Workshop:
Machine Learning Overview
&
Neural Networks

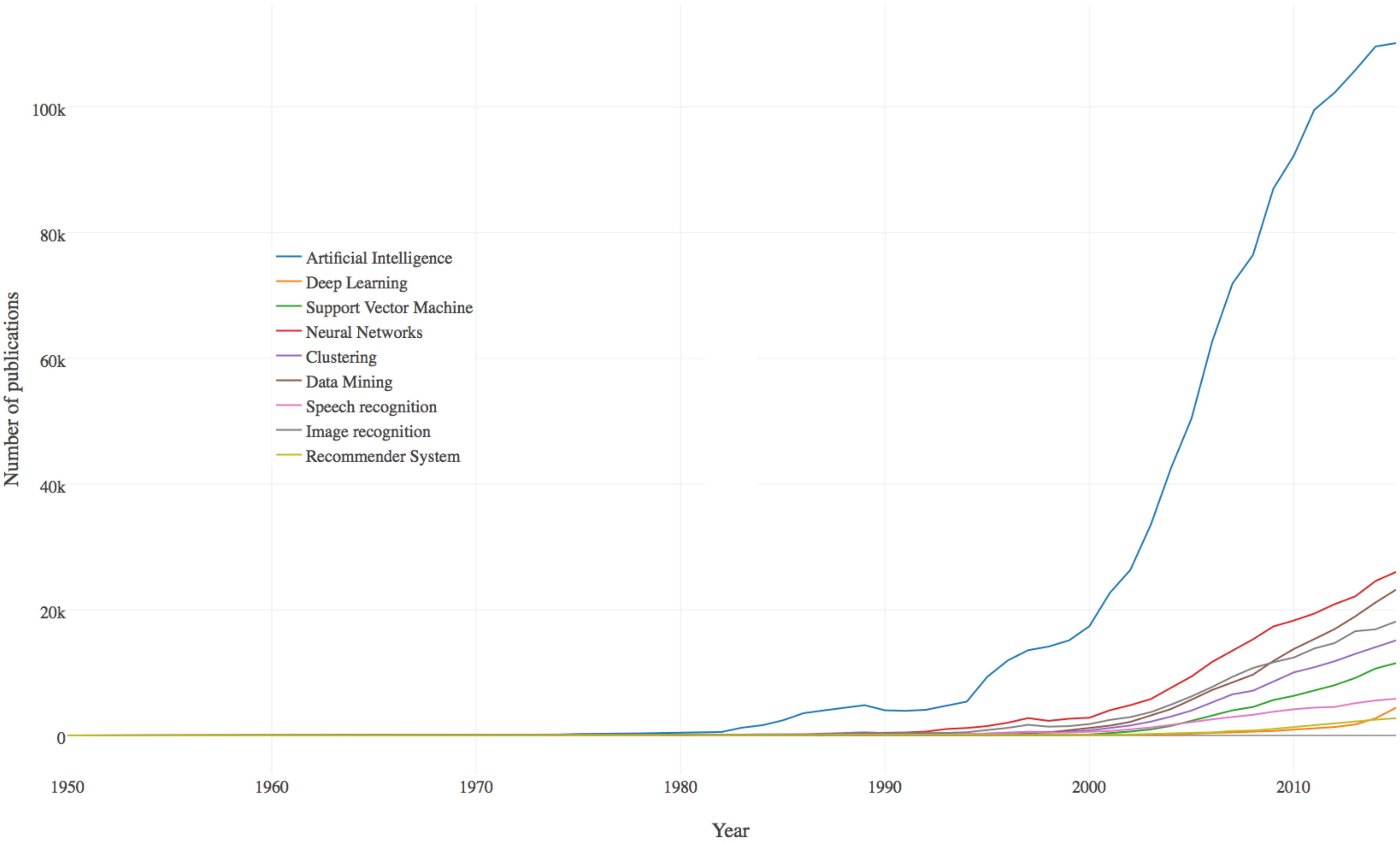
By: Denis Kazakov

For: hackNY2017

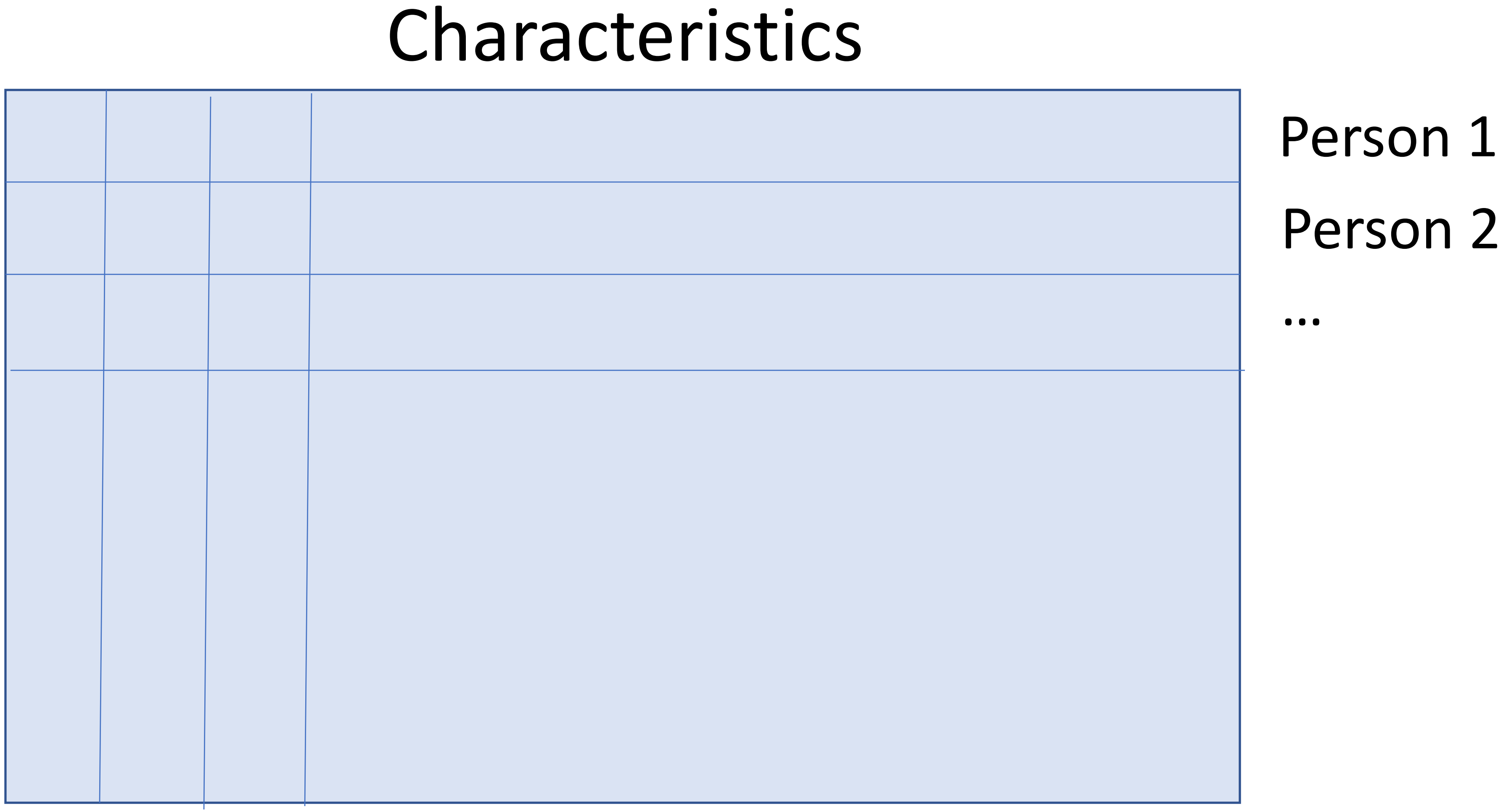
Artificial Intelligence - study of "intelligent agents": any device that perceives its environment and takes actions that maximize its chance of success at some goal

Machine Learning - gives "computers the ability to learn without being explicitly programmed." (Arthur Samuel, 1959)

Neural Networks - do tasks by considering examples, generally without task-specific programming


Artificial Intelligence
Deep Learning
Support Vector Machine
Neural Networks
Clustering
Data Mining
Speech recognition
Image recognition
Recommender System
Number of publications
0
20k
40k
60k
80k
100k
1950
1960
1970
1980
1990
2000
2010
Year

Unsupervised (descriptive)



Characteristics

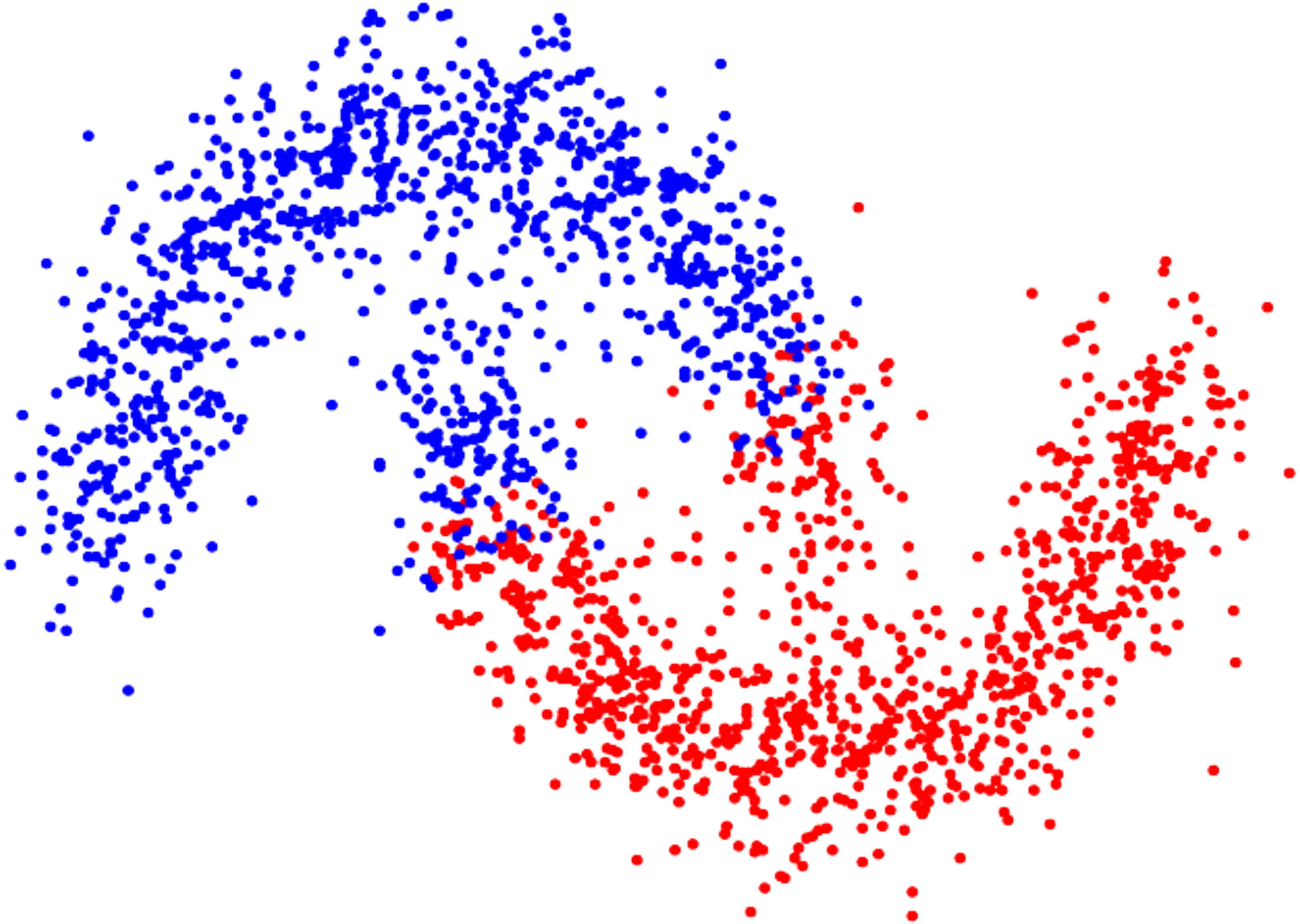
Person 1

Person 2

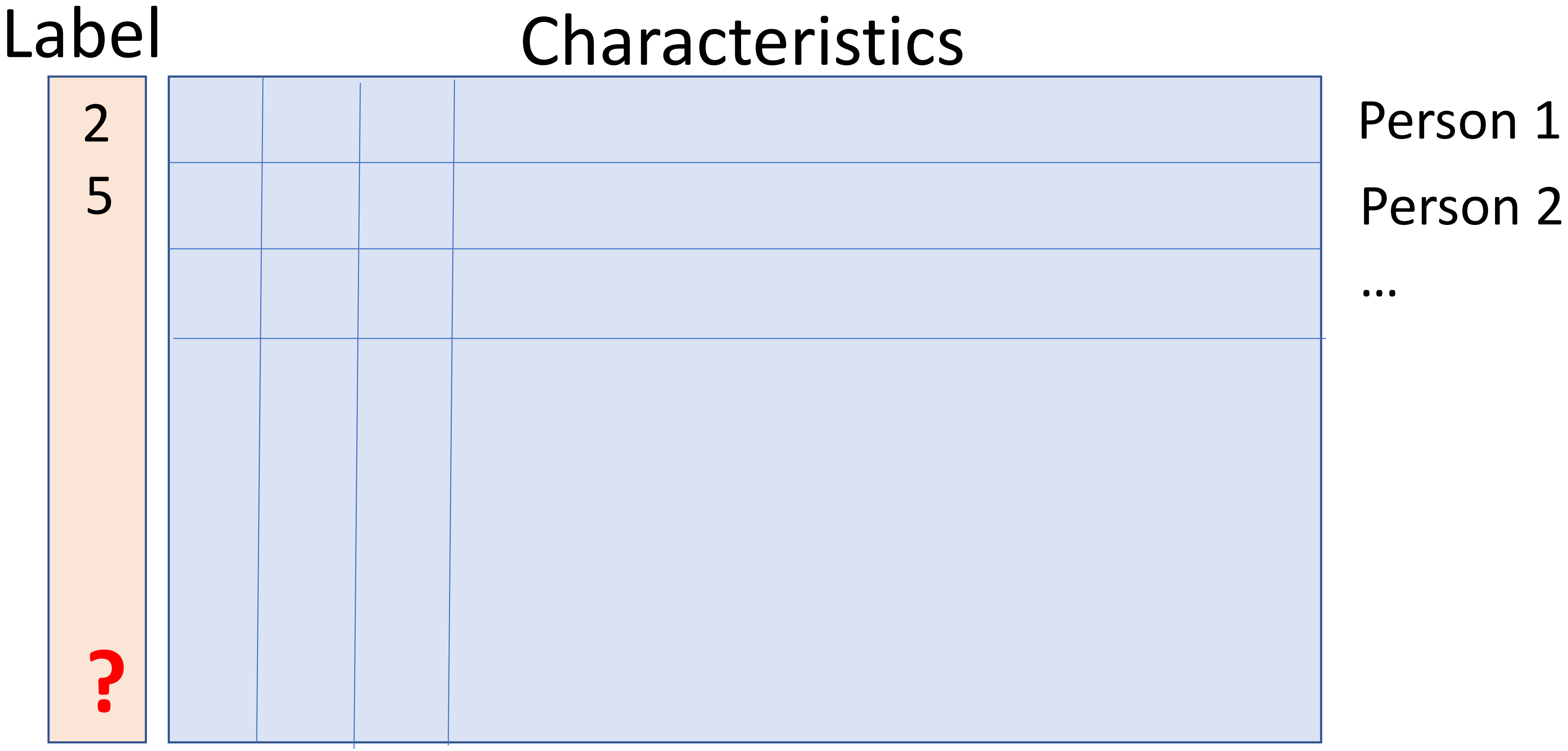
…

Define a **measure of "good"** for something

Minimize(**- measure of "good"**) w.r.t. something



Supervised (predictive)



Label

Characteristics

2

5

?

Person 1

Person 2

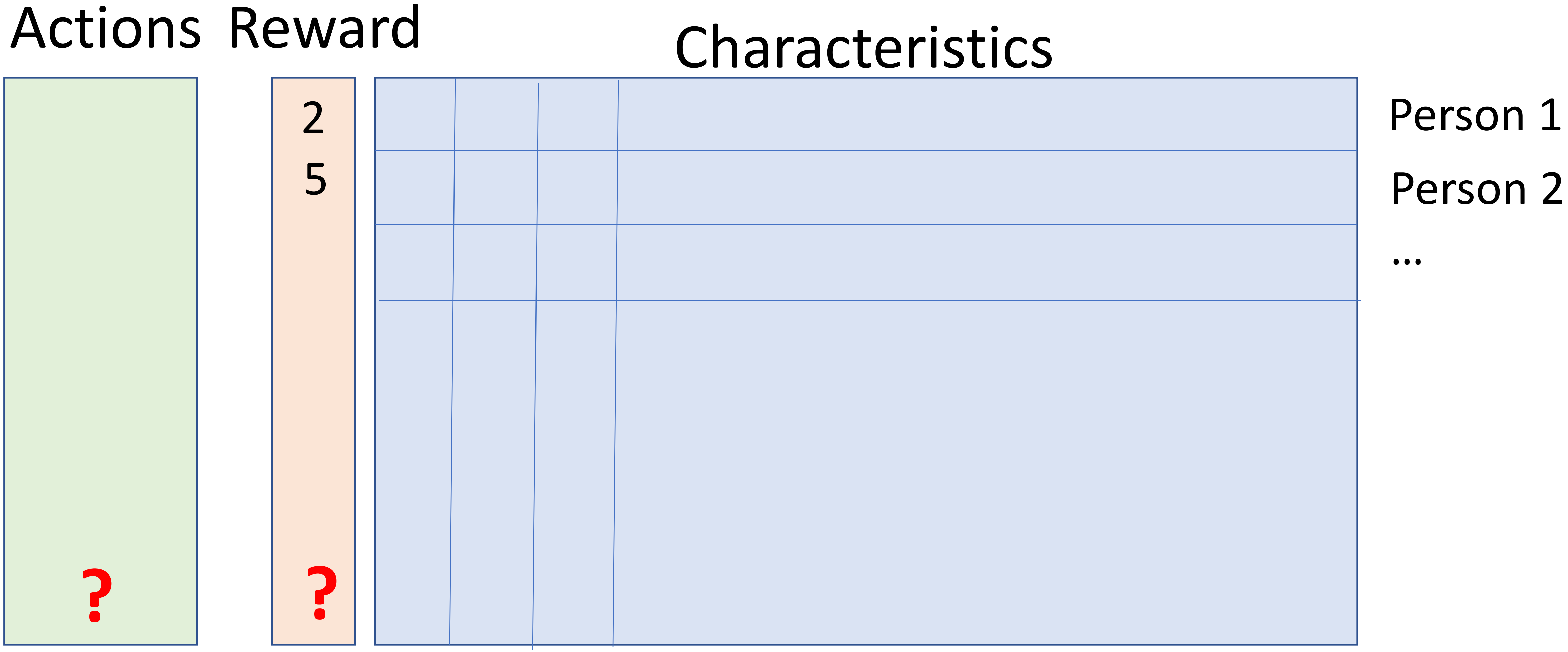
…

Predict **Label** from doing something with **Characteristics:**

Minimize(error of prediction) w.r.t. something

vs

Reinforcement (prescriptive)



Actions

Reward

Characteristics

2

5

?

?

Person 1

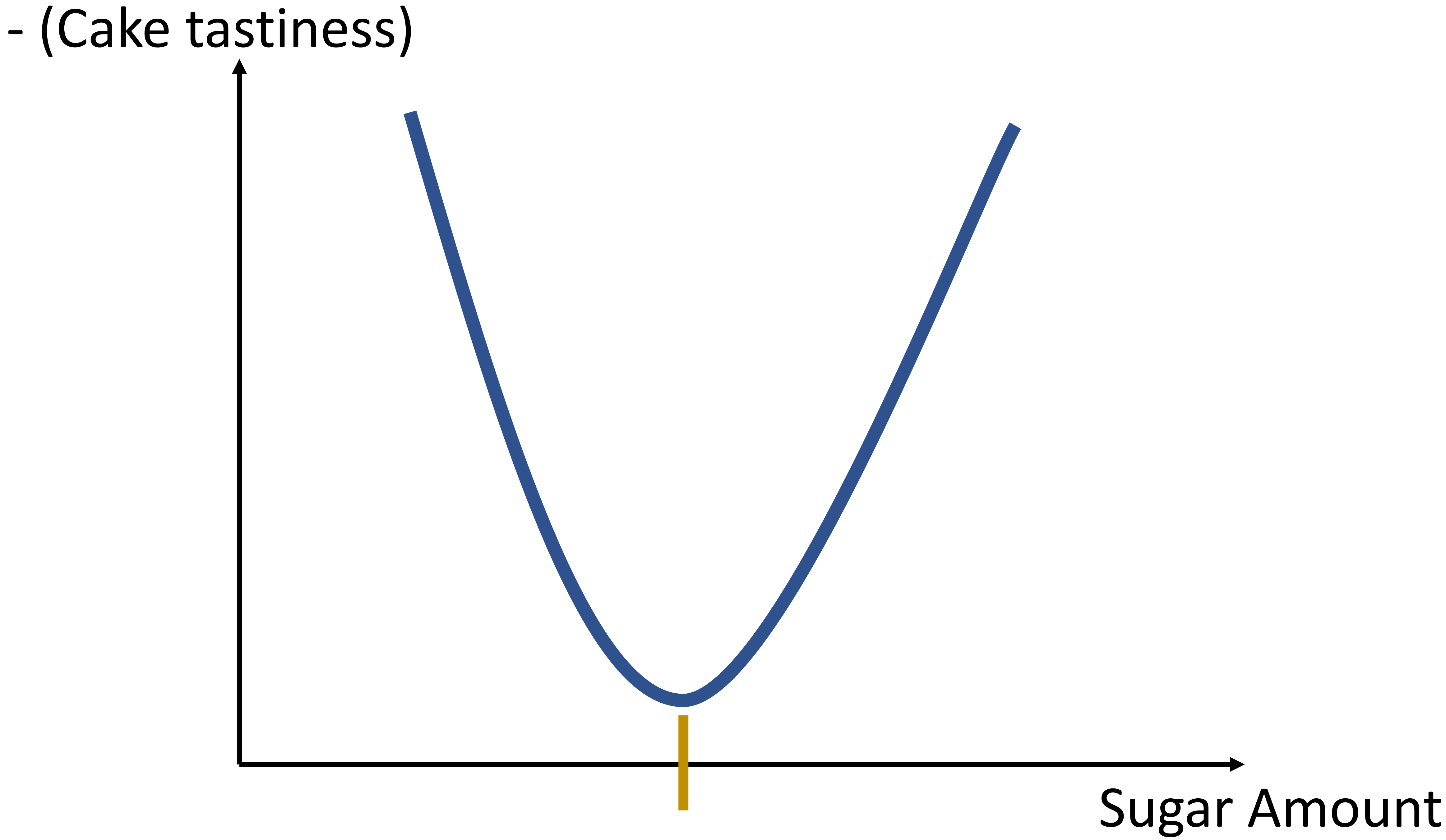
Person 2

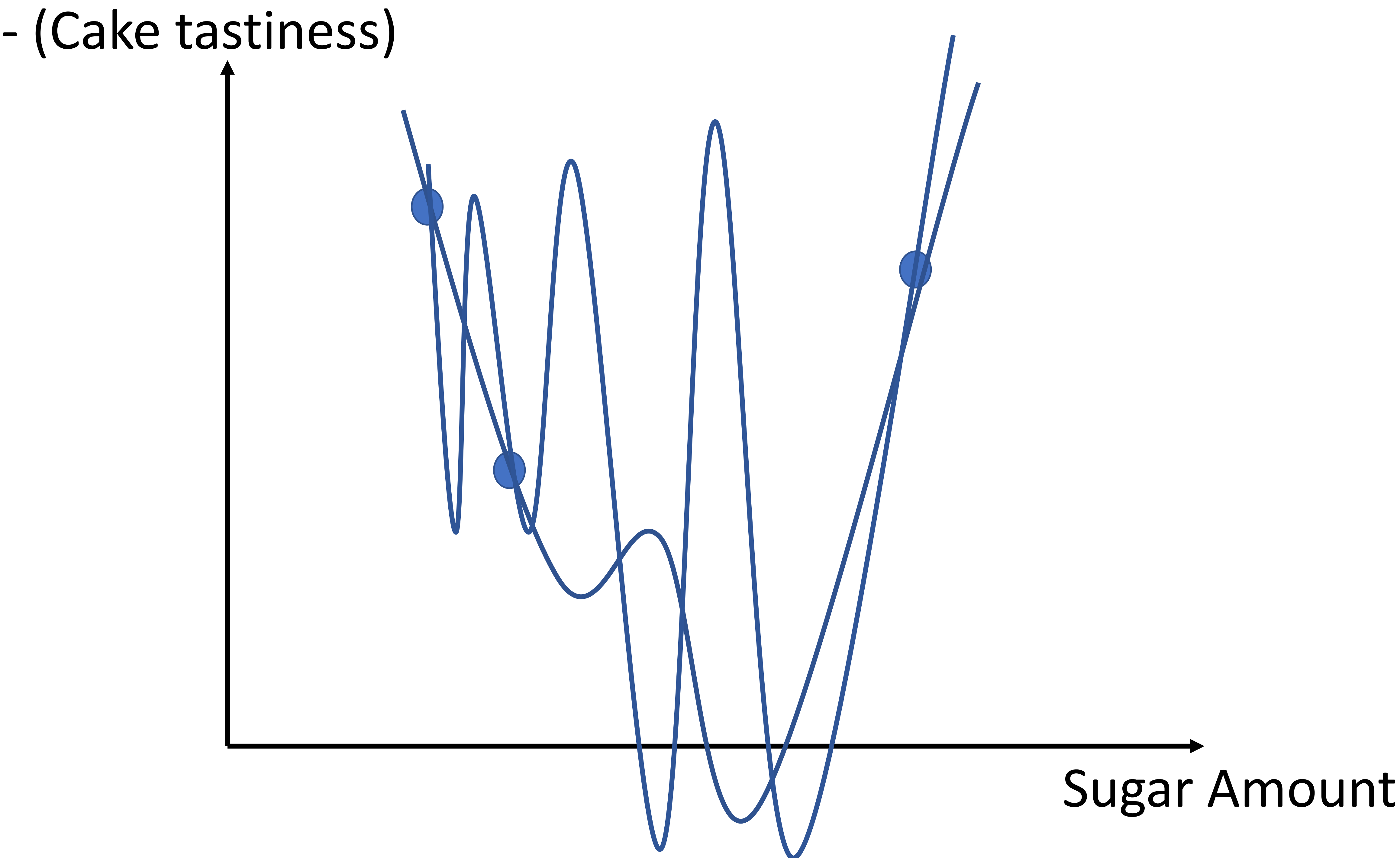
…

Choose **Actions** from doing something with **Characteristics** to get highest **Reward**:

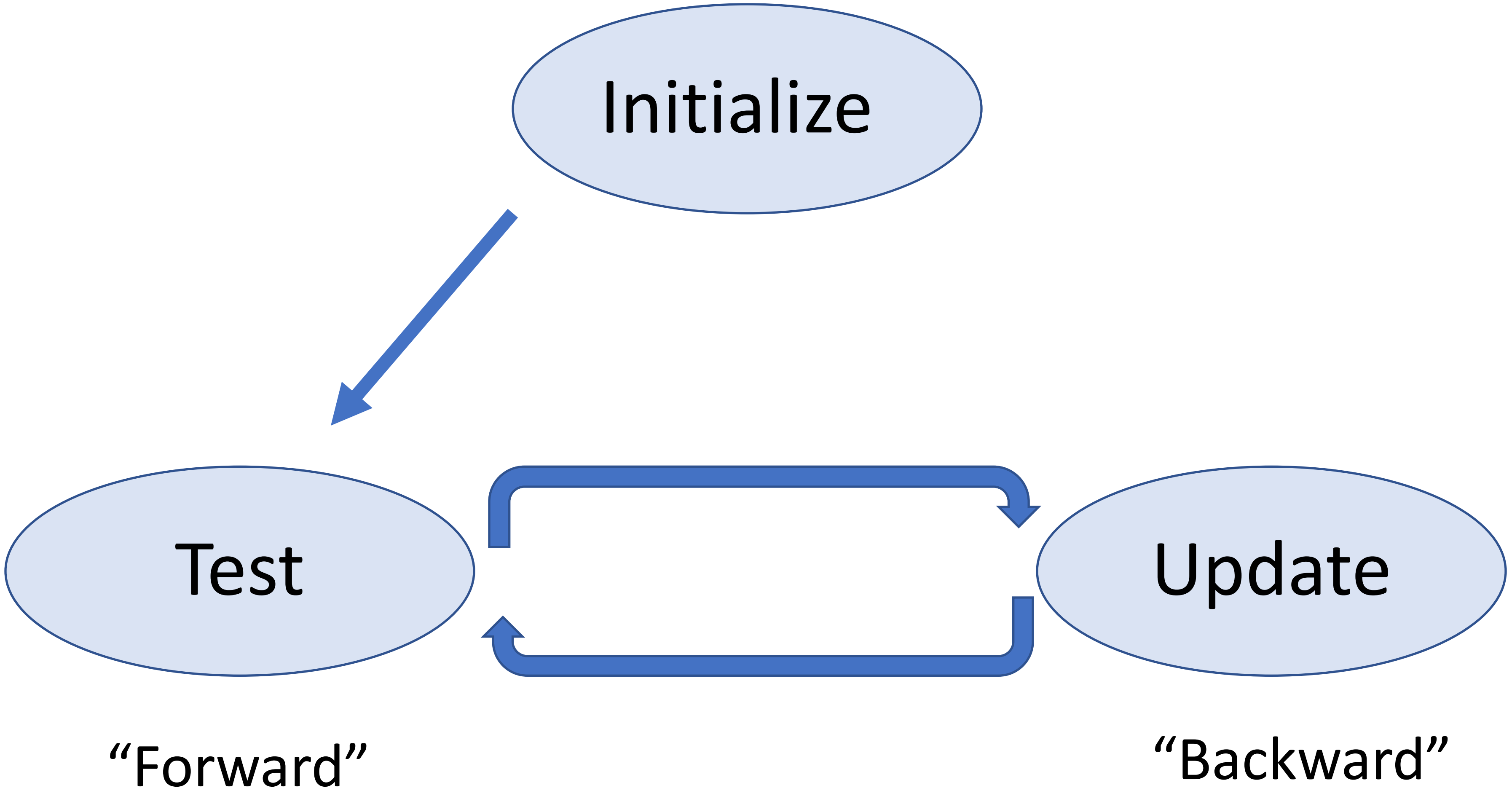
Minimize(- Reward) w.r.t. something

- gait


- (Cake tastiness)
Sugar Amount


- (Cake tastiness)
Sugar Amount

For each ML model:

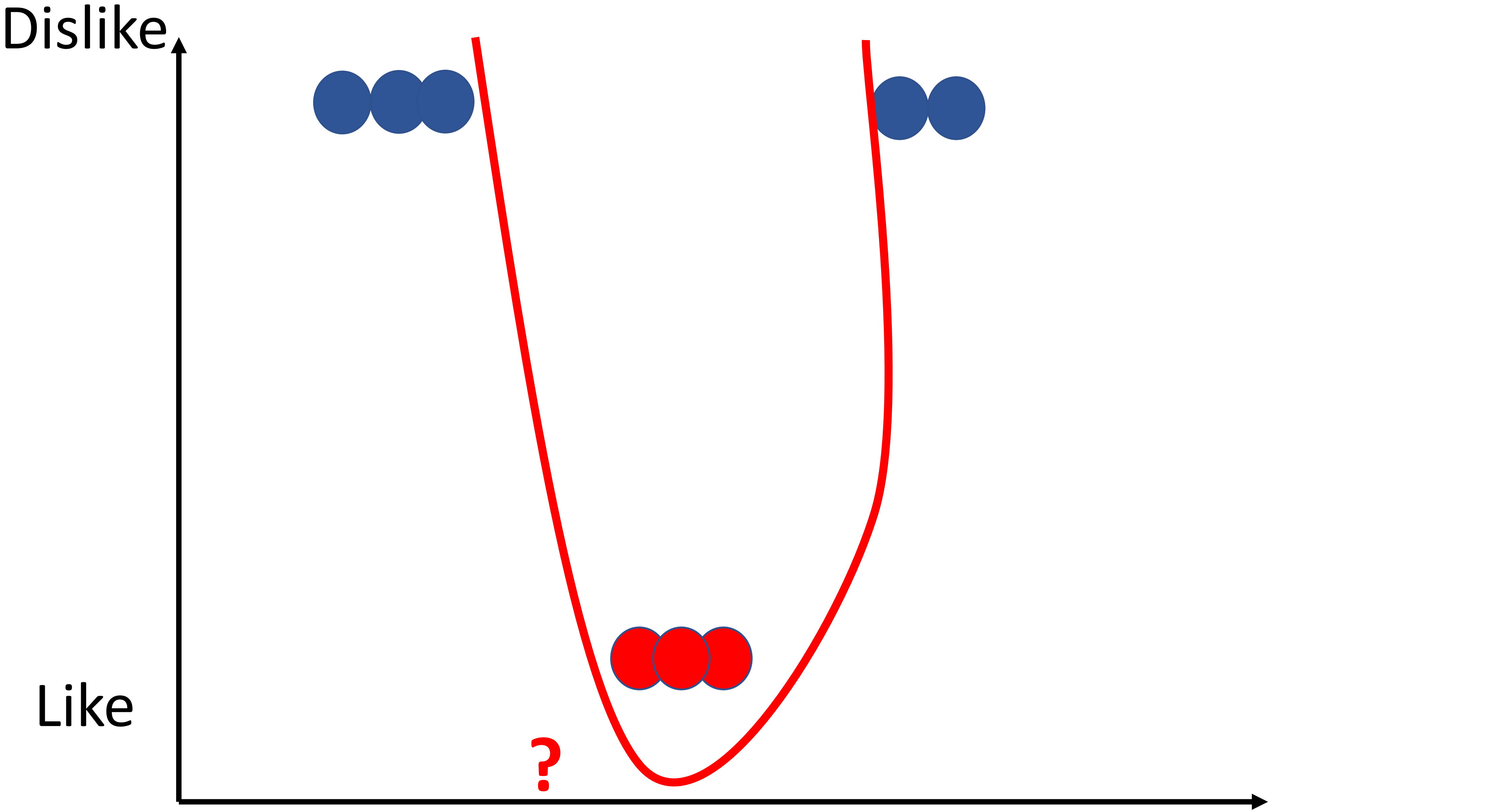

Initialize

Test

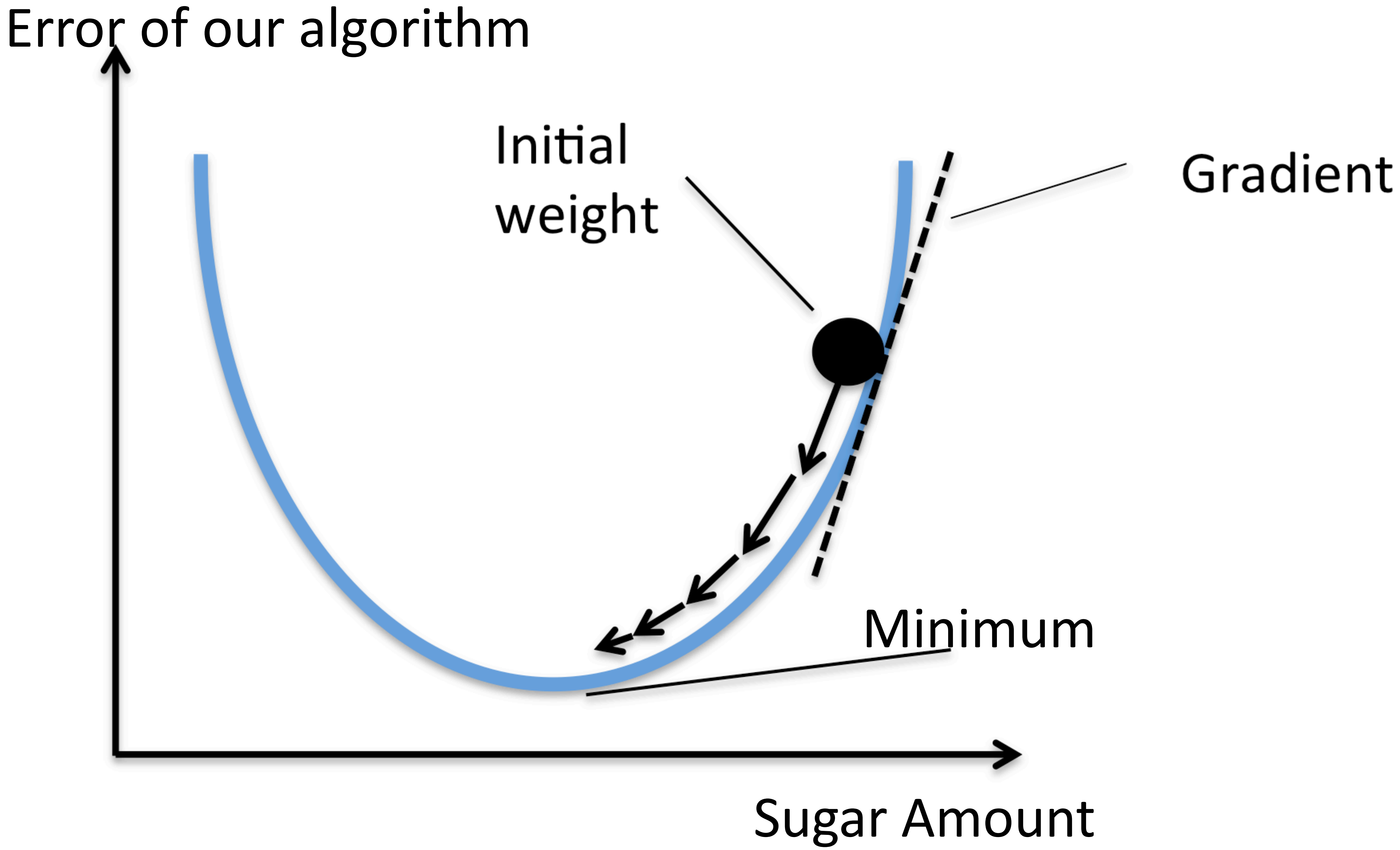
Update

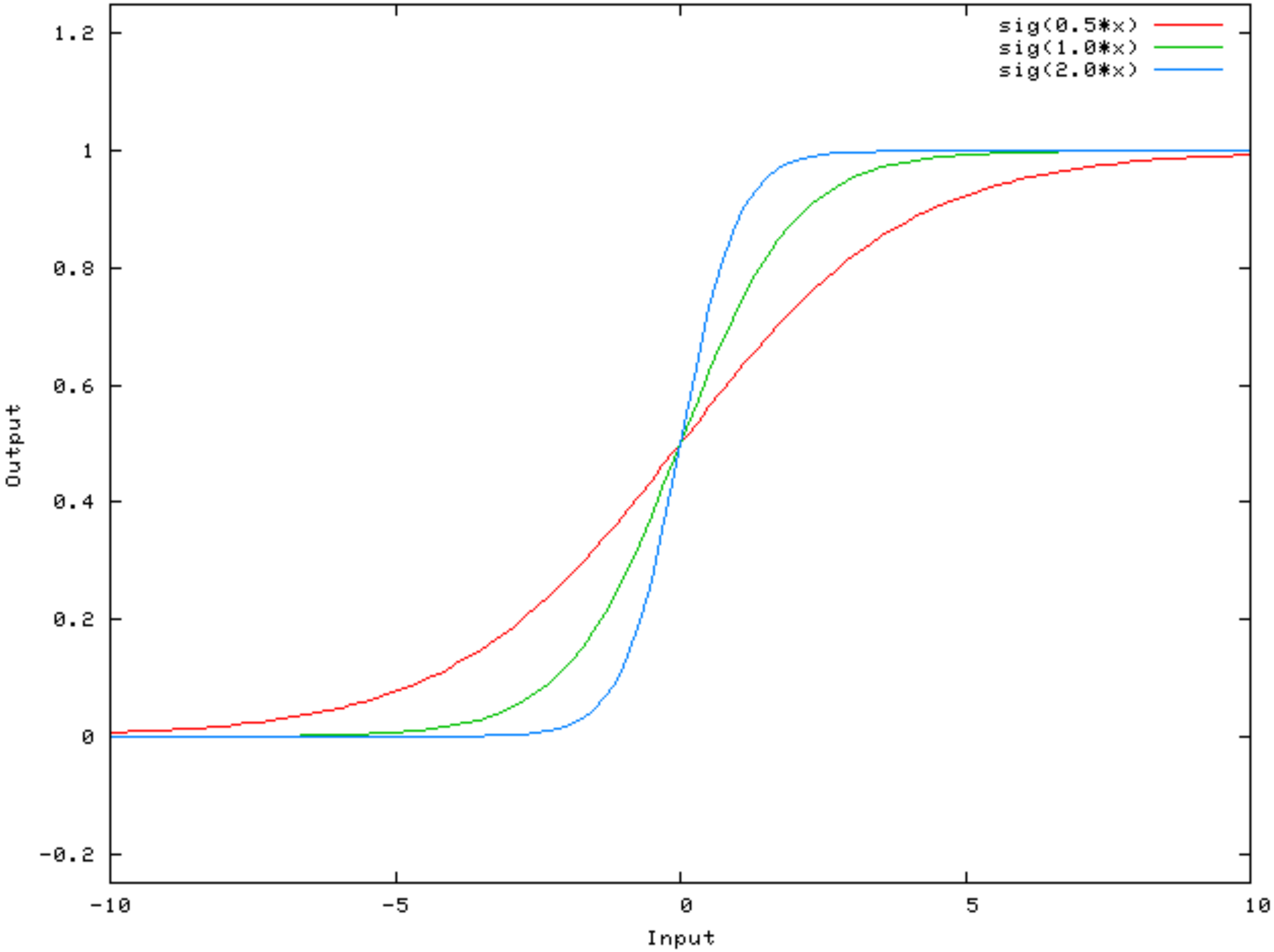
“Forward”

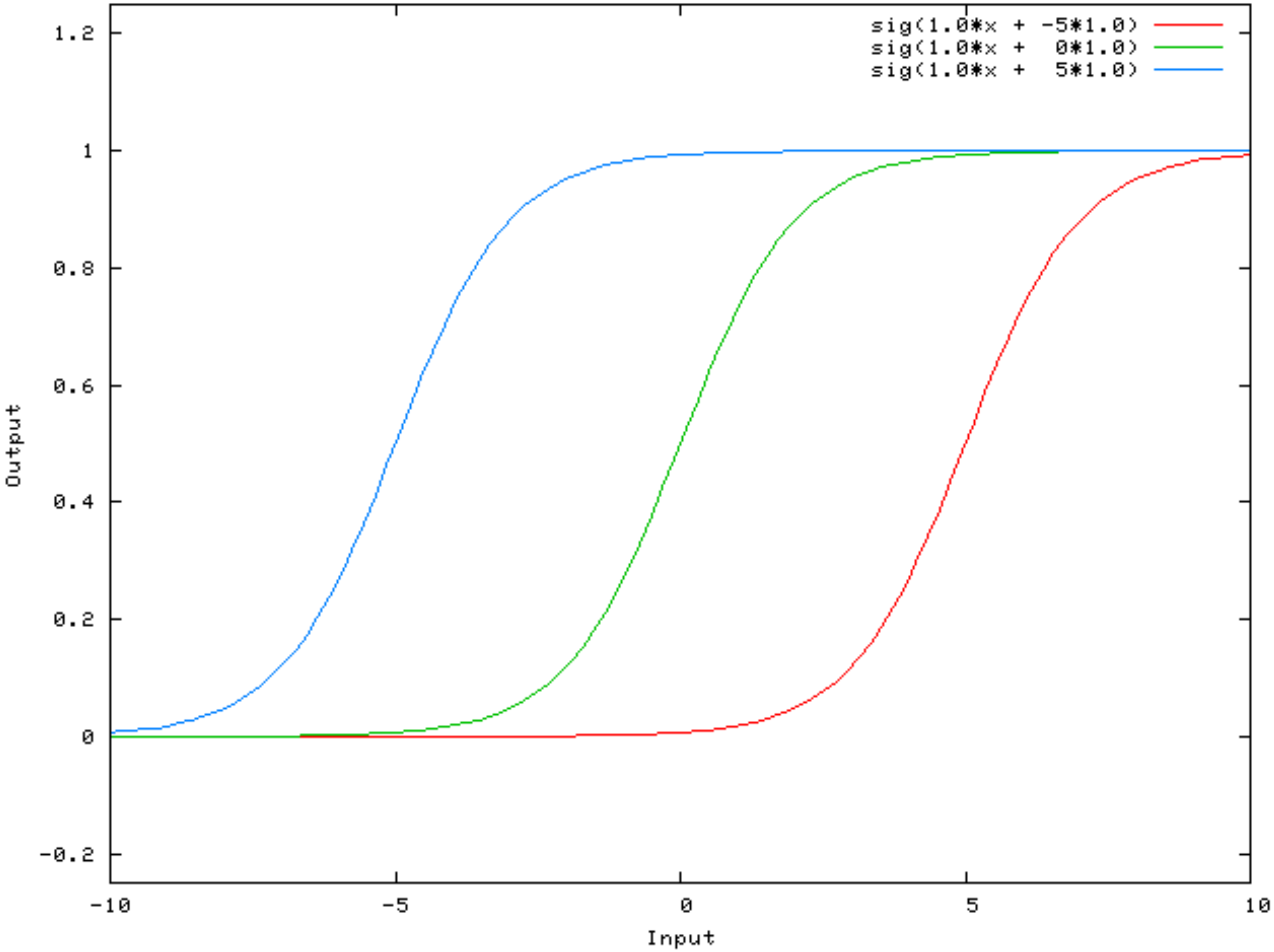
“Backward”



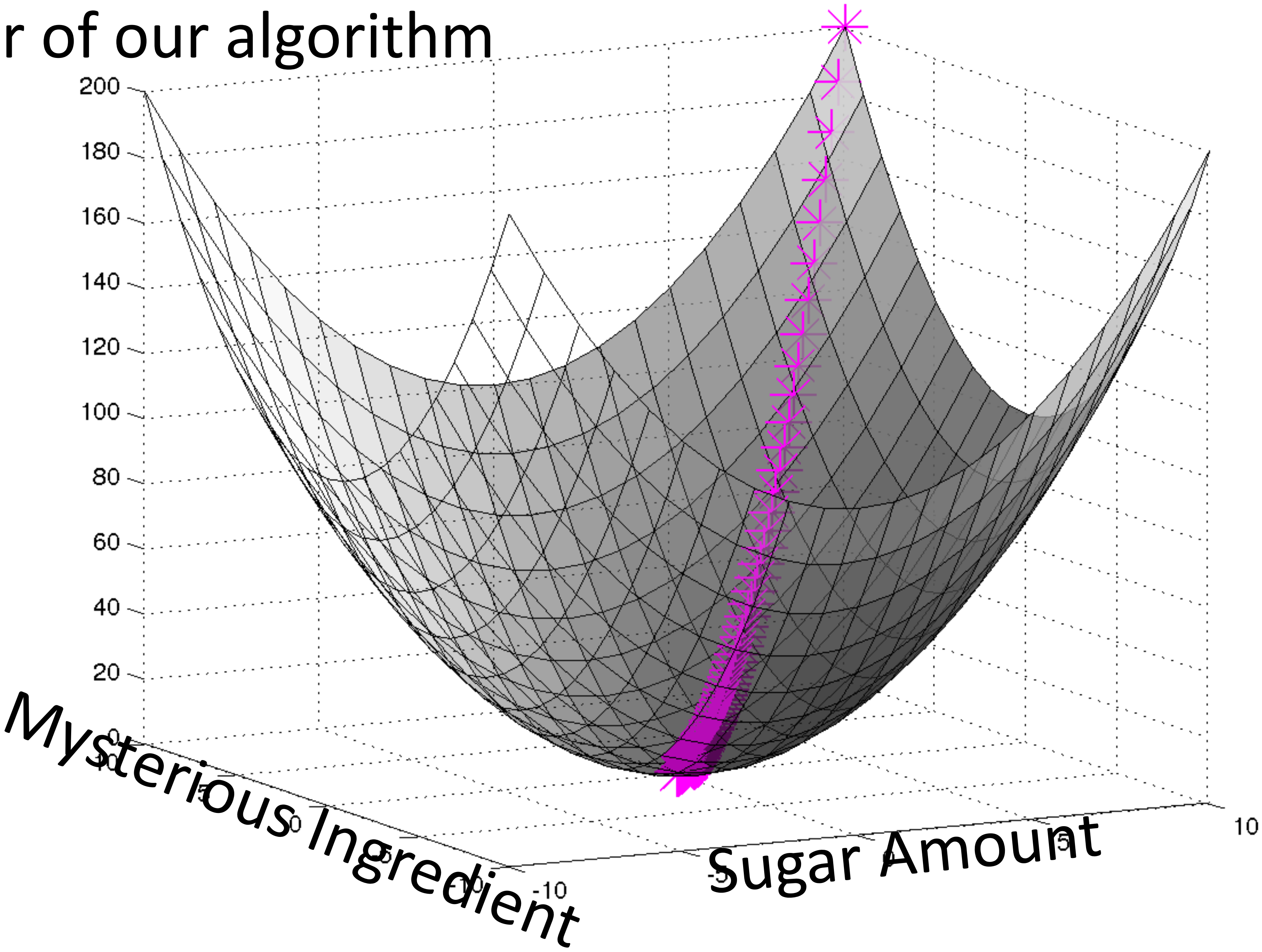
Minimize(error of prediction) w.r.t. something


Error of our algorithm
Initial weight
Gradient
Minimum
Sugar Amount


sig(0.5*x)
sig(1.0*x)
sig(2.0*x)
1.2
1
0.8
0.6
0.4
0.2
0
-0.2
Output
-10
-5
0
5
10
Input

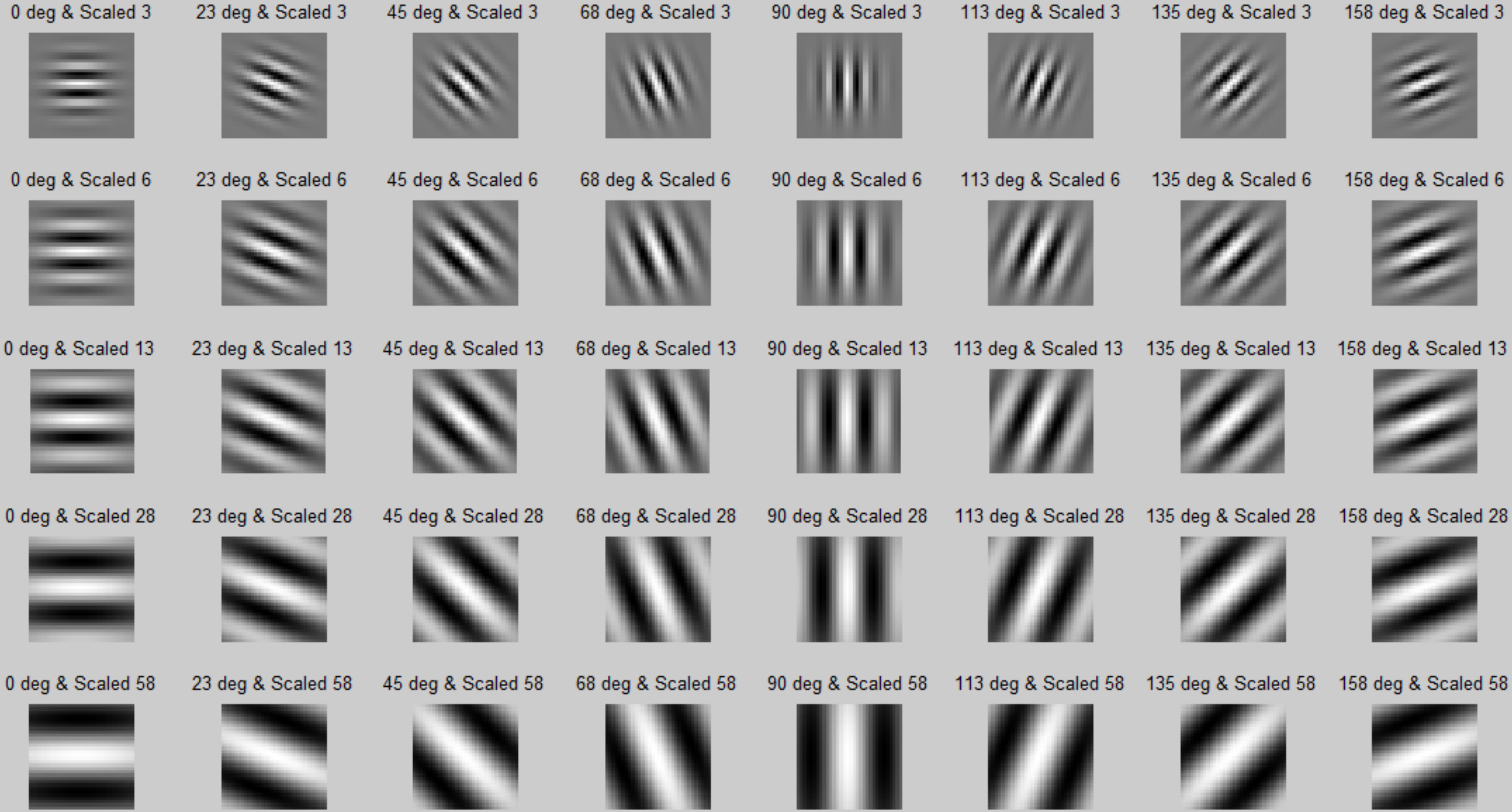

sig(1.0*x + -5*1.0)
sig(1.0*x + 0*1.0)
sig(1.0*x + 5*1.0)
1.2
1
0.8
0.6
0.4
0.2
0
-0.2
Output
-10
-5
0
5
10
Input

Error of our algorithm



- Training animation (people who liked the cake vs who didn’t)

VS



0 deg & Scaled 3 | 23 deg & Scaled 3 | 45 deg & Scaled 3 | 68 deg & Scaled 3 | 90 deg & Scaled 3 | 113 deg & Scaled 3 | 135 deg & Scaled 3 | 158 deg & Scaled 3

0 deg & Scaled 6 | 23 deg & Scaled 6 | 45 deg & Scaled 6 | 68 deg & Scaled 6 | 90 deg & Scaled 6 | 113 deg & Scaled 6 | 135 deg & Scaled 6 | 158 deg & Scaled 6

0 deg & Scaled 13 | 23 deg & Scaled 13 | 45 deg & Scaled 13 | 68 deg & Scaled 13 | 90 deg & Scaled 13 | 113 deg & Scaled 13 | 135 deg & Scaled 13 | 158 deg & Scaled 13

0 deg & Scaled 28 | 23 deg & Scaled 28 | 45 deg & Scaled 28 | 68 deg & Scaled 28 | 90 deg & Scaled 28 | 113 deg & Scaled 28 | 135 deg & Scaled 28 | 158 deg & Scaled 28

0 deg & Scaled 58 | 23 deg & Scaled 58 | 45 deg & Scaled 58 | 68 deg & Scaled 58 | 90 deg & Scaled 58 | 113 deg & Scaled 58 | 135 deg & Scaled 58 | 158 deg & Scaled 58

Unsupervised Feature Learning

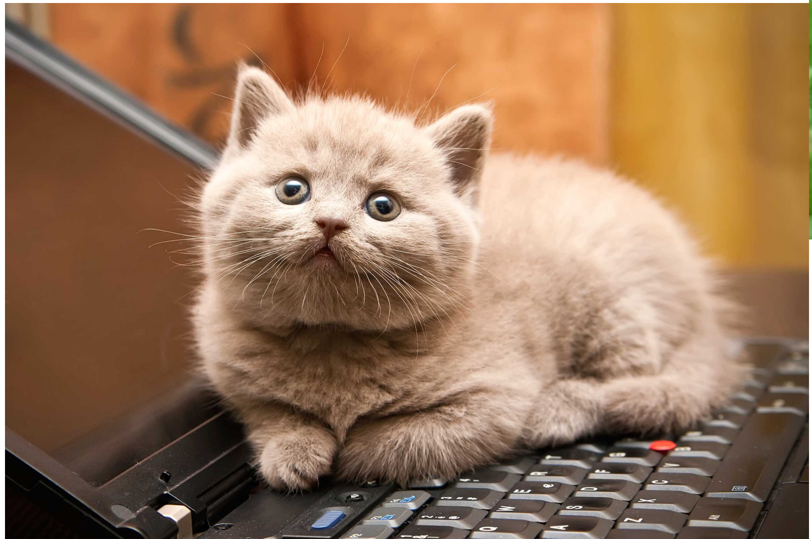
Unsupervised:

Define a **measure of “good”** for something

Minimize(**- measure of “good”**) w.r.t. something

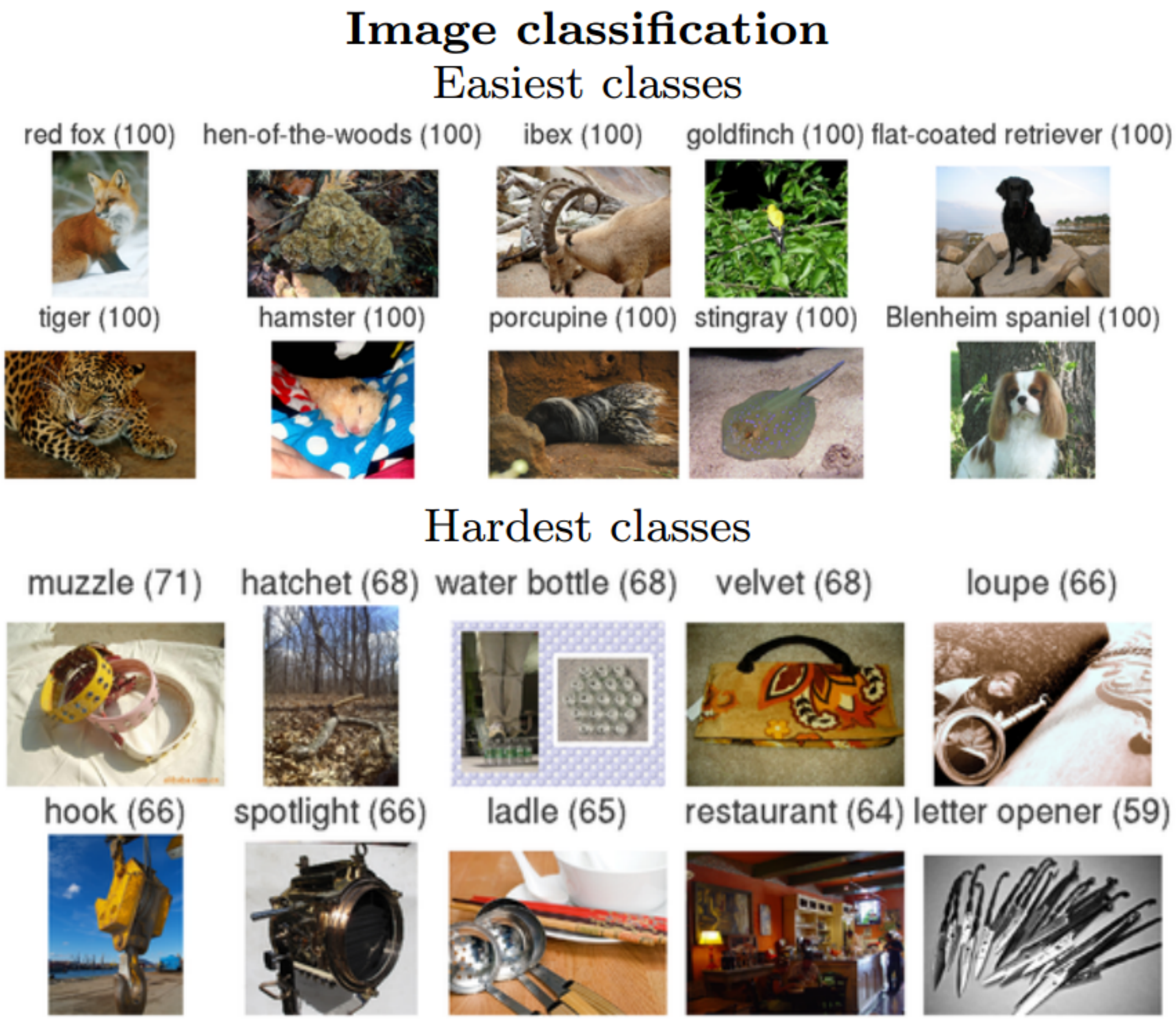
Something – weights of the neural net and its **features**

measure of “good” – performance on some task (labeled or reinforcement)

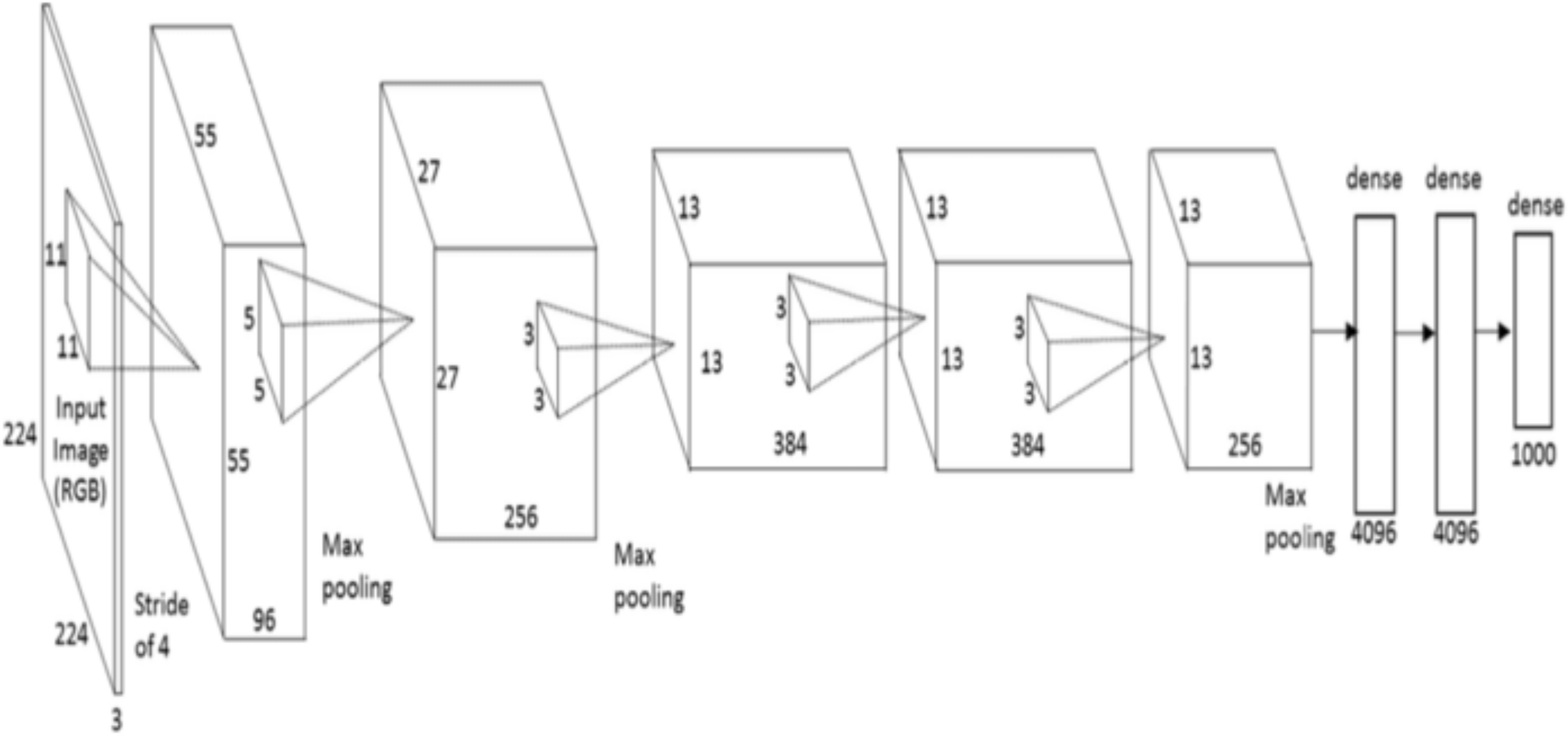


Computer Vision (ILSVRC)

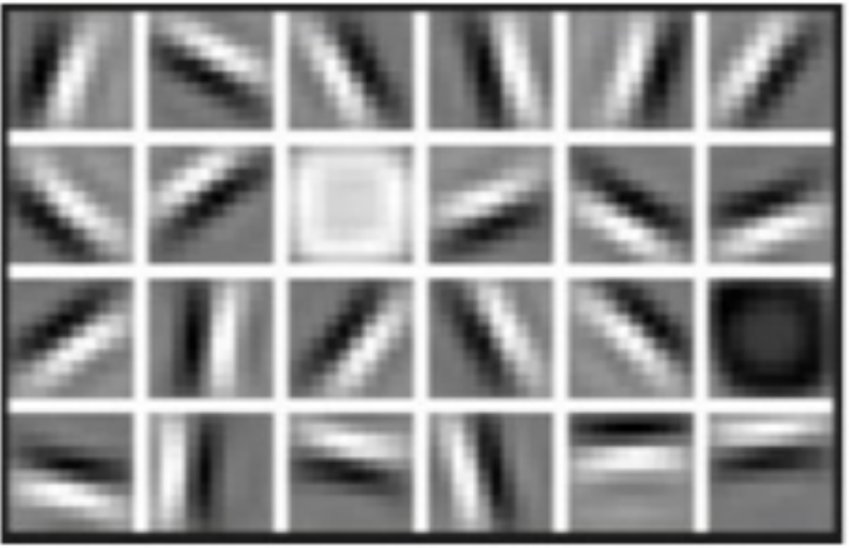
- 2010 - 28.2%
- 2011 - 25.8%
- 2012 - 16.4% (the 2nd best entry had an error rate of 26.2%).
- Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton



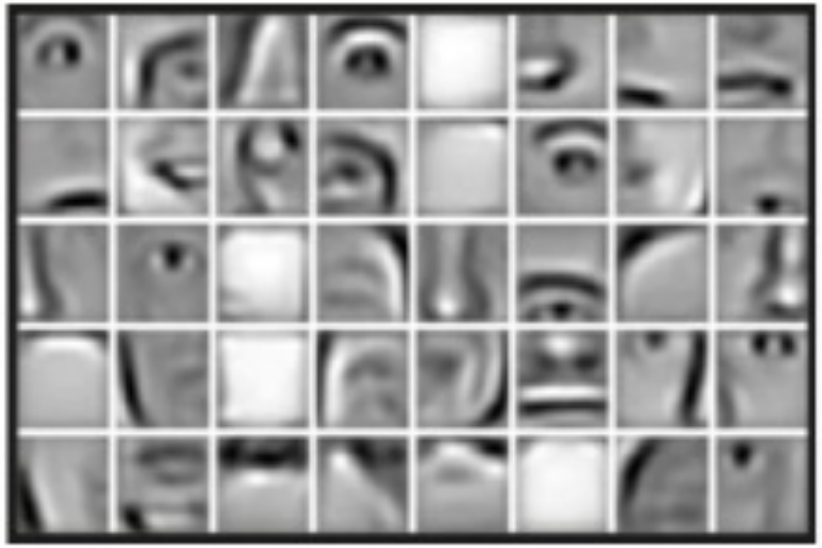
AlexNet



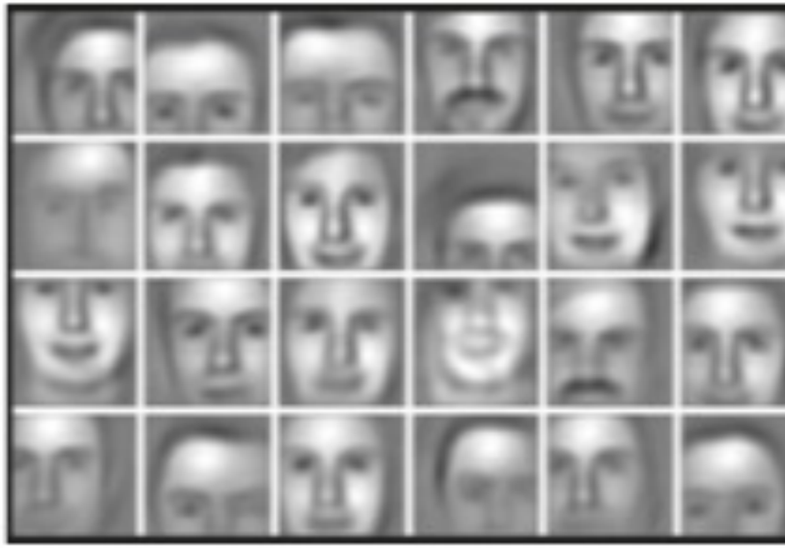
224
224
3
11
11
Input Image (RGB)
Stride of 4
55
55
96
5
5
Max pooling
27
27
256
3
3
Max pooling
13
13
384
3
3
13
13
384
3
3
13
13
256
Max pooling
dense
dense
dense
4096
4096
1000



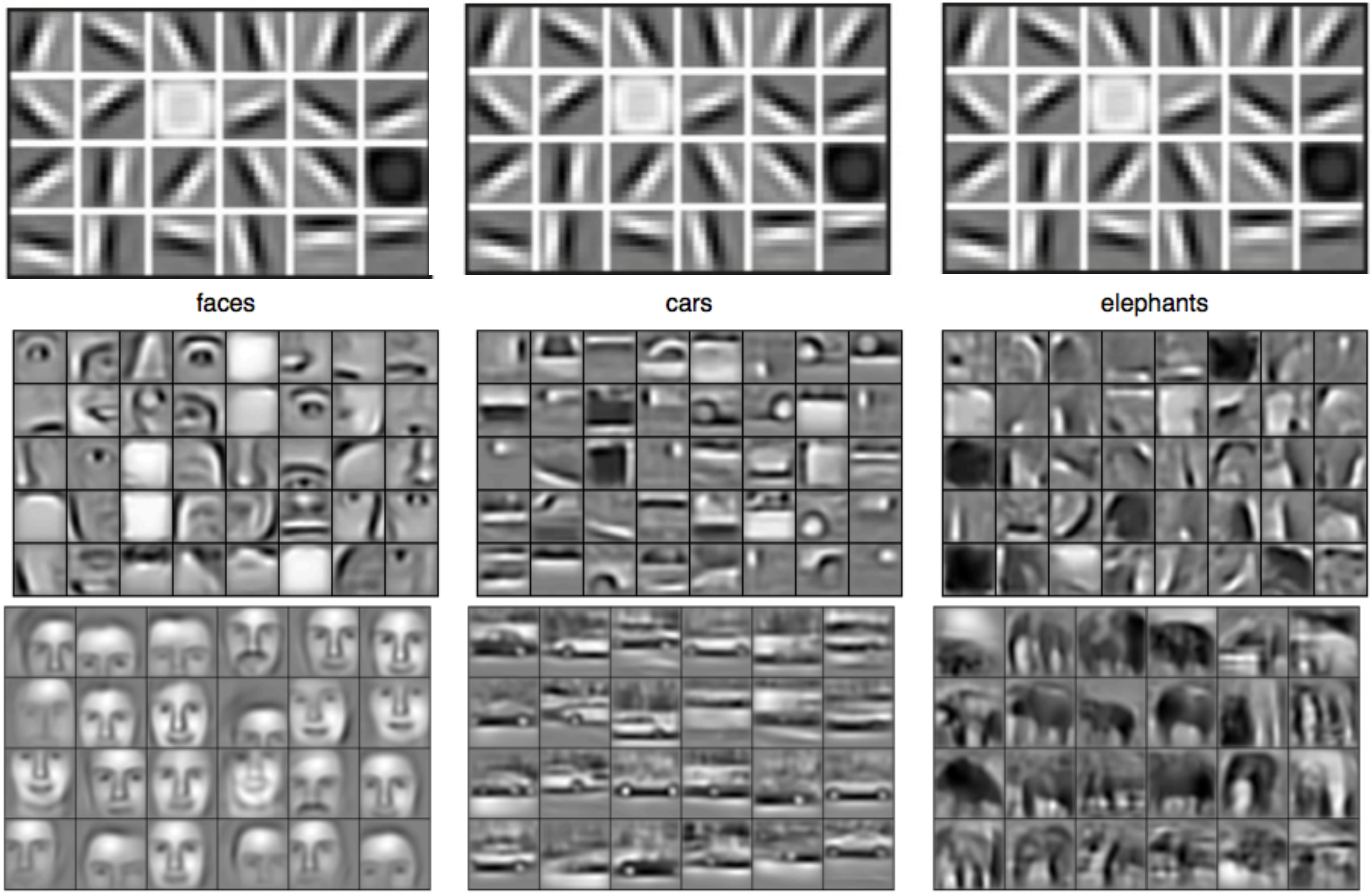
First Layer Representation



Second Layer Representation



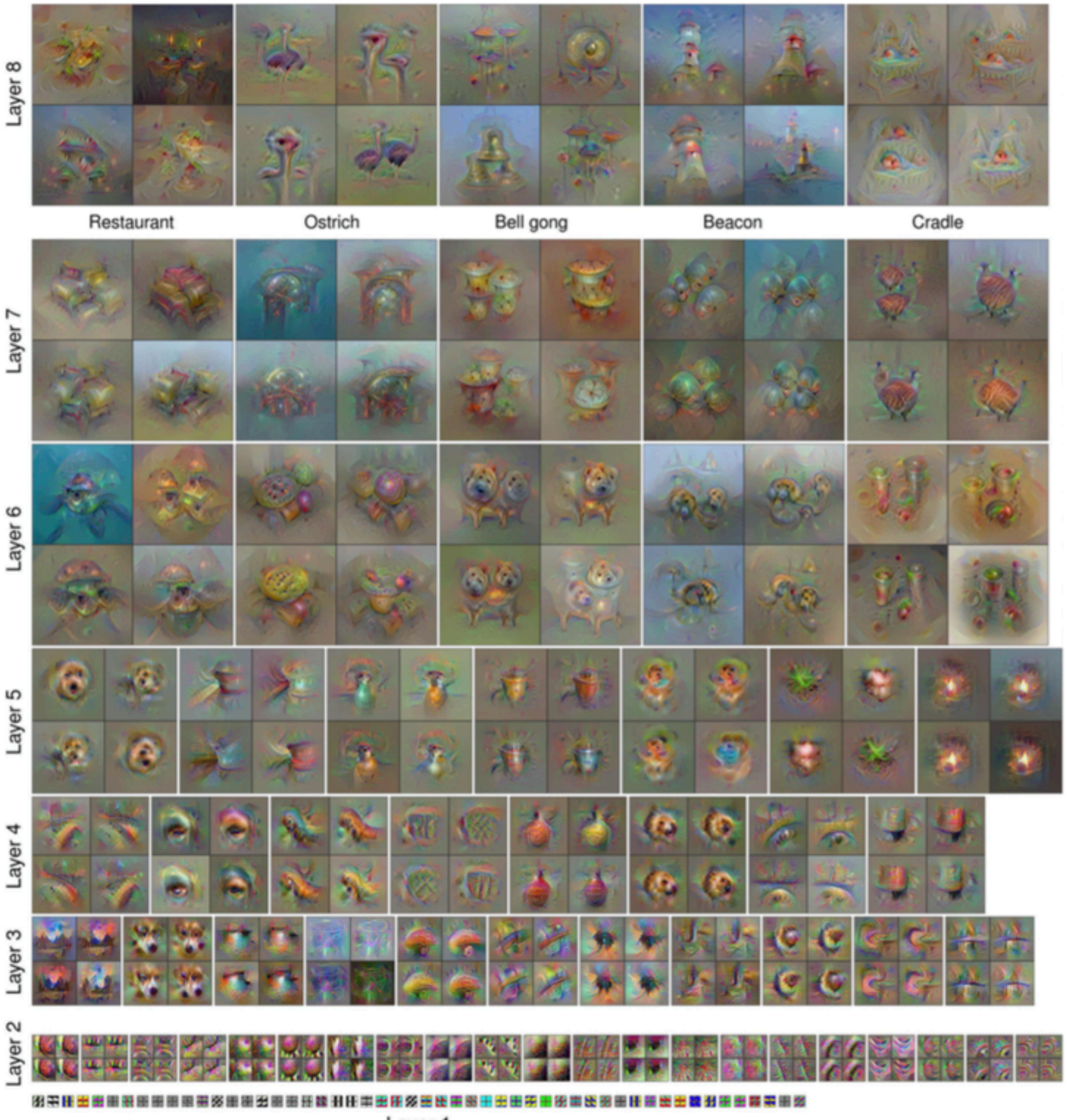
Third Layer Representation

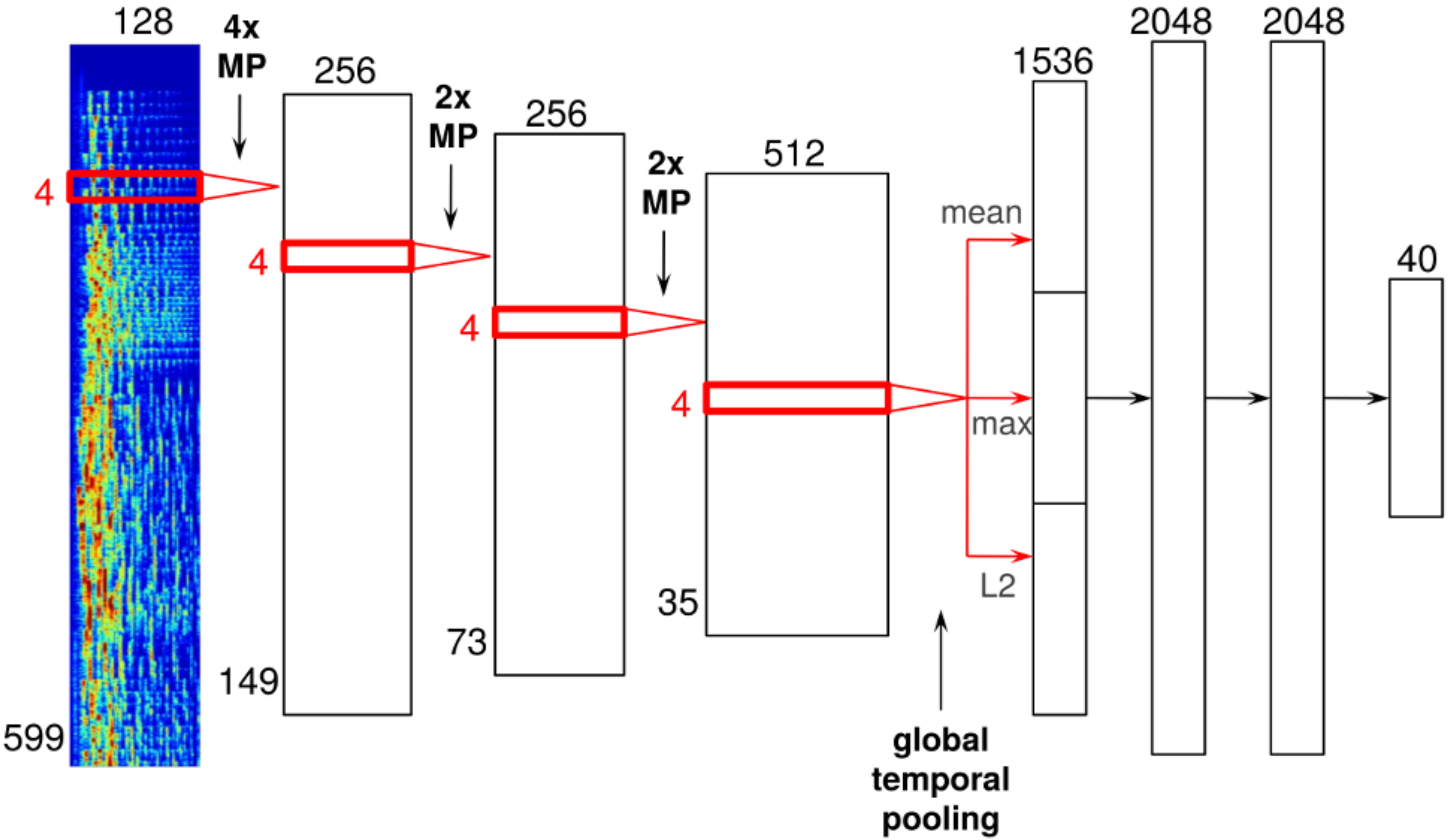


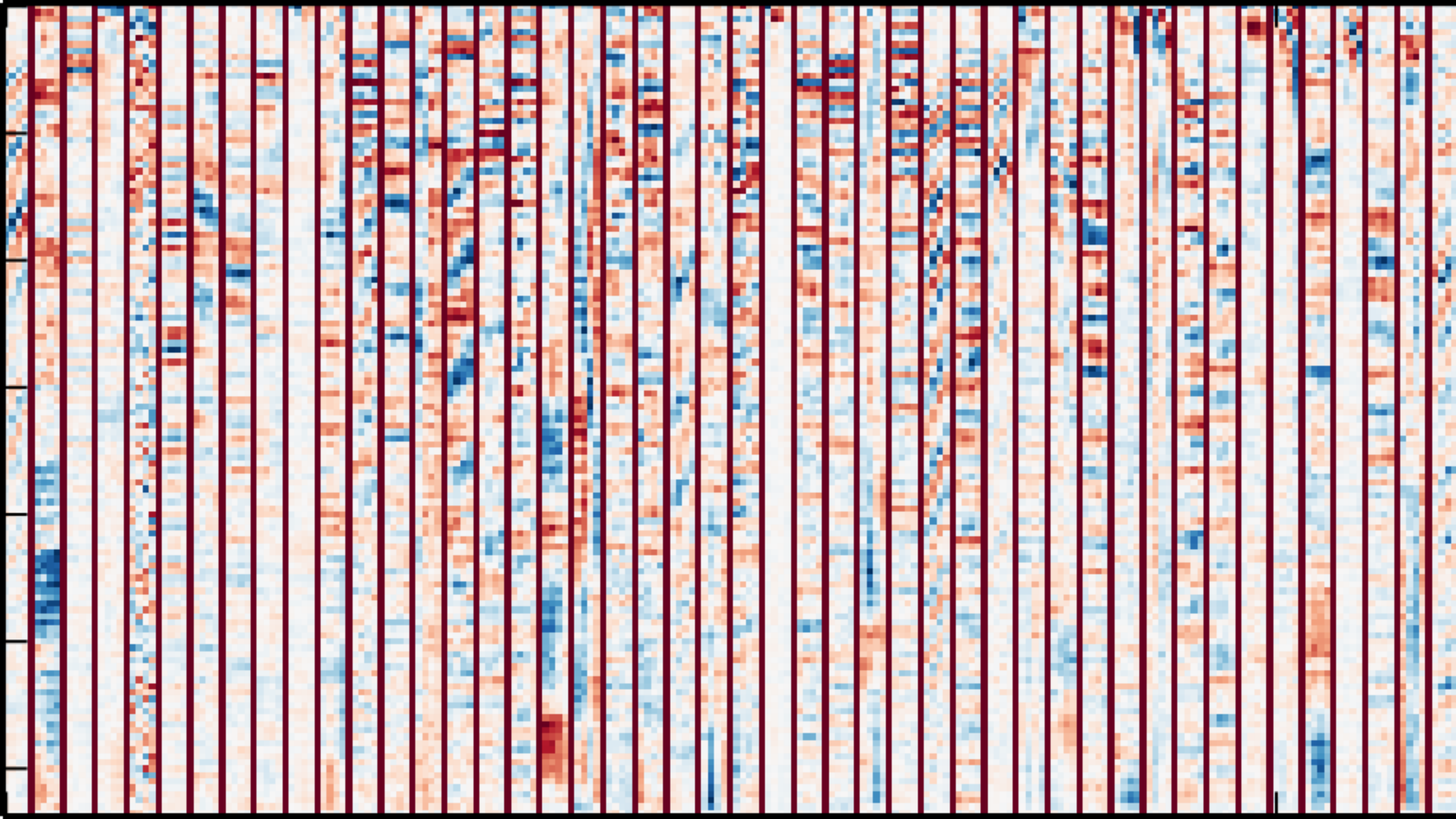
faces

cars

elephants


Layer 8
Restaurant
Ostrich
Bell gong
Beacon
Cradle
Layer 7
Layer 6
Layer 5
Layer 4
Layer 3
Layer 2
Layer 1

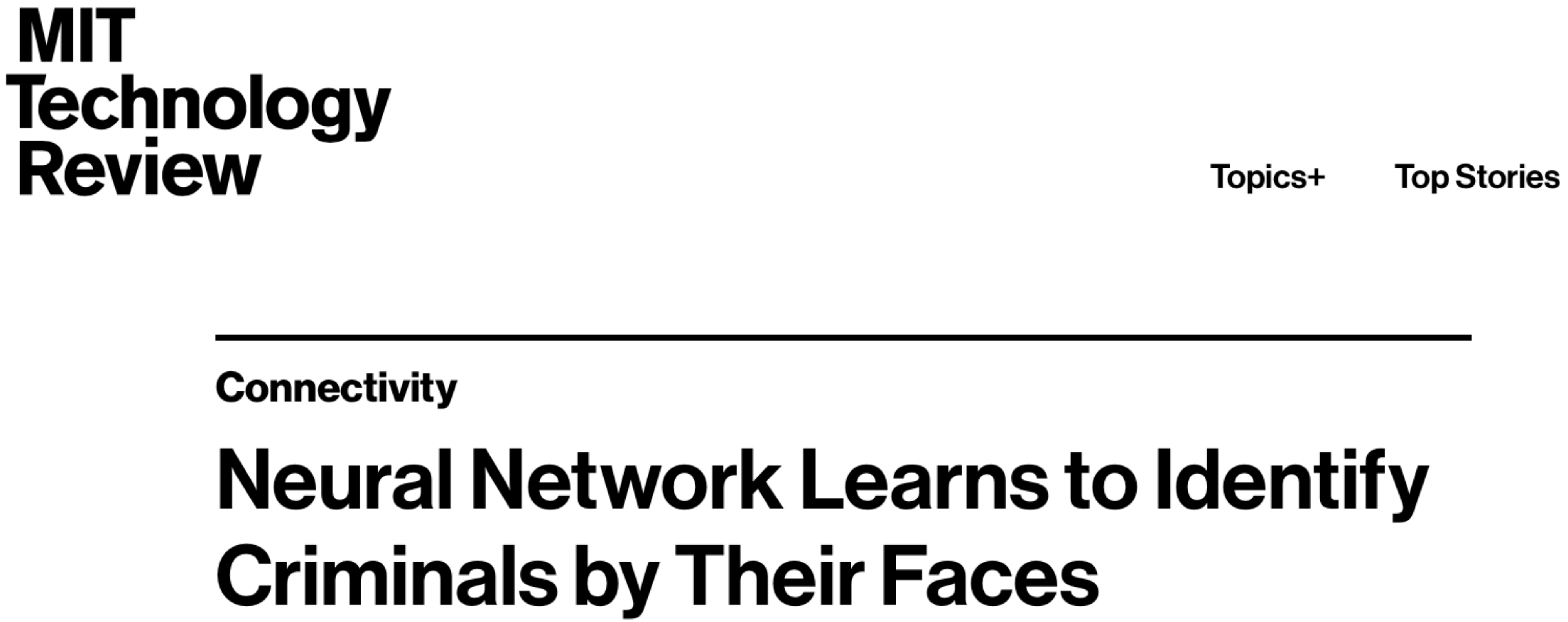

128
4x MP
256
2x MP
256
2x MP
512
1536
2048
2048
40
4
4
4
4
599
149
73
35
mean
max
L2
global temporal pooling



- sounds for filters

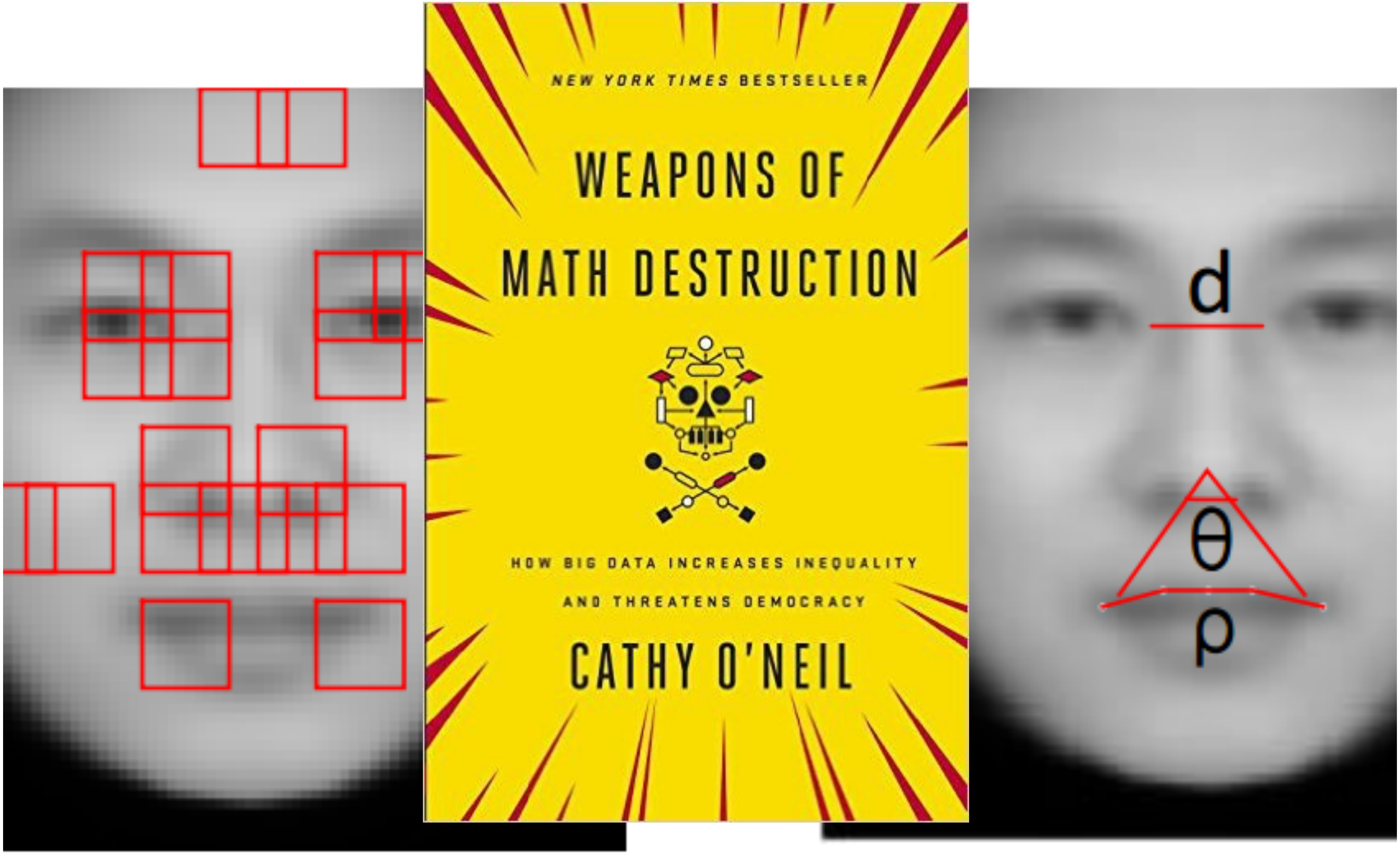
“Garbage in -> Garbage out”

Main St



Connectivity

Neural Network Learns to Identify Criminals by Their Faces

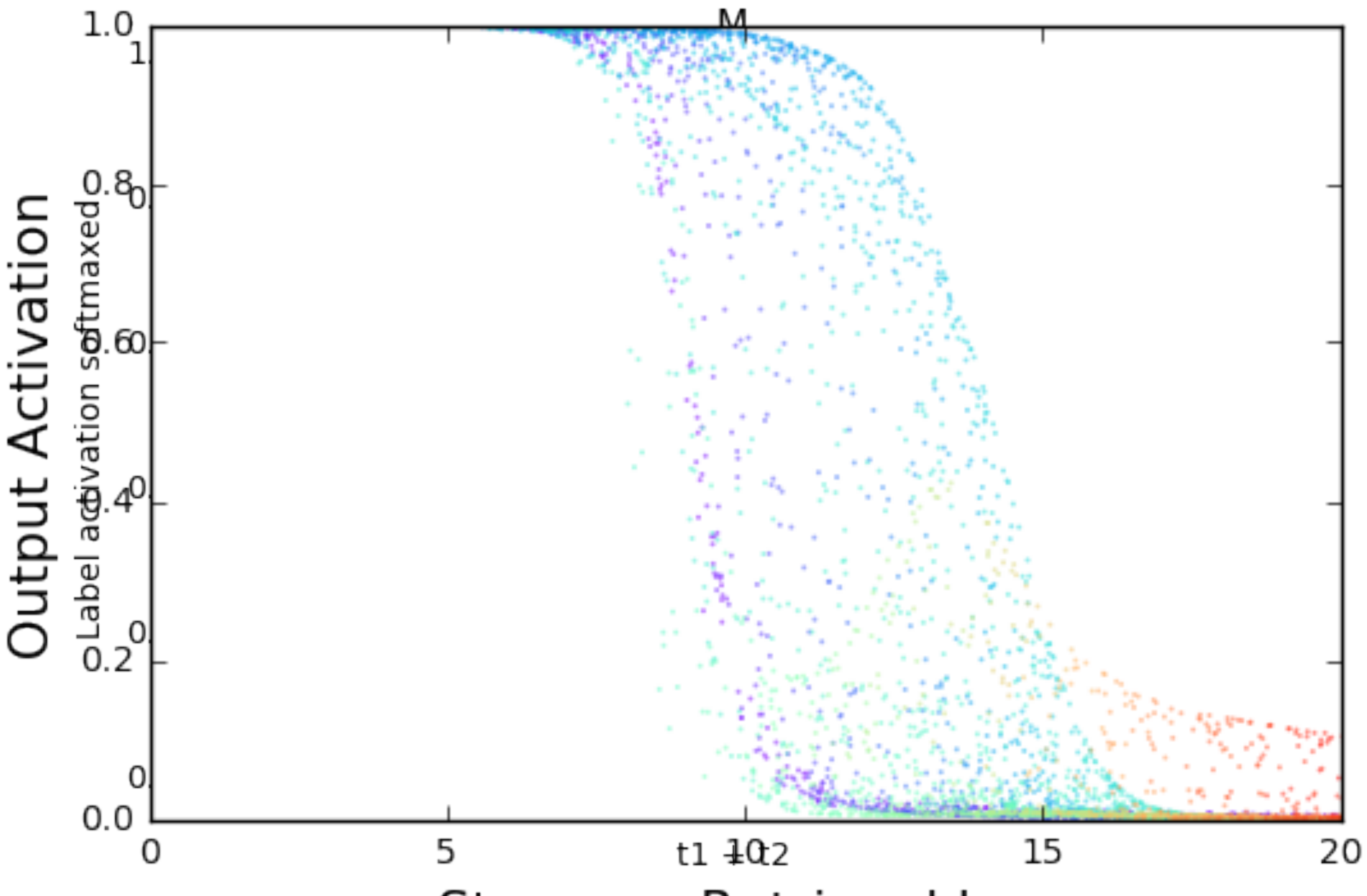


NEW YORK TIMES BESTSELLER

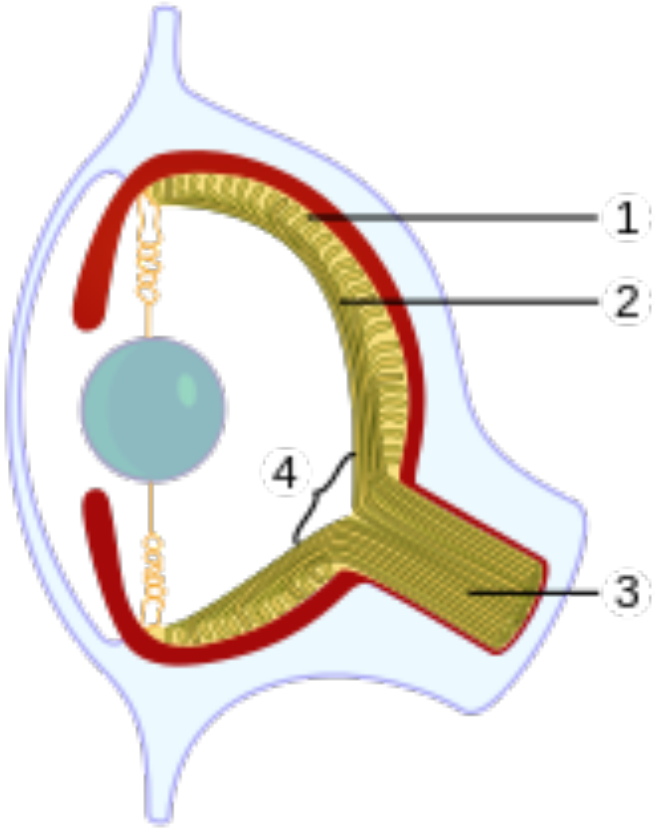
WEAPONS OF MATH DESTRUCTION

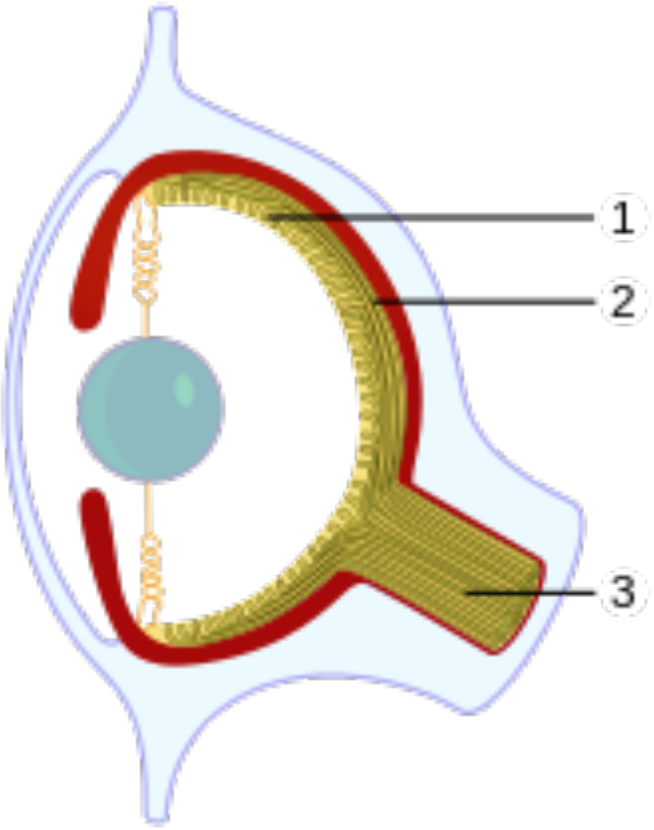
HOW BIG DATA INCREASES INEQUALITY AND THREATENS DEMOCRACY

CATHY O'NEIL


M
Output Activation
Label activation softmaxed
1.0
0.8
0.6
0.4
0.2
0.0
1
0
0
0
0
0.
0
5
10
15
20
t1
t2

Local Optima concerns

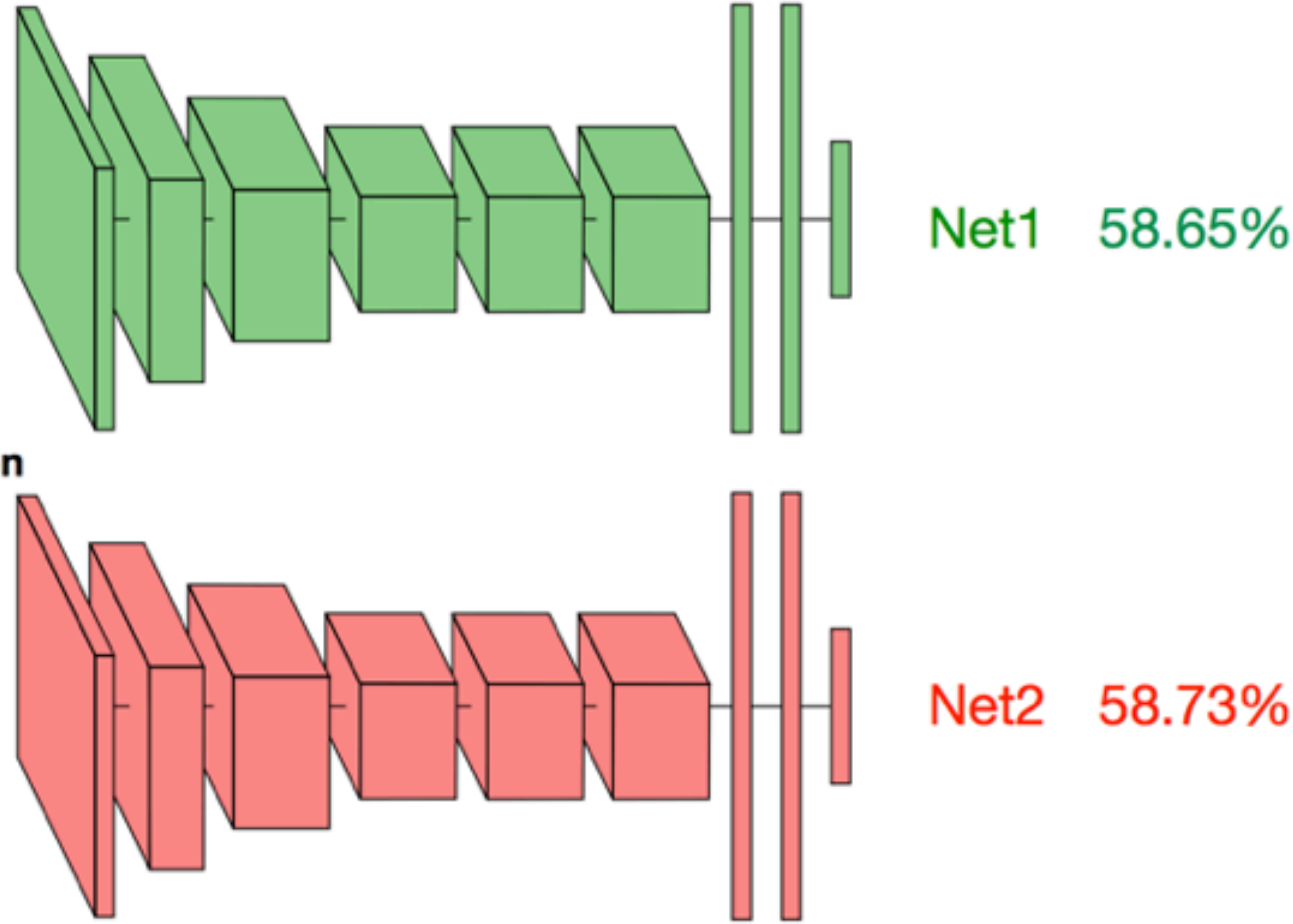




Same architecture

Different random initialization

Same data for training

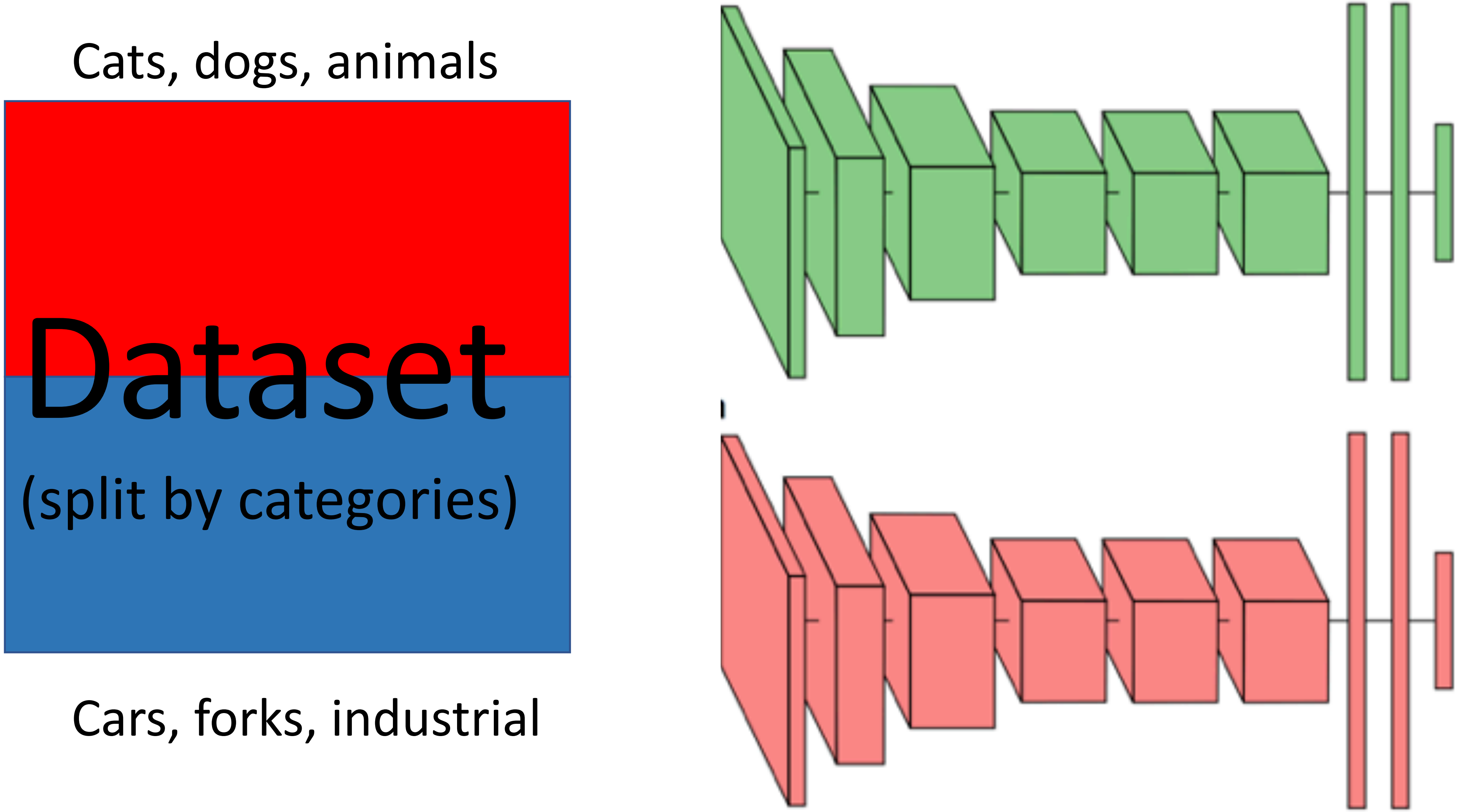


Some features the same, some different, but they learn to do the same thing.

Transfer Learning

Transfer Learning

French	**English**
actuellement	current, present
fabrique	factory
genial	genius

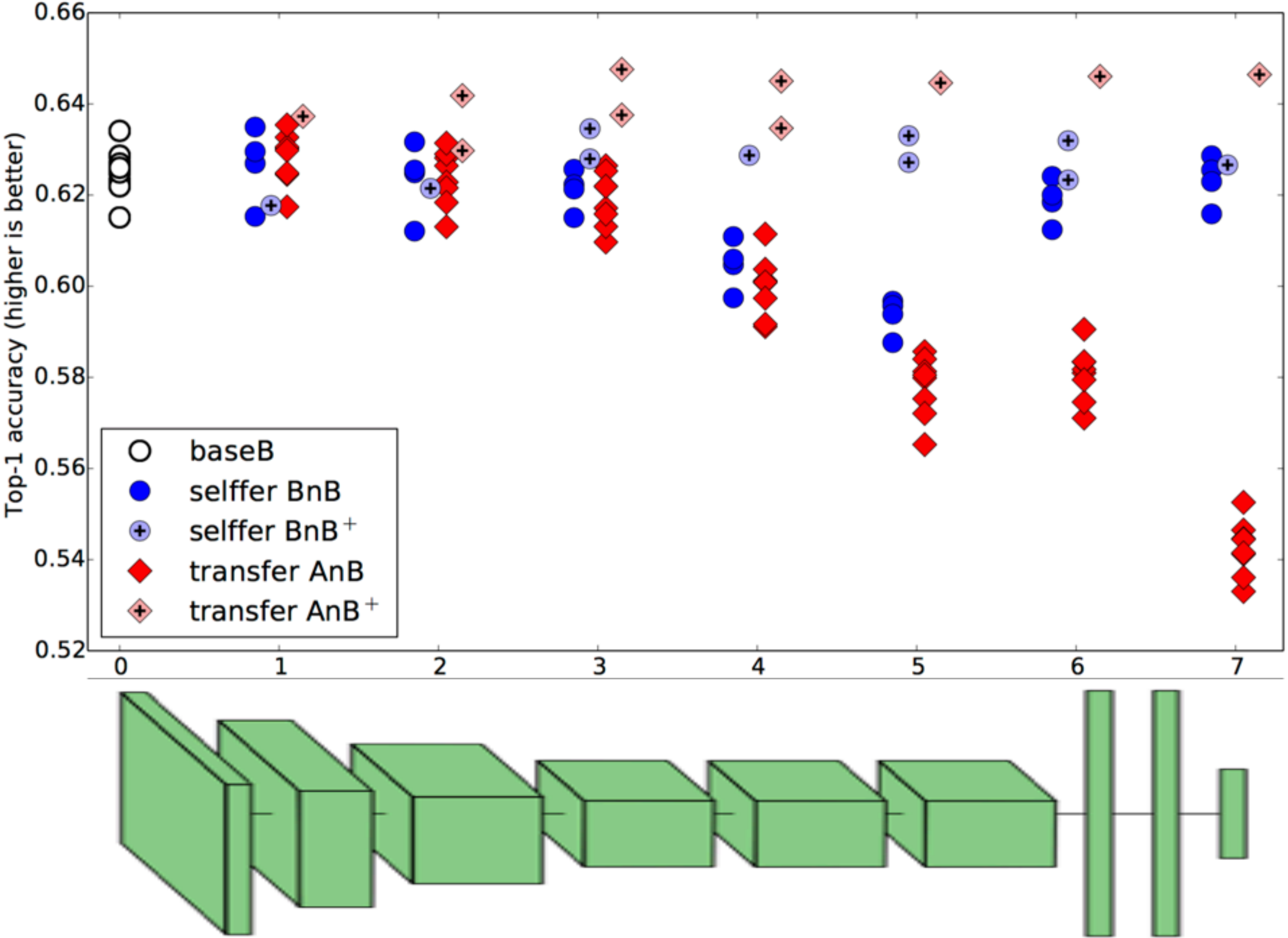


Cats, dogs, animals

Dataset

(split by categories)

Cars, forks, industrial


Top-1 accuracy (higher is better)
0.52
0.54
0.56
0.58
0.60
0.62
0.64
0.66
0
1
2
3
4
5
6
7
○ baseB
● selffer BnB
⊕ selffer BnB$^+$
◆ transfer AnB
◆ transfer AnB$^+$

Neural Networks $\approx$ information encoding

Current themes in research

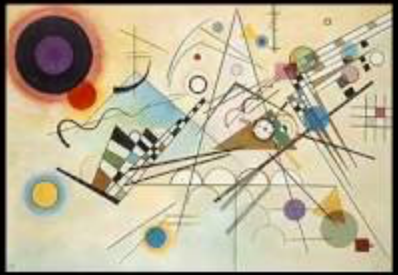
- how to phrase problems
- how to optimize better
- what are we even doing

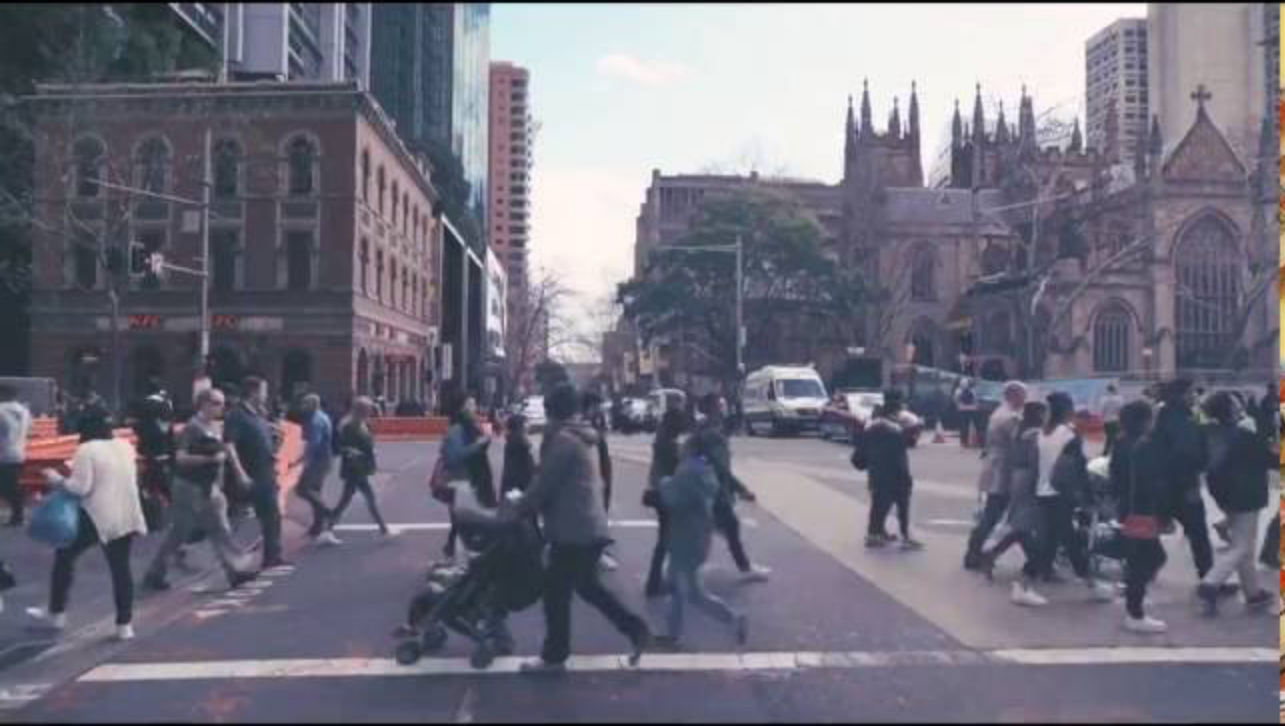
Current ML themes in industry

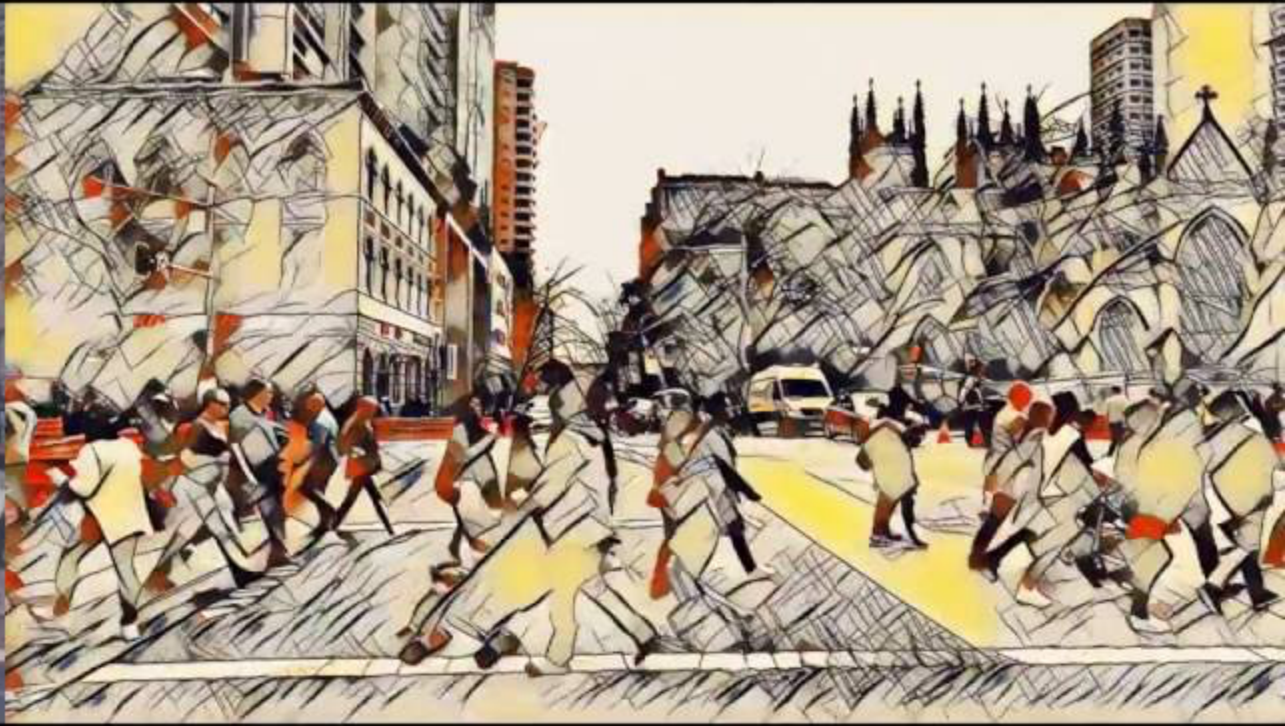
- Abusing ML
- Using ML in real products with well-trained teams and testing.

Takeaway

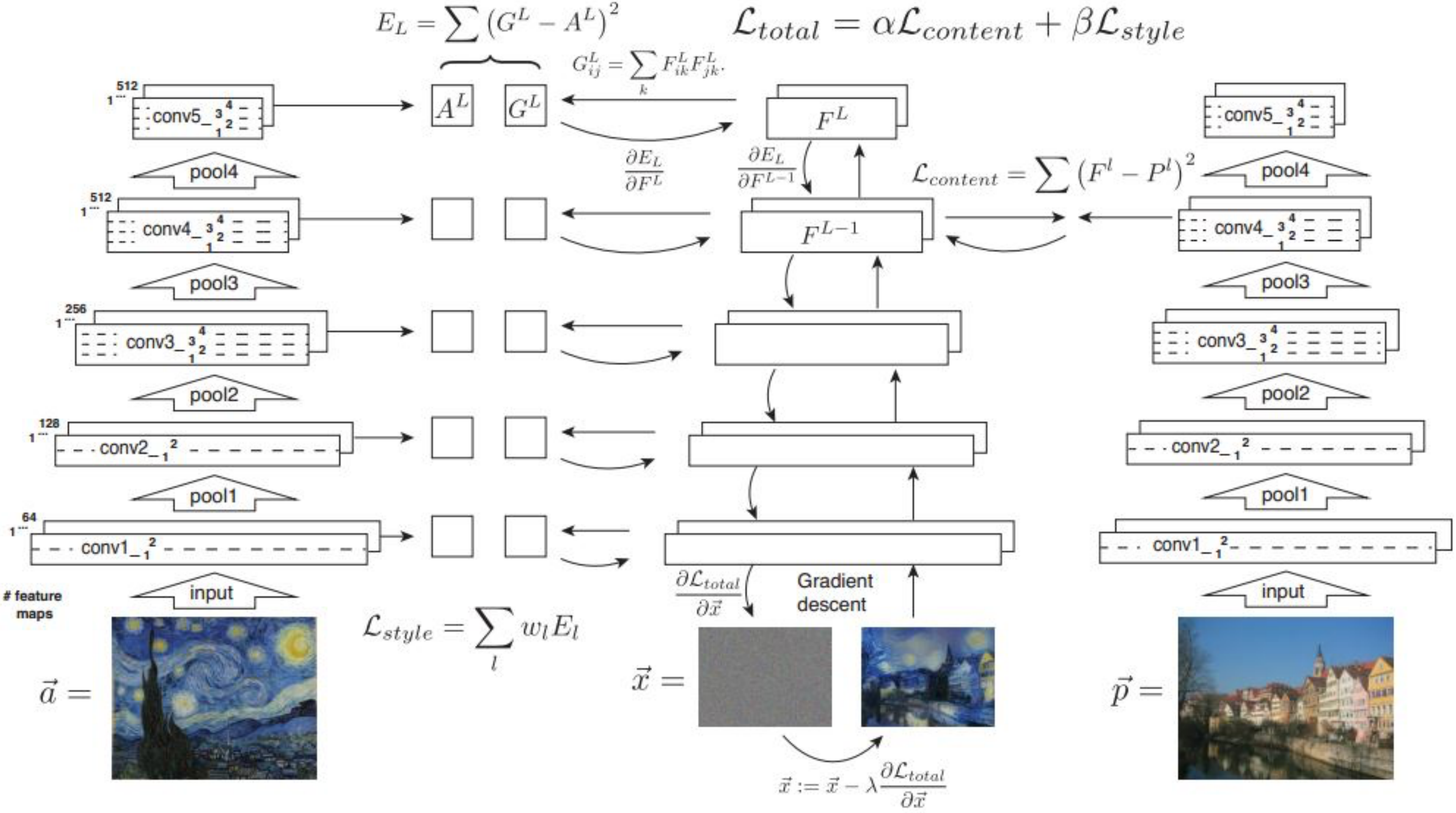
- Hierarchical Feature Learning is **useful**, **transferable**, **robust**
- Ml and NN are really cool, but also really easy to do incorrectly







- nyc



$E_L = \sum (G^L - A^L)^2$
$\mathcal{L}_{total} = \alpha\mathcal{L}_{content} + \beta\mathcal{L}_{style}$
$G^L_{ij} = \sum_k F^L_{ik}F^L_{jk}.$
A^L
G^L
F^L
$\frac{\partial E_L}{\partial F^L}$
$\frac{\partial E_L}{\partial F^{L-1}}$
F^{L-1}
$\mathcal{L}_{content} = \sum (F^l - P^l)^2$
512
256
128
64
conv5_1 2 3 4
pool4
conv4_1 2 3 4
pool3
conv3_1 2 3 4
pool2
conv2_1 2
pool1
conv1_1 2
input
# feature maps
$\vec{a} =$
$\mathcal{L}_{style} = \sum_l w_l E_l$
$\frac{\partial \mathcal{L}_{total}}{\partial \vec{x}}$
Gradient descent
$\vec{x} =$
$\vec{x} := \vec{x} - \lambda \frac{\partial \mathcal{L}_{total}}{\partial \vec{x}}$
$\vec{p} =$

- Video prediction (GAN vs other)

- inverse bike

Additional

- tensorflow + colah: to experiment with colah’s blog post, play with tensorflow playground.
- Notebooks + lectures: general ML. Lectures include derivations of the methods covered.
- Hinton’s class: intuition behind some main methods. Pretty old at this point, but a good thing to skim.
- Blogs: Karpathy, Colah, Distill: really well written and done blogs on neural nets.