(h_{L-1}(\ldots h_1(x, \theta_1), \theta_{L-1}),\ \theta_L)$
  - x:input, θ_l: parameters for layer l, $a_l = h_l(x, \theta_l)$: (non-)linear function
- Given training corpus $\{X, Y\}$ find optimal parameters

$$\theta^* \leftarrow \arg\min_{\theta} \sum_{(x,y)\subseteq(X,Y)} \ell(y, a_L(x; \theta_{1,\ldots,L}))$$

Learning Representations & Features

- Traditional pattern recognition

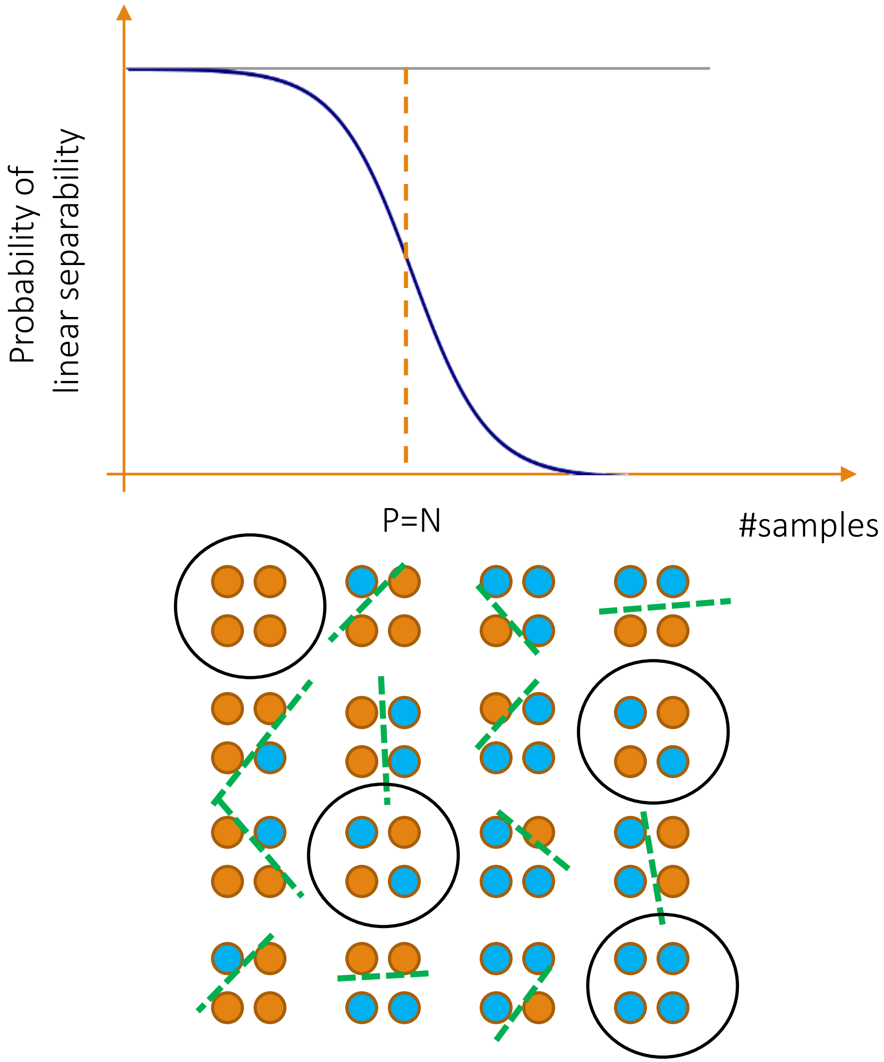
Hand-crafted Feature Extractor → Separate Trainable Classifier → “Lemur”

- End-to-end learning → Features are also learned from data

Trainable Feature Extractor ⇄ Trainable Classifier → “Lemur”

Non-separability of linear machines

- $X = \{x_1, x_2, \dots, x_n\} \in \mathcal{R}^d$
- Given the n points there are in total 2^n dichotomies
- Only about d are linearly separable
- With $n > d$ the probability X is linearly separable converges to 0 very fast
- The chances that a dichotomy is linearly separable is very small



How can we solve the non-separability of linear machines?

A. Apply SVM
B. Have non-linear features
C. Have non-linear kernels
D. Use advanced optimizers, like Adam or Nesterov's Momentum

The question will open when you start your session and slideshow.

shakespeak

Time: 60s

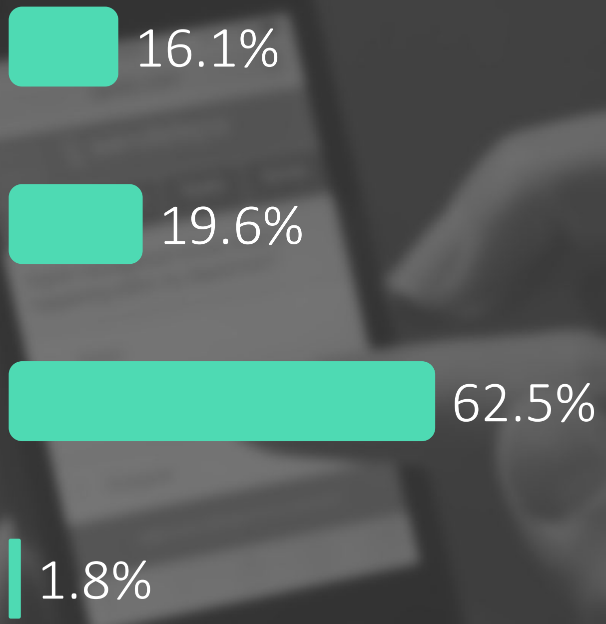
How can we solve the non-separability of linear machines?

A. Apply SVM

B. Have non-linear features

C. Have non-linear kernels

D. Use advanced optimizers, like Adam or Nesterov's Momentum



Closed

Non-linearizing linear machines

- Most data distributions and tasks are non-linear
- A linear assumption is often convenient, but not necessarily truthful
- **Problem:** How to get non-linear machines without too much effort?

Non-linearizing linear machines

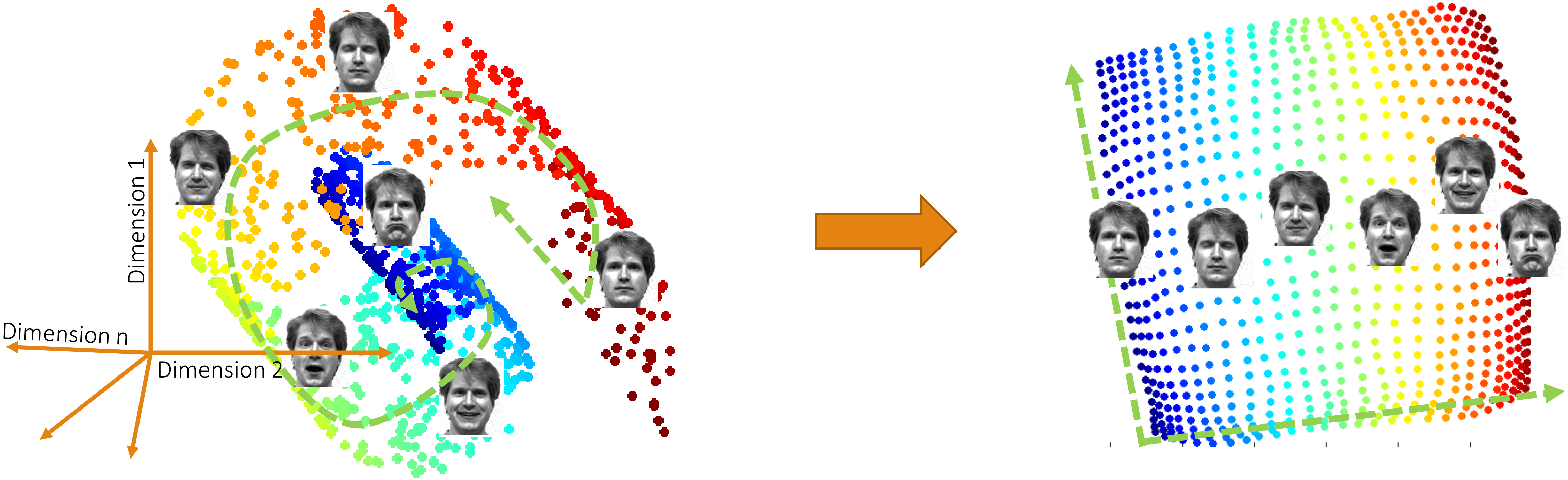
- Most data distributions and tasks are non-linear
- A linear assumption is often convenient, but not necessarily truthful
- **Problem:** How to get non-linear machines without too much effort?
- **Solution:** Make features non-linear
- What is a good non-linear feature?
  - Non-linear kernels, e.g., polynomial, RBF, etc
  - Explicit design of features (SIFT, HOG)?

Good features

- Invariant
  - But not too invariant
- Repeatable
  - But not bursty
- Discriminative
  - But not too class-specific
- Robust
  - But sensitive enough

Manifolds

- Raw data live in huge dimensionalities
- But, effectively lie in lower dimensional manifolds
- Can we discover this manifold to embed our data on?

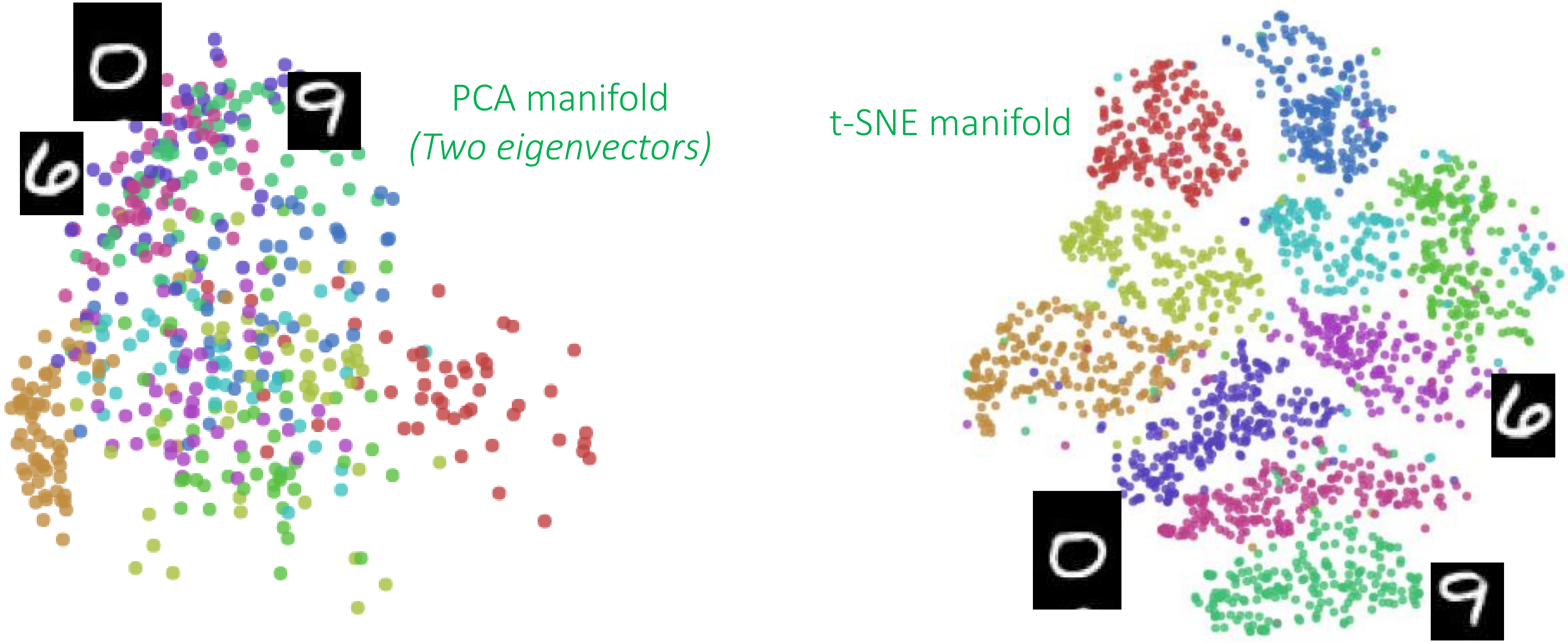


How to get good features?

- **Goal:** discover these lower dimensional manifolds
  - These manifolds are most probably highly non-linear
- **Hypothesis (1):** Compute the coordinates of the input (e.g. a face image) to this non-linear manifold → data become separable
- **Hypothesis (2):** Semantically similar things lie closer together than semantically dissimilar things

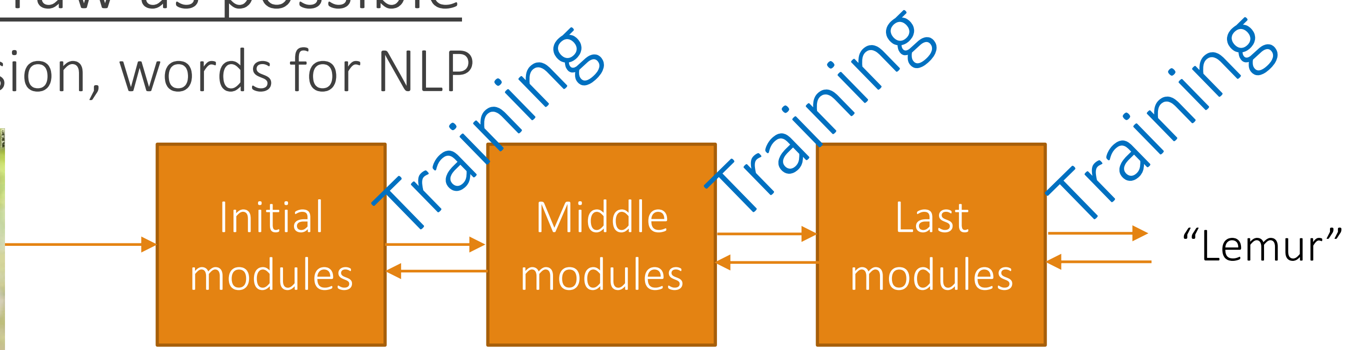
The digits manifolds

- There are good features and bad features
  good manifold representations and bad manifold representations
- 28 pixels x 28 pixels = 784 dimensions



End-to-end learning of feature hierarchies

- A pipeline of successive modules
- Each module's output is the input for the next module
- Modules produce features of higher and higher abstractions
  - Initial modules capture low-level features (e.g. edges or corners)
  - Middle modules capture mid-level features (e.g. circles, squares, textures)
  - Last modules capture high level, class specific features (e.g. face detector)
- Preferably, input as raw as possible
  - Pixels for computer vision, words for NLP



Why learn the features and not just design them?

A. Designing features manually is too time consuming and requires expert knowledge
B. Learned features give us a better understanding of the data
C. Learned features are more compact and specific for the task at hand
D. Learned features are easy to adapt
E. Features can be learnt in a plug-n-play fashion, ease for the layman

The question will open when you start your session and slideshow.

Time: 60s

Why learn the features and not just design them?

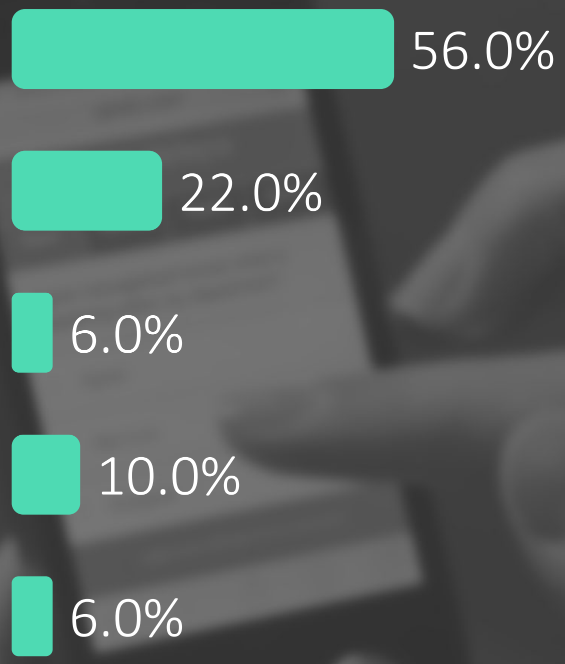
A. Designing features manually is too time consuming and requires expert knowledge

B. Learned features give us a better understanding of the data

C. Learned features are more compact and specific for the task at hand

D. Learned features are easy to adapt

E. Features can be learnt in a plug-n-play fashion, ease for the layman

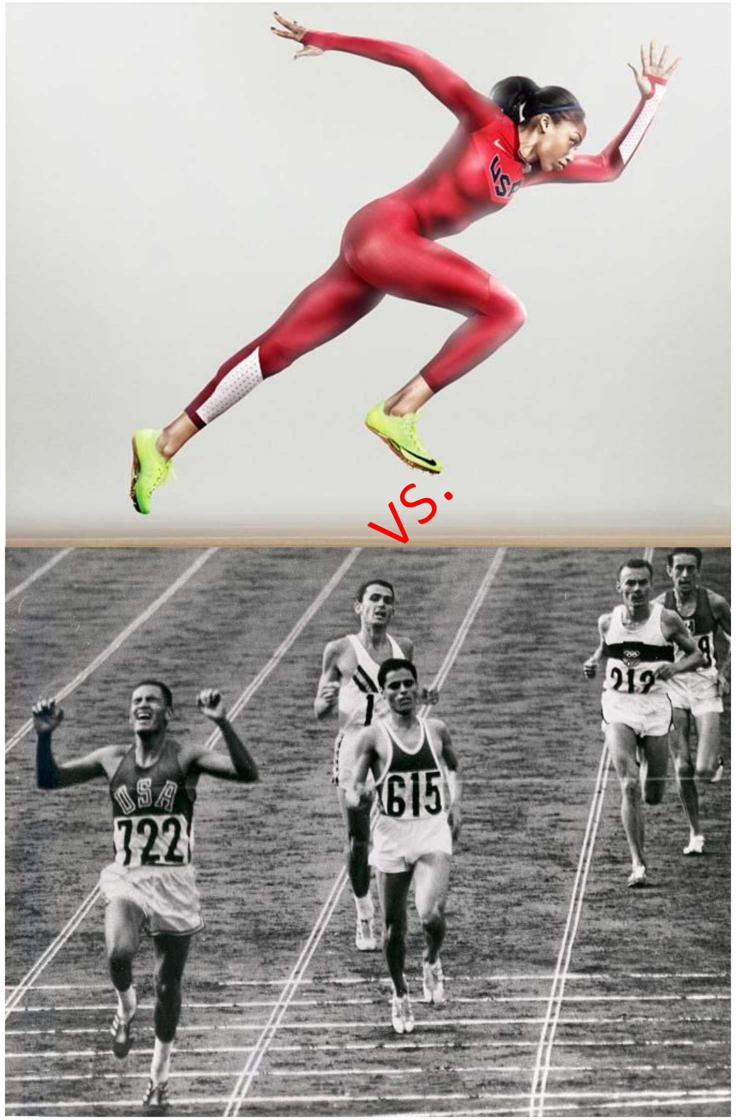


Closed

shakespeak

Why learn the features?

- Manually designed features
  - Often take a lot of time to come up with and implement
  - Often take a lot of time to validate
  - Often they are incomplete, as one cannot know if they are optimal for the task
- Learned features
  - Are easy to adapt
  - Very compact and specific to the task at hand
  - Given a basic architecture in mind, it is relatively easy and fast to optimize
- Time spent for designing features now spent for designing architectures



Types of learning

- Supervised learning
  - (Convolutional) neural networks

Is this a dog or a cat?

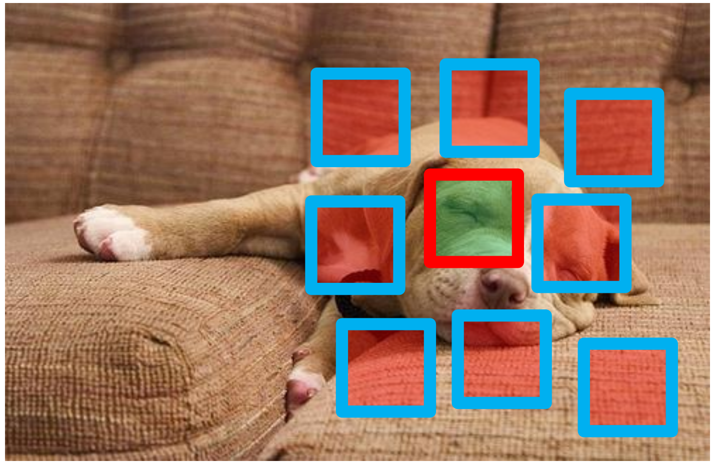
Types of learning

- Supervised learning
  - (Convolutional) neural networks
- Unsupervised learning
  - Autoencoders, layer-by-layer training

Reconstruct this image

Types of learning

- Supervised learning
  - (Convolutional) neural networks
- Unsupervised learning
  - Autoencoders, layer-by-layer training
- Self-supervised learning
  - A mix of supervised and unsupervised learning

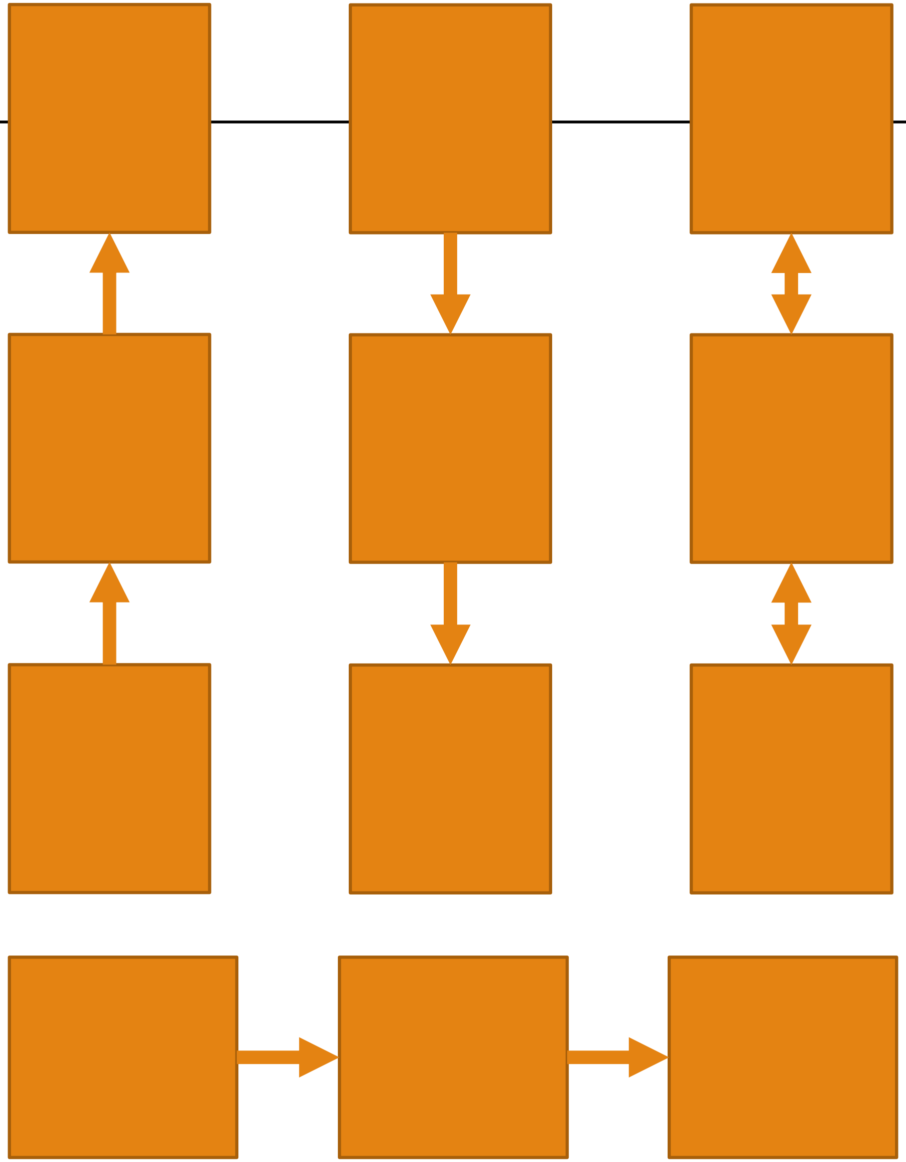


Types of learning

- Supervised learning
  - (Convolutional) neural networks
- Unsupervised learning
  - Autoencoders, layer-by-layer training
- Self-supervised learning
  - A mix of supervised and unsupervised learning
- Reinforcement learning
  - Learn from noisy, delayed rewards from your environment
  - Perform actions in your environment, so as to make decisions what data to collect

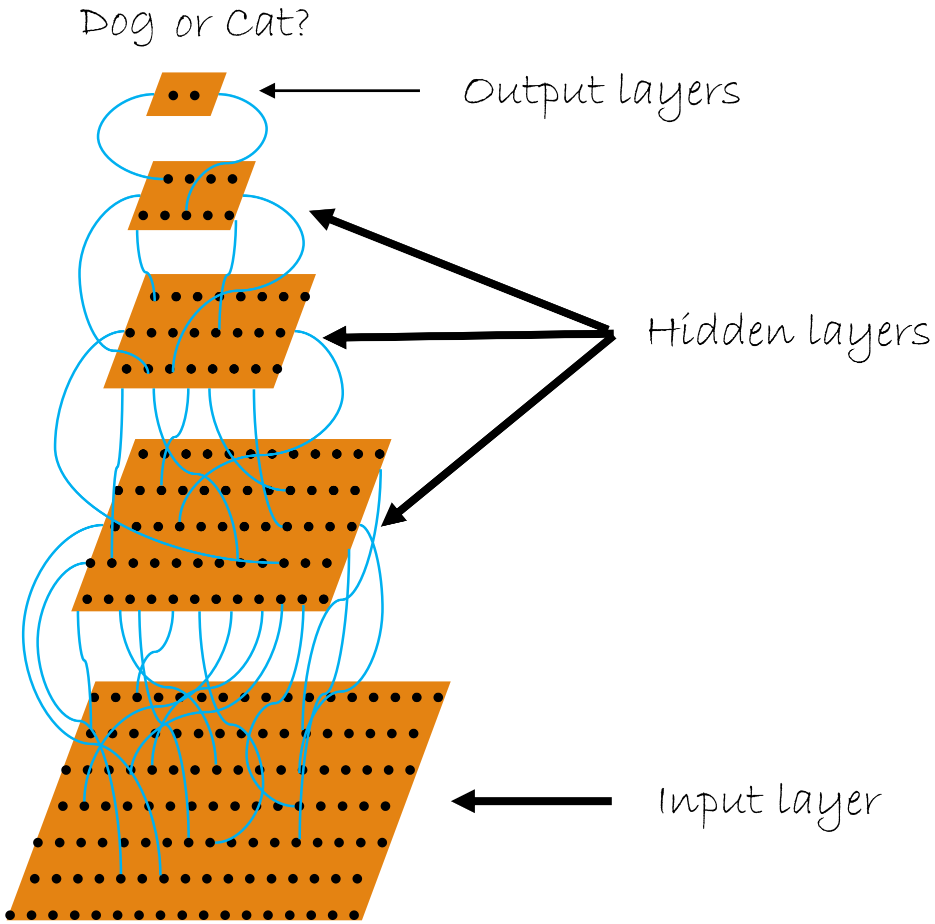
Deep architectures

- Feedforward
  - (Convolutional) neural networks
- Feedback
  - Deconvolutional networks
- Bi-directional
  - Deep Boltzmann Machines, stacked autoencoders
- Sequence based
  - RNNs, LSTMs

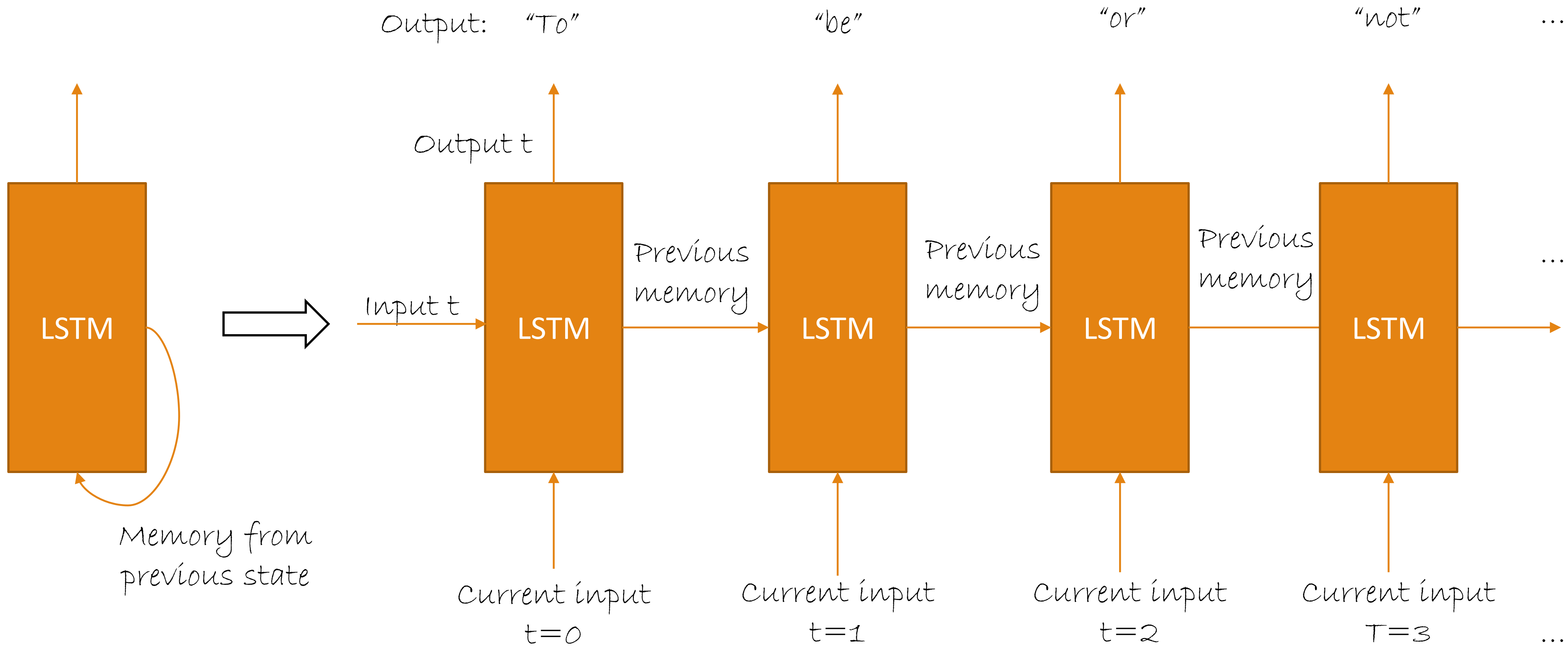


Convolutional networks in a nutshell

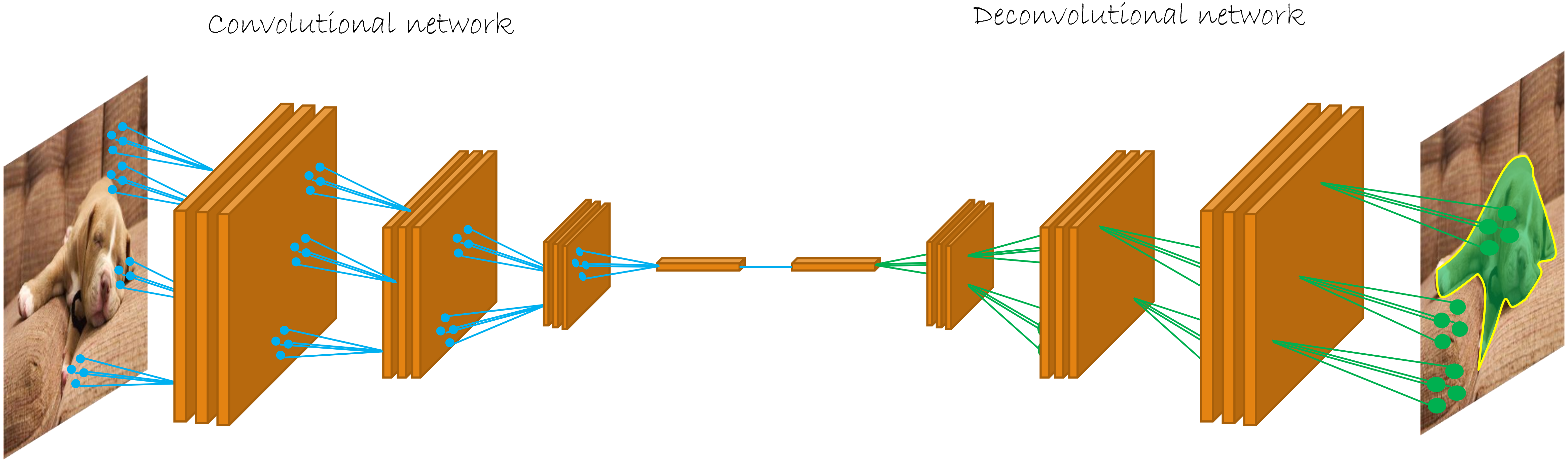
Is this a dog or a cat?



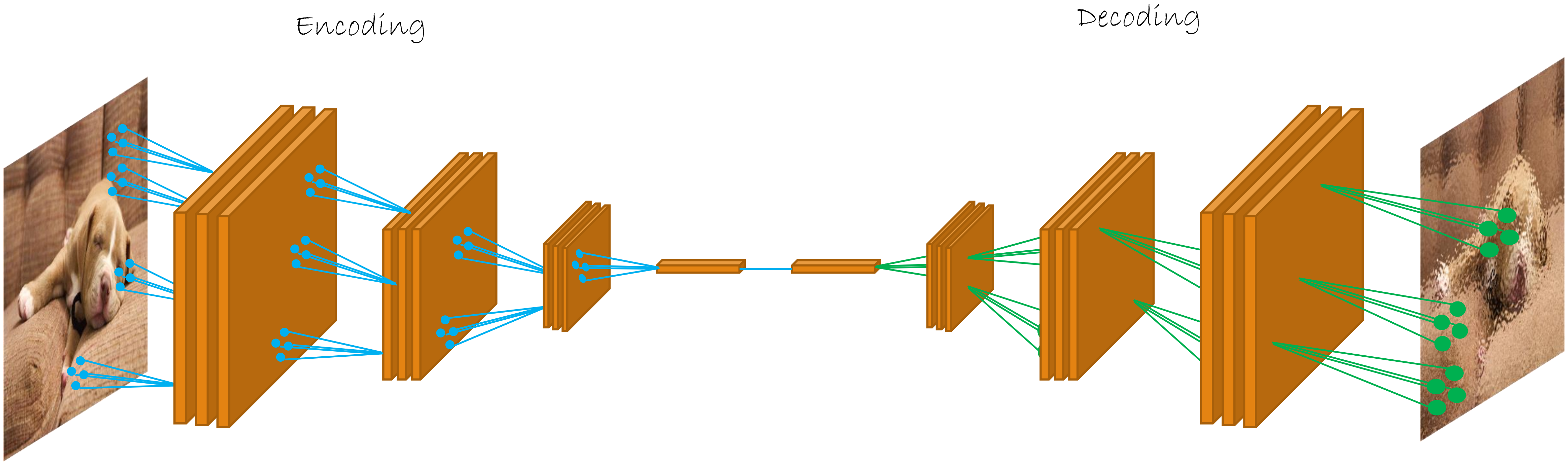
Recurrent networks in a nutshell



Deconvolutional networks



Autoencoders in a nutshell



Philosophy of the course

The bad news ☹

- We only have 2 months = 14 lectures
- Lots of material to cover
- Hence, no time to lose
  - Basic neural networks, learning Tensorflow, learning to program on a server, advanced optimization techniques, convolutional neural networks, recurrent neural networks, generative models
- This course is hard

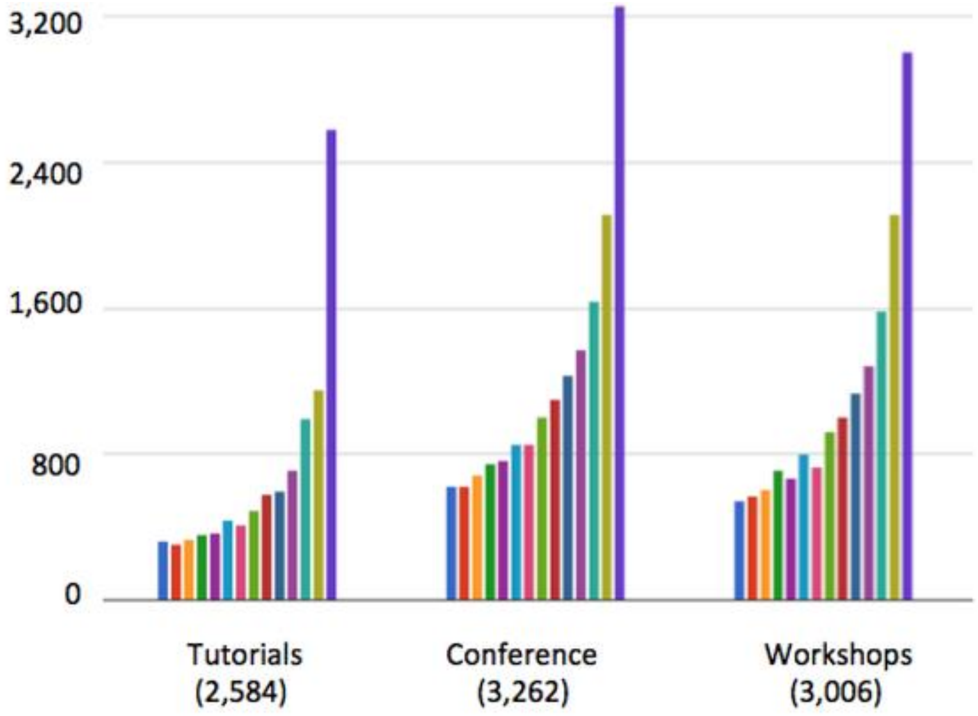
The good news ☺

- We are here to help
  - Kirill, Berkay, Peter and Tom have done some excellent work and we are all ready here to help you with the course
  - Last year we got a great evaluation score, so people like it and learn from it
- We have agreed with SURF SARA to give you access to the Dutch Supercomputer Cartesius with a bunch of (very) expensive GPUs
  - You should have no problem with resources
  - You get to know how it is to do real programming on a server
- You'll get to know some of the hottest stuff in AI today
- You'll get to present your own work to an interesting/ed crowd

The good news ☺

- You'll get to know some of the hottest stuff in AI today
  - in academia

NIPS



CVPR

The good news ☺

- You will get to know some of the hottest stuff in AI today
  - in academia & in industry

The even better news ☺☺☺

- In the end of the course we might give a few MSc Thesis Projects in collaboration with Qualcomm/QUVA Lab
  - Students will become interns in the QUVA lab and get paid during thesis
- Requirements
  - Work hard enough and be motivated
  - Have top performance in the class
  - And interested in working with us
- Come and find me later

Code of conduct

- We encourage you to help each other, to actively participate, give feedback etc
  - 3 students with **highest participation** in Q&A in Piazza get **+1 grade**
  - Your grade depends on what you do, not what others do
  - You have plenty of chances to collaborate for your poster and paper presentation
- However, we do not tolerate **blind** copy
  - Not from each other
  - Not from the internet
  - We have (deep) ways to check that

First lab assignment

Deep Learning Framework

- Tensorflow [https://www.tensorflow.org/]
  - Python, very good documentation, lots of code (get inspired but not blindly copy)
- Learning Goals
  - Multi-layer Perceptrons, Convolutional Neural Networks, make you first neural classifier
  - Solve a neural network in pen and paper
  - Basic hyper-parameter tuning
- Deadline: **14th of November, 23:59**
  - Late submissions: -1 point per day
- Make sure you send your submissior <u>in the correct format</u>
  - We have more than 100 students.
    If the code is not immediately runnable, you will not get a grade.

Submission

Create ZIP archive with the following structure:

```
lastname_assignment_3.zip
|   report_lastname.pdf
|   mlp_numpy.py
|   mlp_tf.py
|   convnet_tf.py
|   train_mlp_numpy.py
|   train_mlp_tf.py
|   train_convnet_tf.py
```

Replace lastname with your last name. After you create the zip file, please send it to uva.deeplearning@gmail.com with subject line Assignment 1: <firstname> <lastname> . Please make sure to send your submission to the correct e-mail address and do **NOT** use Blackboard for submissions. We cannot guarantee a grade for the assignment if the deliverables are not handed in according to these instructions.

⚠ The deadline for the assignment is the **14th of November, 23:59** ⚠

Programming in a super-computer

- Use the servers from SURF-SARA for your assignments
- You can also use them for your workshop presentation
- Probably, I will arrange 1 hour presentation from SURF SARA on how to use their facilities

Summary

- A brief history of neural networks and deep learning
- What is deep learning and why is it happening now?
- What types of deep learning exist?
- Demos and tasks where deep learning is currently the preferred choice of models

Reading material & references

- http://www.deeplearningbook.org/
- Chapter 1: Introduction, p.1-28

Next lecture

- Neural networks as layers and modules
- Build your own modules
- Theory of neural networks