Deep Learning Theory and Applications

Yale

CPSC/AMTH 663

Outline

1. Course logistics
2. What is Deep Learning?
3. Deep learning examples
   - CNNs
   - Word embeddings
   - RNNs
   - Autoencoders
   - Ultra deep learning (ResNet)
   - Generative models (e.g. GANs)
   - Deep reinforcement learning

Course Logistics

- Textbooks (available online)
  - Neural Networks and Deep Learning by Michael Nielsen
  - Deep Learning by Goodfellow, Bengio, and Courville
- Required background
  - Basic probability
  - Basic linear algebra & calculus
  - Programming experience
    - Python and Tensorflow will be used in this course
  - Look at the textbooks and HW 1 for an idea
- Canvas
  - Lectures posted after class
  - Announcements & HW

Course Logistics

- Office hours:
  - **Instructor Office Hours:** Tuesdays and Thursdays 2:00-3:30 pm AKW 104
  - **TA Office Hours:** Mondays and Wednesdays 12:00-1:30 pm AKW 407
- 5-6 HW assignments
  - Assigned about every 2 weeks, due on Thursdays
  - All/most will include some programming (Python & Tensorflow)
- Final project (details forthcoming)
  - In groups of 3-4

Goals of the Course

- A solid understanding of supervised feedforward neural networks
  - Stochastic gradient descent, backpropagation, etc.
  - Cost functions, regularizers, etc.
- The ability to design and train novel architectures
- An understanding of optimization strategies in training deep architectures
- Understanding of important deep architectures (e.g. CNN, RNN, autoencoders, GANs, deep reinforcement learning)

What is deep learning?

- Class of machine learning algorithms that use multiple levels of non-linear processing units in order to perform supervised or unsupervised analysis of data
  - Different levels contain different features of the data
  - Different levels contain different representations of the same data
- In this class we will learn only about the large class of deep artificial neural networks, loosely inspired by computation in the brain
  - Deep belief networks, logic circuits are also “deep learning”

Machine Learning

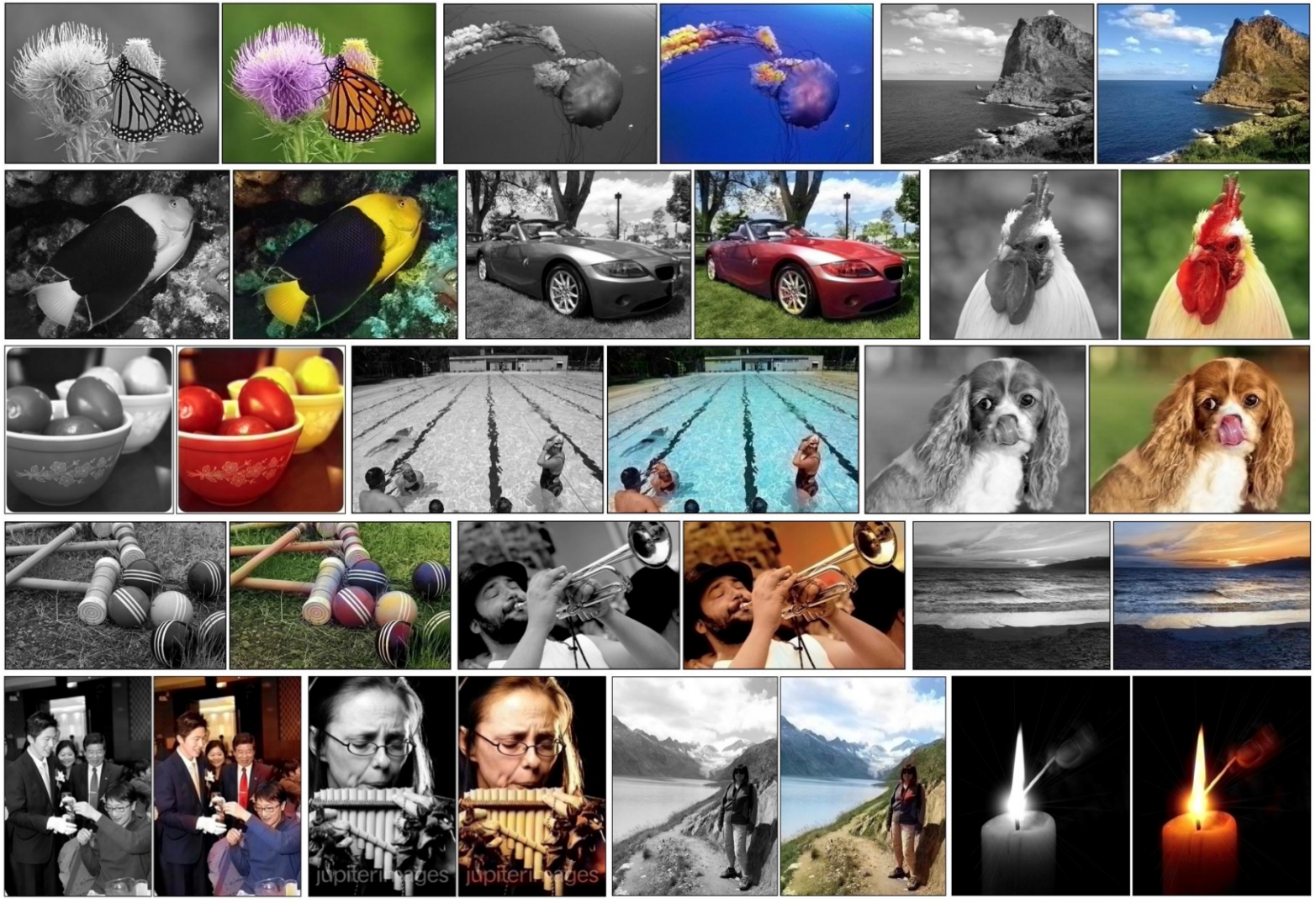
- Field of study that gives computers the ability to learn without being explicitly programmed
- Usually parameters of a selected models are fitted via optimization
- Model is selected to be not too complex and not too simple (overfit vs underfit)

Artificial Neural Networks

- A computing system made up of simple, interconnected processing elements that process information by their dynamic state response to external inputs
- Loosely inspired by neural computation, hence processing elements are called “neurons”
- Trained with instead of programmed
- Capacity to encode vast array of functions and learn makes them capable of learning complicated functions of inputs amenable to various tasks

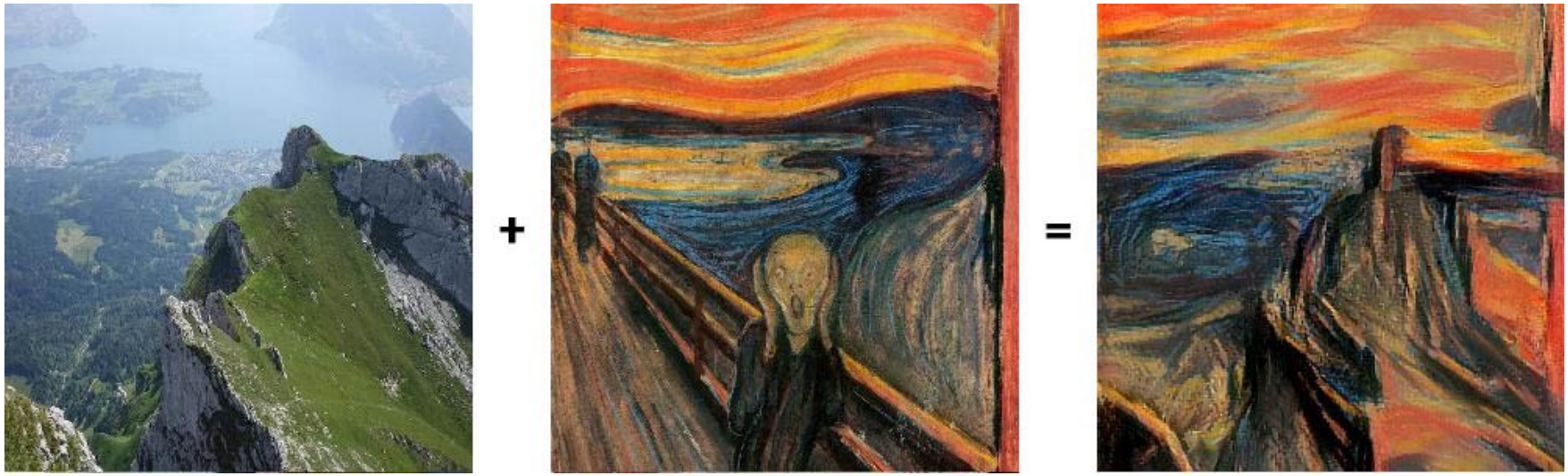
Image Colorization

(Zhang et al., 2016)

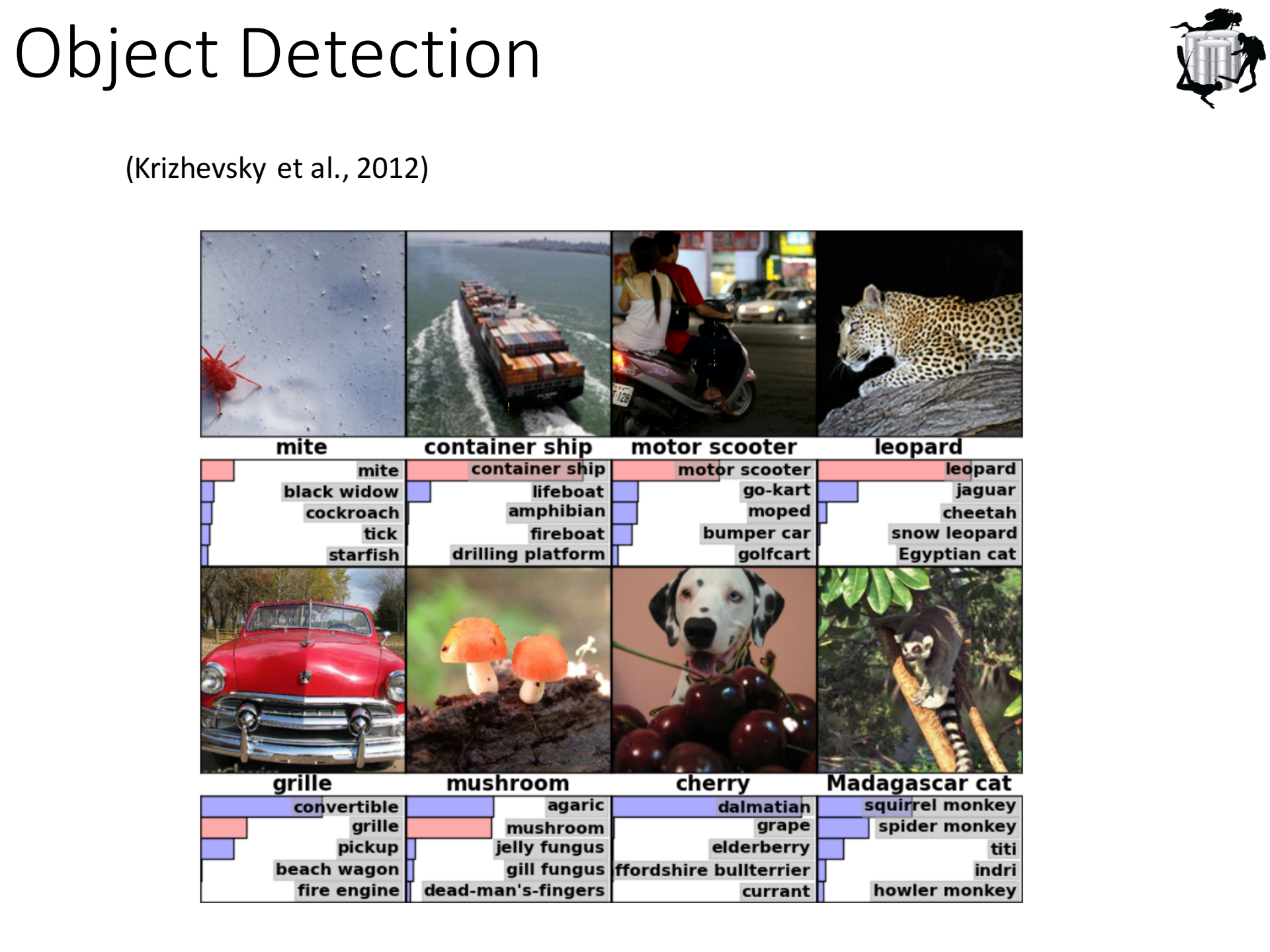


Style Transfer

(Gatys et al., 2015)



Object Detection



(Krizhevsky et al., 2012)

Text Generation

(Andrej Karpathy blog, 2015)

```
PANDARUS:
Alas, I think he shall be come approached and the day
When little srain would be attain'd into being never fed,
And who is but a chain and subjects of his death,
I should not sleep.

Second Senator:
They are away this miseries, produced upon my soul,
Breaking and strongly should be buried, when I perish
The earth and thoughts of many states.

DUKE VINCENTIO:
Well, your wit is in the care of side and that.

Second Lord:
They would be ruled after this chamber, and
my fair nues begun out of the fact, to be conveyed,
Whose noble souls I'll have the heart of the wars.

Clown:
Come, sir, I will make did behold your worship.

VIOLA:
I'll drink it.
```

Image Captioning

(Karpathy & Fei-Fei, 2015)

"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."

"girl in pink dress is jumping in air."

"black and white dog jumps over bar."

"young girl in pink shirt is swinging on swing."

"man in blue wetsuit is surfing on wave."

"little girl is eating piece of cake."

"baseball player is throwing ball in game."

"woman is holding bunch of bananas."

"black cat is sitting on top of suitcase."

Deep Reinforcement

Alpha Go Zero

nature
International journal of science

Access provided by Yale University

Altmetric: 2188 Citations: 1 More detail »

Article

Mastering the game of Go without human knowledge

David Silver ✉, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, Yutian Chen, Timothy Lillicrap, Fan Hui, Laurent Sifre, George van den Driessche, Thore Graepel & Demis Hassabis

Nature **550**, 354–359 (19 October 2017)
doi:10.1038/nature24270
Download Citation

Computational science
Computer science Reward

Received: 07 April 2017
Accepted: 13 September 2017
Published online: 18 October 2017

Alpha Zero

arXiv.org > cs > arXiv:1712.01815

Search or
(Help | Advanc

Computer Science > Artificial Intelligence

Mastering Chess and Shogi by Self-Play with a General Reinforcement Learning Algorithm

David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, Timothy Lillicrap, Karen Simonyan, Demis Hassabis

(Submitted on 5 Dec 2017)

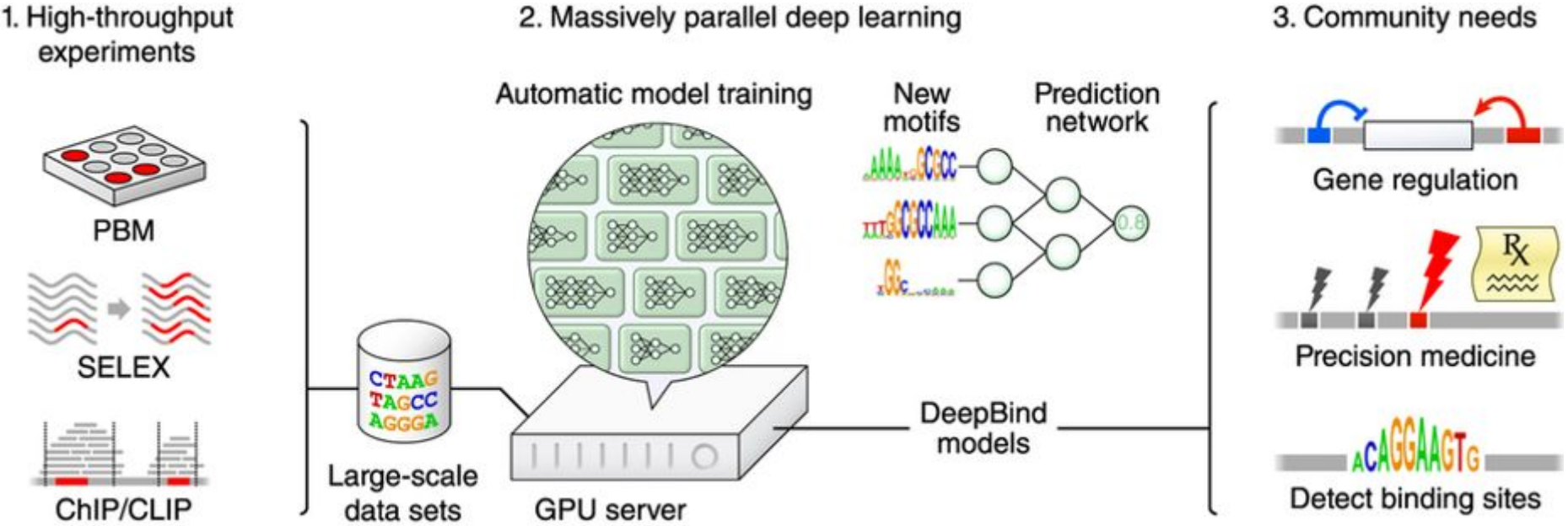
The game of chess is the most widely-studied domain in the history of artificial intelligence. The strongest programs are based on a combination of sophisticated search techniques, domain-specific adaptations, and handcrafted evaluation functions that have been refined by human experts over several decades. In contrast, the AlphaGo Zero program recently achieved superhuman performance in the game of Go, by tabula rasa reinforcement learning from games of self-play. In this paper, we generalise this approach into a single AlphaZero algorithm that can achieve, tabula rasa, superhuman performance in many challenging domains. Starting from random play, and given no domain knowledge except the game rules, AlphaZero achieved within 24 hours a superhuman level of play in the games of chess and shogi (Japanese chess) as well as Go, and convincingly defeated a world-champion program in each case.

Subjects: **Artificial Intelligence (cs.AI)**; Learning (cs.LG)
Cite as: arXiv:1712.01815 **[cs.AI]**
(or arXiv:1712.01815v1 **[cs.AI]** for this version)

Submission history
From: David Silver [view email]
[v1] Tue, 5 Dec 2017 18:45:38 GMT (272kb,D)

Genomics: Enhancer Binding

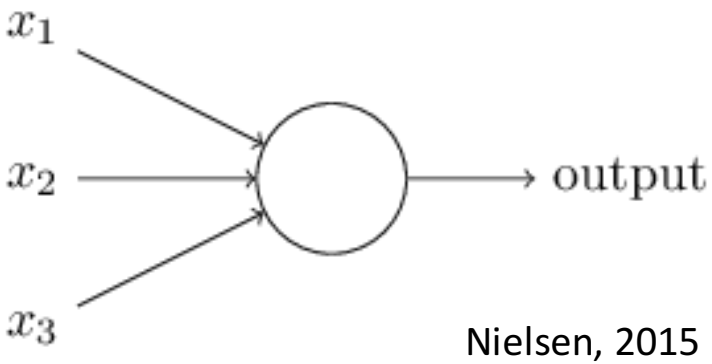
(Alipanahi et al. 2015)



What is a neural network?

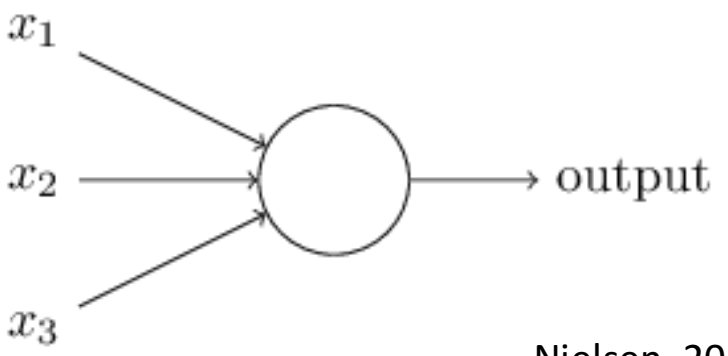
The perceptron

- Developed in 1950's and 1960's by Frank Rosenblatt
- Binary inputs
- Single binary output
- Example:



Nielsen, 2015

The perceptron

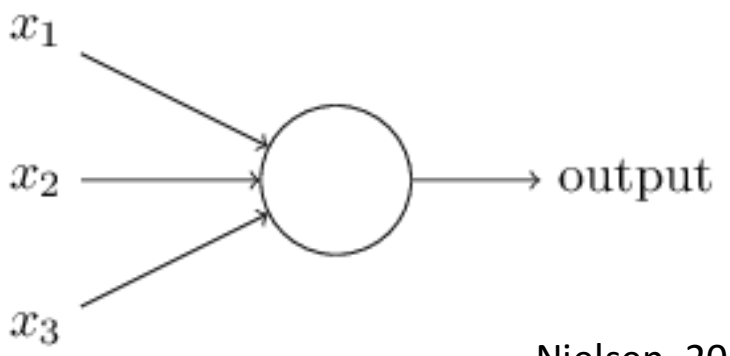


Nielsen, 2015

- Computing the output:
  - Assign weights to each input
  - Determine if weighted sum of inputs is greater than some threshold

$$\text{output} = \begin{cases} 0 & \text{if } \sum_j w_j x_j \leq \text{ threshold} \\ 1 & \text{if } \sum_j w_j x_j > \text{ threshold} \end{cases}$$

The perceptron



Nielsen, 2015

- Example: Decide whether to attend a cheese festival
- Three factors:
  1. Is the weather good? x_1
  2. Does your friend want to accompany you? x_2
  3. Is the festival near public transit? (you don't own a car) x_3

$$x_j = \begin{cases} 0, \text{if no} \\ 1, \text{if yes} \end{cases}$$

The perceptron

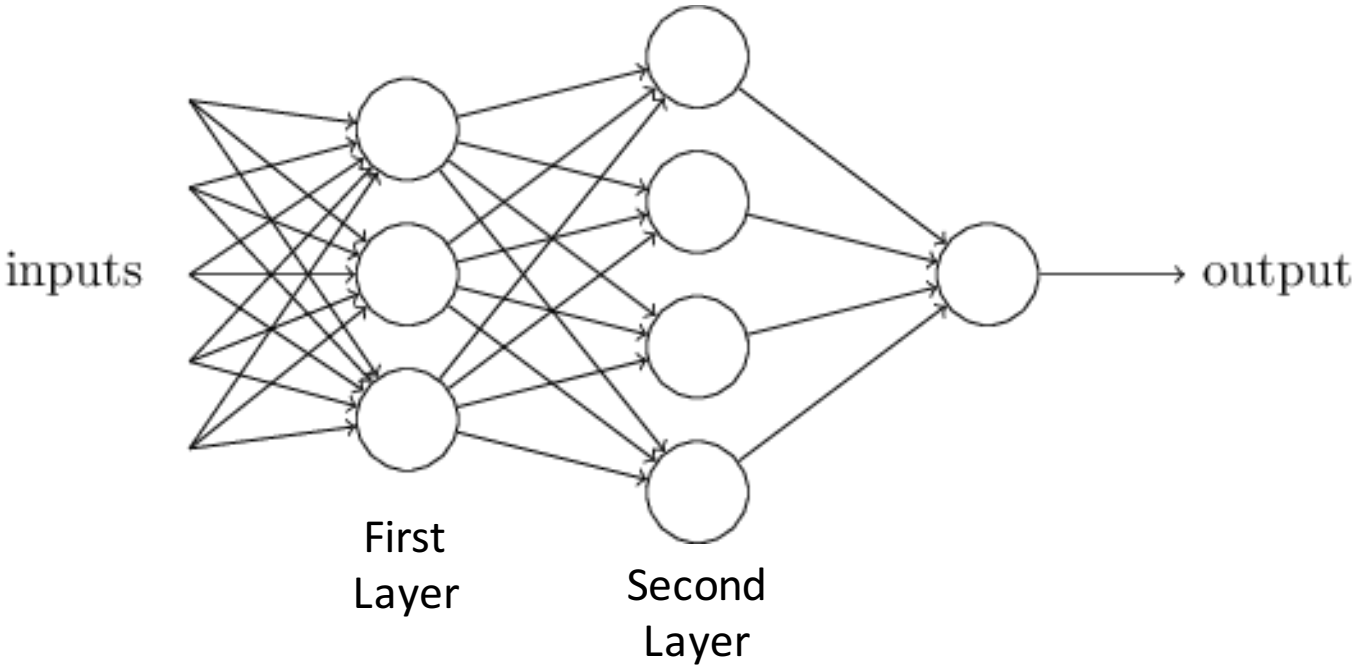
- Example: Decide whether to attend a cheese festival
- Three factors:
  1. Is the weather good? x_1
  2. Does your friend want to accompany you? x_2
  3. Is the festival near public transit? (you don't own a car) x_3
- **Case 1**: Love cheese but hate bad weather
  - $w_1 = 6$
  - $w_2 = 2$
  - $w_3 = 2$
  - Threshold$= 5$
  - $\sum_j w_j x_j >$ threshold whenever weather is **good** ($x_1 = 1$**)**
  - $\sum_j w_j x_j <$ threshold whenever weather is **bad** ($x_1 = 0$**)**

The perceptron

- Example: Decide whether to attend a cheese festival
- Three factors:
  1. Is the weather good? x_1
  2. Does your friend want to accompany you? x_2
  3. Is the festival near public transit? (you don't own a car) x_3
- **Case 2**: Love cheese but don't hate bad weather as much
  - $w_1 = 6$
  - $w_2 = 2$
  - $w_3 = 2$
  - Threshold$= 3$
  - $\sum_j w_j x_j >$ threshold whenever weather is **good** ($x_1 = 1$) *or* friend will go ($x_2 = 1$) *and* when the festival is near public transit ($x_3 = 1$)

The multilayer perceptron (MLP)

- A single perceptron is pretty simple
- A complex network of perceptrons can make subtle decisions



Nielsen, 2015

Notation Simplification

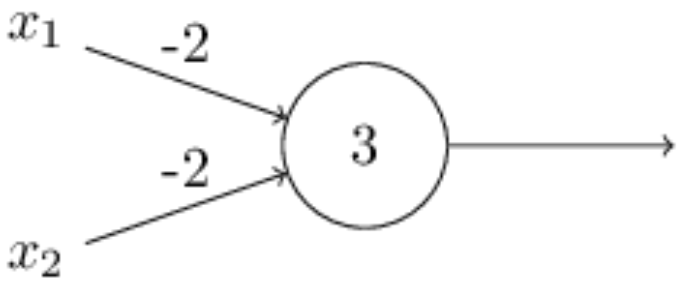
- $w \cdot x = \sum_j w_j x_j$
  - w and x are the weight and input vectors, respectively
- Replace the threshold with perceptron *bias*
  - Bias $b = -\text{threshold}$

$$\text{output} = \begin{cases} 0 & \text{if } w \cdot x + b \leq 0 \\ 1 & \text{if } w \cdot x + b > 0 \end{cases}$$

- Bias is a measure of ease in *firing* the perceptron

Logic circuits with perceptrons

- $w_1, w_2 = -2, b = 3$

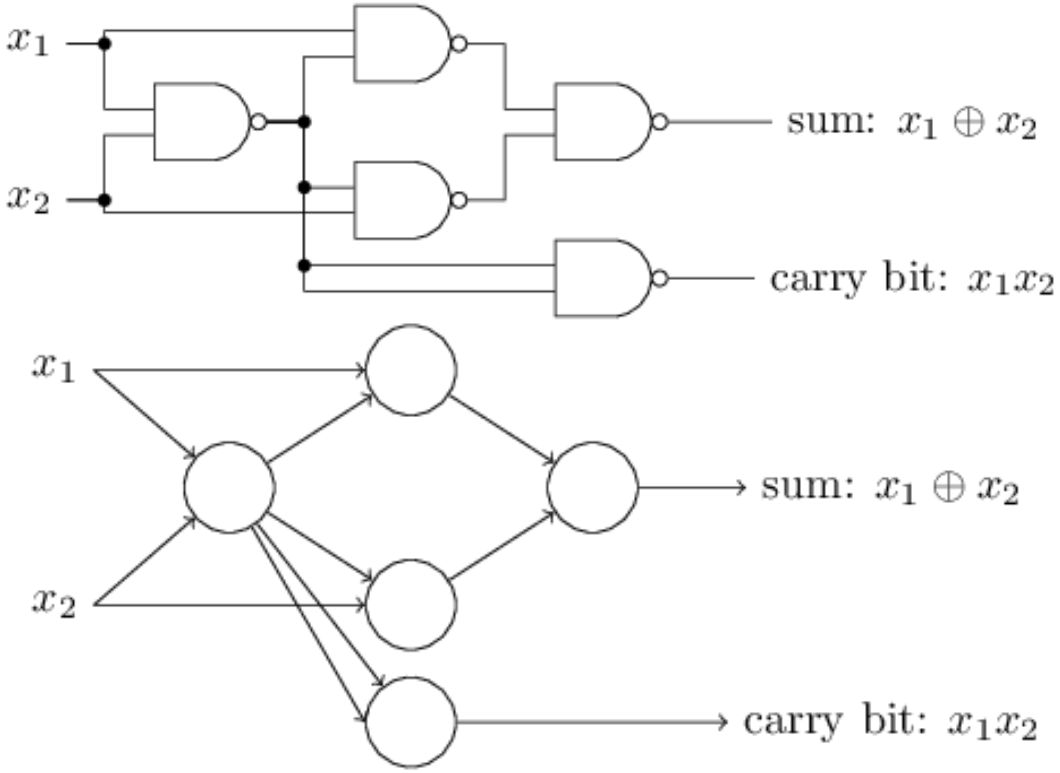


Nielsen, 2015

- What is the output of this perceptron for each possible input?
- What logic circuit is this?
- Input 00 produces 1
- Input 01 or 10 produce 1
- Input 11 produces 0
- This is a NAND gate!

Logic circuits with perceptrons

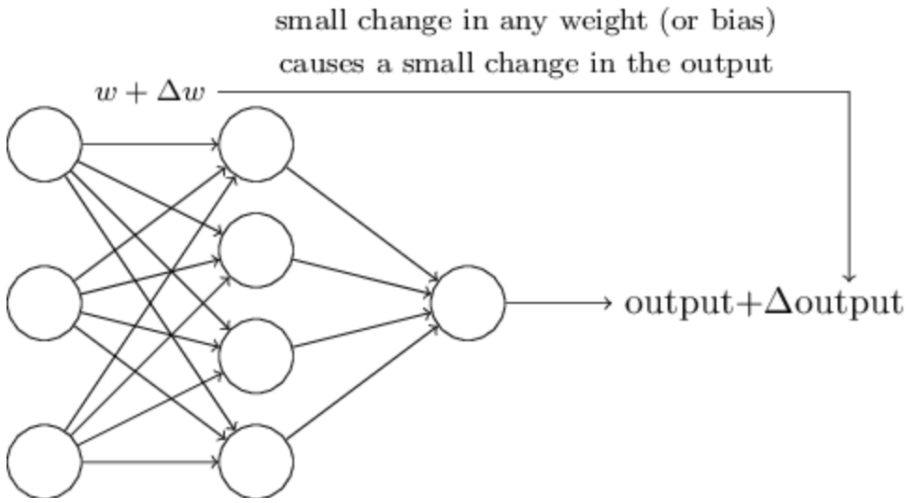
- NAND gates are universal for computation
  - Any computation can be built from NAND gates
  - Therefore, perceptrons are universal for computation
- Bitwise addition:



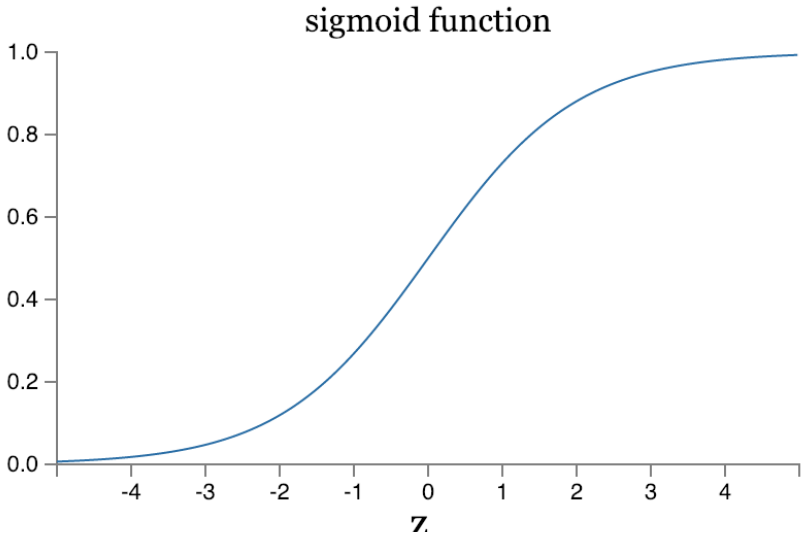
Nielsen, 2015

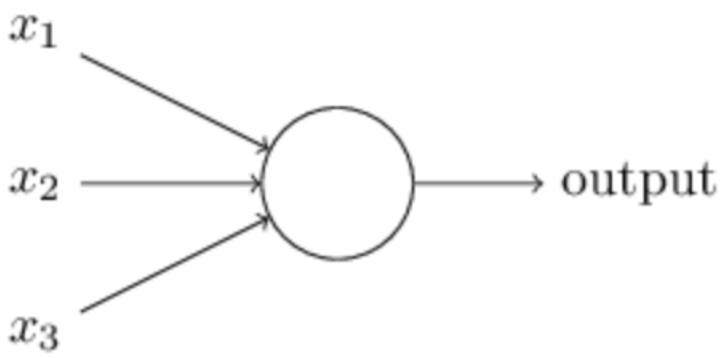
Problem with Logical Functions

- Hard to “tune” in a traditional sense
- Small change in weight can lead to large changes in conditions (or on change at all)
  - Think the volume knob in your car behaving this way
- For this to happen we need neurons to perform a continuous *differentiable* function



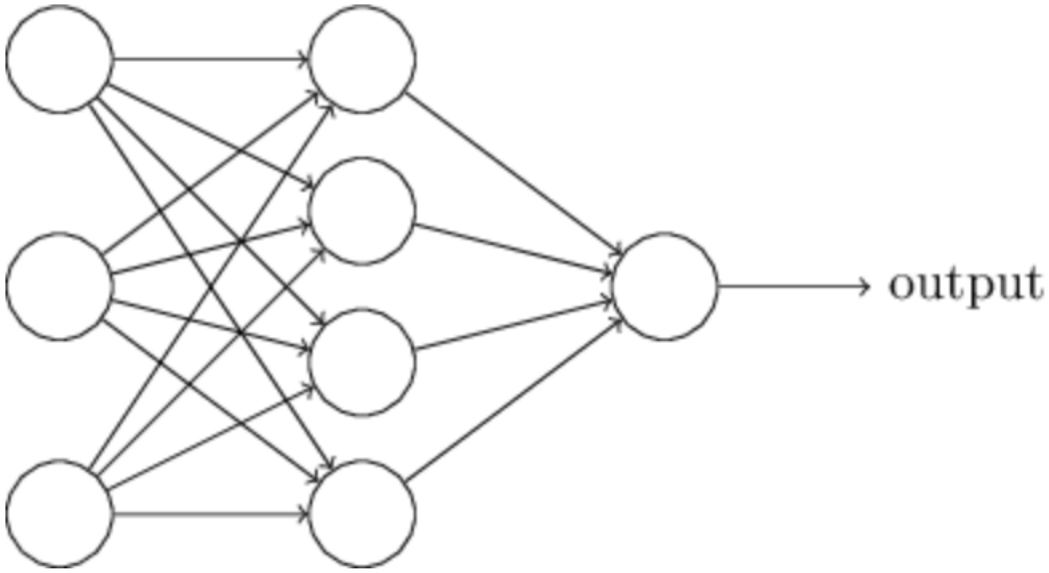
Sigmoidal Neuron





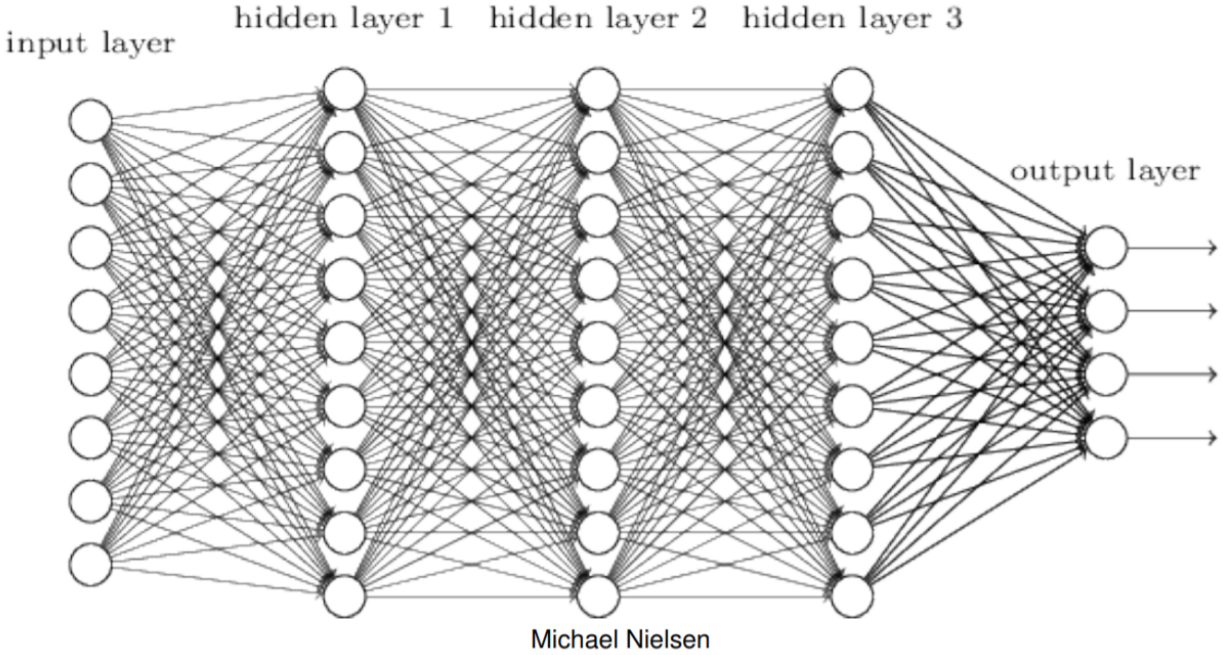
$$\frac{1}{1 + \exp(-\sum_j w_j x_j - b)}$$

Multi-layers of Neurons



Differentiable Computing

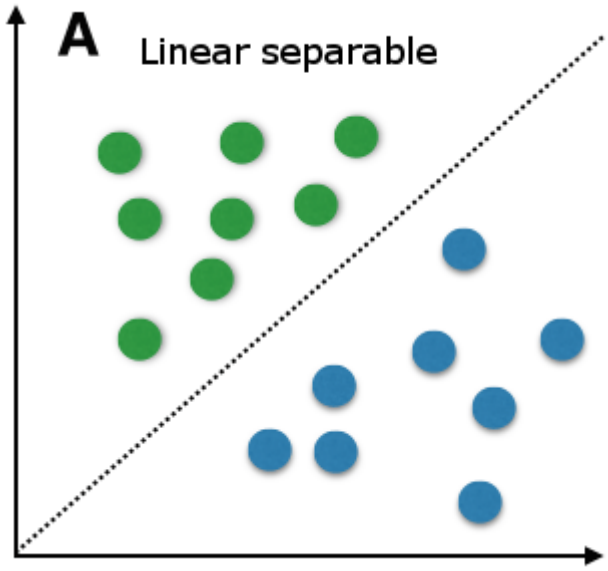
- We can create *learning algorithms* that automatically tune the weights and biases with *Gradient Descent*
  - Tuning occurs in response to external stimuli and w/o direct intervention
  - Creates a circuit designed for the problem at hand

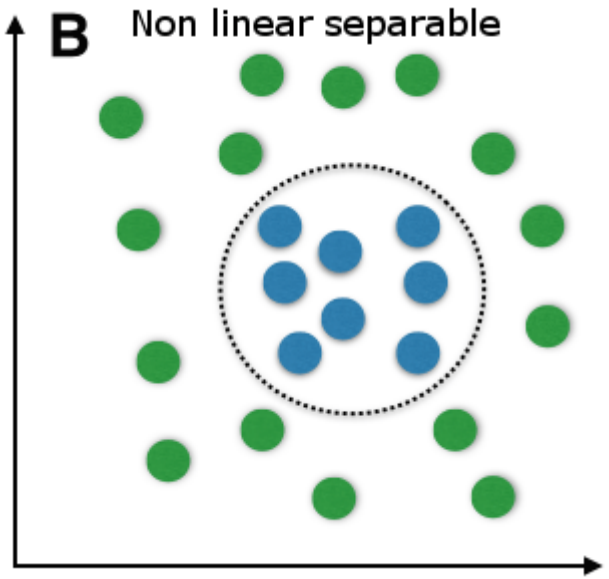


Michael Nielsen

Why go deep?

Depth = complexity

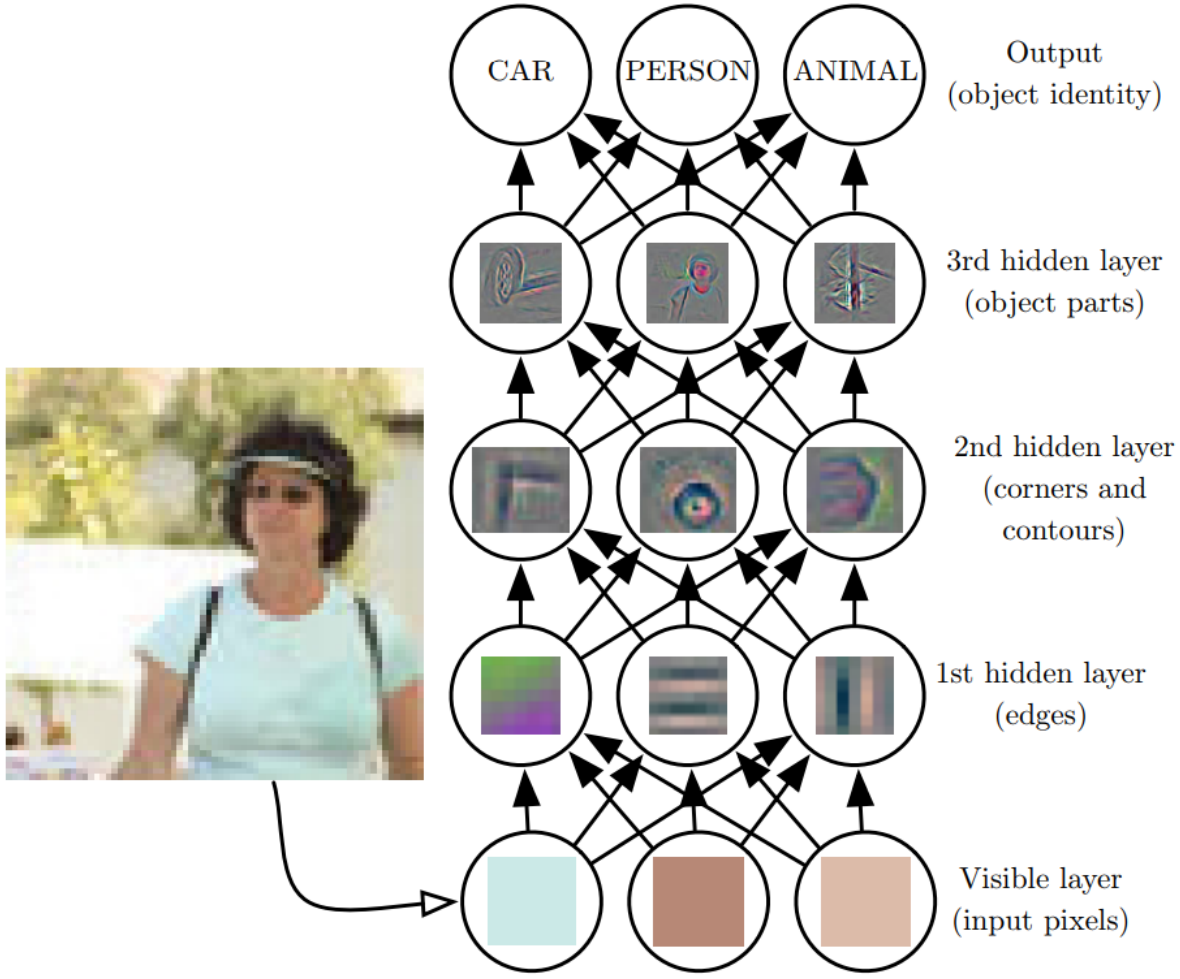




Single layer perceptrons can only handle linear separability

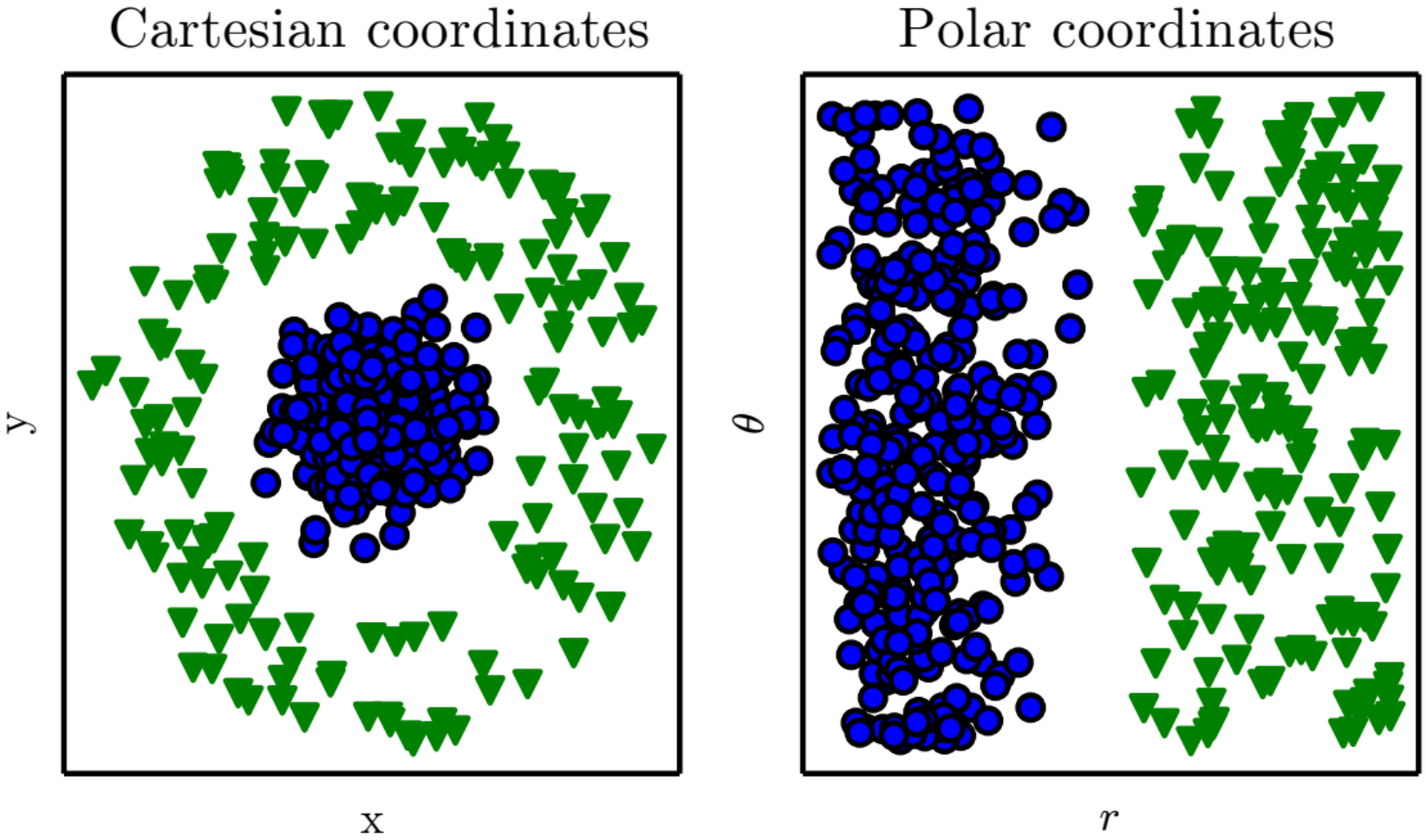
Why go deep?

Representations matter



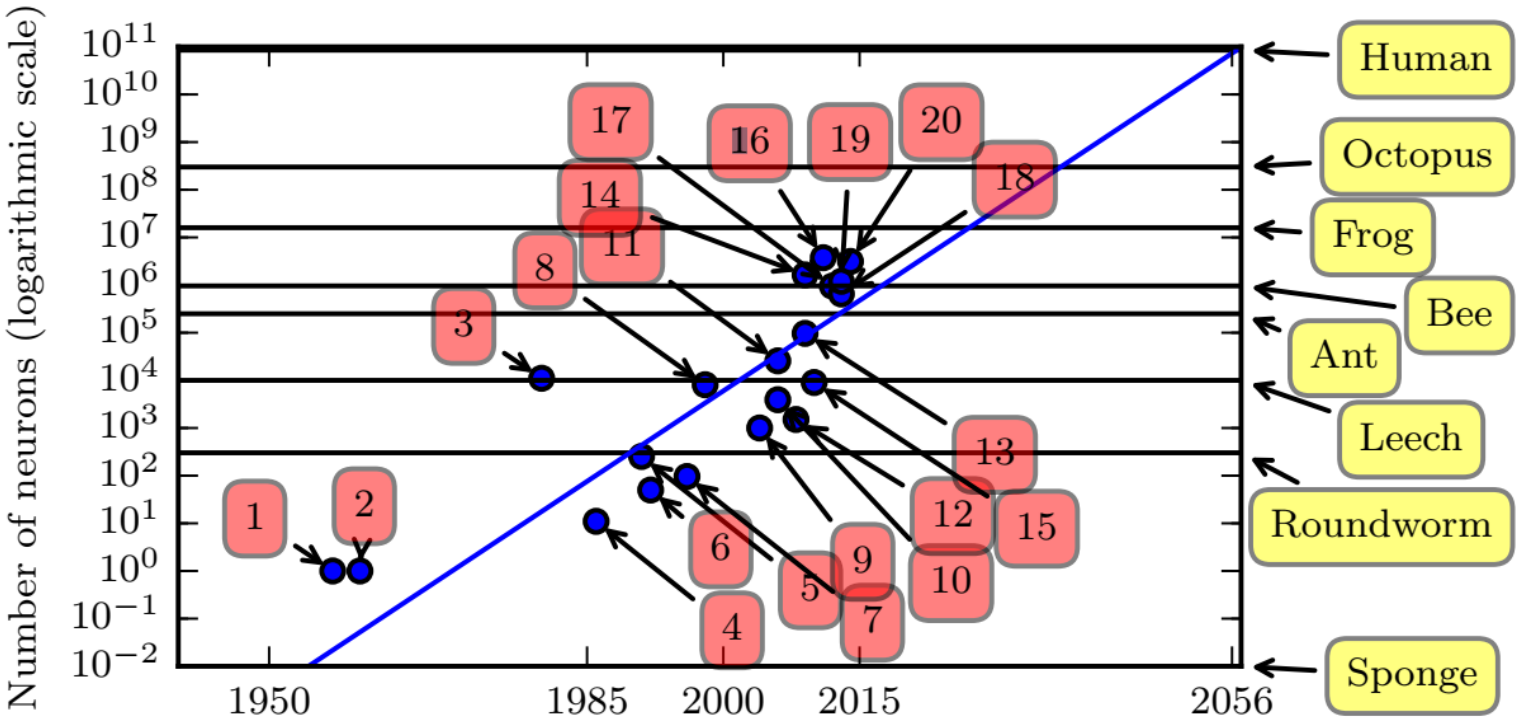
Goodfellow et al., 2016

Representations matter



Goodfellow et al., 2016

Increasing # of neurons



Goodfellow et al., 2016

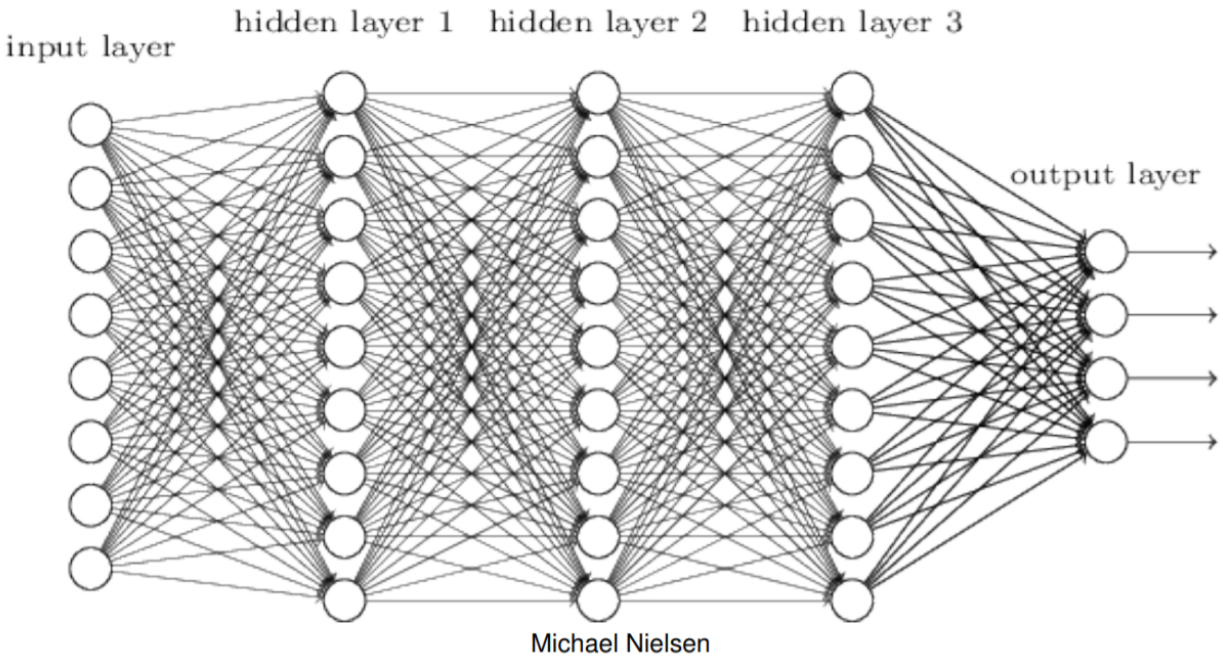
1. Perceptron (Rosenblatt, 1958)
4. Early backpropagation network
6. MLP for speech recognition (Bengio et al, 1991)
11. GPU-accelerated convolutional network (Challeapilla et al., 2006)
20. GoogLeNet (Szegedy et al., 2014a)

Design choices for an ANN

- Learning algorithms
  - Backpropagation
  - Stochastic gradient descent (SGD)
- Activation function (e.g. threshold)
- Cost functions
- Number and dimension of layers
- Connections between layers
- Regularizations
  - Layers
  - Batches
- More…

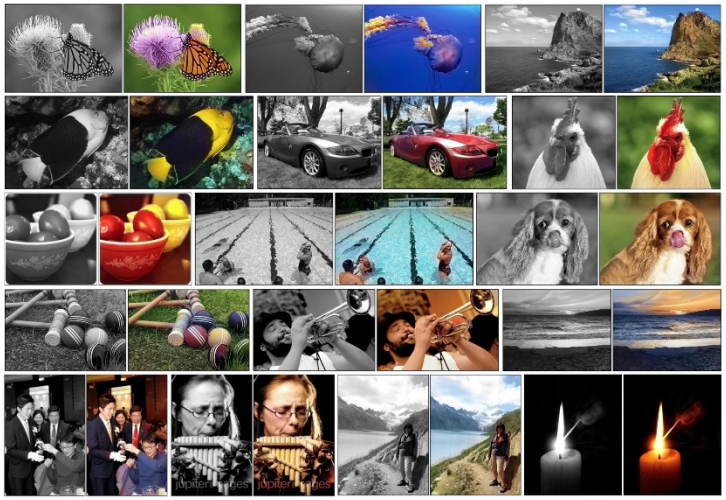
Fully connected network

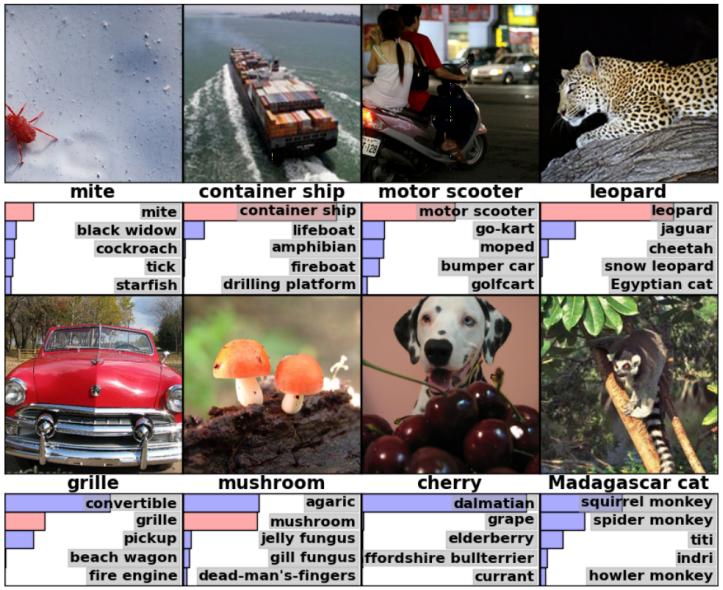
- Every feature interacts with every other feature
- Weight matrix at every level allowed to be dense

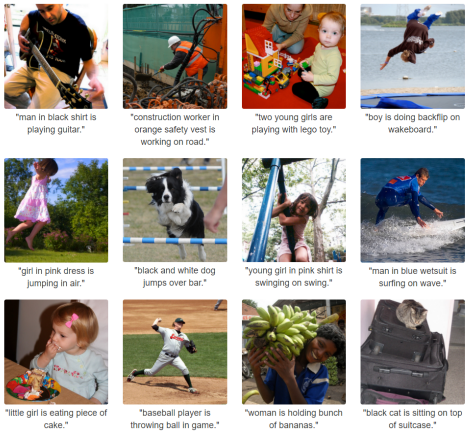


Convolutional Neural Networks (CNNs)

- Very successful in images

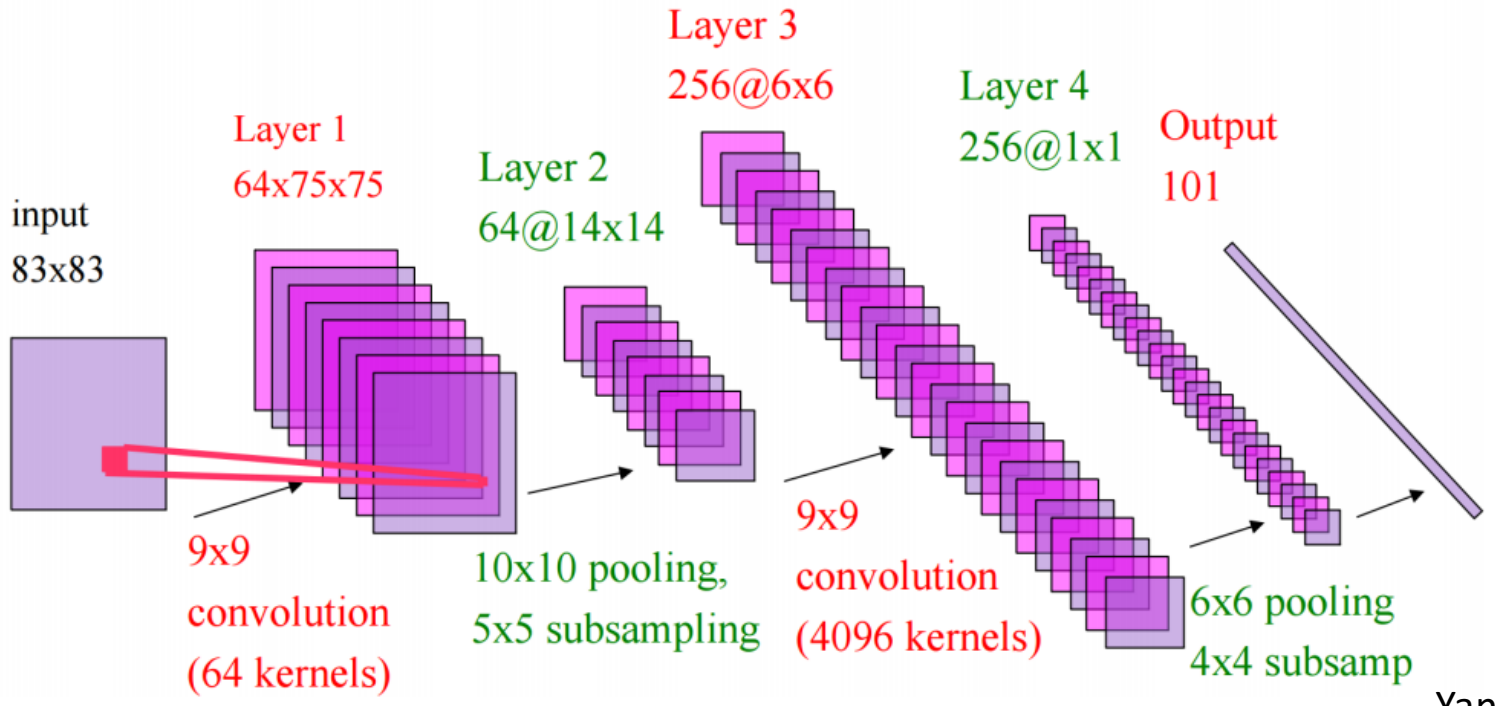






Convolutional Neural Networks (CNNs)

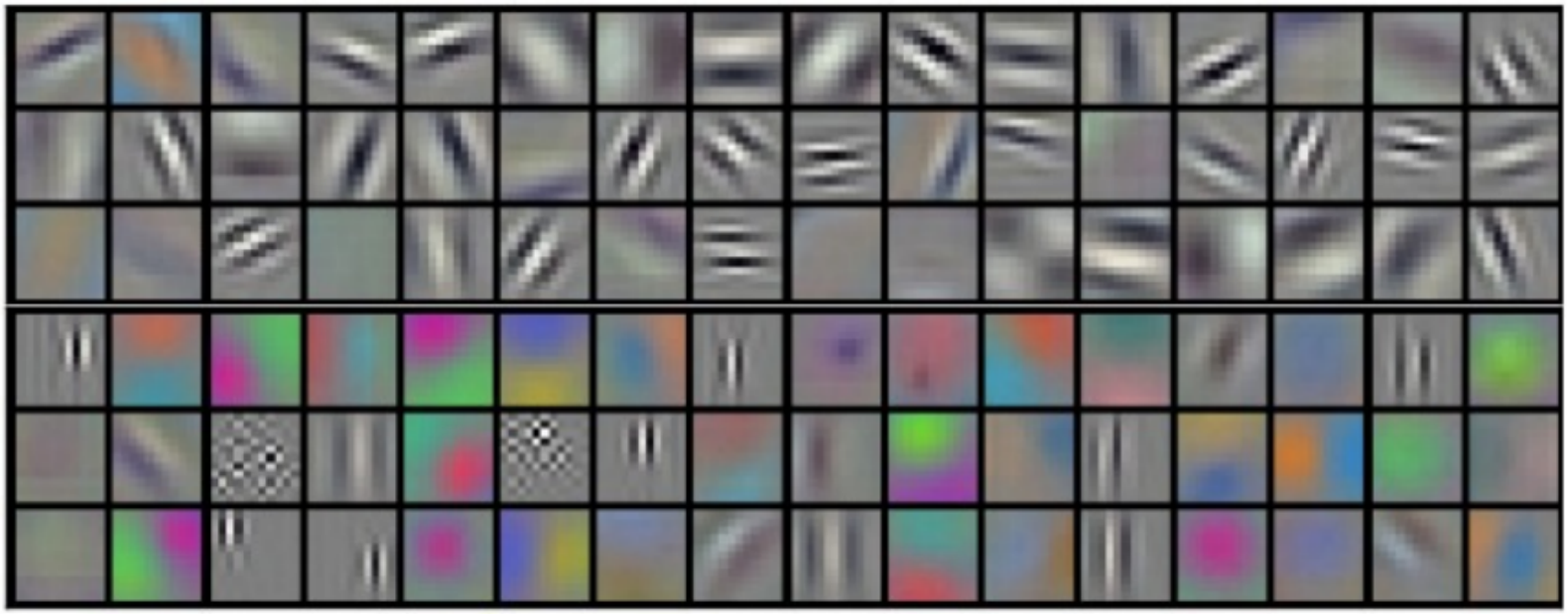
- Only pixels that are close to each other in the image interact with each other (convolution layer)
- Weight matrices are highly structured
- "Pooling" helps to simplify output of convolution layer



Yann LeCun

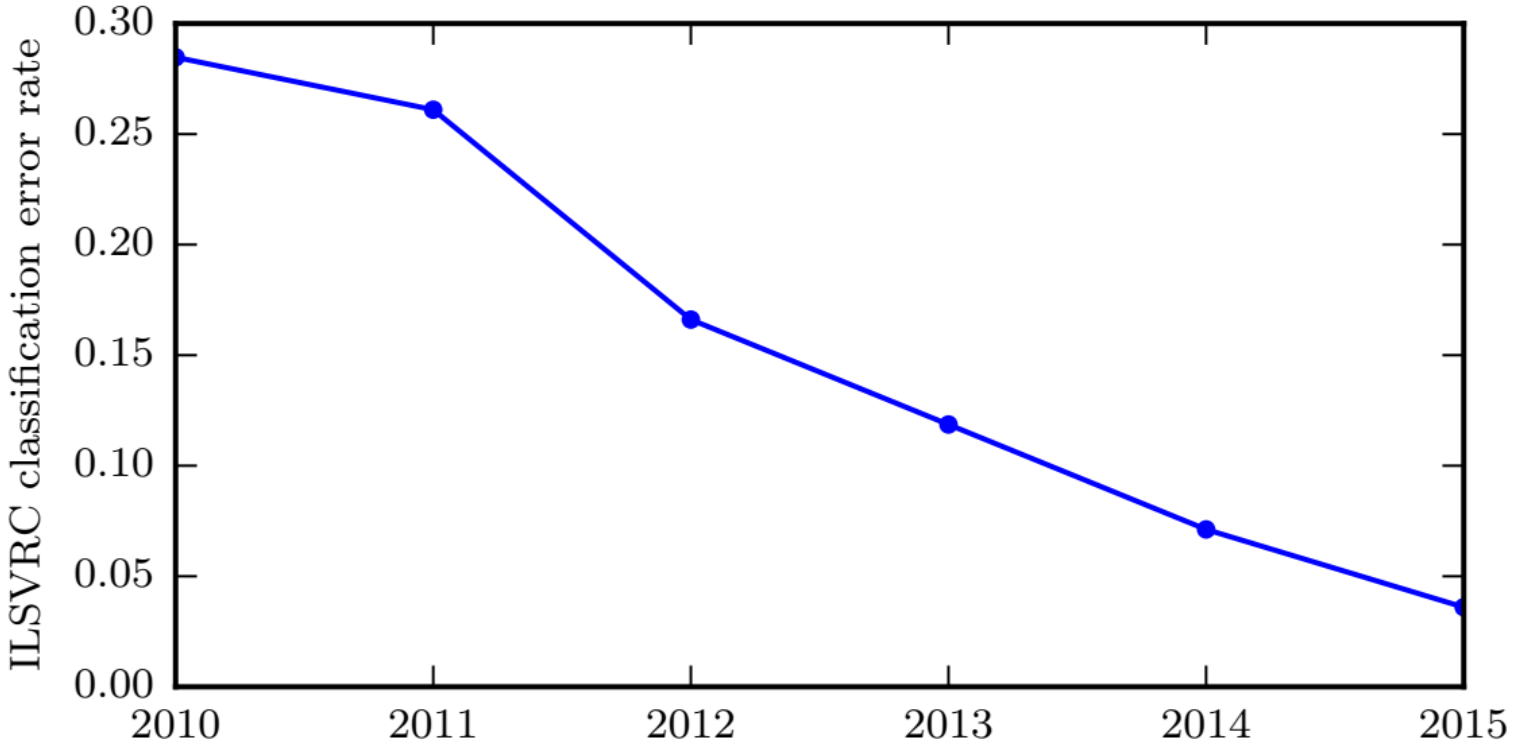
Convolutional Neural Networks (CNNs)

- Weights from the first layer tend to look like directional filters after training
  - Detects edges, color change, etc.



CS 231n, Karpathy

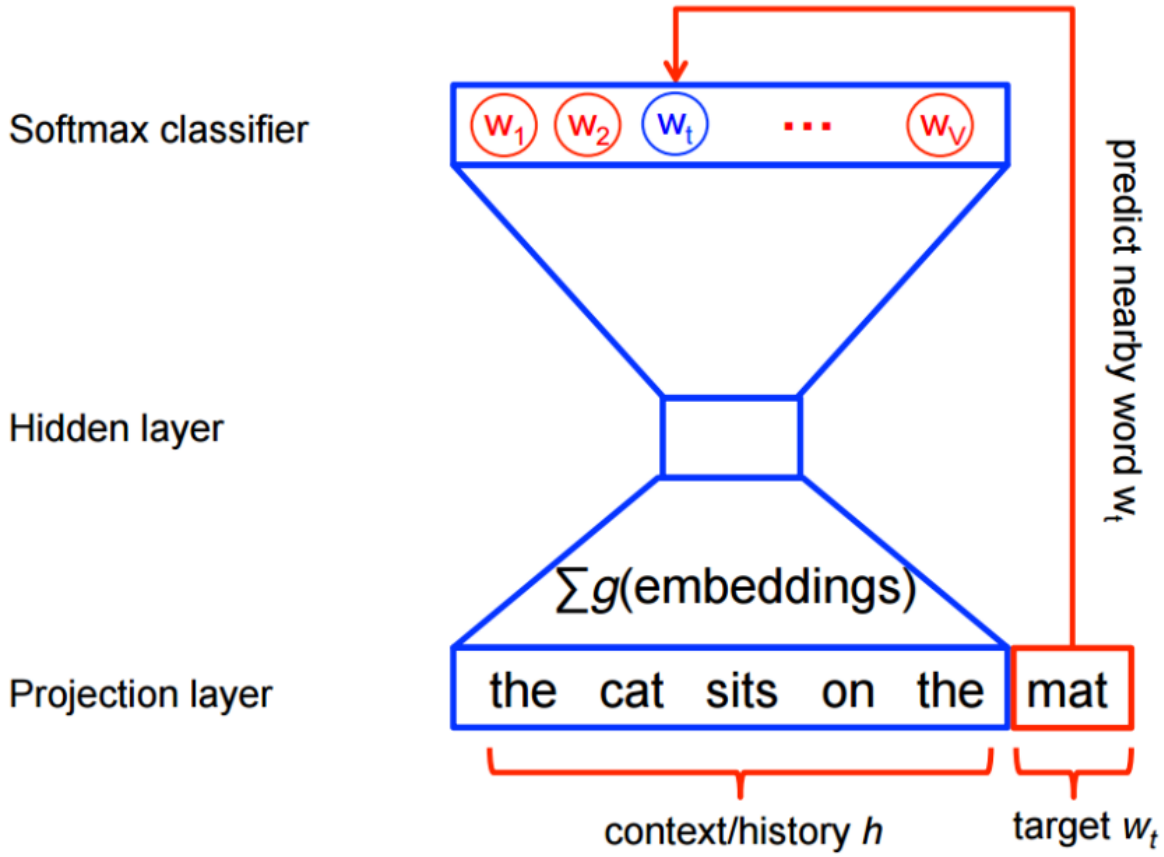
Convolutional Neural Networks (CNNs)



Goodfellow et al., 2016

Word2Vec

- Organization of words via neural networks
- Next word in a sentence can be predicted based on organization



Recurrent Neural Networks (RNNs)

- Useful when time is important

"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."

"girl in pink dress is jumping in air."

"black and white dog jumps over bar."

"young girl in pink shirt is swinging on swing."

"man in blue wetsuit is surfing on wave."

"little girl is eating piece of cake."

"baseball player is throwing ball in game."

"woman is holding bunch of bananas."

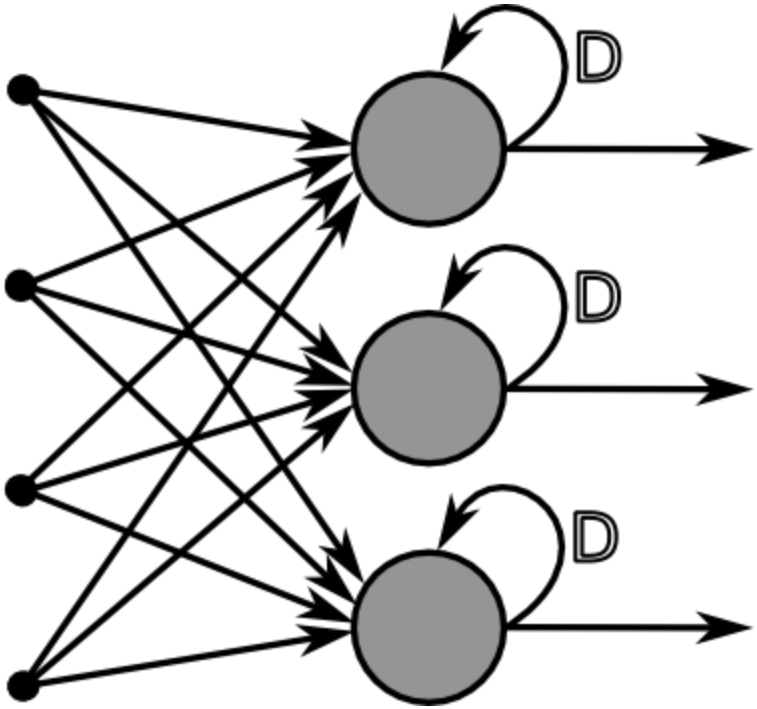
"black cat is sitting on top of suitcase."

```
PANDARUS:
Alas, I think he shall be come approached and the day
When little srain would be attain'd into being never fed,
And who is but a chain and subjects of his death,
I should not sleep.

Second Senator:
They are away this miseries, produced upon my soul,
Breaking and strongly should be buried, when I perish
The earth and thoughts of many states.

DUKE VINCENTIO:
Well, your wit is in the care of side and that.

Second Lord:
They would be ruled after this chamber, and
my fair nues begun out of the fact, to be conveyed,
Whose noble souls I'll have the heart of the wars.

Clown:
Come, sir, I will make did behold your worship.

VIOLA:
I'll drink it.
```

Recurrent Neural Networks (RNNs)

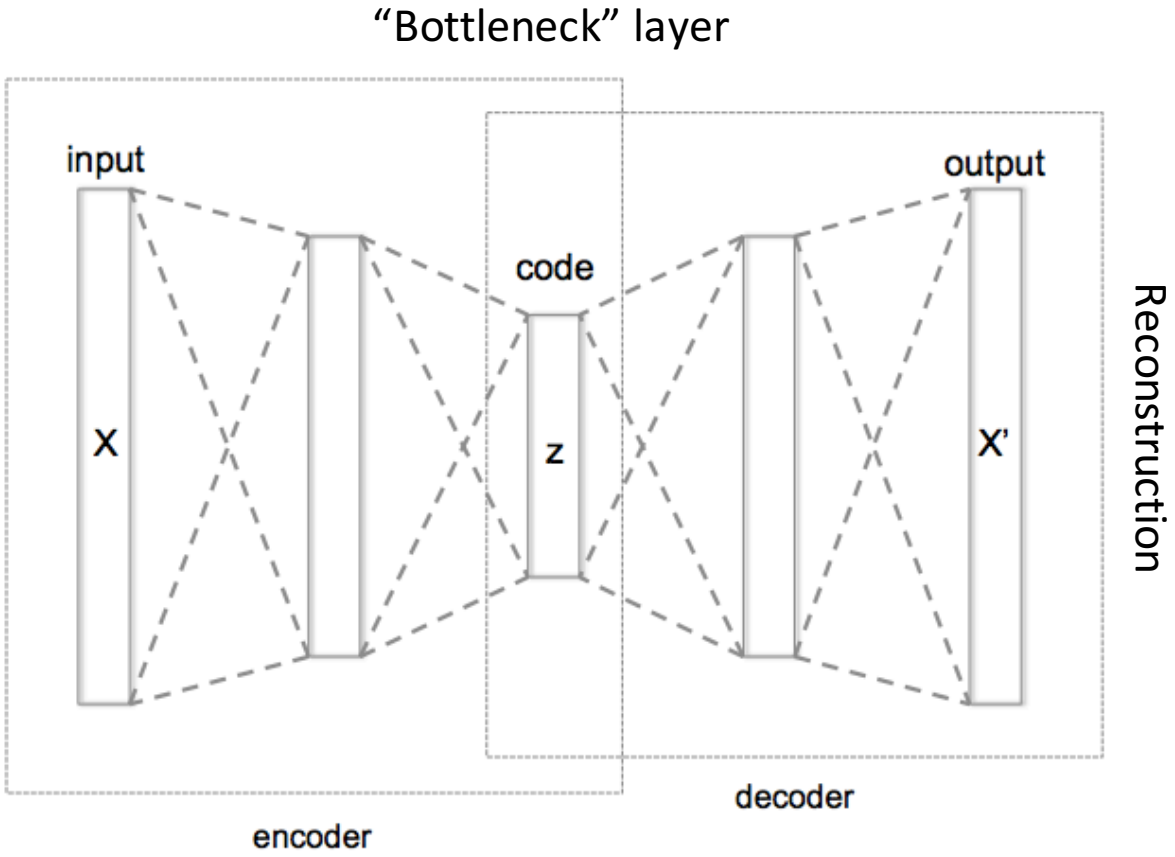
- In feedforward nets (everything we've considered so far), activations of later layers are completely determined by the input
- RNNs allow the hidden layers to be affected by activations at earlier times (i.e. feedback)
  - E.g. a neuron's activation may include as input its activation at an earlier time
  - Cycles are now included in the network
- This time-varying behavior make RNNs useful for analyzing data that change over time (e.g. speech)
- Training can be difficult for long-term dependencies

Fully Recurrent Network

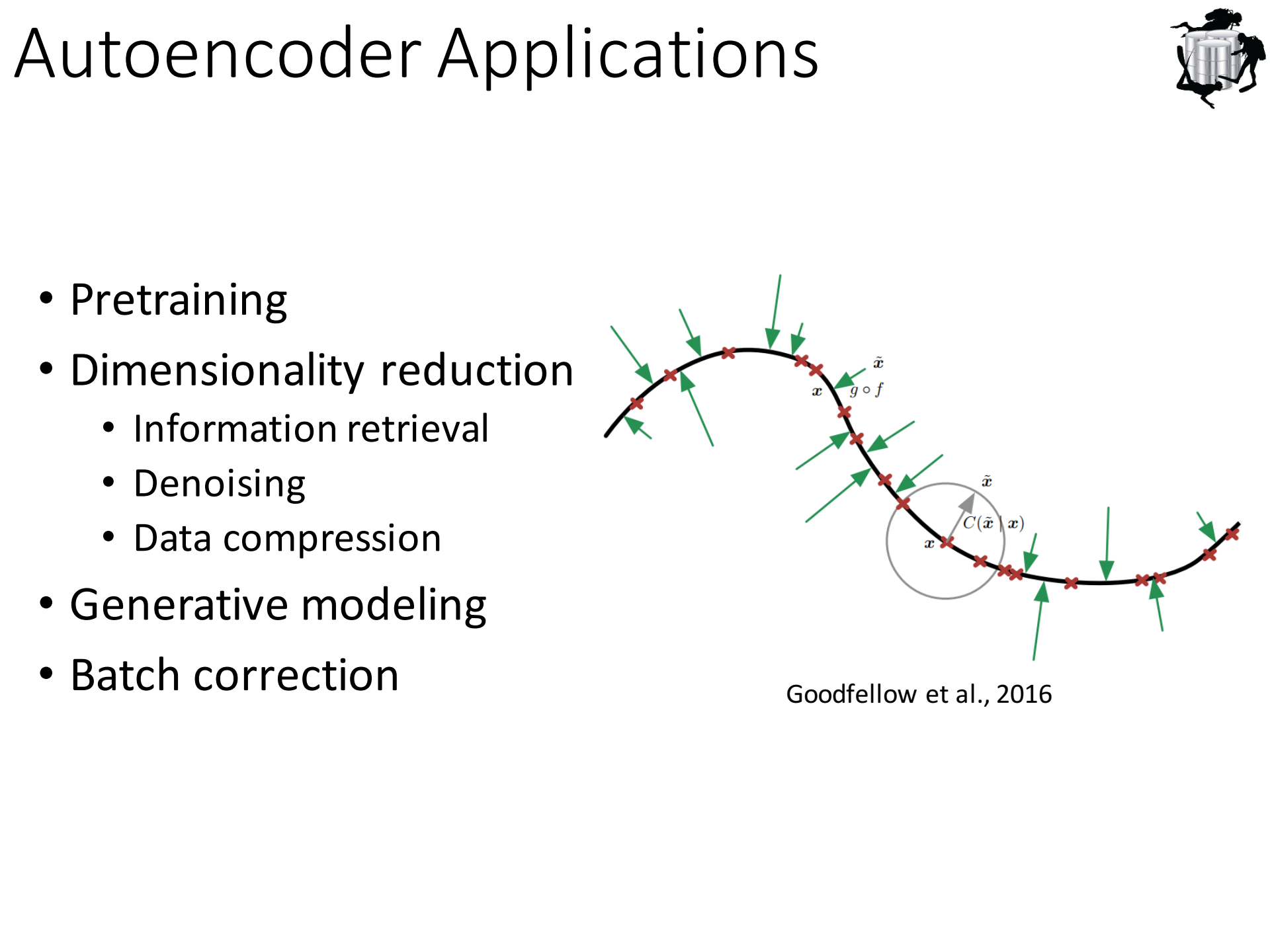


Autoencoders

- Attempts to compress the data and then reconstruct the input



Autoencoder Applications

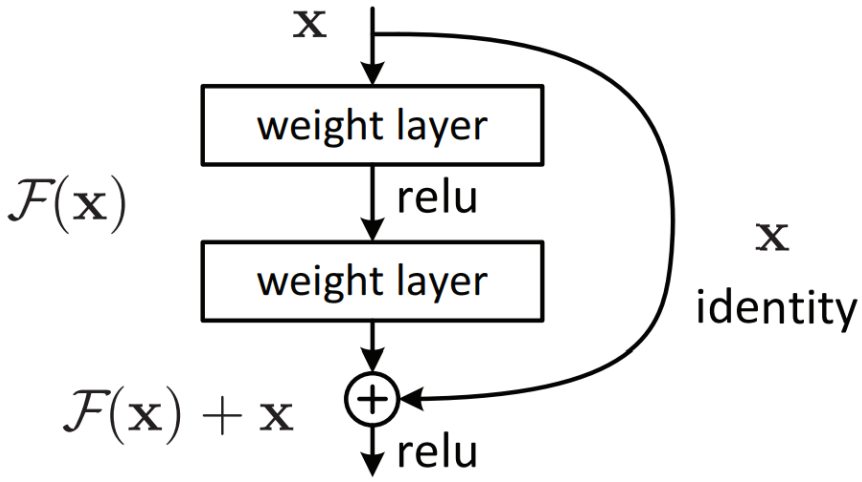


- Pretraining
- Dimensionality reduction
  - Information retrieval
  - Denoising
  - Data compression
- Generative modeling
- Batch correction

Goodfellow et al., 2016

Ultra Deep Learning (e.g. ResNet)

- Very deep neural nets are difficult to train
  - Accuracy can degrade with deeper networks
- ResNet developed a framework to address this degradation
  - Successfully trained a 152 layer network
  - Won the ILSVRC 2015 image classification task



arxiv.org/abs/1512.03385

Generative Models

- Create a map from random noise into distribution of training data to generate samples
- Generative Adversarial Net (GAN)
  - Generative model is pitted against an discriminative model that determines whether a sample is from the model or the data
  - Improves both generation and discrimination

Generative Models

a) b) c) d)

Goodfellow

Deep Reinforcement Learning

Alpha Go Zero

Alpha Zero

nature
International journal of science

Access provided by Yale University

Altmetric: 2188 Citations: 1 More detail »

Article

Mastering the game of Go without human knowledge

David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, Yutian Chen, Timothy Lillicrap, Fan Hui, Laurent Sifre, George van den Driessche, Thore Graepel & Demis Hassabis

Nature **550**, 354–359 (19 October 2017)
doi:10.1038/nature24270
Download Citation

Computational science
Computer science Reward

Received: 07 April 2017
Accepted: 13 September 2017
Published online: 18 October 2017

arXiv.org > cs > arXiv:1712.01815

Search or
(Help | Advanc

Computer Science > Artificial Intelligence

Mastering Chess and Shogi by Self-Play with a General Reinforcement Learning Algorithm

David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, Timothy Lillicrap, Karen Simonyan, Demis Hassabis

(Submitted on 5 Dec 2017)

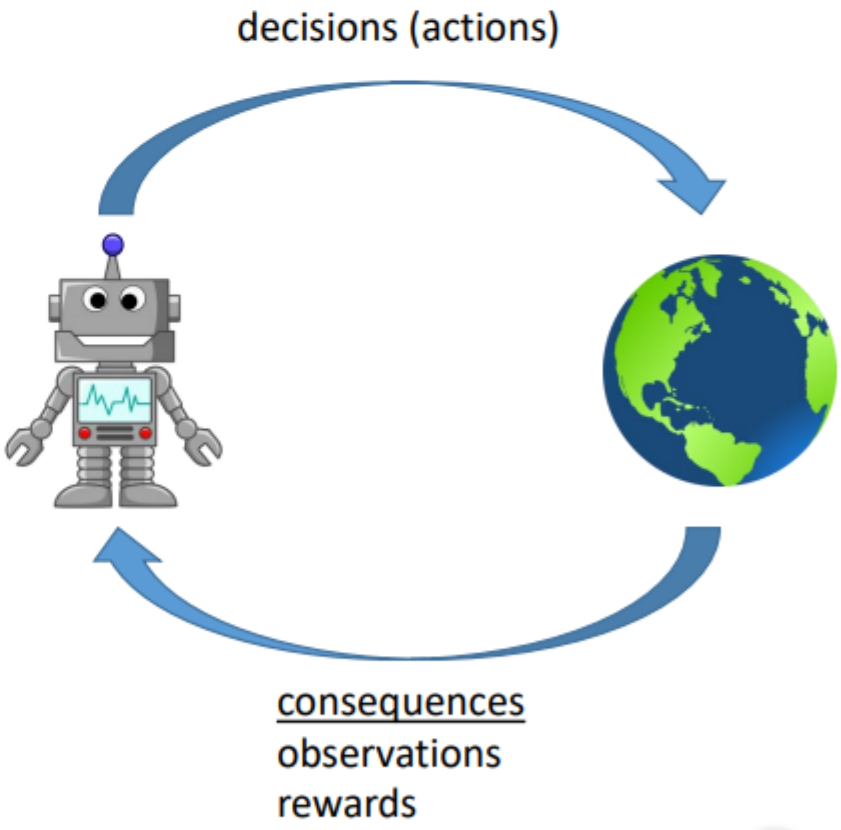
The game of chess is the most widely-studied domain in the history of artificial intelligence. The strongest programs are based on a combination of sophisticated search techniques, domain-specific adaptations, and handcrafted evaluation functions that have been refined by human experts over several decades. In contrast, the AlphaGo Zero program recently achieved superhuman performance in the game of Go, by tabula rasa reinforcement learning from games of self-play. In this paper, we generalise this approach into a single AlphaZero algorithm that can achieve, tabula rasa, superhuman performance in many challenging domains. Starting from random play, and given no domain knowledge except the game rules, AlphaZero achieved within 24 hours a superhuman level of play in the games of chess and shogi (Japanese chess) as well as Go, and convincingly defeated a world-champion program in each case.

Subjects: **Artificial Intelligence (cs.AI)**; Learning (cs.LG)
Cite as: arXiv:1712.01815 **[cs.AI]**
(or arXiv:1712.01815v1 **[cs.AI]** for this version)

Submission history
From: David Silver [view email]
[v1] Tue, 5 Dec 2017 18:45:38 GMT (272kb,D)

Deep Reinforcement Learning

- What is reinforcement learning?



Deep Reinforcement Learning

- Examples

Actions: muscle contractions
Observations: sight, smell
Rewards: food

Actions: motor current or torque
Observations: camera images
Rewards: task success measure
(e.g., running speed)

Actions: what to purchase
Observations: inventory levels
Rewards: profit

Further reading

- Nielsen, Chapter 1
- Goodfellow et al., Chapter 1