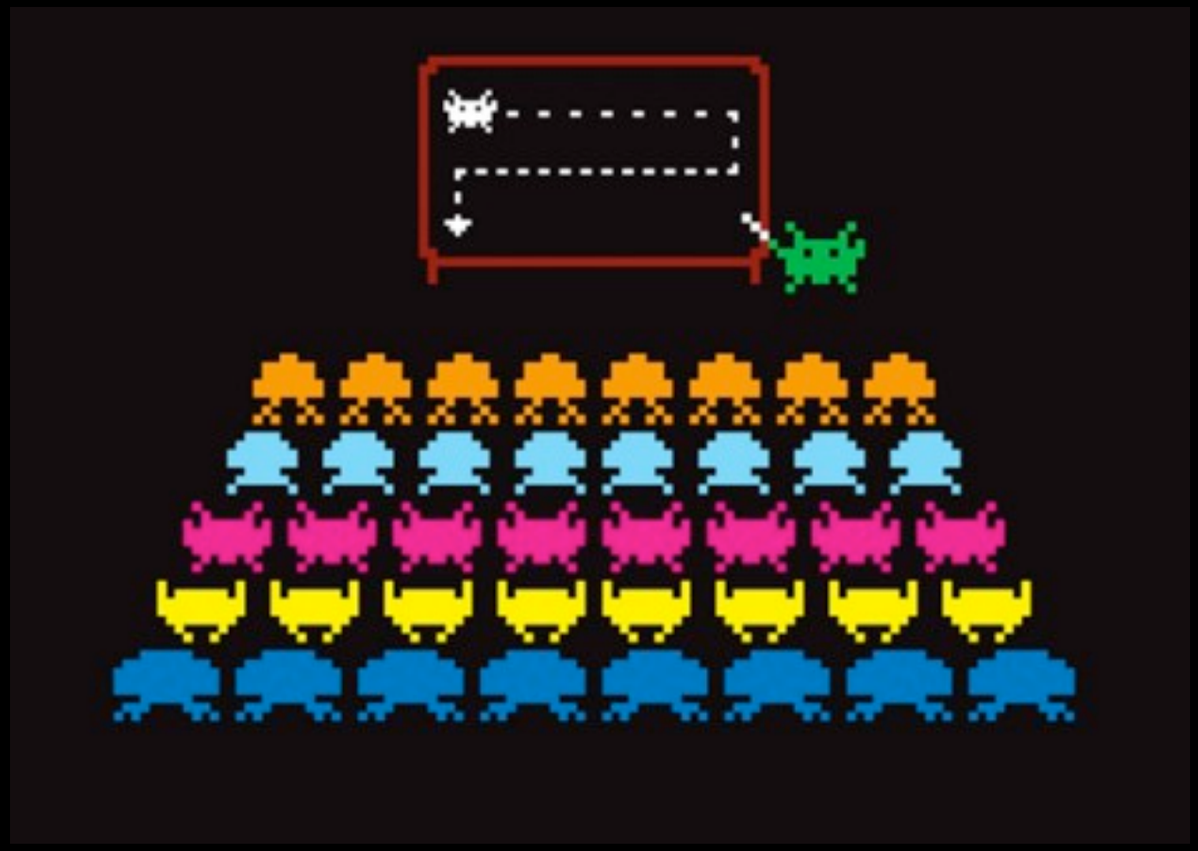
Deep Learning



Raul Vicente

Institute of Computer Science, University of Tartu

There is a revolution in AI & machine learning

Deep Learning

MIT Technology Review

10 BREAKTHROUGH TECHNOLOGIES 2013

Deep Learning

With massive amounts of computational power, machines can now recognize objects and translate speech in real time. Artificial intelligence is finally getting smart.

Is "Deep Learning" A Revolution in Artificial Intelligence?

New Yorker 11/2012

Facebook taps 'Deep Learning' Giant for new AI Lab

Wired 12/2013

Why Did Google Pay $400 Million for DeepMind

MIT Technology Reviews 1/2014

New Techniques from Google Are Taking Artificial Intelligence to Another Level

MIT Technology Reviews 5/2013

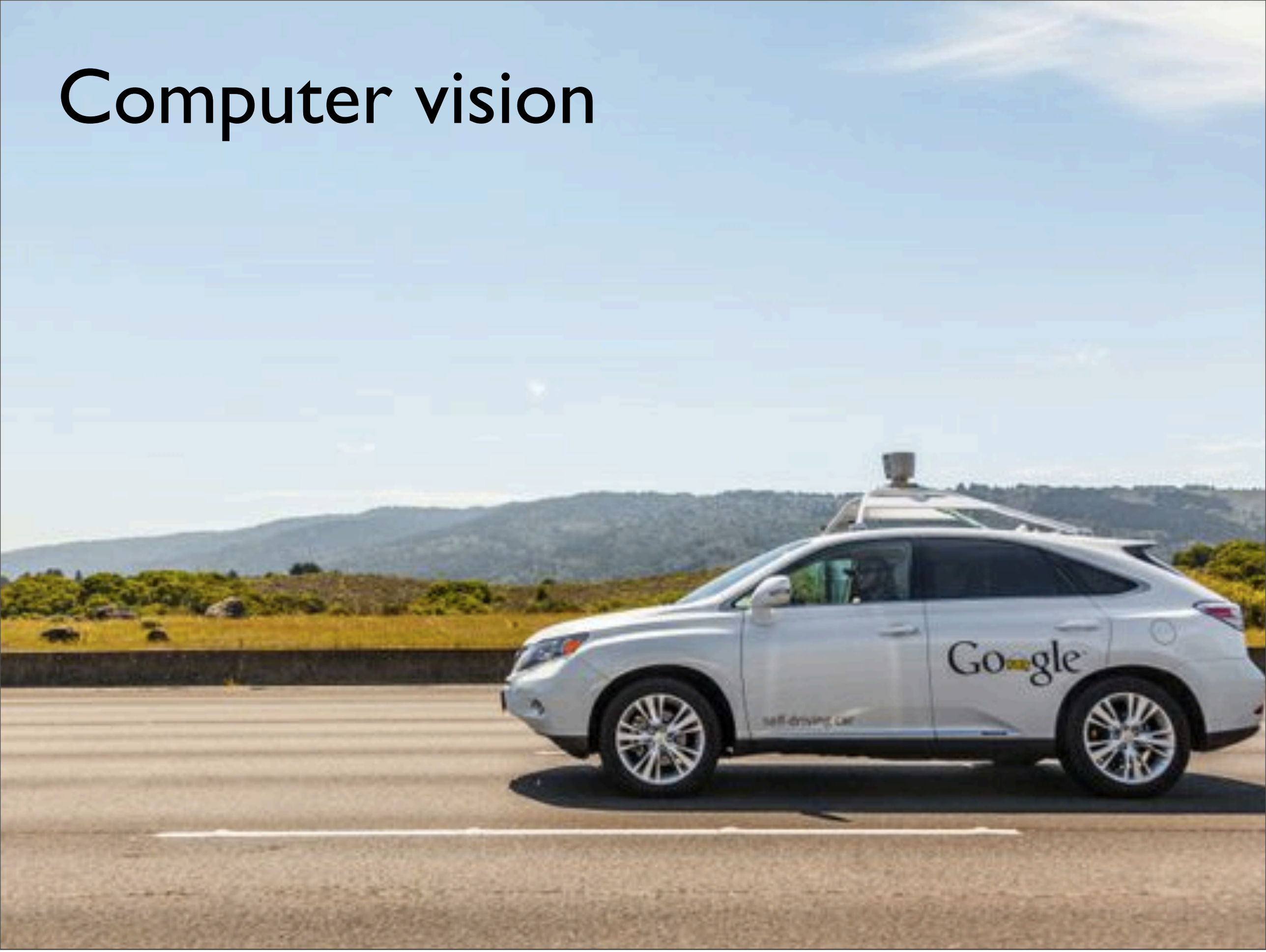
Google Hires Brains that Helped Supercharge Machine Learning

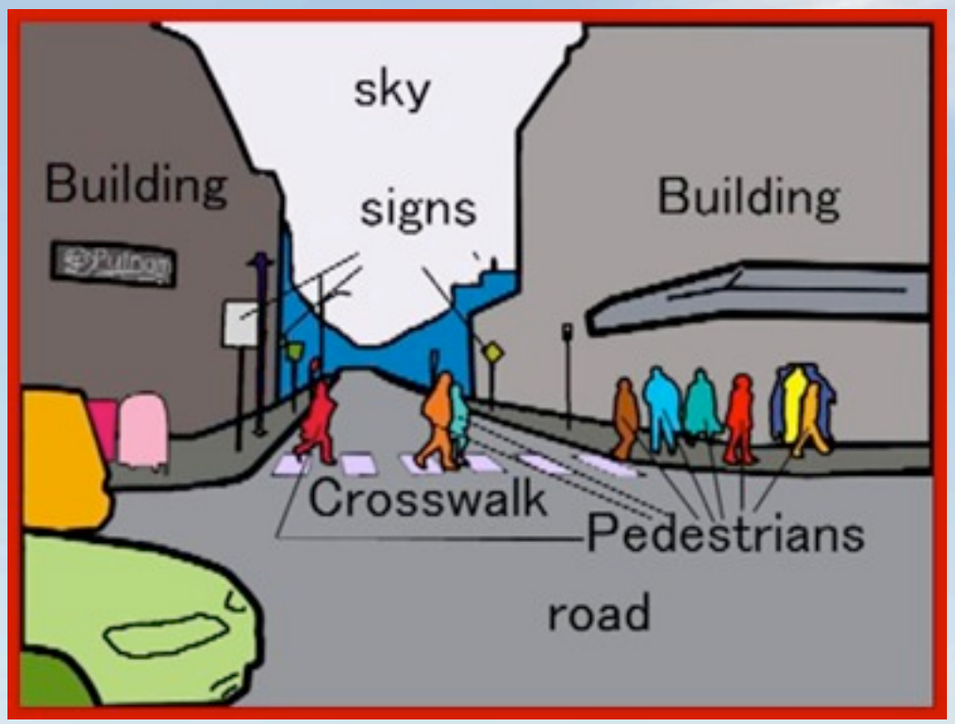
Wired 3/2013

Real-time speech translation

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

Computer vision




sky
Building
signs
Building
Crosswalk
Pedestrians
road

Google

You see this

But the camera see this

194	210	201	212	199	213	215	195	178	158	182	209
180	189	190	221	209	205	191	167	147	115	129	163
114	126	140	188	176	165	152	140	170	106	78	88
87	103	115	154	143	142	149	153	173	101	57	57
102	112	106	131	122	138	152	147	128	84	58	66
94	95	79	104	105	124	129	113	107	87	69	67
68	71	69	98	89	92	98	95	89	88	76	67
41	56	68	99	63	45	60	82	58	76	75	65
20	43	69	75	56	41	51	73	55	70	63	44
50	50	57	69	75	75	73	74	53	68	59	37
72	59	53	66	84	92	84	74	57	72	63	42
67	61	58	65	75	78	76	73	59	75	69	50

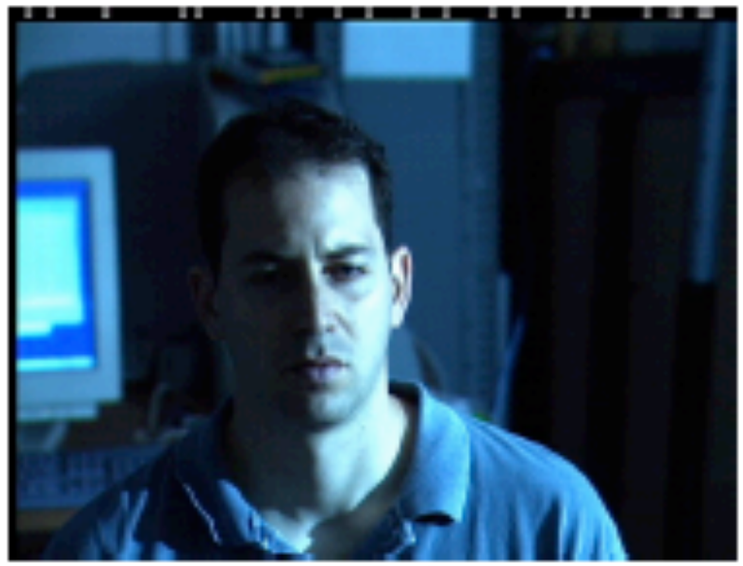
1.22

1.04

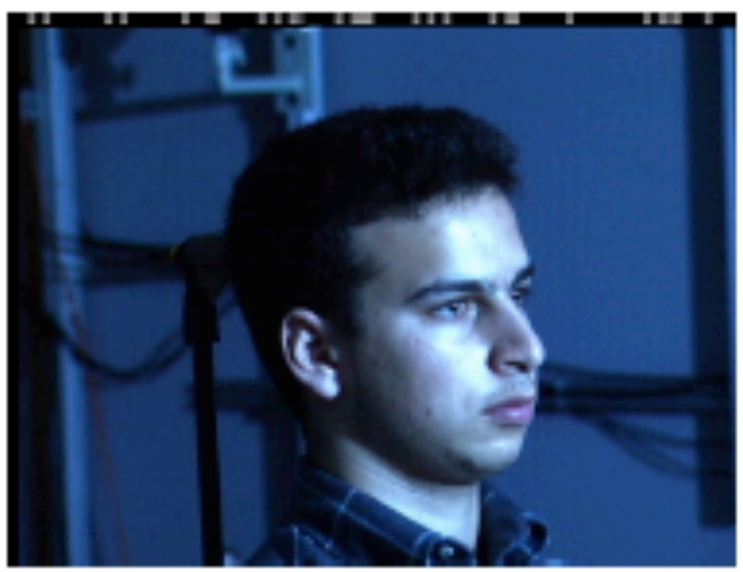
1.33

1.33

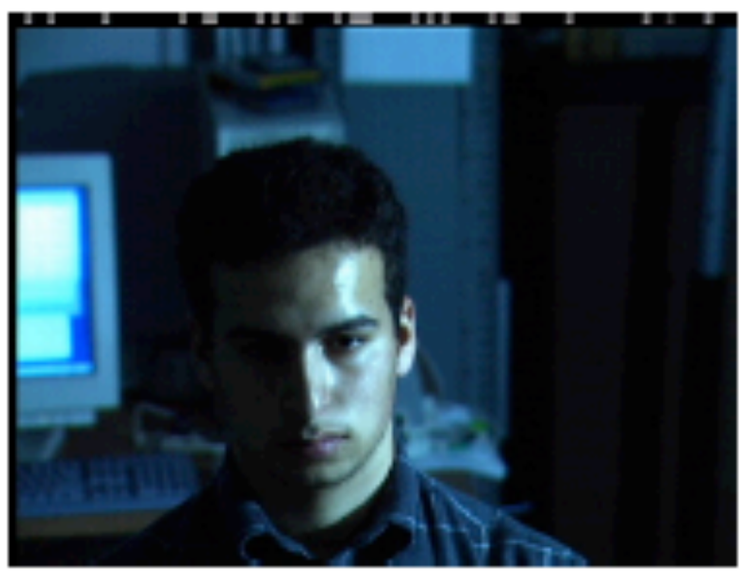
0.78



1.26



0.99




ВЫХОД В ГОРОД
Russian
English
ACCESS TO CITY

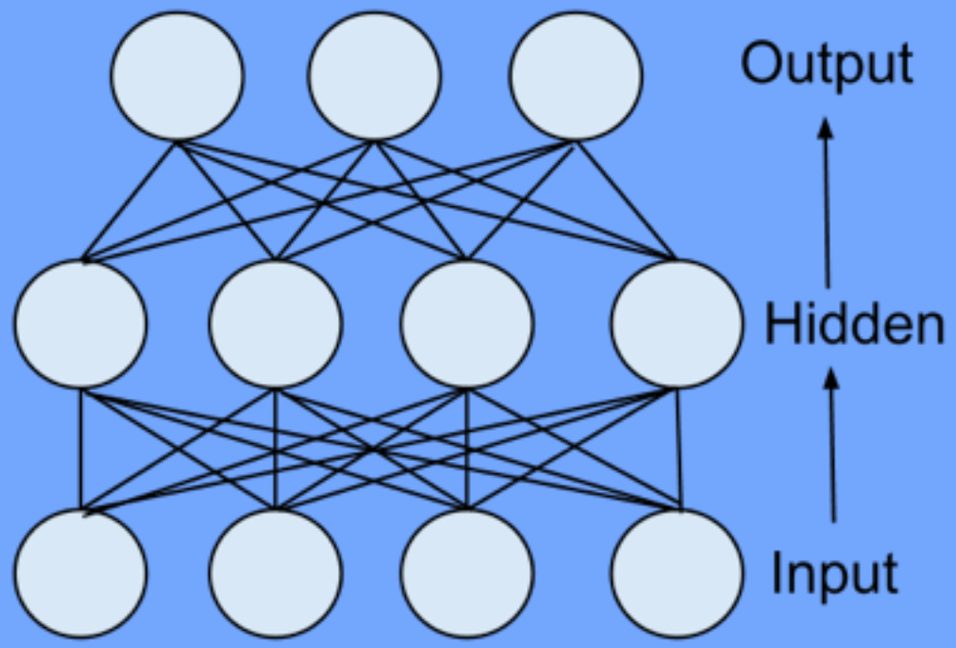
a man riding a horse drawn carriage down a street

a boat is docked in a canal with a large building in the background

a young boy is holding a baseball bat

2 themes

Deep Learning



DeepMind

Machine Learning: algorithms that learn from data

- Supervised

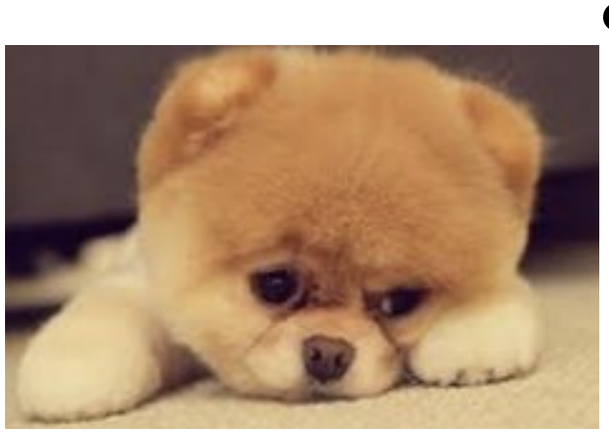
{“cat”, “dog”}

Machine Learning: algorithms that learn from data

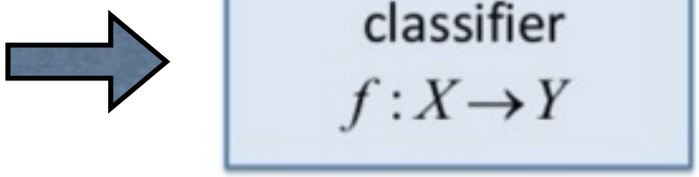
- Supervised

Machine Learning: algorithms that learn from data

- Supervised



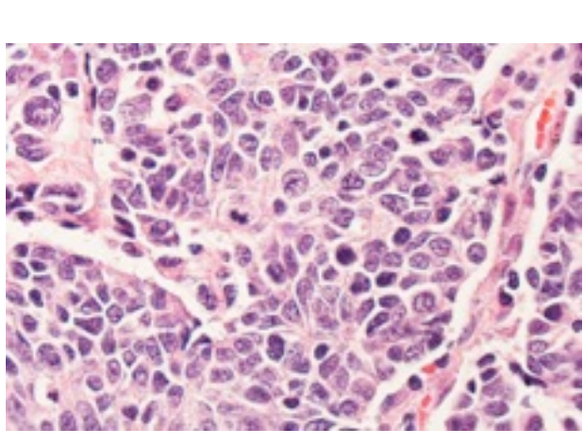
?



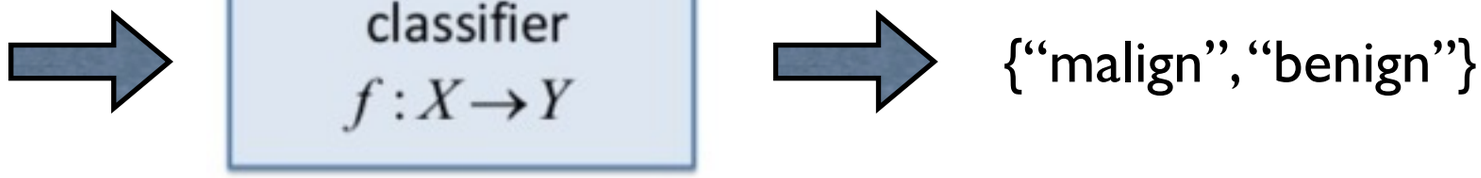
“dog”

Machine Learning: algorithms that learn from data

- Supervised



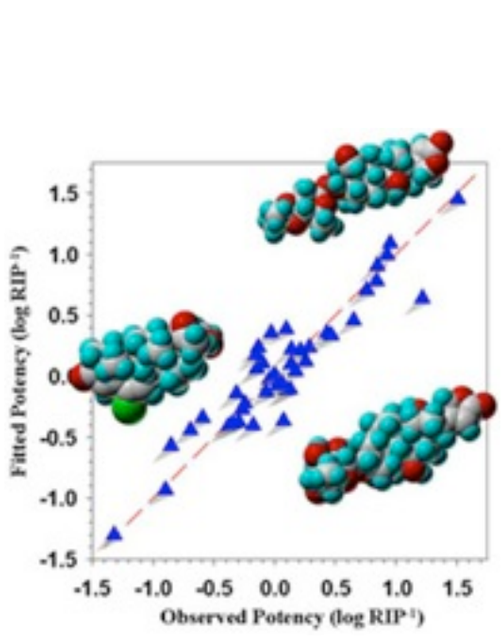
?



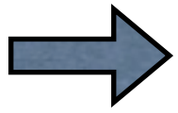
{“malign”,“benign”}

Machine Learning: algorithms that learn from data

- Supervised

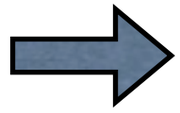


?



classifier

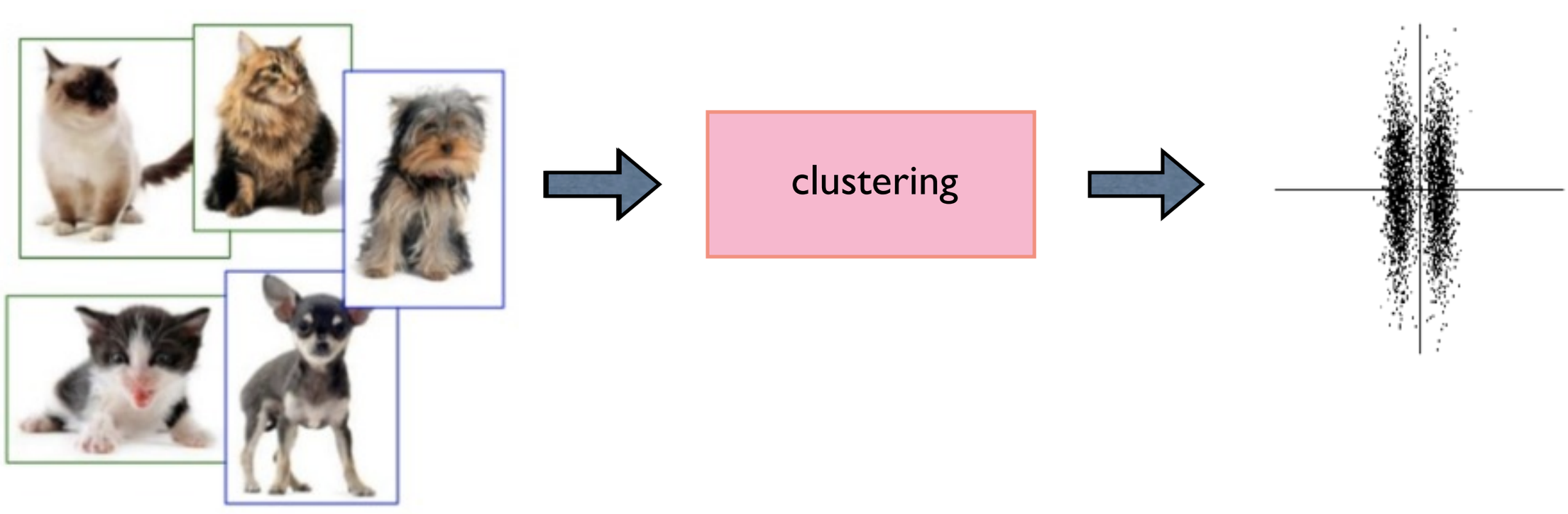
$f : X \rightarrow Y$



{“toxic”, “non-toxic”}

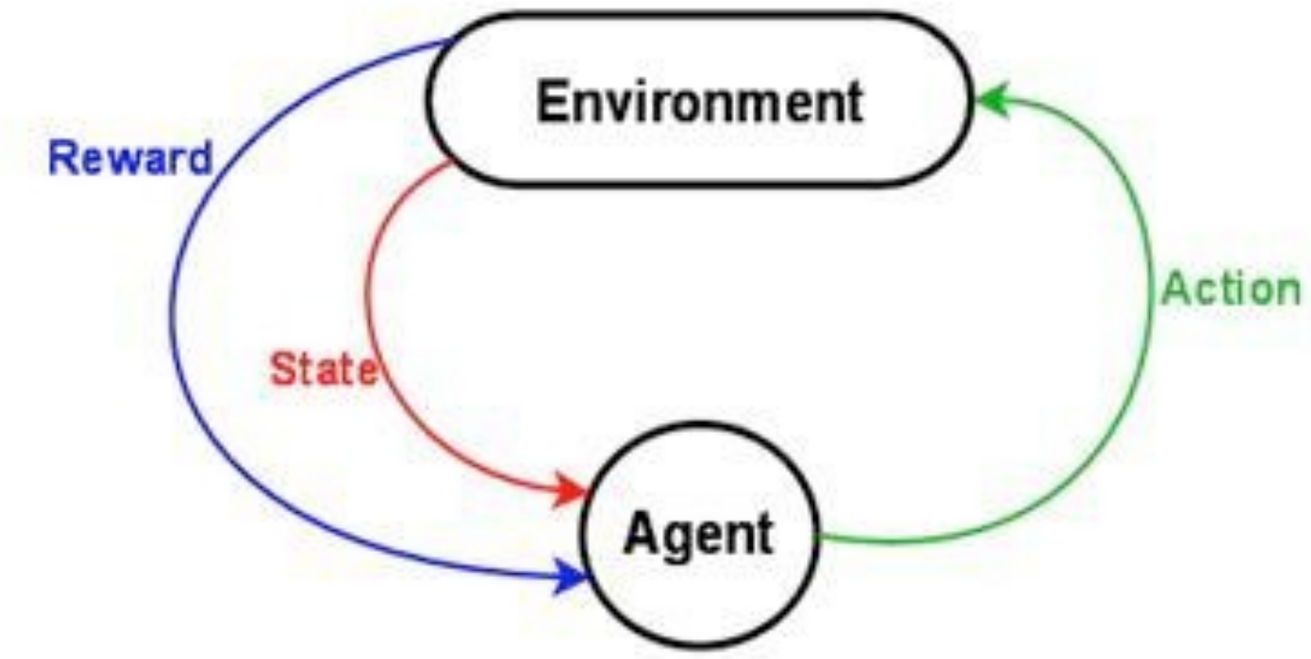
Machine Learning: algorithms that learn from data

- Supervised
- Unsupervised



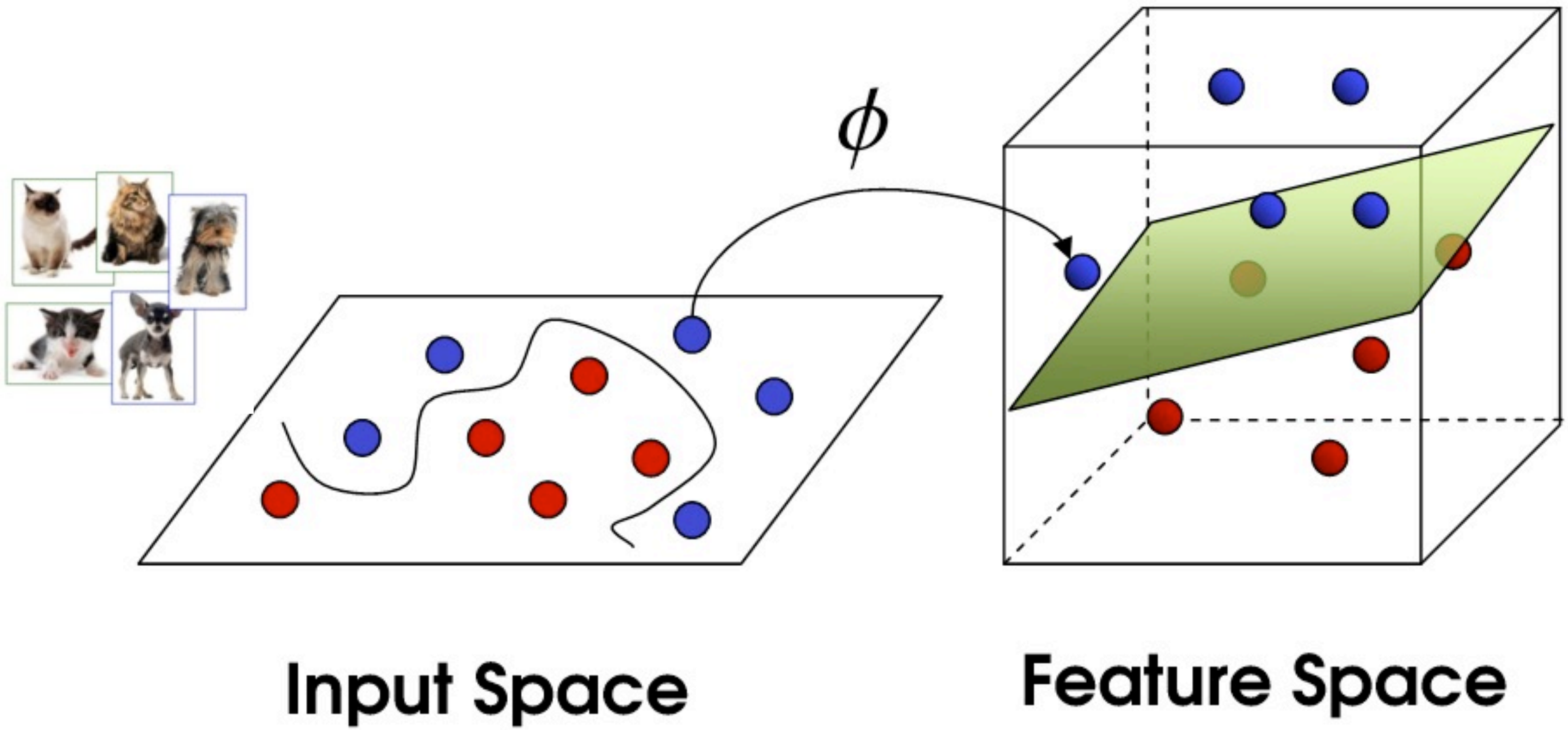
Machine Learning: algorithms that learn from data

- Supervised
- Unsupervised
- Reinforcement

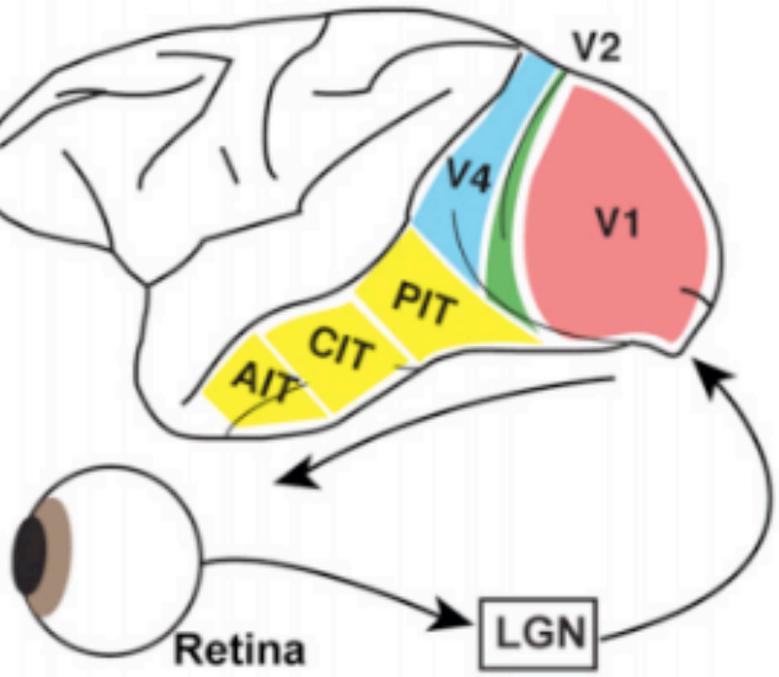


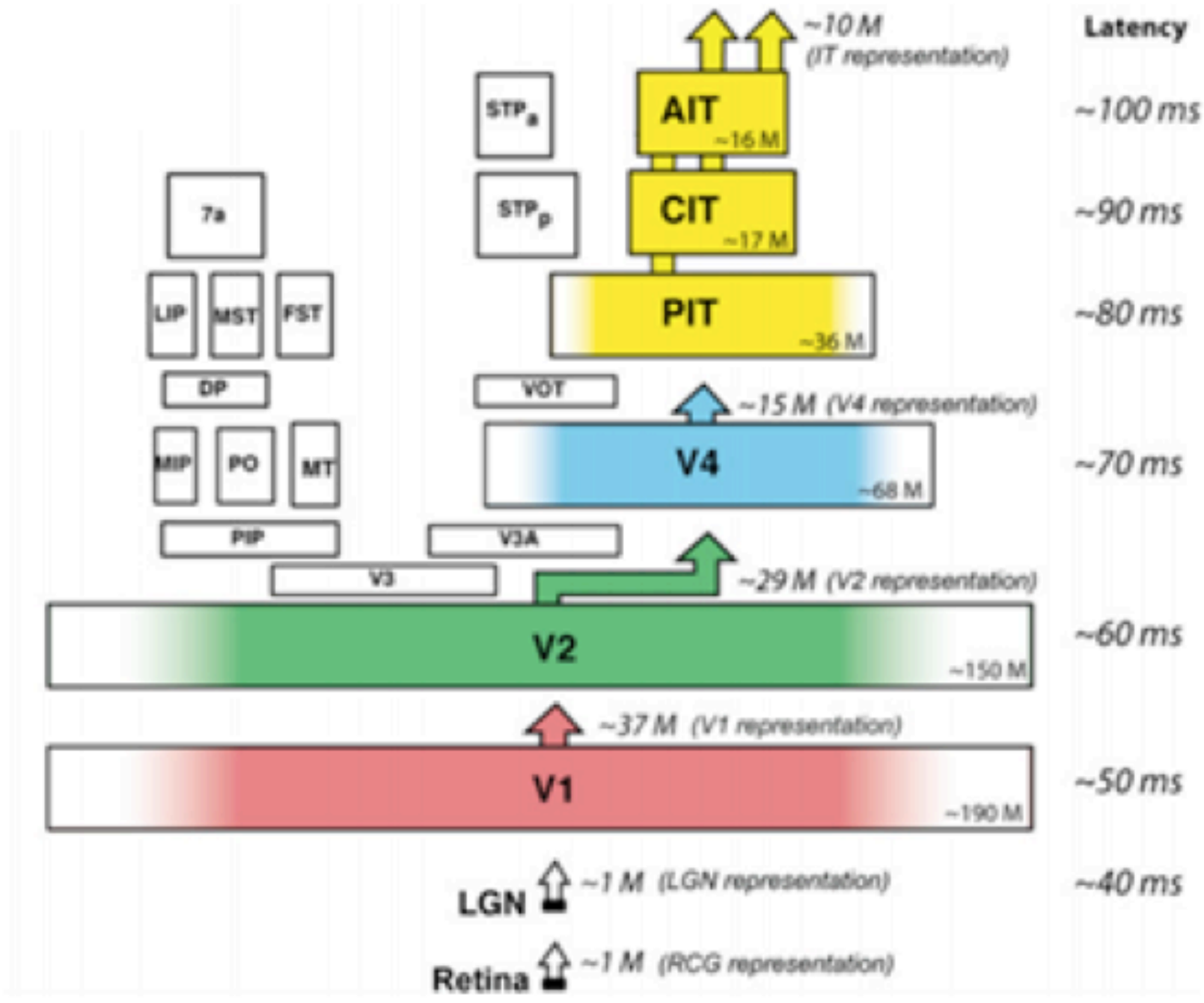
Shallow Machine Learning

0 or 1 abstraction layer (feature transformation)



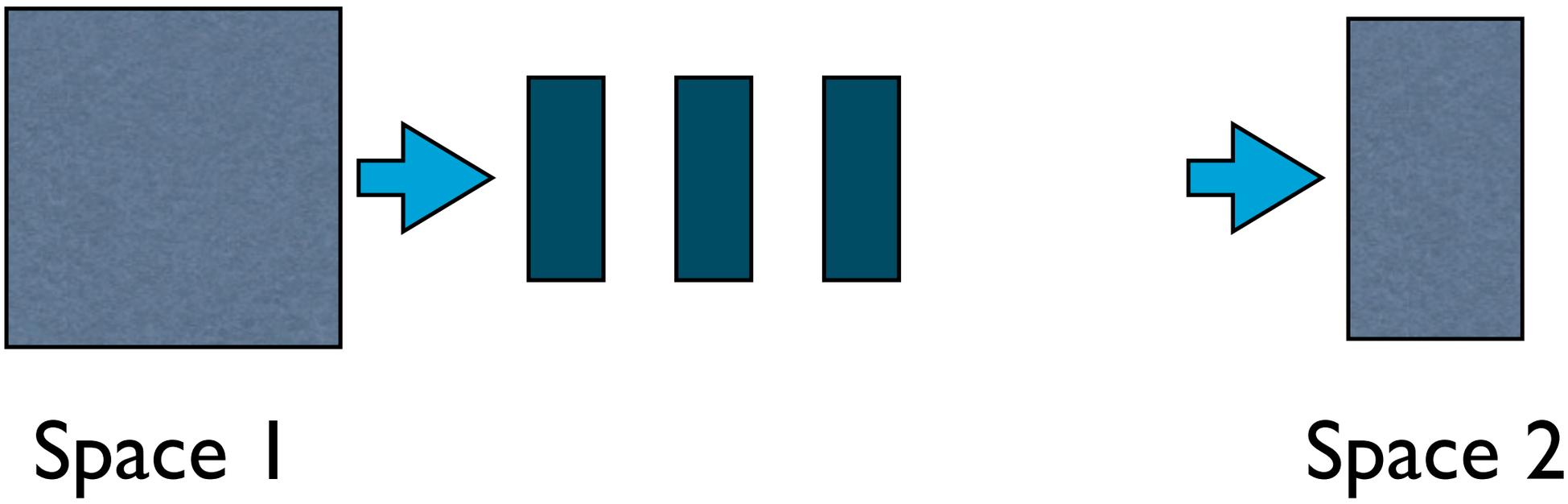
Deep Human Learning



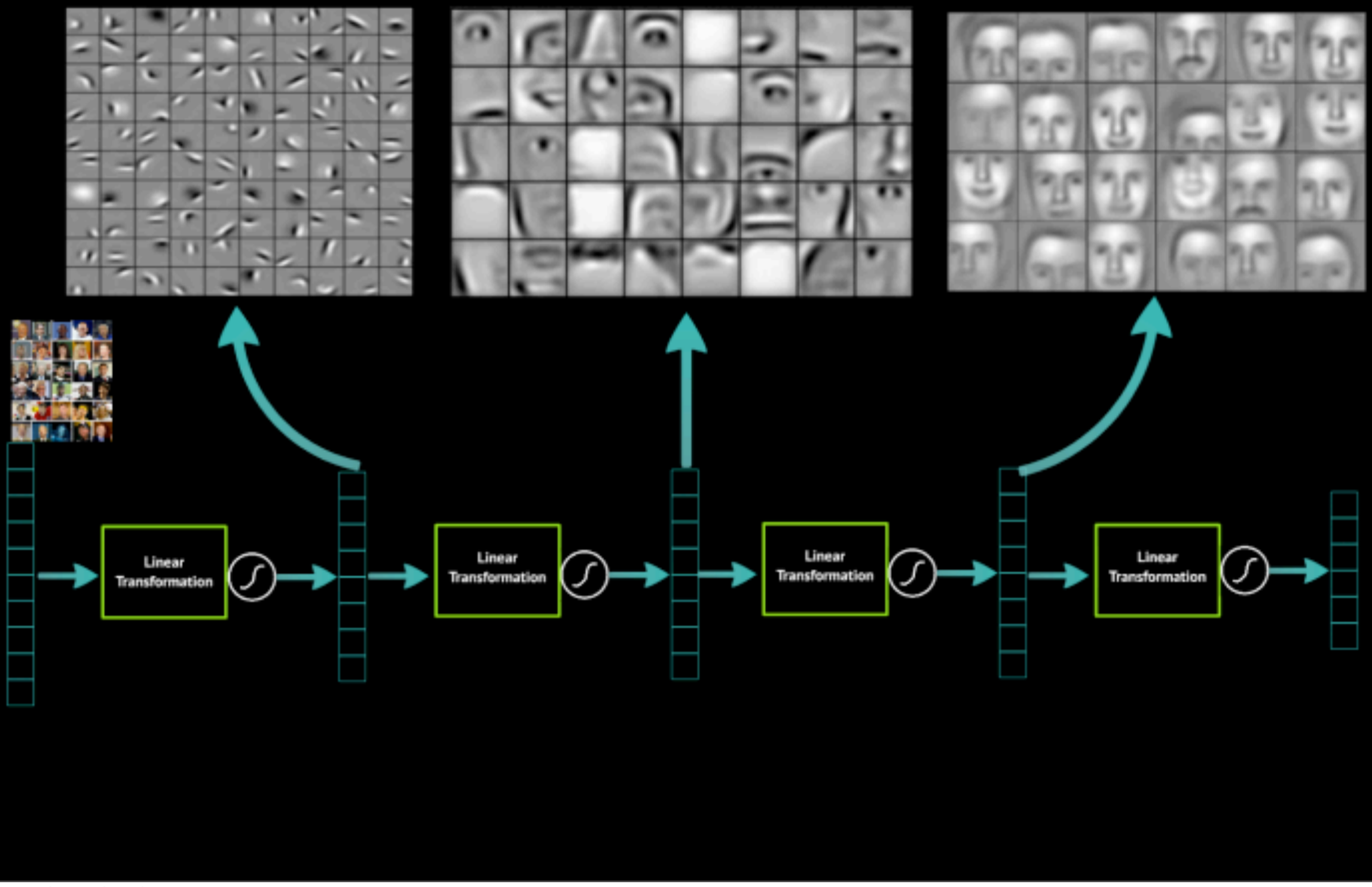


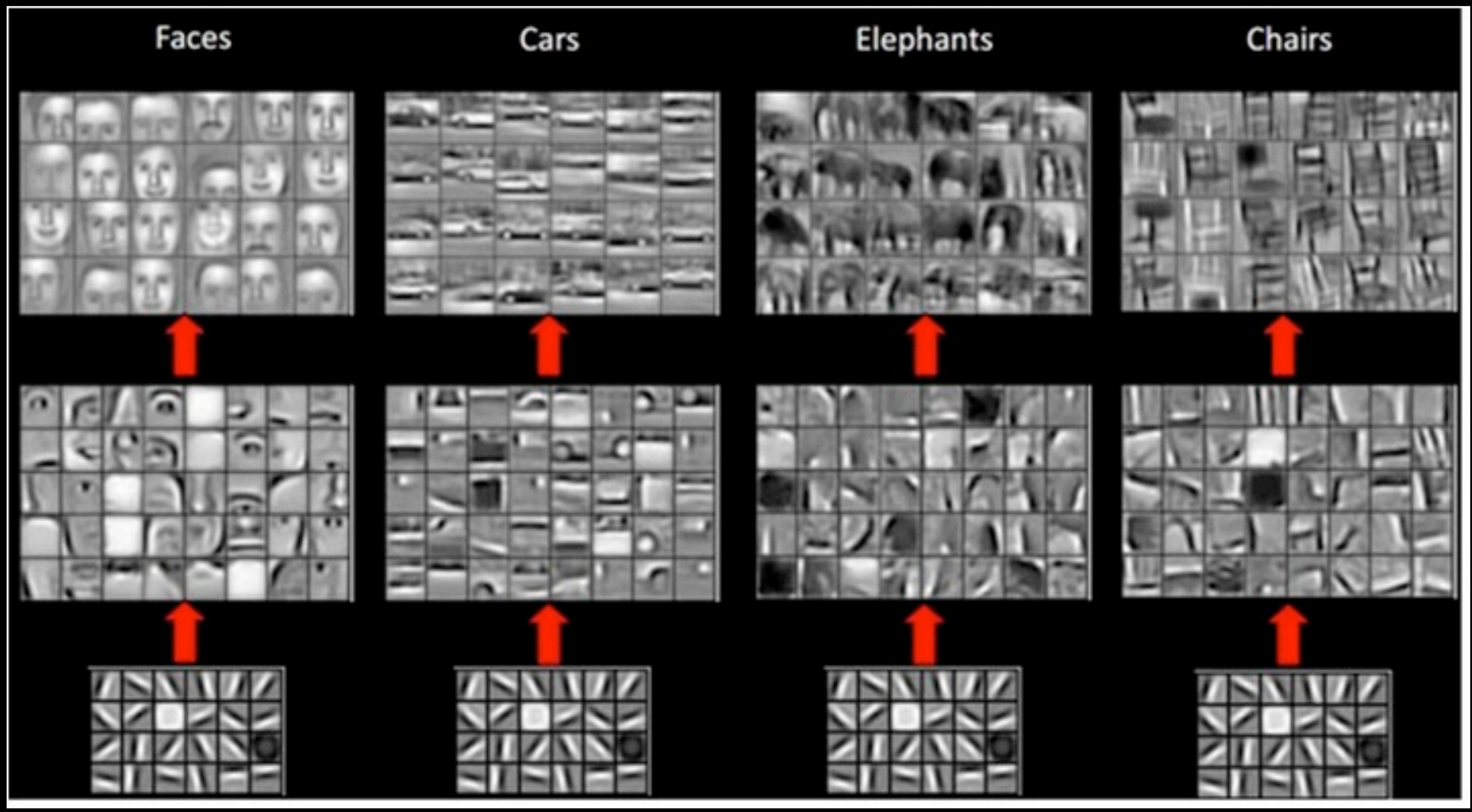
What is deep learning?

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



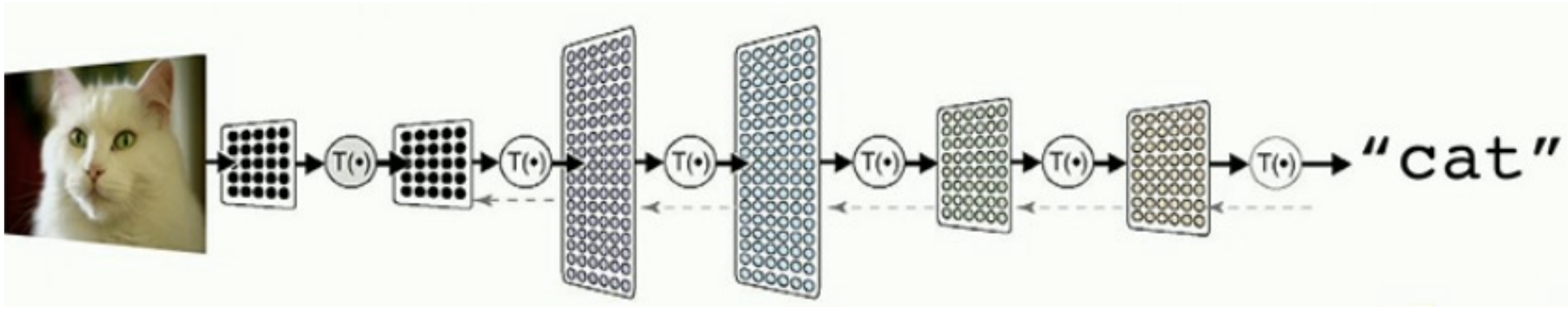
Deep Learning learns layers of features




Faces
Cars
Elephants
Chairs

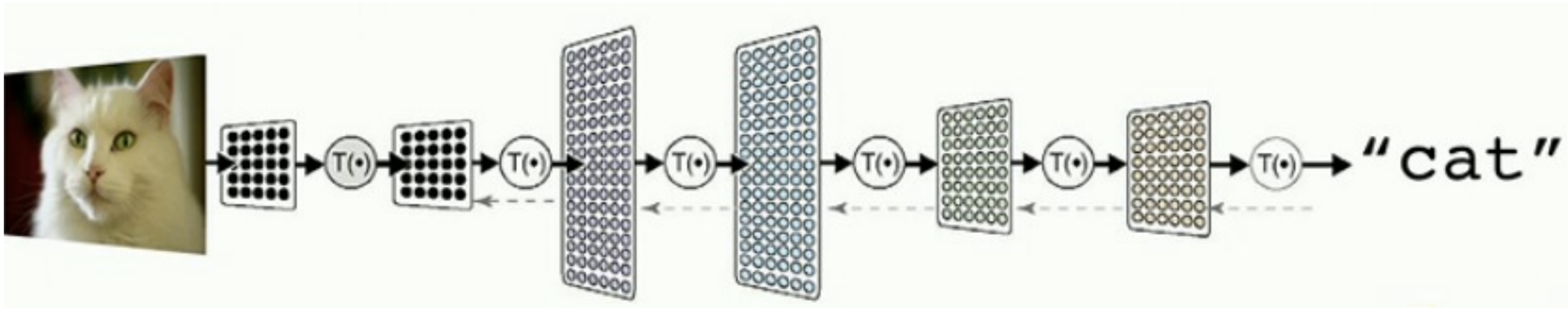
What is deep learning?

DL = Artificial Neural Networks with many layers



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

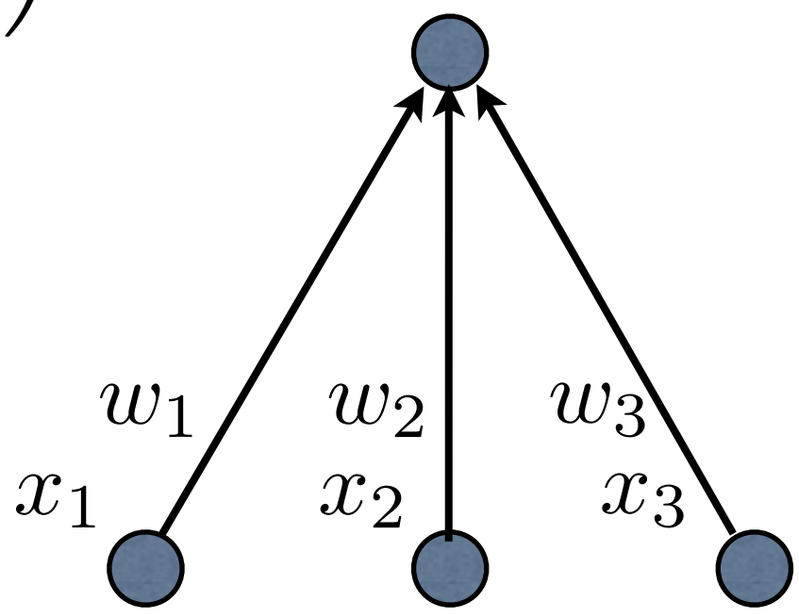


The neuron

- Different weights compute different functions

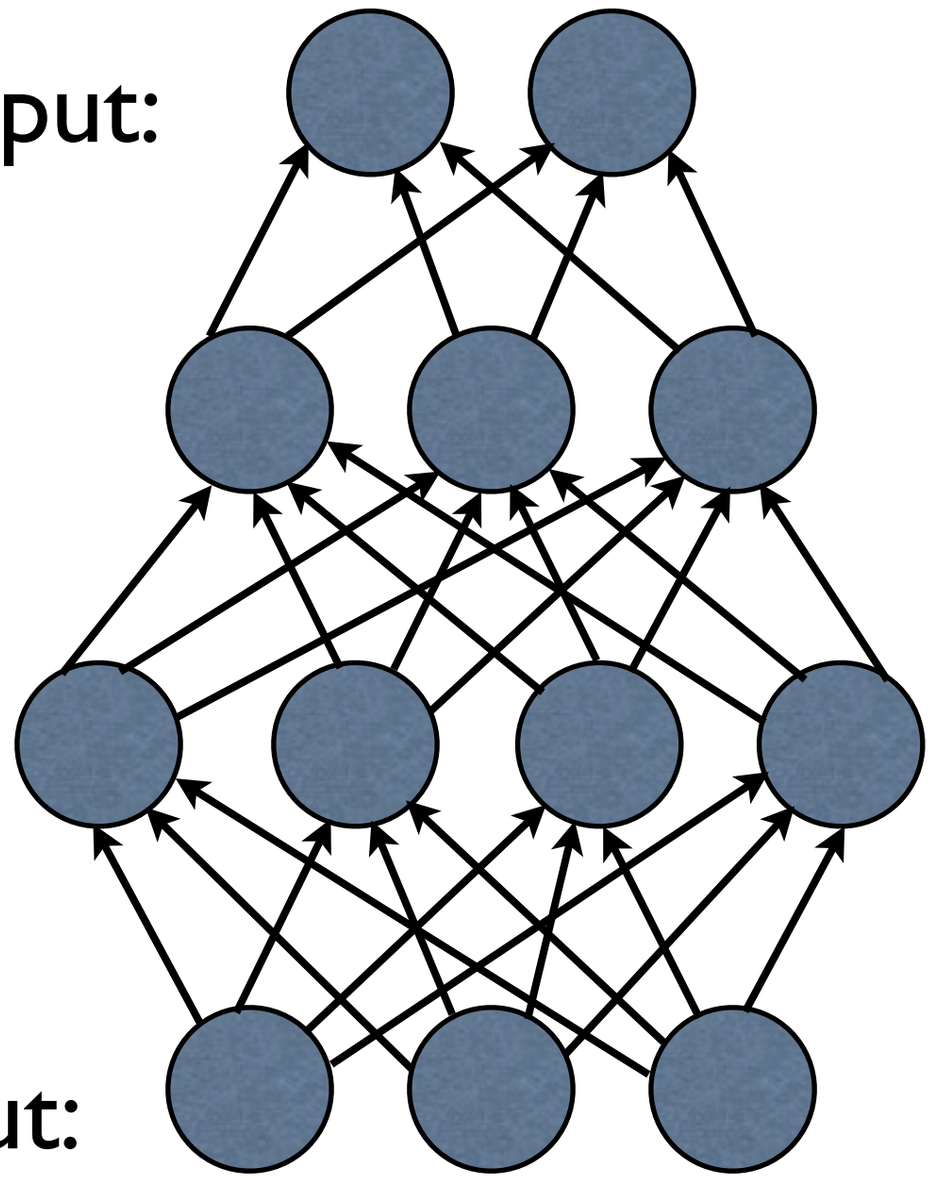
$$y_i = F\left(\sum_i w_i x_i\right)$$

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



Neural networks

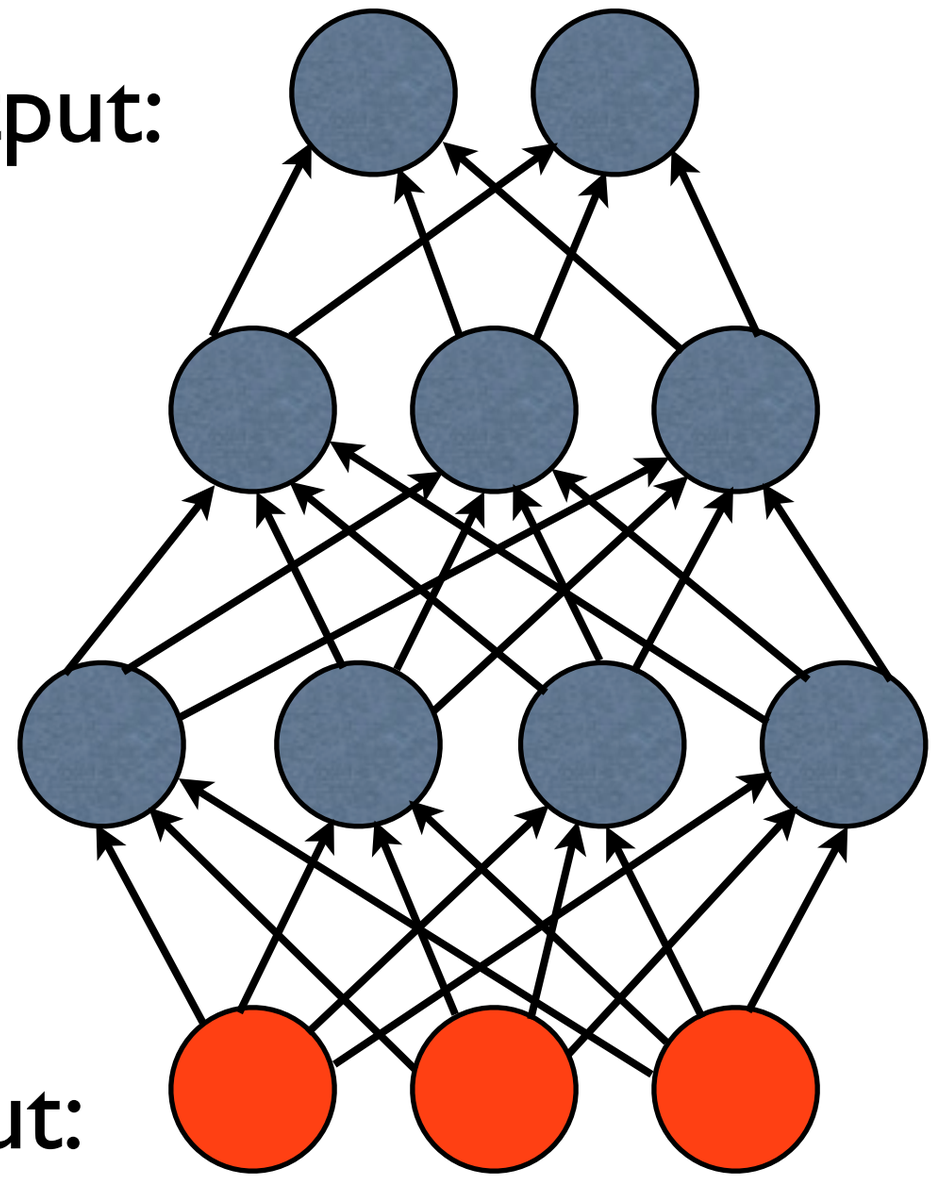
Output:



Input:

Neural networks

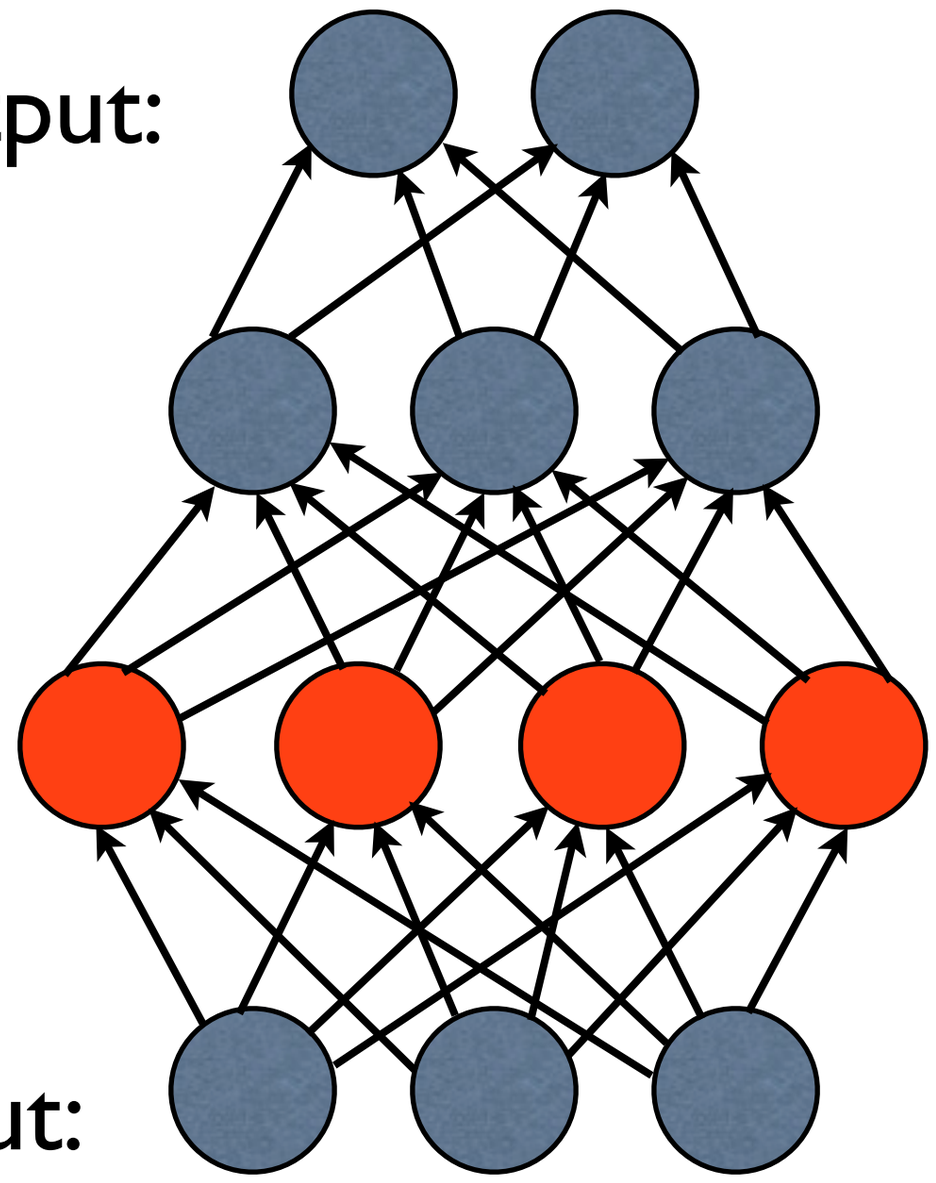
Output:



Input:

Neural networks

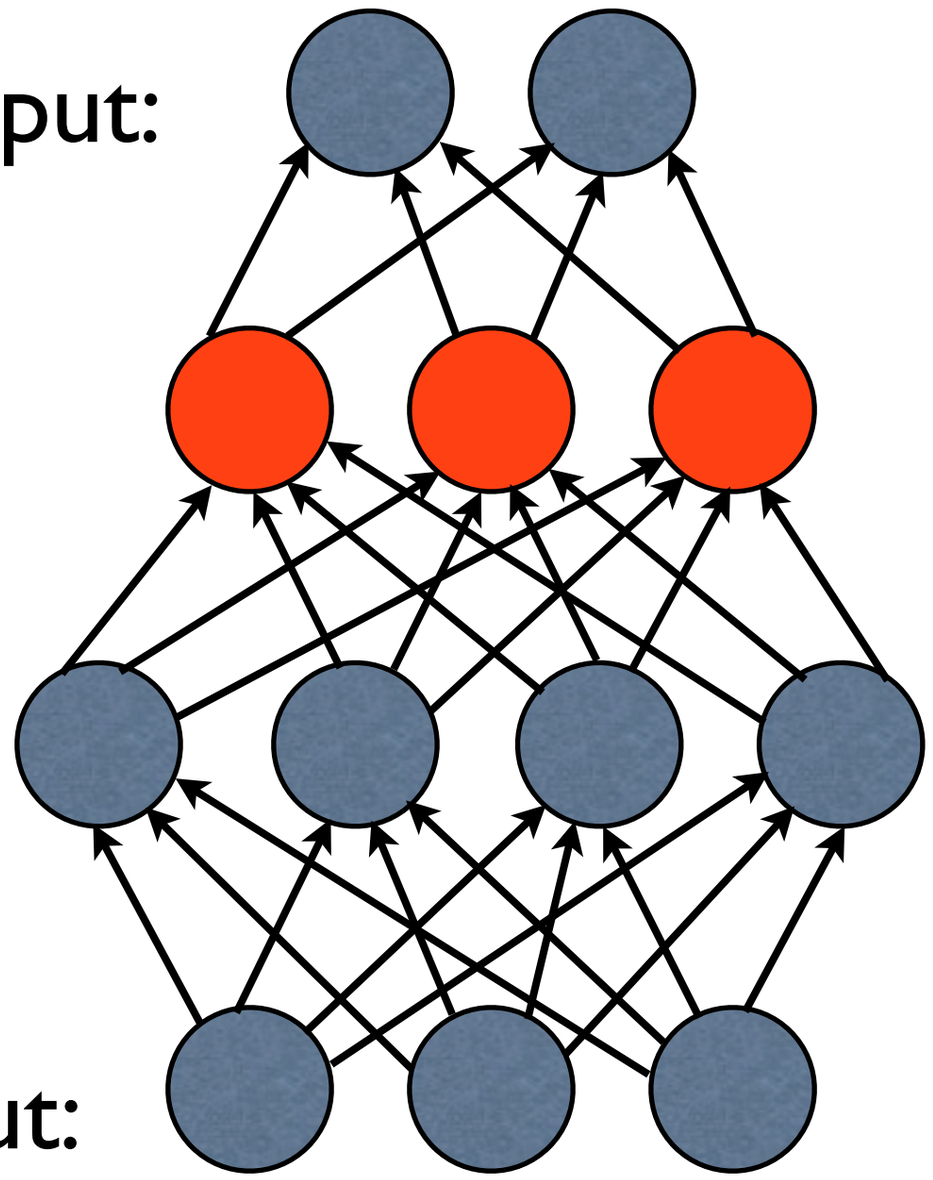
Output:



Input:

Neural networks

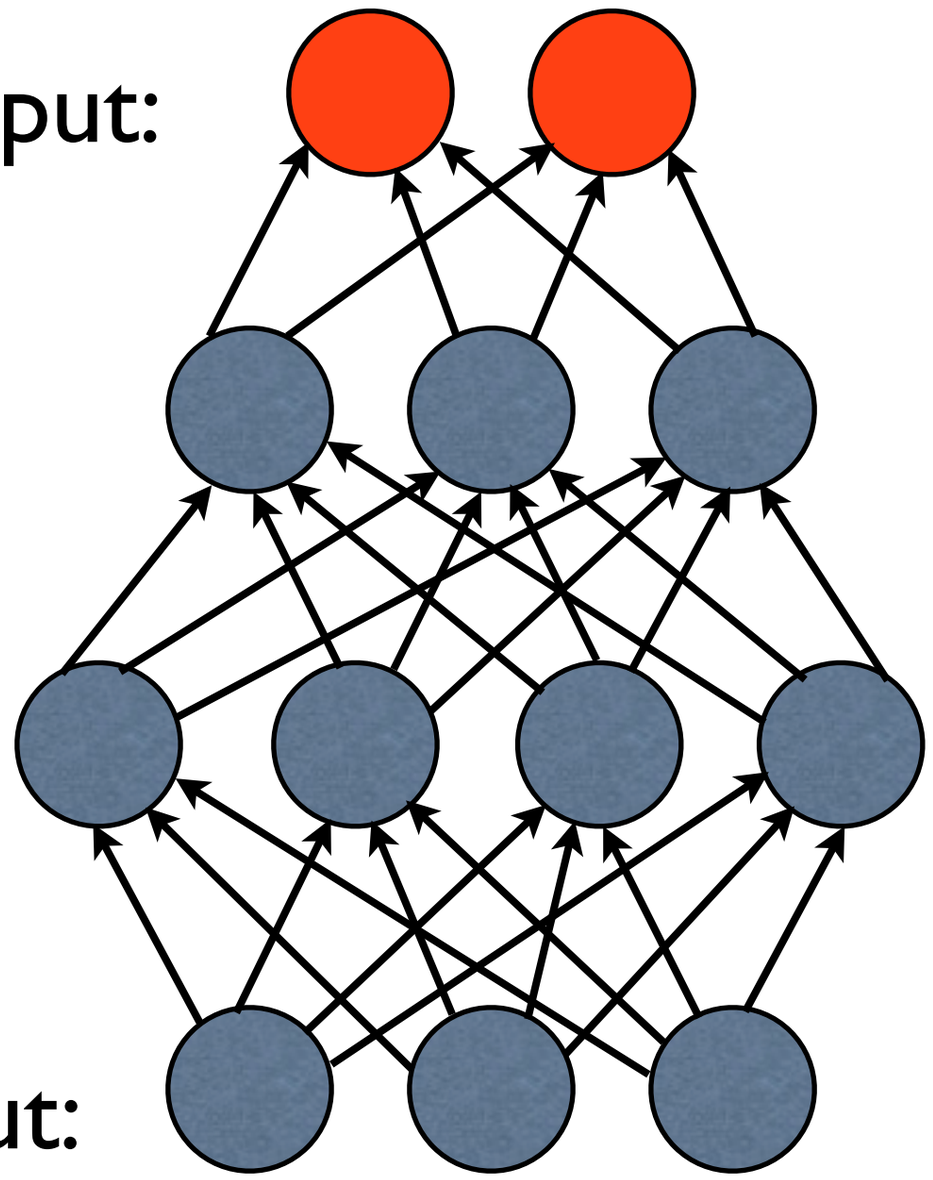
Output:



Input:

Neural networks

Output:



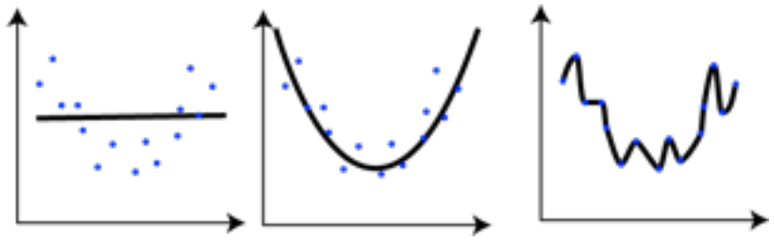
Input:

Learning algorithm

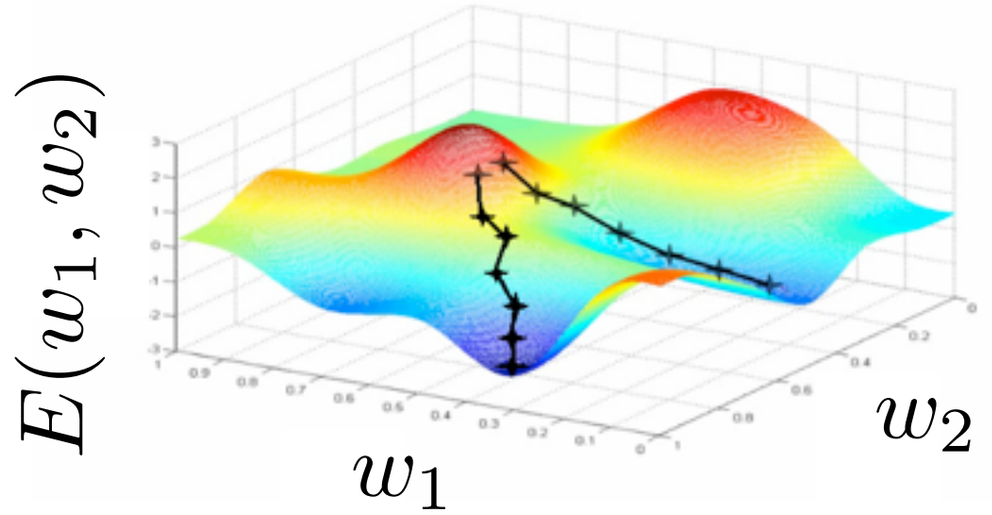
- **while** not done
    - pick a random training case (**x**, **y**)
    - run neuronal network on input **x**
    - modify connection weights to make prediction closer to **y**

Very difficult to train

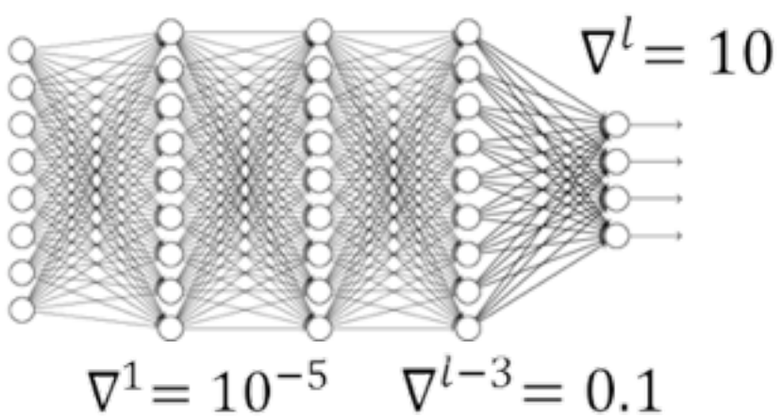
Overfitting



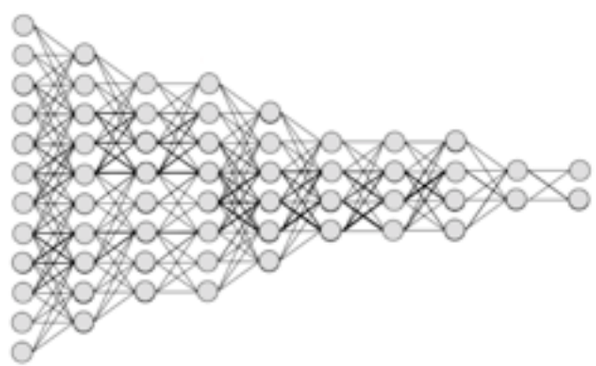
Complex landscape



Vanishing gradients



Dimensionality

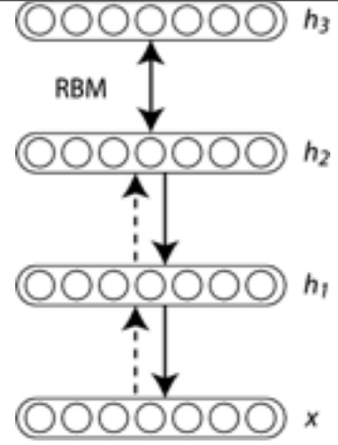



KEEP
CALM
AND
STUDY
ALGORITHMS

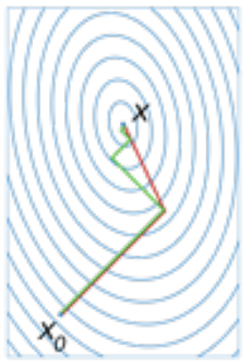

WE ♥
DATA


TITAN Z

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

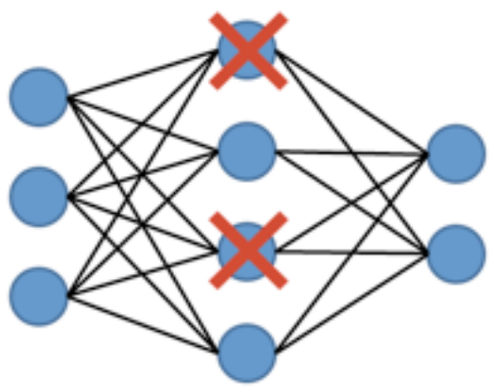


- Efficient descent algorithms
  (complex landscape)

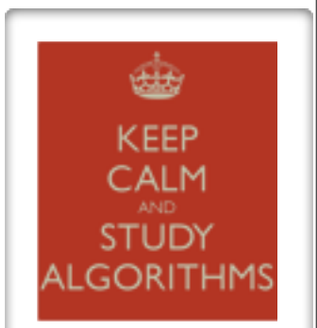


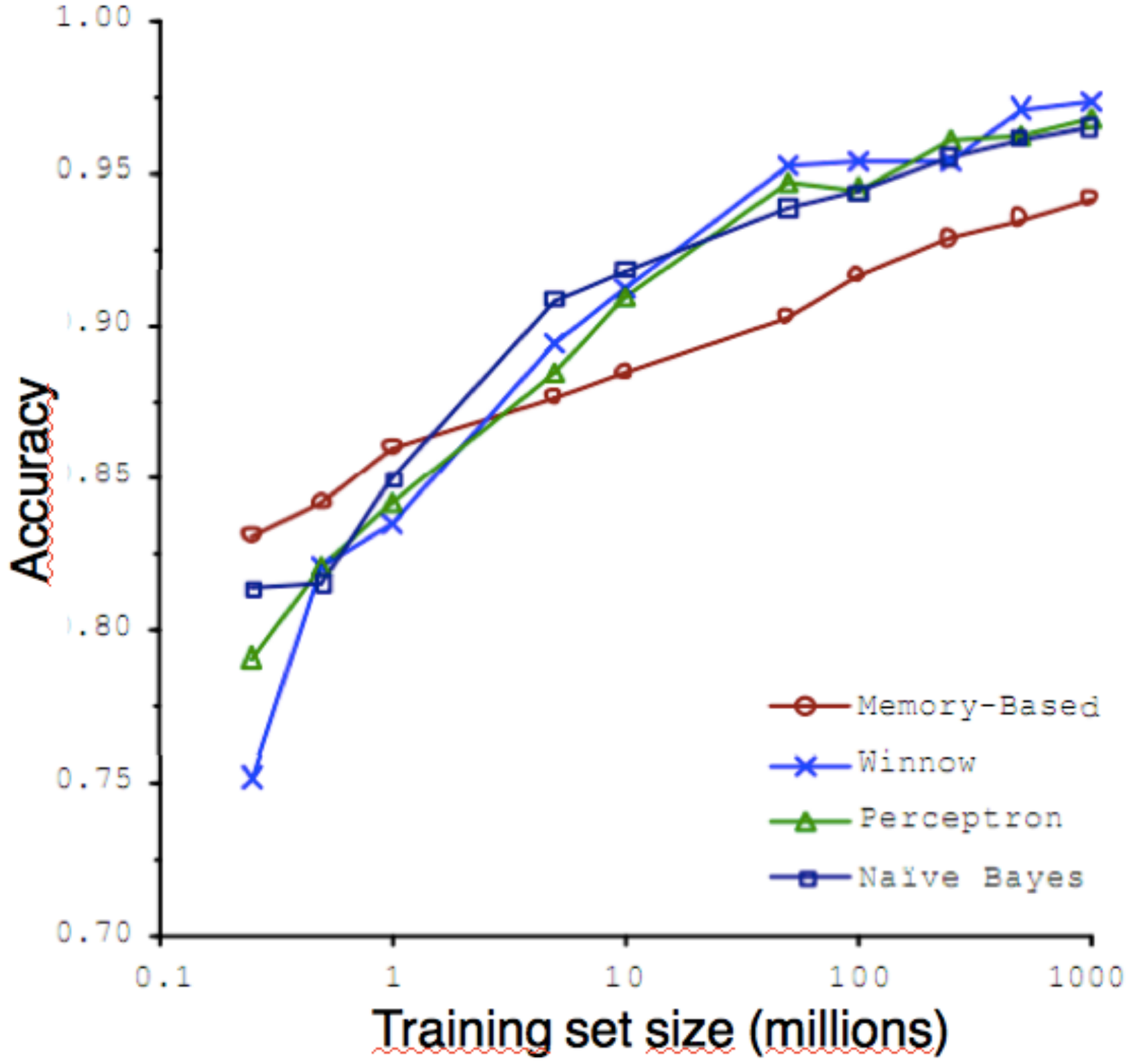
- Activation
  (vanishing gradient)

- Dropout
  (overfitting)



- Domain Prior Knowledge



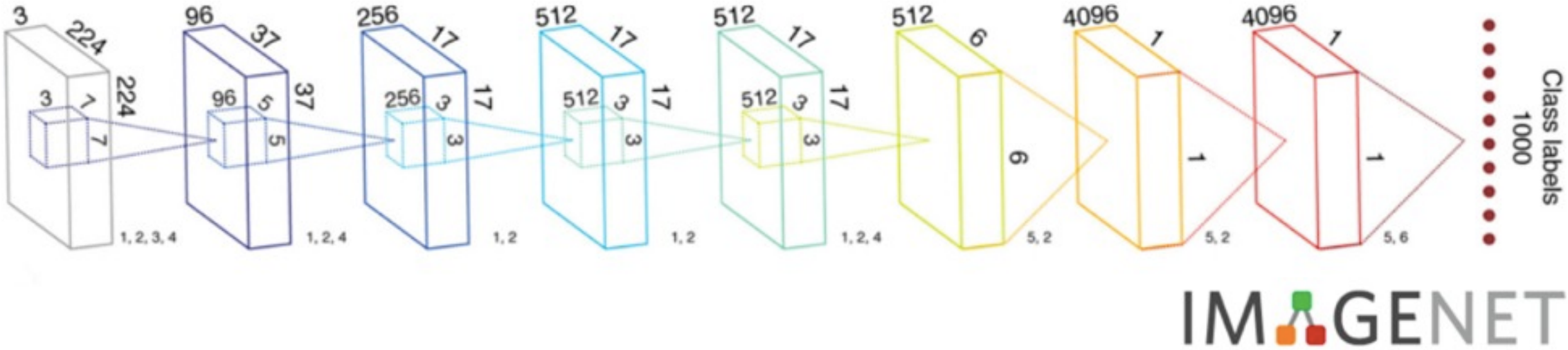

1.00
0.95
.90
.85
.80
0.75
0.70
Accuracy
0.1
1
10
100
1000
Training set size (millions)
Memory-Based
Winnow
Perceptron
Naïve Bayes

Now that we are deep...

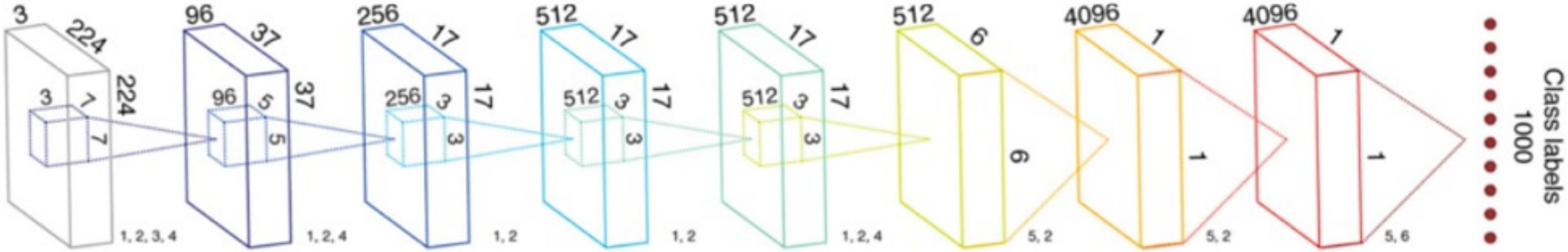
- Instead of hand-crafted features, let the algorithm build the relevant features for your problem

- More representational power

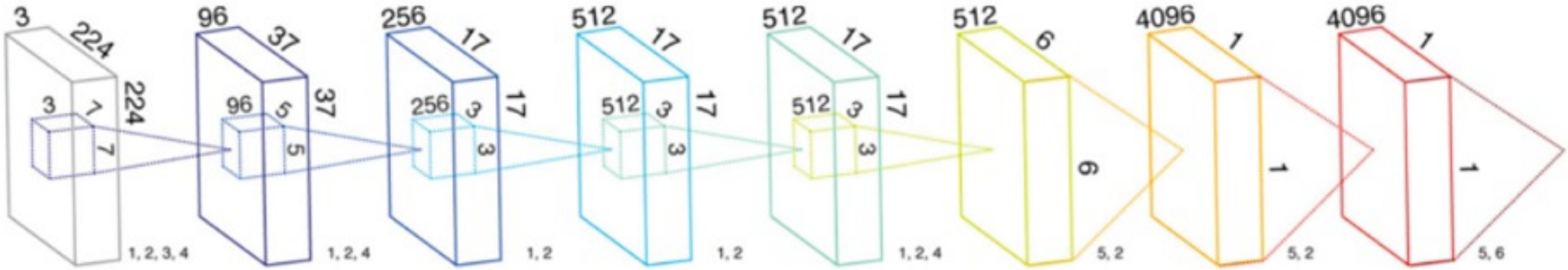
- Powerful function approximator

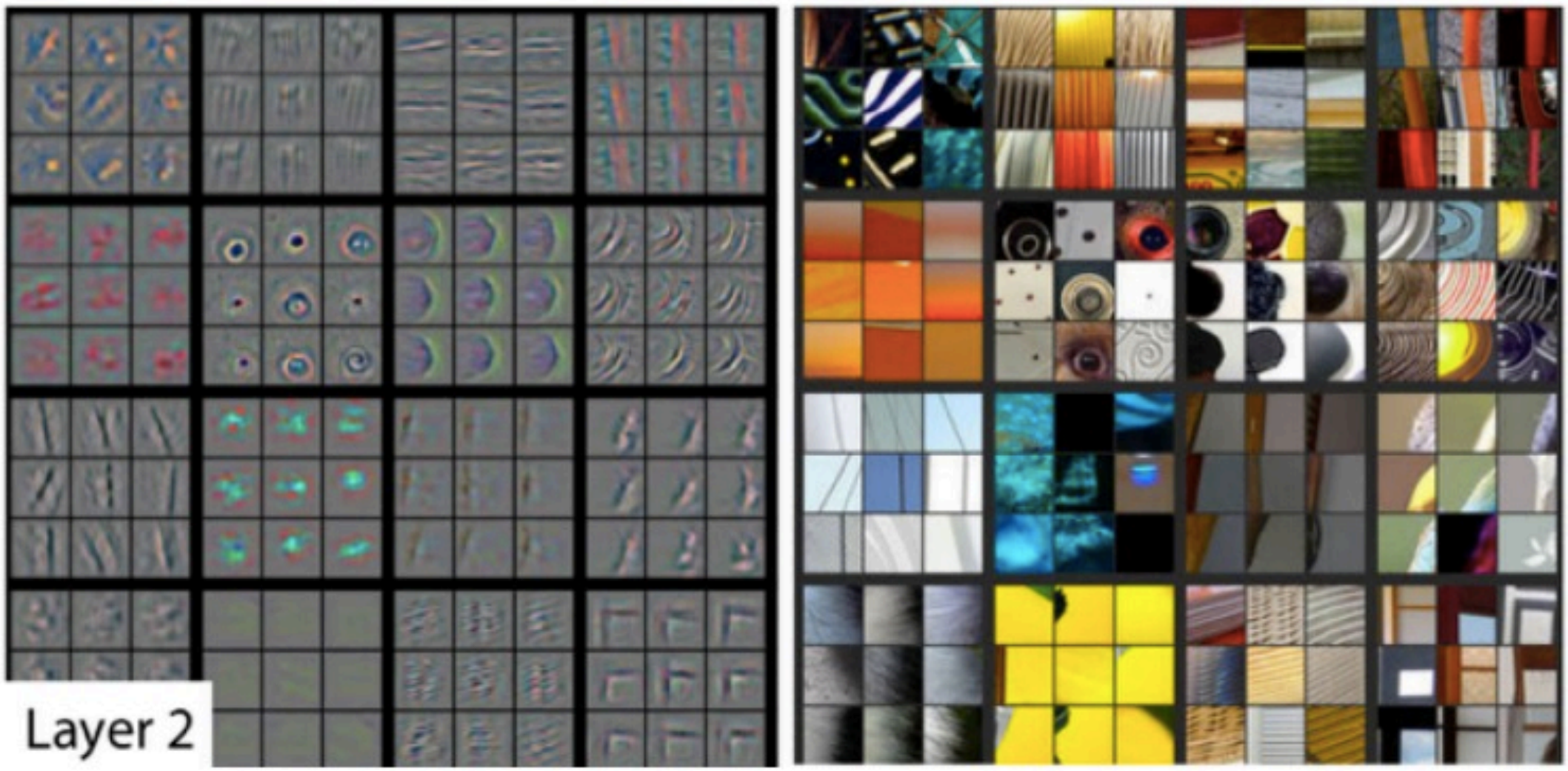

3
224
224
3
7
7
1, 2, 3, 4
96
37
37
96
5
5
1, 2, 4
256
17
17
256
3
3
1, 2
512
17
17
512
3
3
1, 2
512
17
17
512
3
3
1, 2, 4
512
6
6
5, 2
4096
1
1
5, 2
4096
1
1
5, 6
Class labels
1000

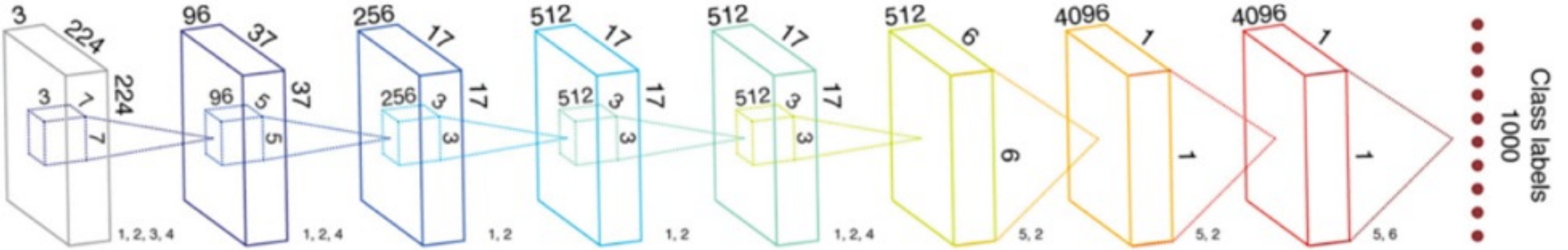
IMAGENET

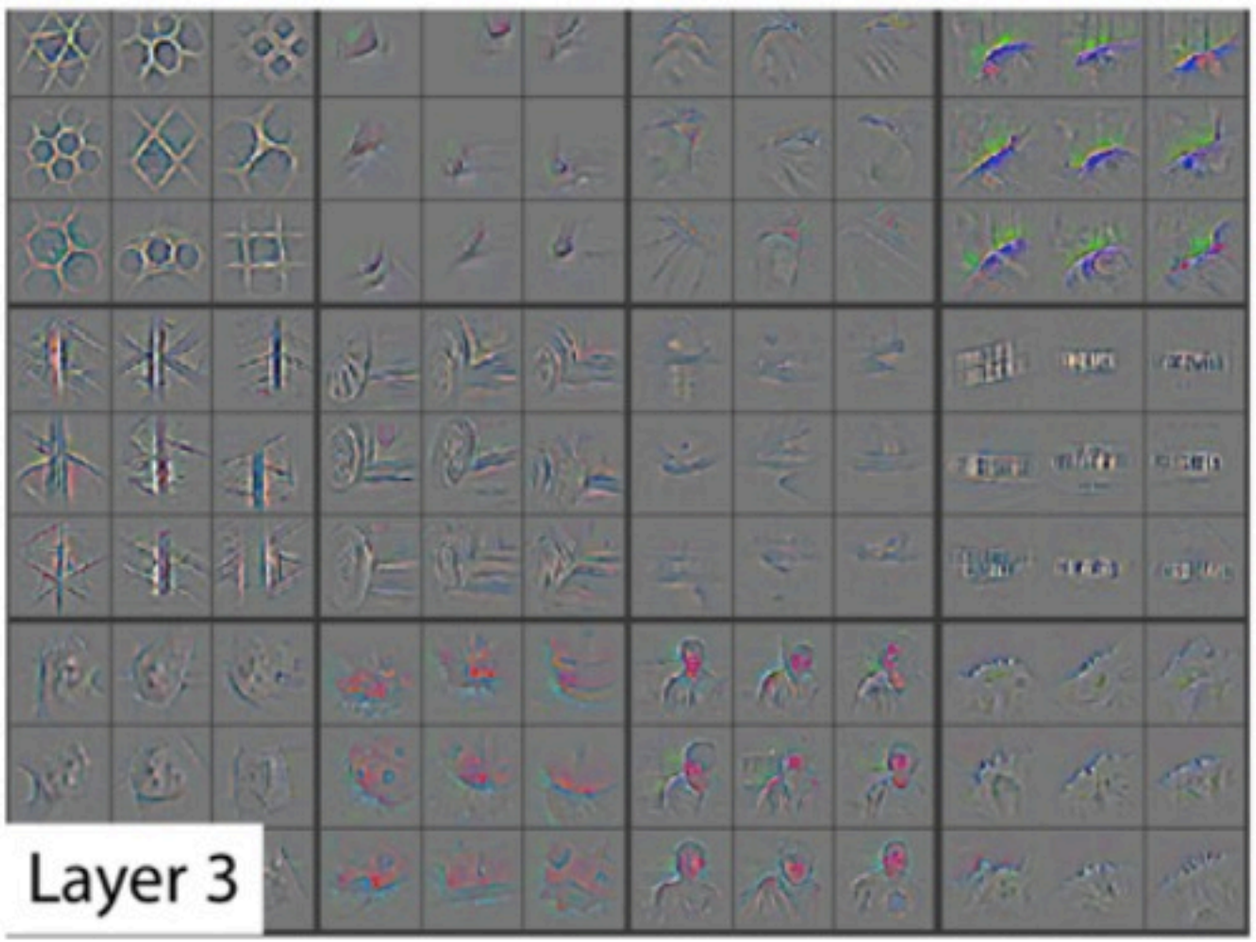

3
224
224
7
7
1, 2, 3, 4
96
37
37
5
5
1, 2, 4
256
17
17
3
3
1, 2
512
17
17
3
3
1, 2
512
17
17
3
3
1, 2, 4
512
6
6
5, 2
4096
1
1
5, 2
4096
1
1
5, 6
Class labels
1000

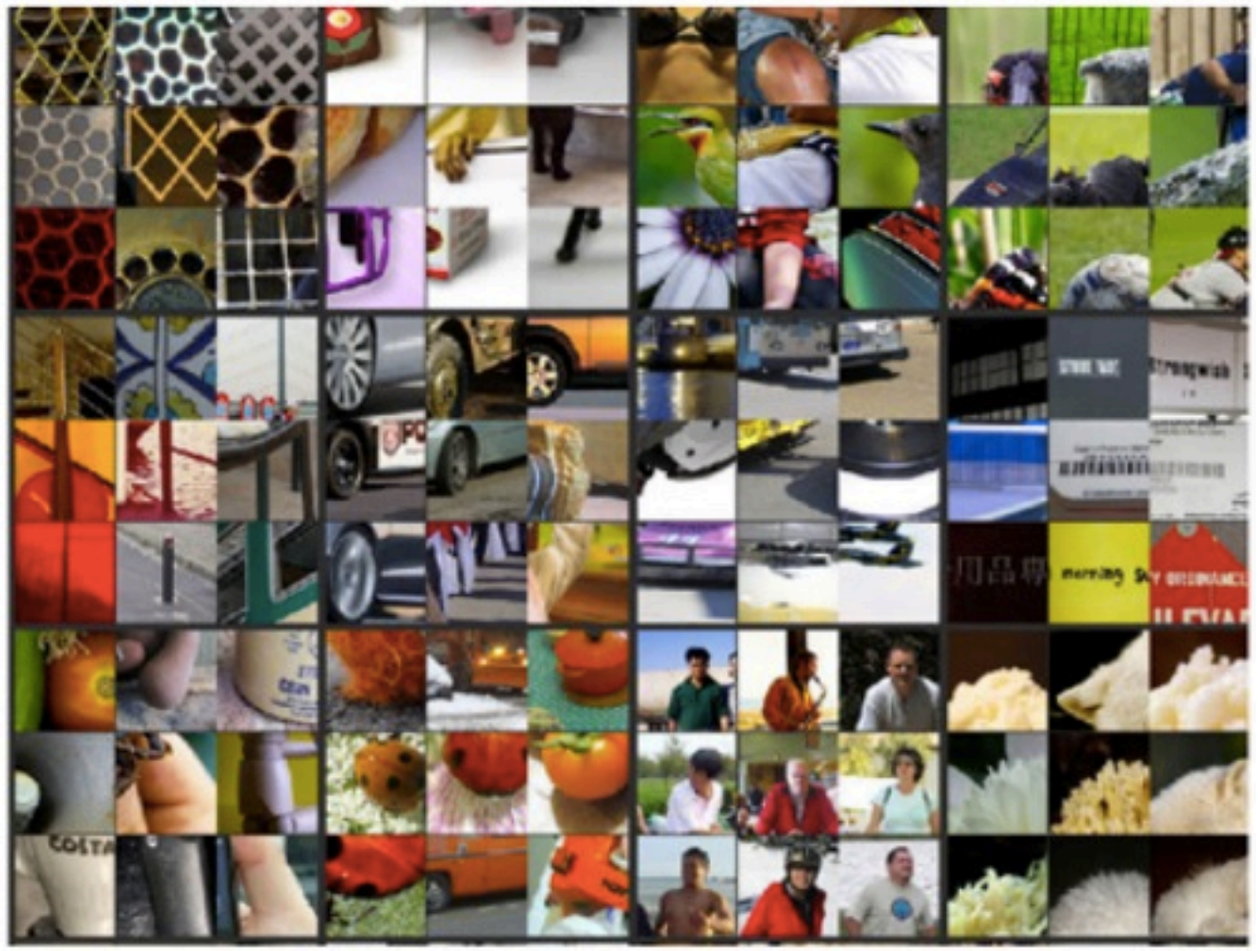
Layer 1

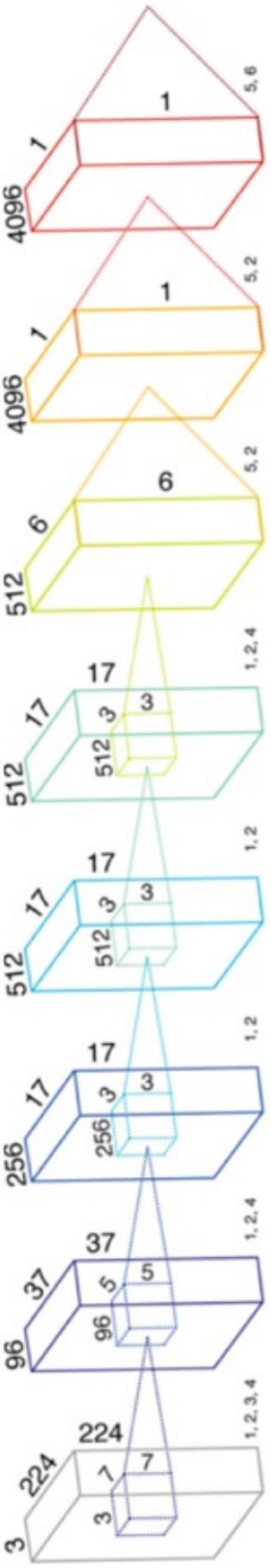

3
224
224
3
7
7
96
37
37
96
5
5
256
17
17
256
3
3
512
17
17
512
3
3
512
17
17
512
3
3
512
6
6
6
4096
1
1
1
4096
1
1
1
Class labels
1000
1, 2, 3, 4
1, 2, 4
1, 2
1, 2
1, 2, 4
5, 2
5, 2
5, 6

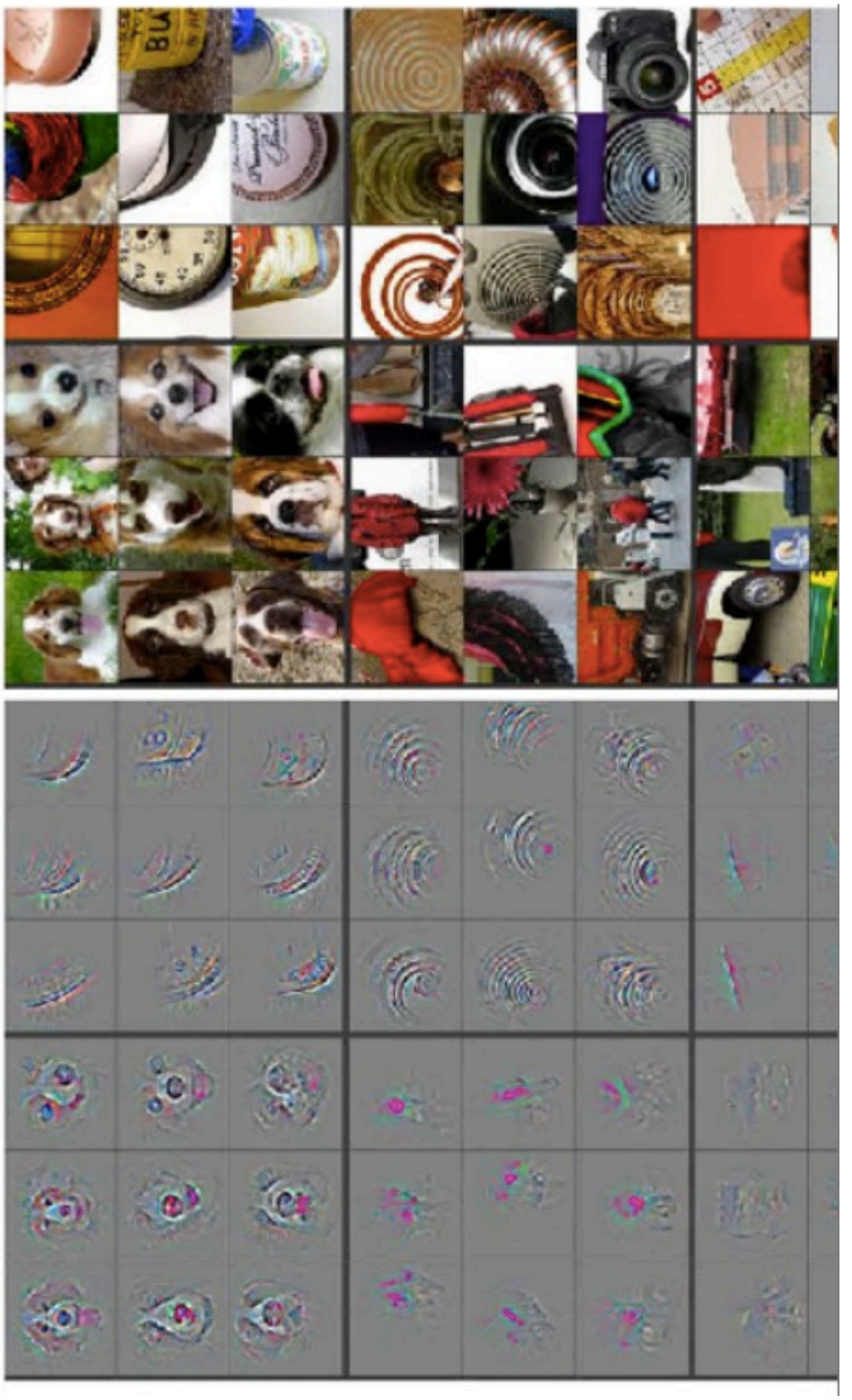

Layer 2


3
224
224
3
7
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
6
4096
1
1
1
4096
1
1
1
Class labels
1000
1, 2, 3, 4
1, 2, 4
1, 2
1, 2
1, 2, 4
5, 2
5, 2
5, 6

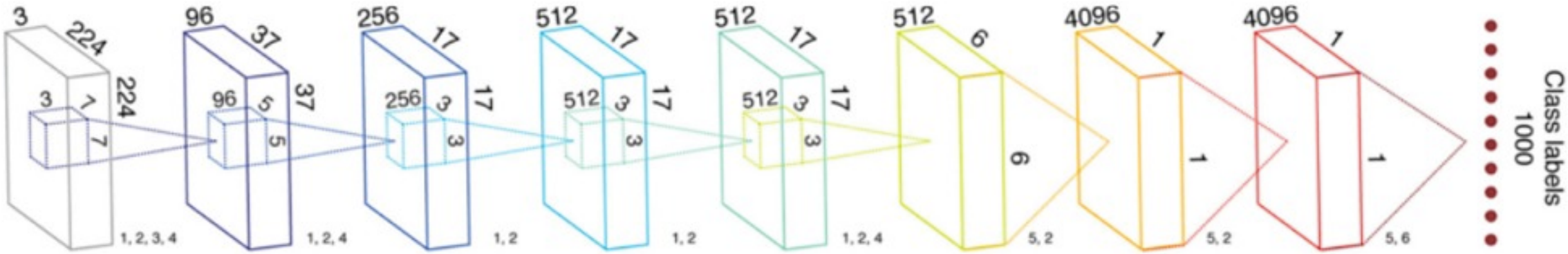

Layer 3

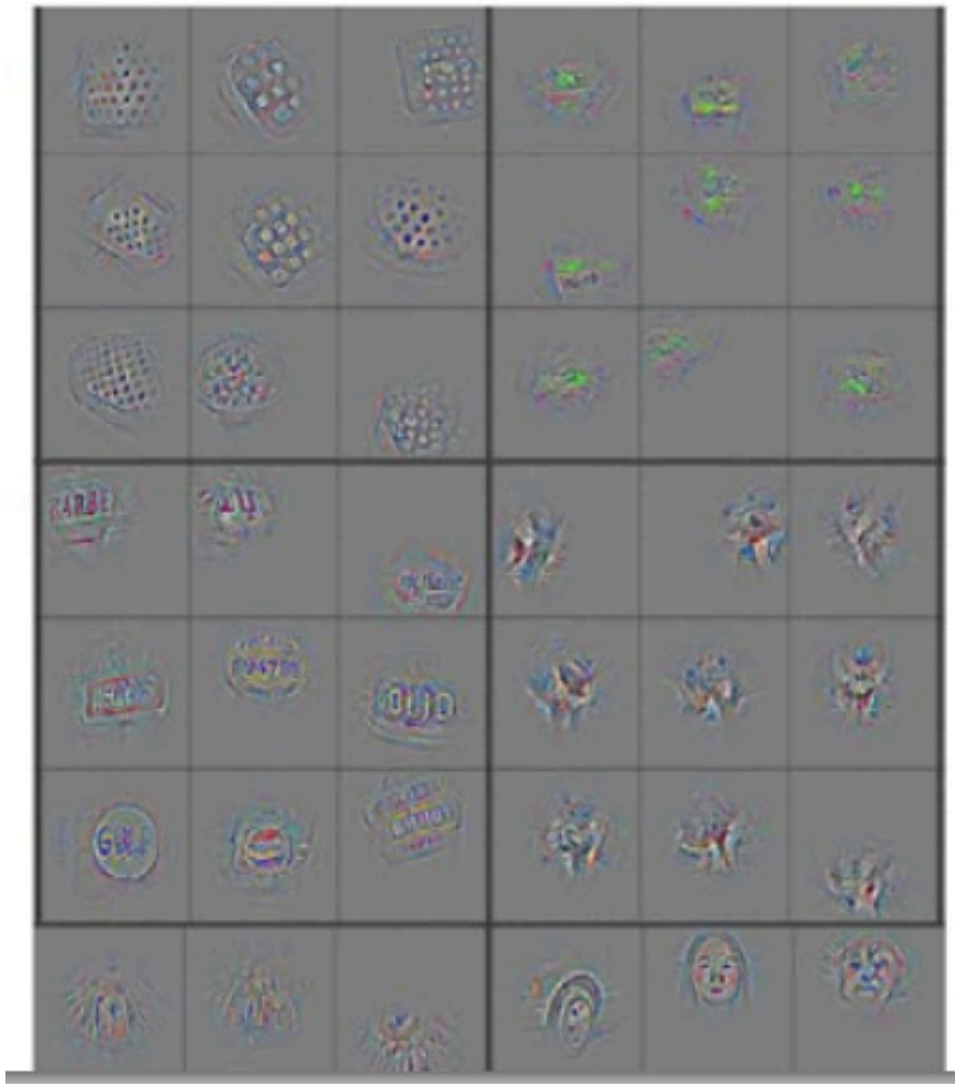


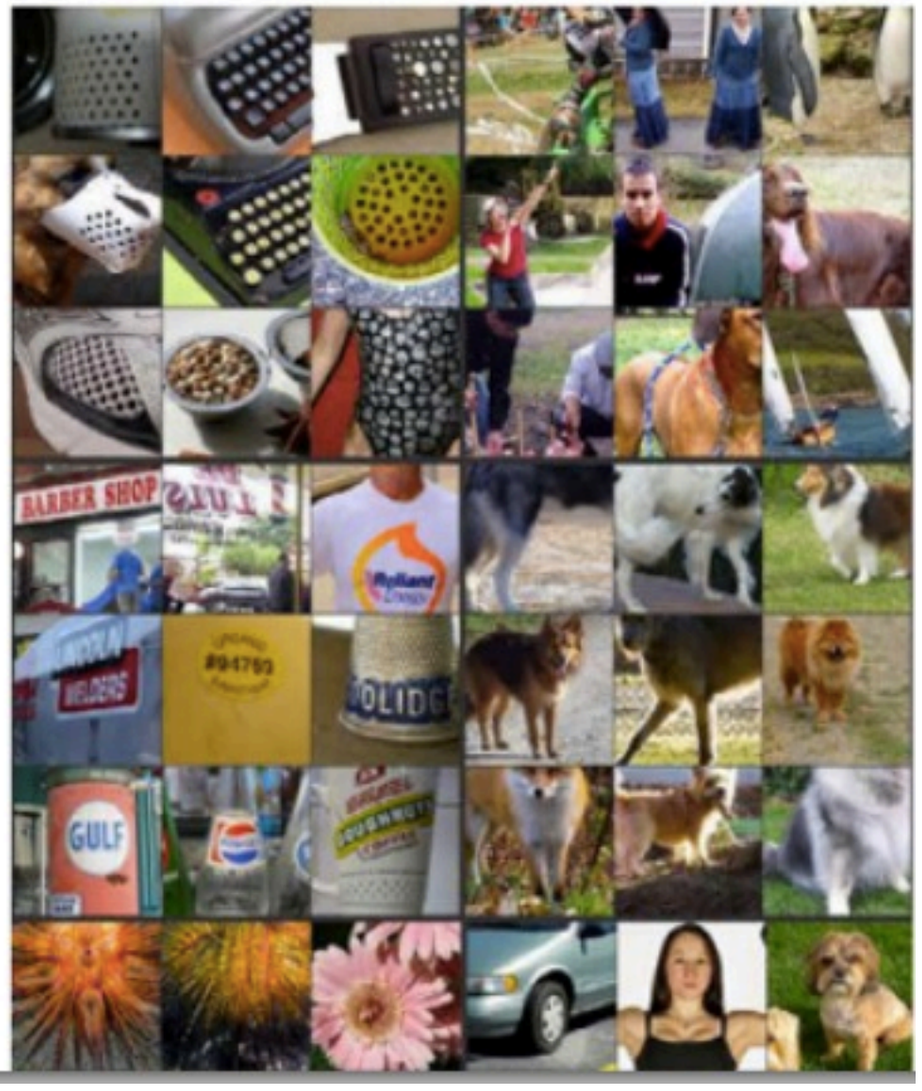




sábado, 17 de octubre de 15

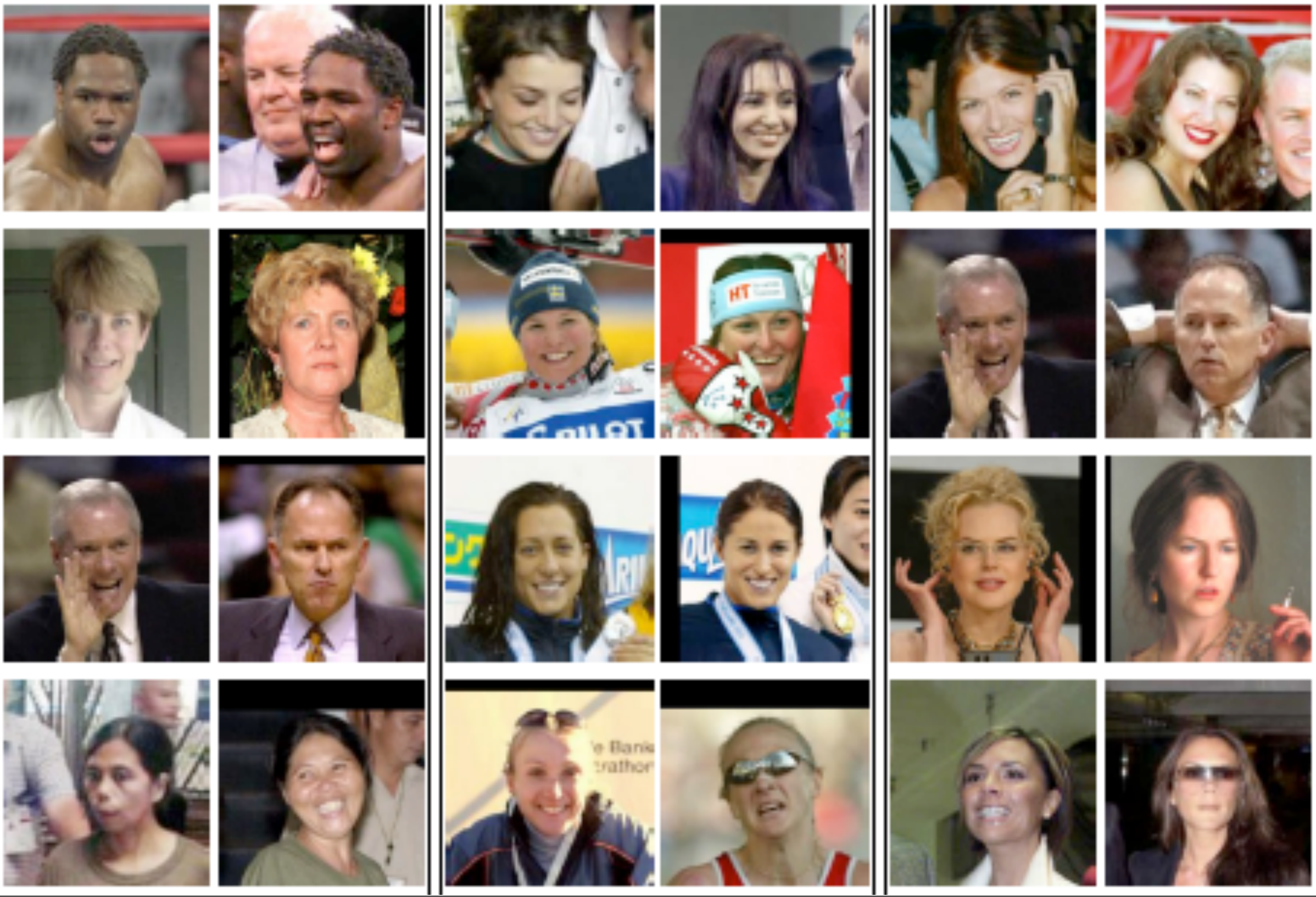

3
224
224
3
7
7
96
37
37
96
5
5
256
17
17
256
3
3
512
17
17
512
3
3
512
17
17
512
3
3
512
6
6
6
4096
1
1
1
4096
1
1
1
Class labels
1000
1, 2, 3, 4
1, 2, 4
1, 2
1, 2
1, 2, 4
5, 2
5, 2
5, 6

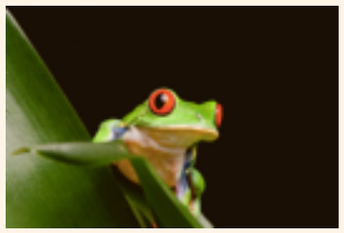
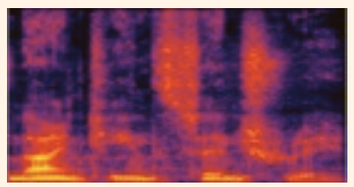
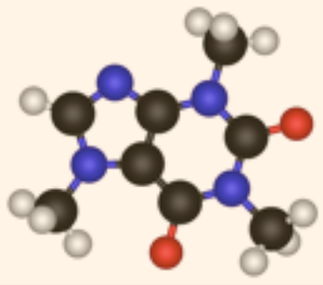
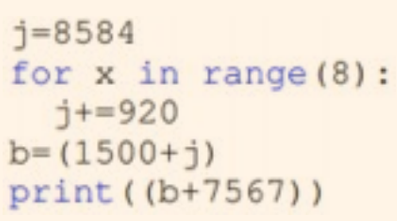



BARBER SHOP
GULF

False accept

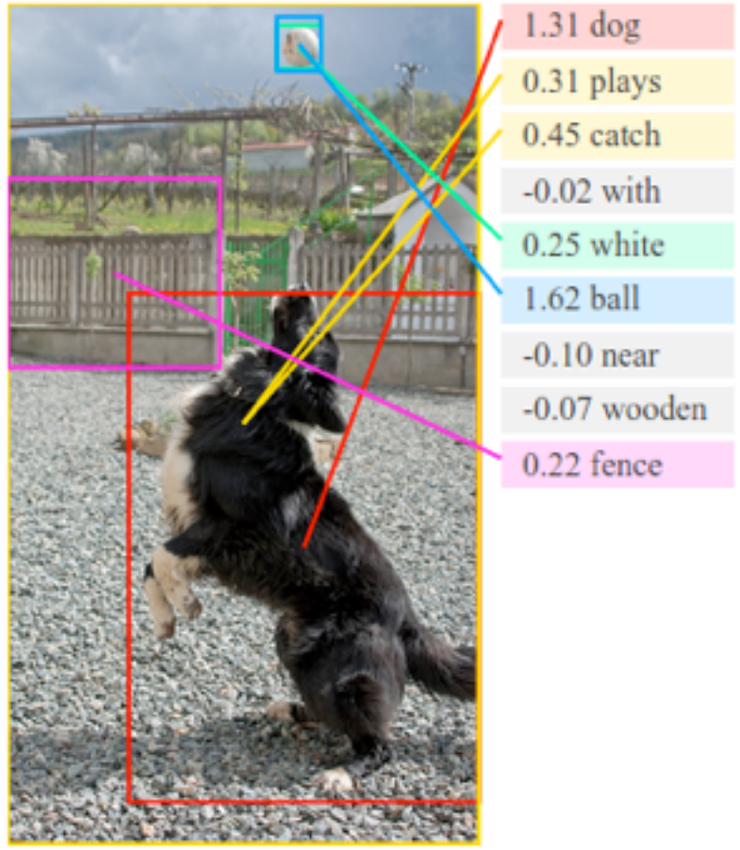
False reject



Input	Output
Pixels: 	“frog”
Audio: 	“na in tiz sen tu rii”
“Buenos dias, que tal estas?”	“Guten morgen, wie geht es dir?”
	“Toxic”
* 	25011.

"man in black shirt is playing guitar."

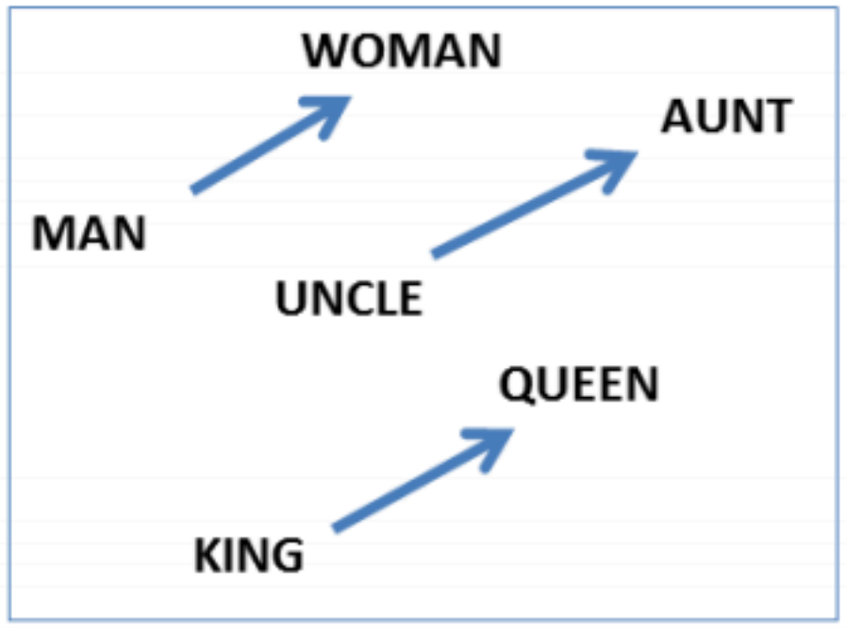
"construction worker in orange safety vest is working on road."

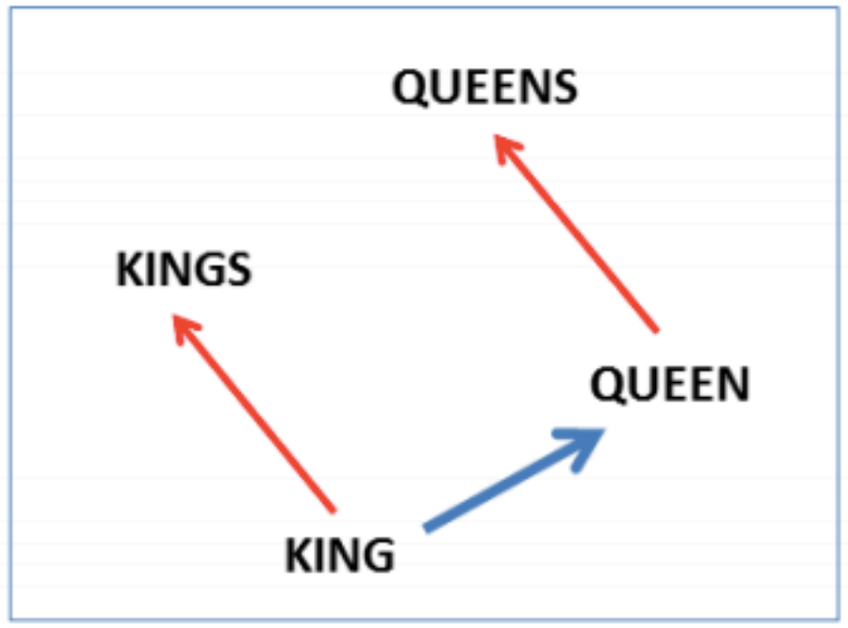


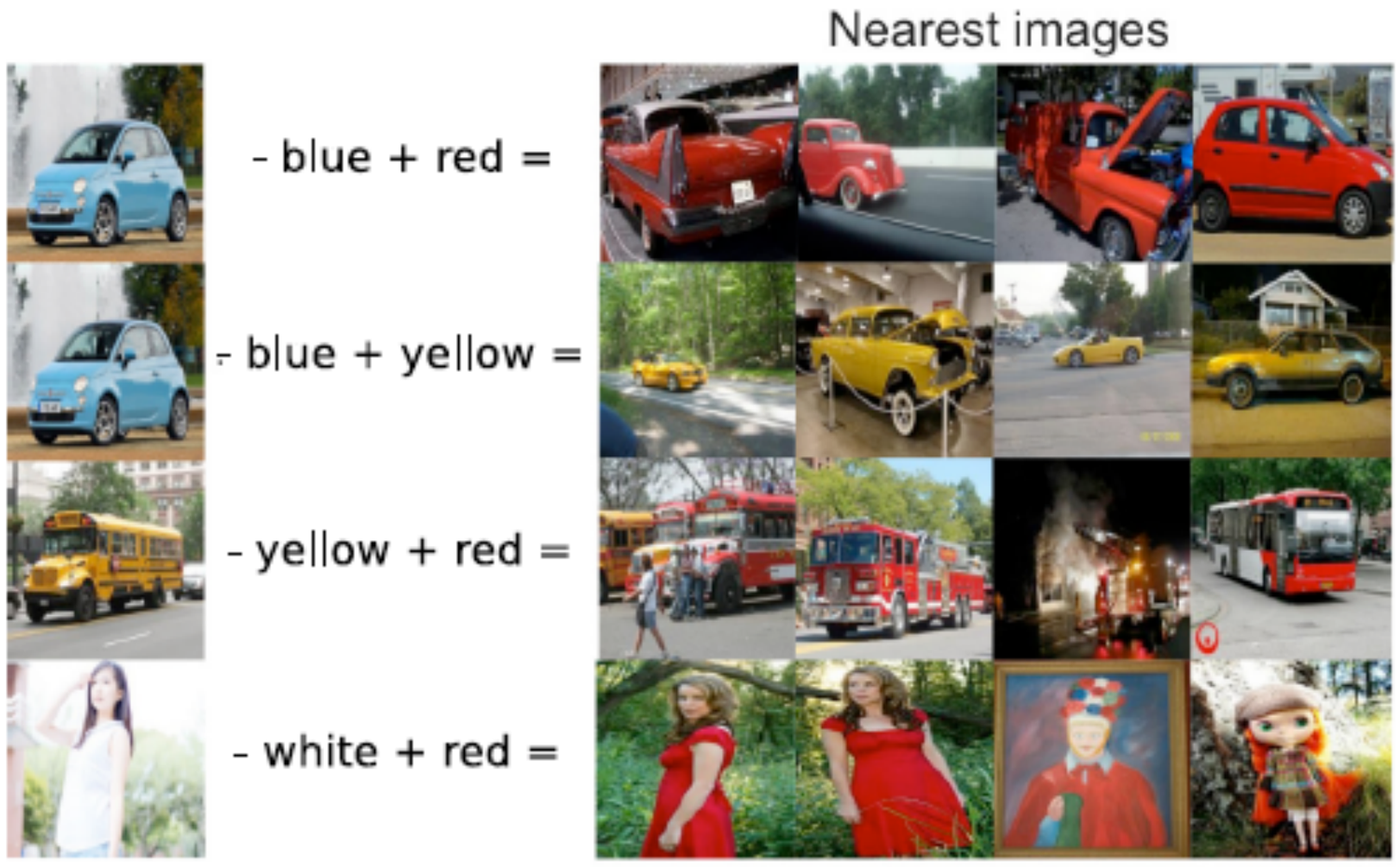
"girl in pink dress is jumping in air."

"black and white dog jumps over bar."

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


WOMAN
AUNT
MAN
UNCLE
QUEEN
KING


QUEENS
KINGS
QUEEN
KING

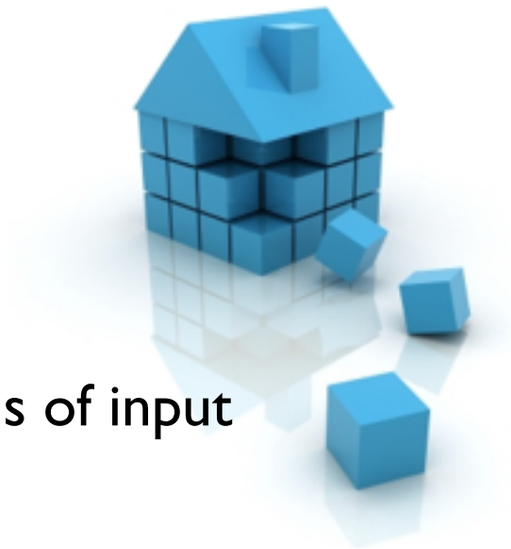

Nearest images
- blue + red =
- blue + yellow =
- yellow + red =
- white + red =

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

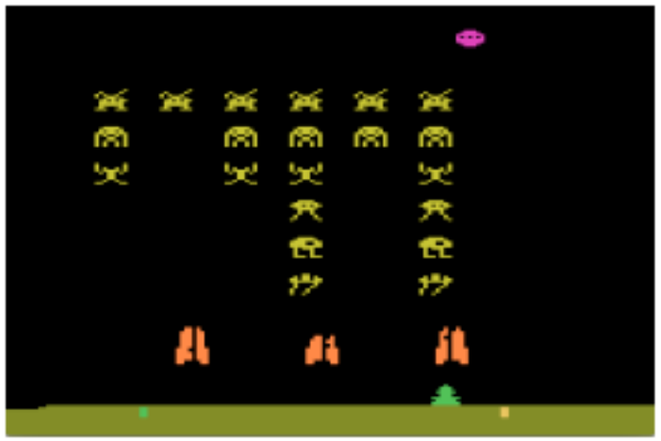
Playing Atari with Deep Reinforcement Learning

Playing Atari with Deep Reinforcement Learning

Playing Atari with Deep Reinforcement Learning

Self-taught AI that learns to play better than humans

Playing Atari with Deep Reinforcement Learning



Self-taught AI that learns to play better than humans

19.12.2013

DeepMind submits paper to ArXiv

28.01.2014

1st mention of DM in list

26.05.2014

Works but learning not confirmed

07.12.2014

Learning confirmed

2014

26.01.2014

Google acquires DM for €400M

31.03.2014

Decided to replicate it

25.09.2014

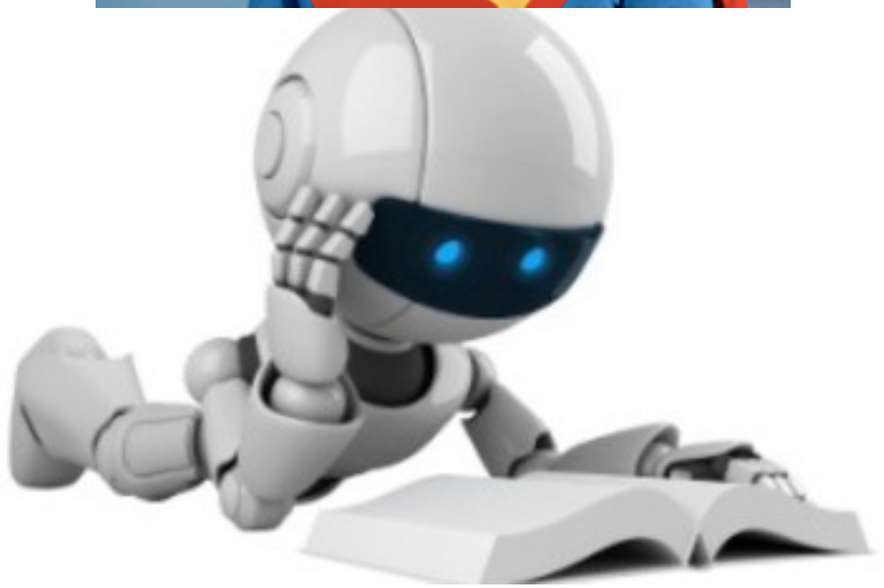
Robohub article

25.02.2015

DM's article

Why all the buzz...?

- Super-human level control
- Same algorithm learnt different games
- Learn from raw input



Playing Atari with Deep Reinforcement Learning

Playing Atari with Deep Reinforcement Learning
Mnih, et al Dec. 2013, arXiv:1312.5602

Playing **Atari** with Deep Reinforcement Learning

Playing Atari with Deep Reinforcement Learning
Mnih, et al Dec. 2013, arXiv:1312.5602

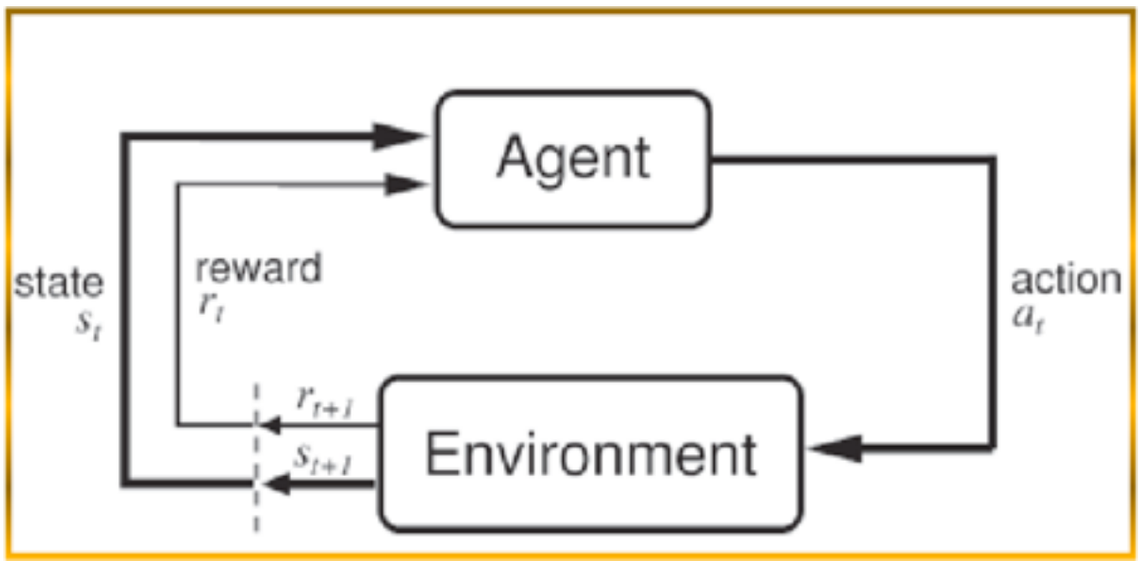
Playing **Atari** with Deep Reinforcement Learning

Playing Atari with Deep Reinforcement Learning
Mnih, et al Dec. 2013, arXiv:1312.5602

Playing Atari with Deep **Reinforcement Learning**

Playing Atari with Deep Reinforcement Learning
Mnih, et al Dec. 2013, arXiv:1312.5602

Playing Atari with Deep **Reinforcement Learning**

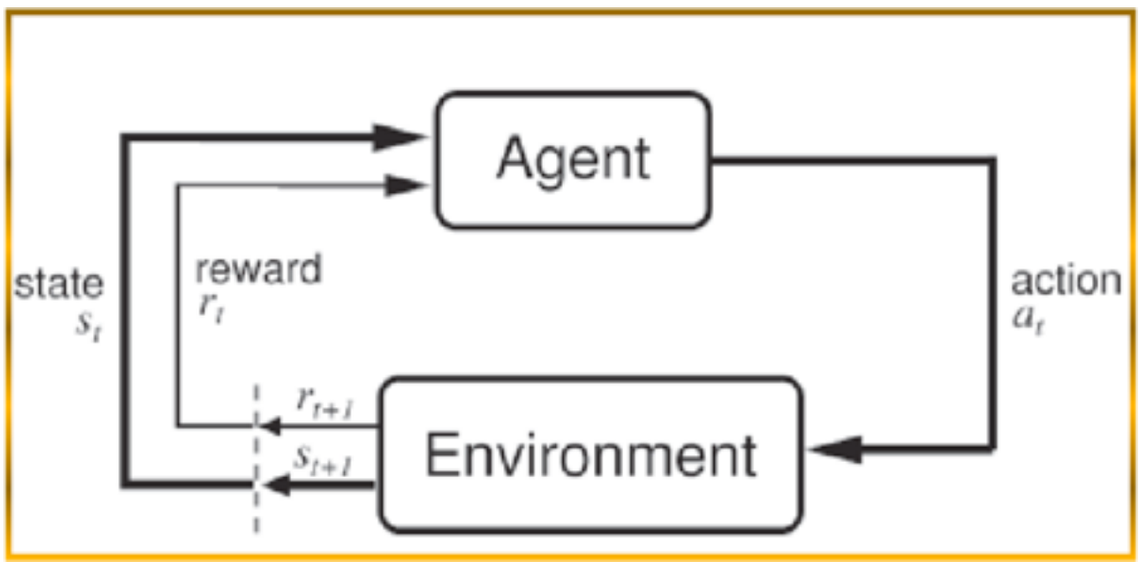


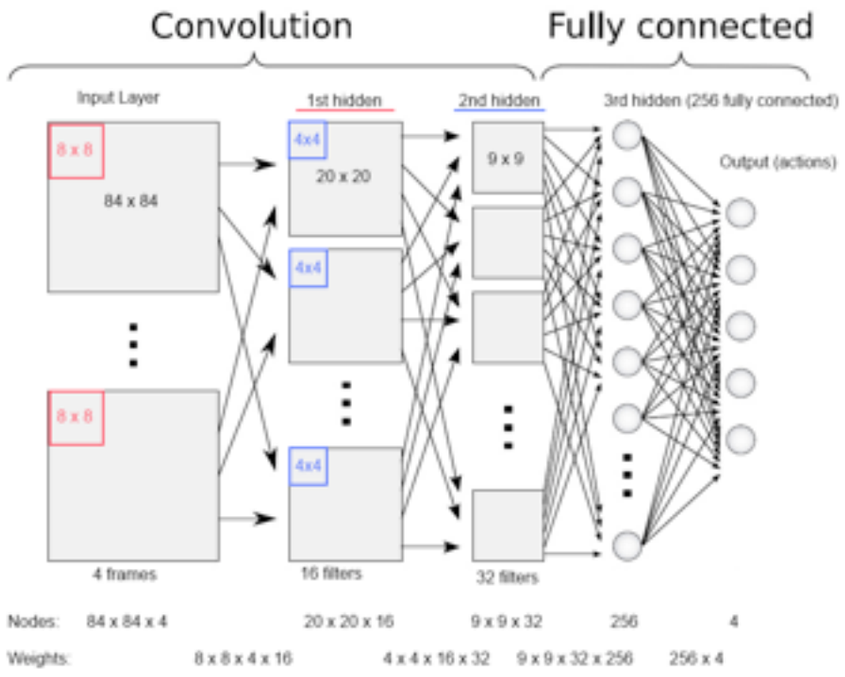
Playing Atari with Deep Reinforcement Learning
Mnih, et al Dec. 2013, arXiv:1312.5602

Playing Atari with **Deep** Reinforcement Learning

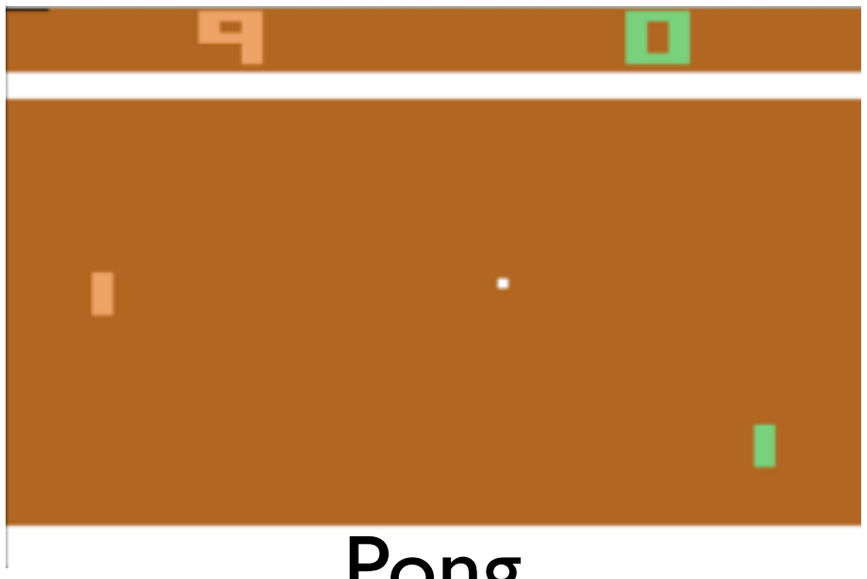
Playing Atari with Deep Reinforcement Learning
Mnih, et al Dec. 2013, arXiv:1312.5602

Playing Atari with **Deep** Reinforcement Learning

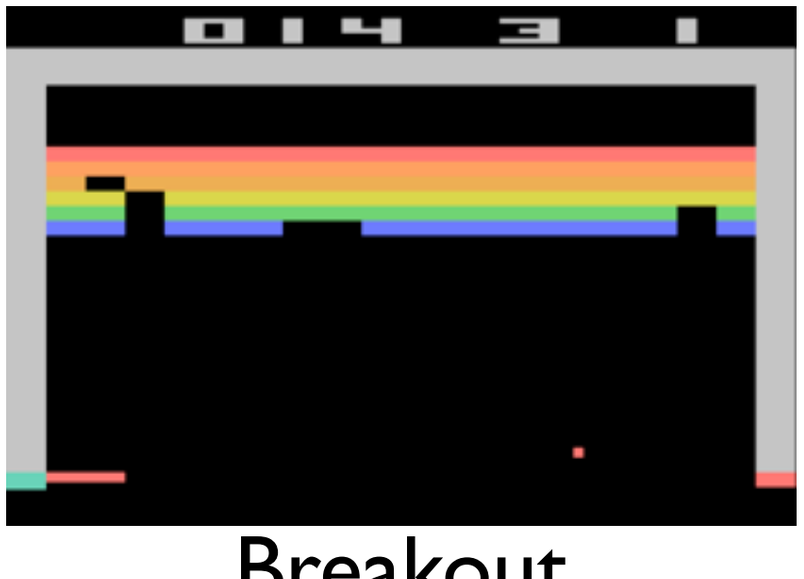




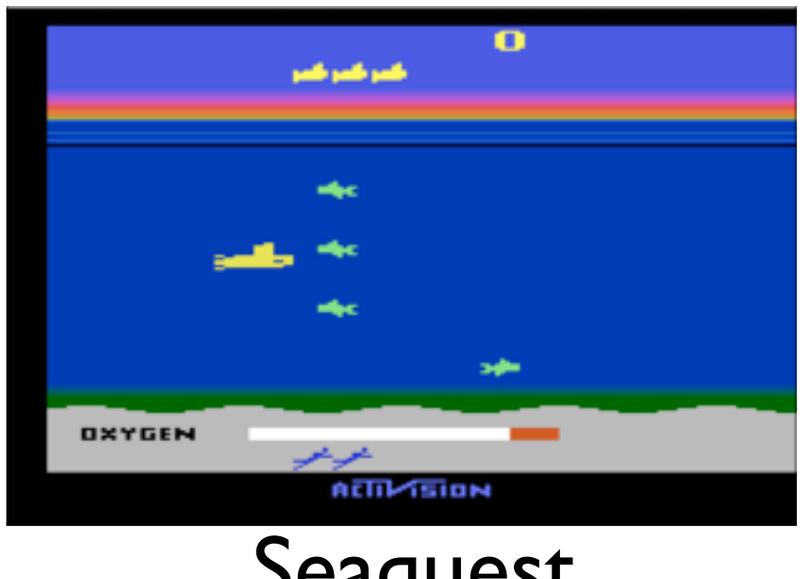
Playing Atari with Deep Reinforcement Learning
Mnih, et al Dec. 2013, arXiv:1312.5602



Pong

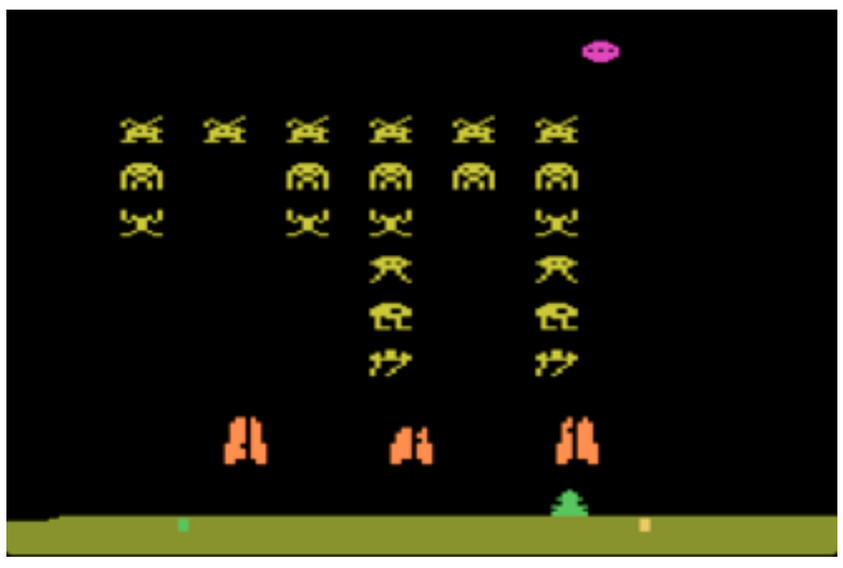


Breakout



Seaquest

Beam Rider

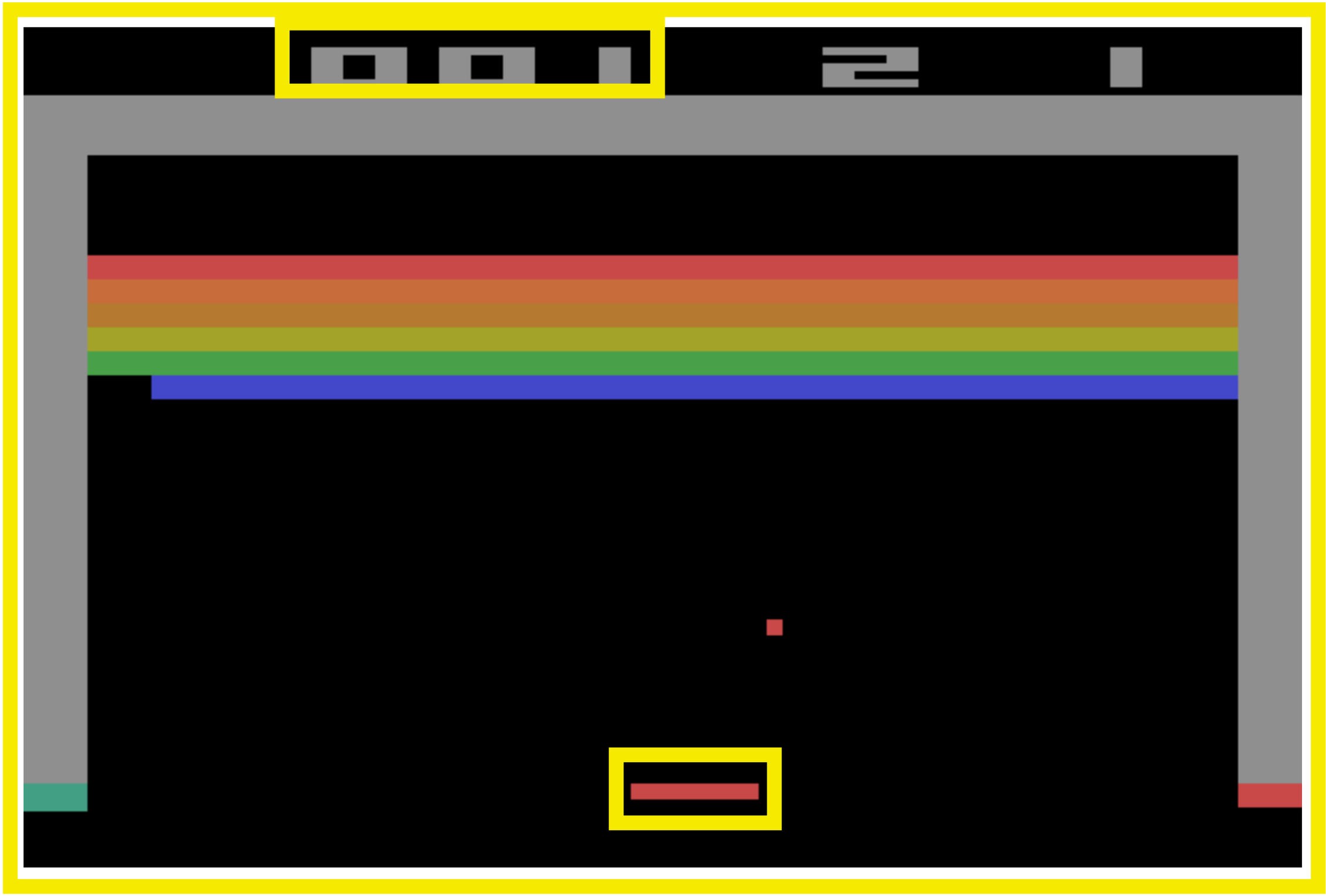


Space Invaders

Enduro

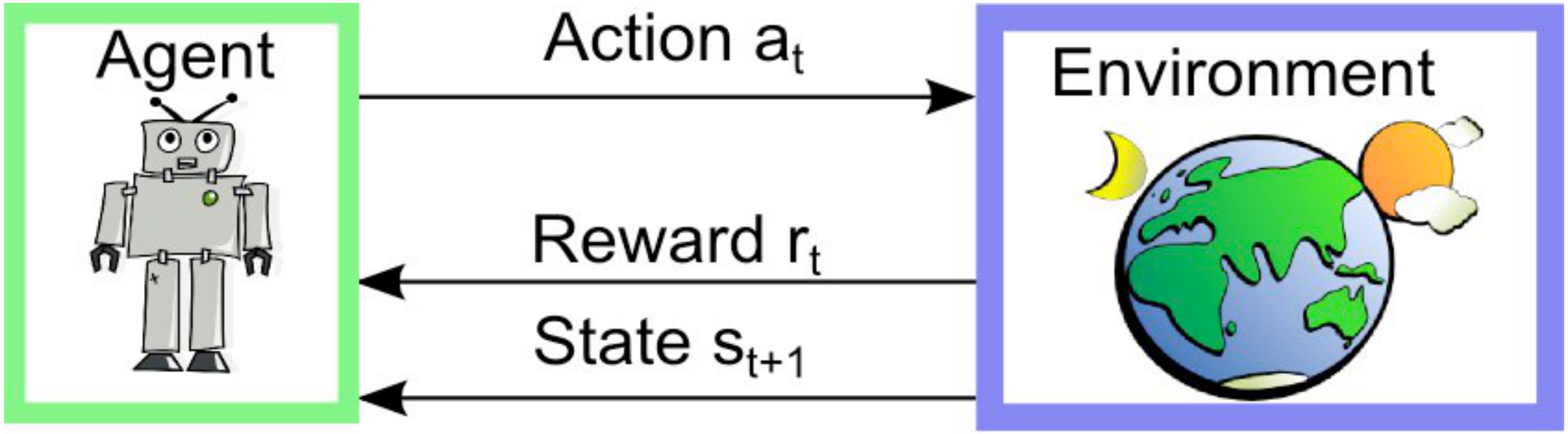
...and 43 more games

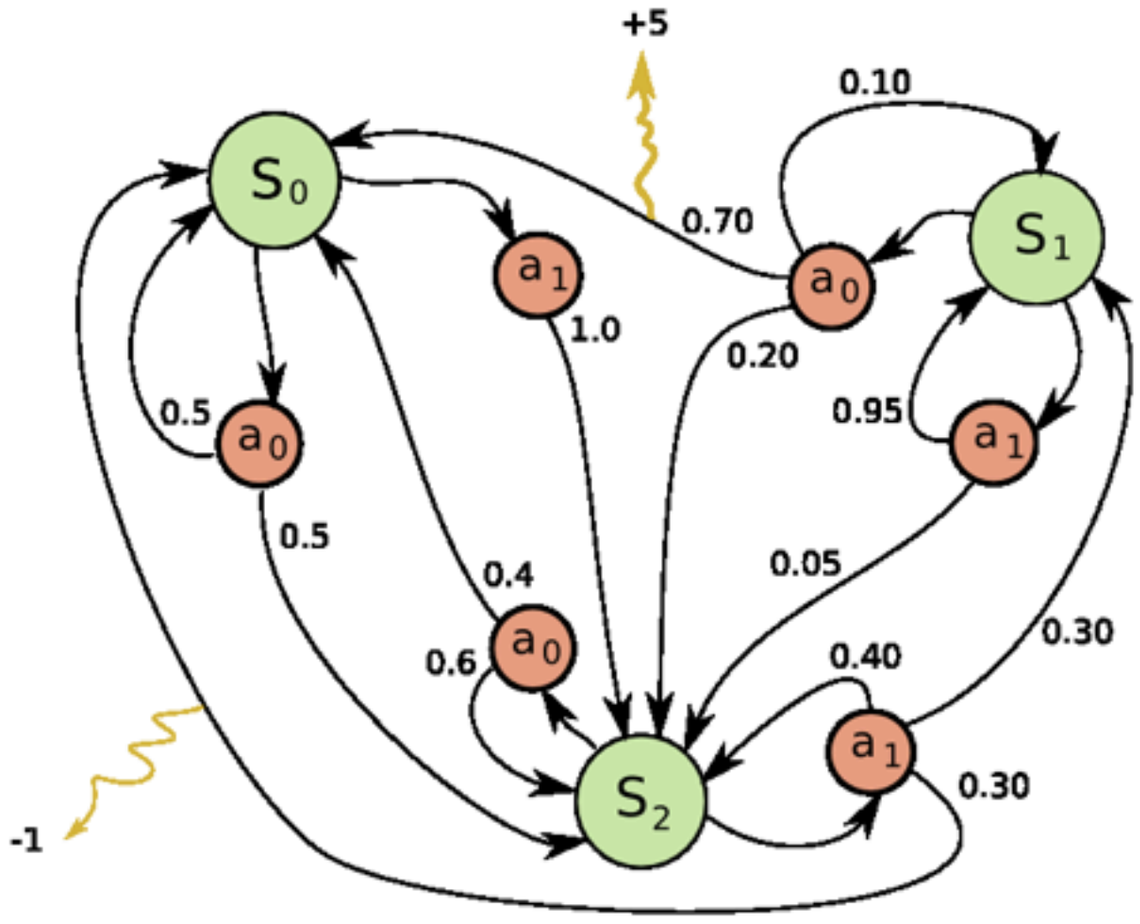
Reward



State

Action





Markov decision process

s, a, r, s, a, r,...

Why is hard?

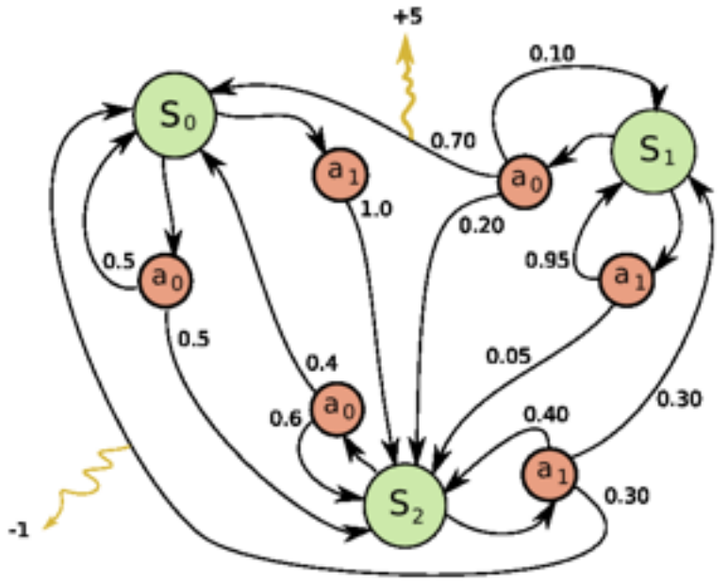
"I must have done something bad, but what is it?"

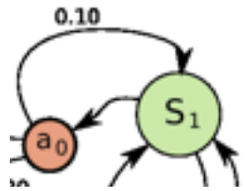
"Shall I continue with this strategy or try others?"

First, it has to learn to see!

It starts from scratch (pixels) with no high-level concept such as ball, or block, or anything!

Environment is not known!





s, a, r, s, a, r,...

84x84 pixels

256 gray

4 frames

...and it is huge!

Q-learning

- **Total reward** at time step t:

$$R_t = \sum_{i=0} \gamma^i r_{t+i} = r_t + \gamma R_{t+1}$$

- γ is the **discount factor** $0 \leq \gamma < 1$, that makes future rewards worth less than current

- Define **optimal action value function** $Q^*(s, a)$ as maximum expected return after taking an action *a* in state s, and then following the most optimal policy

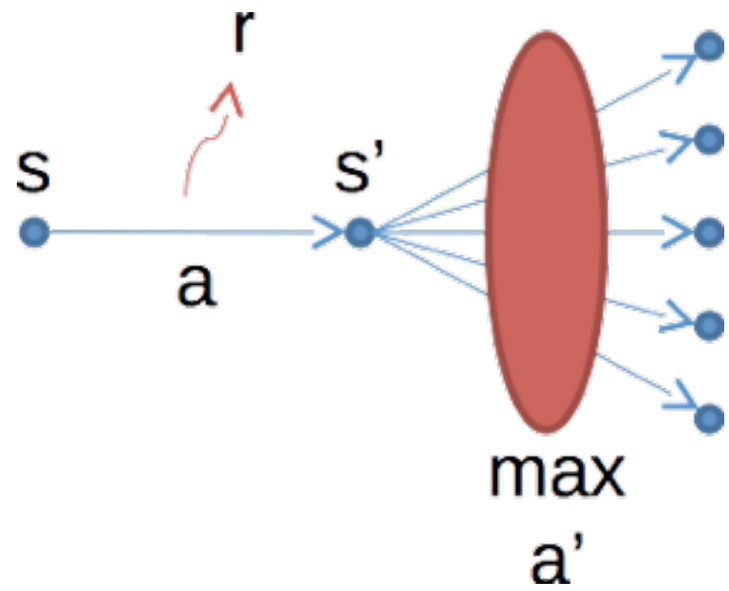
$$Q^*(s, a) = \max_{\pi} E\left[R | s_t = s, a_t = a, \pi\right]$$

- The **optimal strategy** in state s is to select an action *a* for which $Q^*(s, a)$ is maximum

Q-learning (continued)

- **Bellman equation:**

$$Q^*(s,a) = \underset{s'}{E}\left[r + \gamma \max_{a'} Q^*(s',a')\right]$$



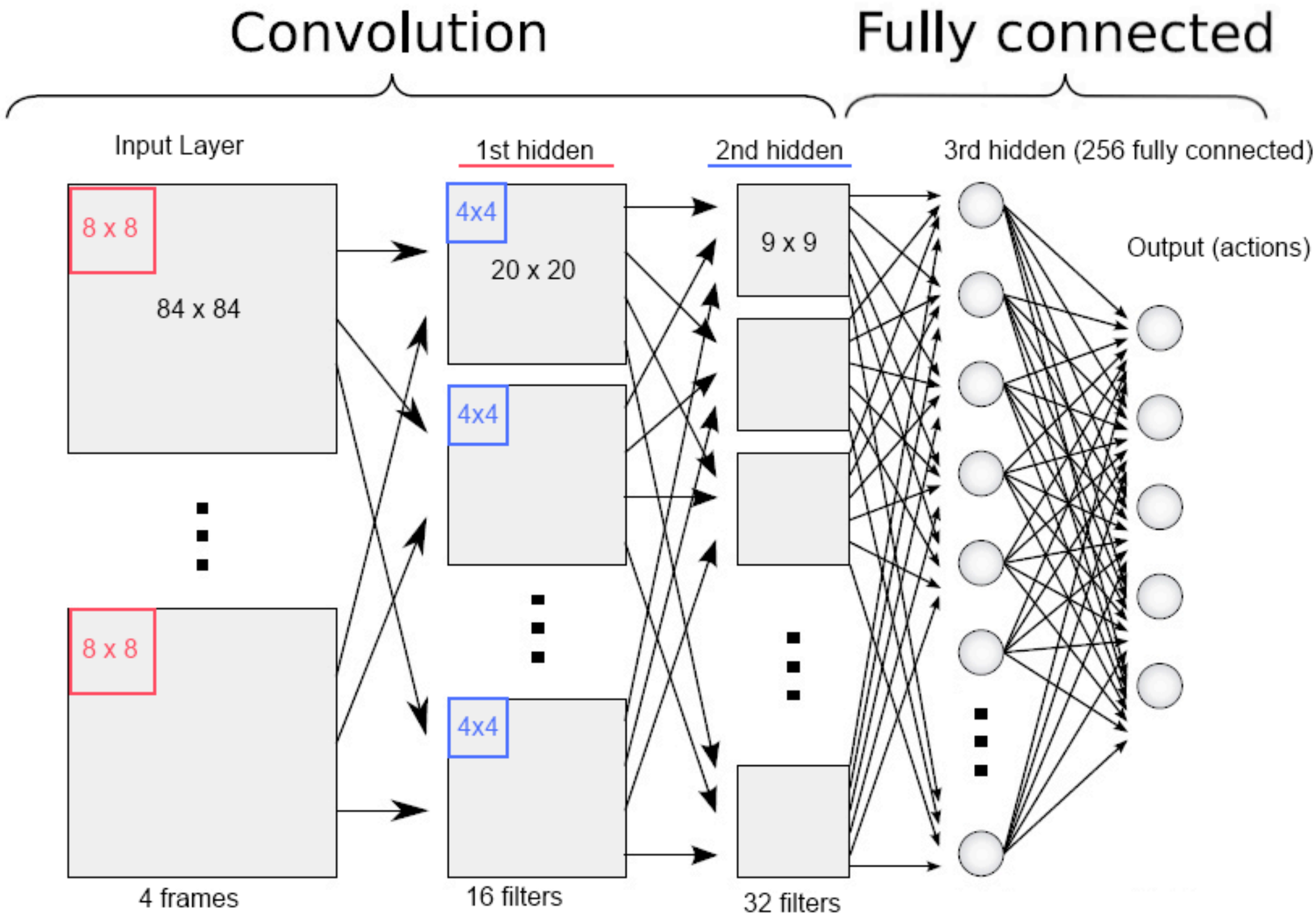
- **Iterative approximation**

$$Q_{i+1}(s,a) = Q_i(s,a) + \alpha\left(r + \gamma \max_{a'} Q_i(s',a') - Q_i(s,a)\right)$$

$$\lim_{i\to\infty} Q_i = Q^*$$

- If state and action spaces are finite, Q-function can be implemented as a table. But for large state spaces this table might be huge

- **Use convolutional neural networks to approximate the Q-function!**


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

Learning algorithm

- Play a lot of games
  - Play a lot of frames
    - Get a frame and **pre-process it**

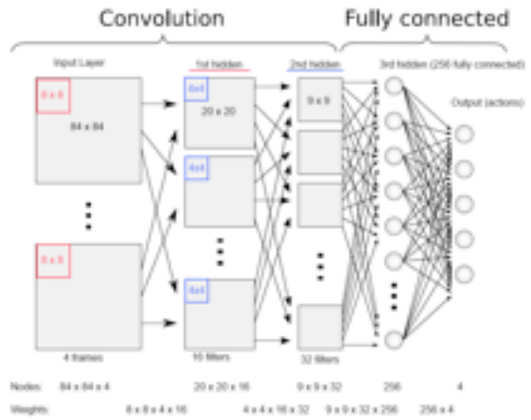
Pre-processing

- Image
    1. 210 x 160 RGB
    2. 110 x 84 gray-scale
    3. 84 x 84

- Every 4th frame from the emulator
- State contains 4 frames
- Reward is binarized

Learning algorithm

- Play a lot of games
  - Play a lot of frames
    - Get a frame and pre-process it
    - Chose action using **Deep Neural Net (DNN)** and current state
      - Do it randomly with probability e



Learning algorithm

- Play a lot of games
  - Play a lot of frames
    - Get a frame and pre-process it
    - Chose action using Deep Neural Net (DNN) and current state
      - Do it randomly with probability e
    - Observe reward and next image and save everything to the memory
    - Train the network
      - 32 random transitions: 32x{state,action,reward,state}
      - Q-learning

$$Q(s_t, a_t) = r_t + \gamma \max_a DNN(state_t + 1)$$

      - Gradient descent

In a nutshell...

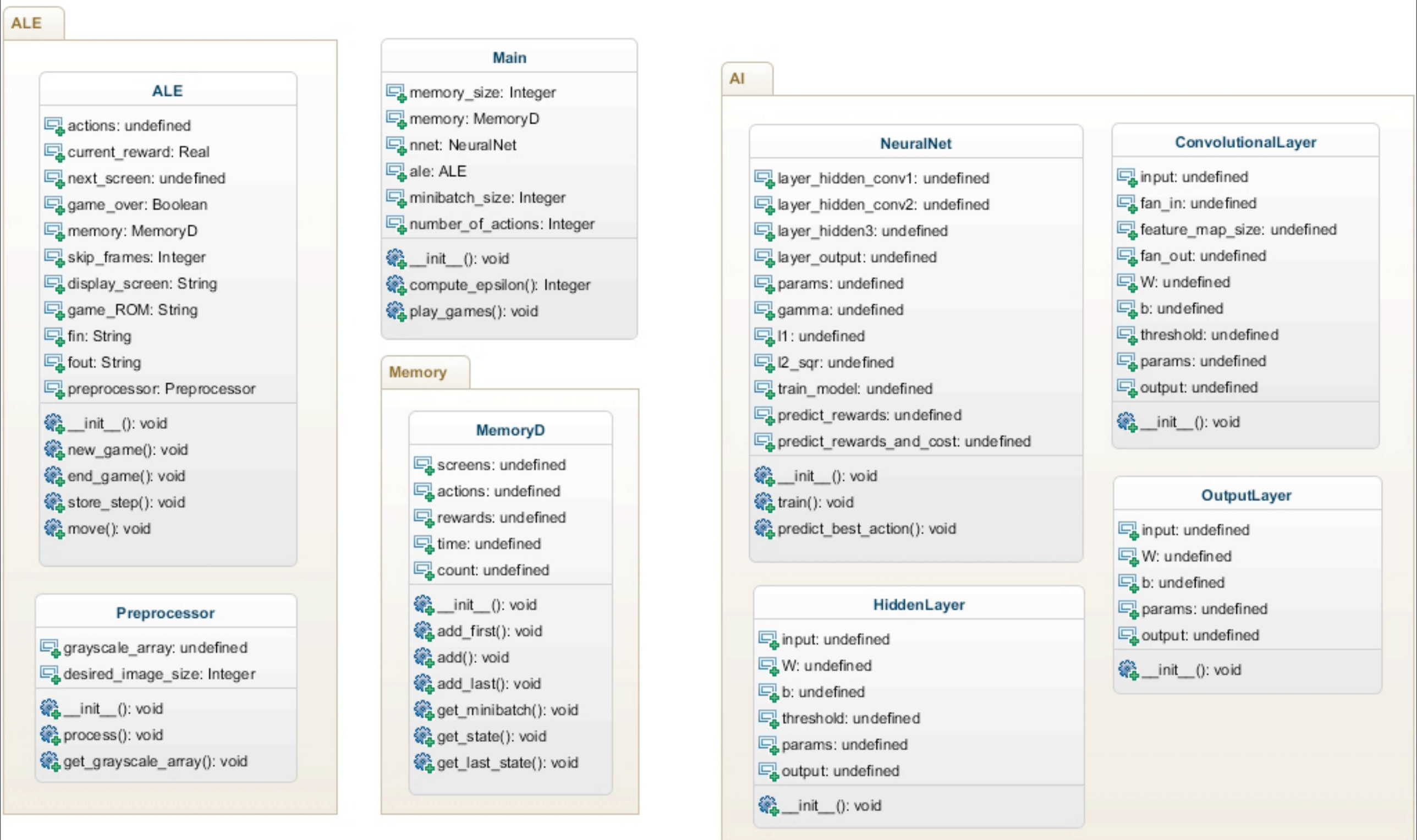
1. At each step choose action with **maximum total reward**

2. Use **convolutional neural network** to estimate this total reward

3. Use **Q-learning rule** to iteratively train this network to estimate better

4. Use **experience replay memory** to stabilize the learning

In real life...



https://github.com/kristjankorjus/Replicating-DeepMind/

Let's play

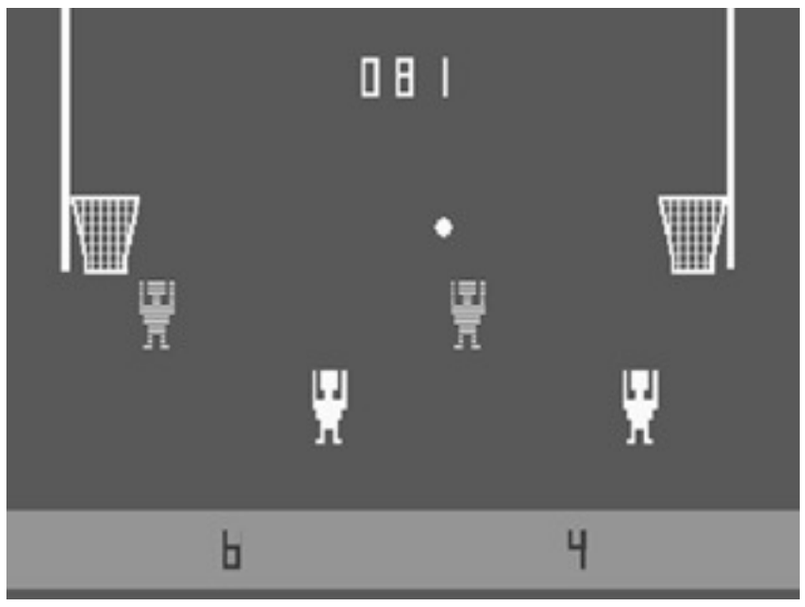
Why is it important?

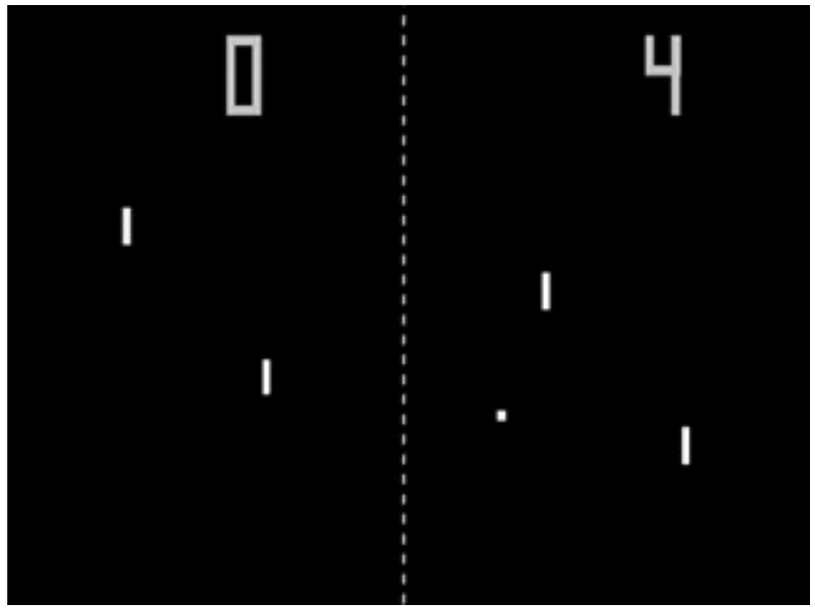
- Complex and different tasks
  - challenging even for humans
- Model free
  - same architecture for all games
- Beats all other results so far
  - benchmark for future research
- Easy to understand
  - sparked a lot of interest
- Parallels with Neuroscience

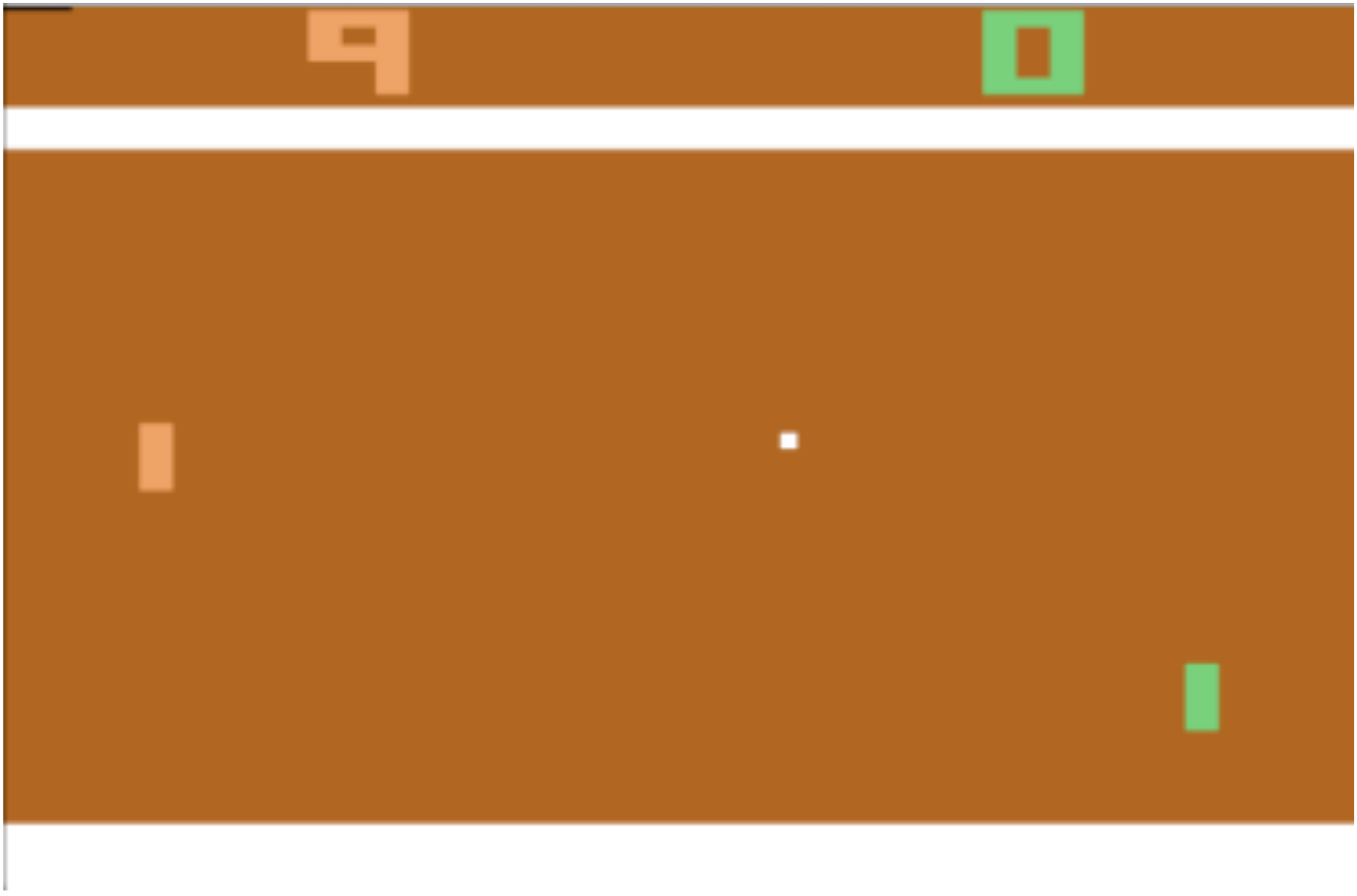
Ongoing projects

Collective phenomena: cooperation/competition in multi-agent learning systems

- Activate 2nd player (or more)
- Engineer the reward



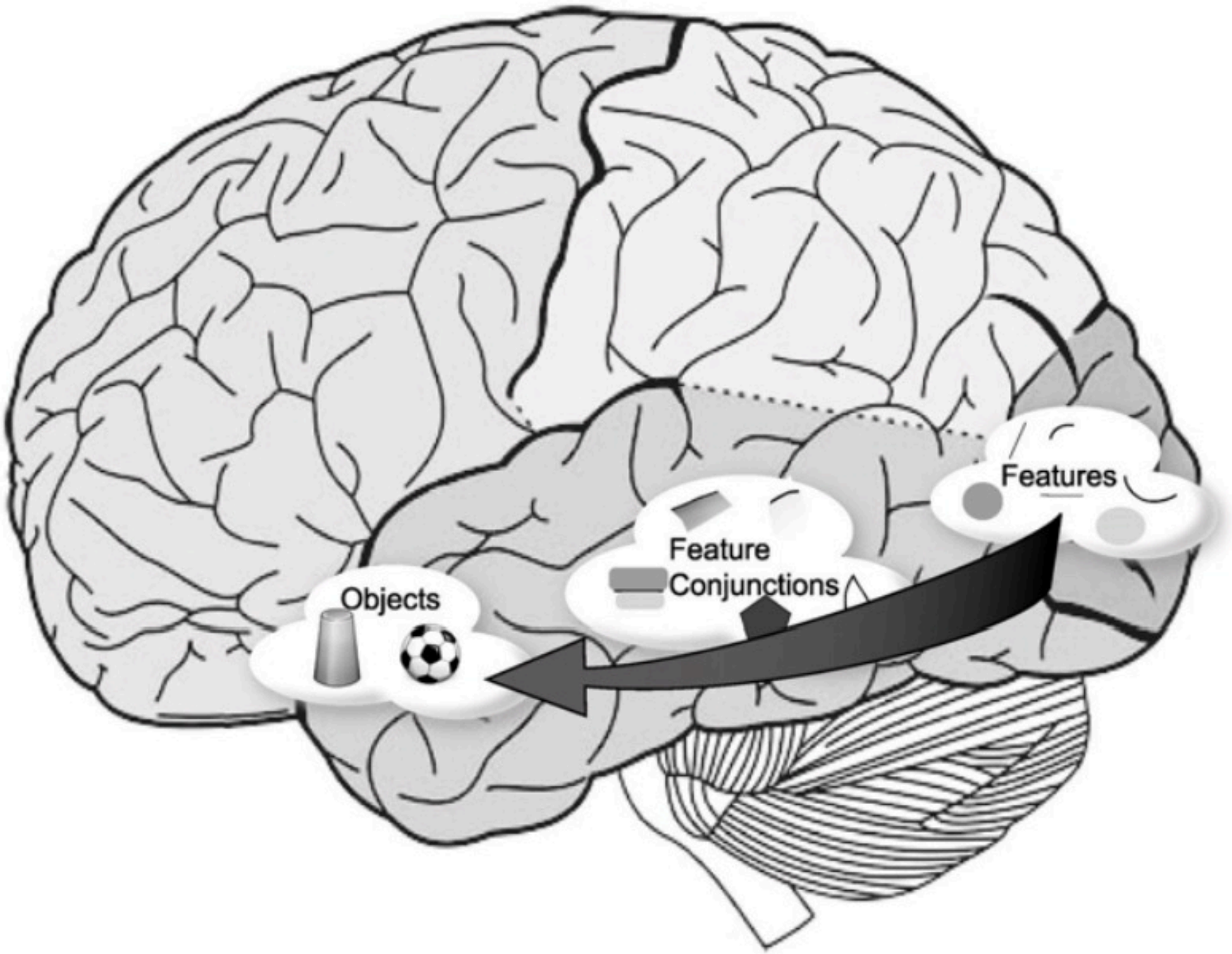


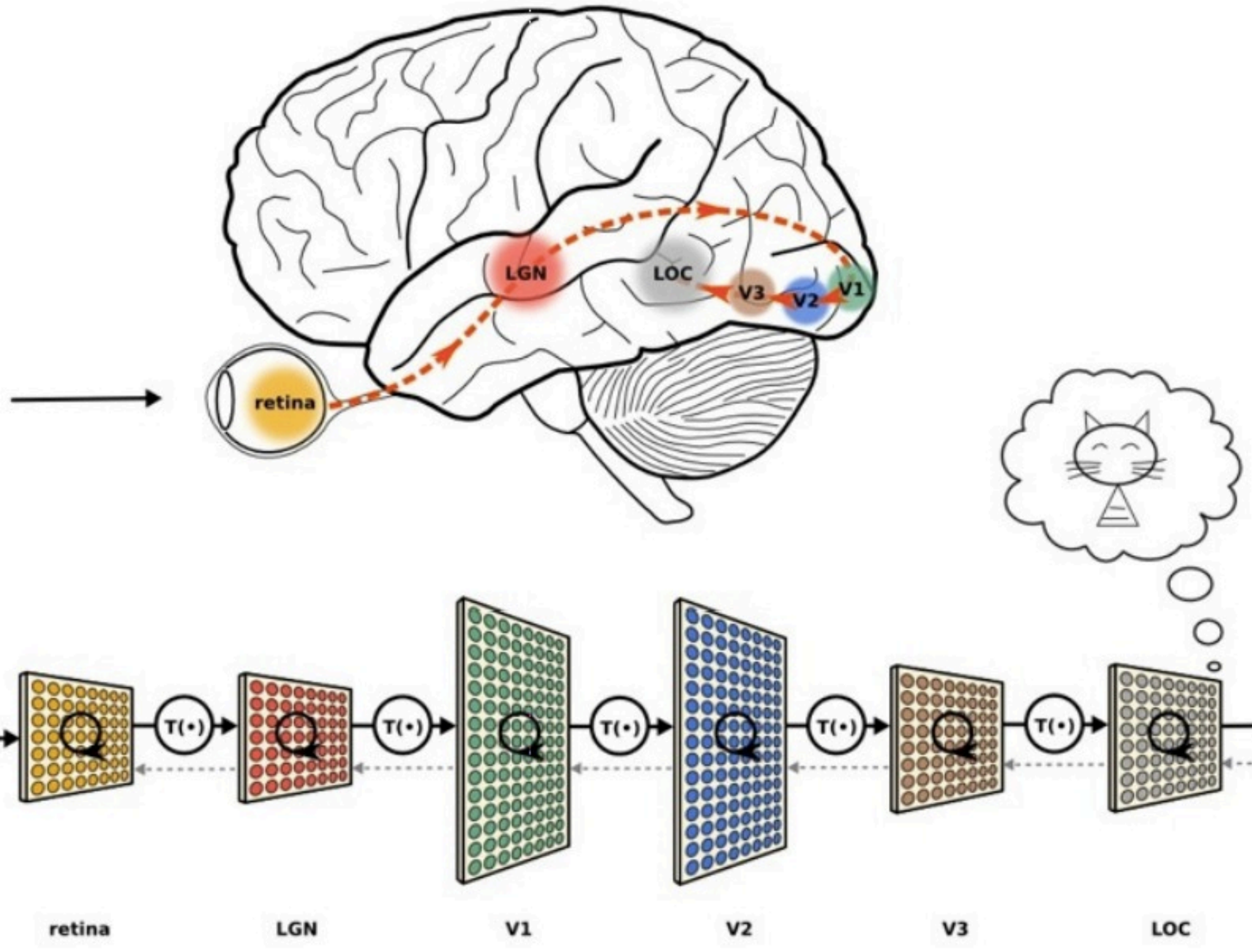

9
0

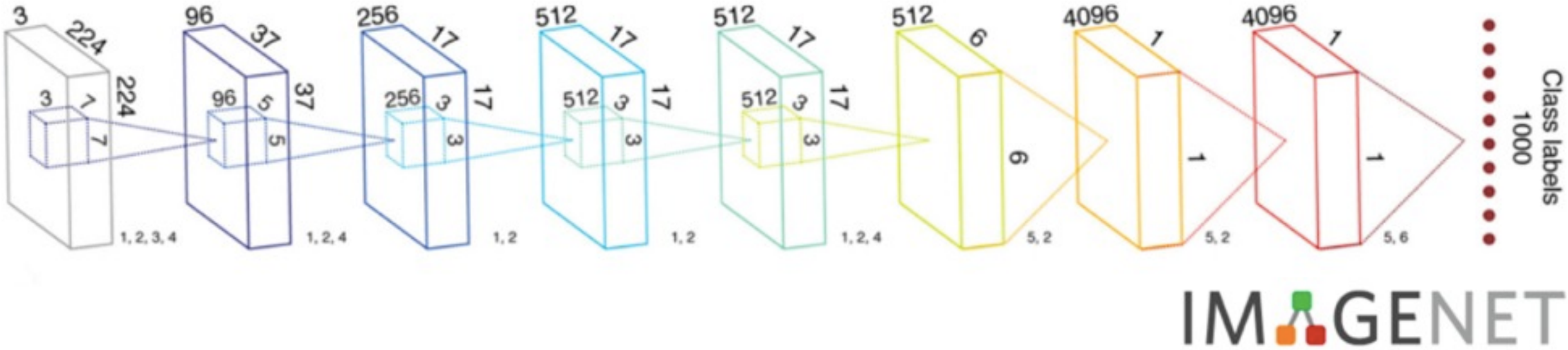
sábado, 17 de octubre de 15

Ongoing projects

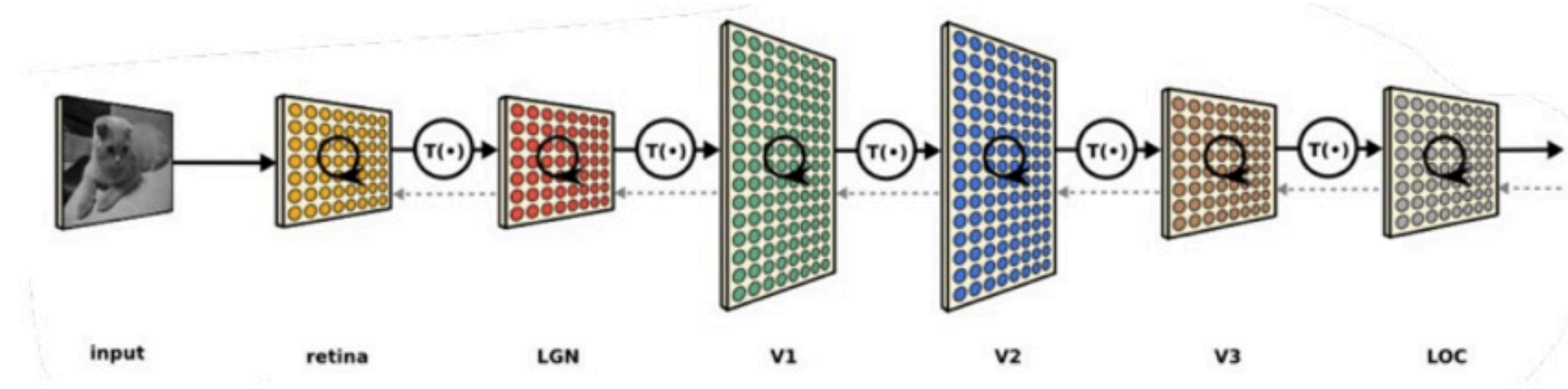
Deep Networks vs brain: compare CNN vs brain responses to natural images using intra-craneal recordings

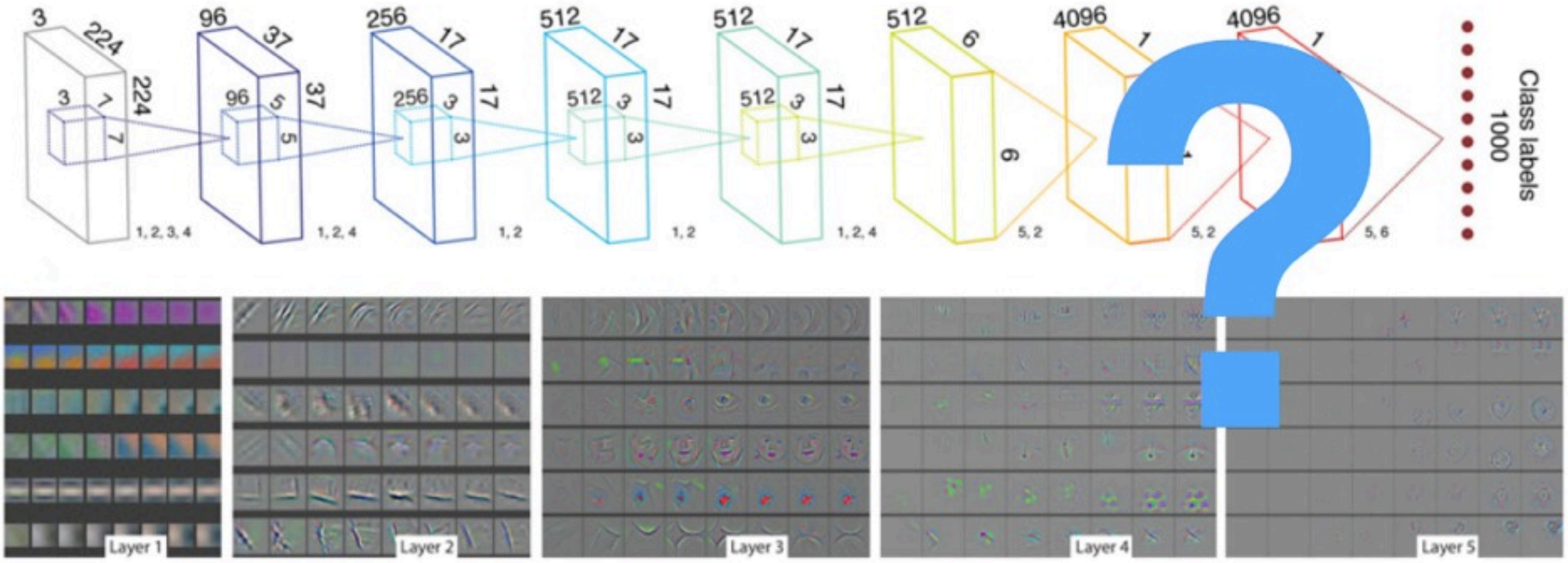

Features
Feature
Conjunctions
Objects


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

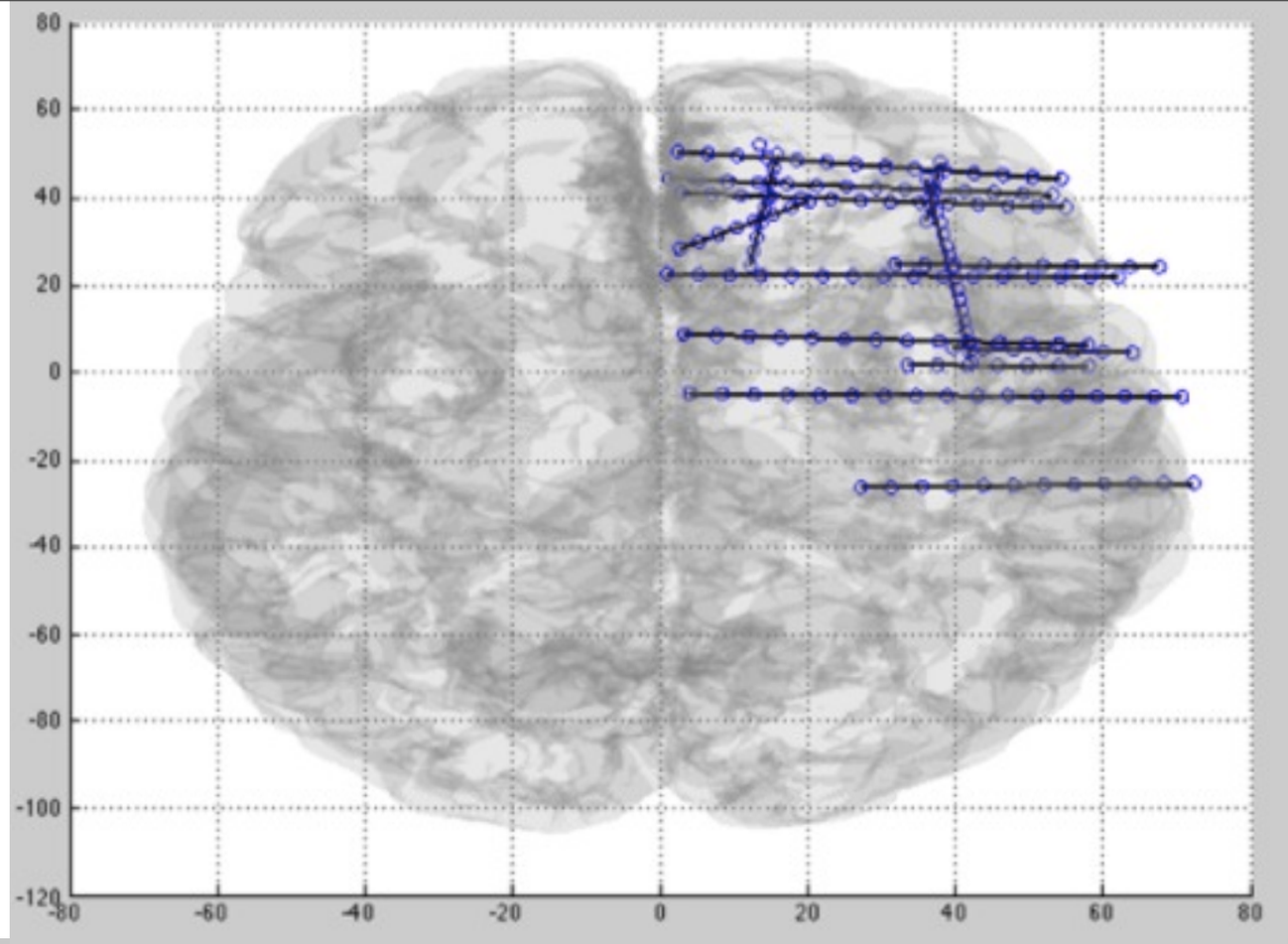

3
224
224
3
7
7
1, 2, 3, 4
96
37
37
96
5
5
1, 2, 4
256
17
17
256
3
3
1, 2
512
17
17
512
3
3
1, 2
512
17
17
512
3
3
1, 2, 4
512
6
6
5, 2
4096
1
1
5, 2
4096
1
1
5, 6
Class labels
1000

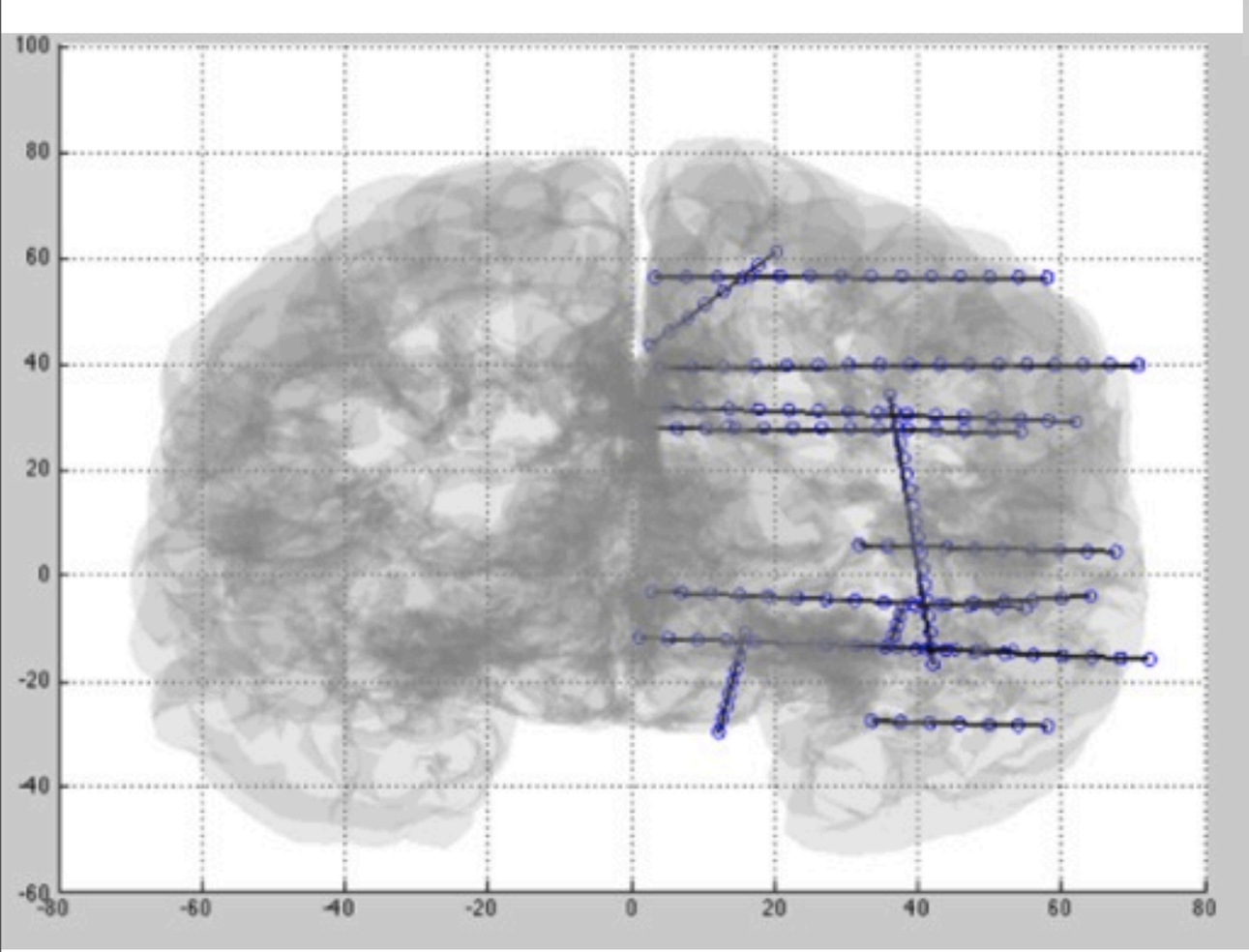
IMAGENET

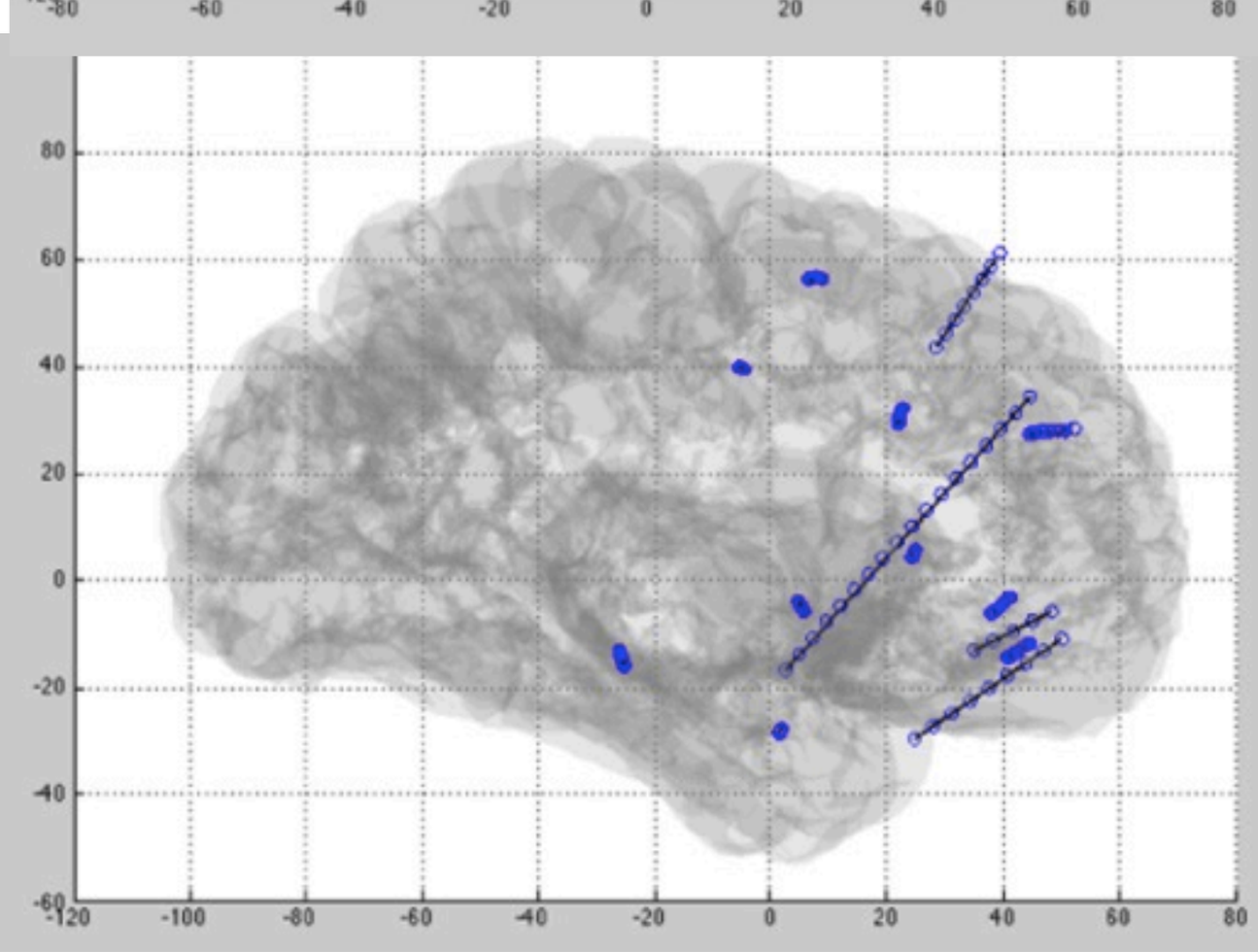

T(•)
T(•)
T(•)
T(•)
T(•)
input
retina
LGN
V1
V2
V3
LOC


3
224
224
3
7
1
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
Class labels
1000
1, 2, 3, 4
1, 2, 4
1, 2
1, 2
1, 2, 4
5, 2
5, 2
5, 6
Layer 1
Layer 2
Layer 3
Layer 4
Layer 5

109 patients (>1000 electrodes)
Hospital de Lyon (France)

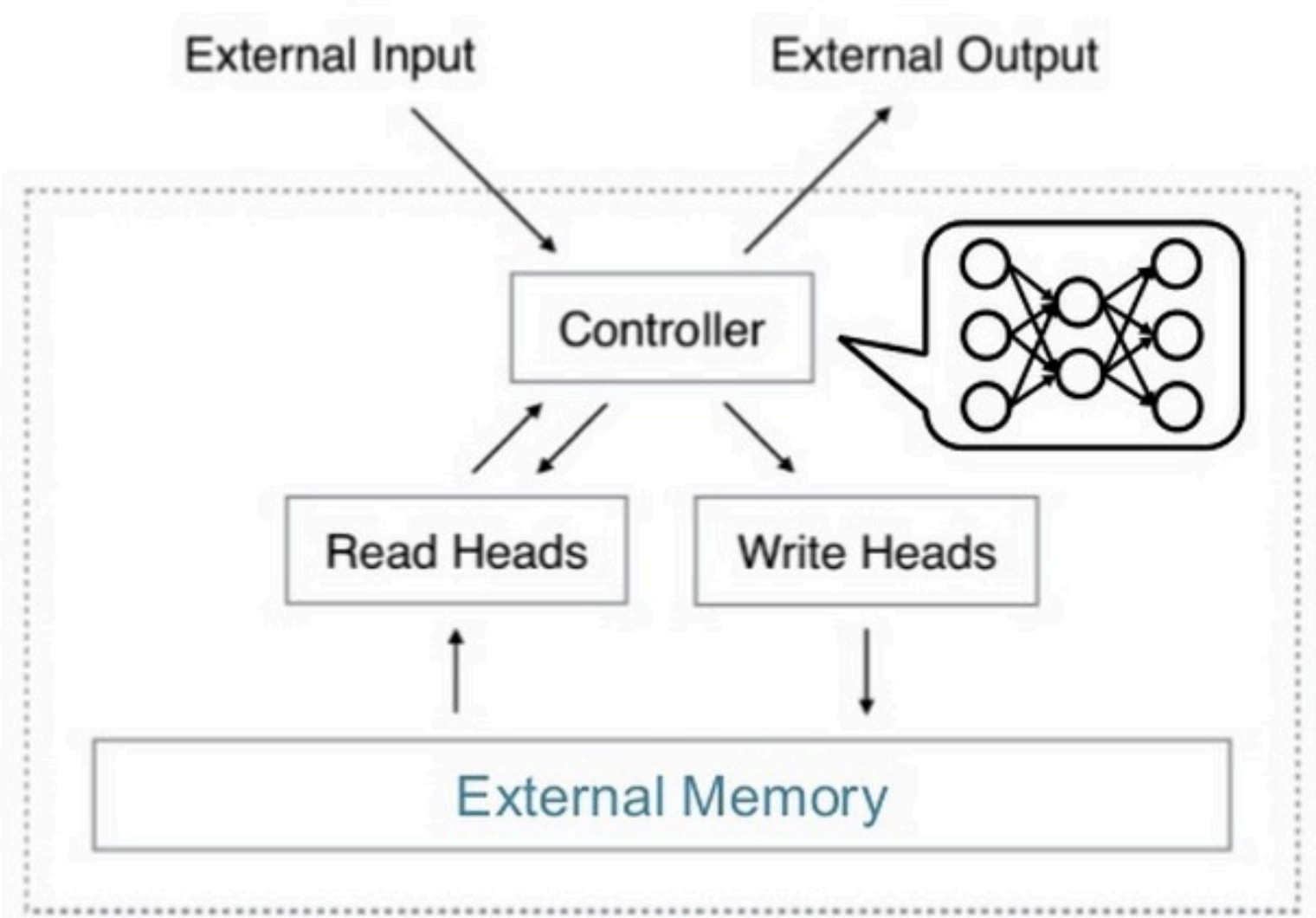




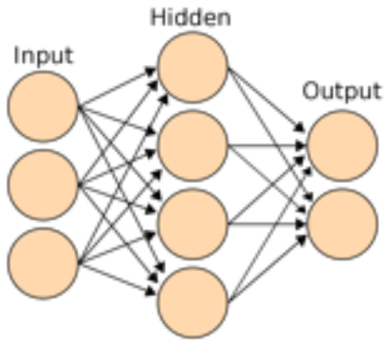


What about memory?

Neural Turing Machines learn to approximate algorithms from examples!



Take home message



- Deep Learning = Neural Networks 2.0

- Feature learning & approximate wild functions

- Pushing ML and AI to unthinkable applications

COMPUTATIONAL NEUROSCIENCE
Research group at the University of Tartu

Thanks!

Tambet

Ilya

Taivo

Kristjan

Ardi

TARTU ÜLIKOOL
UNIVERSITAS TARTUENSIS
1632