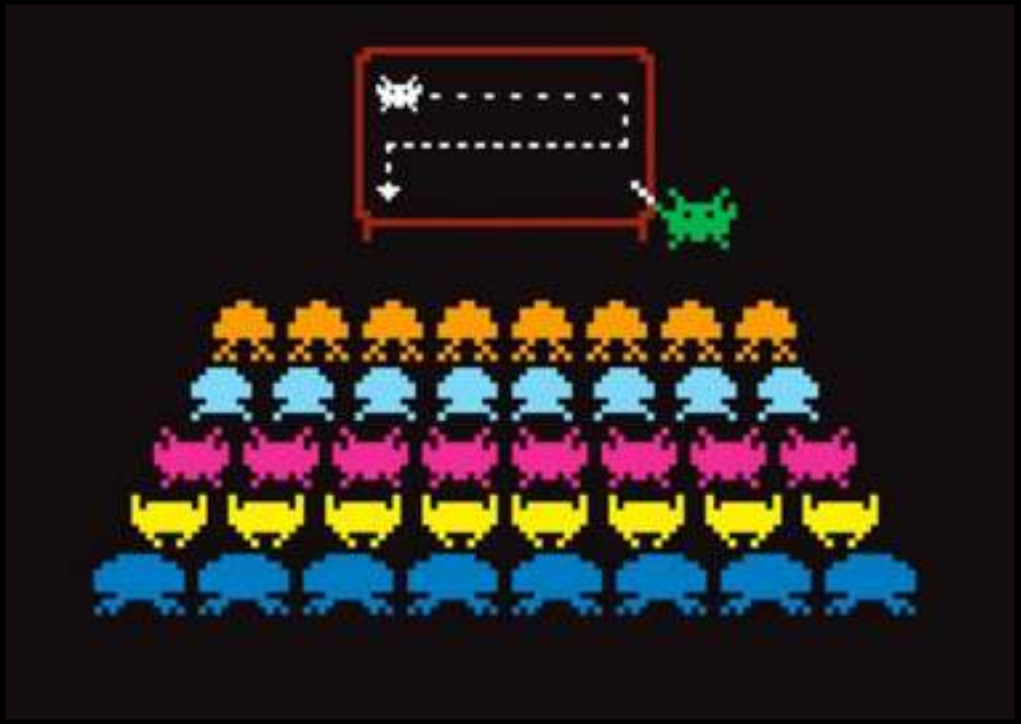
From the brain to deep learning

(and back)



Dmytro Fishman

based on venia legendi lecture by Raul Vincente

Institute of Computer Science, University of Tartu

NATURE | NEWS FEATURE

عربي

A world where everyone has a robot: why 2040 could blow your mind

SCIENCE

Scientists See Promise in Deep-Learning Progr

By JOHN MARKOFF NOV. 23, 2012

BBC Sign in News Sport Weather Sho

NEWS

Home Video World UK Business Tech Science Magazi

NATURE | NEWS

عربي

Game-playing software holds lessons
neuroscience

DeepMind computer provides new way to investigate how the brai

Forbes / Tech

Top 20 Stocks for 2016

DEC 29, 2014 @ 11:37 AM 89,471 VIEWS

Tech 2015: Deep Learning And Machine Intelligence Will Eat The World

'Deep learning' technology inspired by human brain

Google a step closer to developin
machines with human-like intell

culture business lifestyle fashion environment tech trave

What Google's winning Go algorithm will do r

AlphaGo's techniques could have broad uses, but moving beyond games is a c

Elizabeth Gibney

15 March 2016

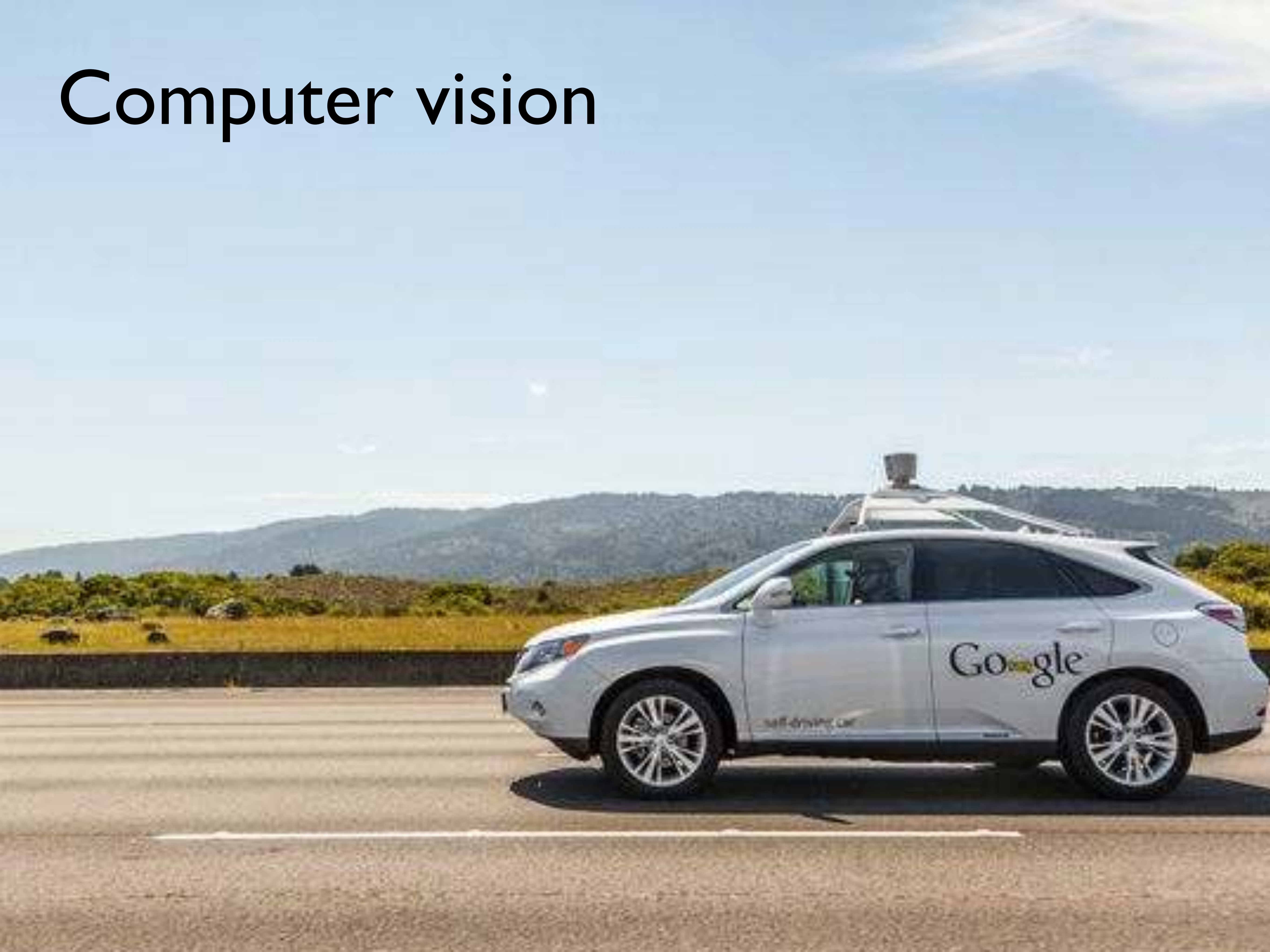
ndroids do dream of electric sheep

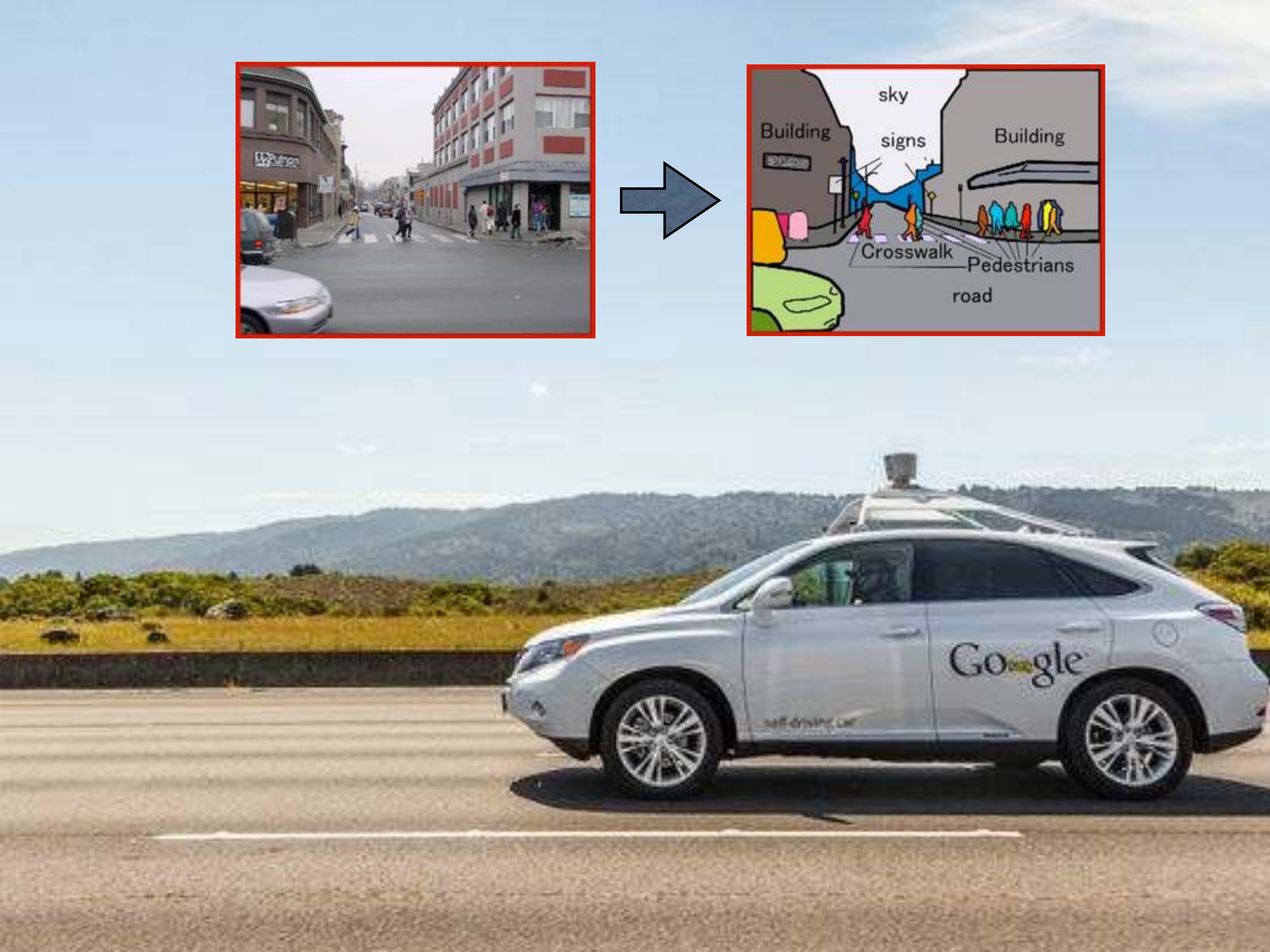
up feedback loop in its image recognition neural network - which

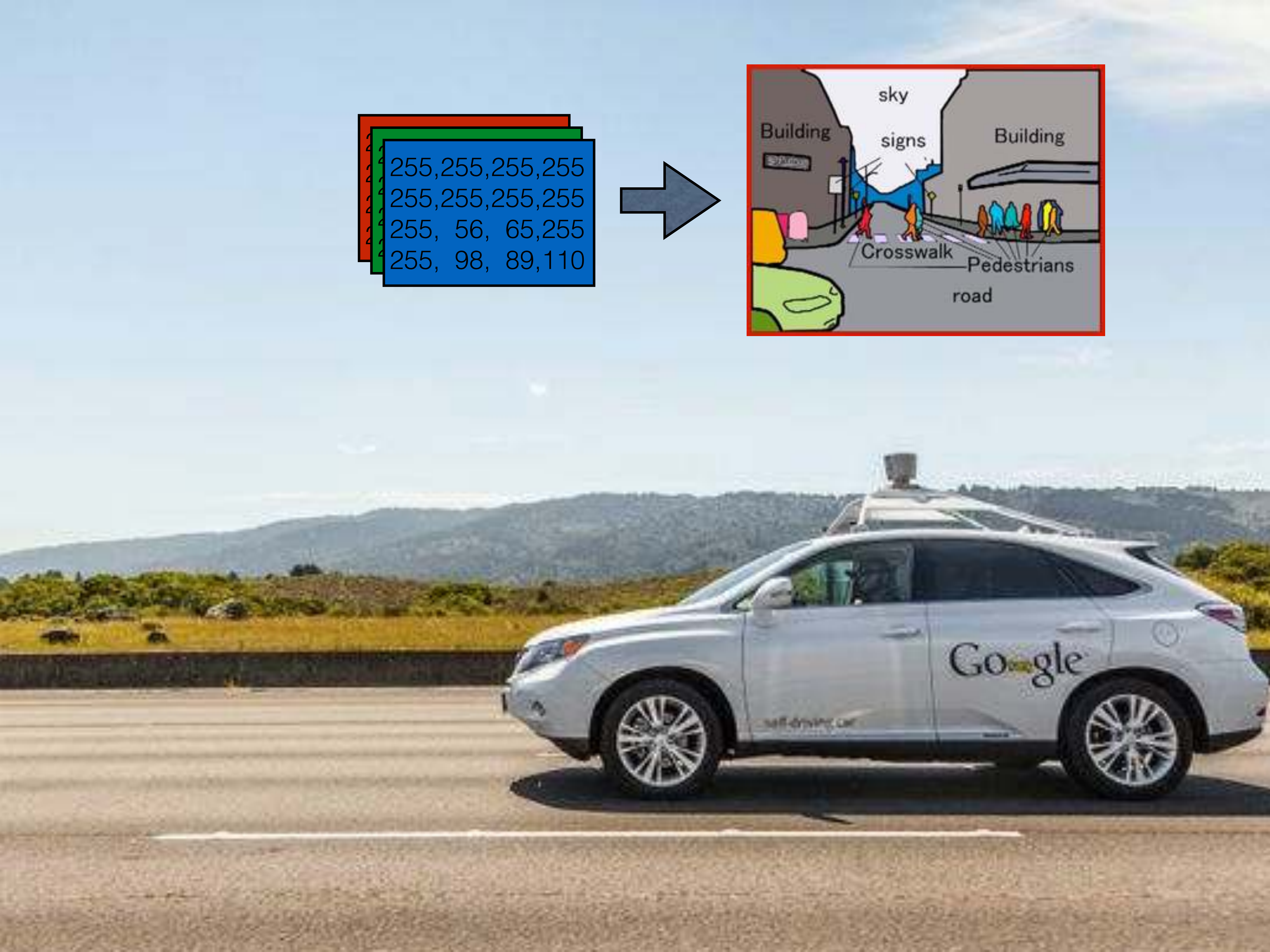
Empowering technologies

https://www.youtube.com/watch?v=eu9kMIeS0wQ

Computer vision




sky
Building
signs
Building
Crosswalk
Pedestrians
road
Google

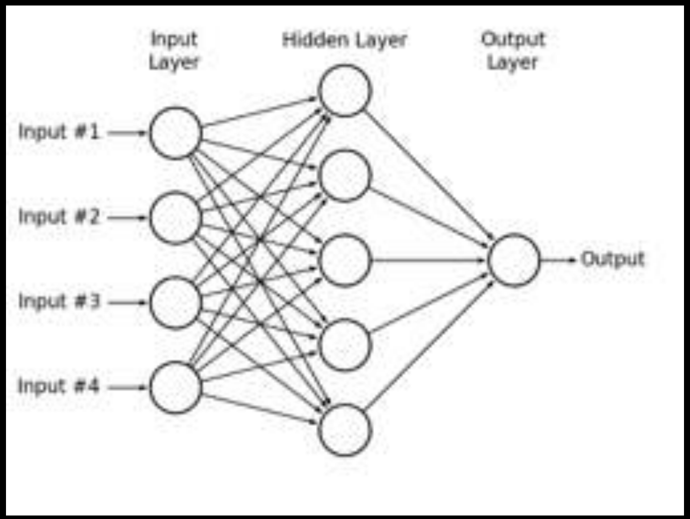


255,255,255,255
255,255,255,255
255, 56, 65,255
255, 98, 89,110
sky
Building
signs
Building
Crosswalk
Pedestrians
road
Google

DeepMind AlphaGo vs World Champion

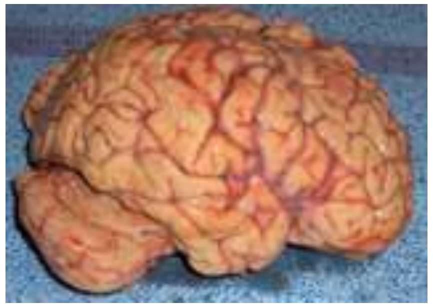
DeepMind AlphaGo vs World Champion
4 - 1


ALPHAGO
01:59:50
AlphaGo
Lee Sedol

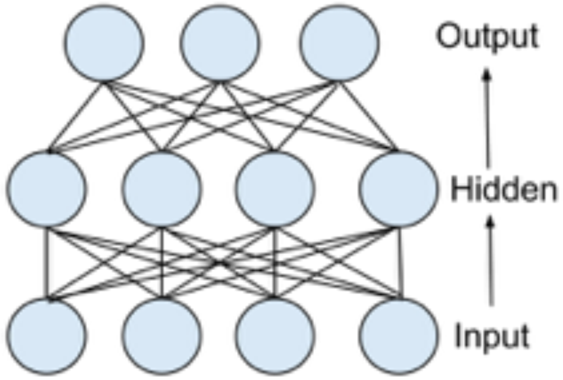

Input Layer
Hidden Layer
Output Layer
Input #1
Input #2
Input #3
Input #4
Output

VS

Brains 101

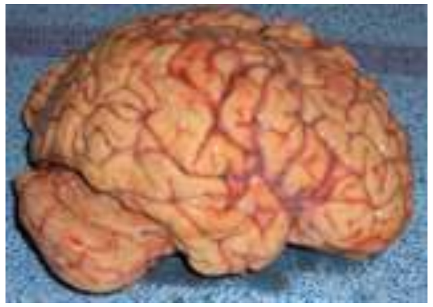


Deep Learning

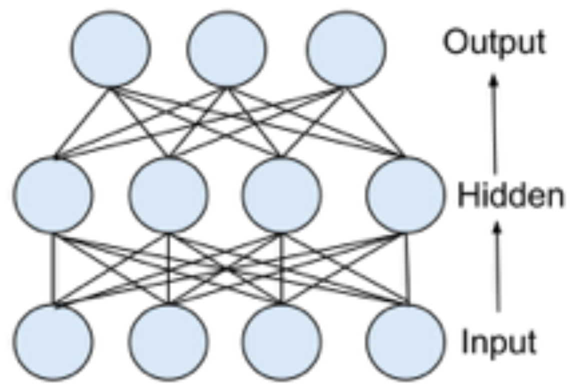


Feynman dictum

Brains 101



Deep Learning



Feynman dictum

your brain (a spongy, 1.5 kg of tissue) is the *CEO* + *Data scientist* of your body

your brain (a spongy, 1.5 kg of tissue) is the *CEO* + *Data scientist* of your body

- >1000 disorders of the brain and the nervous system

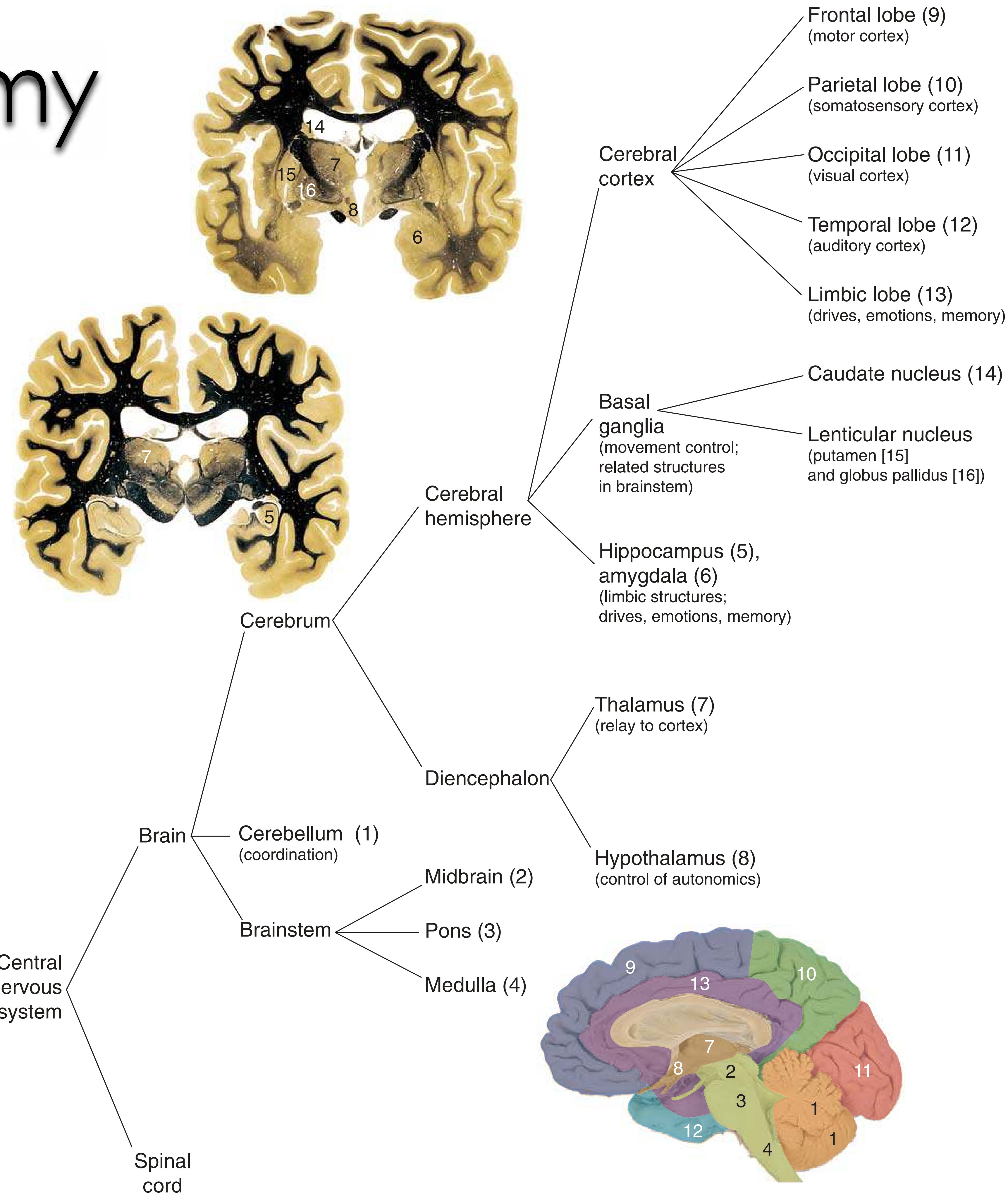
"One of the difficulties in understanding the brain is that it is like nothing so much as a lump of porridge"

R.L. Gregory

Eye and the Brain: the psychology of seeing,
New York, 1966, McGraw-Hill

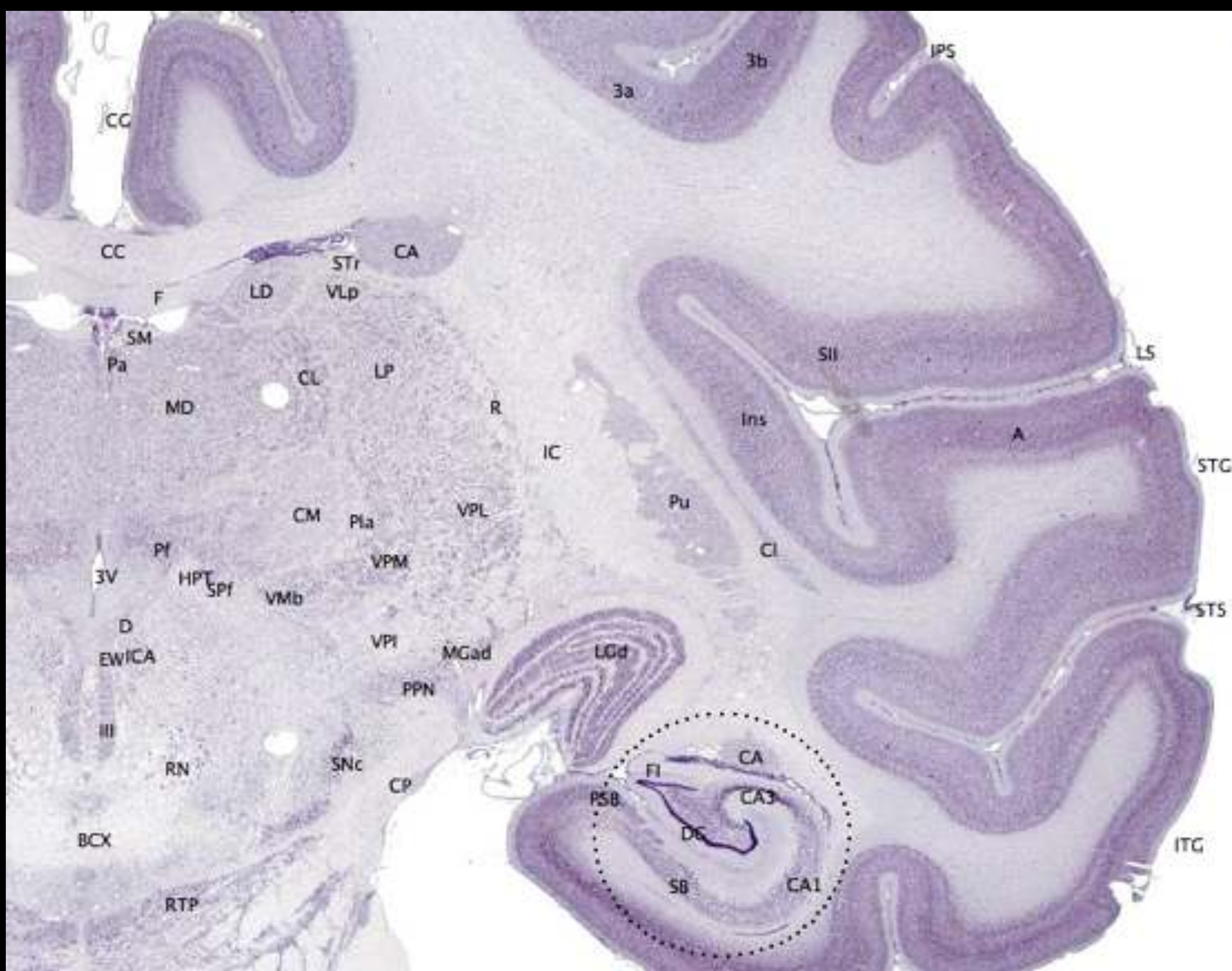
Gross anatomy

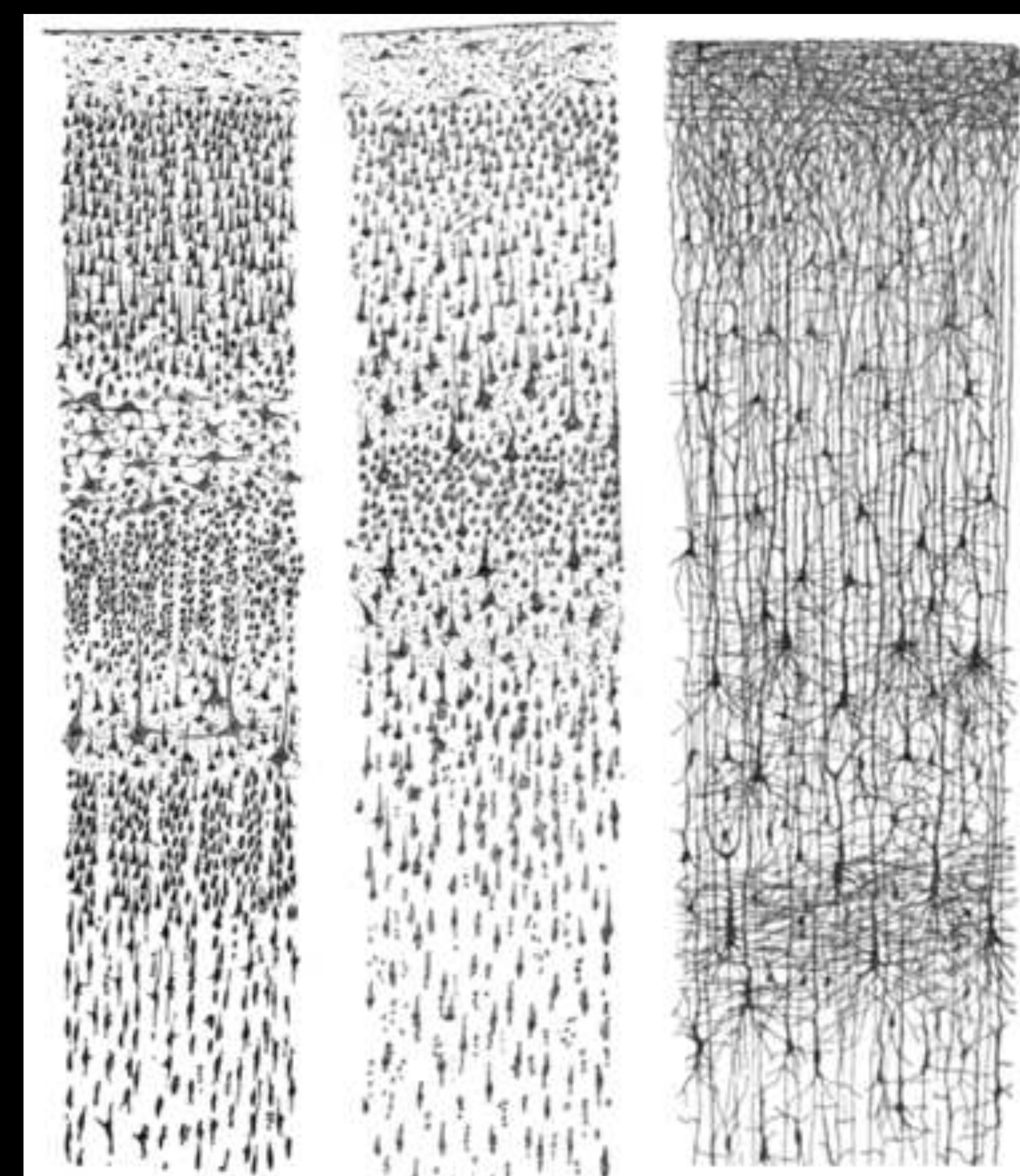
Overview of the subdivisions of the CNS



Gross anatomy

Cerebral cortex: outermost 6 layered structure of the neural tissue of human and other mammals (2-4 mm). Key role in high cognitive functions (memory, attention, language, ...)



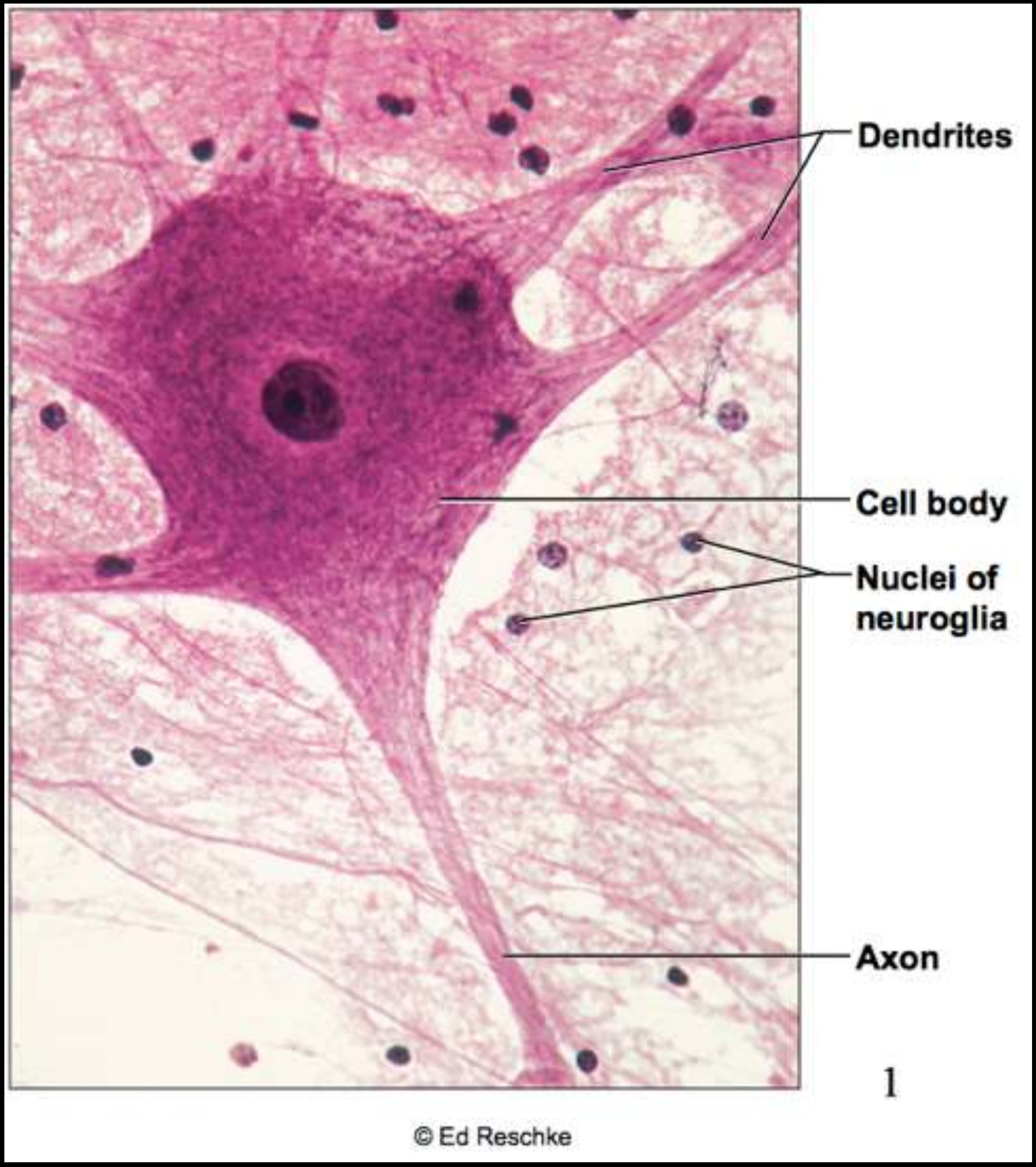


Cellular elements

2 cell categories in the nervous system:

Neurons (information processing, signaling elements, 100 billion)

Glial cells (10 x neurons)



Neurons as computational elements

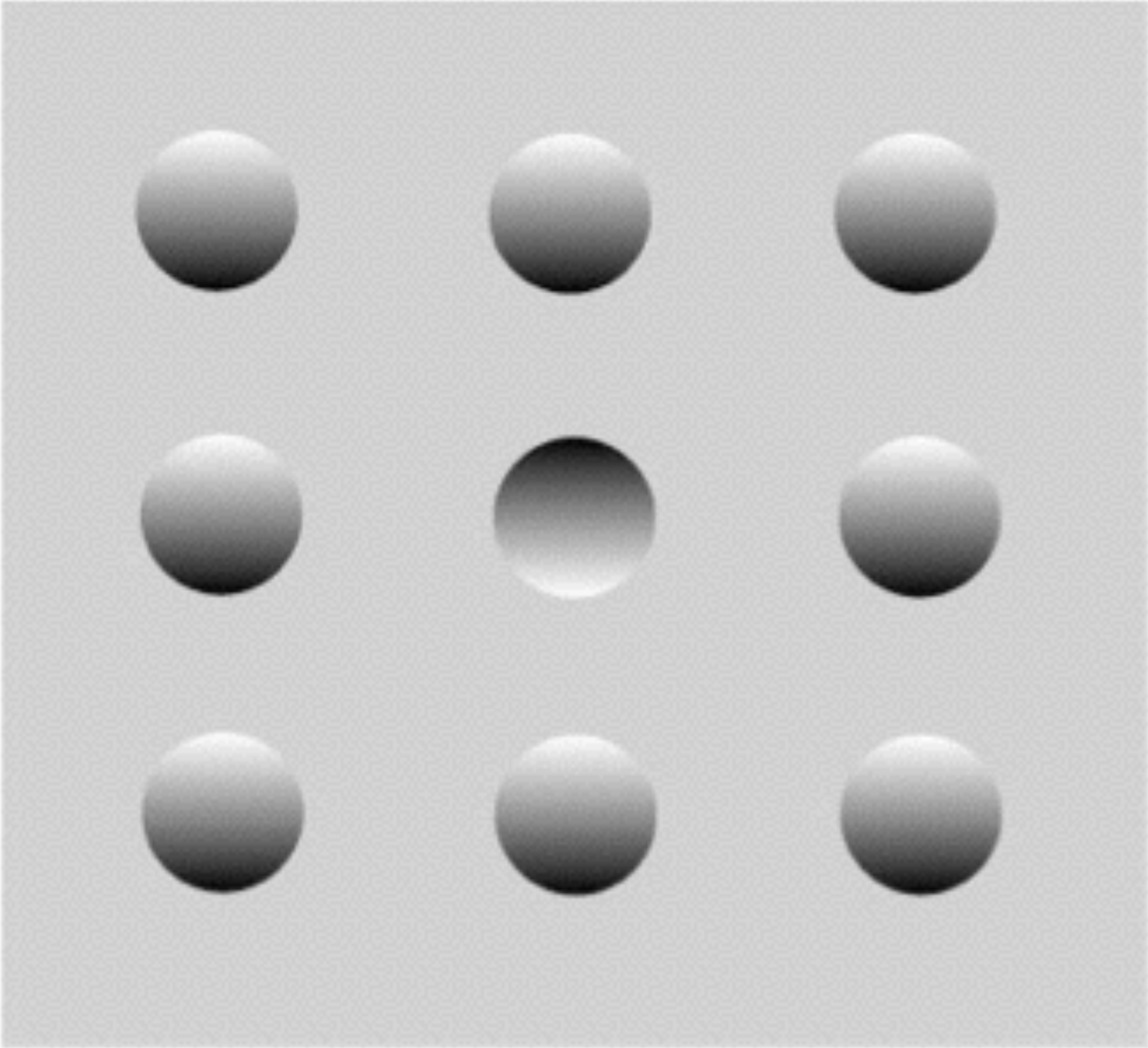
- Slow (below kHz)
- Unreliable (synaptic failure p=0.5)
- Aging (50-80k neurons die every day)
- Re-organization of their connections (learning & memory)

Neurons as computational elements

- Slow (below kHz)
- Unreliable (synaptic failure p=0.5)
- Aging (50-80k neurons die every day)
- Re-organization of their connections (learning & memory)

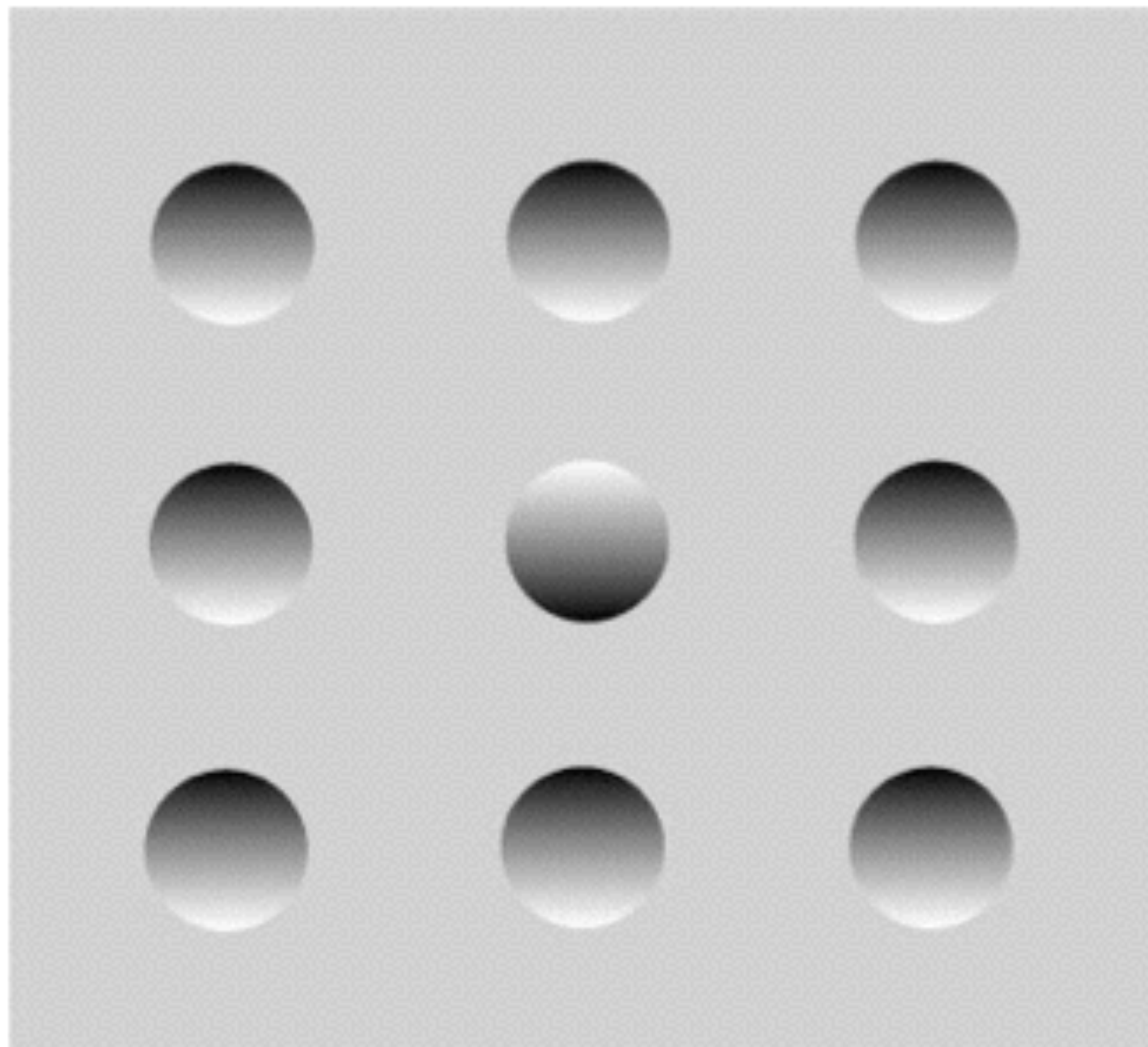
Neurons simultaneously transmit, store, and modify information

How the brain makes sense of the world?



Which dimples are popping out and which popping in?

Which dimples are popping out and which popping in?



$$P(B \mid A) = \frac{P(A \mid B)\, P(B)}{P(A)}$$

Bayes' rule

$$P(B \mid A) = \frac{P(A \mid B)\, P(B)}{P(A)}$$

Bayes' rule

Formula for computing:

P(what's in the world | sensory data)

this is what your brain wants to know!

$$P(B \mid A) = \frac{P(A \mid B)\,P(B)}{P(A)}$$

Bayes' rule

Formula for computing:

P(what's in the world | sensory data)

this is what your brain wants to know!

from

P(sensory data | what's in the world)

"Likelihood"
given by laws of physics;
ambiguous because many world states give rise to same sense data

$$P(B \mid A) = \frac{P(A \mid B)\, P(B)}{P(A)}$$

Bayes' rule

Formula for computing:

P(what's in the world I sensory data)

this is what your brain wants to know!

from

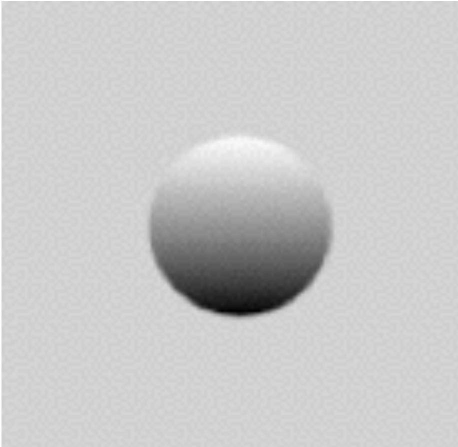
P(sensory data I what's in the world)

"Likelihood"
given by laws of physics;
ambiguous because many world states give rise to same sense data

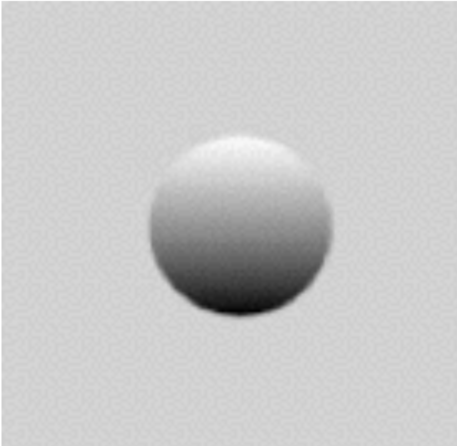
&

P(what's in the world)

"prior"
given by past experience

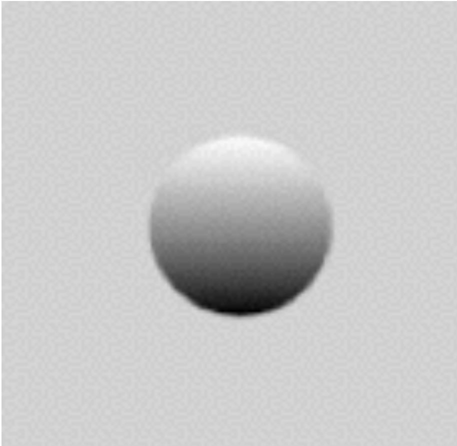


P(IN | image) VS P(OUT | image)



$P(\text{IN} \mid \text{image})$ **vs** $P(\text{OUT} \mid \text{image})$

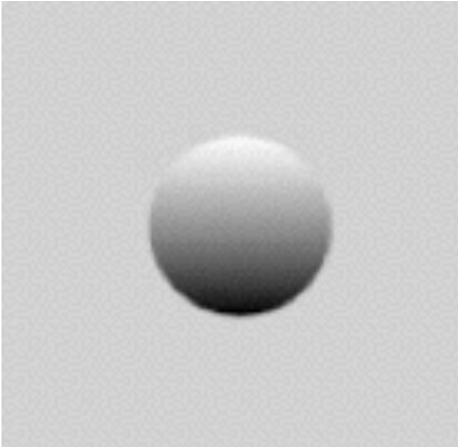
Applying Bayes rule: $\overbrace{P(\text{world} \mid \text{sense data})}^{\text{posterior}} \propto \overbrace{P(\text{sense data} \mid \text{world})}^{\text{likelihood}} \overbrace{P(\text{world})}^{\text{prior}}$



P(IN | image) **vs** P(OUT | image)

Applying Bayes rule: $\overbrace{P(\text{world} \mid \text{sense data})}^{\text{posterior}} \propto \overbrace{P(\text{sense data} \mid \text{world})}^{\text{likelihood}}\ \overbrace{P(\text{world})}^{\text{prior}}$

P(IN | image) = P(image | IN & light below) × P(IN) × P(light below)



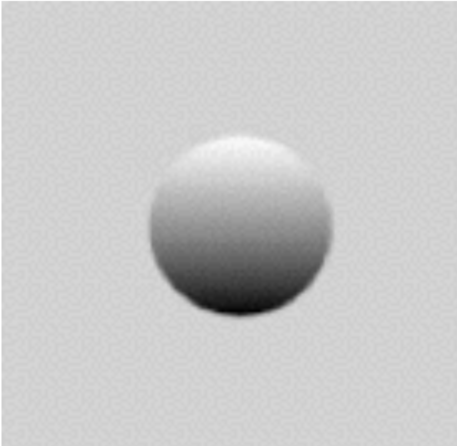
P(IN | image) **vs** P(OUT | image)

Applying Bayes rule: $\underbrace{P(\text{world} \mid \text{sense data})}_{\text{posterior}} \propto \underbrace{P(\text{sense data} \mid \text{world})}_{\text{likelihood}} \underbrace{P(\text{world})}_{\text{prior}}$

P(IN | image) = P(image | IN & light below) × P(IN) × P(light below)

vs

P(OUT | image) = P(image | OUT & light above) × P(OUT) × P(light above)



$P(\text{IN} \mid \text{image})$ **vs** $P(\text{OUT} \mid \text{image})$

Applying Bayes rule: $\overbrace{P(\text{world} \mid \text{sense data})}^{\text{posterior}} \propto \overbrace{P(\text{sense data} \mid \text{world})}^{\text{likelihood}} \overbrace{P(\text{world})}^{\text{prior}}$

$$P(\text{IN} \mid \text{image}) = P(\text{image} \mid \text{IN} \,\&\, \text{light below}) \times P(\text{IN}) \times P(\text{light below})$$

vs

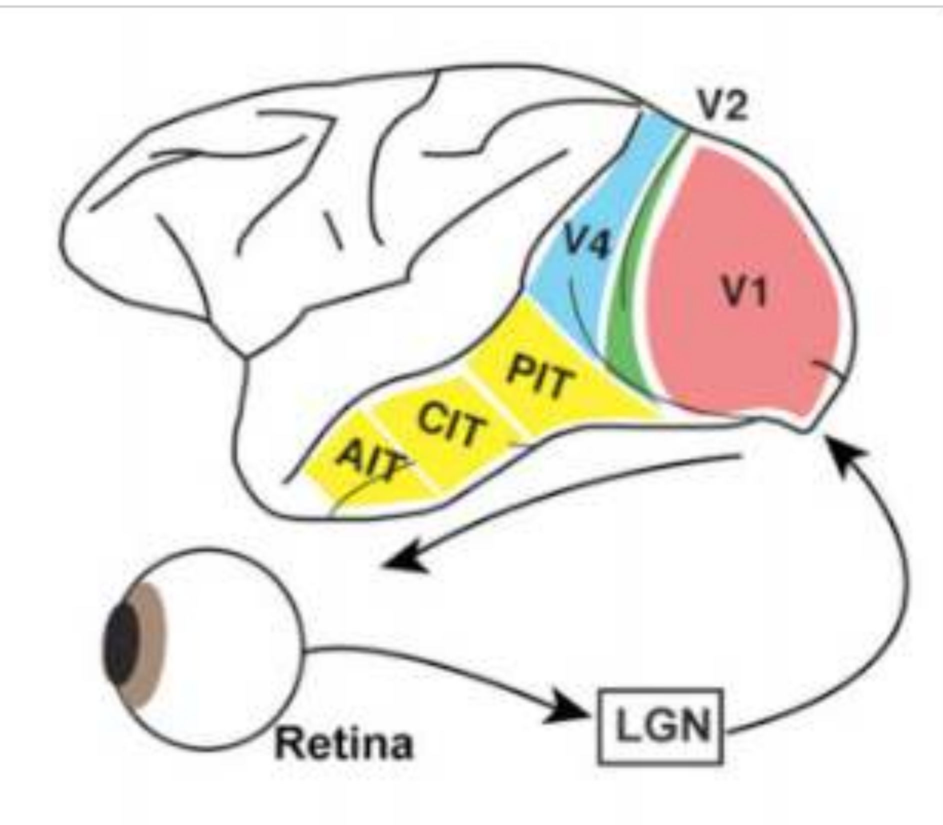
$$P(\text{OUT} \mid \text{image}) = P(\text{image} \mid \text{OUT} \,\&\, \text{light above}) \times P(\text{OUT}) \times P(\text{light above})$$

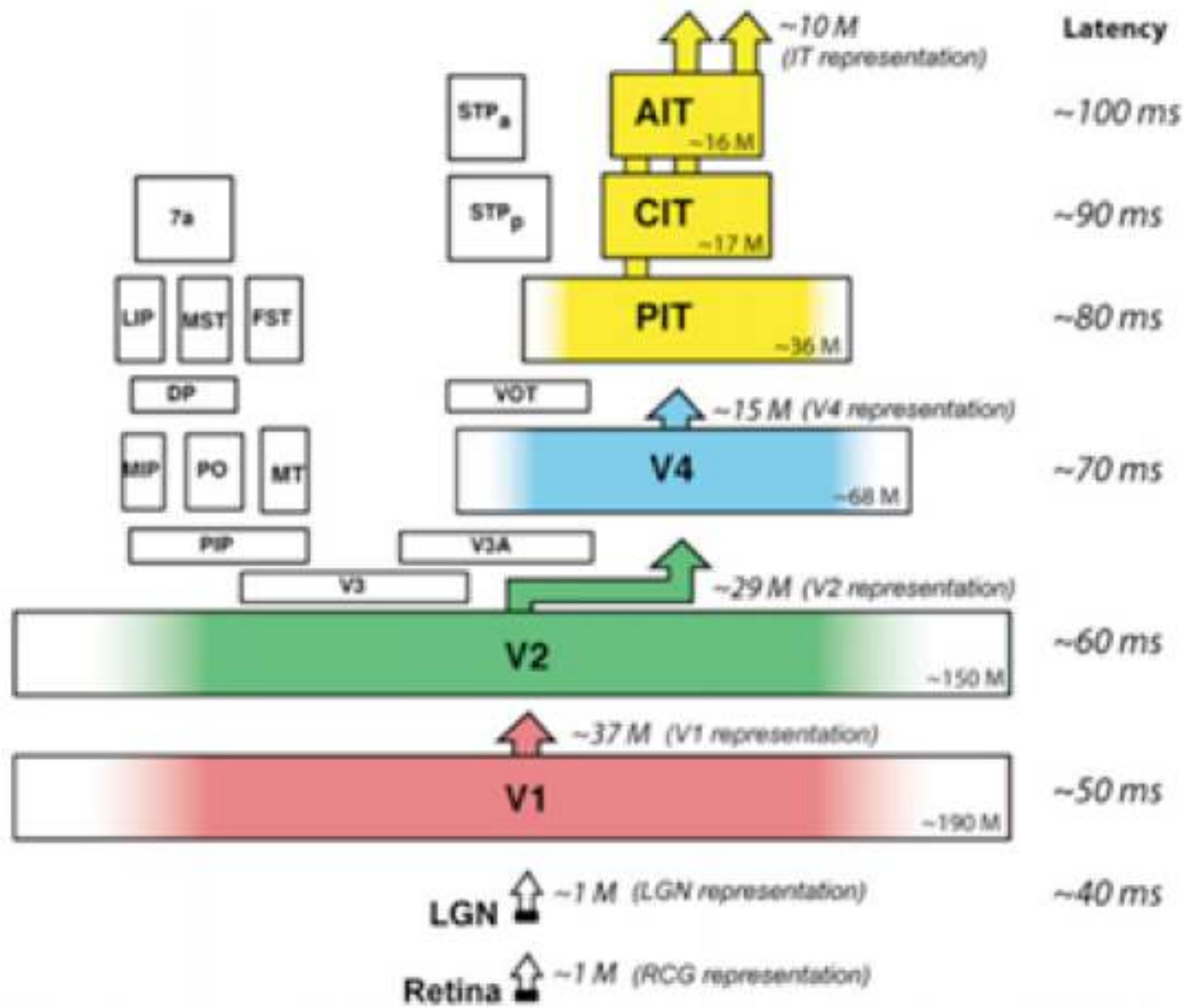
“OUT” is much more probable because Sun is usually up and your brain uses that fact automatically!

How the brain makes sense of the world?

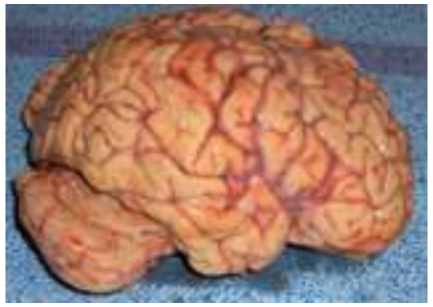
The brain takes into account “prior knowledge” to figure out what’s in the world given our sensory information

Primate Vision

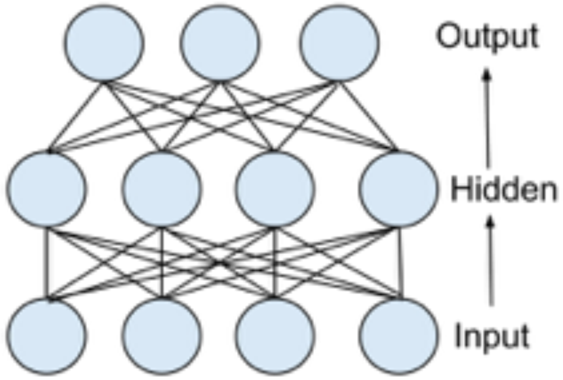




Brains 101



Deep Learning



Feynman dictum

Machine Learning: algorithms that learn from data

Machine Learning tasks

Supervised Learning

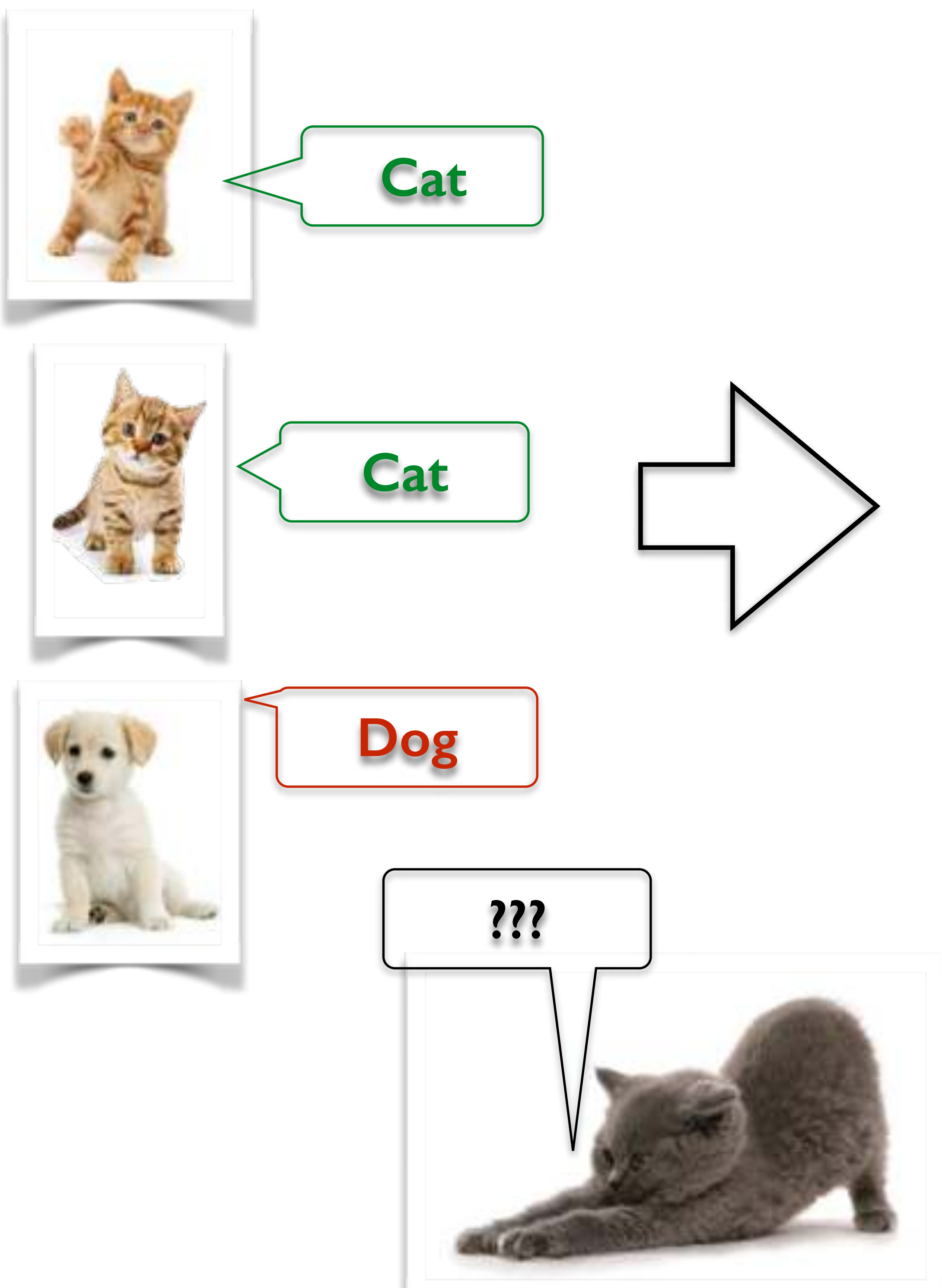
- Classification
- Regression

Unsupervised Learning

- Clustering
- Dimensionality reduction
- Density Estimation

Reinforcement Learning - taking actions in an environment to maximise reward

Supervised Learning



Features | **Labels**

$x1 = (87, 205, 255, 155, \ldots); \; y1 = 1$

$x2 = (95, 195, 245, 155, \ldots); \; y2 = 1$

$x3 = (43, 159, 255, 5, \ldots); \; y3 = -1$

Adopted from P.Vincent http://videolectures.net/deeplearning2015_vincent_machine_learning/

Supervised Learning

Features **Labels**

Cat

$x1 = (87, 205, 255, 155, \ldots);$ $y1 = 1$

Cat

$x2 = (95, 195, 245, 155, \ldots);$ $y2 = 1$

Dog

$x3 = (43, 159, 255, 5, \ldots);$ $y3 = -1$

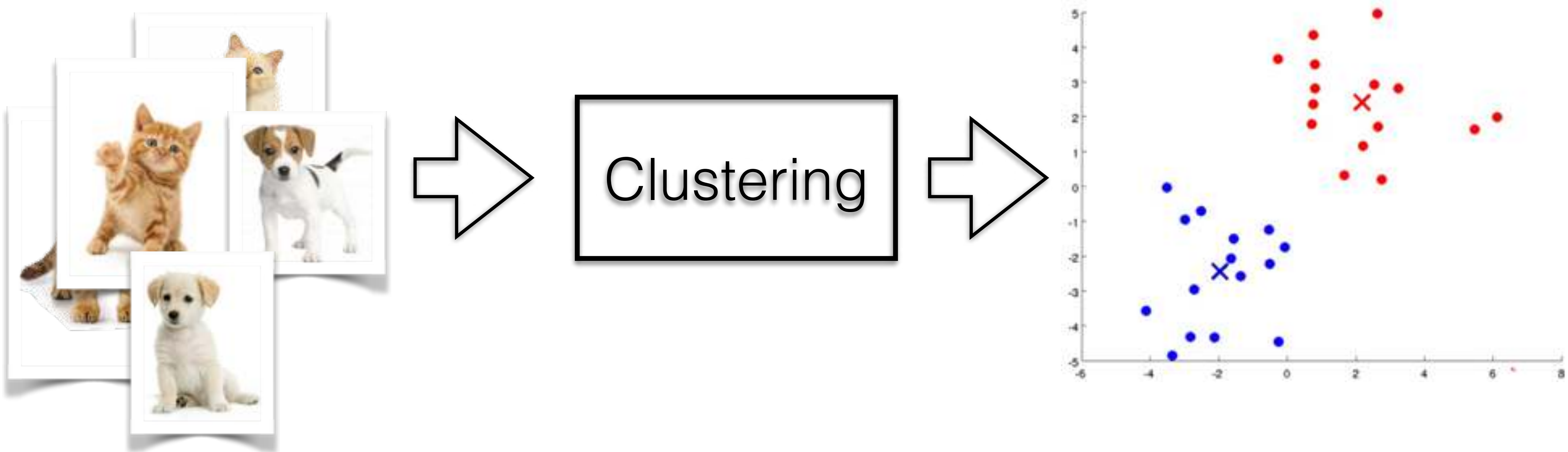
???

$x4 = (180, 66, 245, \ldots);$

$y4 = f(x4) = 1$

Adopted from P.Vincent http://videolectures.net/deeplearning2015_vincent_machine_learning/

Unsupervised Learning



Reinforcement Learning

Machine Learning tasks

Supervised Learning

- Classification
- Regression

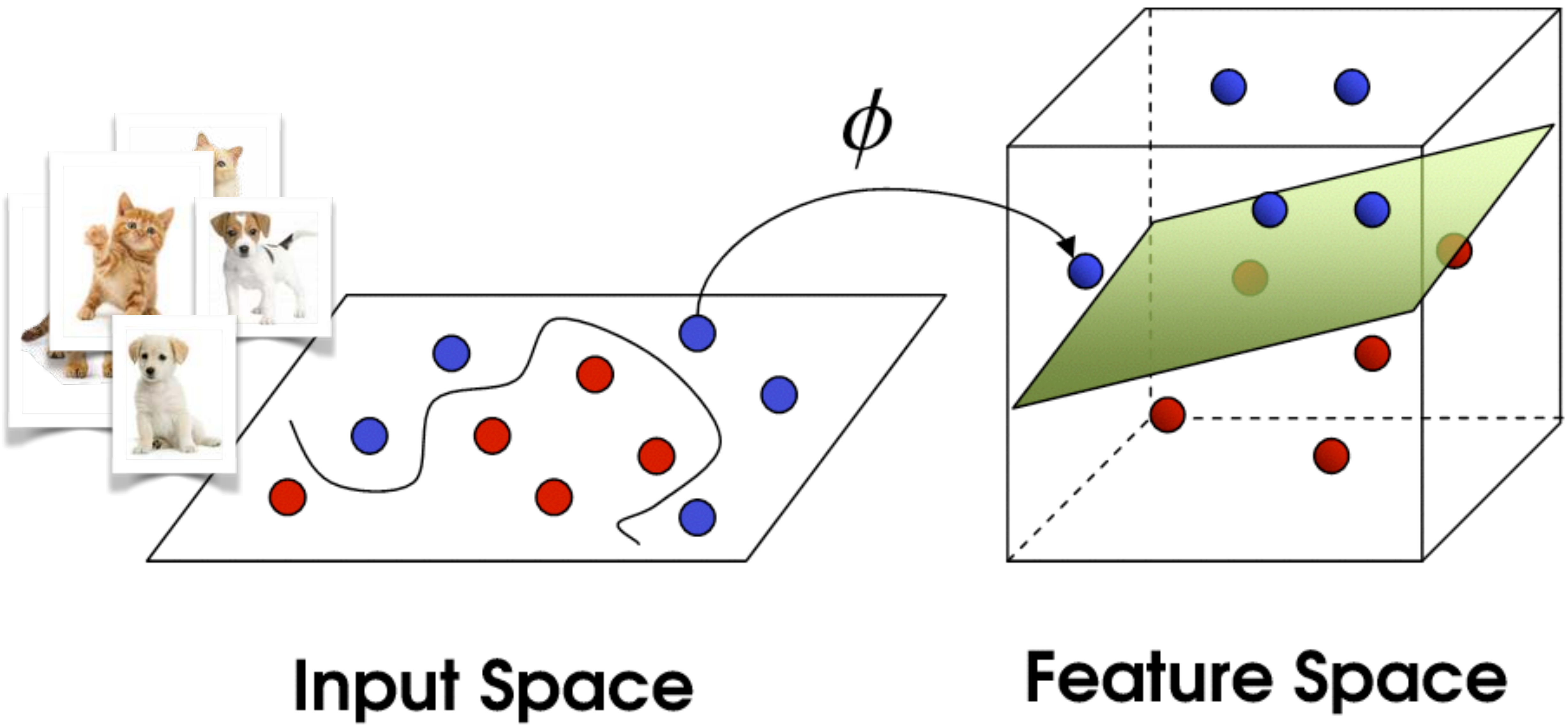
Unsupervised Learning

- Clustering
- Dimensionality reduction
- Density Estimation

Reinforcement Learning - taking actions in an environment to maximise reward

Shallow Machine Learning

0 or 1 abstraction layer (feature transformation)

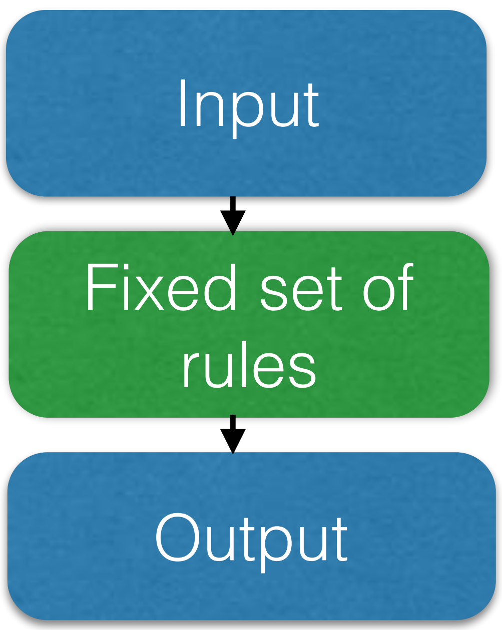


Evolution of ML methods

Adopted from Y.Bengio http://videolectures.net/deeplearning2015_bengio_theoretical_motivations/

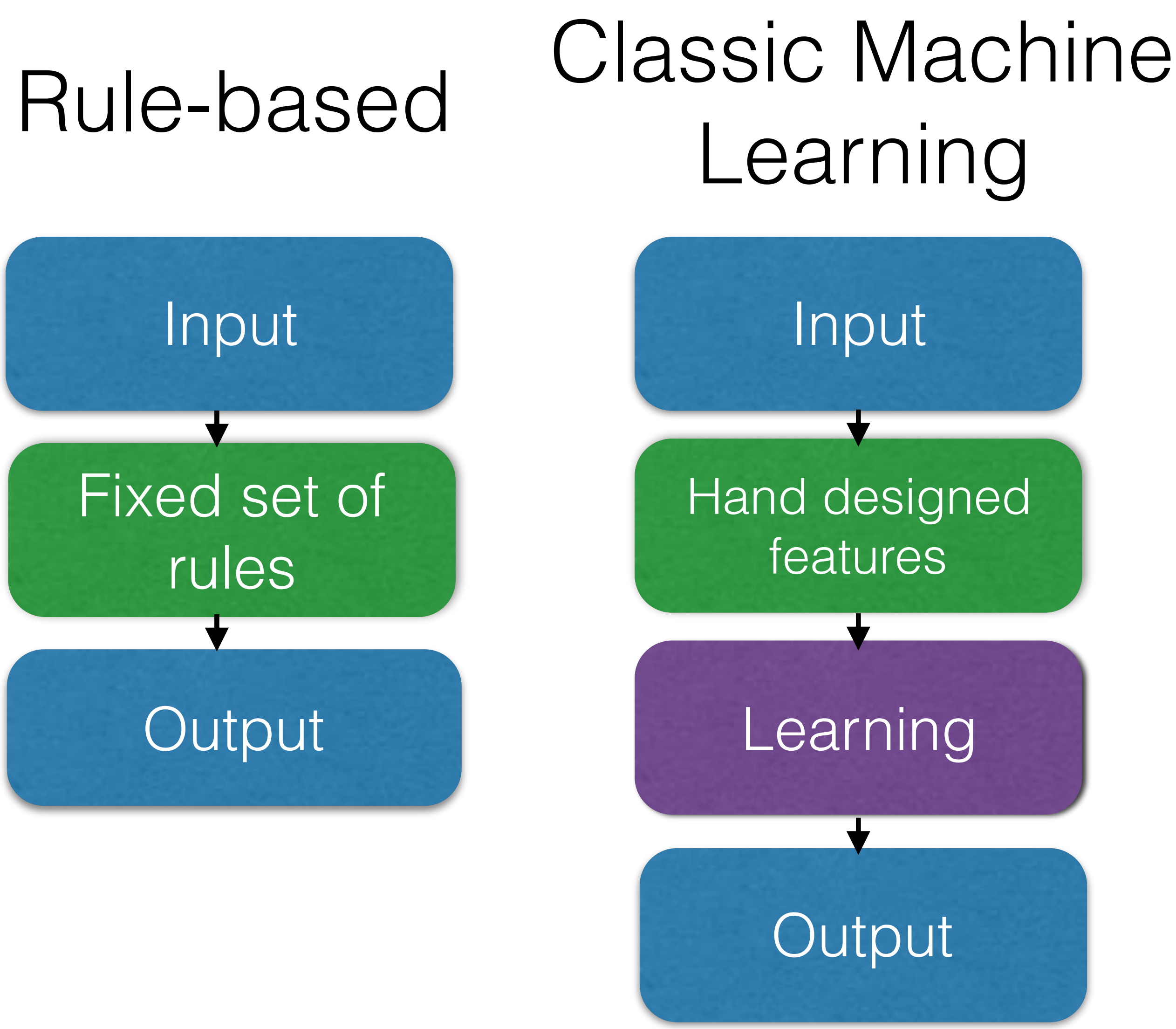
Evolution of ML methods

Rule-based



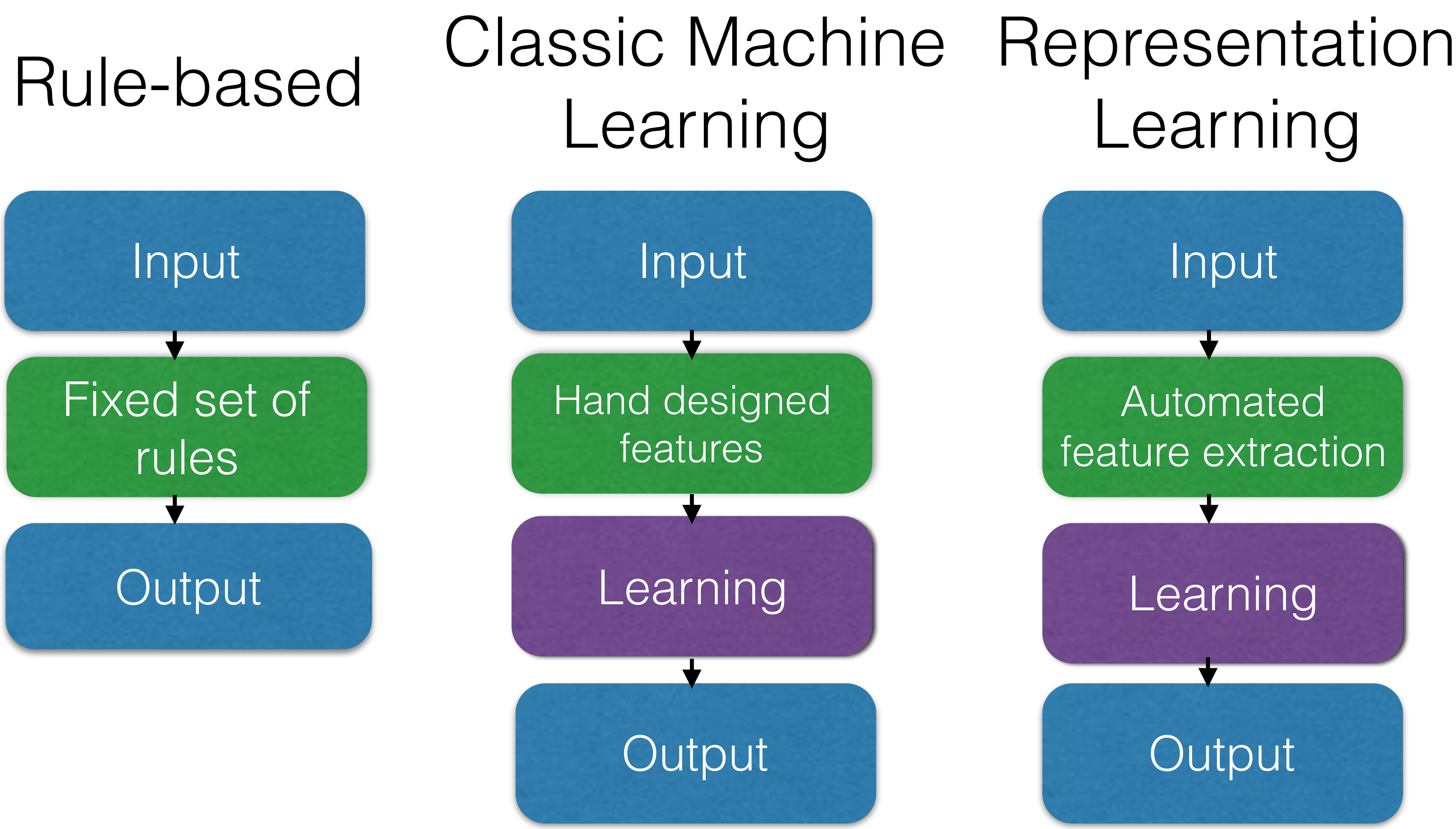
Adopted from Y.Bengio http://videolectures.net/deeplearning2015_bengio_theoretical_motivations/

Evolution of ML methods



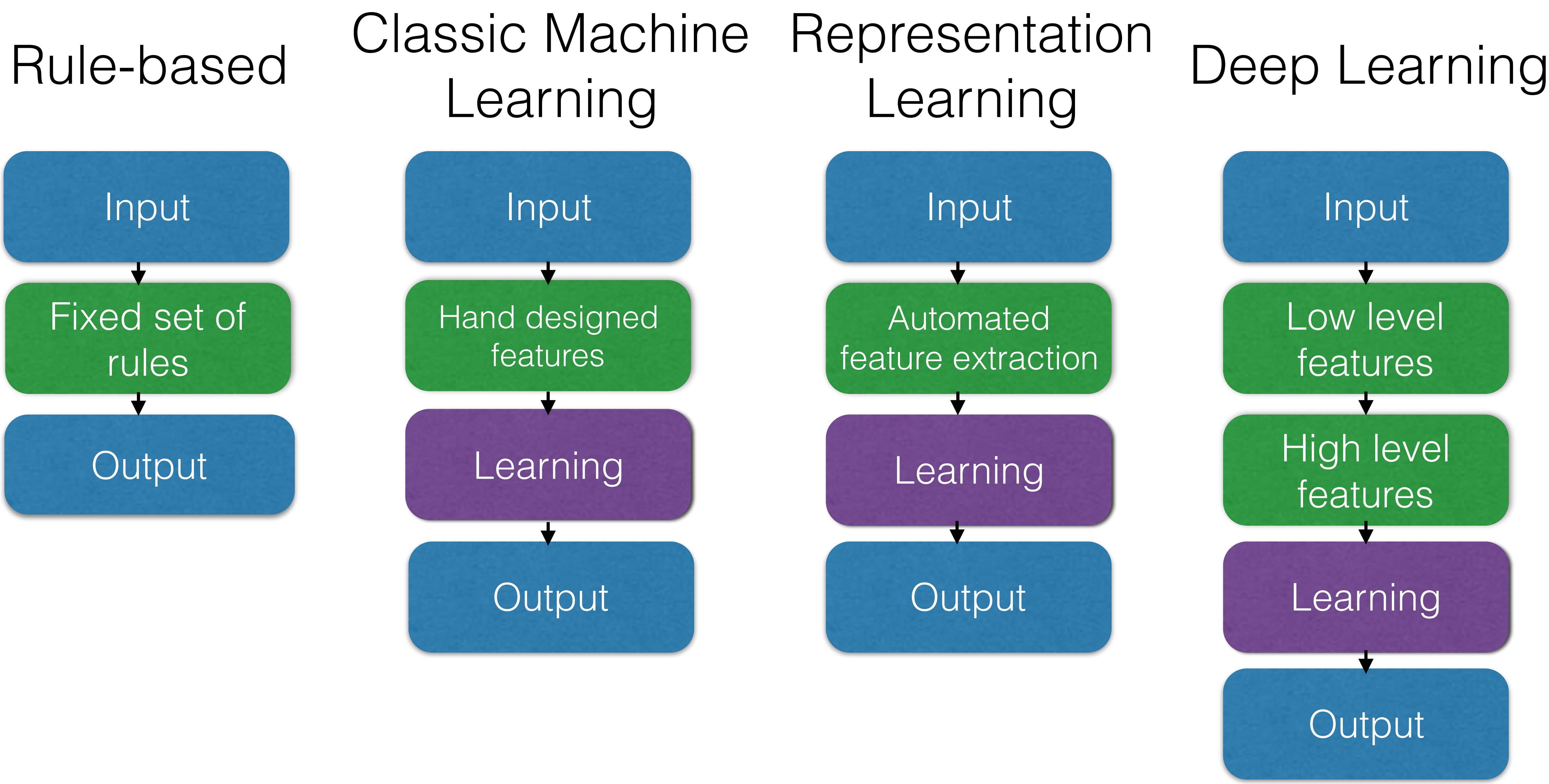
Adopted from Y.Bengio http://videolectures.net/deeplearning2015_bengio_theoretical_motivations/

Evolution of ML methods



Adopted from Y.Bengio http://videolectures.net/deeplearning2015_bengio_theoretical_motivations/

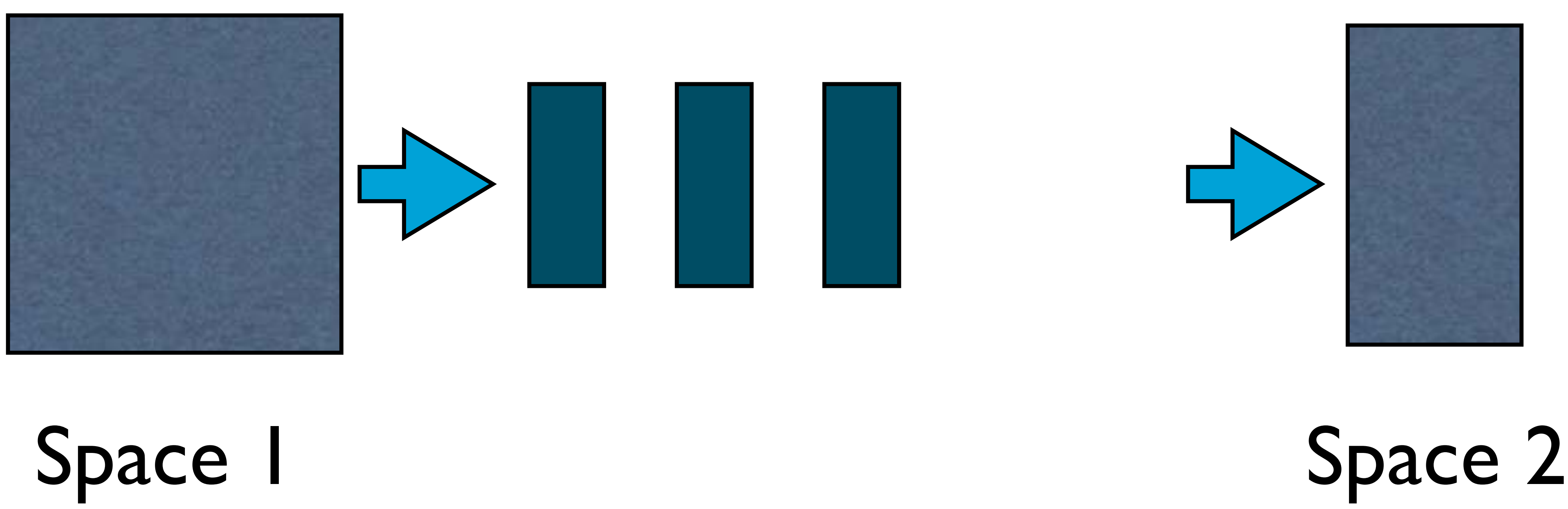
Evolution of ML methods



Adopted from Y.Bengio http://videolectures.net/deeplearning2015_bengio_theoretical_motivations/

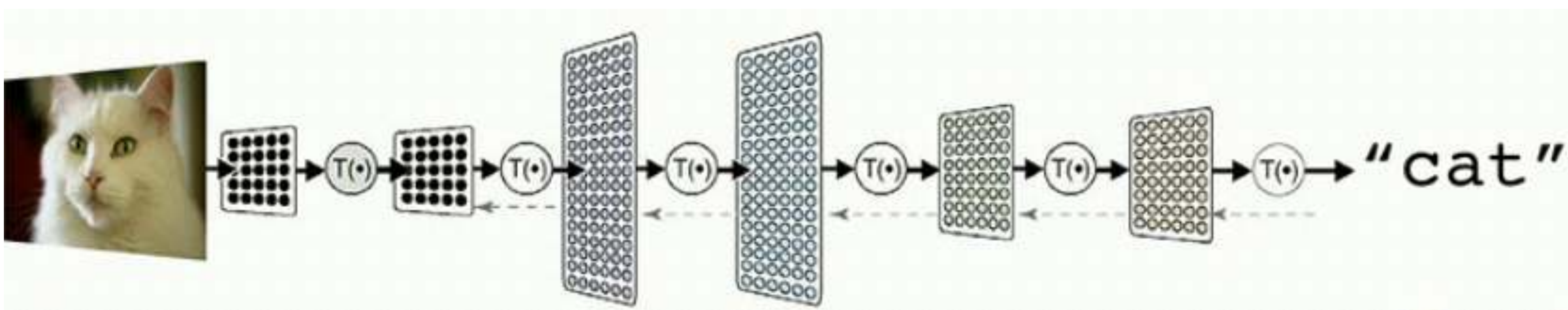
What is deep learning?

many layers of adaptive non-linear processing to model complex relationships among data



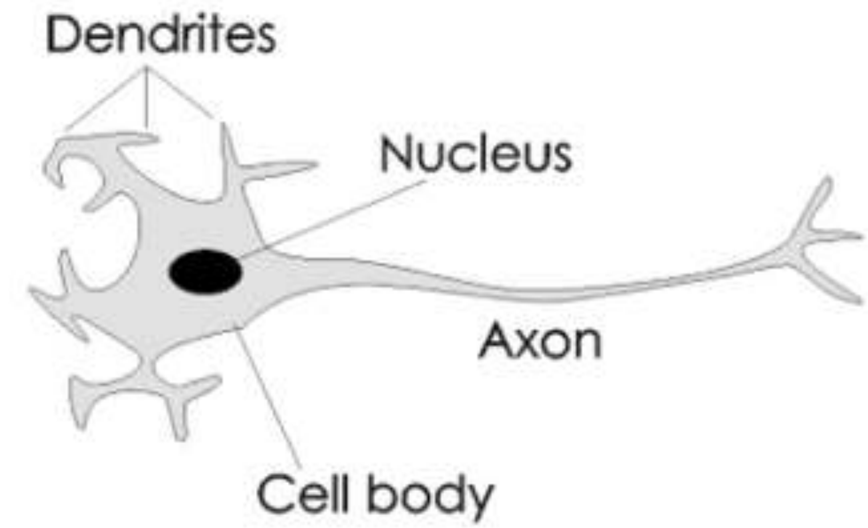
In practice

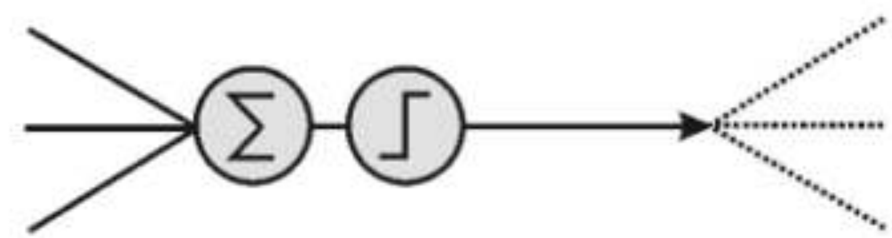
DL = Artificial Neural Networks with many layers



McCulloch & Pitts (1943)

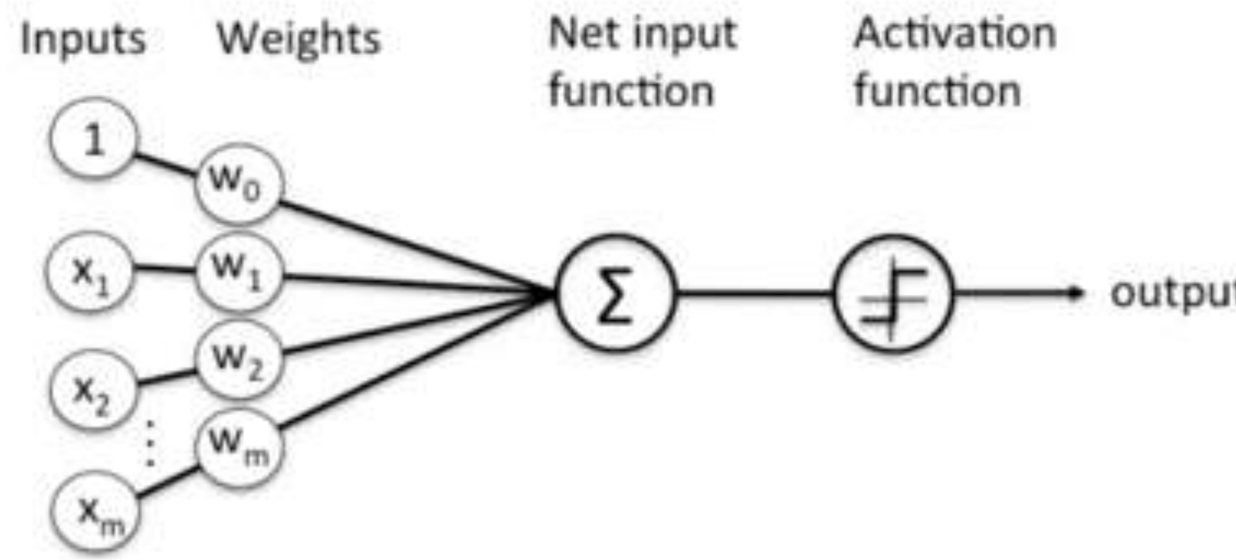
A Logical Calculus of the Ideas Immanent in Nervous Activity





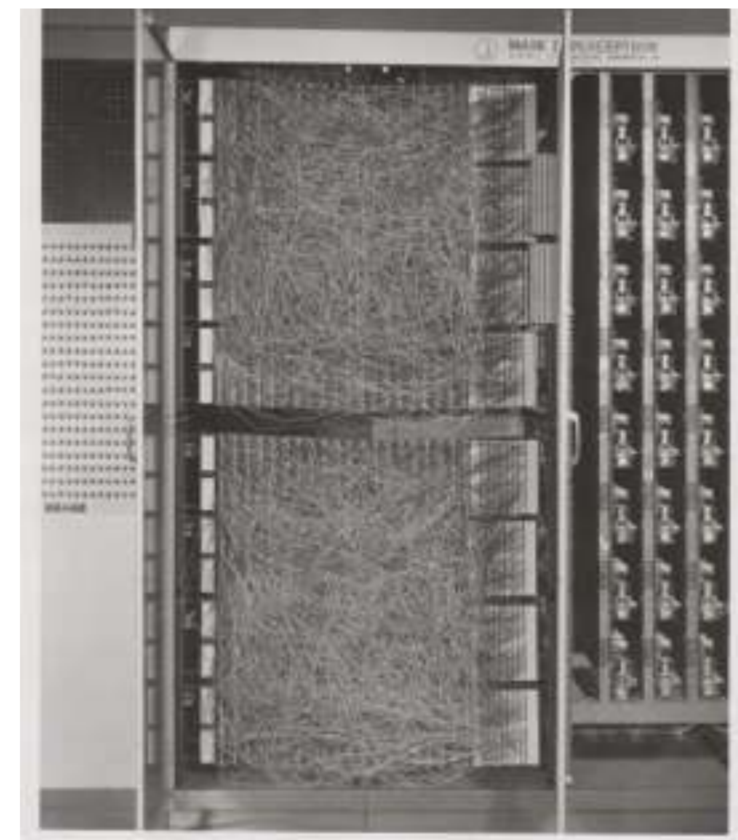
McCulloch & Pitts (1943)

A Logical Calculus of the Ideas Immanent in Nervous Activity



Rosenblatt (1957)

Perceptron



New York Times: "(The perceptron) is the embryo of an electronic computer that is expected to be able to walk, talk, see, write, reproduce itself and be conscious of its existence"

McCulloch & Pitts (1943)

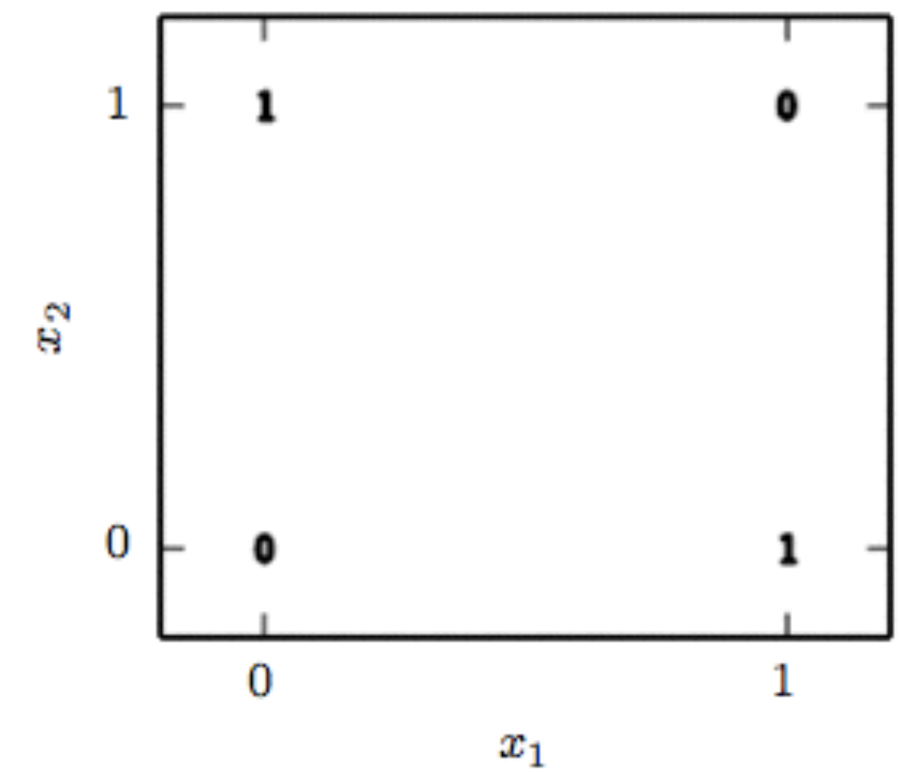
A Logical Calculus of the Ideas Immanent in Nervous Activity

Rosenblatt (1957)

Perceptron

Minsky & Papert (1969)

Perceptrons: an introduction to computational geometry



McCulloch & Pitts (1943)

A Logical Calculus of the Ideas Immanent in Nervous Activity

Rosenblatt (1957)

Perceptron

Minsky & Papert (1969)

Perceptrons: an introduction to computational geometry

Blum & Rivest (1992)

Training a 3-node neural network is NP-complete

McCulloch & Pitts (1943)

A Logical Calculus of the Ideas

Immanent in Nervous Activity

Rosenblatt (1957)

Perceptron

Minsky & Papert (1969)

Perceptrons: an introduction to computational geometry

Blum & Rivest (1992)

Training a 3-node neural network is NP-complete

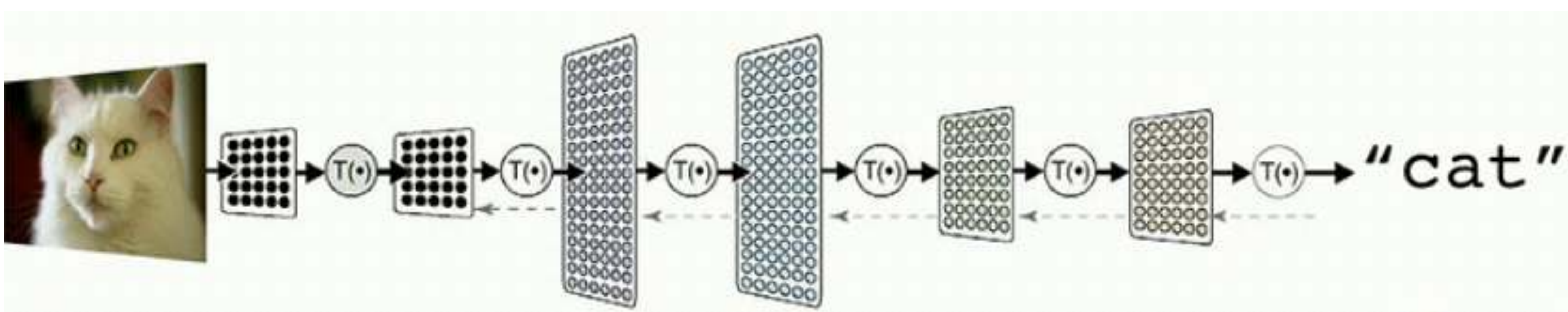
Werbos (1974)

Rumelhart, Hinton & Williams (1986)

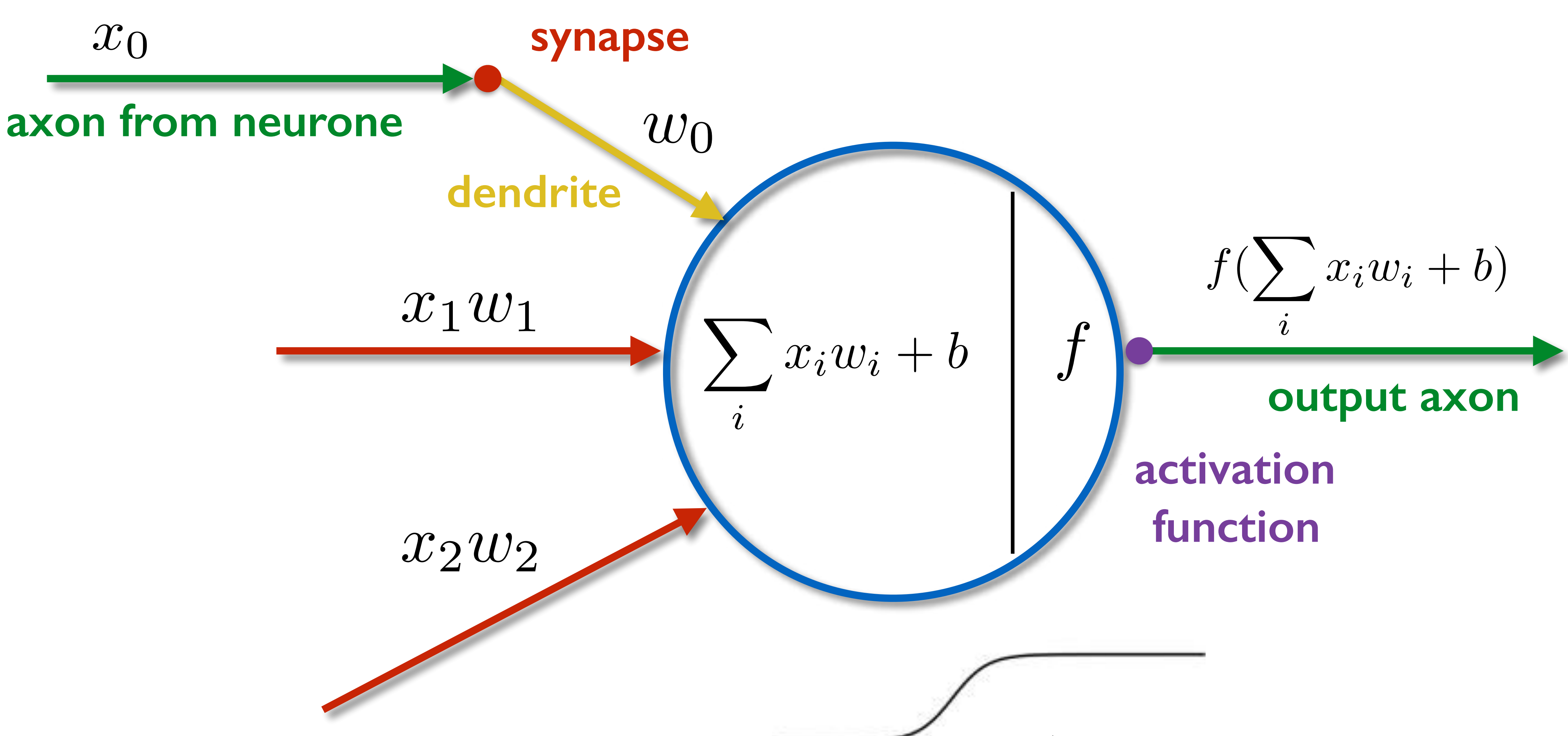
Learning representations by back-propagating errors

Artificial neural network

- A collection of simple trainable mathematical units, which collaborate to compute a complicated function
- Compatible with supervised, unsupervised, and reinforcement
- Brain inspired (loosely)



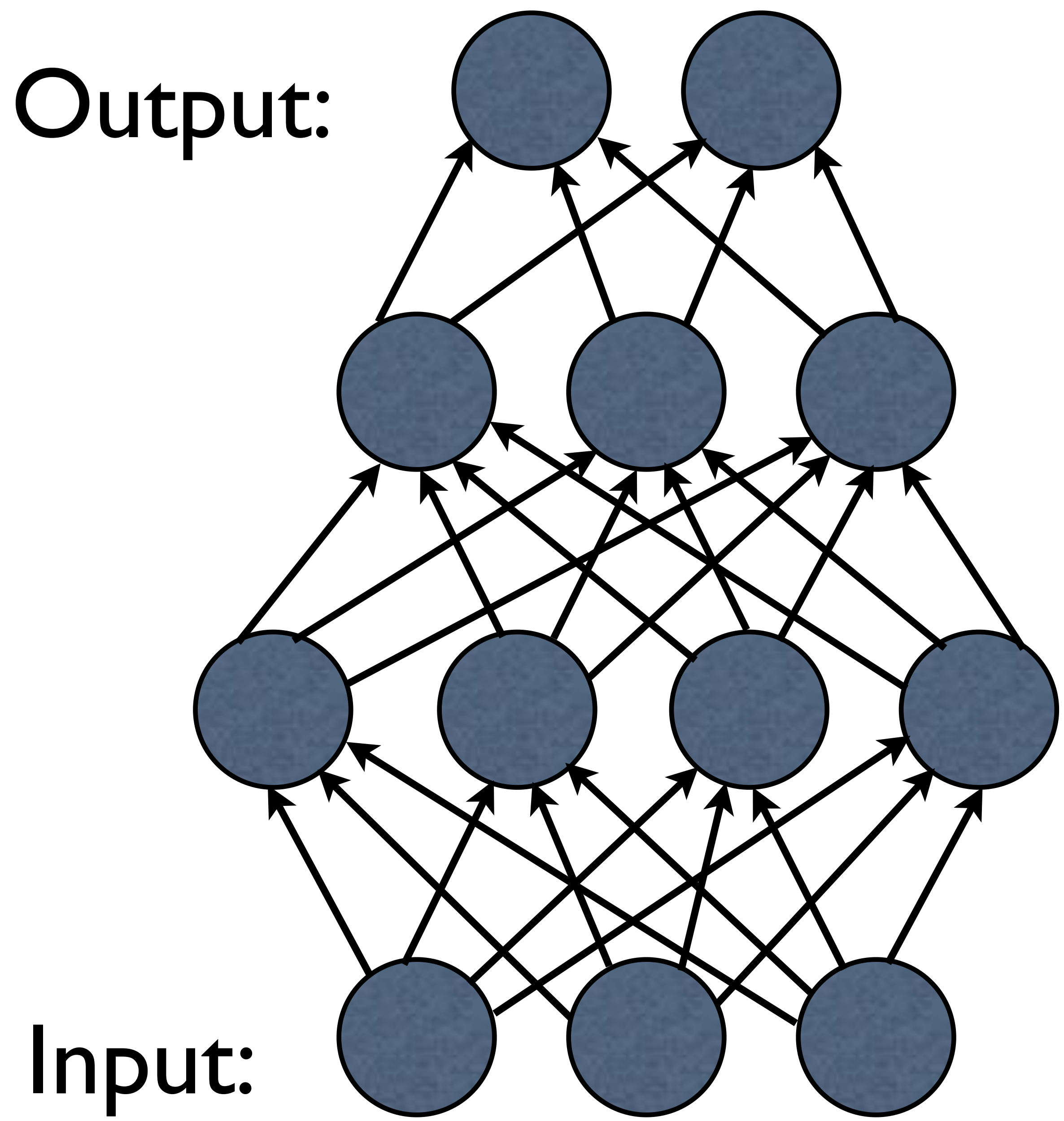
Artificial Neuron



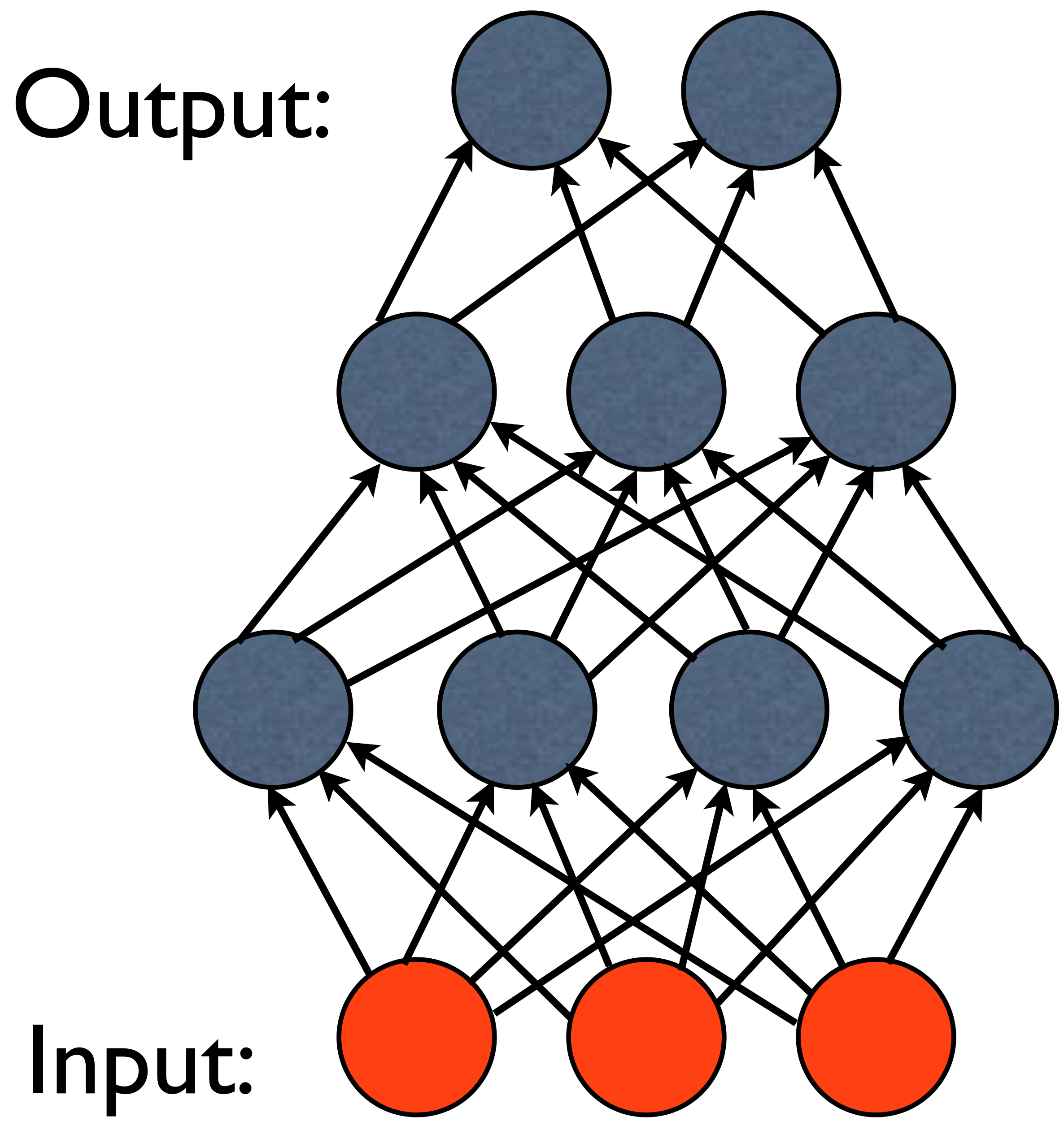
$$F(x) = \frac{1}{1 + \exp(-x)}$$

Slide from CS231n: Convolutional Neural Networks for Visual Recognition http://cs231n.github.io/ by Andrej Karpathy

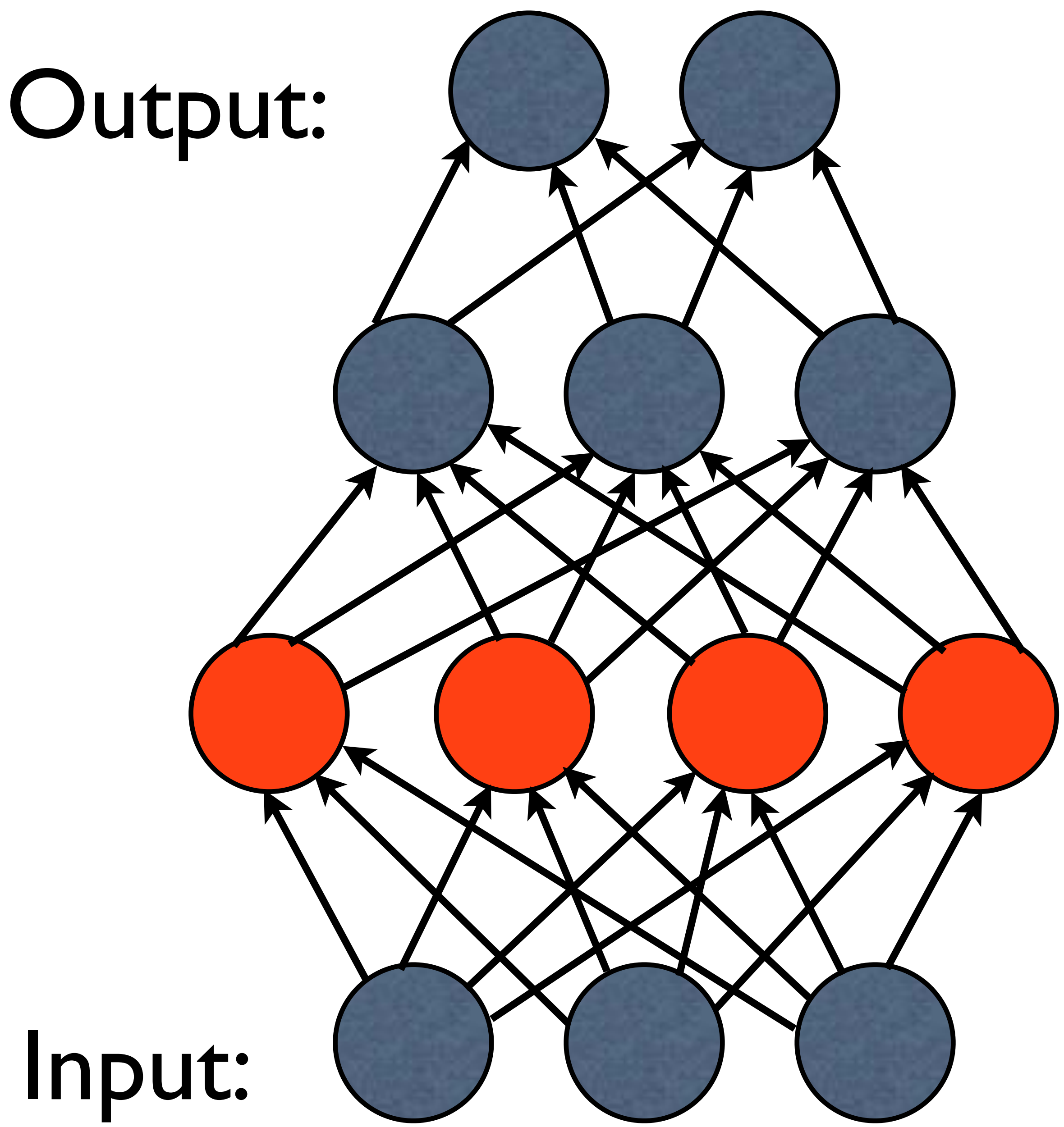
Neural networks



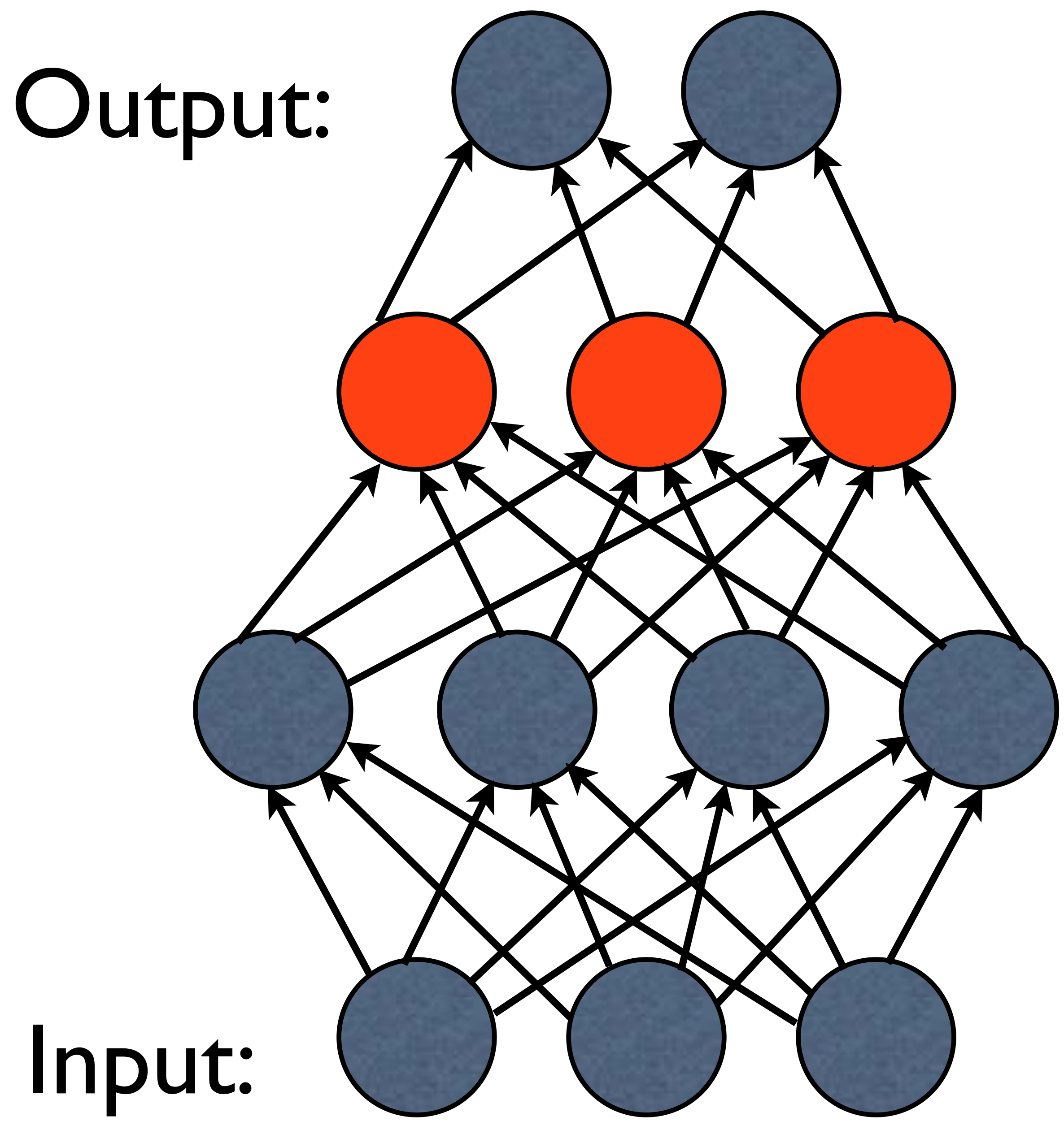
Neural networks



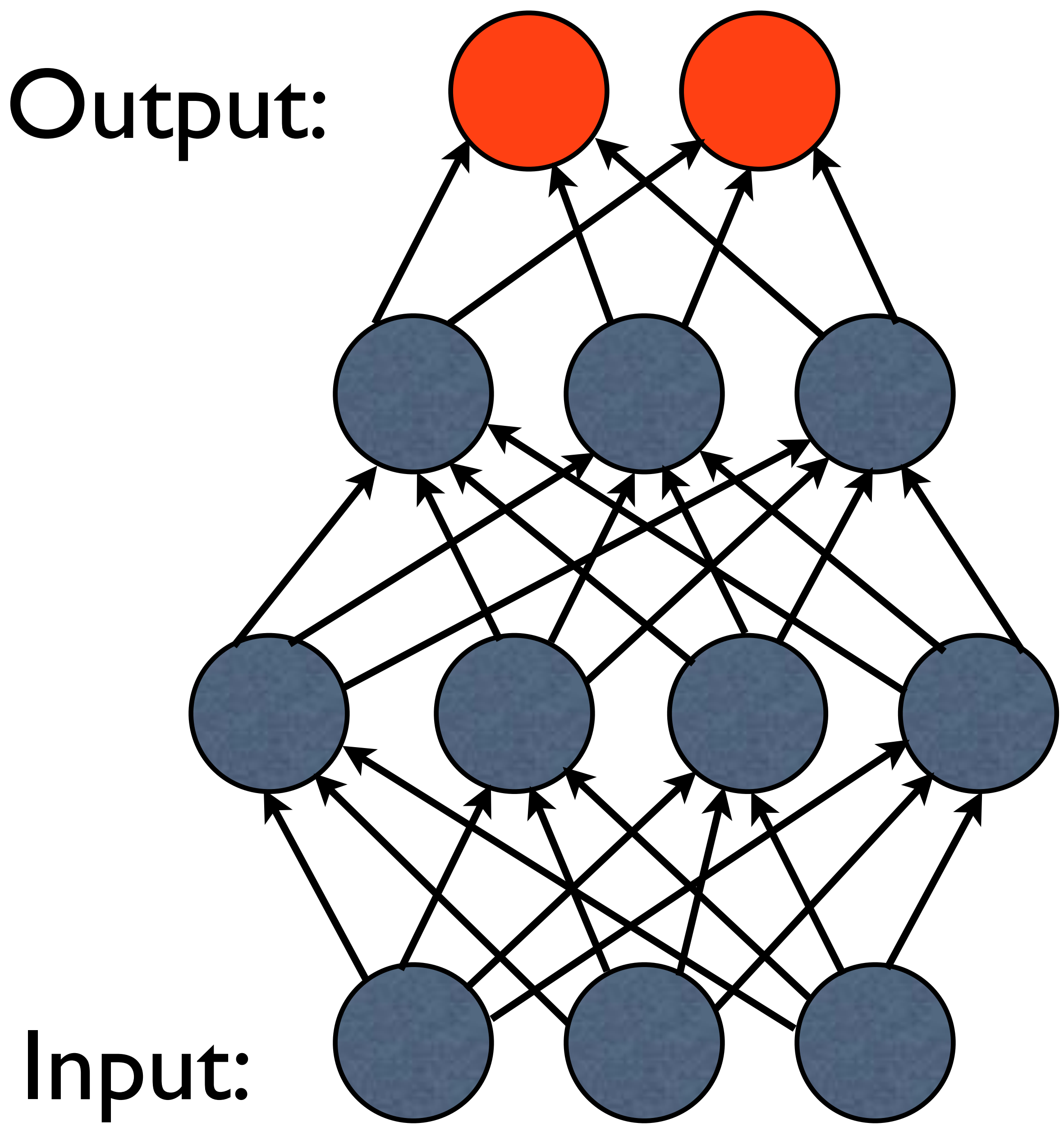
Neural networks



Neural networks



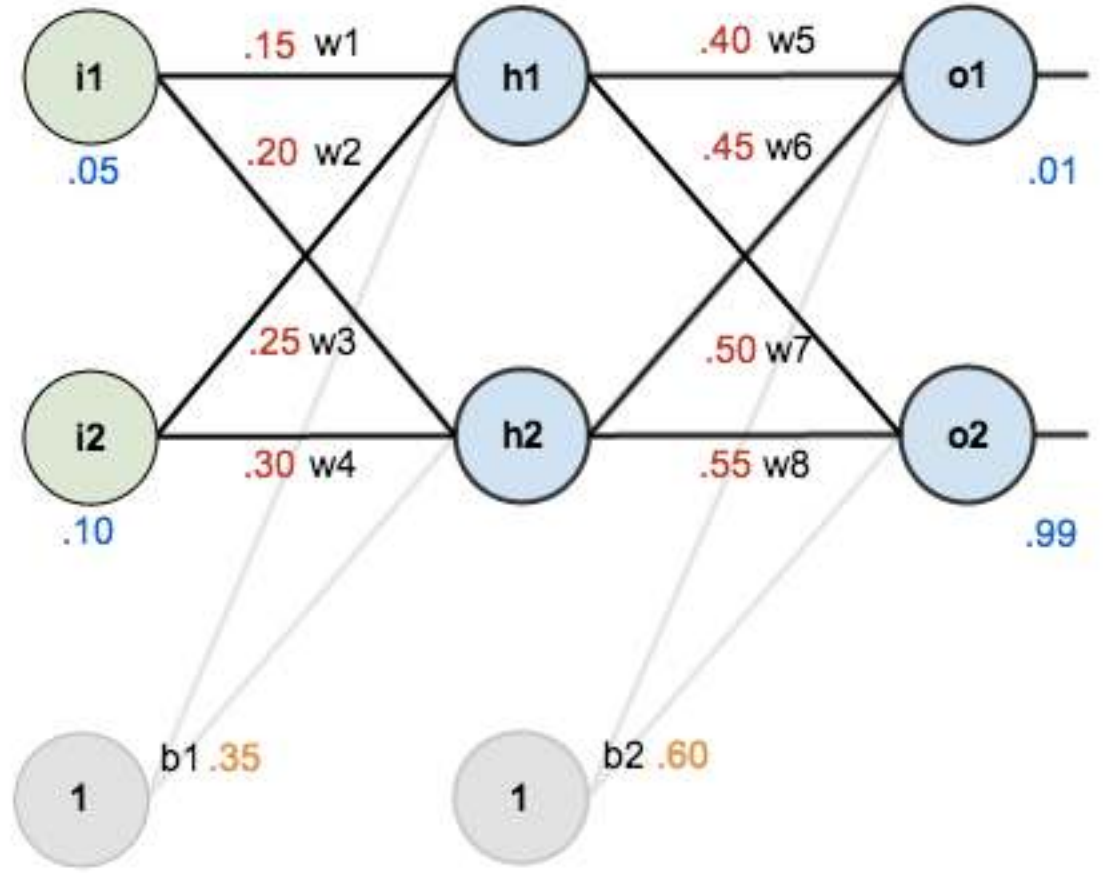
Neural networks



Learning algorithm

- **while** not done
  - pick a random training case (**x**, **y**)
  - run neuronal network on input **x**
  - modify connection weights to make prediction closer to **y**

Lets backpropagate

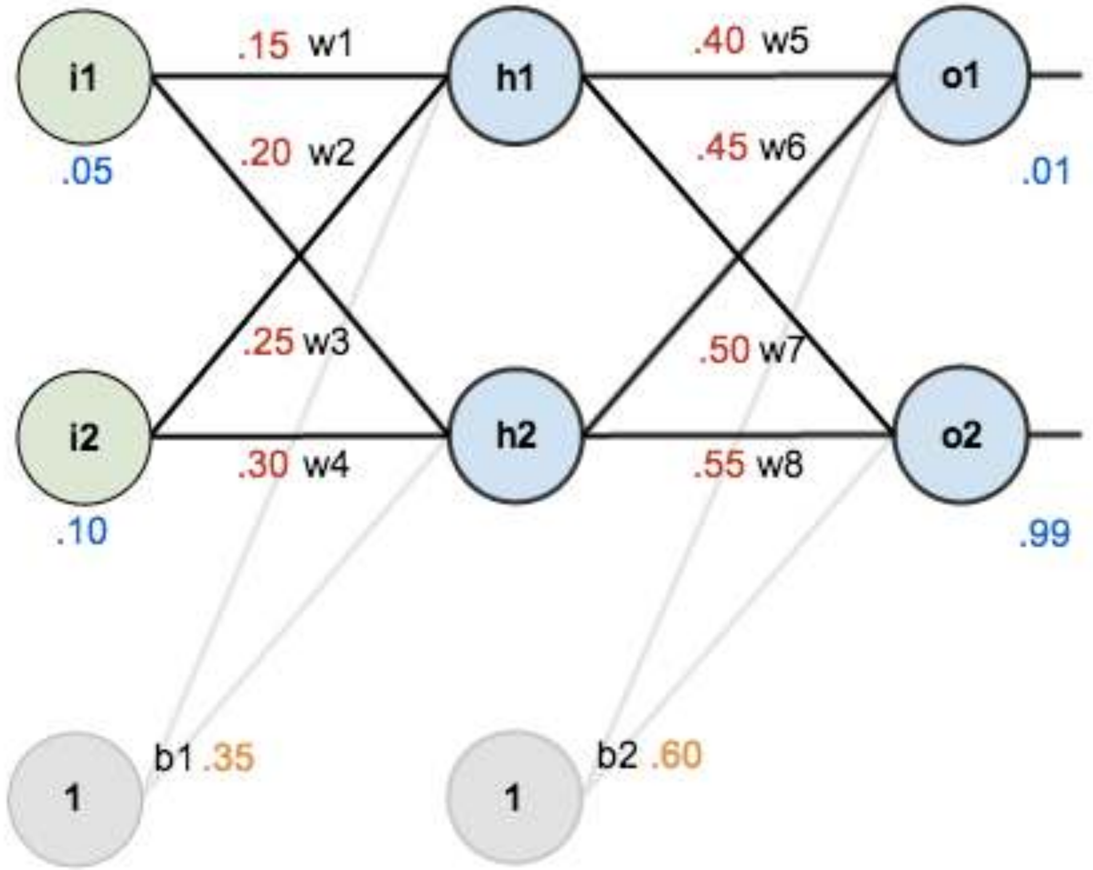


INPUT	TARGET
i1 = 0.05	o1 = 0.01
i2 = 0.10	o2 = 0.99

1. The Forward pass - Compute total error

$$net_{h1} = w_1 * i_1 + w_2 * i_2 + b_1 * 1$$

Lets backpropagate



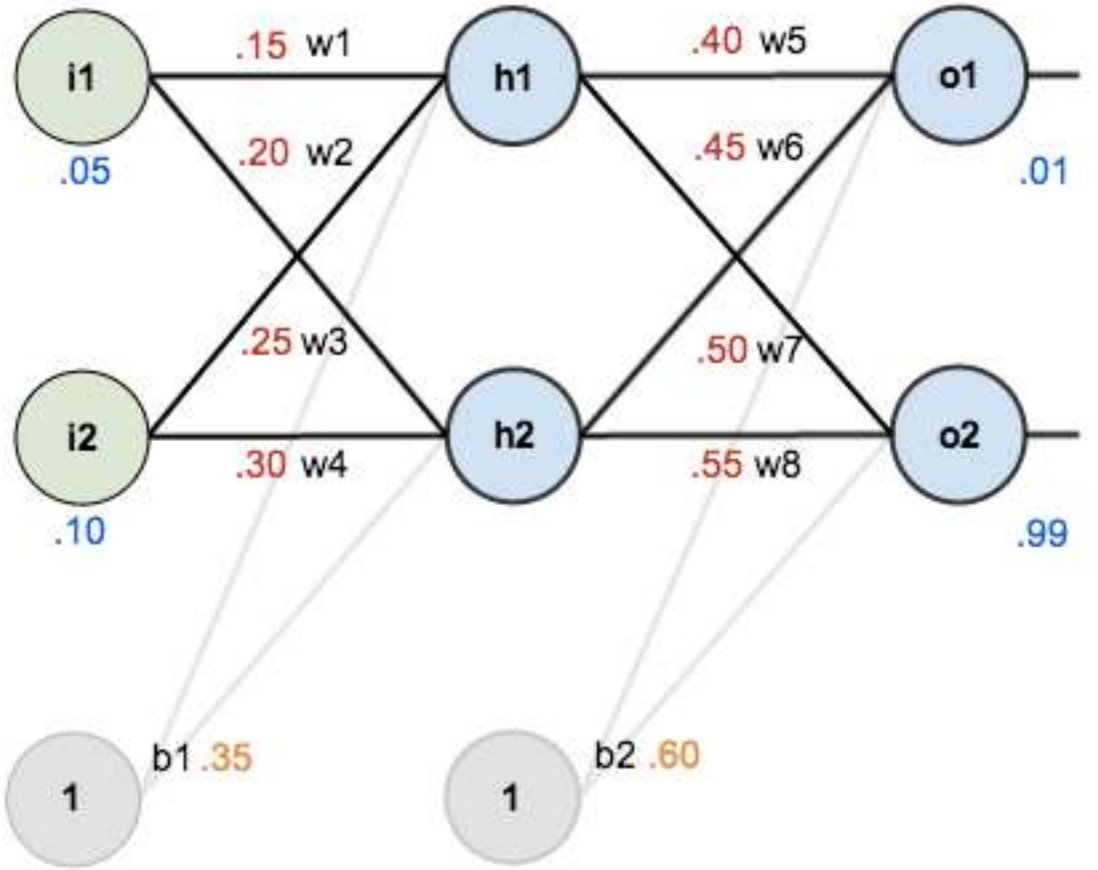
INPUT	TARGET
i1 = 0.05	o1 = 0.01
i2 = 0.10	o2 = 0.99

1. The Forward pass - Compute total error

$$net_{h1} = w_1 * i_1 + w_2 * i_2 + b_1 * 1$$

$$net_{h1} = 0.15 * 0.05 + 0.2 * 0.1 + 0.35 * 1 = 0.3775$$

Lets backpropagate



INPUT

i1 = 0.05

i2 = 0.10

TARGET

o1 = 0.01

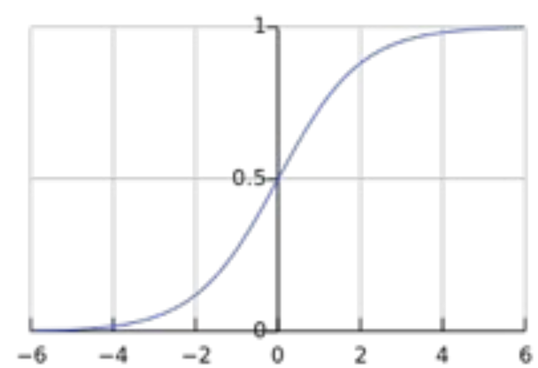
o2 = 0.99

1. The Forward pass - Compute total error

$$net_{h1} = w_1 * i_1 + w_2 * i_2 + b_1 * 1$$

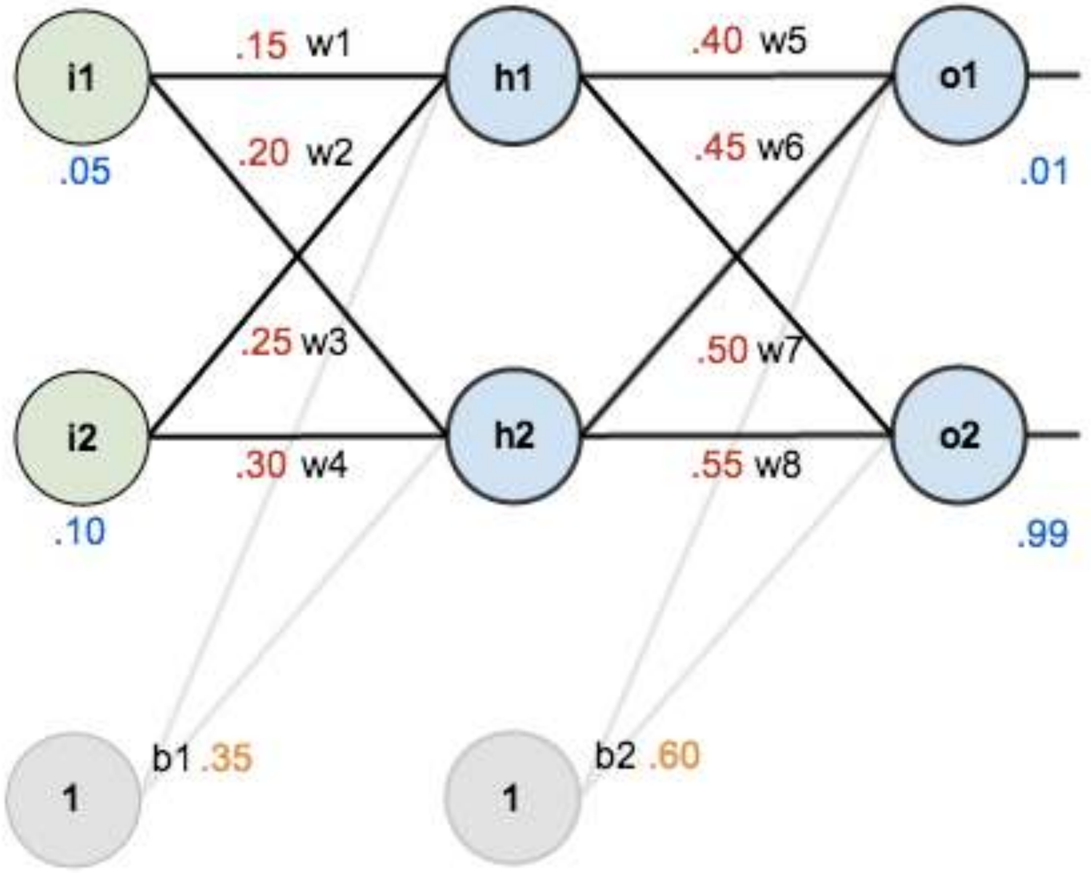
$$net_{h1} = 0.15 * 0.05 + 0.2 * 0.1 + 0.35 * 1 = 0.3775$$

$$out_{h1} = \frac{1}{1 + e^{-net_{h1}}} = \frac{1}{1 + e^{-0.3775}} = 0.5933$$



$$f(x) = \frac{1}{1 + e^{-x}}$$

Lets backpropagate



INPUT

i1 = 0.05

i2 = 0.10

TARGET

o1 = 0.01

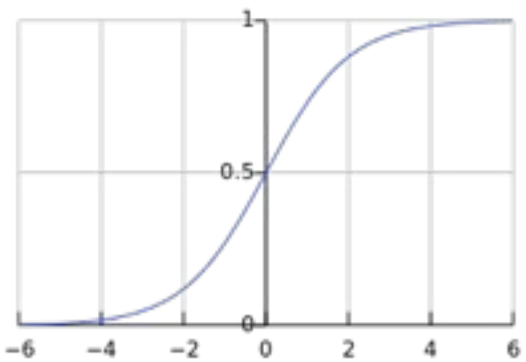
o2 = 0.99

1. The Forward pass - Compute total error

$$net_{h1} = w_1 * i_1 + w_2 * i_2 + b_1 * 1$$

$$net_{h1} = 0.15 * 0.05 + 0.2 * 0.1 + 0.35 * 1 = 0.3775$$

$$out_{h1} = \frac{1}{1 + e^{-net_{h1}}} = \frac{1}{1 + e^{-0.3775}} = 0.5933$$

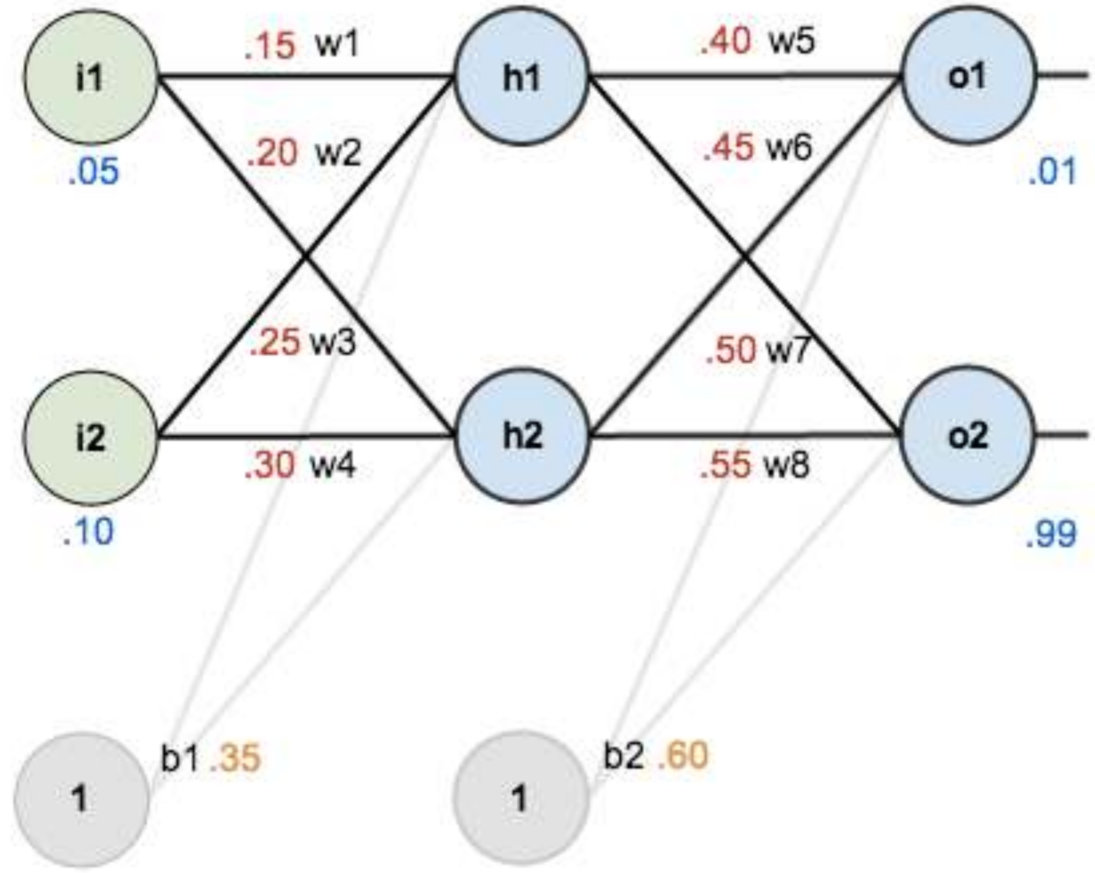


$$f(x) = \frac{1}{1 + e^{-x}}$$

Repeat for h2 = 0.596; o1 = 0.751; o2 = 0.773

Lets backpropagate

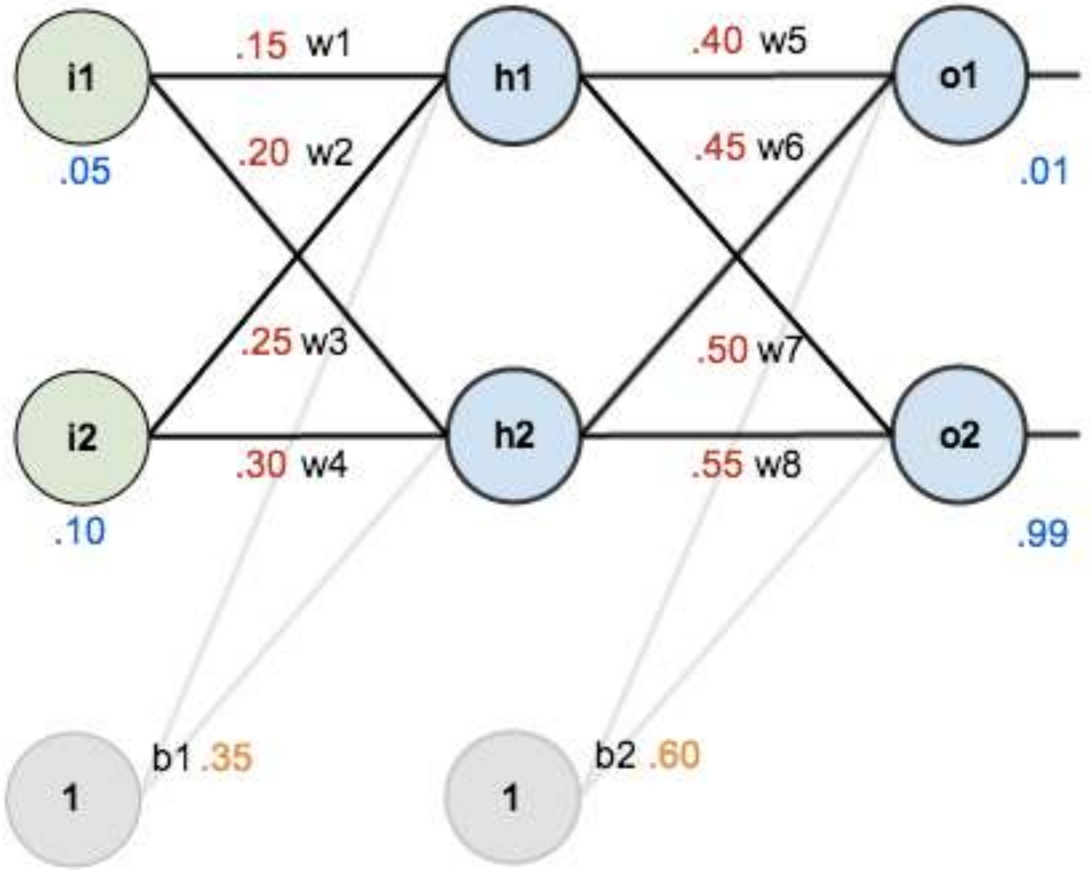
INPUT	TARGET
i1 = 0.05	o1 = 0.01
i2 = 0.10	o2 = 0.99



1. The Forward pass - Compute total error

We have o1, o2

Lets backpropagate



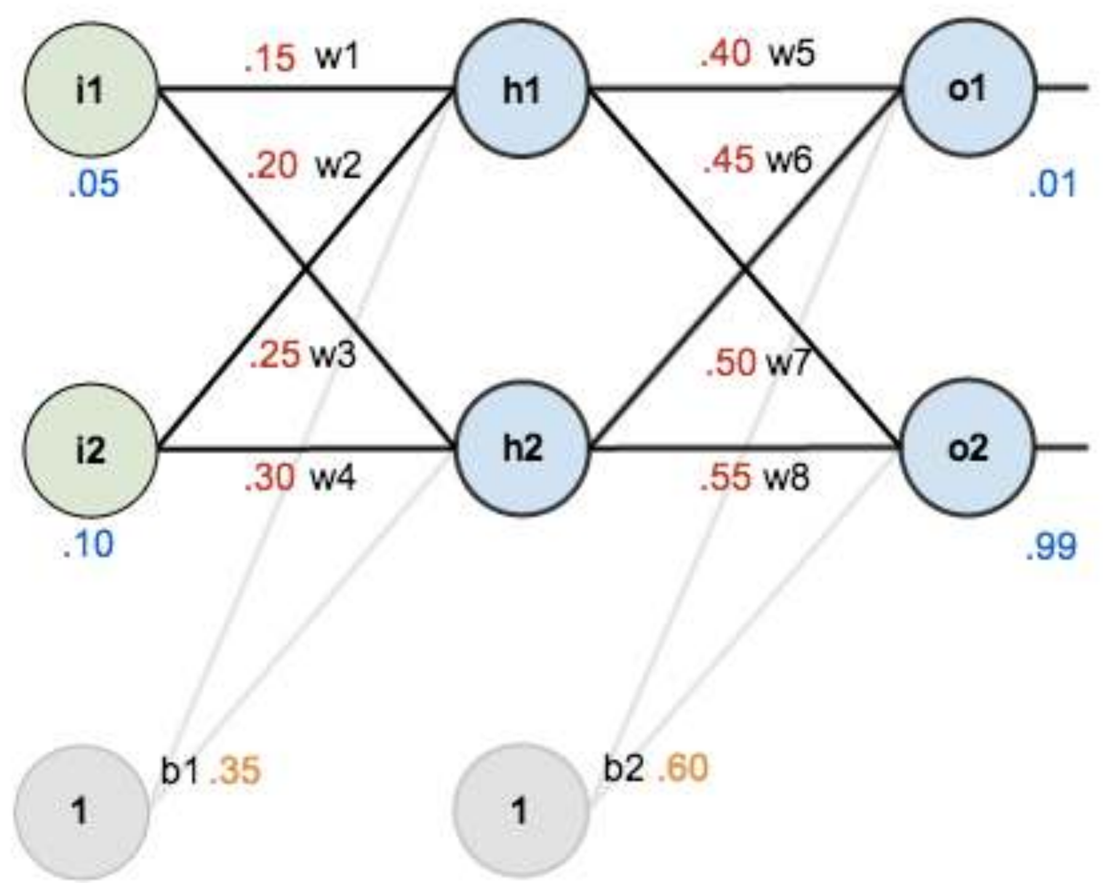
INPUT	TARGET
i1 = 0.05	o1 = 0.01
i2 = 0.10	o2 = 0.99

1. The Forward pass - Compute total error

We have o1, o2

$$E_{total} = \sum \frac{1}{2}(target - output)^2$$

Lets backpropagate



INPUT	TARGET
i1 = 0.05	o1 = 0.01
i2 = 0.10	o2 = 0.99

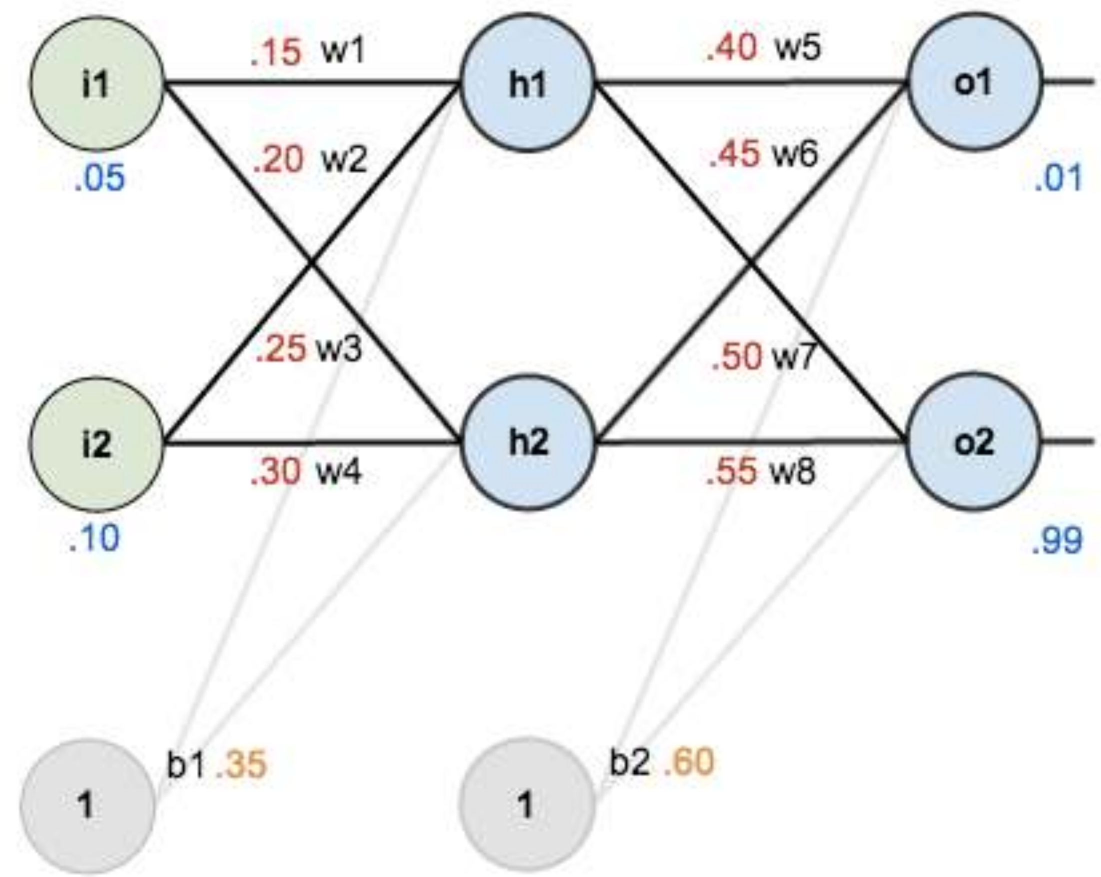
1. The Forward pass - Compute total error

We have o1, o2

$$E_{total} = \sum \frac{1}{2}(target - output)^2$$

$$E_{o1} = \frac{1}{2}(target_{o1} - out_{o1})^2 = \frac{1}{2}(0.01 - 0.7514)^2 = 0.2748$$

Lets backpropagate



INPUT	TARGET
i1 = 0.05	o1 = 0.01
i2 = 0.10	o2 = 0.99

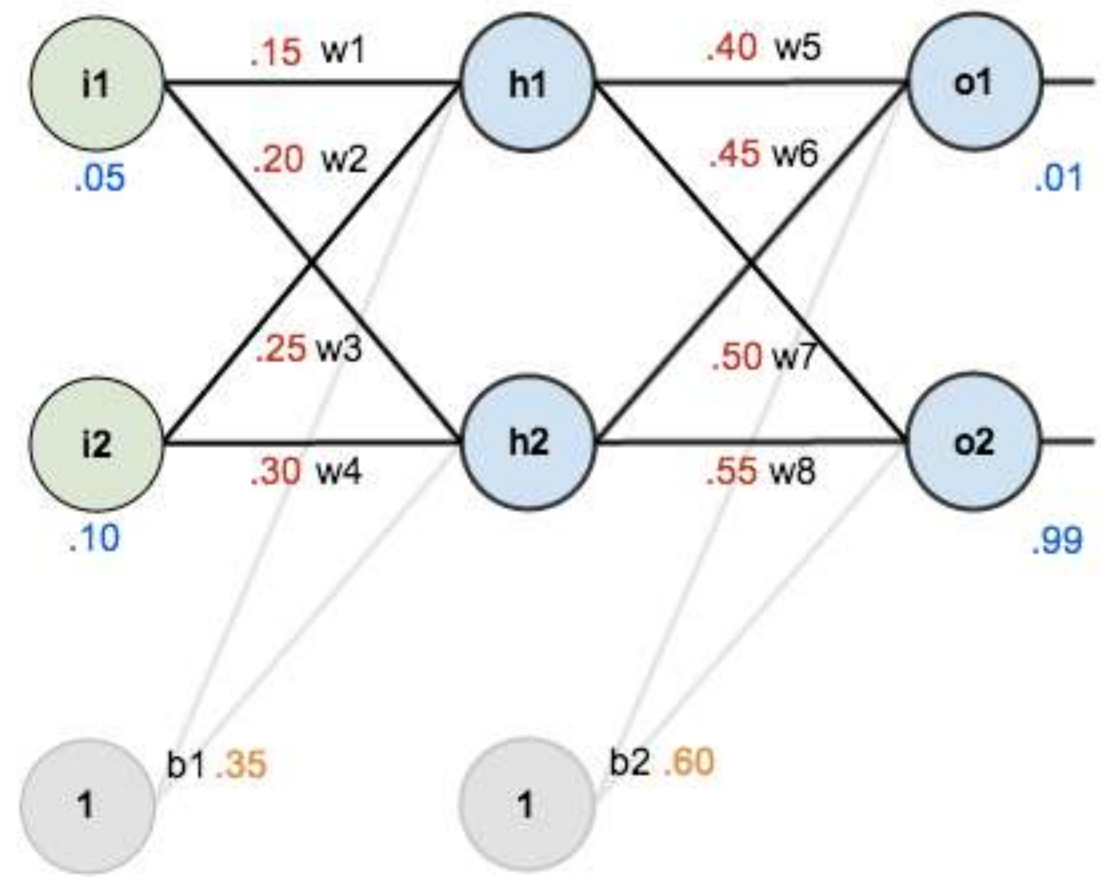
1. The Forward pass - Compute total error

We have o1, o2 $$E_{total} = \sum \frac{1}{2}(target - output)^2$$

$$E_{o1} = \frac{1}{2}(target_{o1} - out_{o1})^2 = \frac{1}{2}(0.01 - 0.7514)^2 = 0.2748$$

$$E_{o2} = 0.02356$$

Lets backpropagate



INPUT	TARGET
i1 = 0.05	o1 = 0.01
i2 = 0.10	o2 = 0.99

1. The Forward pass - Compute total error

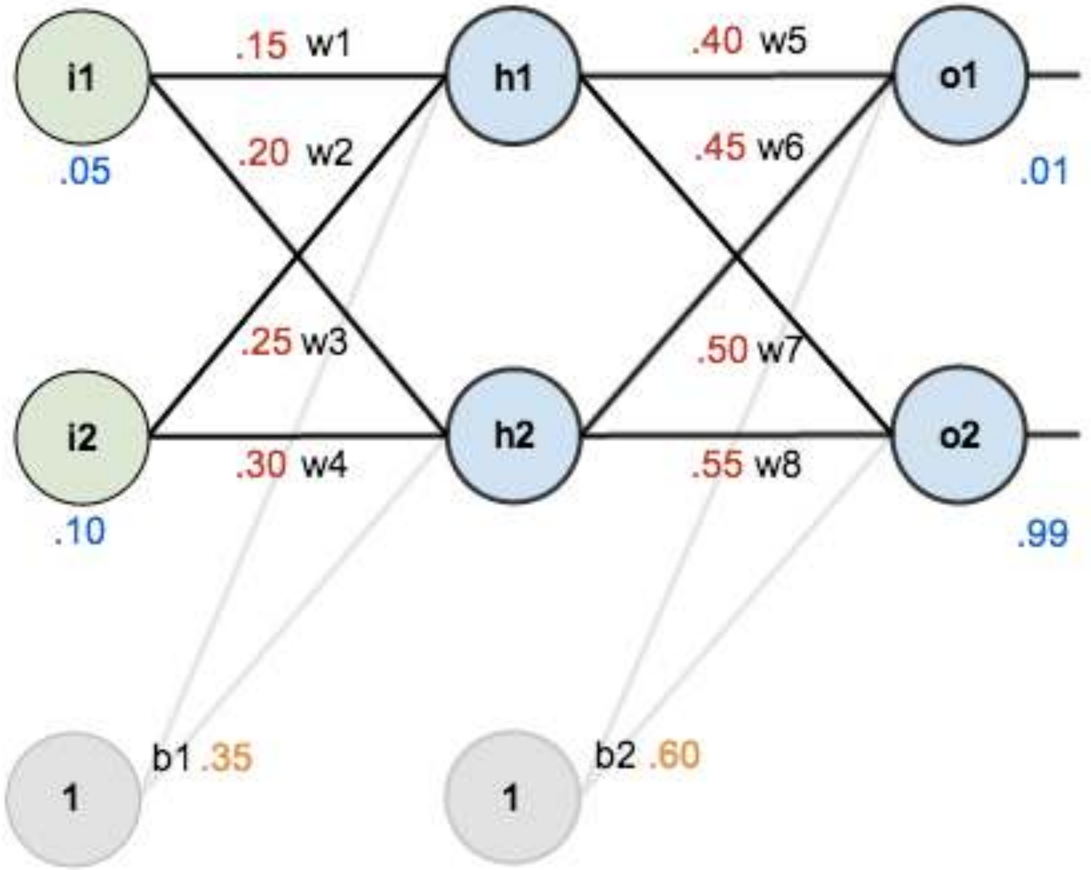
We have o1, o2 $$E_{total} = \sum \frac{1}{2}(target - output)^2$$

$$E_{o1} = \frac{1}{2}(target_{o1} - out_{o1})^2 = \frac{1}{2}(0.01 - 0.7514)^2 = 0.2748$$

$$E_{o2} = 0.02356$$

$$E_{total} = E_{o1} + E_{o2} = 0.2748 + 0.02356 = 0.29836$$

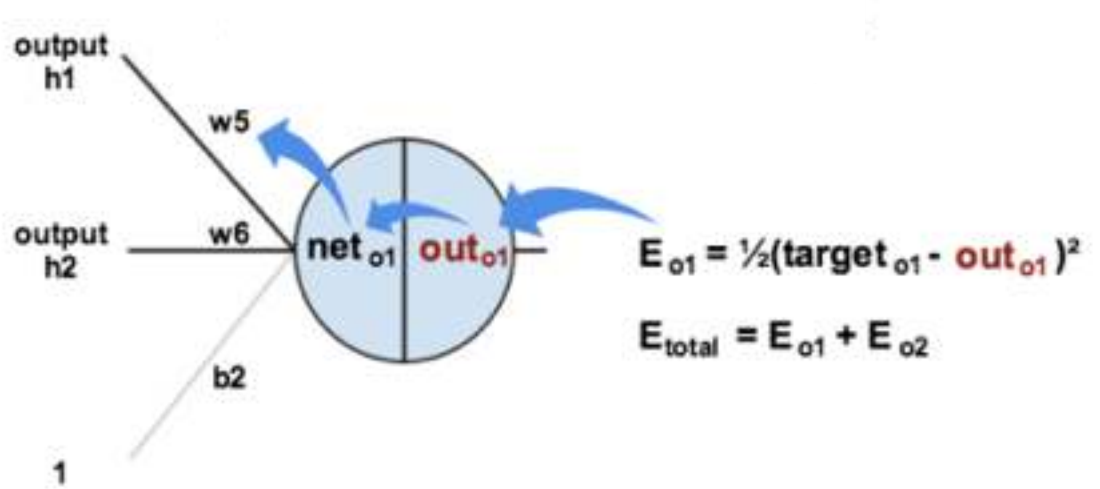
Lets backpropagate



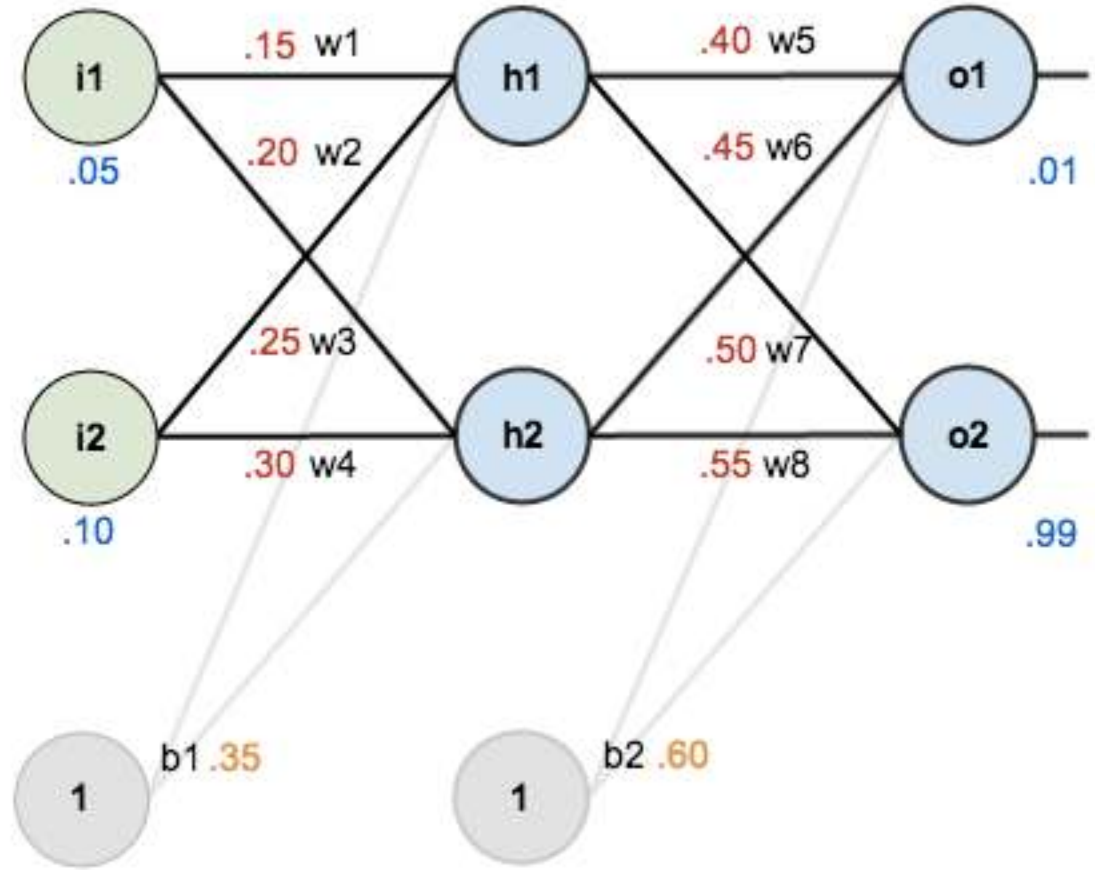
INPUT	TARGET
i1 = 0.05	o1 = 0.01
i2 = 0.10	o2 = 0.99

2. The Backward pass - Updating weights

We want to know how much a change in W5 affects the total error



Lets backpropagate



INPUT

TARGET

i1 = 0.05

o1 = 0.01

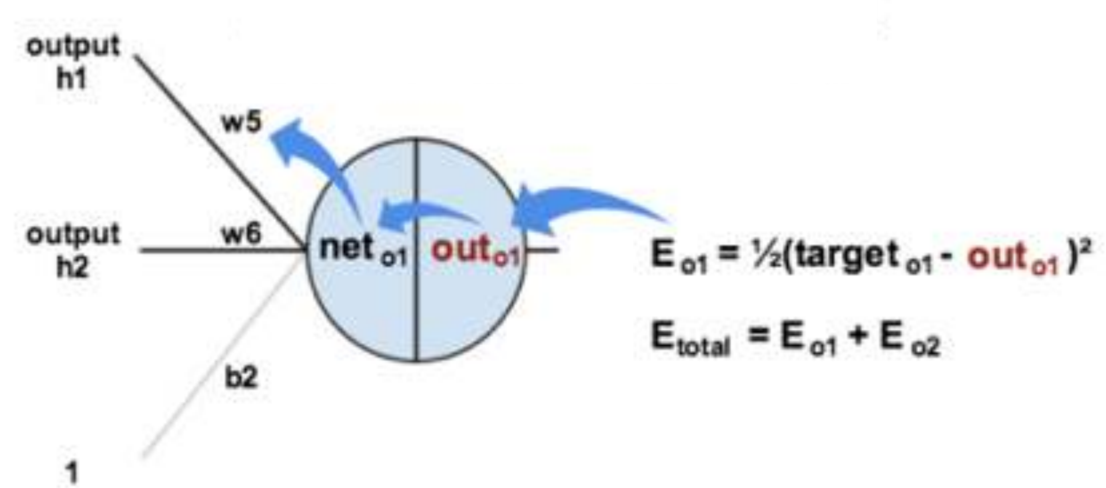
i2 = 0.10

o2 = 0.99

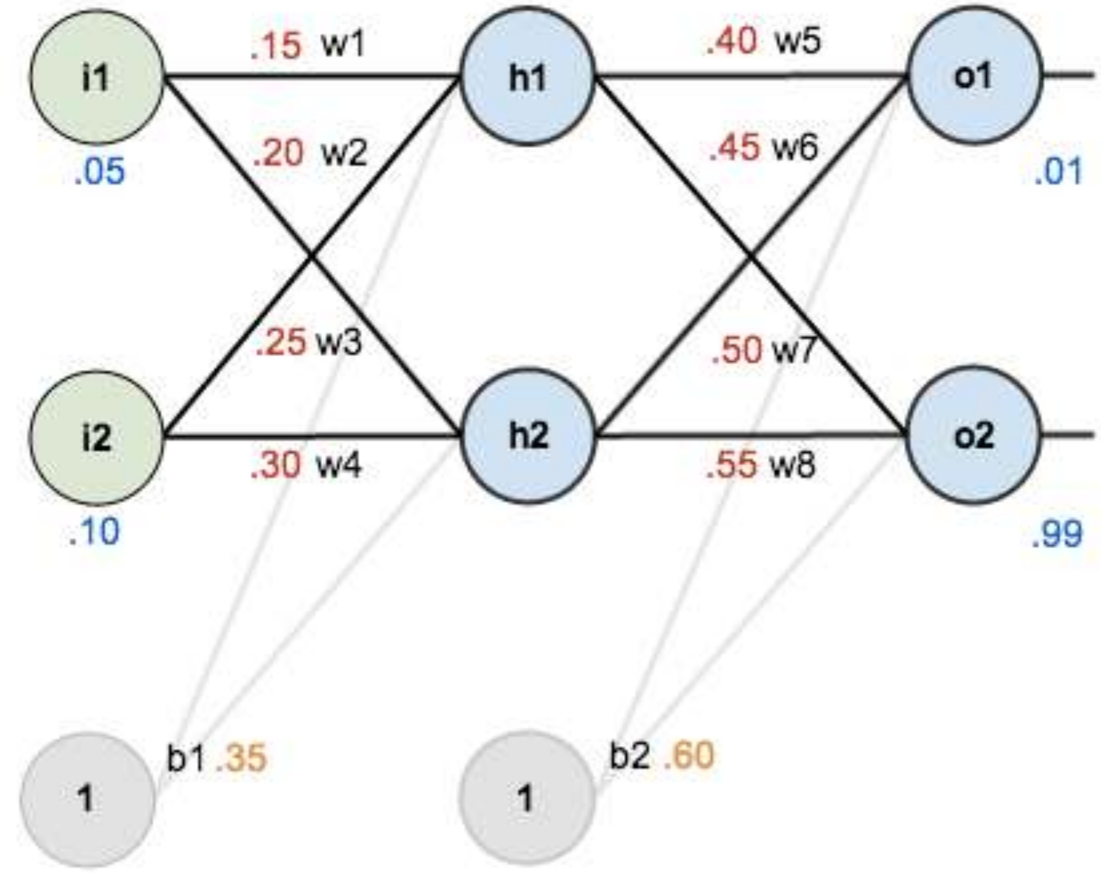
2. The Backward pass - Updating weights

We want to know how much a change in w5 affects the total error

$$\frac{\partial E_{total}}{\partial w_5} = \frac{\partial E_{total}}{\partial out_{o1}} * \frac{\partial out_{o1}}{\partial net_{o1}} * \frac{\partial net_{o1}}{\partial w_5}$$



Lets backpropagate

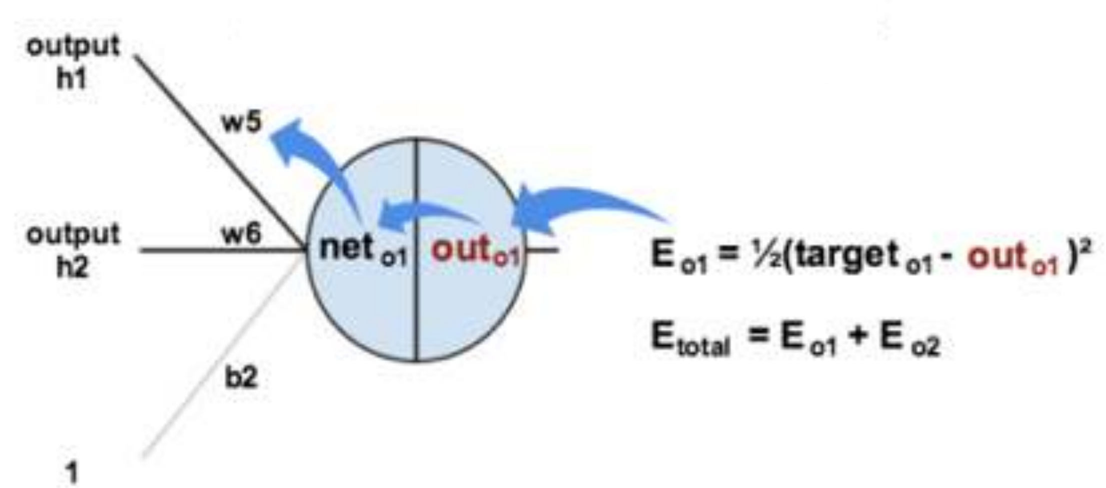


INPUT	TARGET
i1 = 0.05	o1 = 0.01
i2 = 0.10	o2 = 0.99

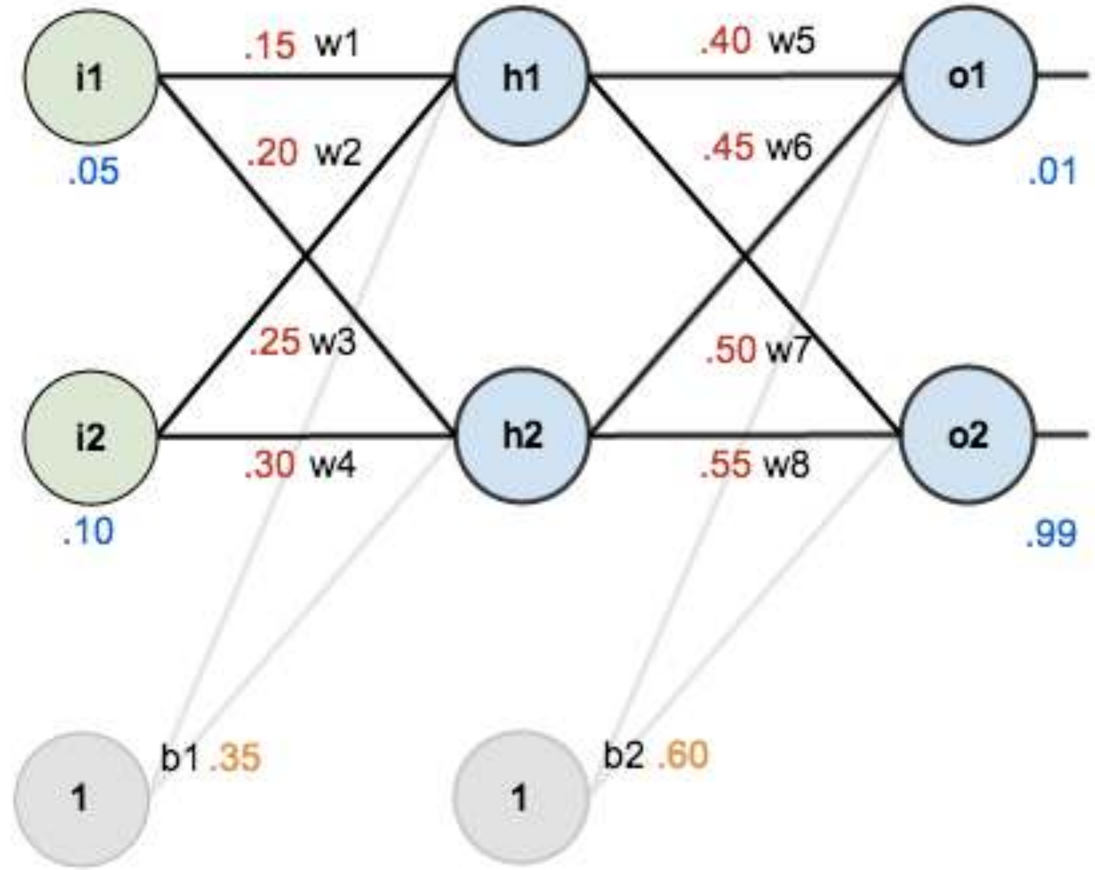
2. The Backward pass - Updating weights

We want to know how much a change in w5 affects the total error

$$\frac{\partial E_{total}}{\partial w_5} = \boxed{\frac{\partial E_{total}}{\partial out_{o1}}} * \frac{\partial out_{o1}}{\partial net_{o1}} * \frac{\partial net_{o1}}{\partial w_5}$$



Lets backpropagate



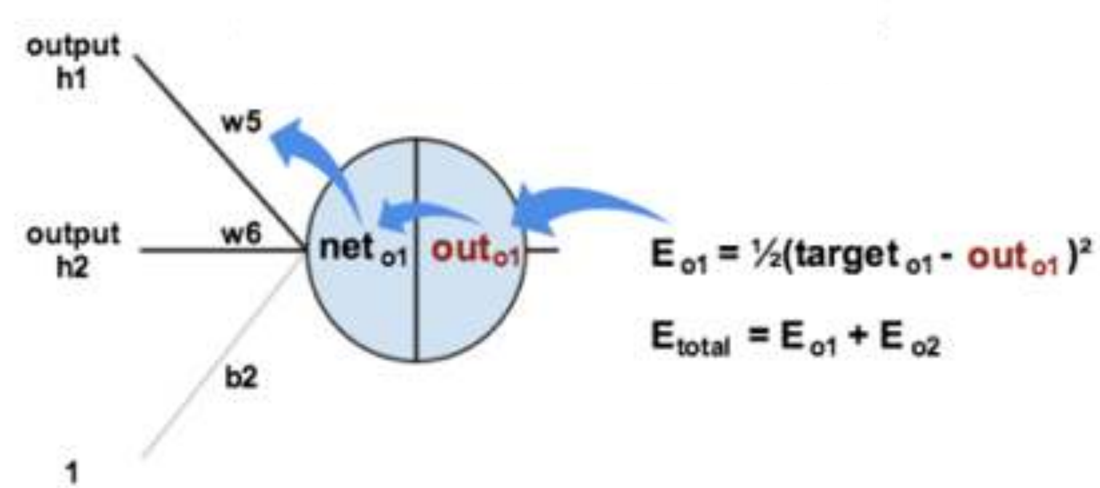
INPUT	TARGET
i1 = 0.05	o1 = 0.01
i2 = 0.10	o2 = 0.99

2. The Backward pass - Updating weights

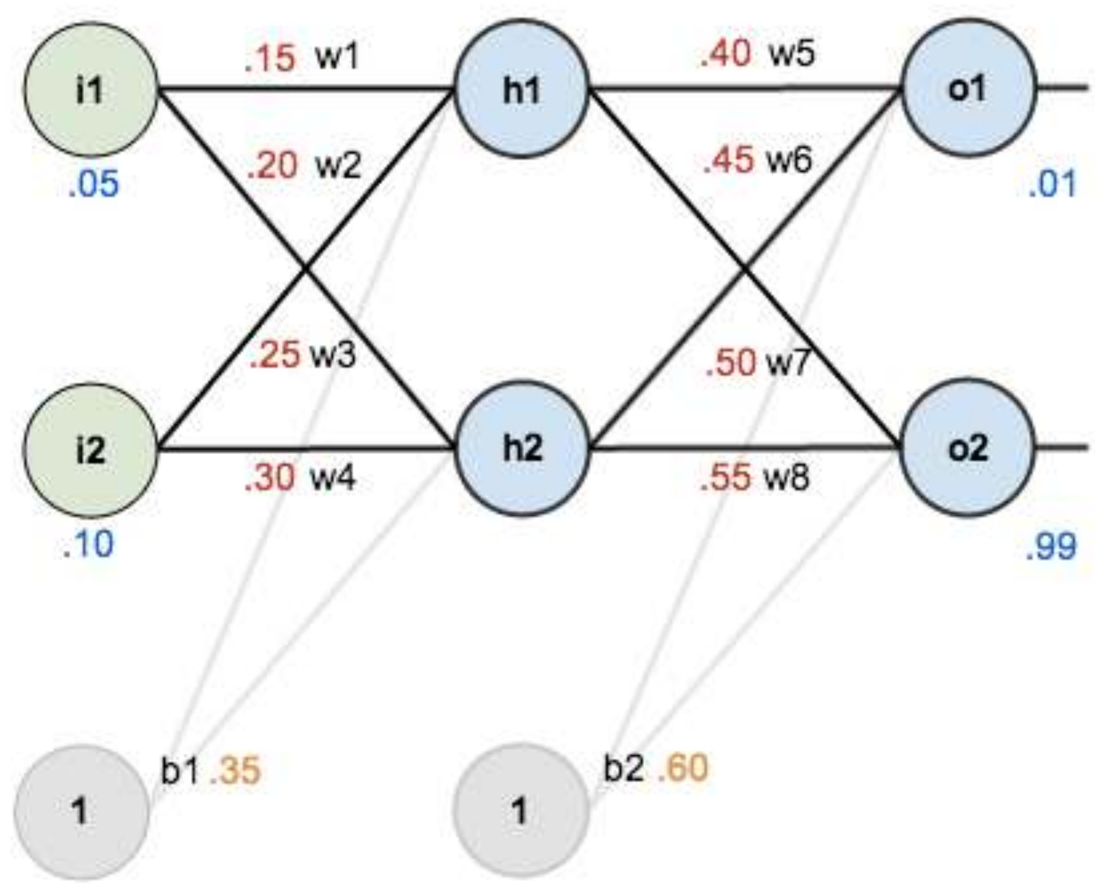
We want to know how much a change in w5 affects the total error

$$\frac{\partial E_{total}}{\partial w_5} = \boxed{\frac{\partial E_{total}}{\partial out_{o1}}} * \frac{\partial out_{o1}}{\partial net_{o1}} * \frac{\partial net_{o1}}{\partial w_5}$$

$$E_{total} = \sum \frac{1}{2}(target - output)^2$$



Lets backpropagate



INPUT

i1 = 0.05

i2 = 0.10

TARGET

o1 = 0.01

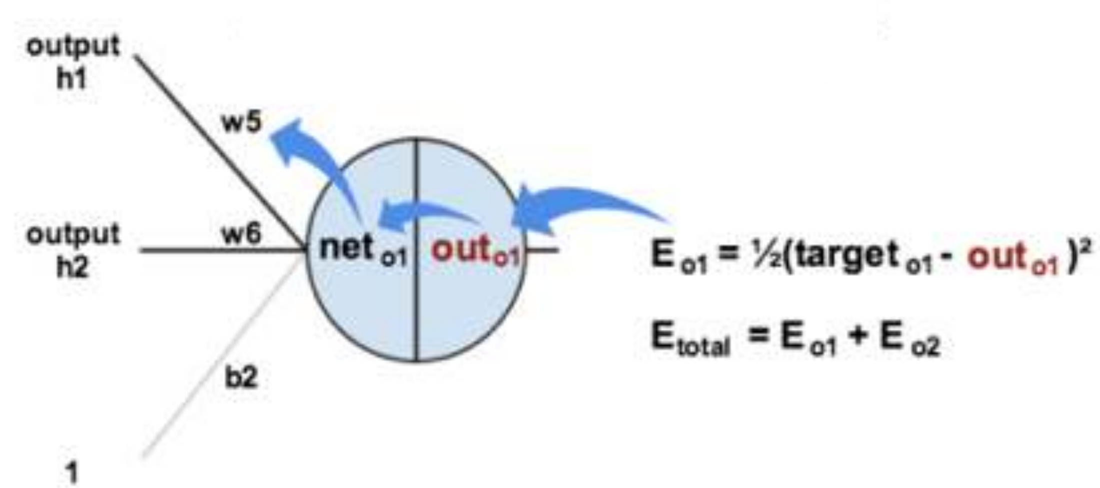
o2 = 0.99

2. The Backward pass - Updating weights

We want to know how much a change in w5 affects the total error

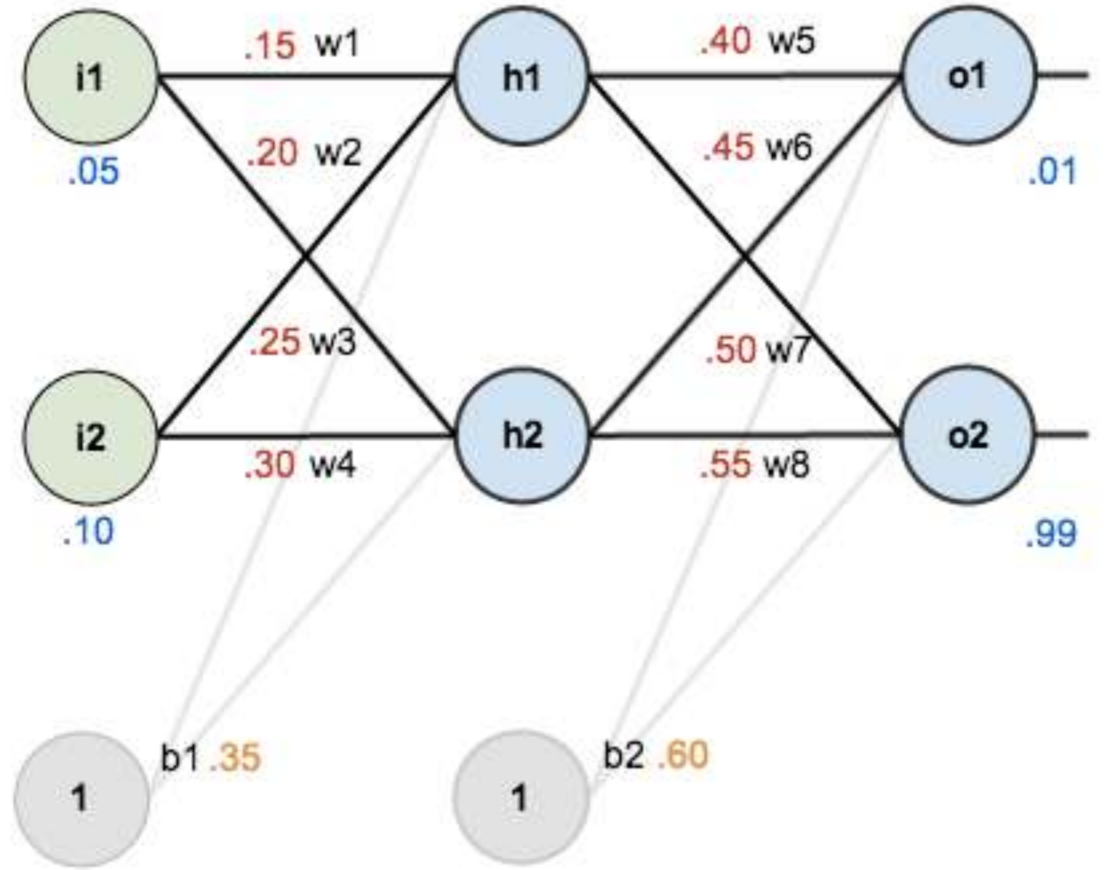
$$\frac{\partial E_{total}}{\partial w_5} = \boxed{\frac{\partial E_{total}}{\partial out_{o1}}} * \frac{\partial out_{o1}}{\partial net_{o1}} * \frac{\partial net_{o1}}{\partial w_5}$$

$$E_{total} = \sum \frac{1}{2}(target - output)^2$$



$$\frac{\partial E_{total}}{\partial out_{o1}} = 2 * \frac{1}{2}(target_{o1} - out_{o1}) * -1 + 0 = -(0.01 - 0.751) = 0.741$$

Lets backpropagate



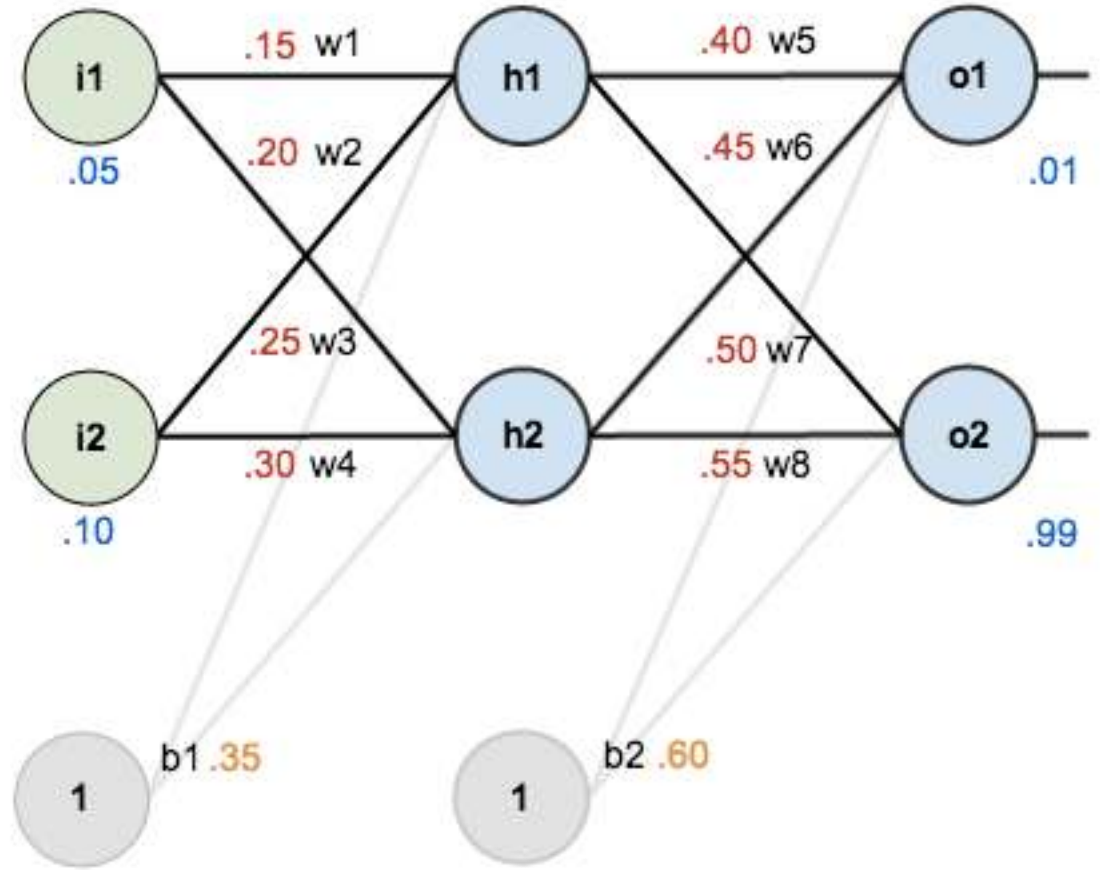
INPUT	TARGET
i1 = 0.05	o1 = 0.01
i2 = 0.10	o2 = 0.99

2. The Backward pass - Updating weights

We want to know how much a change in w5 affects the total error

$$\frac{\partial E_{total}}{\partial w_5} = \frac{\partial E_{total}}{\partial out_{o1}} * \boxed{\frac{\partial out_{o1}}{\partial net_{o1}}} * \frac{\partial net_{o1}}{\partial w_5}$$

Lets backpropagate



INPUT	TARGET
i1 = 0.05	o1 = 0.01
i2 = 0.10	o2 = 0.99

2. The Backward pass - Updating weights

We want to know how much a change in w5 affects the total error

$$\frac{\partial E_{total}}{\partial w_5} = \frac{\partial E_{total}}{\partial out_{o1}} * \boxed{\frac{\partial out_{o1}}{\partial net_{o1}}} * \frac{\partial net_{o1}}{\partial w_5}$$

$$out_{o1} = \frac{1}{1+e^{-net_{o1}}}$$

Lets backpropagate

INPUT

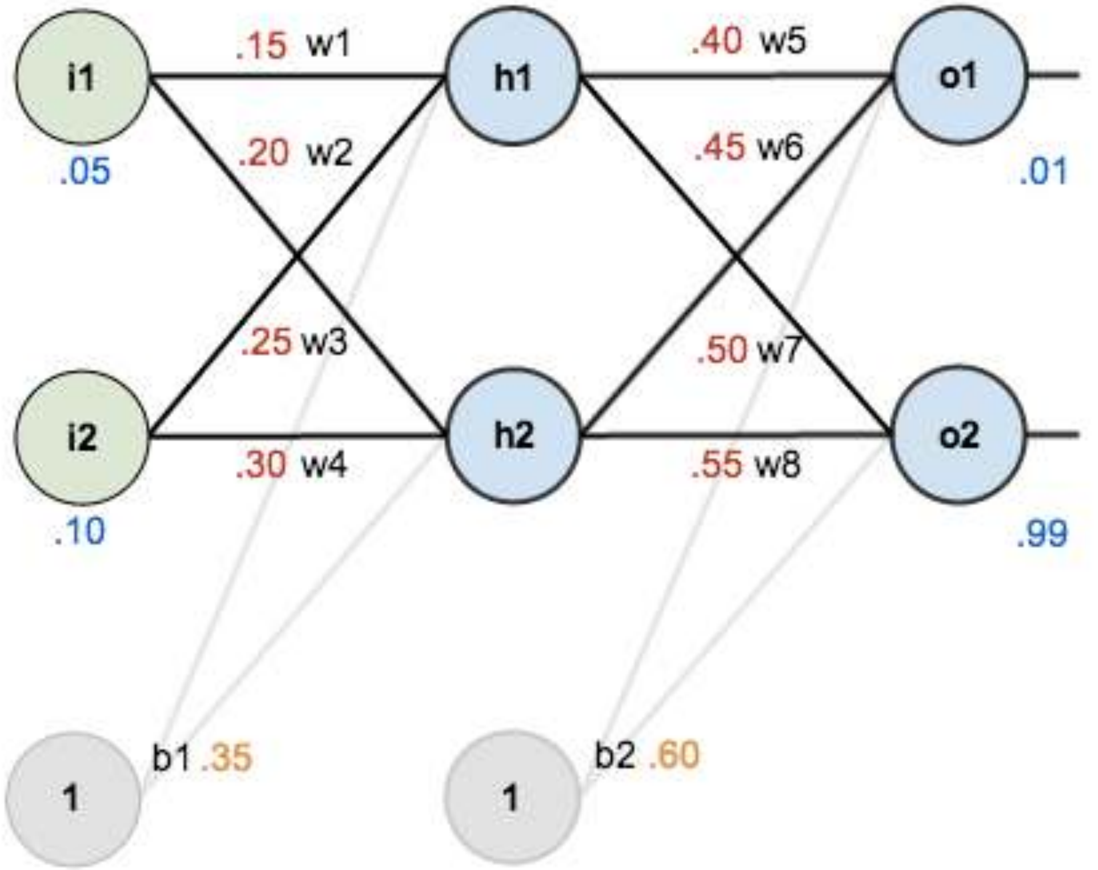
i1 = 0.05

i2 = 0.10

TARGET

o1 = 0.01

o2 = 0.99



2. The Backward pass - Updating weights

We want to know how much a change in w5 affects the total error

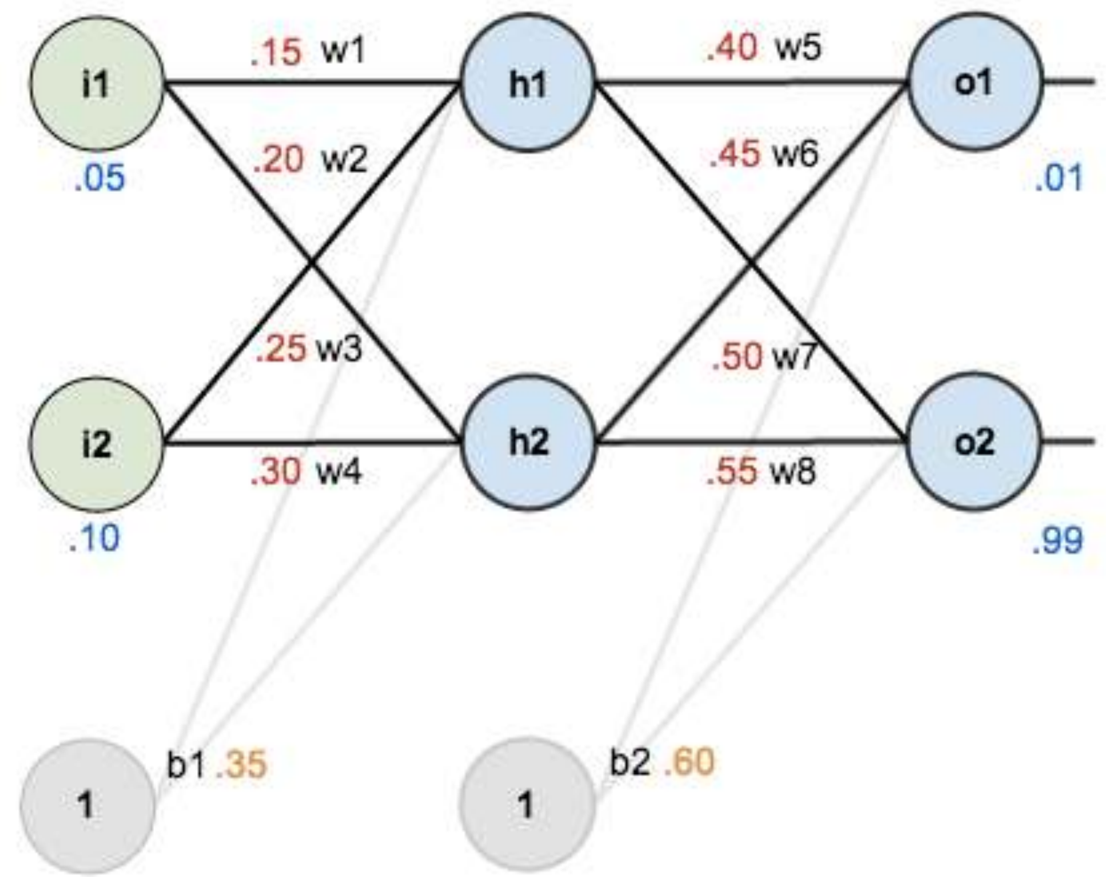
$$\frac{\partial E_{total}}{\partial w_5} = \frac{\partial E_{total}}{\partial out_{o1}} * \boxed{\frac{\partial out_{o1}}{\partial net_{o1}}} * \frac{\partial net_{o1}}{\partial w_5}$$

$$out_{o1} = \frac{1}{1 + e^{-net_{o1}}}$$

$$\frac{\partial out_{o1}}{\partial net_{o1}} = out_{o1}\left(1 - out_{o1}\right) = 0.1868$$

Lets backpropagate

INPUT	TARGET
i1 = 0.05	o1 = 0.01
i2 = 0.10	o2 = 0.99



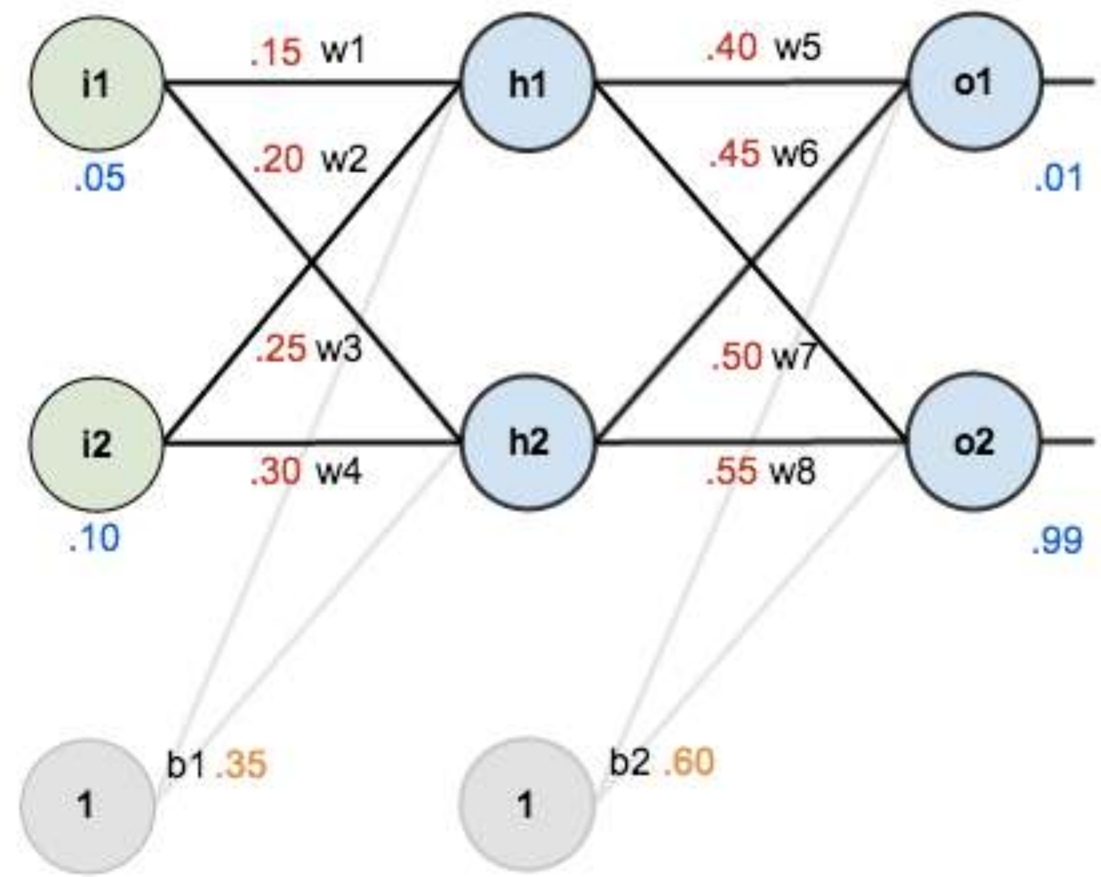
2. The Backward pass - Updating weights

We want to know how much a change in w5 affects the total error

$$\frac{\partial E_{total}}{\partial w_5} = \frac{\partial E_{total}}{\partial out_{o1}} * \frac{\partial out_{o1}}{\partial net_{o1}} * \boxed{\frac{\partial net_{o1}}{\partial w_5}}$$

Lets backpropagate

INPUT	TARGET
i1 = 0.05	o1 = 0.01
i2 = 0.10	o2 = 0.99



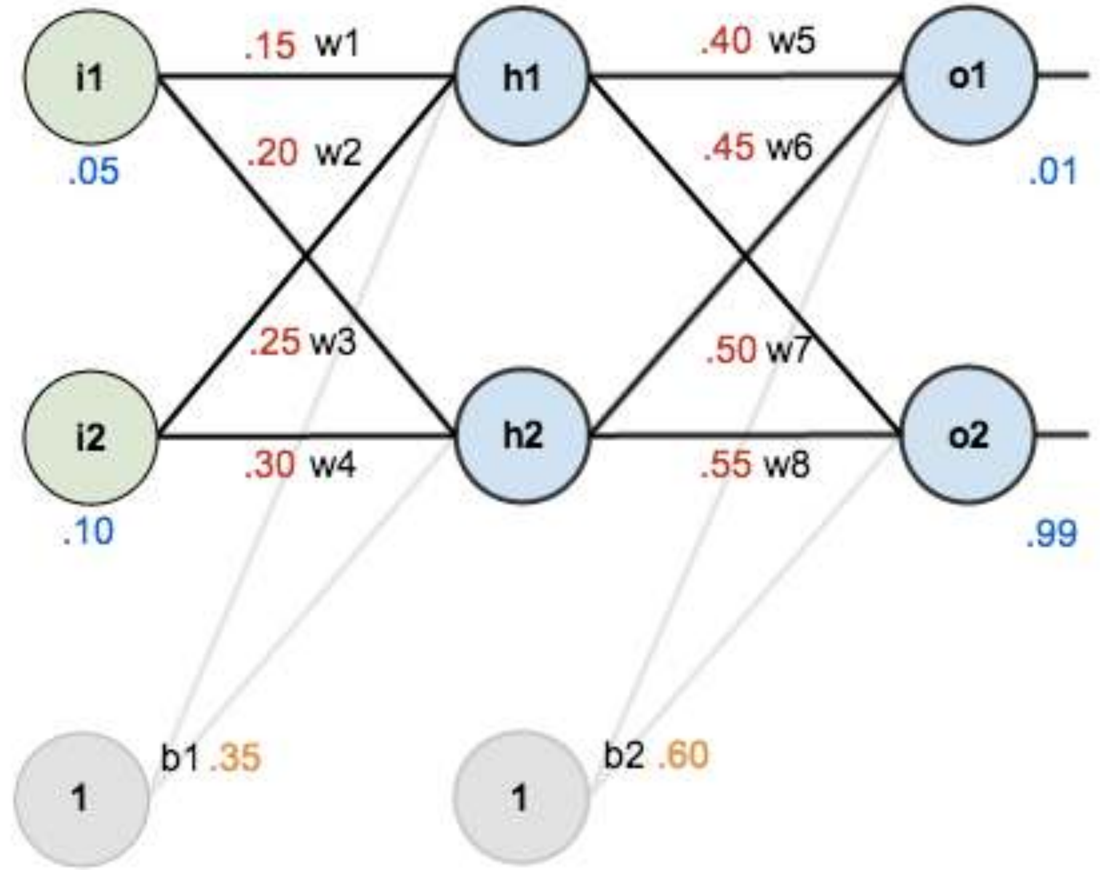
2. The Backward pass - Updating weights

We want to know how much a change in w5 affects the total error

$$\frac{\partial E_{total}}{\partial w_5} = \frac{\partial E_{total}}{\partial out_{o1}} * \frac{\partial out_{o1}}{\partial net_{o1}} * \boxed{\frac{\partial net_{o1}}{\partial w_5}}$$

$$net_{o1} = w_5 * out_{h1} + w_6 * out_{h2} + b_2 * 1$$

Lets backpropagate



INPUT	TARGET
i1 = 0.05	o1 = 0.01
i2 = 0.10	o2 = 0.99

2. The Backward pass - Updating weights

We want to know how much a change in w5 affects the total error

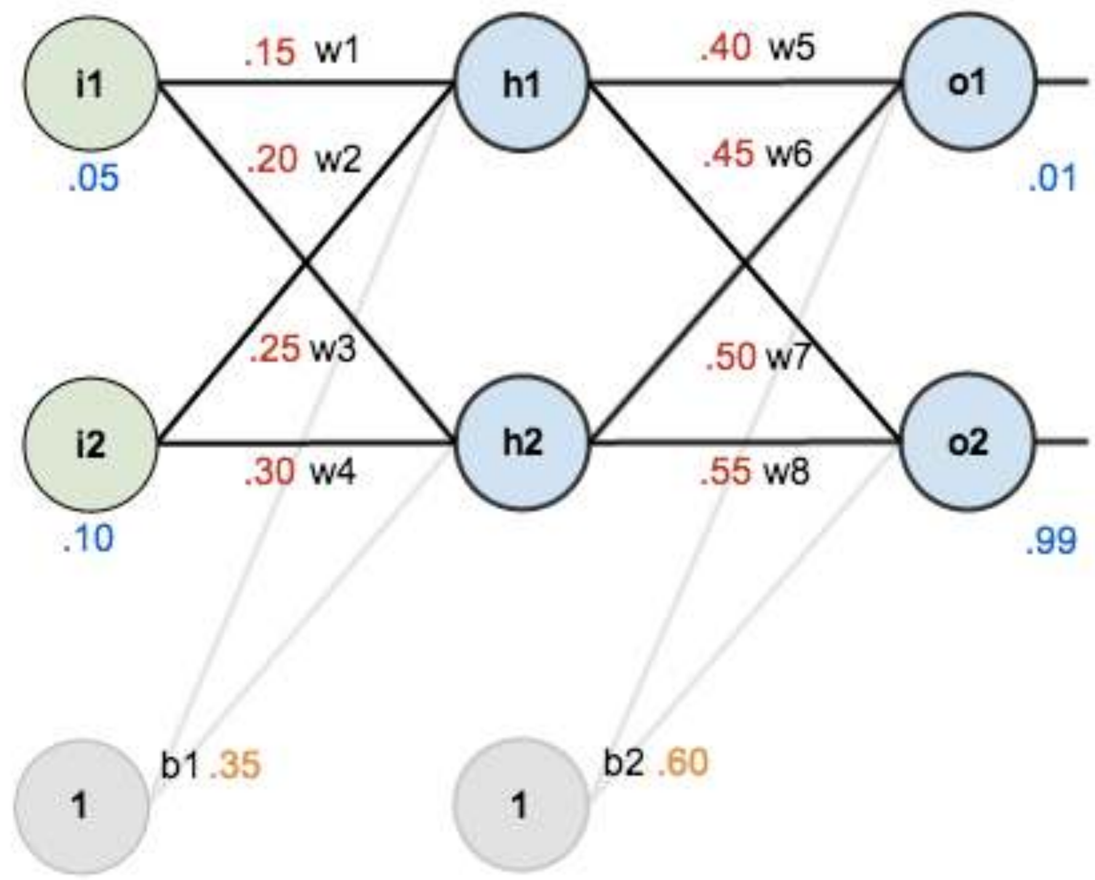
$$\frac{\partial E_{total}}{\partial w_5} = \frac{\partial E_{total}}{\partial out_{o1}} * \frac{\partial out_{o1}}{\partial net_{o1}} * \boxed{\frac{\partial net_{o1}}{\partial w_5}}$$

$$net_{o1} = w_5 * out_{h1} + w_6 * out_{h2} + b_2 * 1$$

$$\frac{\partial net_{o1}}{\partial w_5} = out_{h1} = 0.5933$$

Lets backpropagate

INPUT	TARGET
i1 = 0.05	o1 = 0.01
i2 = 0.10	o2 = 0.99

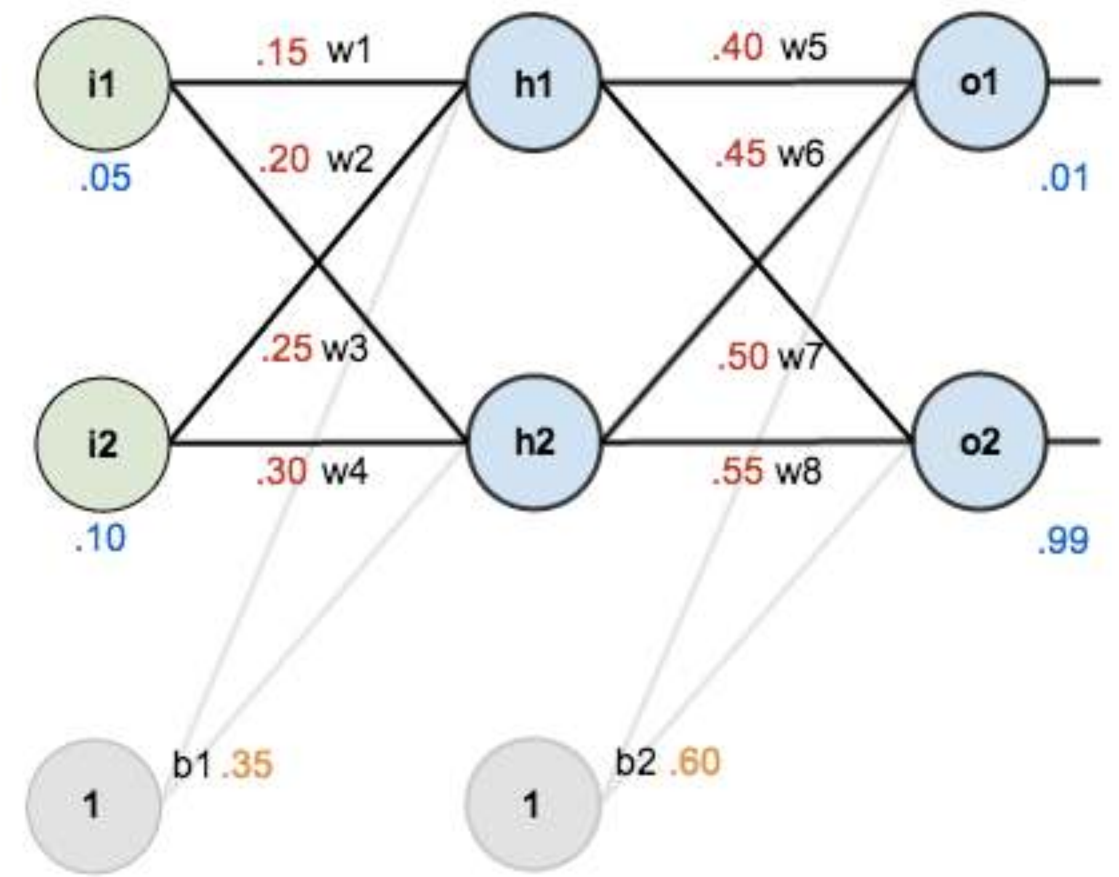


2. The Backward pass - Updating weights

We want to know how much a change in w5 affects the total error

$$\frac{\partial E_{total}}{\partial w_5} = \frac{\partial E_{total}}{\partial out_{o1}} * \frac{\partial out_{o1}}{\partial net_{o1}} * \frac{\partial net_{o1}}{\partial w_5}$$

Lets backpropagate



INPUT	TARGET
i1 = 0.05	o1 = 0.01
i2 = 0.10	o2 = 0.99

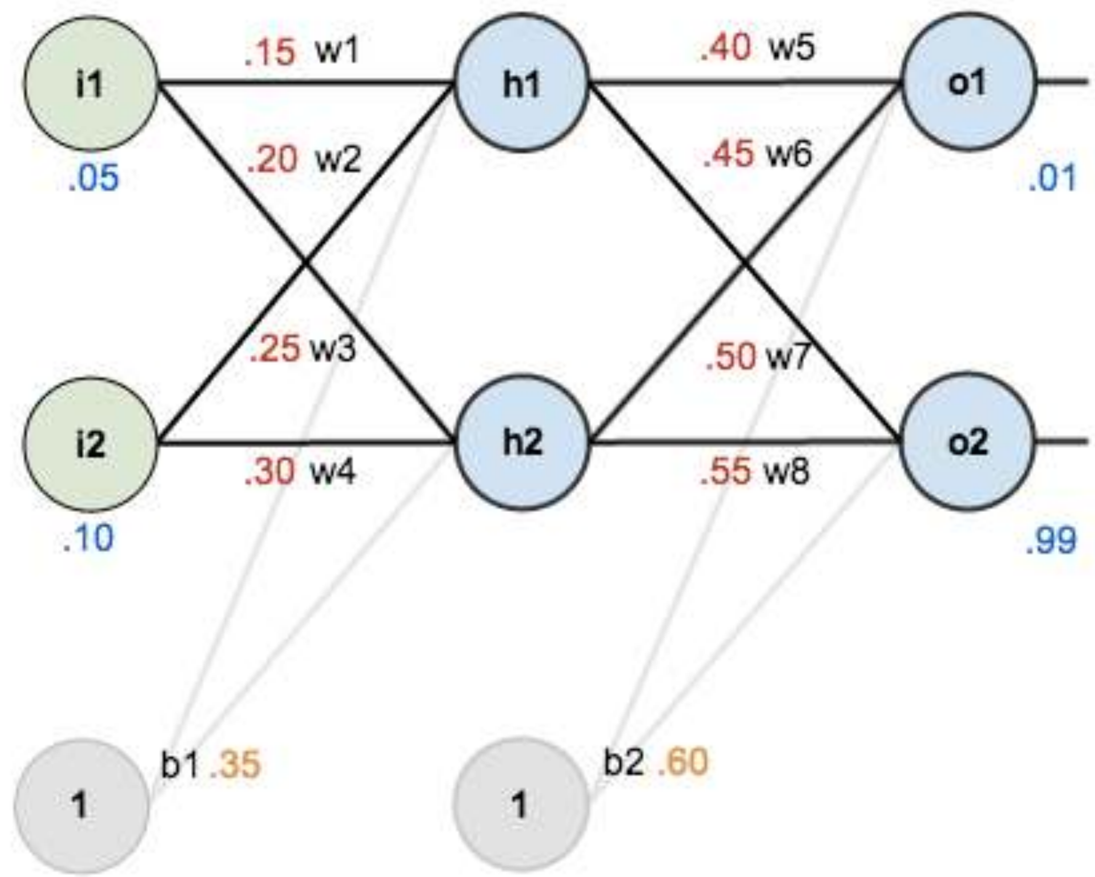
2. The Backward pass - Updating weights

We want to know how much a change in w5 affects the total error

$$\frac{\partial E_{total}}{\partial w_5} = \frac{\partial E_{total}}{\partial out_{o1}} * \frac{\partial out_{o1}}{\partial net_{o1}} * \frac{\partial net_{o1}}{\partial w_5}$$

$$\frac{\partial E_{total}}{\partial w_5} = 0.7414 * 0.1868 * 0.5933 = 0.0821$$

Lets backpropagate



INPUT	TARGET
i1 = 0.05	o1 = 0.01
i2 = 0.10	o2 = 0.99

2. The Backward pass - Updating weights

We want to know how much a change in w5 affects the total error

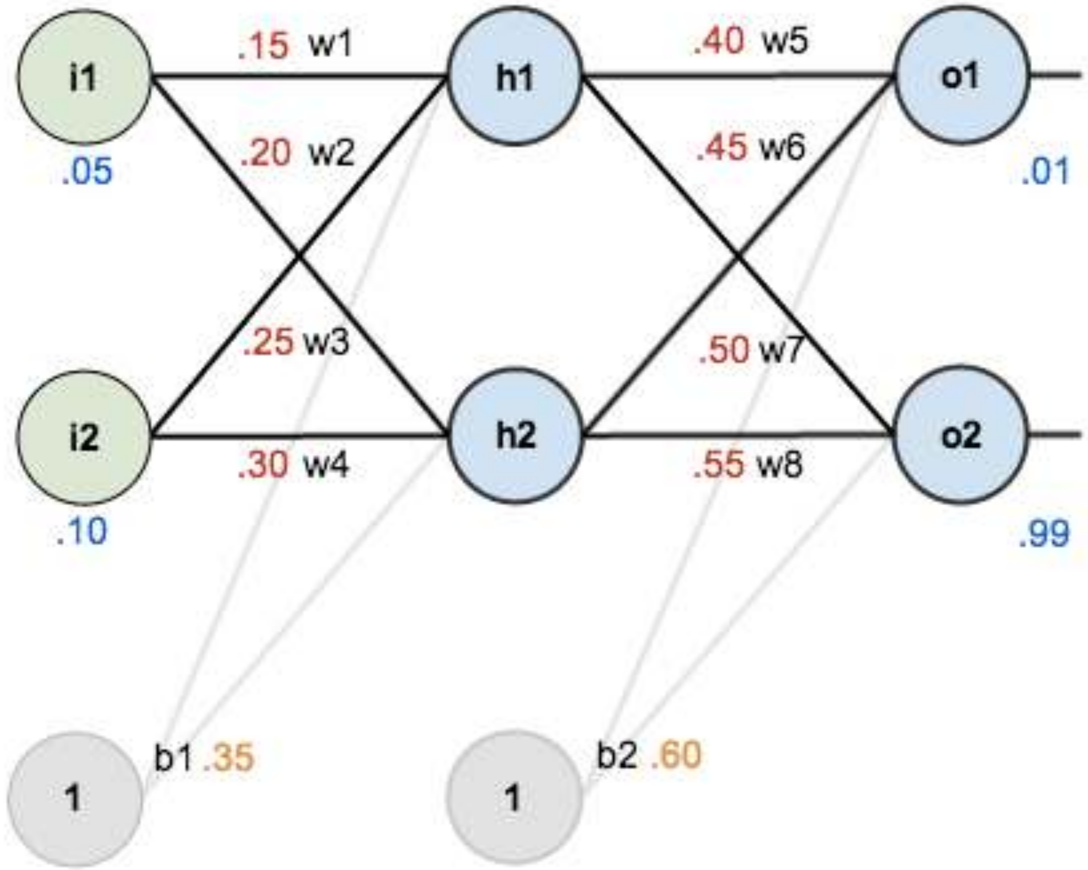
$$\frac{\partial E_{total}}{\partial w_5} = \frac{\partial E_{total}}{\partial out_{o1}} * \frac{\partial out_{o1}}{\partial net_{o1}} * \frac{\partial net_{o1}}{\partial w_5}$$

$$\frac{\partial E_{total}}{\partial w_5} = 0.7414 * 0.1868 * 0.5933 = 0.0821$$

$$w_5^{new} = w_5^{old} - \eta * \frac{\partial E_{total}}{\partial w_5} = 0.4 - 0.5 * 0.0821 = 0.3589$$

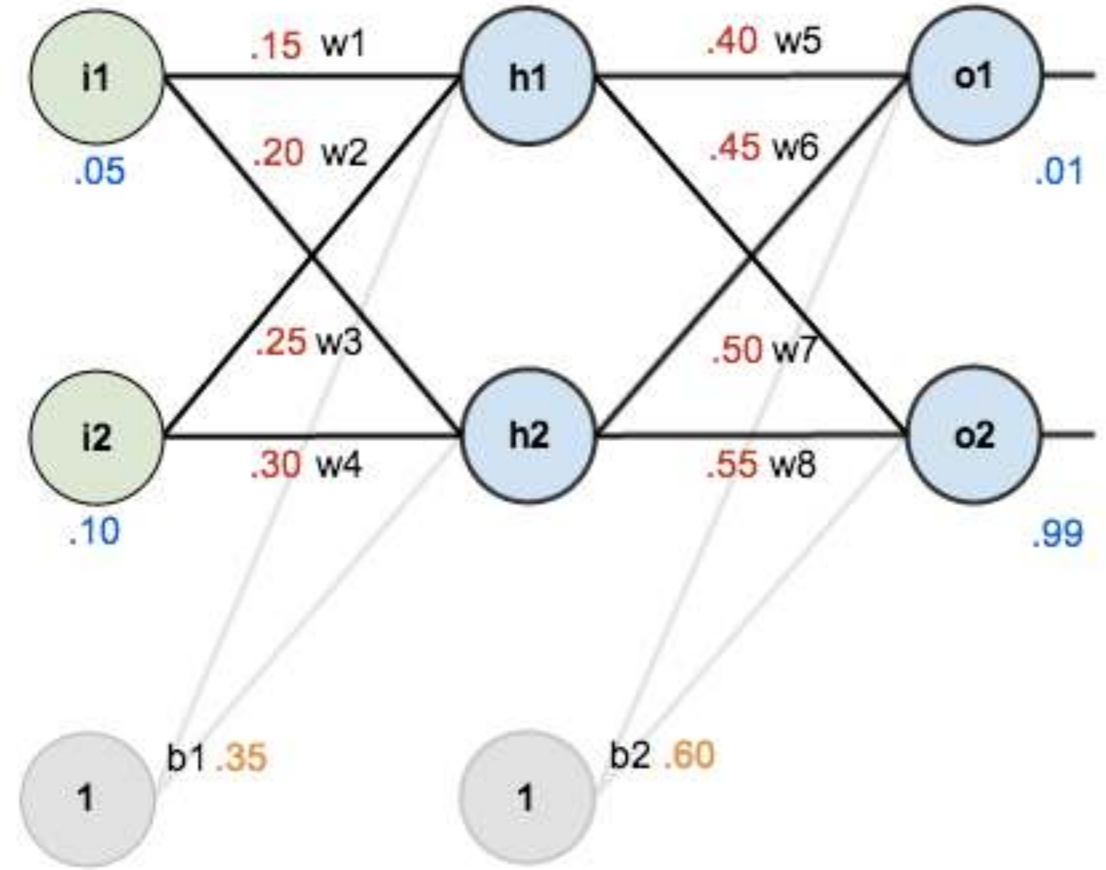
Lets backpropagate

INPUT	TARGET
i1 = 0.05	o1 = 0.01
i2 = 0.10	o2 = 0.99



- Repeat for w6, w7, w8

Lets backpropagate

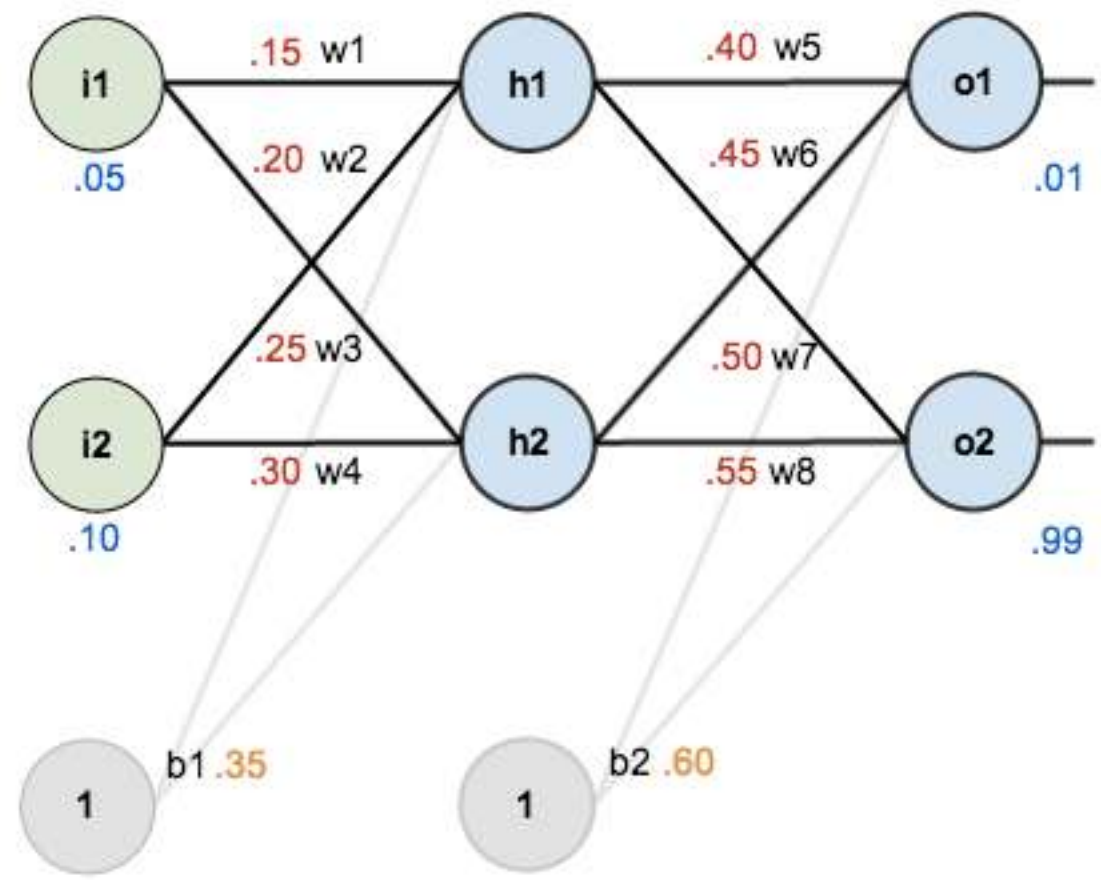


INPUT	TARGET
i1 = 0.05	o1 = 0.01
i2 = 0.10	o2 = 0.99

- Repeat for w6, w7, w8
- In analogous way for w1, w2, w3, w4

Lets backpropagate

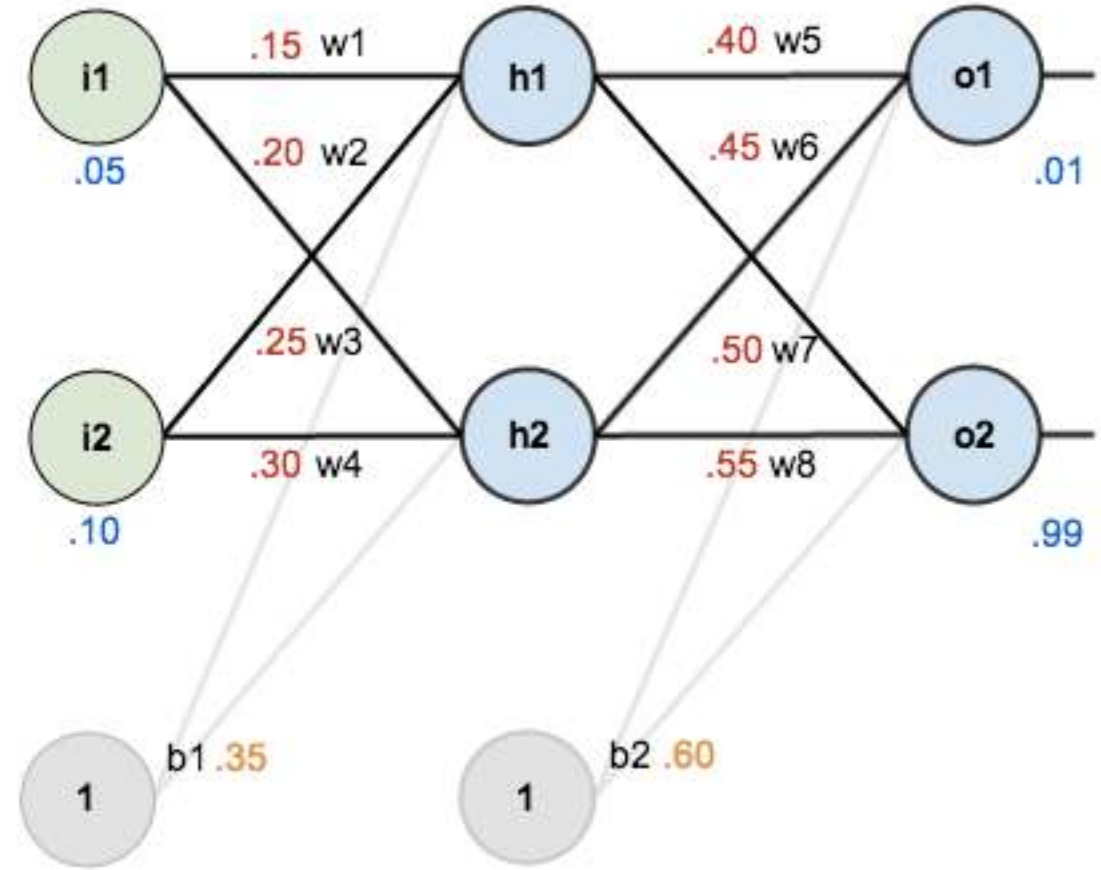
INPUT	TARGET
i1 = 0.05	o1 = 0.01
i2 = 0.10	o2 = 0.99



- Repeat for w6, w7, w8
- In analogous way for w1, w2, w3, w4
- Compute the total error before: **0.298371109**

Lets backpropagate

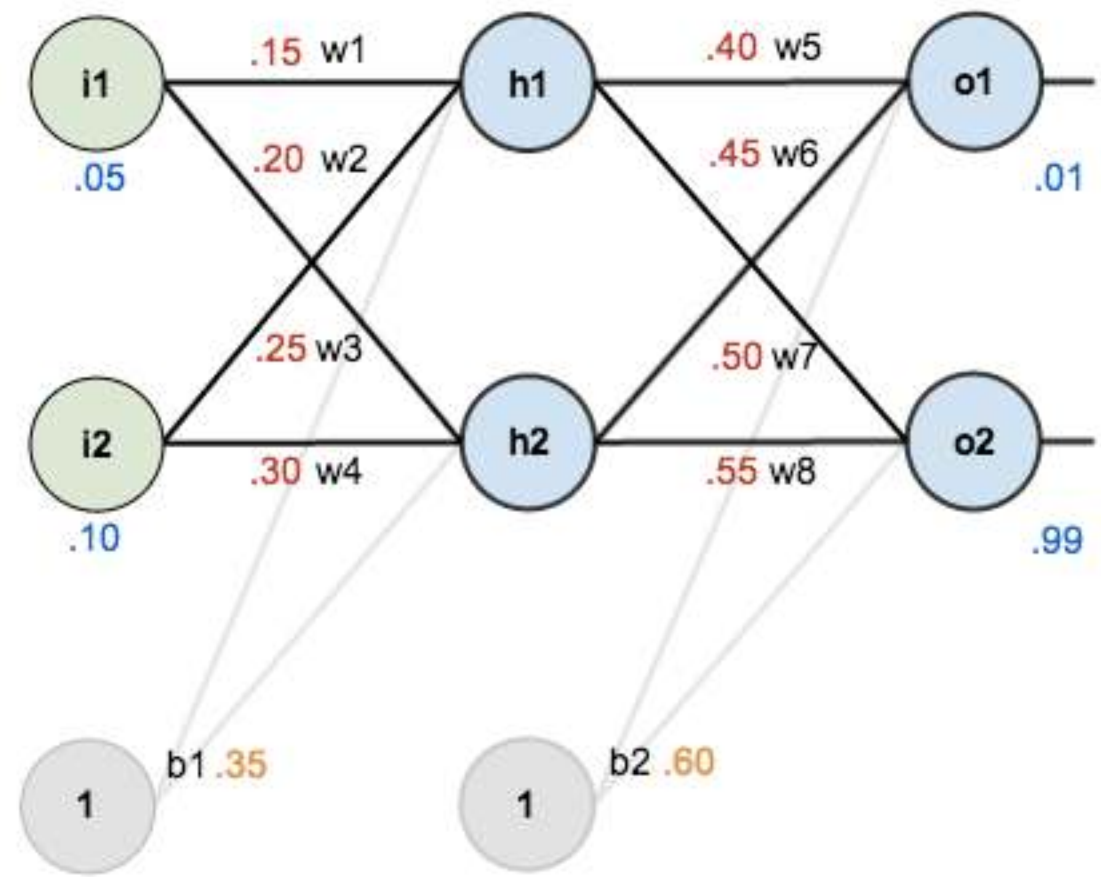
INPUT	TARGET
i1 = 0.05	o1 = 0.01
i2 = 0.10	o2 = 0.99



- Repeat for w6, w7, w8
- In analogous way for w1, w2, w3, w4
- Compute the total error before: **0.298371109**

  now: **0.291027924**

Lets backpropagate



INPUT	TARGET
i1 = 0.05	o1 = 0.01
i2 = 0.10	o2 = 0.99

- Repeat for w6, w7, w8
- In analogous way for w1, w2, w3, w4
- Compute the total error before: **0.298371109**

  now: **0.291027924**

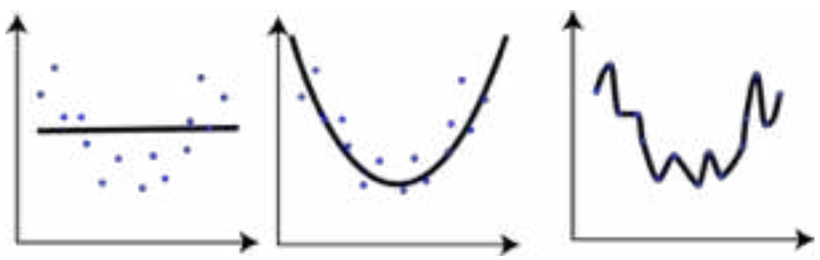
- Repeat x10000: **0.000035085**

Training ANNs

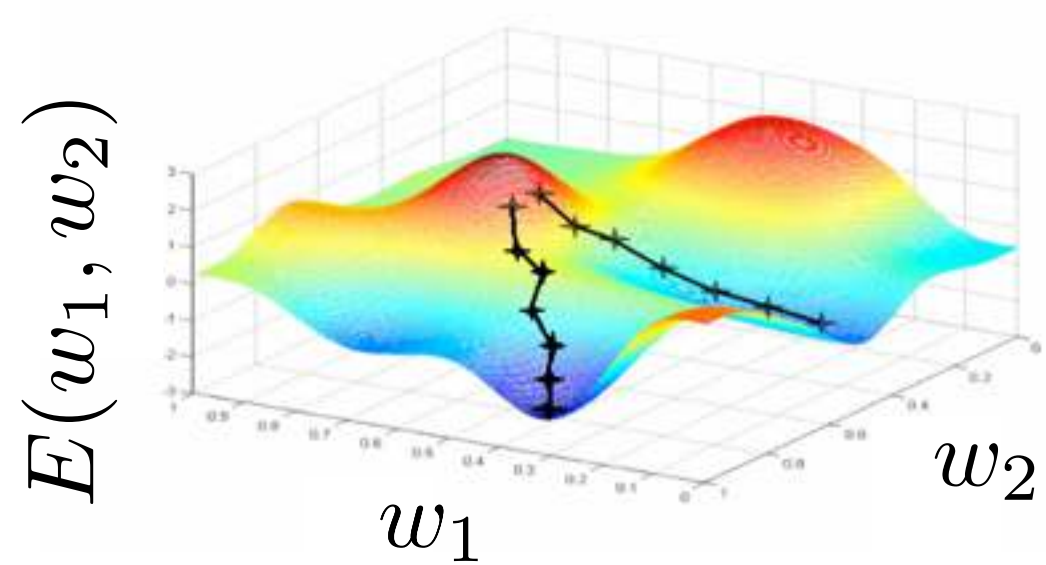
http://playground.tensorflow.org/

Deep networks were difficult to train

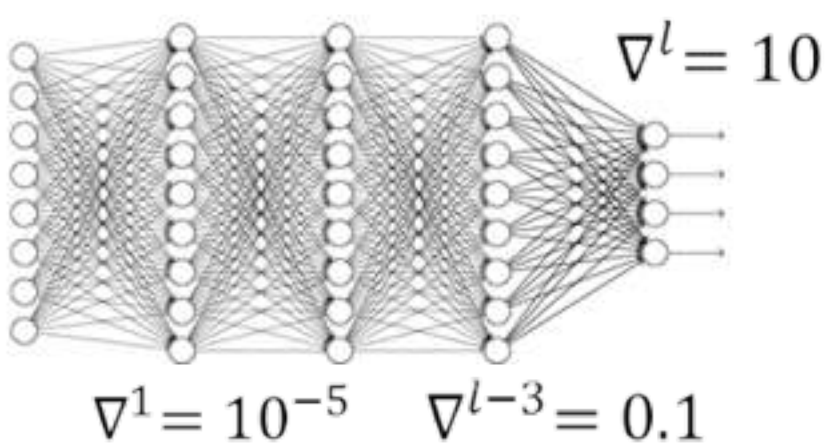
Overfitting



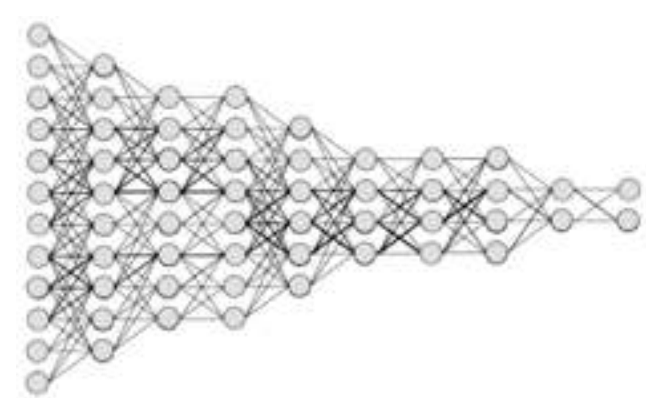
Complex landscape



Vanishing gradients



Dimensionality



Why DL revolution did not happen in 1986?

Why DL revolution did not happen in 1986?

Not enough data
(datasets 1000 too small)

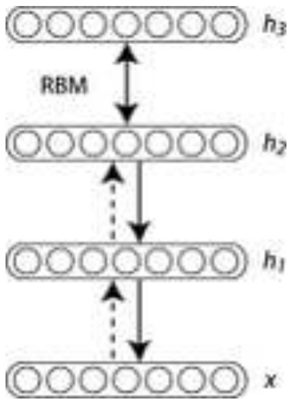
Why DL revolution did not happen in 1986?

Not enough data
(datasets 1000 too small)

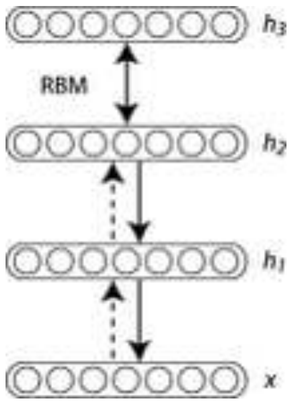
Computers were too slow
(1000000 times)


KEEP
CALM
AND
STUDY
ALGORITHMS

- Pre-training (weights initialization)
  (complex landscape)

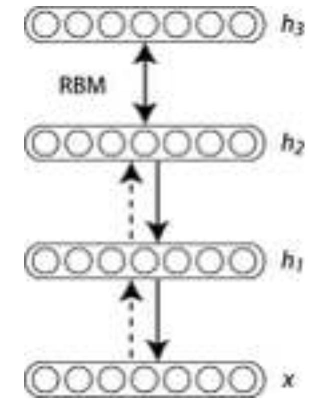


• Pre-training (weights initialization)
(complex landscape)

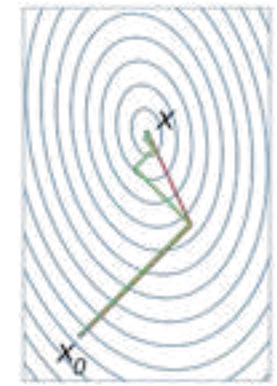


• Efficient descent algorithms
(complex landscape)

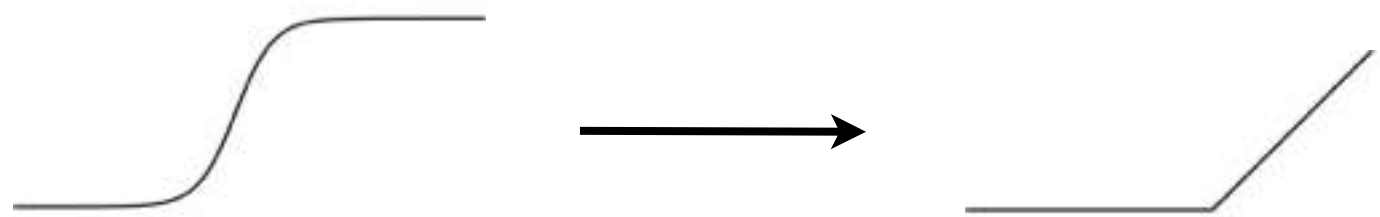
- Pre-training (weights initialization)
  (complex landscape)

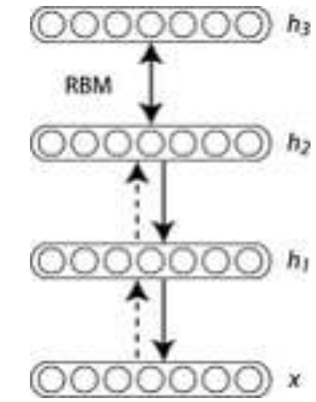


- Efficient descent algorithms
  (complex landscape)

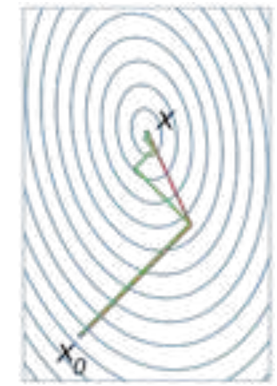


- Activation
  (vanishing gradient)



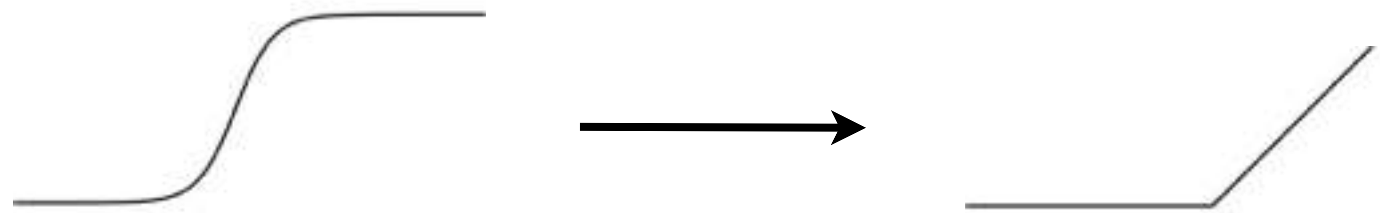


- Pre-training (weights initialization)
  (complex landscape)

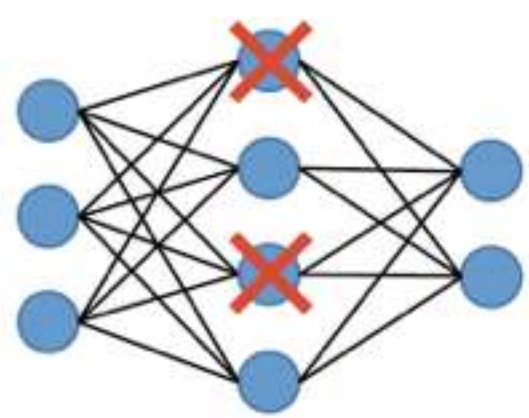


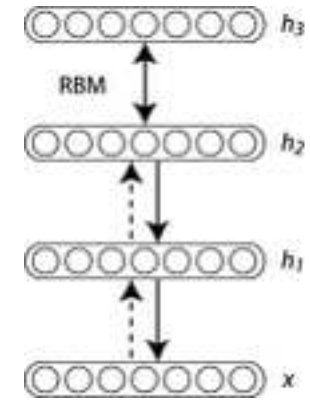
- Efficient descent algorithms
  (complex landscape)

- Activation
  (vanishing gradient)

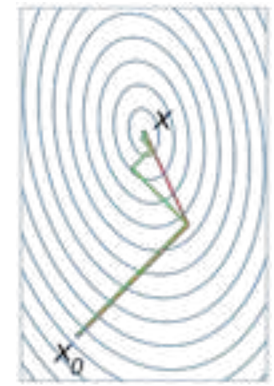


- Dropout
  (overfitting)



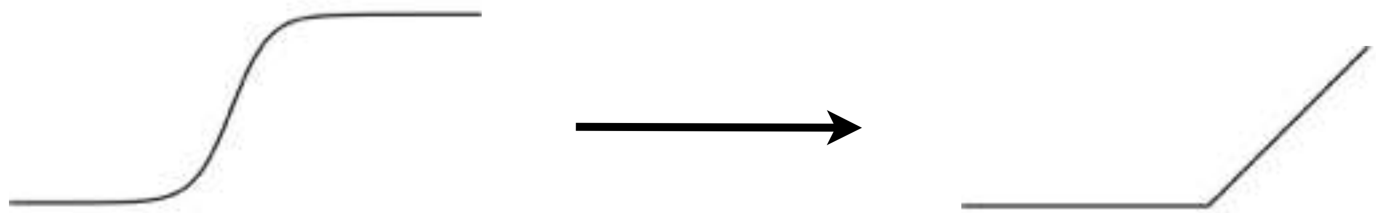


- Pre-training (weights initialization)
  (complex landscape)

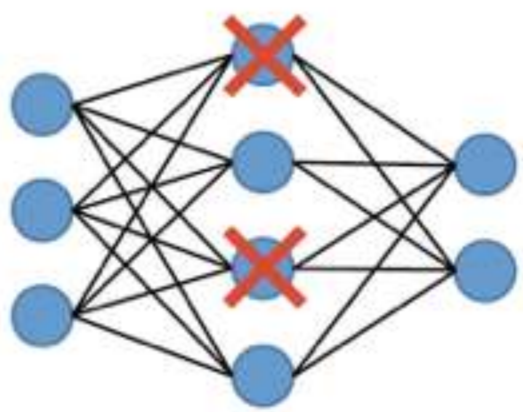


- Efficient descent algorithms
  (complex landscape)

- Activation
  (vanishing gradient)



- Dropout
  (overfitting)



- Domain Prior Knowledge

Now that we are deep...

- Instead of hand-crafted features, let the algorithm build the relevant features for your problem

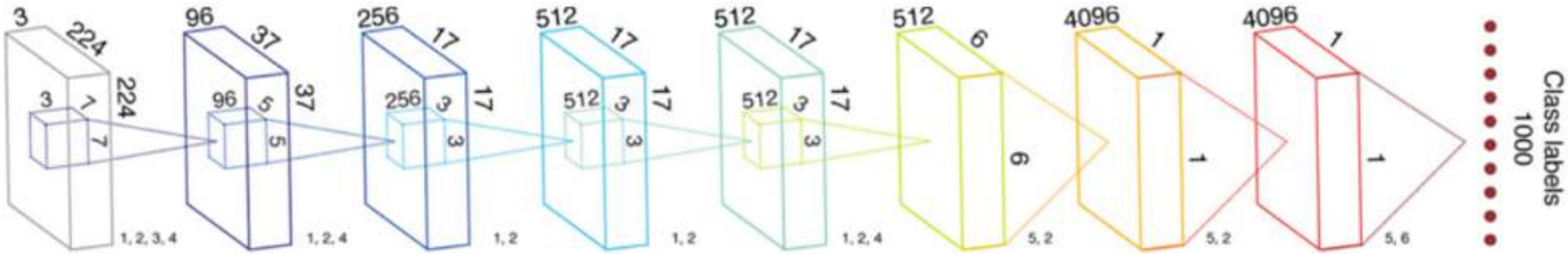
- More representational power for learning

- Powerful function approximator


ImageNet Large Scale Visual Recognition
Challenge (ILSVRC)

2012

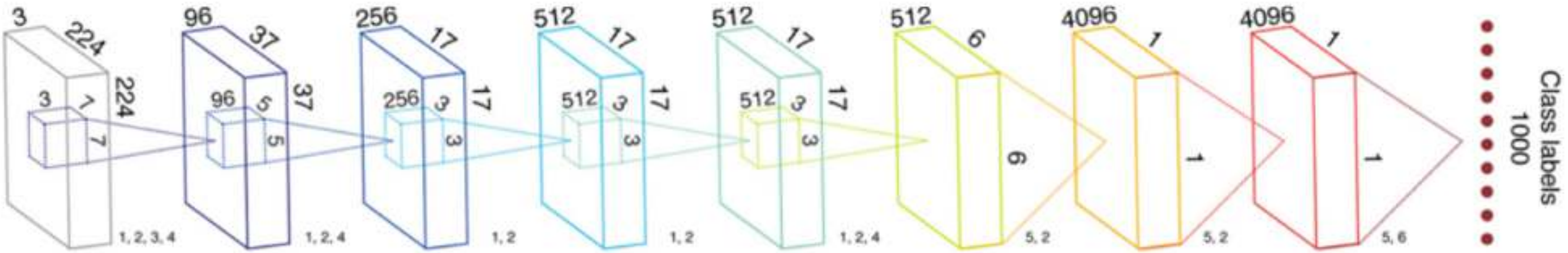
25% ⟶ 15%



IMAGENET

2012

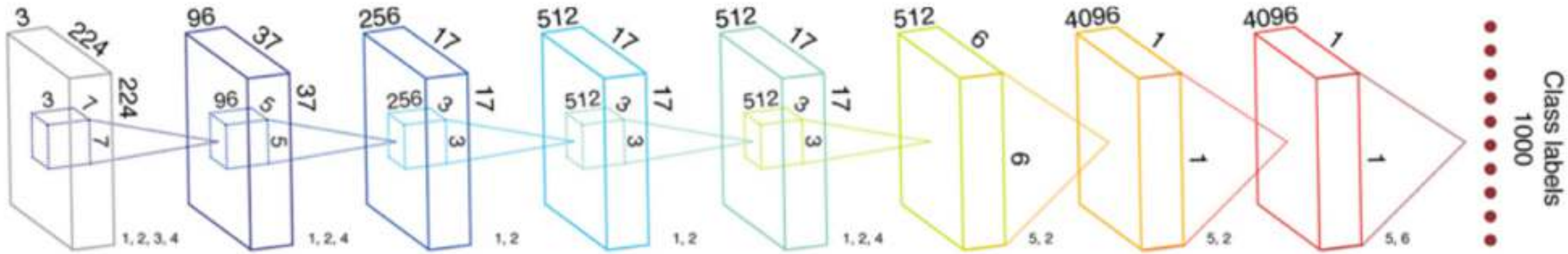
25% ⟶ 15%



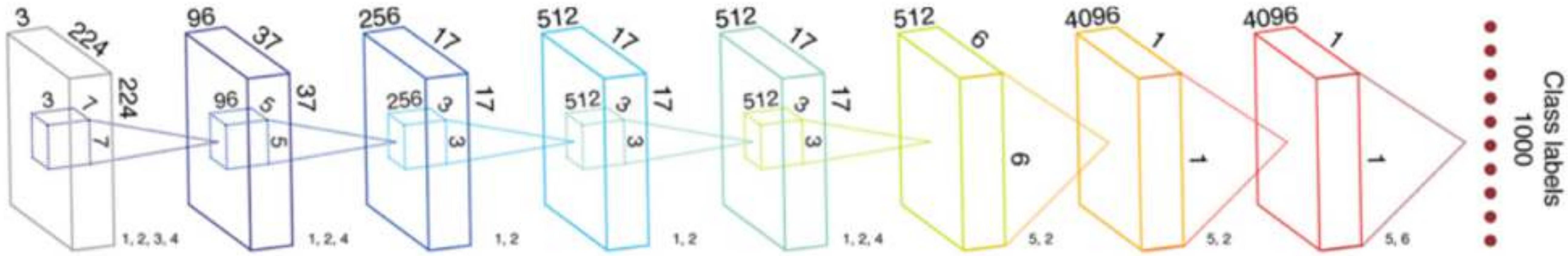
IMAGENET

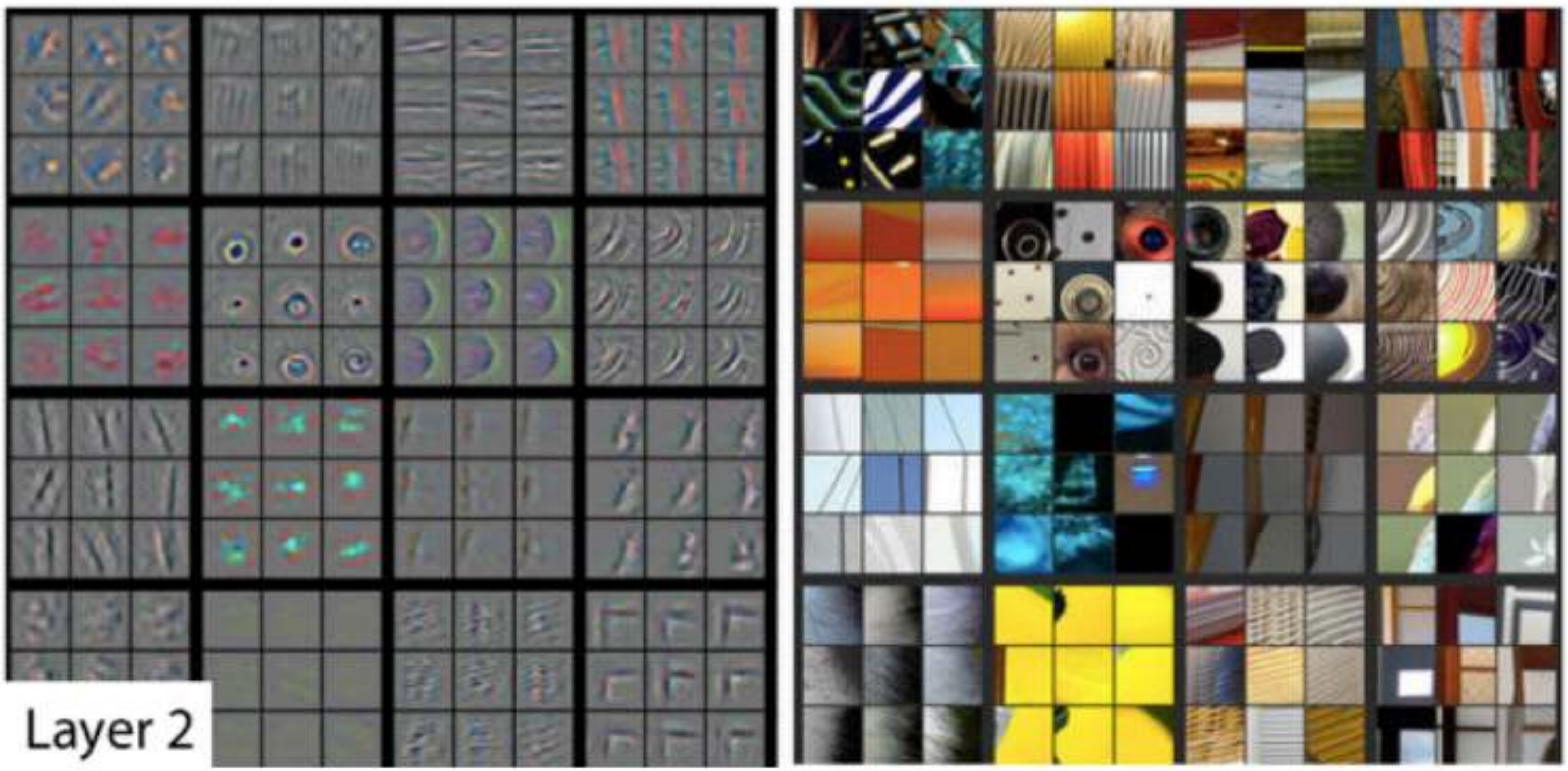
2016

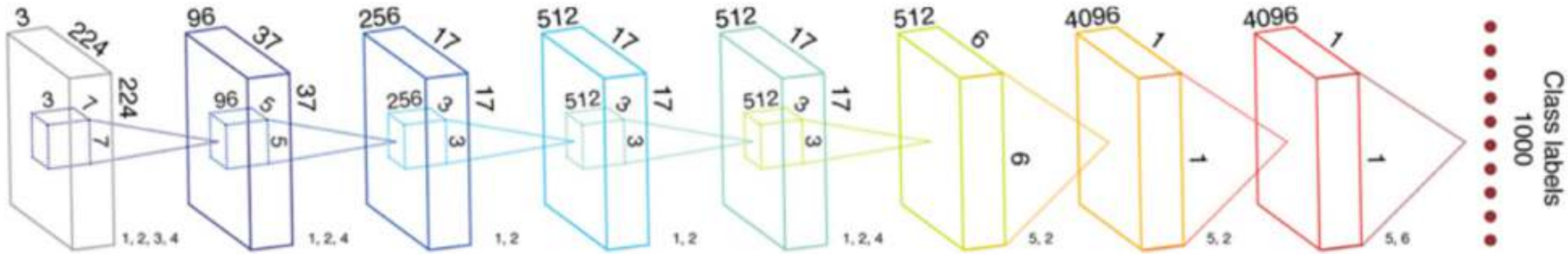
3%

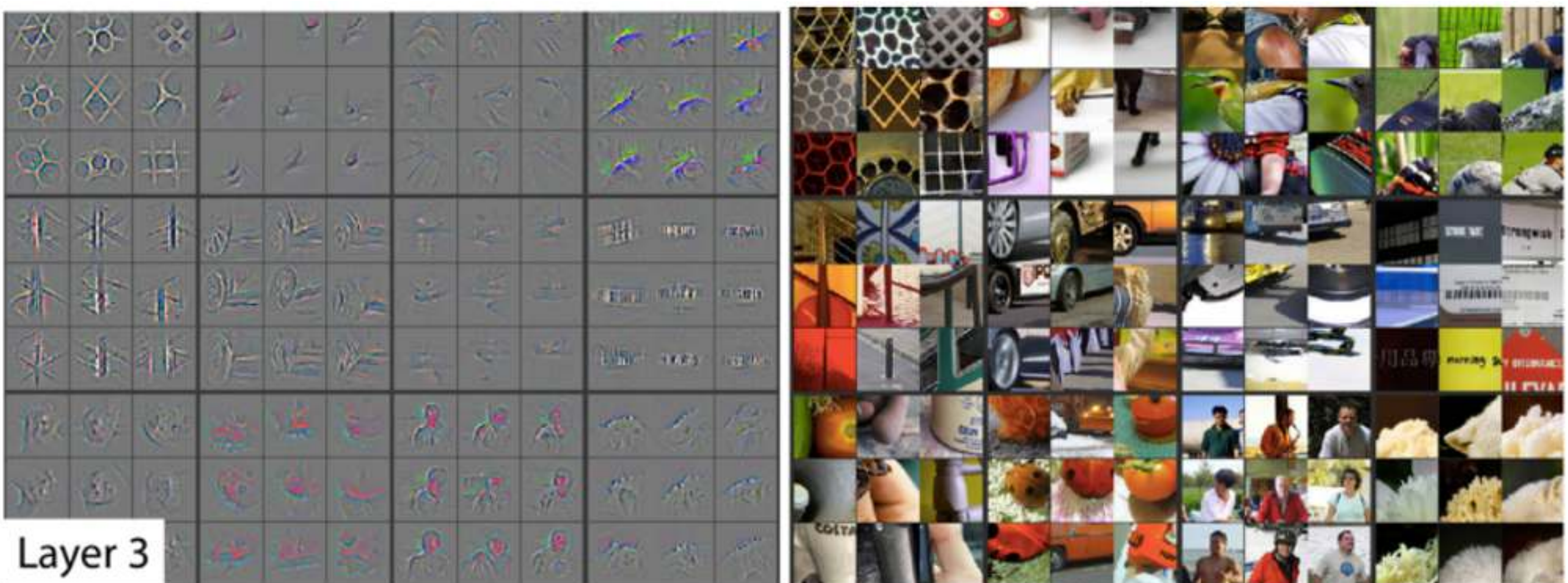

3
224
224
3
1
7
96
37
37
96
5
5
256
17
17
256
3
3
512
17
17
512
3
3
512
17
17
512
3
3
512
6
6
4096
1
1
4096
1
1
1, 2, 3, 4
1, 2, 4
1, 2
1, 2
1, 2, 4
5, 2
5, 2
5, 6
Class labels
1000

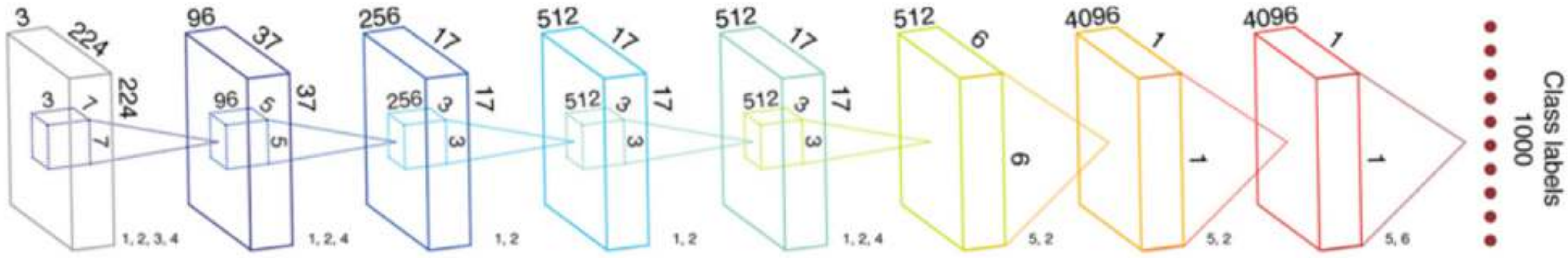
Layer 1

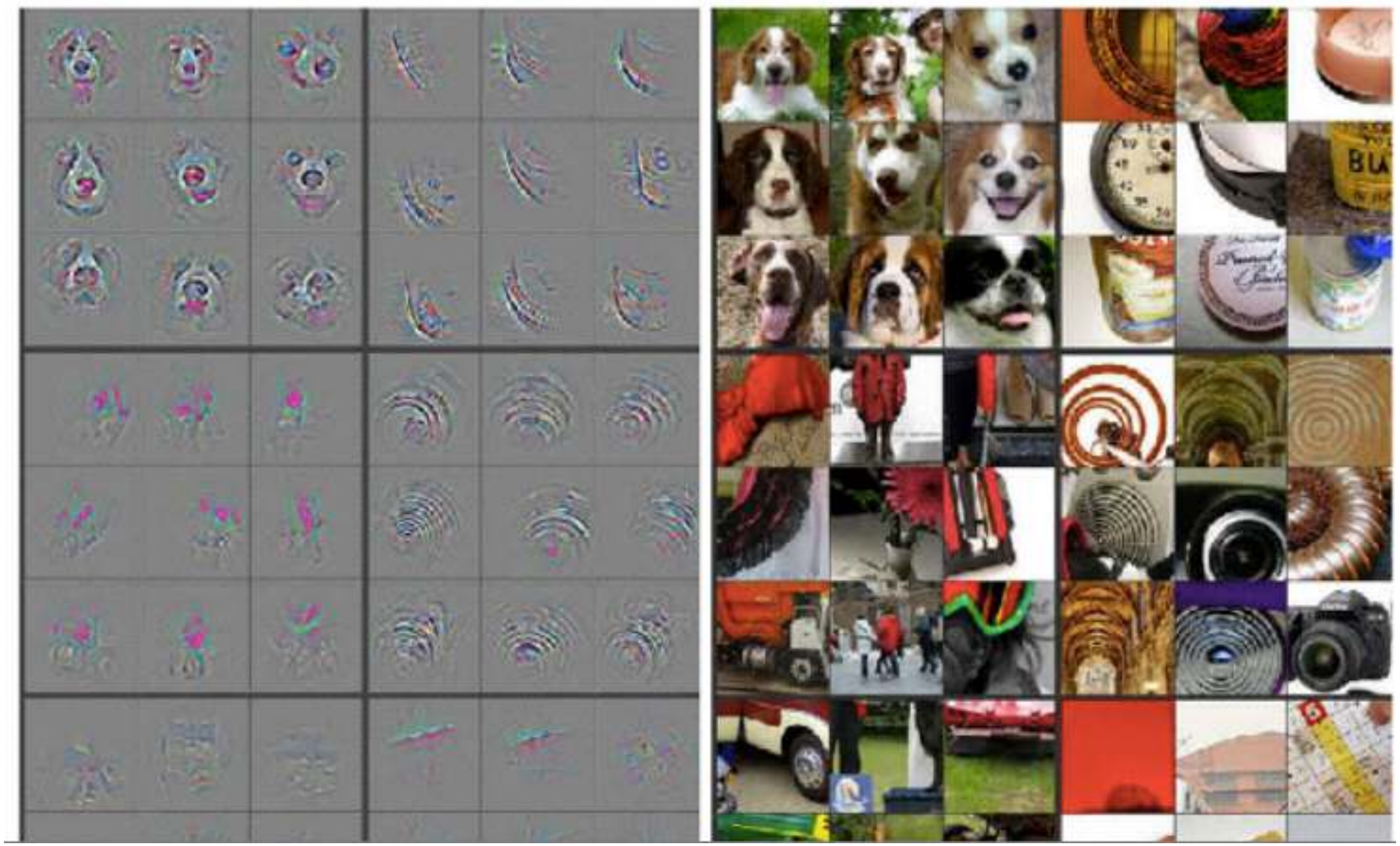

3
224
224
96
37
37
256
17
17
512
17
17
512
17
17
512
6
6
4096
1
1
4096
1
1
1, 2, 3, 4
1, 2, 4
1, 2
1, 2
1, 2, 4
5, 2
5, 2
5, 6
Class labels
1000

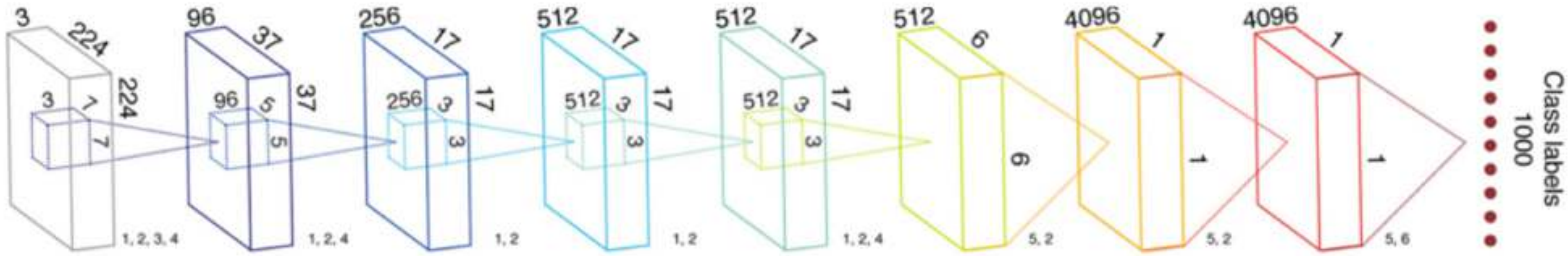

Layer 2

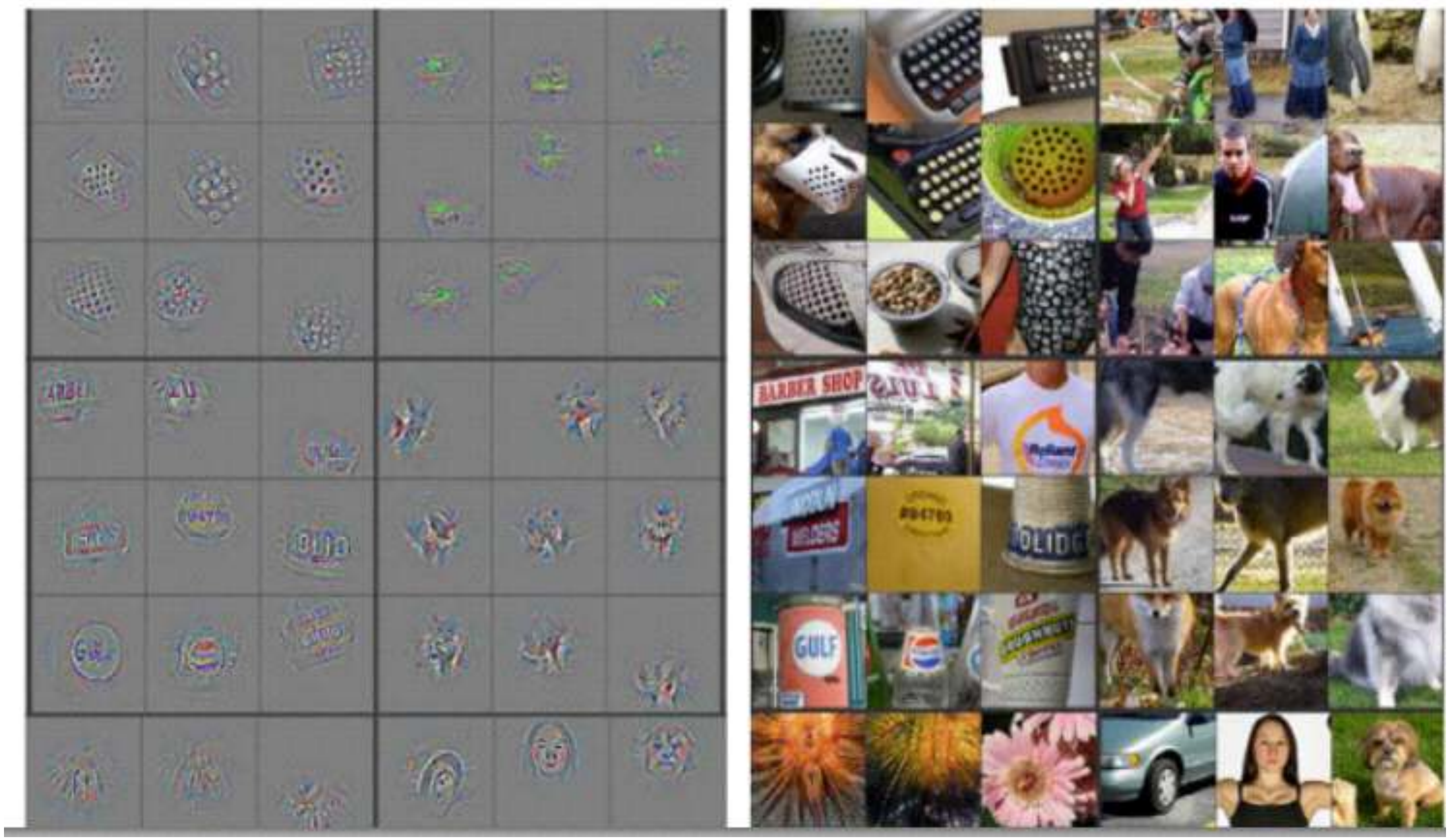

3
224
224
3
7
7
1, 2, 3, 4
96
37
37
96
5
5
1, 2, 4
256
17
17
256
3
3
1, 2
512
17
17
512
3
3
1, 2
512
17
17
512
3
3
1, 2, 4
512
6
6
5, 2
4096
1
1
5, 2
4096
1
1
5, 6
Class labels
1000


Layer 3


3
224
224
3
7
7
1, 2, 3, 4
96
37
37
96
5
5
1, 2, 4
256
17
17
256
3
3
1, 2
512
17
17
512
3
3
1, 2
512
17
17
512
3
3
1, 2, 4
512
6
6
5, 2
4096
1
1
5, 2
4096
1
1
5, 6
Class labels
1000

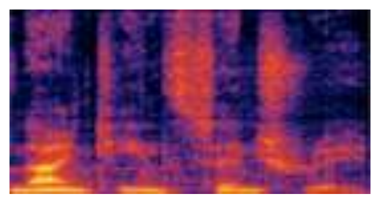
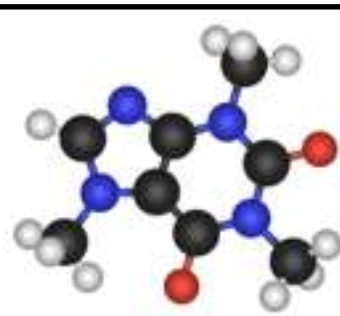



3
224
224
96
37
37
256
17
17
512
17
17
512
17
17
512
6
6
4096
1
1
4096
1
1
3
7
7
96
5
5
256
3
3
512
3
3
512
3
3
1, 2, 3, 4
1, 2, 4
1, 2
1, 2
1, 2, 4
5, 2
5, 2
5, 6
Class labels
1000



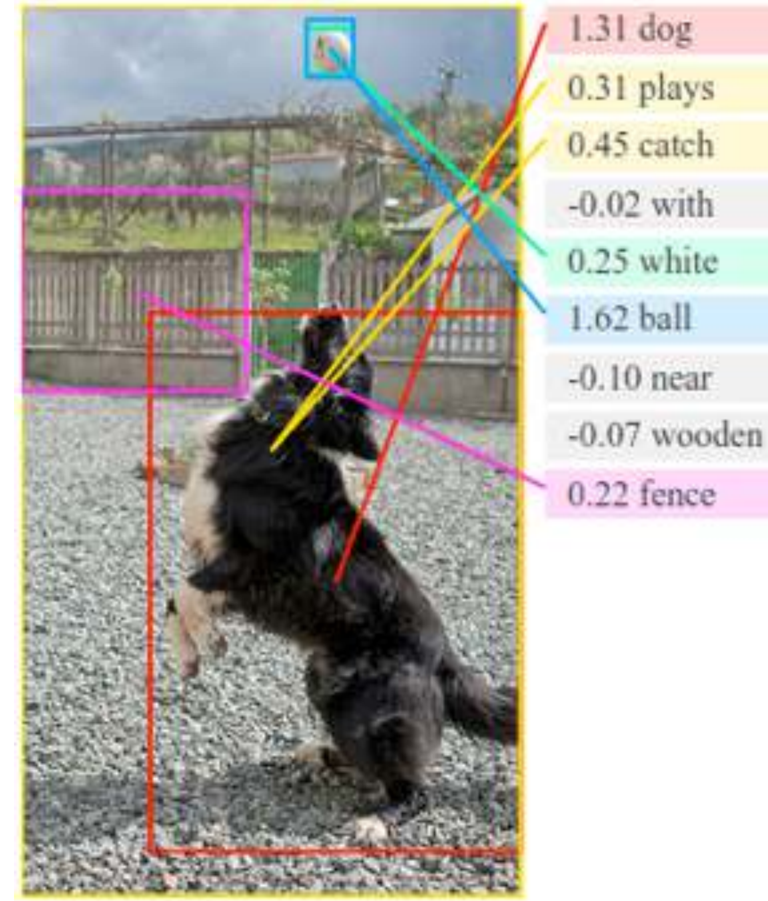
False positives

False negatives

<table>
<tr><th>Input</th><th>Output</th></tr>
<tr><td>Pixels: </td><td>“frog”</td></tr>
<tr><td>Audio: </td><td>“na in tiz   sen tu rii”</td></tr>
<tr><td>“Buenos dias,
que tal estas?”</td><td>“Guten morgen,
wie geht es dir?”</td></tr>
<tr><td></td><td>“Toxic”</td></tr>
<tr><td>*<pre>j=8584
for x in range(8):
  j+=920
b=(1500+j)
print((b+7567))</pre></td><td>25011.</td></tr>
</table>

"man in black shirt is playing guitar."

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

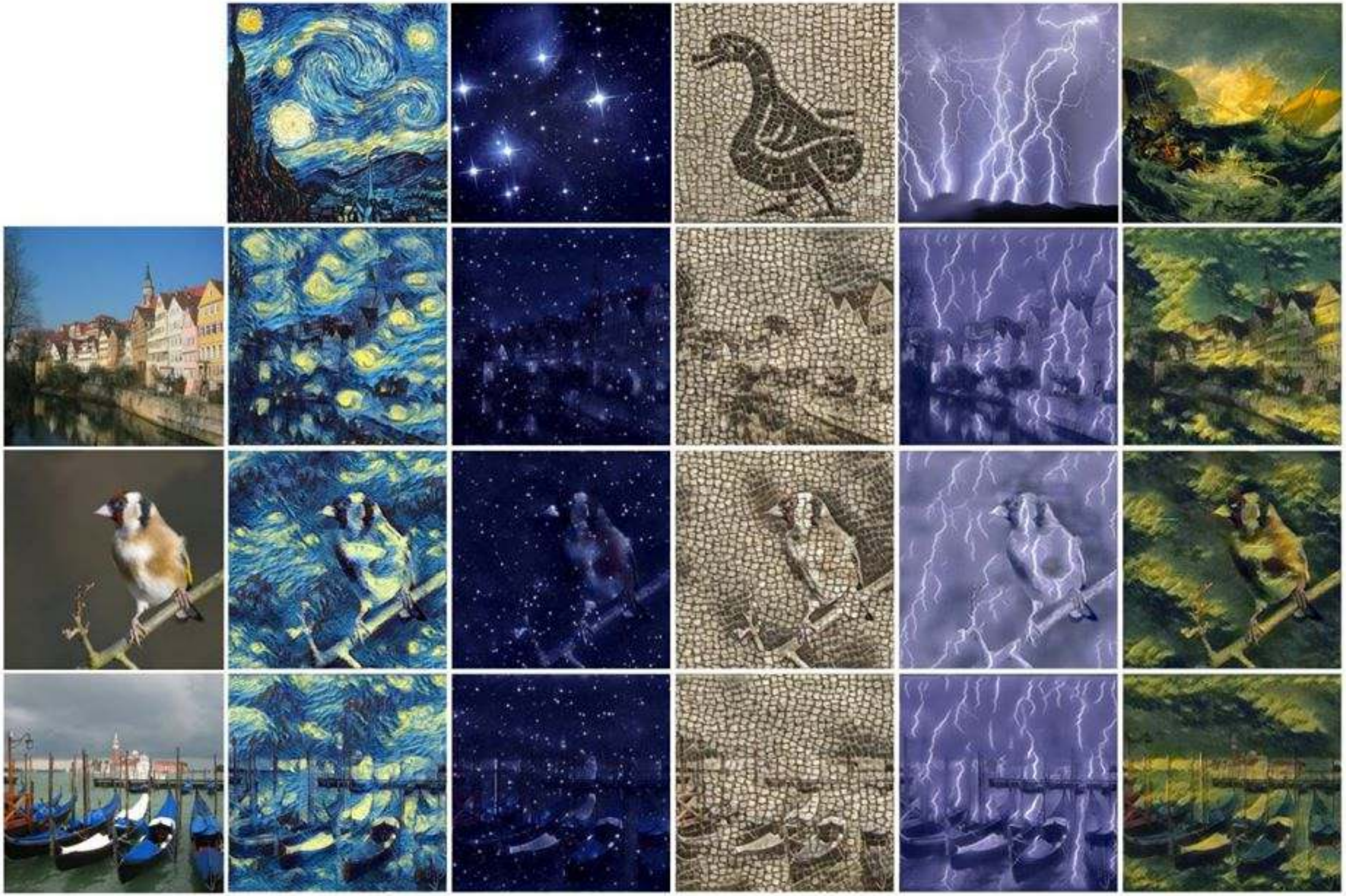


"girl in pink dress is jumping in air."

"black and white dog jumps over bar."

Karpathy, Fei-Fei, "Deep Visual-Semantic Alignments for Generating Image Descriptions" (2014)

Style transferring



Визуализация работы Texture Networks by Dmitry Ulyanov et al.

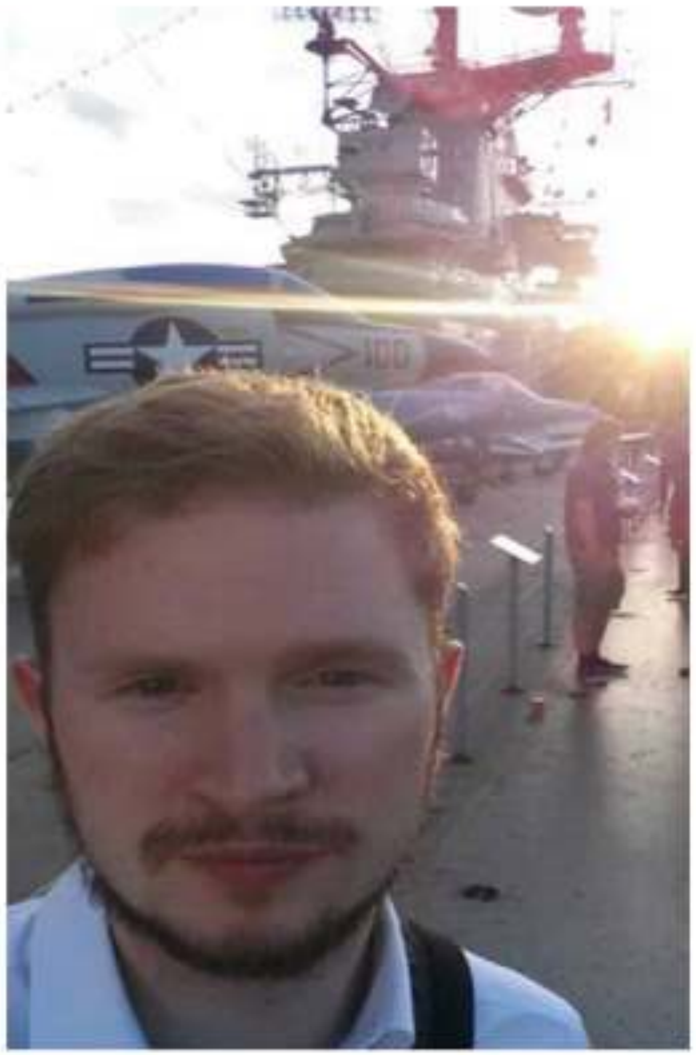
Visual and Textual Question Answering

what is on the table?

Answer

Answer	Confidence
laptop	0.2342
food	0.0387
no	0.0176
cat	0.0149
yes	0.0136

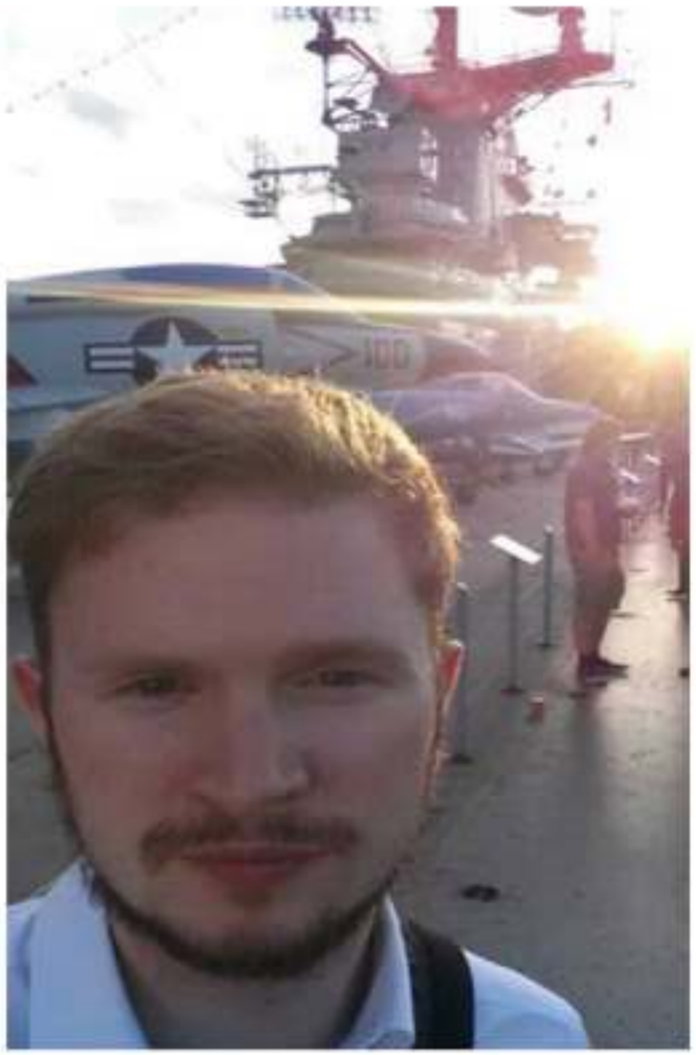
Visual and Textual Question Answering



Where is a man?

Answer

Visual and Textual Question Answering



Where is a man?

Answer

Answer	Confidence
airport	0.0114
outside	0.0110
sidewalk	0.0096
skate park	0.0077
car	0.0077

http://cloudcv.org/vqa/

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
PROJECT MAC

Artificial Intelligence Group July 7, 1966
Vision Memo. No. 100.

THE SUMMER VISION PROJECT

Seymour Papert

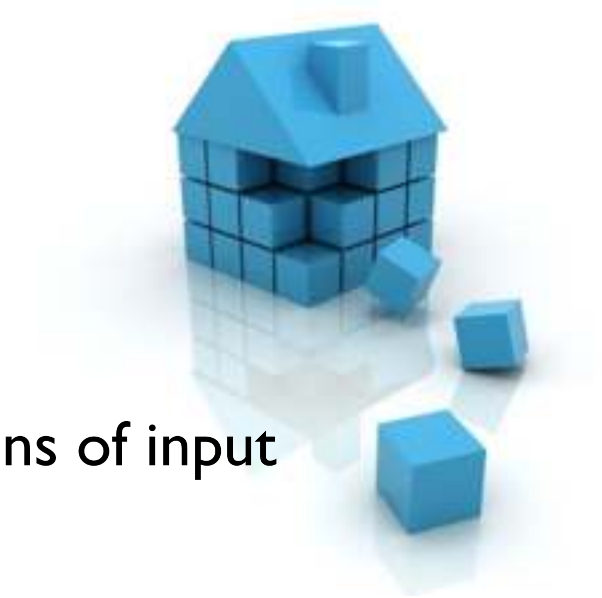
The summer vision project is an attempt to use our summer workers effectively in the construction of a significant part of a visual system.

The final goal is OBJECT IDENTIFICATION which will actually name objects by matching them with a vocabulary of known objects.

If all you have is a hammer in the toolbox, everything looks like a nail.”

– Bernard Baruch

Best potential when:



- the structure of your problem/data naturally maps to a multilayer architecture

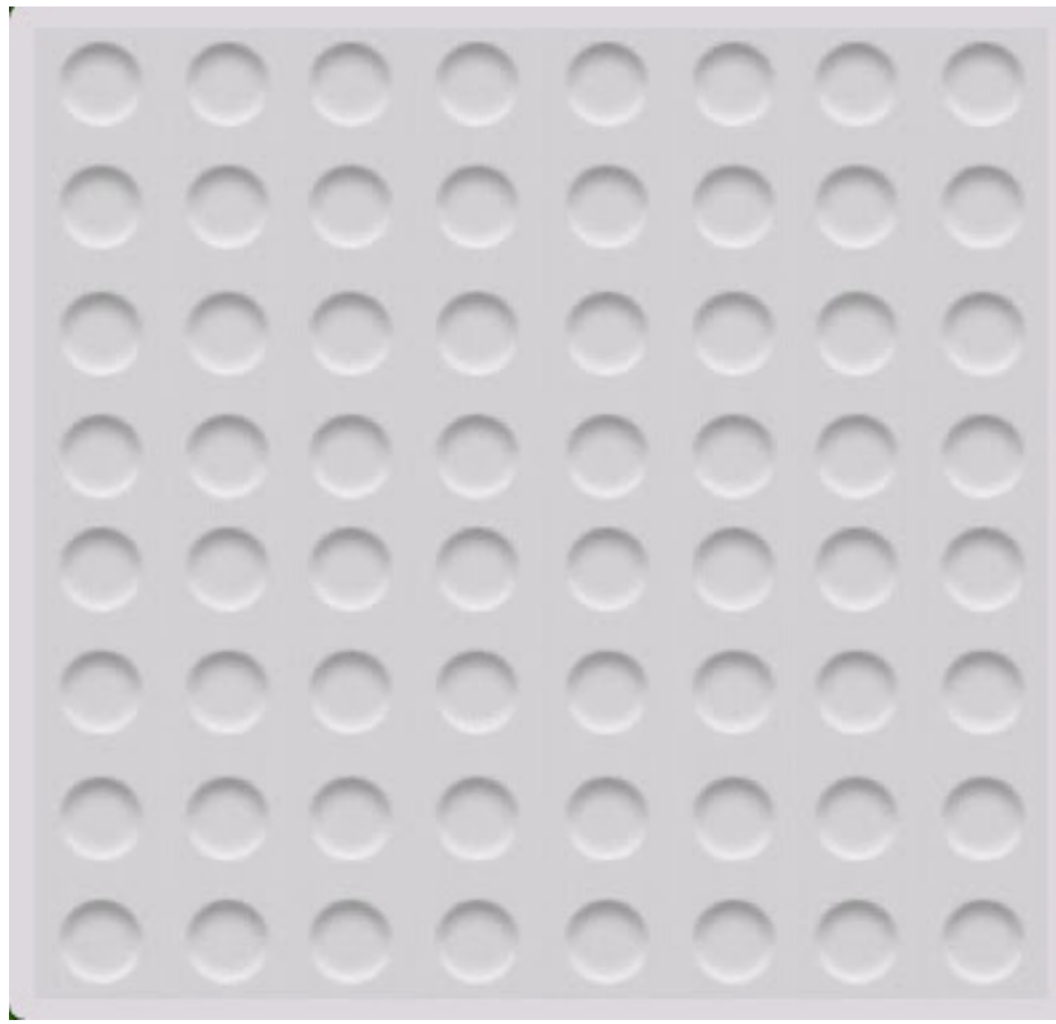
  - hierarchy of abstract features derivable from non-linear transformations of input

- enough data to learn features

  - unlabeled data can also be used for learning features

Why Deep Learning works so well?

Brain takes into account "prior knowledge" to figure out what is the world given ambiguous sensory data

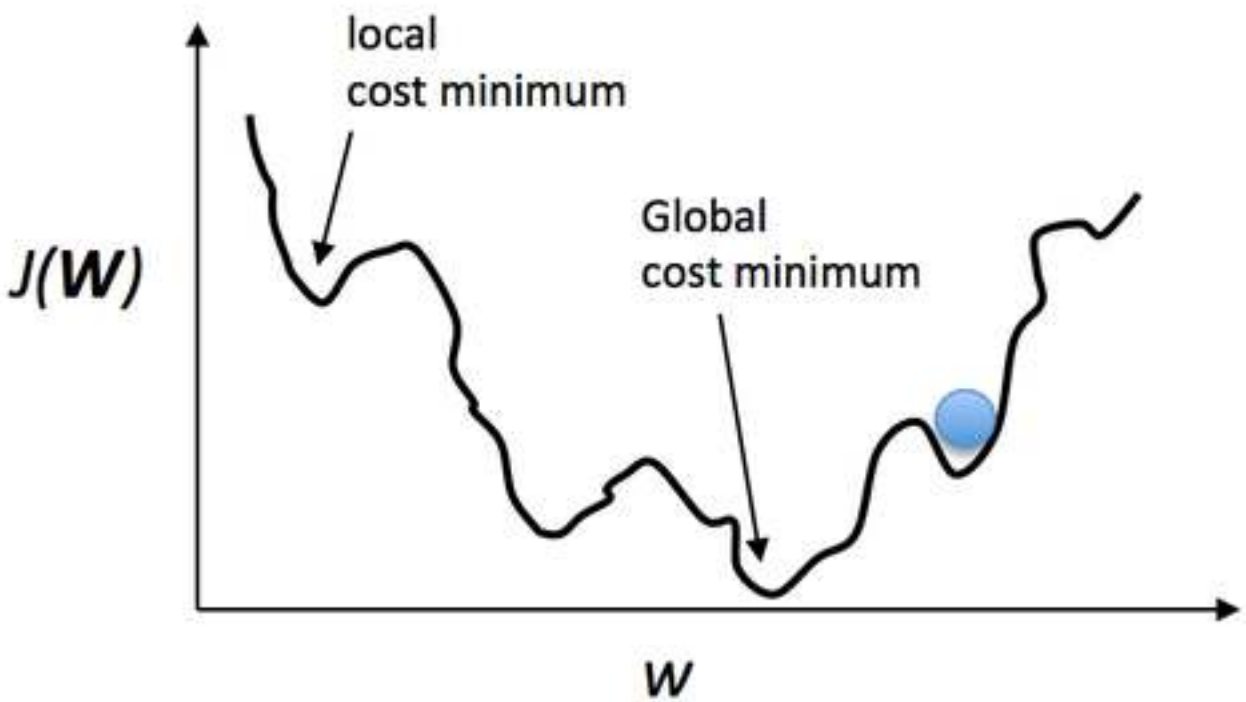


$$\overbrace{P(\text{world} \mid \text{sense data})}^{\text{posterior}} \propto \overbrace{P(\text{sense data} \mid \text{world})}^{\text{likelihood}}\ \overbrace{P(\text{world})}^{\text{prior}}$$

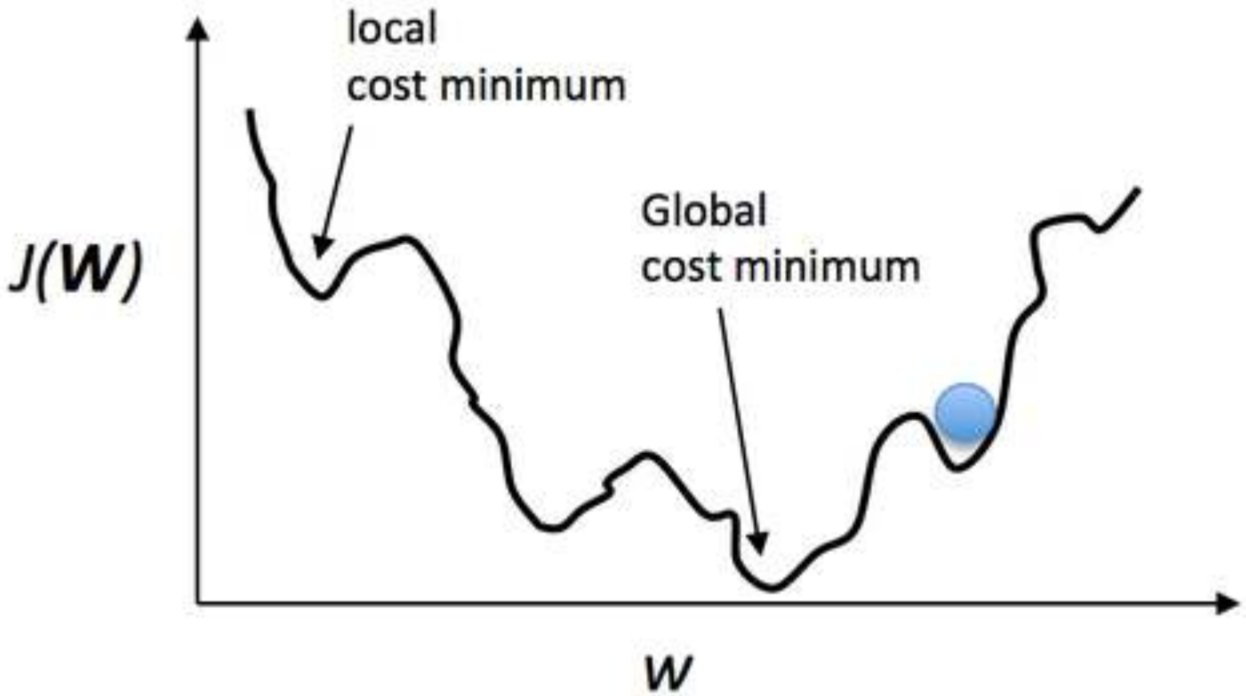
What “priors” are used in ML for generalizing to unseen data?

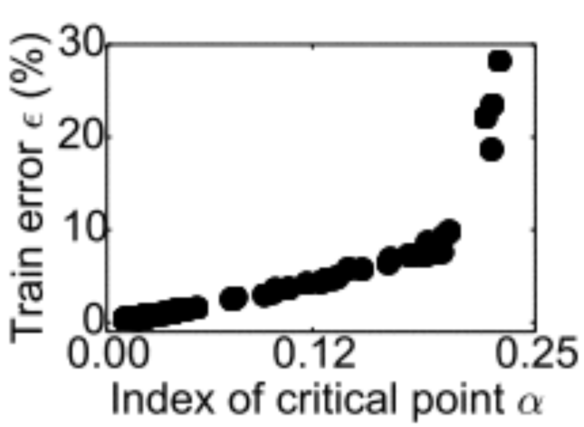
- No free lunch theorem: there is no ML algorithm that generalizes well for all data
- Luckily data in real world contains a lot of structure

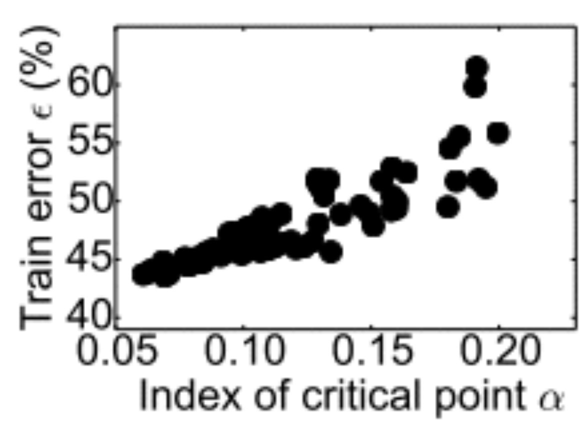
Why it learns at all?



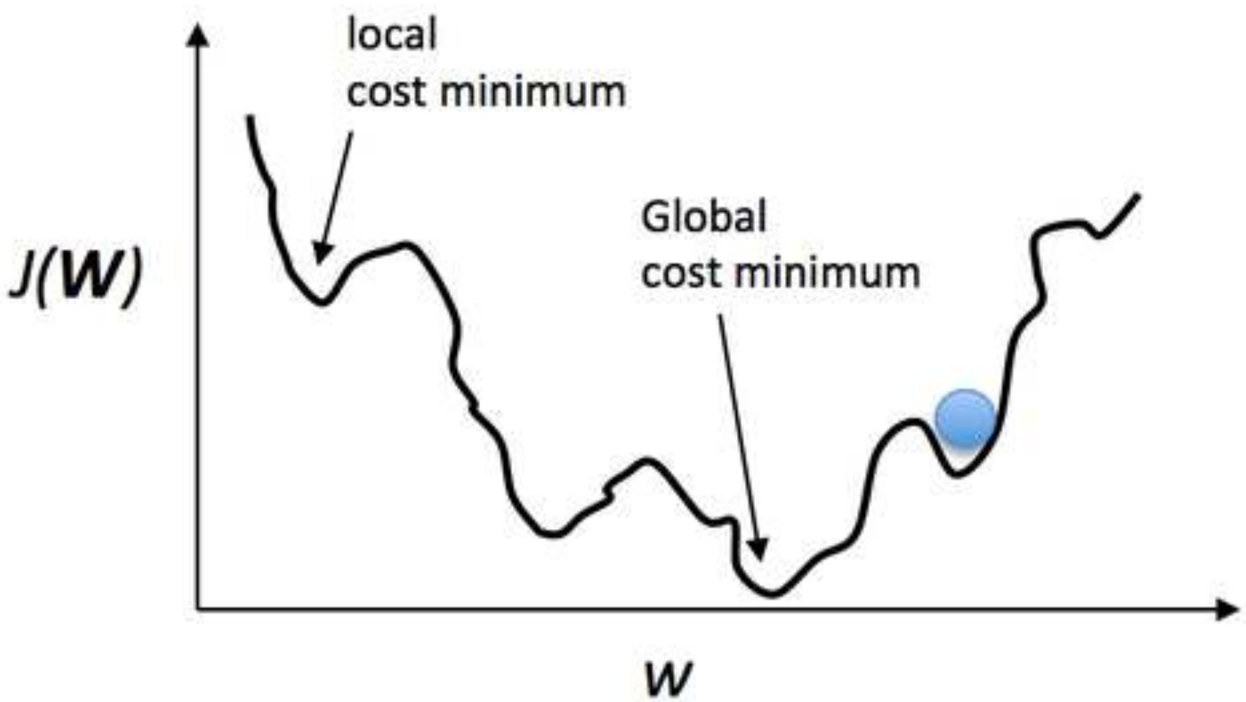
Why it learns at all?

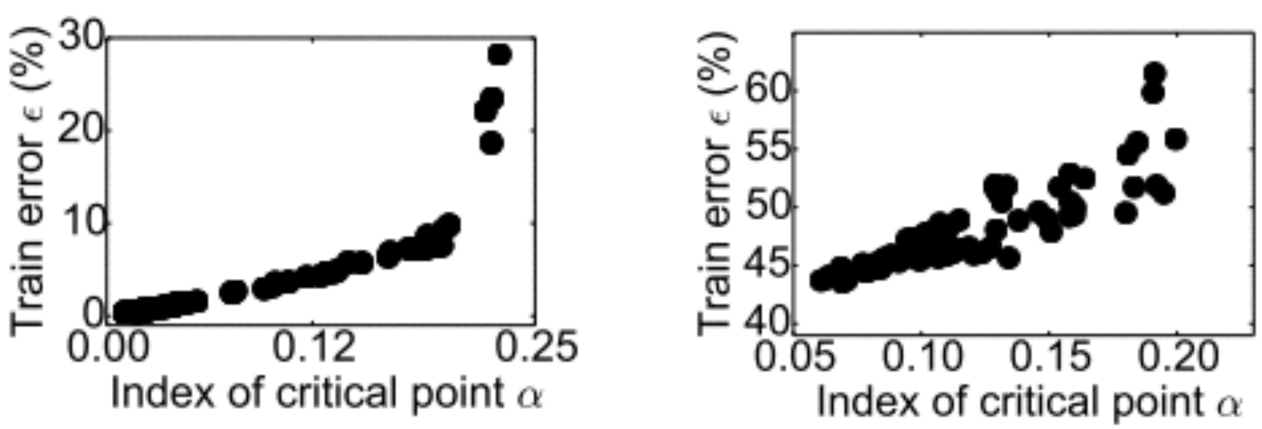






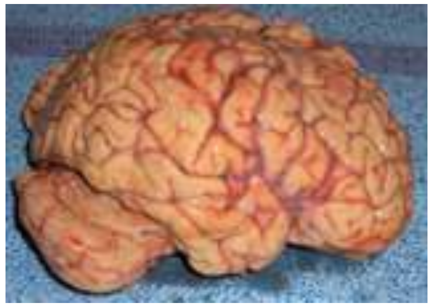
Why it learns at all?



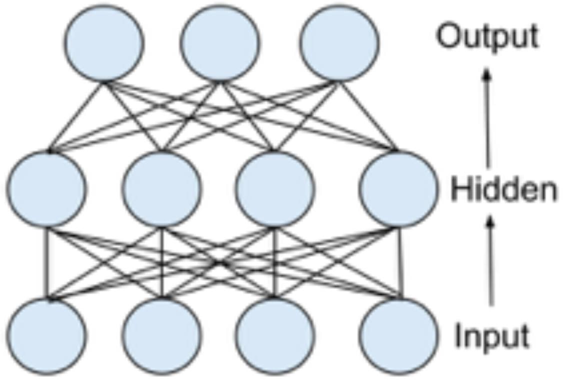


- For large networks most local minima are equivalent
- "Bad" local minima are exponentially improbable with network size
- Do not struggle to find absolute minimum!

Brains 101



Deep Learning

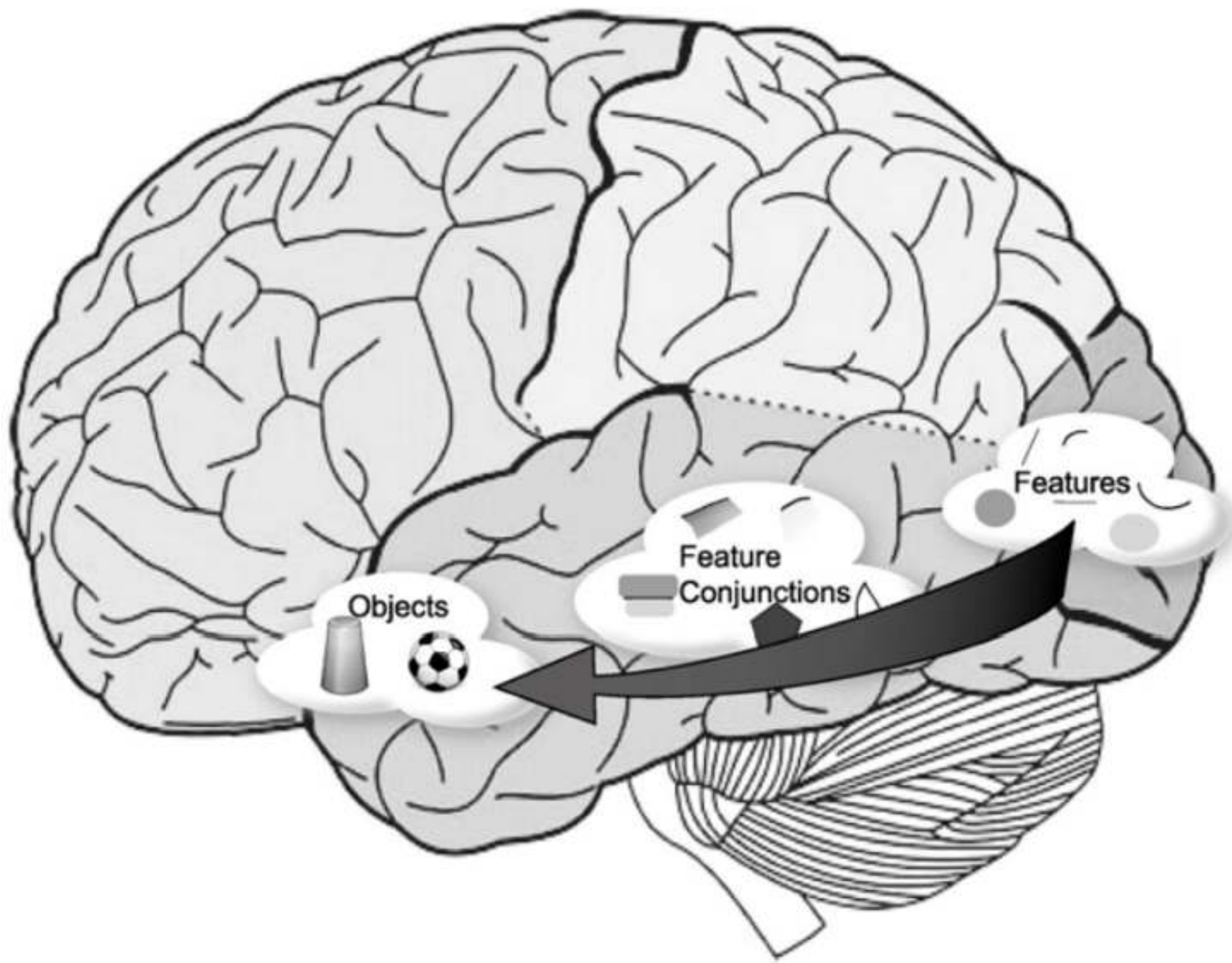


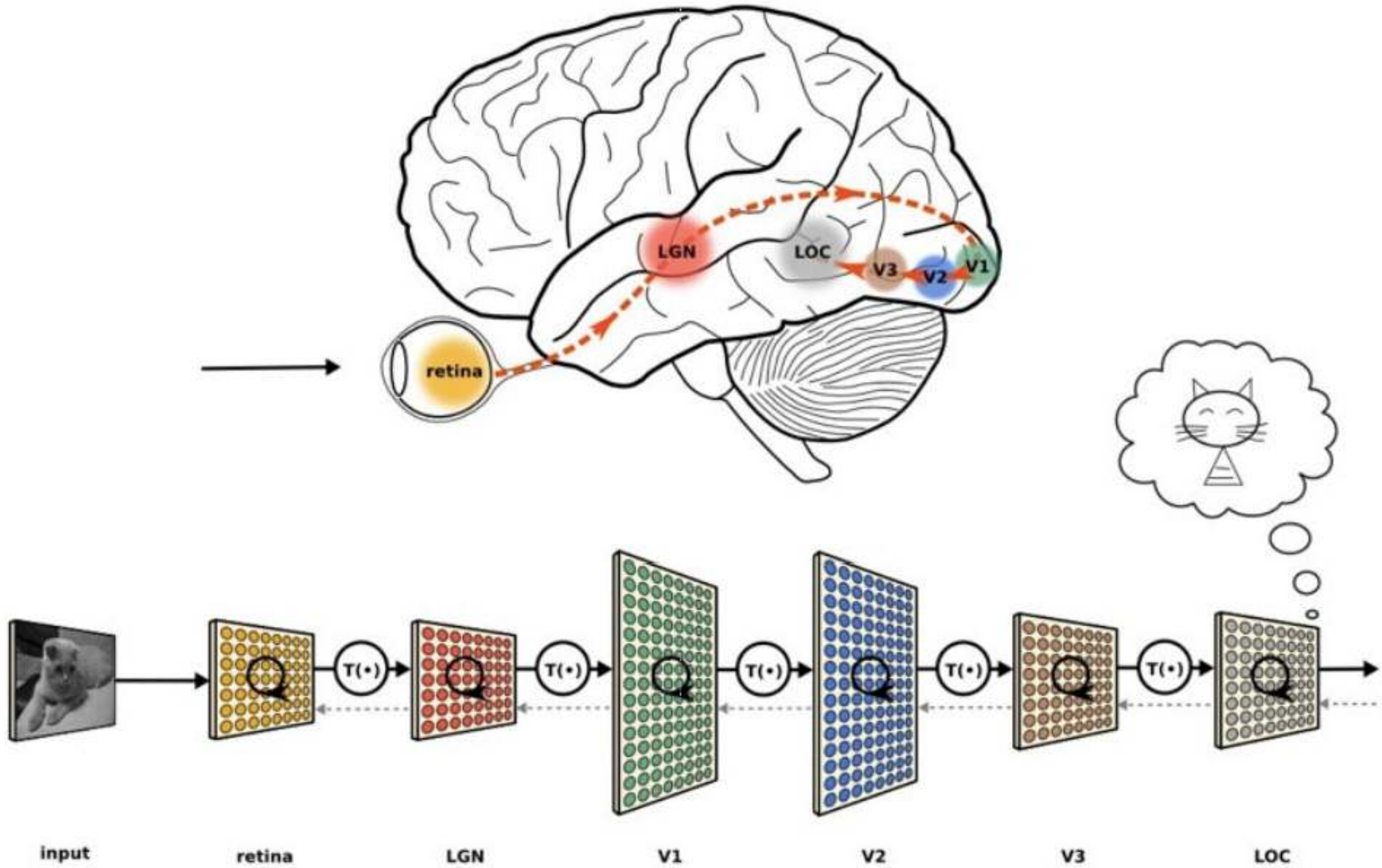
Feynman dictum

“What I cannot create,
I don’t understand”

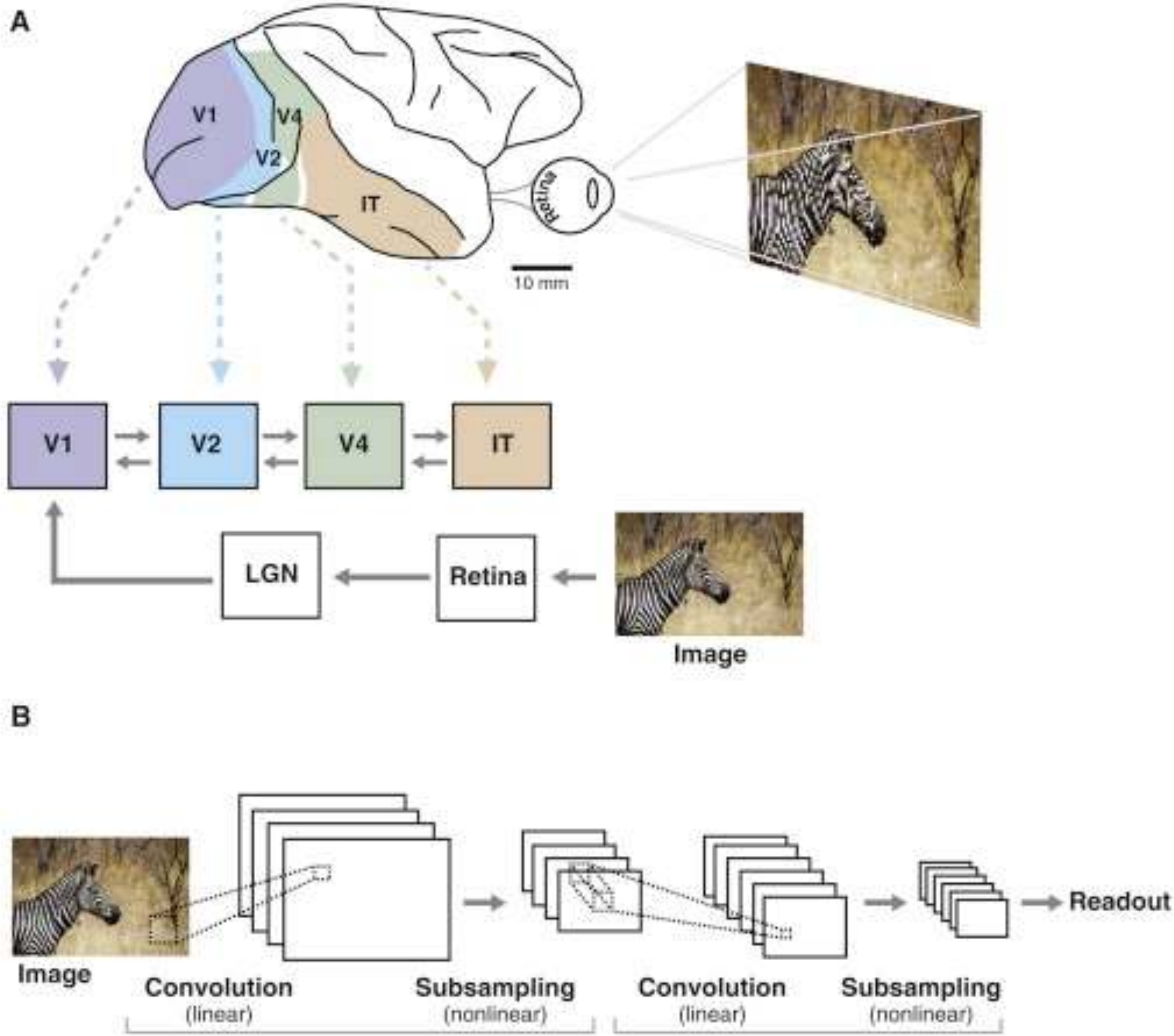
R. Feynman

How we perform object recognition?





retina
LGN
LOC
V3
V2
V1
T(•)
input
retina
LGN
V1
V2
V3
LOC



A
V1
V4
V2
IT
Retina
10 mm
V1
V2
V4
IT
LGN
Retina
Image
B
Image
Convolution
(linear)
Subsampling
(nonlinear)
Convolution
(linear)
Subsampling
(nonlinear)
Readout

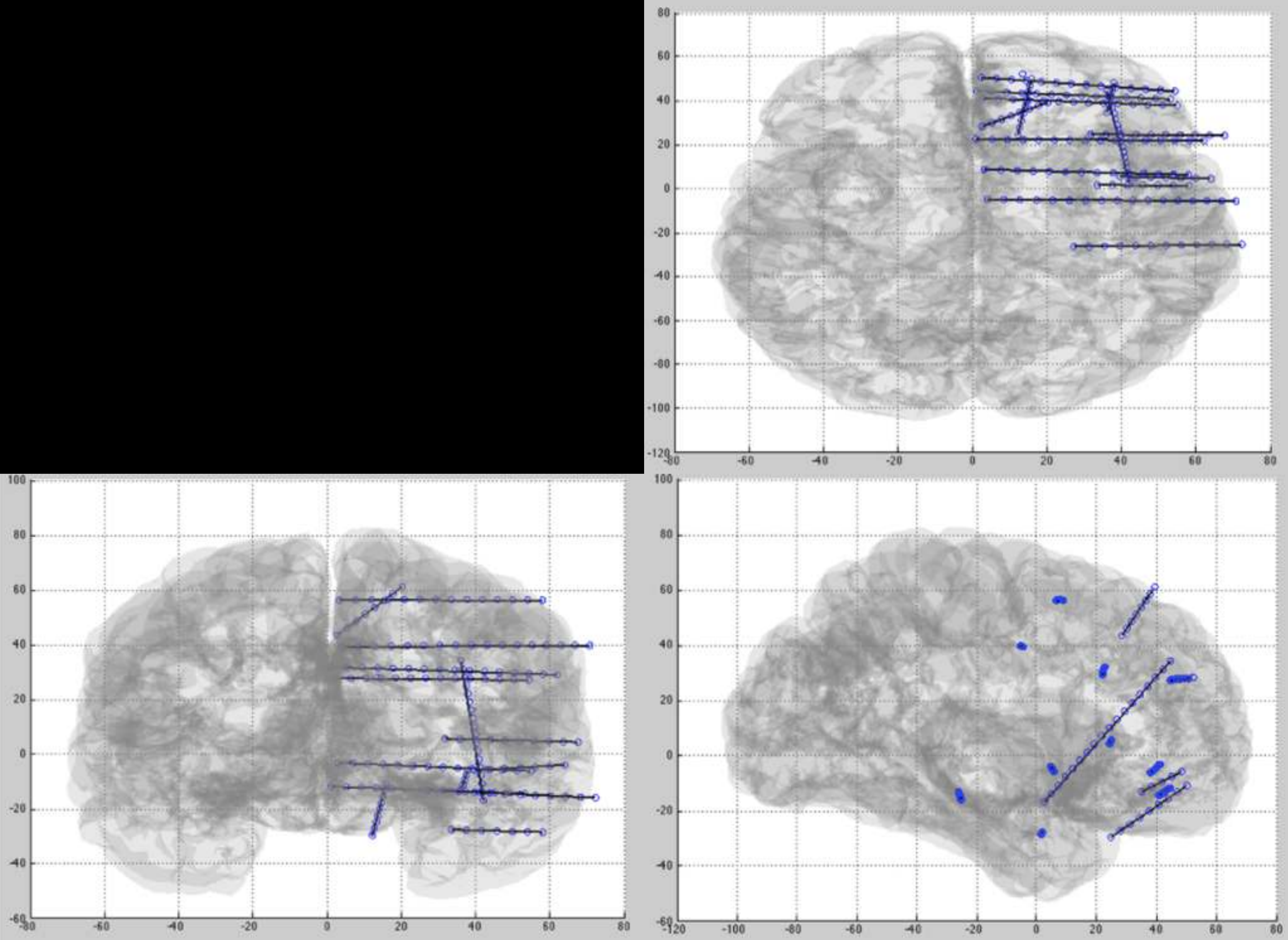
Are they processing stimuli similarly?

How would you compare them?

Question?

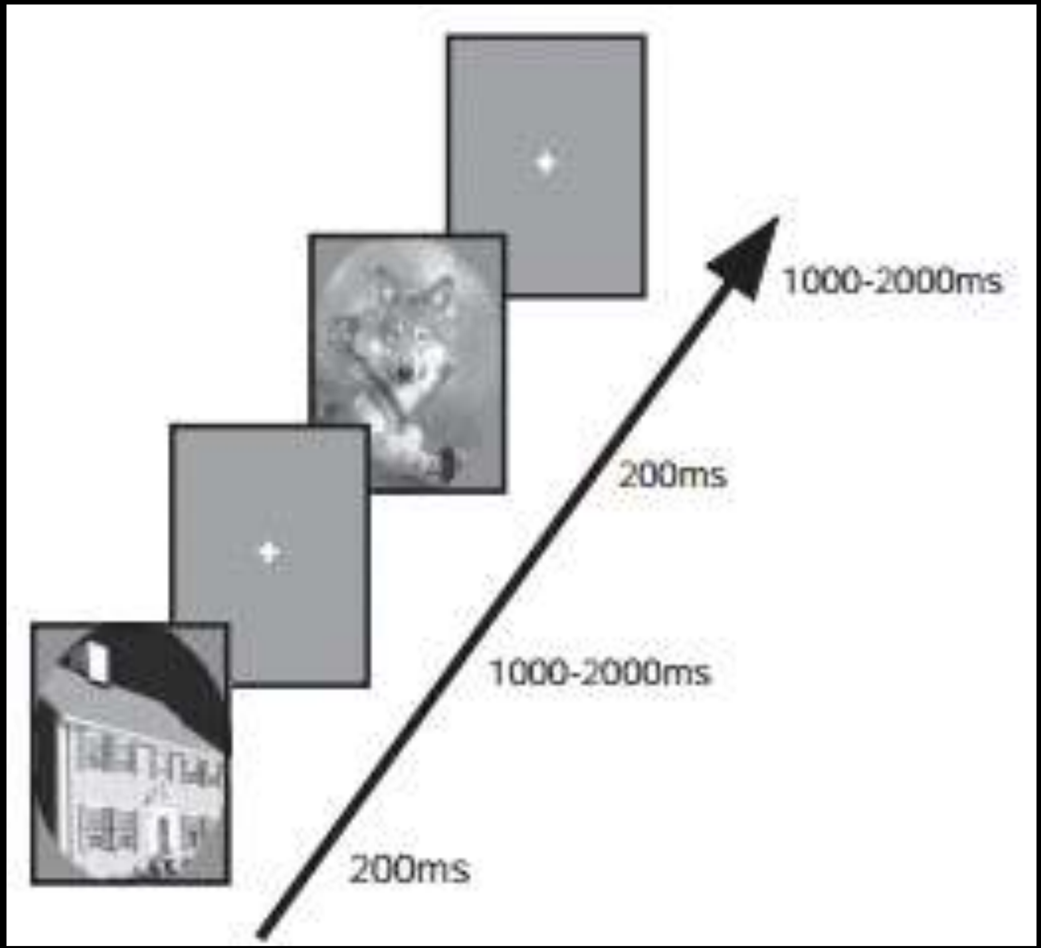
Are deep convolutional networks good models of biological vision?

Collaboration with University Hospital of Lyon



80 60 40 20 0 -20 -40 -60 -80 -100 -120
-80 -60 -40 -20 0 20 40 60 80
100 80 60 40 20 0 -20 -40 -60
-80 -60 -40 -20 0 20 40 60 80
100 80 60 40 20 0 -20 -40 -60
-120 -100 -80 -60 -40 -20 0 20 40 60 80

320 images
109 patients
>10000 total electrodes



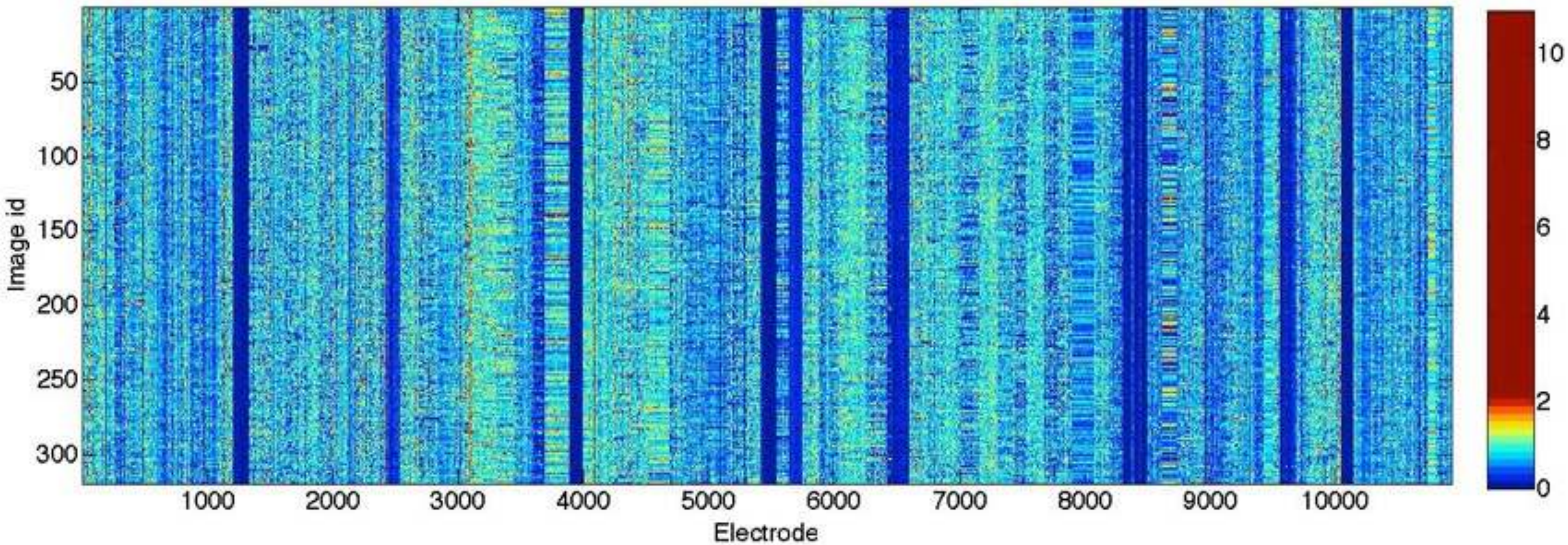
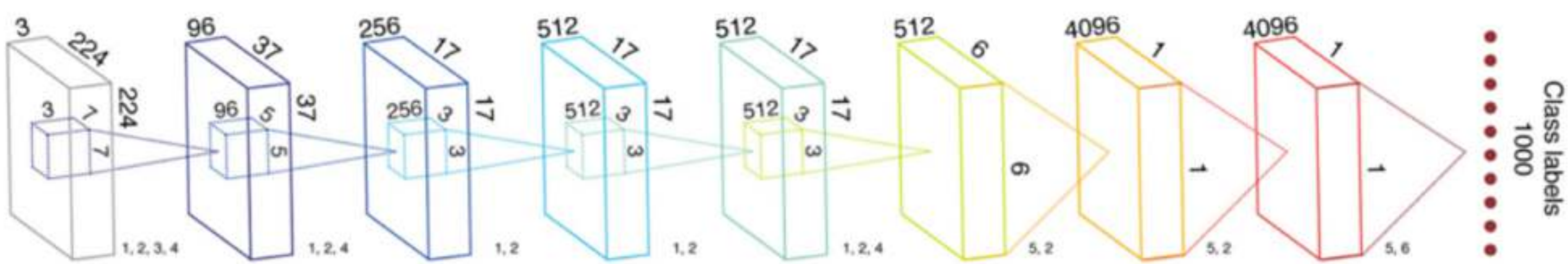
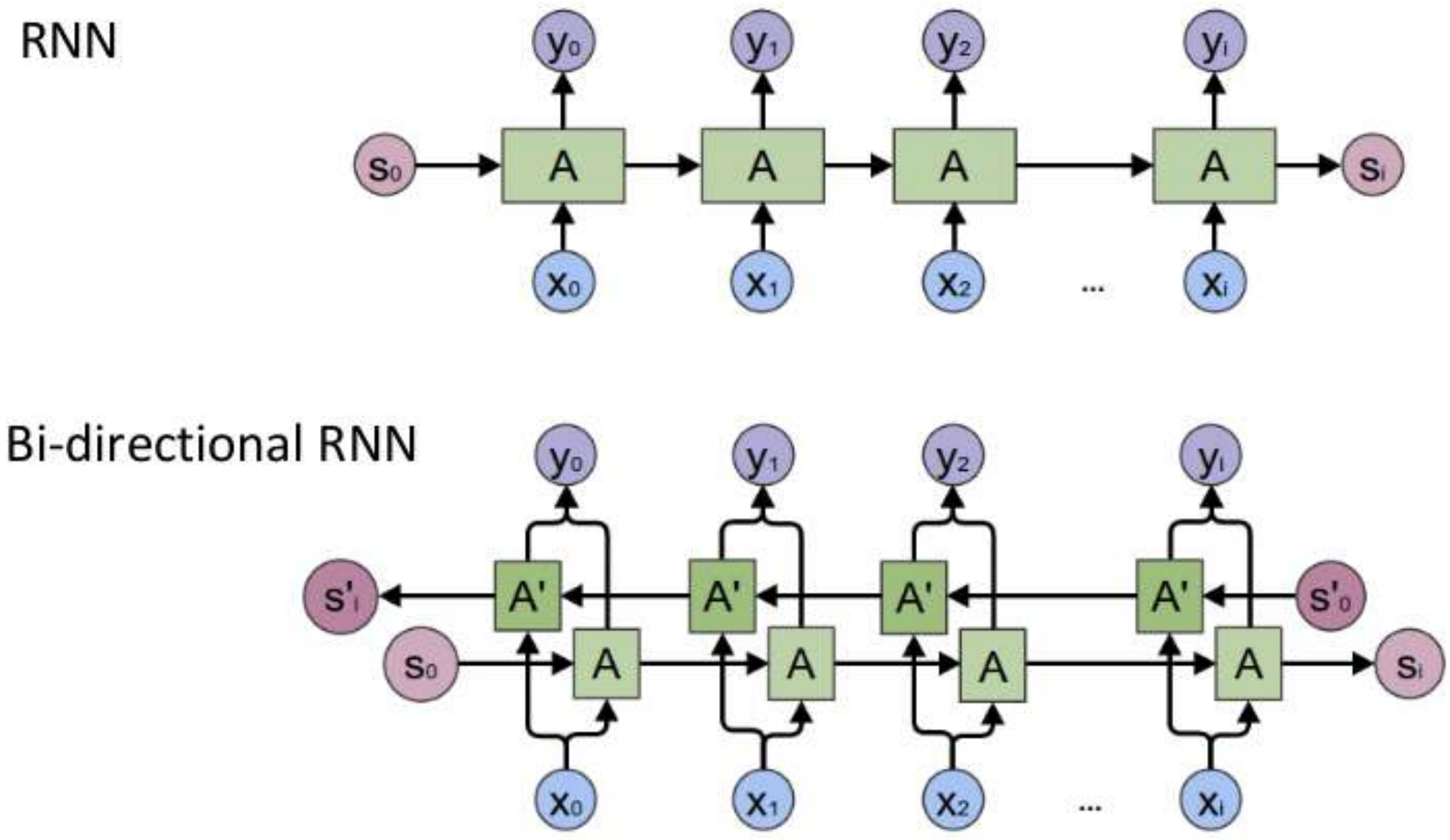


Image id
50
100
150
200
250
300
1000
2000
3000
4000
5000
6000
7000
8000
9000
10000
Electrode
0
2
4
6
8
10



3
224
224
3
1
7
1, 2, 3, 4
96
37
37
96
5
5
1, 2, 4
256
17
17
256
3
3
1, 2
512
17
17
512
3
3
1, 2
512
17
17
512
3
3
1, 2, 4
512
6
6
5, 2
4096
1
1
5, 2
4096
1
1
5, 6
Class labels
1000

Recurrent Neural Networks



4.4MB Shakespeare texts

Shakespeare?

Recurrent Neural Network

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.

VIOLA.
On your attendance, my lord; here.

DUKE.
Stand you awhile aloof.—Cesario,
Thou know'st no less but all; I have unclasp'd
To thee the book even of my secret soul:
Therefore, good youth, address thy gait unto her;
Be not denied access, stand at her doors,
And tell them there thy fixed foot shall grow
Till thou have audience.

VIOLA.
Sure, my noble lord,
If she be so abandon'd to her sorrow
As it is spoke, she never will admit me.

DUKE.
Be clamorous and leap all civil bounds,
Rather than make unprofited return.

VIOLA.
Say I do speak with her, my lord. What then?

Generated text is from The Unreasonable Effectiveness of Recurrent Neural Networks http://karpathy.github.io/2015/05/21/rnn-effectiveness/ by Andrej Karpathy

Shakespeare?

Recurrent Neural Network

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.

VIOLA.
On your attendance, my lord; here.

DUKE.
Stand you awhile aloof.—Cesario,
Thou know'st no less but all; I have unclasp'd
To thee the book even of my secret soul:
Therefore, good youth, address thy gait unto her;
Be not denied access, stand at her doors,
And tell them there thy fixed foot shall grow
Till thou have audience.

VIOLA.
Sure, my noble lord,
If she be so abandon'd to her sorrow
As it is spoke, she never will admit me.

DUKE.
Be clamorous and leap all civil bounds,
Rather than make unprofited return.

VIOLA.
Say I do speak with her, my lord. What then?

Generated text is from The Unreasonable Effectiveness of Recurrent Neural Networks http://karpathy.github.io/2015/05/21/rnn-effectiveness/ by Andrej Karpathy

Linux

474MB of the source code

```
/*
 * Increment the size file of the new incorrect UI_FILTER group information
 * of the size generatively.
 */
static int indicate_policy(void)
{
  int error;
  if (fd == MARN_EPT) {
    /*
     * The kernel blank will coeld it to userspace.
     */
    if (ss->segment < mem_total)
      unblock_graph_and_set_blocked();
    else
      ret = 1;
    goto bail;
  }
  segaddr = in_SB(in.addr);
  selector = seg / 16;
  setup_works = true;
  for (i = 0; i < blocks; i++) {
    seq = buf[i++];
    bpf = bd->bd.next + i * search;
    if (fd) {
      current = blocked;
    }
  }
  rw->name = "Getjbbregs";
  bprm_self_clearl(&iv->version);
  regs->new = blocks[(BPF_STATS << info->historidac)] | PFMR_CLOBATHINC_SECONDS << 12;
  return segtable;
}
```

Generated text is from The Unreasonable Effectiveness of Recurrent Neural Networks http://karpathy.github.io/2015/05/21/rnn-effectiveness/ by Andrej Karpathy

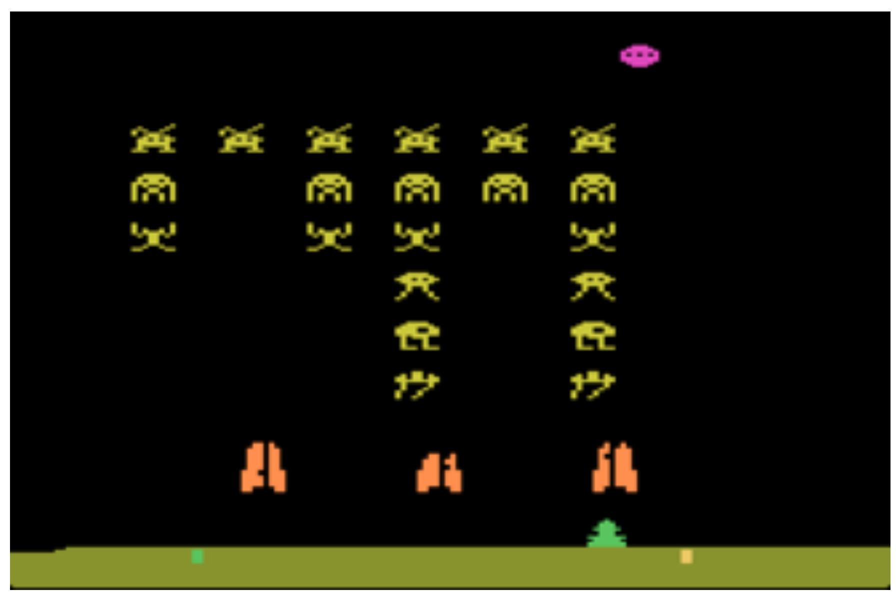
Playing Atari with Deep Reinforcement Learning

Pong

Breakout

Seaquest

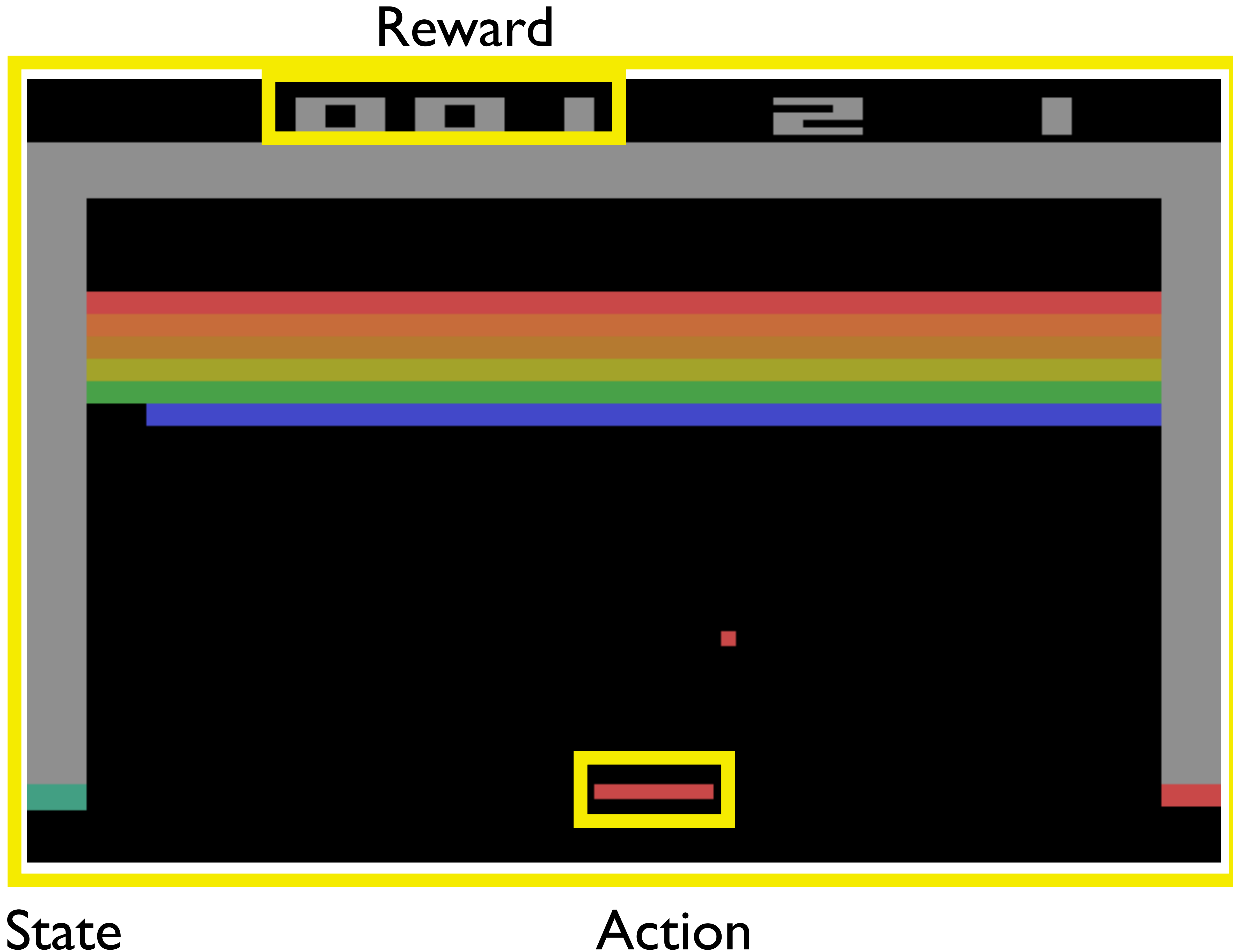
Beam Rider



Space Invaders

Enduro

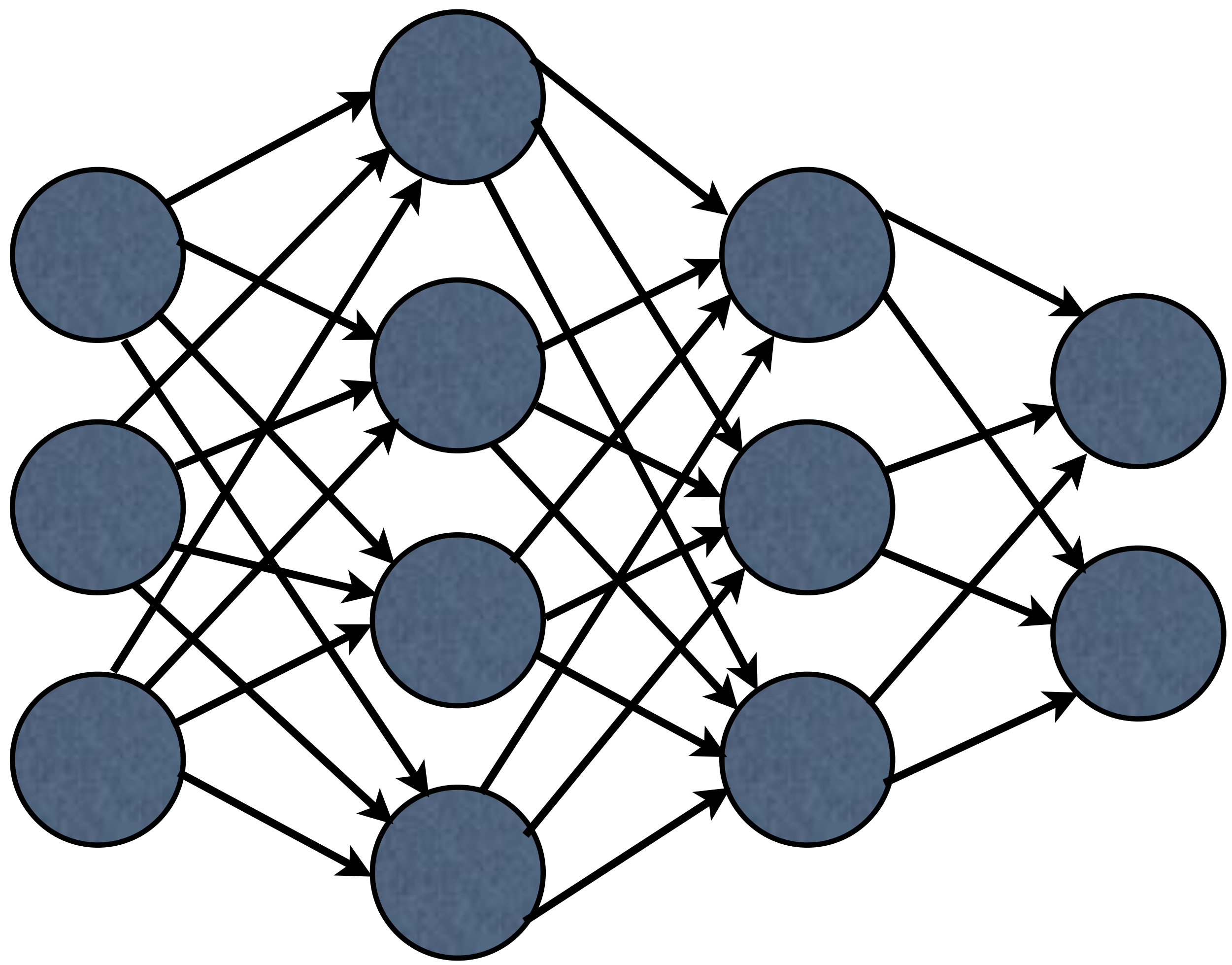
self-taught AI that learns to play better than humans


Reward
001
2
1
State
Action

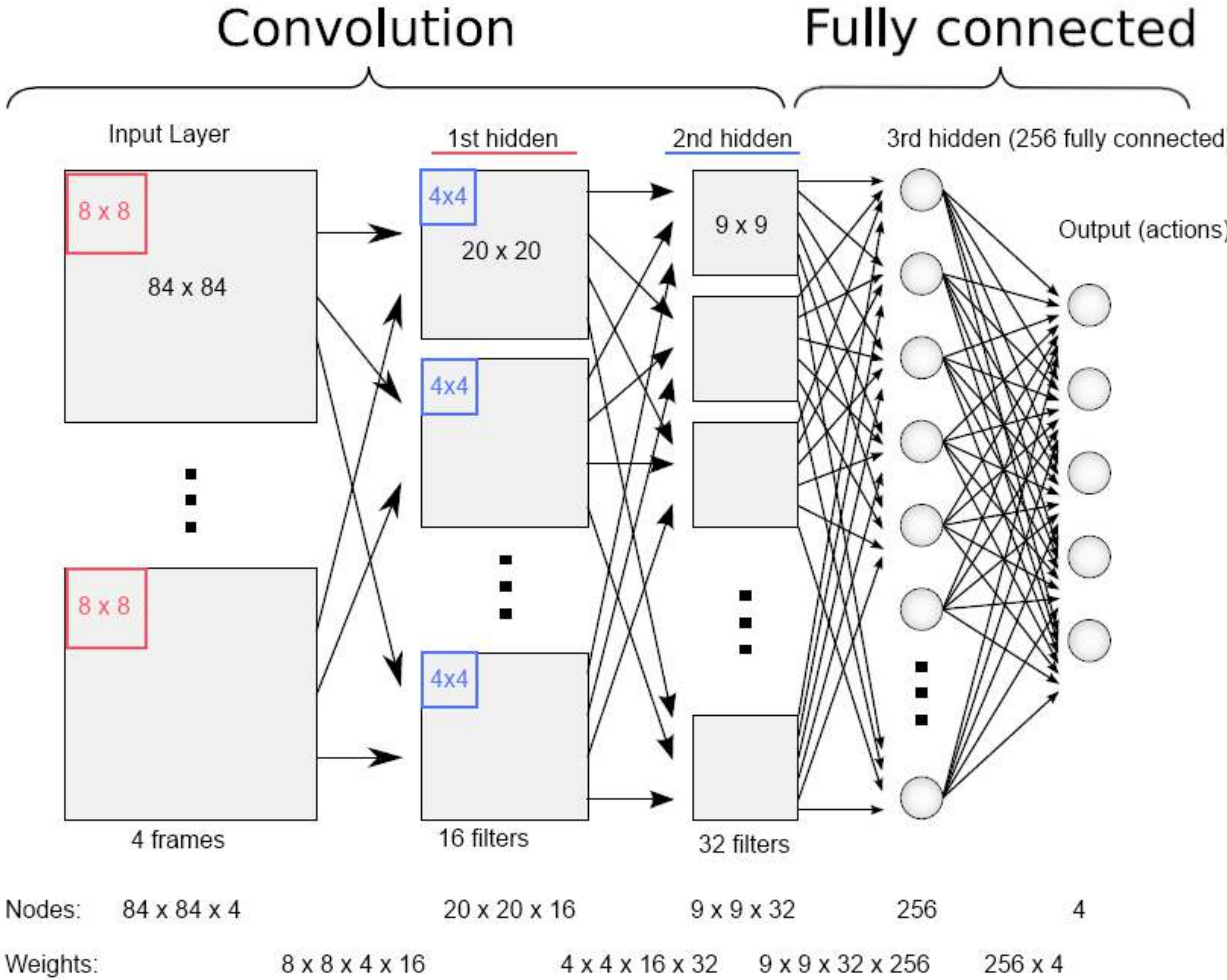
DeepMind

Train DeepMind

Input:



Output:

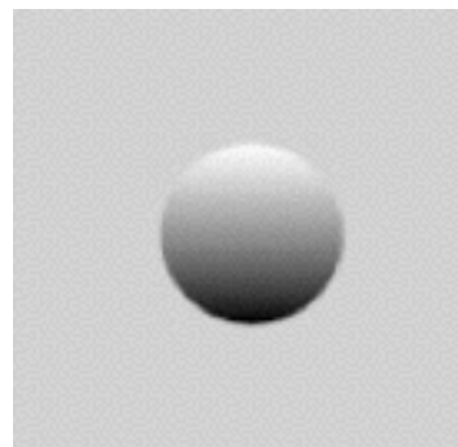


Convolution
Fully connected
Input Layer
1st hidden
2nd hidden
3rd hidden (256 fully connected)
8 x 8
84 x 84
8 x 8
4 frames
4x4
20 x 20
4x4
4x4
16 filters
9 x 9
32 filters
Output (actions)
Nodes: 84 x 84 x 4
20 x 20 x 16
9 x 9 x 32
256
4
Weights: 8 x 8 x 4 x 16
4 x 4 x 16 x 32
9 x 9 x 32 x 256
256 x 4

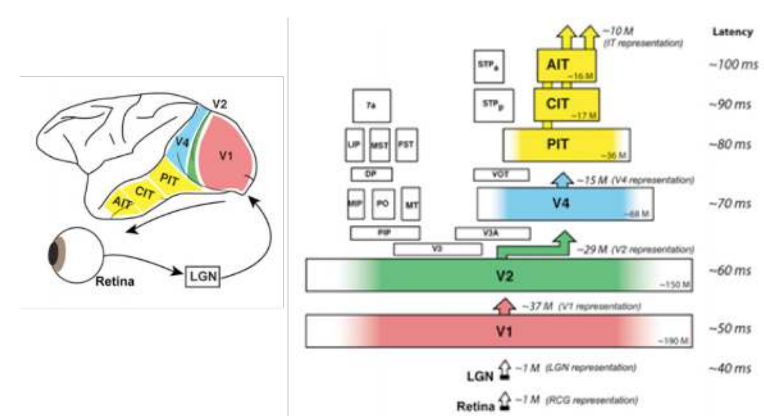
Training CALL OF DUTY

Take home message (I)

- Brains excel at solving ill-posed inference problems

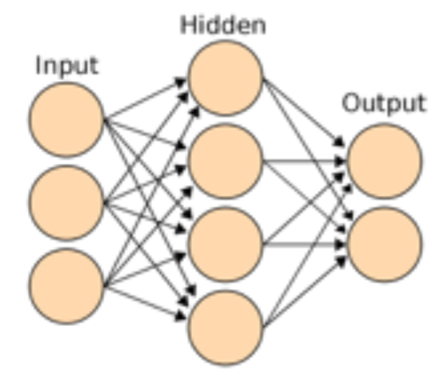


- Sample the environment and store "priors" in their hierarchical/recurrent synaptic matrix



Take home message (II)

- Deep Learning = Neural Networks 3.0



- Feature learning & approximate wild functions

- Pushing ML and AI to unthinkable applications

Take home message (III)

ANNs can be a useful tool for studying brain functions:

- biological visual recognition
- spatial navigation
- emergence of cooperation & communication

Tools

- **Keras** (http://keras.io/)
- **Torch** (http://torch.ch/)
- **Theano** (http://deeplearning.net/software/theano/)
- **Caffe** (http://caffe.berkeleyvision.org/)

Deep Learning Libraries by programming language: http://www.teglor.com/b/deep-learning-libraries-language-cm569

Online courses

Machine Learning
by Andrew Ng
https://www.coursera.org/learn/machine-learning

CS231n: Convolutional Neural Networks for Visual Recognition
by Andrej Karpathy
http://cs231n.github.io/

Neural Networks for Machine Learning
by Geoffrey Hinton
https://www.coursera.org/course/neuralnets

CS224d: Deep Learning for Natural Language Processing
by Richard Socher
http://cs224d.stanford.edu/

Offline places

Summer School in Computer Science
at CS@UCU
http://cs.ucu.edu.ua/en/

The Deep Learning Summer School
at Montreal, Canada
https://sites.google.com/site/deeplearningsummerschool/

International Conference on Machine Learning
at New York, USA
http://icml.cc/2016/

International Conference on Learning Representations
at San Juan, Puerto Rico
http://www.iclr.cc/doku.php?id=start

Popular Blogs

10 Machine Learning Terms Explained in Simple English
by Aylien
http://blog.aylien.com/post/121281850733/10-machine-learning-terms-explained-in-simple

The Unreasonable Effectiveness of Recurrent Neural Networks
by Andrej Karpathy
http://karpathy.github.io/2015/05/21/rnn-effectiveness/

Understand LSTM Networks
by Christopher Olah
https://colah.github.io/posts/2015-08-Understanding-LSTMs/

A Neural Network in 11 lines of Python
by iamtrask
http://iamtrask.github.io/2015/07/12/basic-python-network/

Sign up for Data Elixir mailing list: http://dataelixir.com/

Books

- Deep Learning (Goodwill, Bengio, Courville), MIT Press, 2016
- Theoretical Neuroscience (Abbott & Dayan), MIT Press, 2005

BIIT

TARTU ÜLIKOOL · UNIVERSITAS TARTUENSIS 1632

COMPUTATIONAL NEUROSCIENCE
Research group at the University of Tartu

dmytrofishman@gmail.com