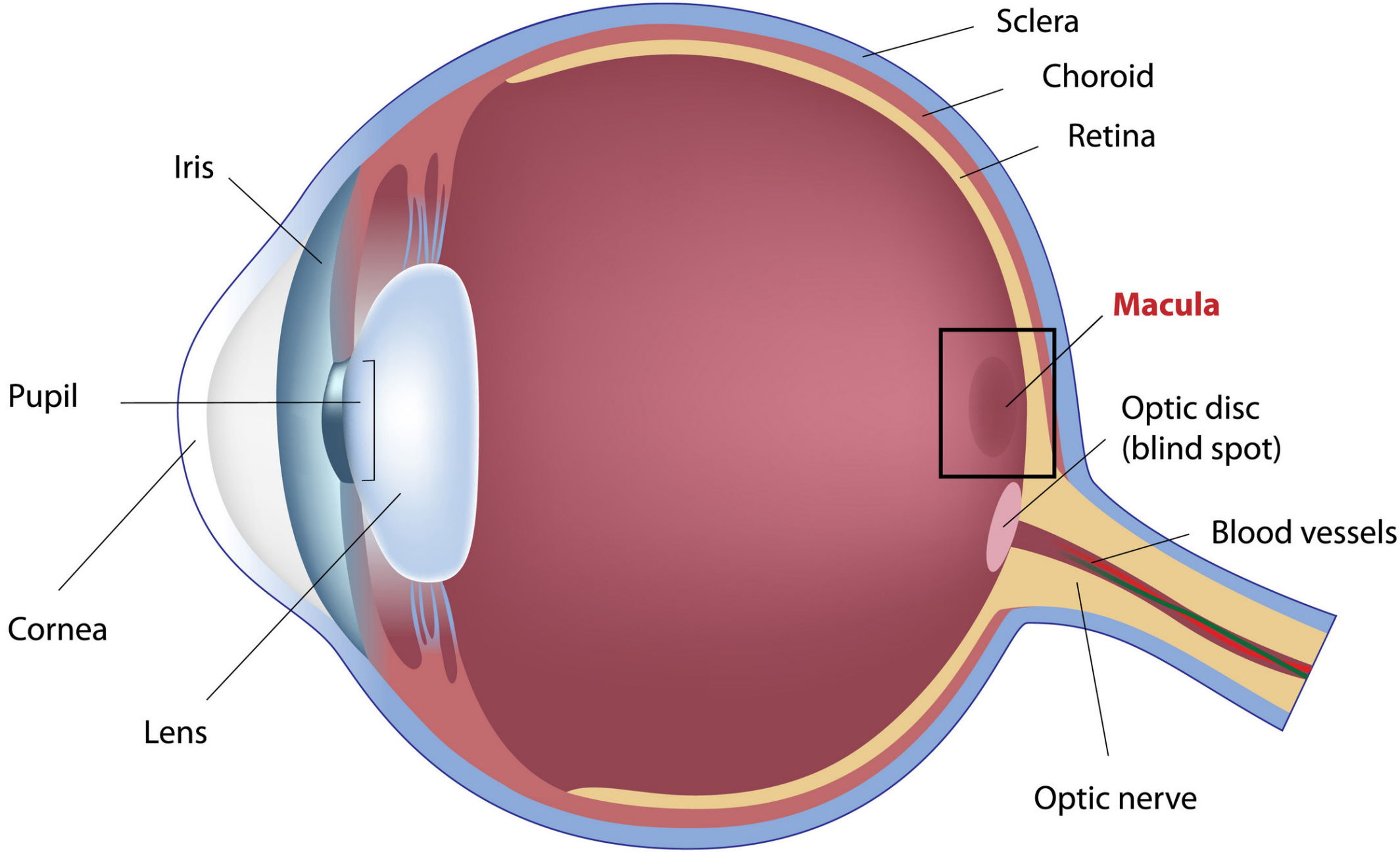
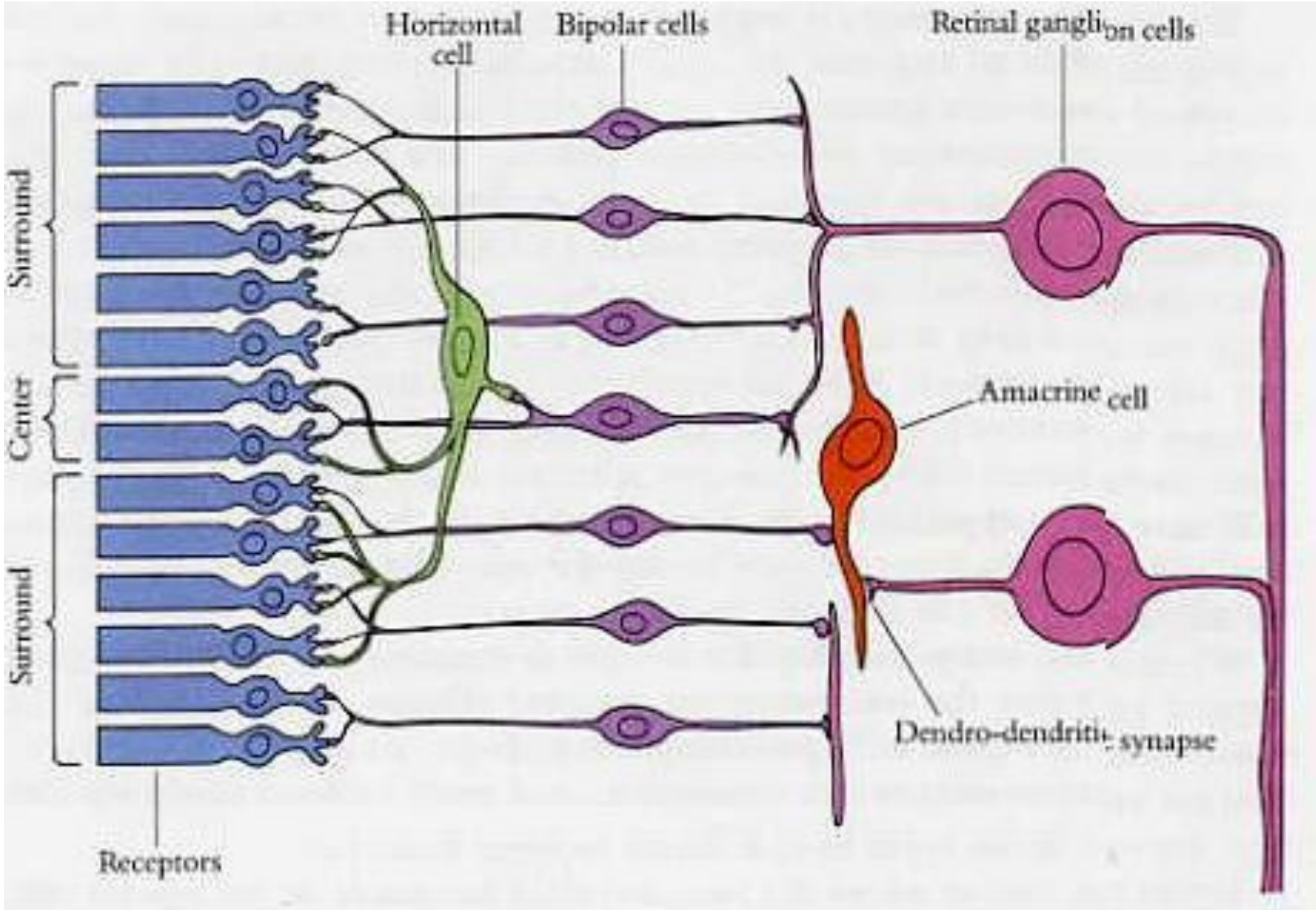
Computation and the Brain 2019

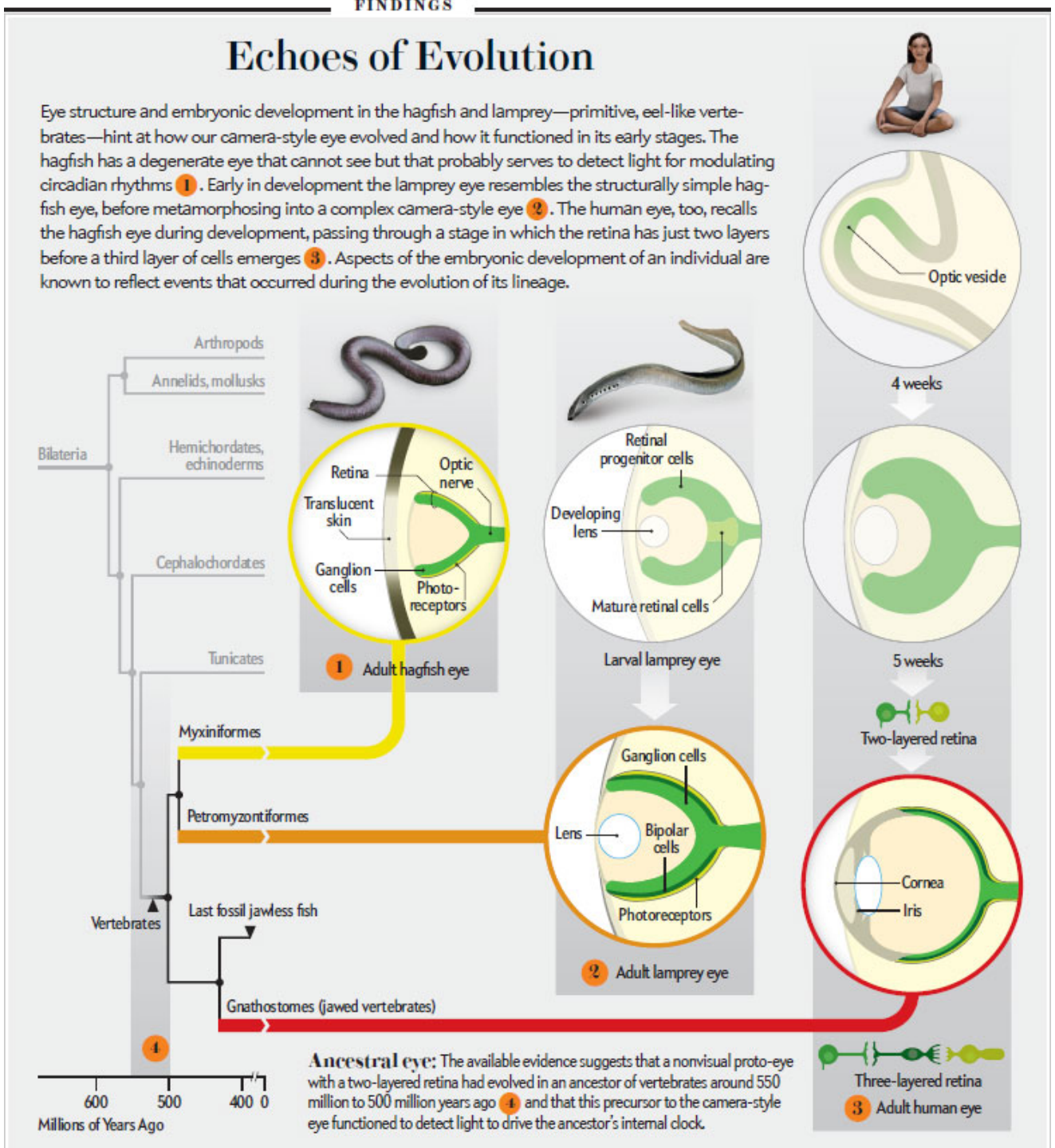
welcome to Week 3

First: What happened last Wednesday

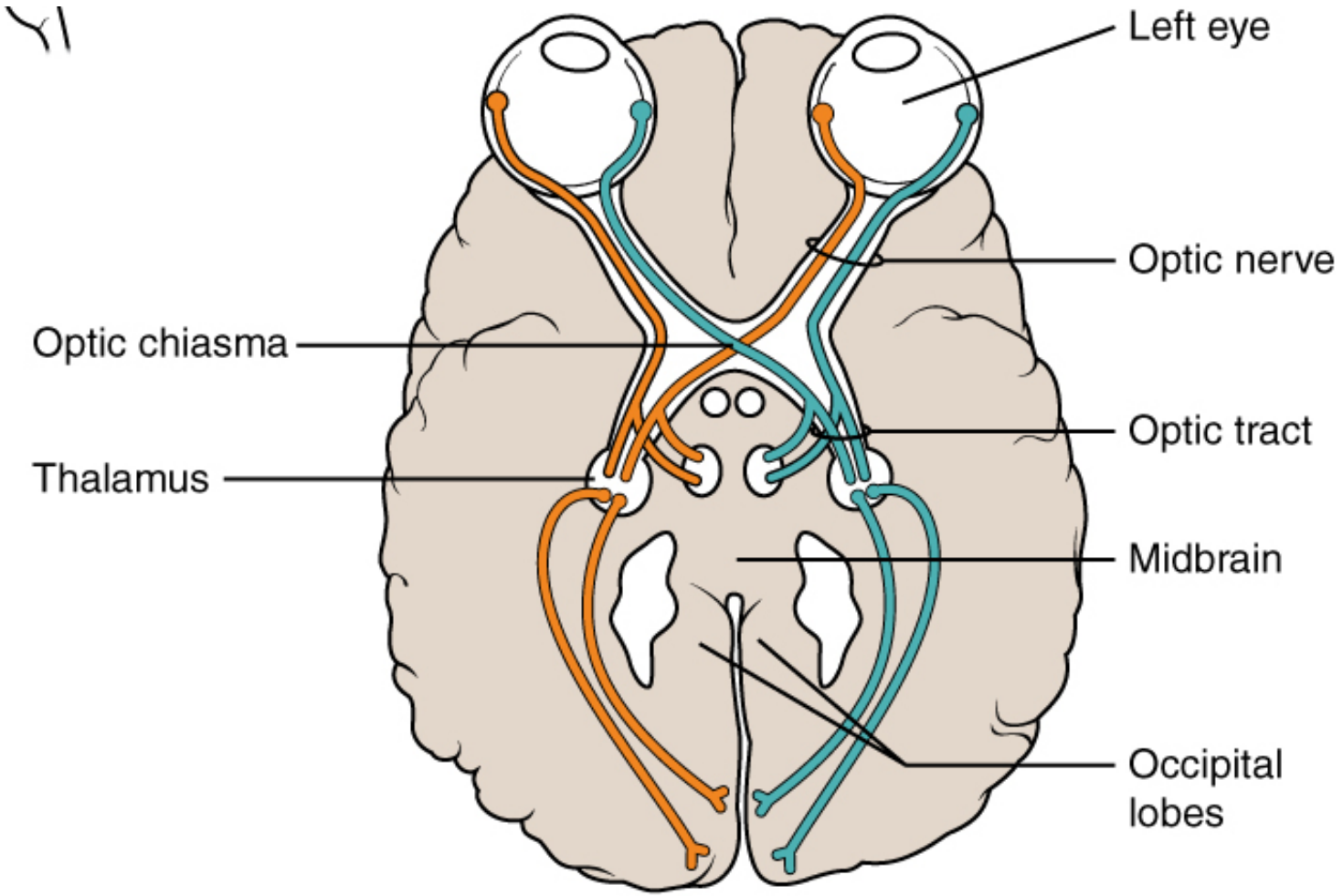

Sclera
Choroid
Retina
Iris
Macula
Pupil
Optic disc
(blind spot)
Blood vessels
Cornea
Lens
Optic nerve


Horizontal cell
Bipolar cells
Retinal ganglion cells
Surround
Center
Surround
Amacrine cell
Dendro-dendritic synapse
Receptors

One of them is the vertebrate eye…



The chiasm


Left eye
Optic nerve
Optic chiasma
Optic tract
Thalamus
Midbrain
Occipital lobes

The archetypical deep net

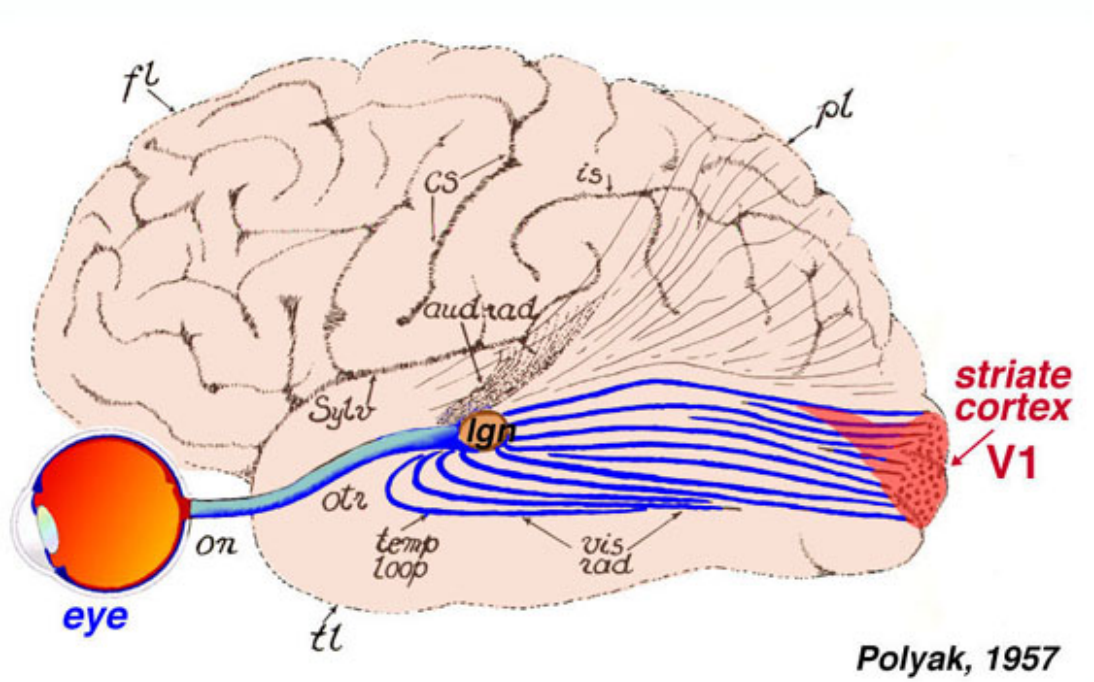
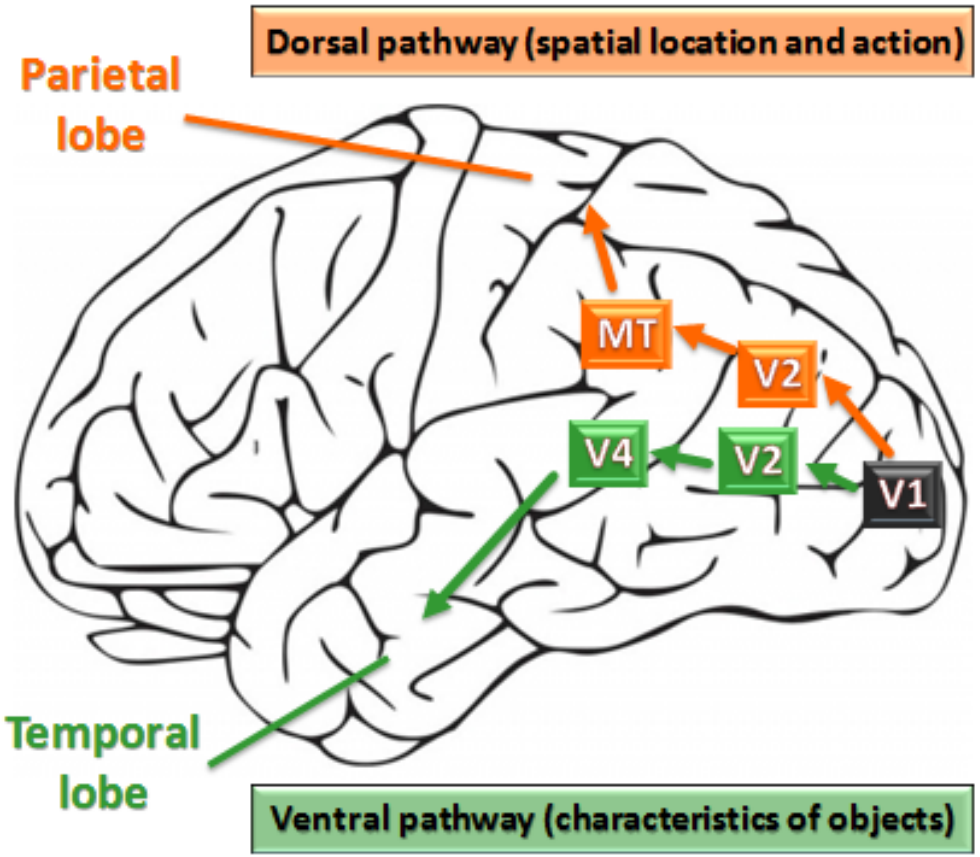
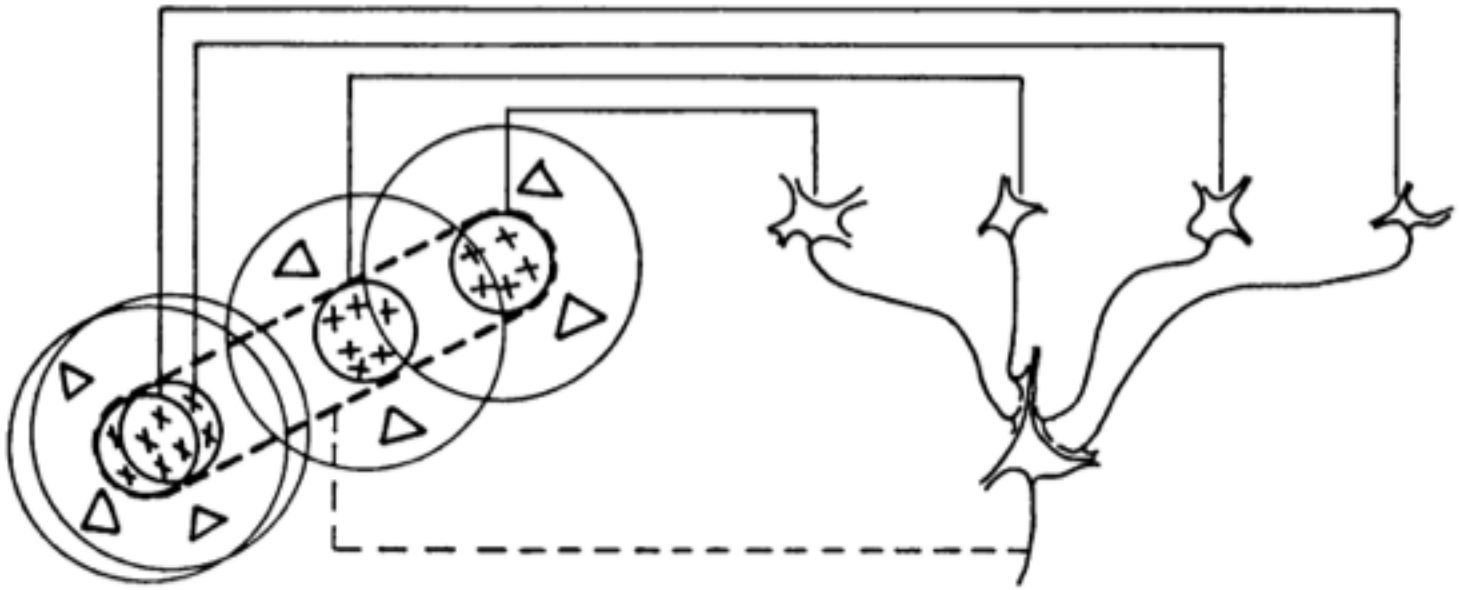


Figure 8. Visual input to the brain goes from eye to LGN and then to primary visual cortex, or area V1, which is located in the posterior of the occipital lobe.
Adapted from Polyak (1957).



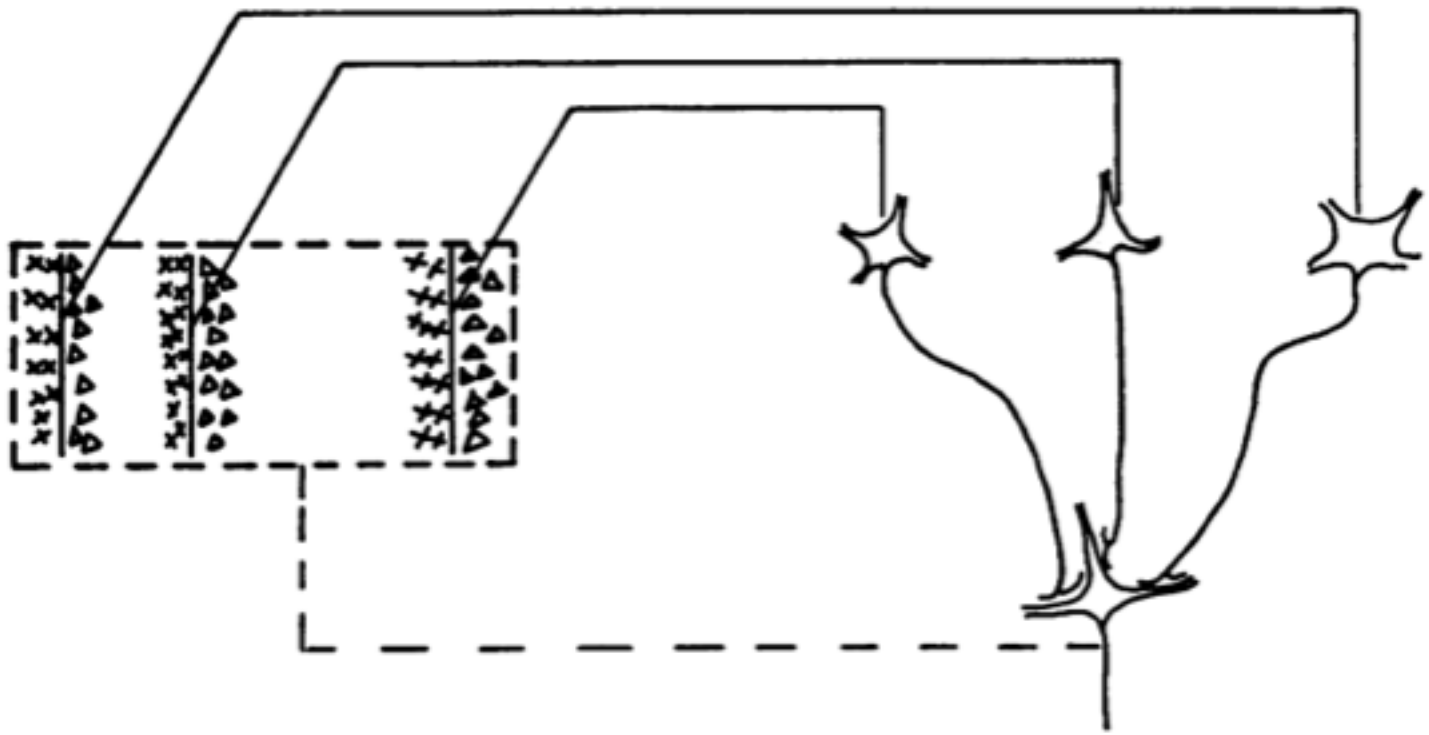
The Hubel – Wiesel conjecture: simple cells



Text-fig. 19. Possible scheme for explaining the organization of simple receptive fields. A large number of lateral geniculate cells, of which four are illustrated in the upper right in the figure, have receptive fields with 'on' centres arranged along a straight line on the retina. All of these project upon a single cortical cell, and the synapses are supposed to be excitatory. The receptive field of the cortical cell will then have an elongated 'on' centre indicated by the interrupted lines in the receptive-field diagram to the left of the figure.

The Hubel – Wiesel conjecture: complex cells

-
-



Text-fig. 20. Possible scheme for explaining the organization of complex receptive fields. A number of cells with simple fields, of which three are shown schematically, are imagined to project to a single cortical cell of higher order. Each projecting neurone has a receptive field arranged as shown to the left: an excitatory region to

K Fukushima 1980: The Neocognitron

Explicitly inspired by Hubel and Wiesel

”unsupervised learning” of weights

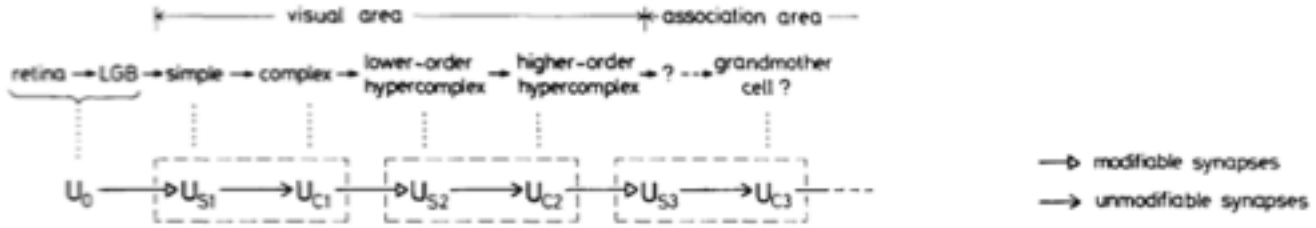


Fig. 1. Correspondence between the hierarchy model by Hubel and Wiesel, and the neural network of the neocognitron

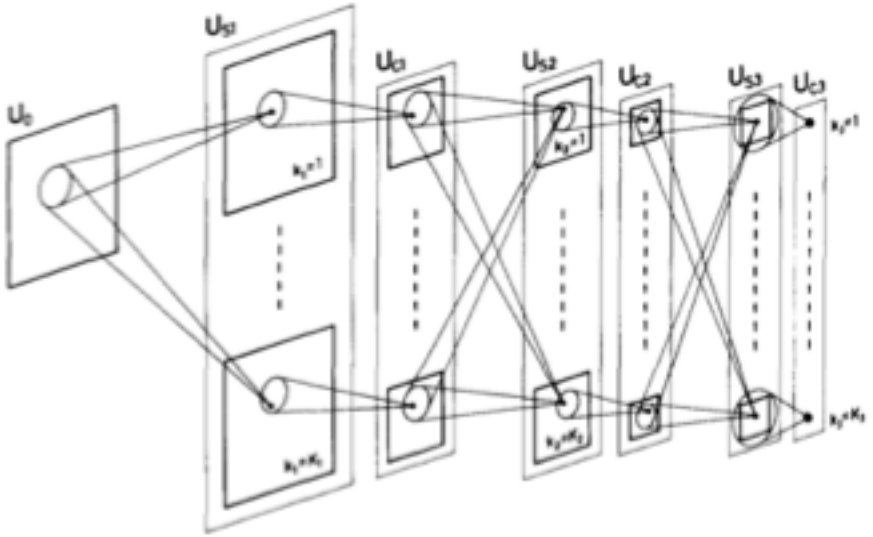
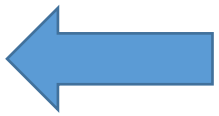


Fig. 2. Schematic diagram illustrating the interconnections between layers in the neocognitron

Gradient descent

To approximate the minimum of a differentiable function
$f: R^d \rightarrow R$

Set $t = 0$, and set x^0 to some initial point
Repeat
  calculate the gradient $\nabla f(x^t)$
  and the ``learning rate'' $\boldsymbol{\alpha}_t$
  $x^{t+1} = x^t - \boldsymbol{\alpha}_t \nabla f(x^t)$
Until termination condition (typically, $|x^{t+1} - x^t|$ very small)



This is the science/ dark art of the matter

What is known about gradient descent

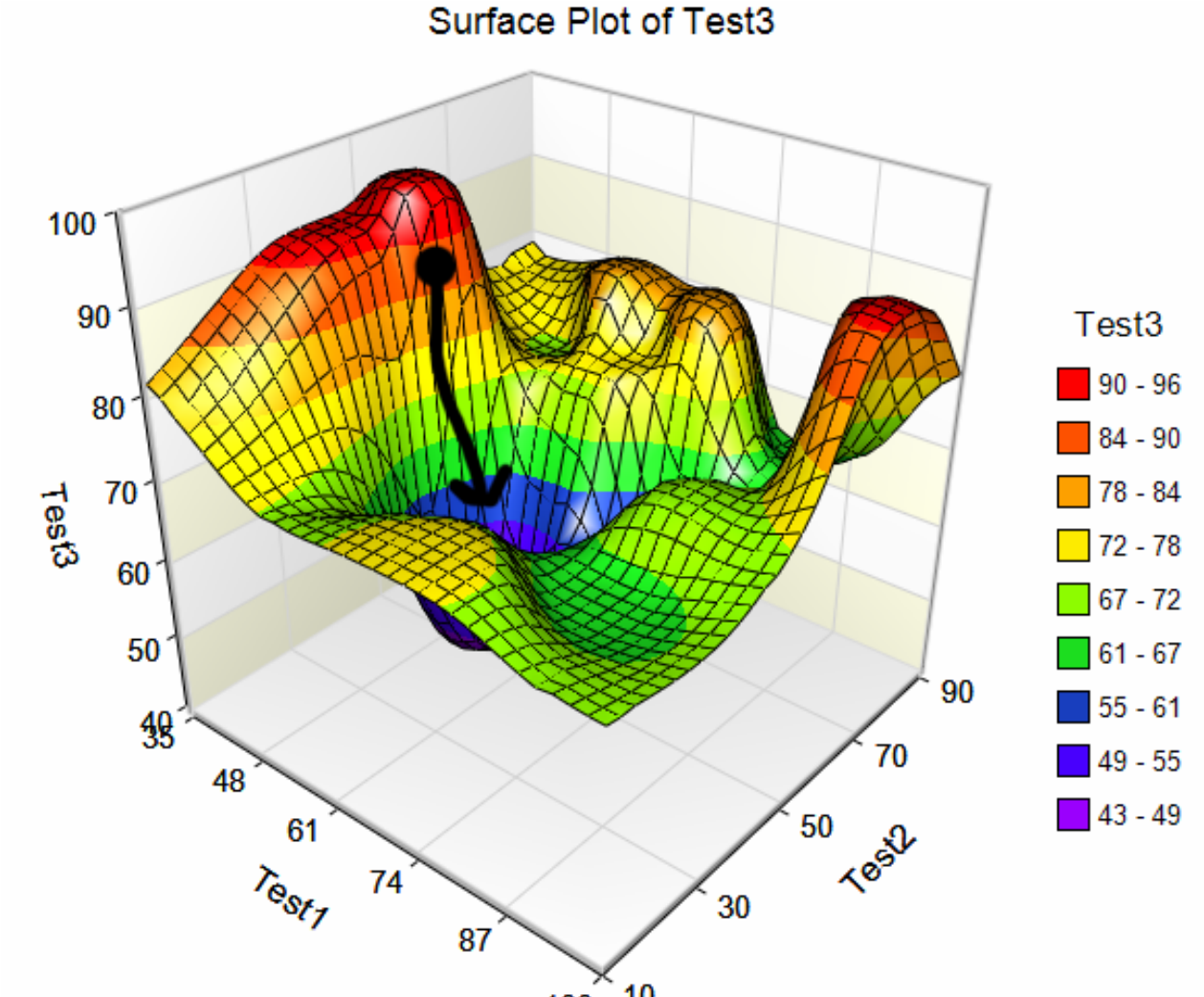
Theorem: If f is a strictly convex quadratic function $x^T A\, x + b\, x$, and L is the largest eigenvalue of A, taking $\alpha_t = 1/L$ guarantees logarithmic convergence (the distance from the optimum decreases exponentially fast).

Theorem: Ditto if f is a general strictly convex function, and L is an upper bound on the Hessian's eigenvalues.

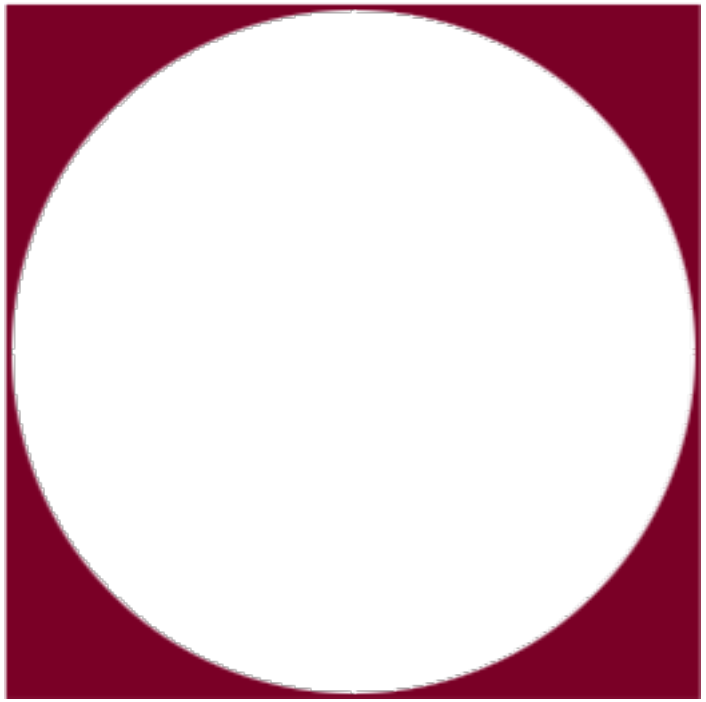
Theorem: If f is not convex, then no convergence can be guaranteed whatsoever

It doesn't matter!

Why?


Surface Plot of Test3
100
90
80
70
60
50
40
Test3
35
48
61
74
87
Test1
90
70
50
30
Test2
Test3
90 - 96
84 - 90
78 - 84
72 - 78
67 - 72
61 - 67
55 - 61
49 - 55
43 - 49

Can you visualize
10-dimensional data?
Are you sure?



…volume of the amazing shrinking inscribed ball

1-d	1.00
2-d	0.79…
3-d	0.52…
4-d	0.31…
5-d	0.16…
…	
10-d	0.002…
16-d	0.000004…

Questions?

Feedback from Readings and Viewings?

Olhausen claims that 85% of V1 is still not understood, while Gallant claims that we already have computational models that can predict the behavior of V1 and V2 neurons up to the theoretical limit. Are they in fundamental disagreement?

Olhausen repeatedly proposes using distributed systems as computational models of the brain, in particular in the resistive model of horizontal cells and in using local network representations to compute sparse coding coefficients. Is this something that we could make better use of in ANNs

Why are rods only of type on? Is it better for capturing minimal amounts of light since it would be transmitted as inhibitory and therefore need less of a threshold to fire?

If the object can change appearance depending on which cones the information lands on, how are the cones that the information lands on chosen?

What do the other 85% of V1 do? (I have my theory but I want to hear from more knowledgeable people.)

Referring to the Rodney Brooks quote, should the amount of time taken for a subject to evolve be used as a proxy for importance? and what role could increasingly varied evolutionary pressure play in neurological evolution?

Understanding the underlying mechanisms of vision and artificially simulating processes that can accomplish vision are both important, but which approach offers better prospects and gives us the most benefit?

Is there a maximum extent to which ANNs can be used to understand and model the brain? What are other techniques used to understand how the vision system works?
How have classifiers evolved with discoveries in vision? Jack Gallant's talk said that his model out-performed other out of the box deep neural nets, have these models been put into production systems?

What are the end goals of these discoveries? Complete brain modeling and understanding or improving current technologies. Both?

Temporal reasoning is inherent to human cognition. What are the “crutches” of machine learning that might explain this gap? Lack of sufficient data? IID assumptions? Are the bounds for online learning algorithms too pessimistic to explain fast adaptability? Should we be motivated to study restricted classes that can be learned online with very little data to identify models of concept representation, similar to how Valiant’s “Evolvability” takes a learning-theory-first approach to understanding possible target concepts for evolution?

Why are cone cells depolarized by default and hyperpolarized when activated? Does that just make it work more easily with how the light sensitive molecule in the cone reacts to light and the subsequent response pathway? I don’t think it would affect how ON/OFF cells work, you could just swap which was which.

How well do we understand the spider brain? As they mentioned in the last video, it has a lot of very complex behavior and I’d imagine raising and dissecting spiders would be much easier than mice. Of course it doesn’t have a cortex so you would lose some human analogy but still would probably yield a lot of good insights - how much have people studied the jumping spider?

In the paper they find sufficient requirements/constraints to construct receptive fields (basis functions) that resemble what simple cells do. How could we find the necessary conditions? And does evolution even care about necessary conditions?

Why is it a good idea to have RGCs subsample more and more cones with increase in eccentricity? It seems inefficient to have many more cones that just get sampled. Perhaps there is a balance between energy and sensitivity that I don't know about which dictates constraints on the number of RGCs vs. cones.

There seems to be a common theme that too much detail is not ideal for human visual processing. How would broader cognitive functioning be effected if there was (even slightly) more precision in the detail that the retinal cells process?

What physical properties of retinal cells allow them to process images with the same level of detail as TVs and projectors, but with far less computational effort?

Are ANN being used by neuroscientists to study the function of brain areas? Besides using ANN to study visual areas, is ANN used to study other part of the brain, i.e. language?

Is there any study on functions of other brain areas, i.e. IT, and how do research conduct experiment on understanding in these areas?

I keep thinking about how V1 works, and how this relates to the input being a time-series. My guess is that V1 controls what areas of the visual input gets attention. This could be in an attempt identify areas of interest, gathering information from periphery for context clues, or isolating attention to a specific area. Am I close?

The eye makes small movements, gathering greater clarity (super-resolution theory). Is this a weakness of the eye? Is it an advantage for computer vision to have complete clarity in the entire visual input? Or is the eye much more efficient by only gathering information needed?

Are there any theories why we do not have photoreceptors that span more area other than that for some reason we can't make them bigger? Or any theories why we can't?

Gallant mentions that his model of V4 is as accurate as the classical model of V1. Is the classical model of V1 built with as many constraints?

(

The deep learning community believes that as the depth of the neuron network increases, the accuracy will increase. Does our neuron system has lots of layers?

In Jack Gallant's model, they state that their model predicts accurately. Is it just because of the "magic" od DNNs?

At the end of last class, Professor Papadimitriou briefly mentioned phantom limb pain. One of the "remedies" for this natural phenomenon includes tricking our eye through mirror box therapy. Can we use illusions to learn more about our visual feedback process?

Somewhat going off what one audience member questioned, could our attempt at building computer systems that mimic the neural mechanisms of the brain help us better understand the underpinnings of our visual system?

The discussion in the Olshausen talk about the sparse coding in the visual cortex to represent images in compact neurons shows how there are often more cells, or "hardware", dedicated to a certain task than is often necessary. Why do we think there are cells dedicated to a certain task than necessary? perhaps for evolutionary purposes or to promote brain plasticity?

The talks and readings all had a similar theme of equating cell functions to mathematical functions. As neuroscience researchers investigate more about parts of the brain that are currently unknown to us, is it possibly dangerous to assume that every part of the brain has a mathematical analog to it since this assumption can lead us to create inaccurate or approximate models of the brain?

Today: Learning Theory

- Suppose that you want to learn to recognize edible mushrooms
- You know that edibility depends on two things:
  - How red is the cap
  - How tall is the stalk
- And suppose further that in fact, you know the true answer is one of 100 given rectangles in the height – redness plane
- You have an oracle that tells you the answer for any mushroom you see in the forest
- How do you find the correct rectangle?

Learning Theory: formalism

- Set X of objects (all mushrooms)
- Labeling function f: X → {+,-} (edible/not)
- Distribution D on X × {+, -} (in the forest)
- Set of hypotheses H = {h1, …, h100} (the 100 rectangles)
- What do you do?

Learning Theory: The Algorithm

- Simple idea: Walk in the forest and pick many mushrooms, say m
- Find out the answer for each
- Find the rectangle of the 100 given ones that fits the most of the data (hint: it will fit all of them!)

$$\min_{h \text{ in } H} (\#\text{wrong labels} / m) \leftarrow \textbf{\textit{Loss}}$$

- How high should m be?

Learning Theory: Theorem

Theorem: $m = \log(|H| / \delta) / \varepsilon$ iid samples from D suffice to guarantee that

with probability at least $1 - \delta$,

the probability of error of the learned rule (rectangle) is at most ε

Proof: easy probabilistic calculation (union bound)

BUT this assumes that…

1. One of the hypotheses in H is exactly correct
2. And that H is finite

(Recall the formula $m = \log(|H| / \delta) / \varepsilon$)

Long story short: The learning theorem still holds, except now there is ε^2 in the denominator, and $\log|H|$ is replaced by the VC dimension of H

VC-dimension?

- We say that H shatters a subset S of X for every subset T of S there is an h in H that contains all of T and nothing from S – T
- VC(H) is the size of the largest set shattered by H
- Example: halfspaces have VCD d+1, intervals 2, rectangles 4, circles 3
- (Notice: = number of parameters!)

The moral of the story

- To learn a classification task, you need labeled data S
- The amount of data |S|you need depends on the approximation and certainty you want, and on the difficulty of H (log of size, VC dimension)

The Bottom Line

- To learn a classification task, you need labeled data S
- The amount of data |S| you need depends on the approximation and certainty you want, and on the difficulty of H (log of size, VC dimension)
- ***Learning reduces to minimization***

$$\min_{h \text{ in } H} \text{Loss}(S, h)$$

For example: The Perceptron algorithm

- Learning a half space (NB: VCD = d+1)
- Sample ~$d/\ \varepsilon^2$ points and find any hyperplane that separates positive and negative examples
- Remember the Rosenblatt perceptron, the NYT quote from 1957, XOR and the first AI winter…
- Of rather limited use…

BUT… the kernel trick and SVMs

- https://www.youtube.com/watch?v=3liCbRZPrZA

Connection between Learning Theory and Deep Neural Nets?

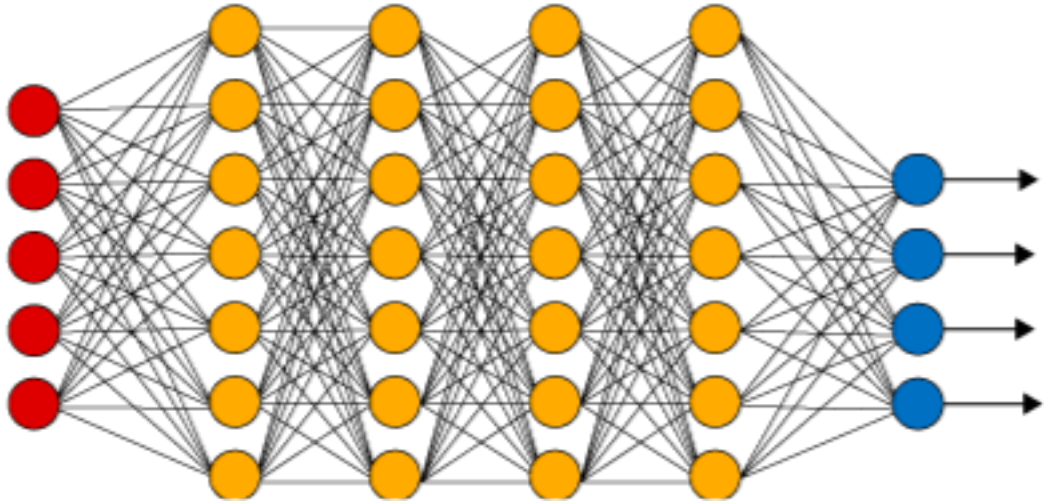
Yes there is!

Choose H = all DNNs of this shape and any possible weight combination W **(Vcdim ~ number of weights!)**

You have to

$\text{minimize}_W\ L(W, S) =$

$\Sigma_S\ (\text{classification error})^2$



Do it by ***stochastic gradient descent***

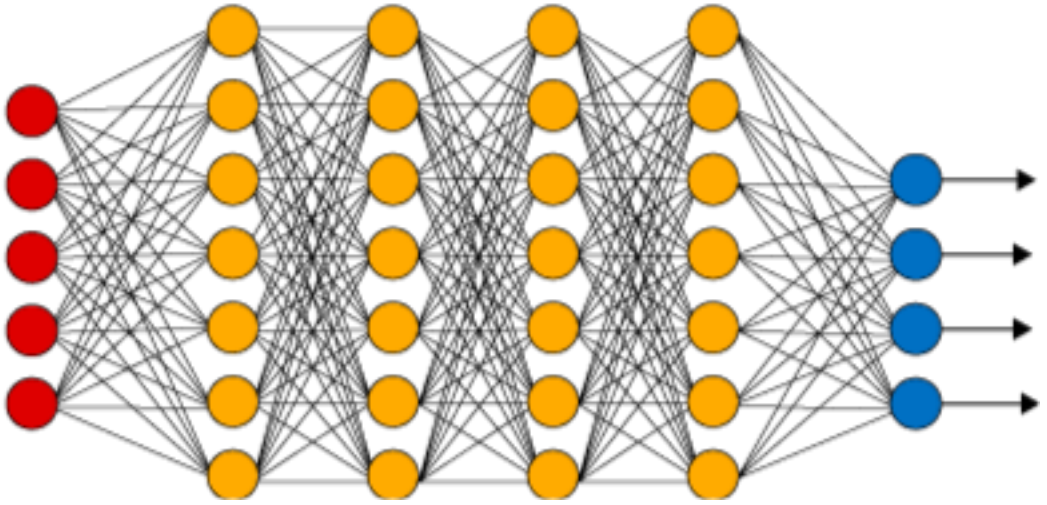
Connection to Deep Neural Nets?

Do it by ***stochastic gradient descent***

To compute the gradient wrt W, take a small sample from S (a minibatch), find its loss as a fcn of W, and use the chain rule from calculus to update W

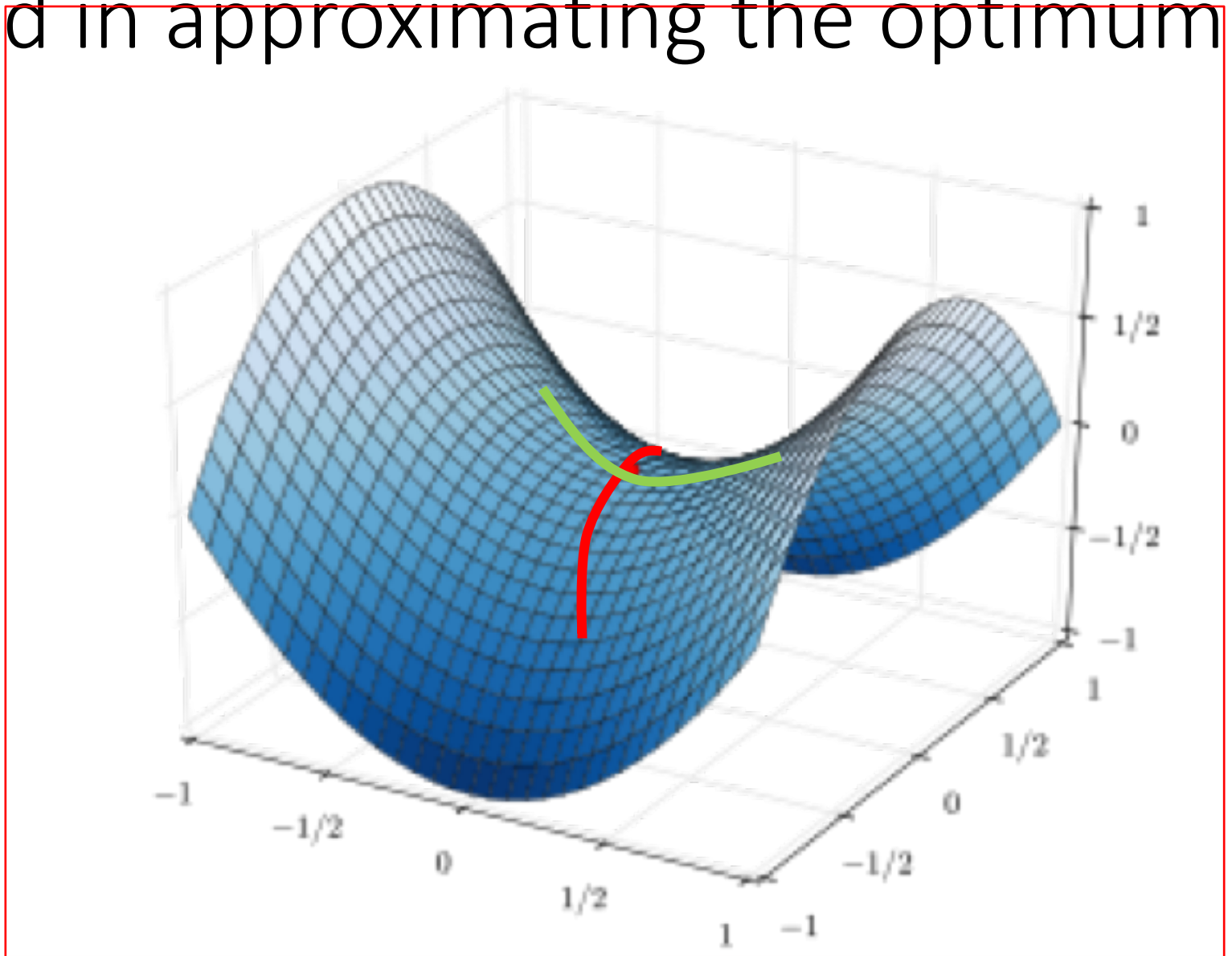
This is called **back propagation**

[Rumelhart - Sejnowksy 1986]



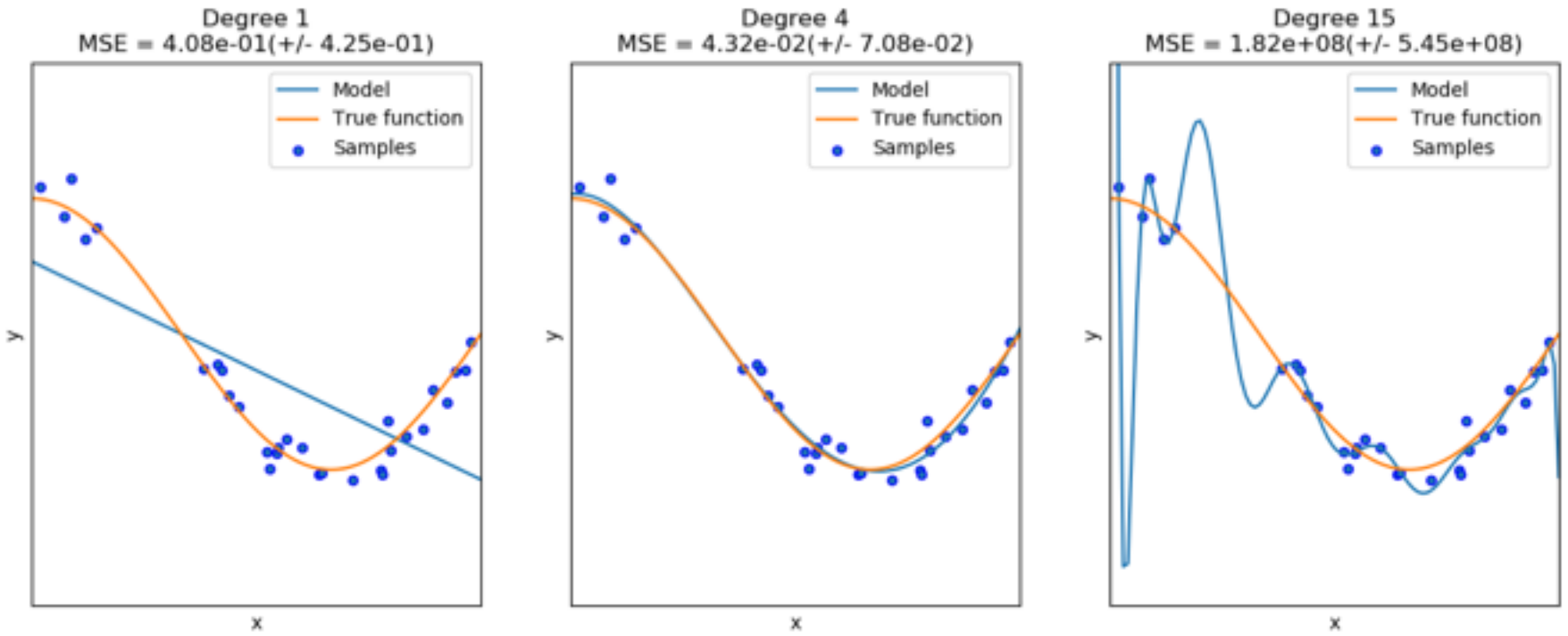
Why does stochastic gradient descent succeed in approximating the optimum?

saddle points



But getting to the optimum is the (relatively) easy part…

Overfitting



How does SGD avoid overfitting?

How does SGD
avoid overfitting?

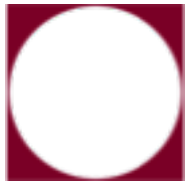
One old answer: dropout

BUT….

… and may be reinforced…

Unsupervised Learning: Reducing the dimensionality of data

- The true curse of dimensionality: you have never been
- We lack geometric intuition



- Q: Is there such thing as ``redundantly high dimension?"
- E.g., 10^8 points in 10^5 dimensions?
- YES! If $\sqrt{\ln(\#\text{points})/\text{dimension}} << 1$, *you are wasting dimensions….*

The Johnson – Lindenstrauss Lemma (JLL)

Theorem (JL, 1984): For any set of n d-dimensional vectors $x_1, \ldots, x_n$ there is another set of vectors $x'_1, \ldots, x'_n$ with dimension $k = 8(\ln n)/\varepsilon^2$ such that for all i, j:

$$|x_i - x_j|^2 \approx |x'_i - x'_j|^2 (1 \pm \varepsilon)$$

What happens if you project a specific d-dim vector on a random k-dim space?

What happens if you project a random d-dim vector on a specific k-dim space?

The Johnson – Lindenstrauss Lemma: Proof sketch

You want to project in a direction R such that no distance between two pints is distorted too much

You must avoid $O(n^2)$ directions (the directions of the line segments connecting two points in your set)

Key insight: If the dimension is huge, it's very easy to avoid getting even close to n^2 directions

A random direction will do so with high probability

See paper by Sanjoy Dasgupta "a simple proof of JLL"

The Johnson – Lindenstrauss Lemma (JLL): The Moral

- for any data set
- if the dimensions have no special meaning for you
- and you are mainly interested in the distances between the data
- and you can tolerate an error of $\boldsymbol{\varepsilon}$
- then your dimension should not exceed $8 \ln n/\boldsymbol{\varepsilon}^2$

Aside: an application to complexity theory:

- The d-dimensional traveling salesman problem can be approximated when d is constant (1, 2, 3, …)
- But when dimension ~ cities, it is NP-complete to approximate
- Q: What happens when dimension ~ √cities?
- A: NP-complete, because you can project so the distances are essentially the same
- All the way down to ~ log (cities)

What if…

- Dimension is not super high
- ***“My data is not any data!”***

- Principal component analysis (PCA, or SVD: singular value decomposition, or LSI: latent semantic indexing)

	Alice	Bob	Claire	Dave	Eve
• Saving Private Ryan	3	2	3	5	4
• Crouching tiger…	5	3	3	3	4
• Memento	4	1	1	3	3
• Almost famous	4	5	4	1	2
• Eternal sunshine…	4	3	1	3	3
• The departed	2	2	5	3	2
• White men can’t jump	1	3	5	2	3
• No country for old men	2	1	4	5	4
• Gladiator	1	2	3	2	2
• Zoolander	2	2	5	3	1

3	2	3	5	4
5	3	3	3	4
4	1	1	3	3
4	5	4	1	2
4	3	1	3	3
2	2	5	3	2
1	3	5	2	3
2	1	4	5	4
1	2	3	2	2
2	2	5	3	1

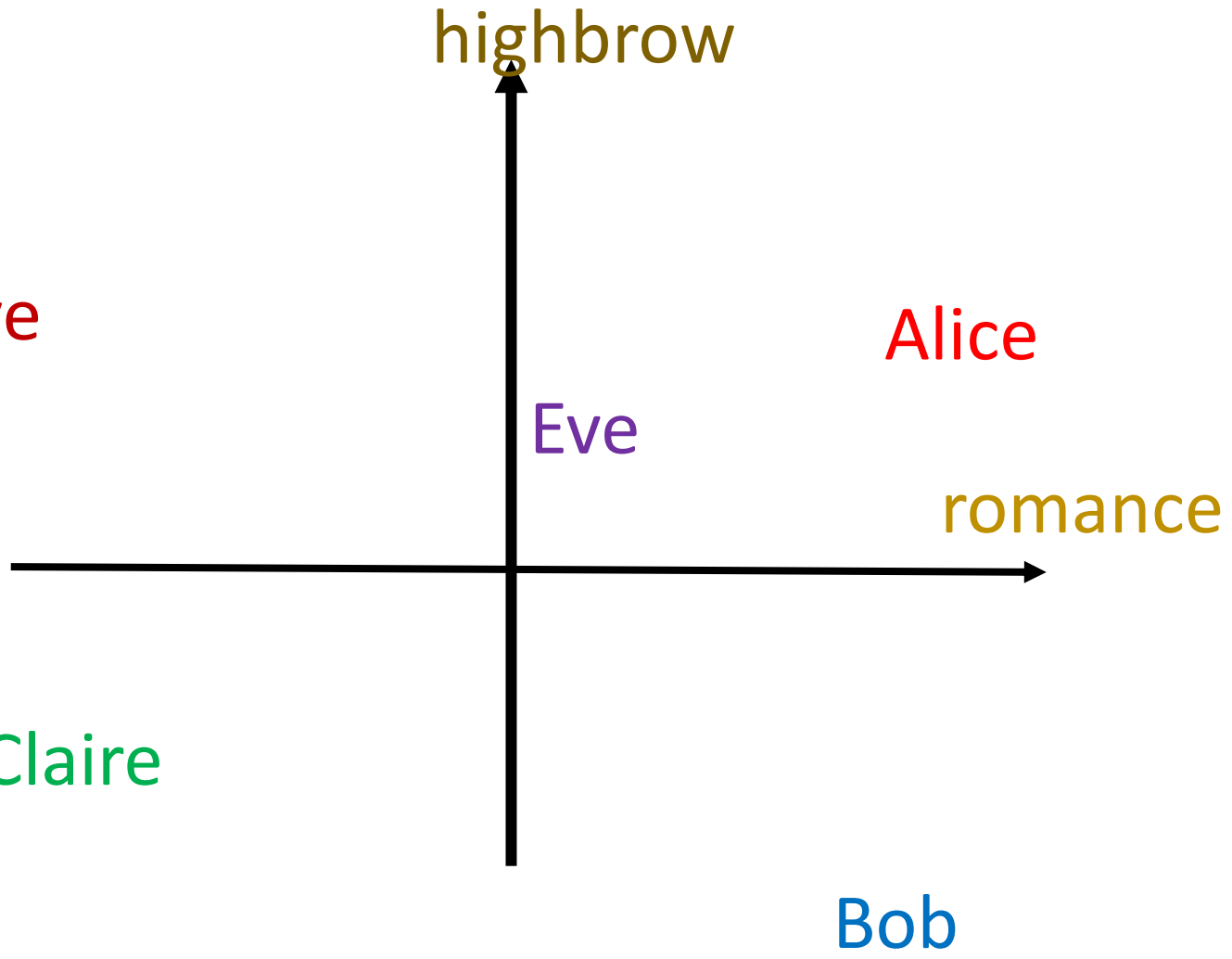
PCA to the rescue!

Q: can you approximate these five 10-d vectors x_1, x_1, …, x_5 as

$$x_i \approx x_{ave} + a_i v + b_i u$$

for two appropriate vectors u, v?

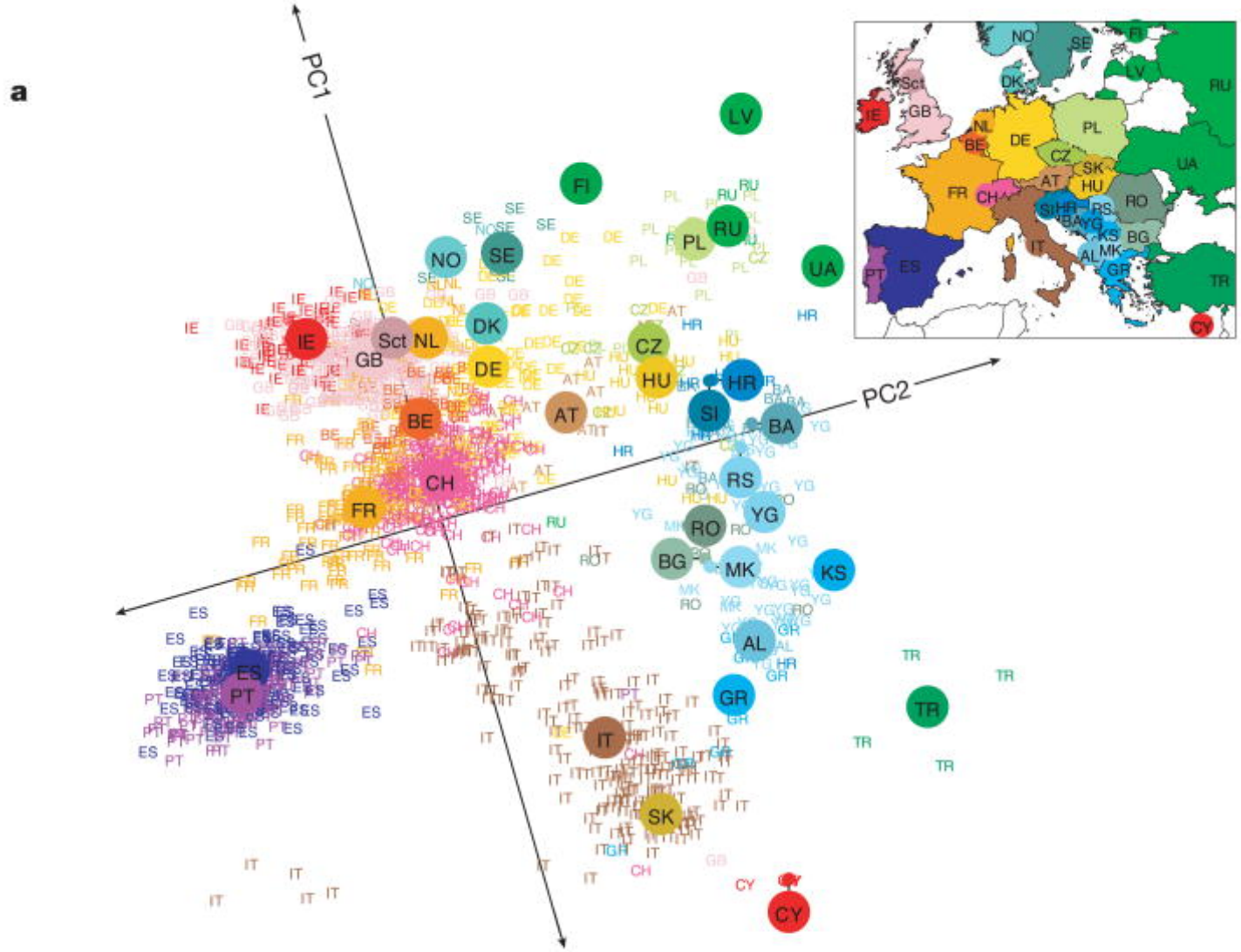
PCA to the rescue!



200 “eigenfaces” are enough to approximate reasonably well most faces (1990)

Imagine the compression achieved!

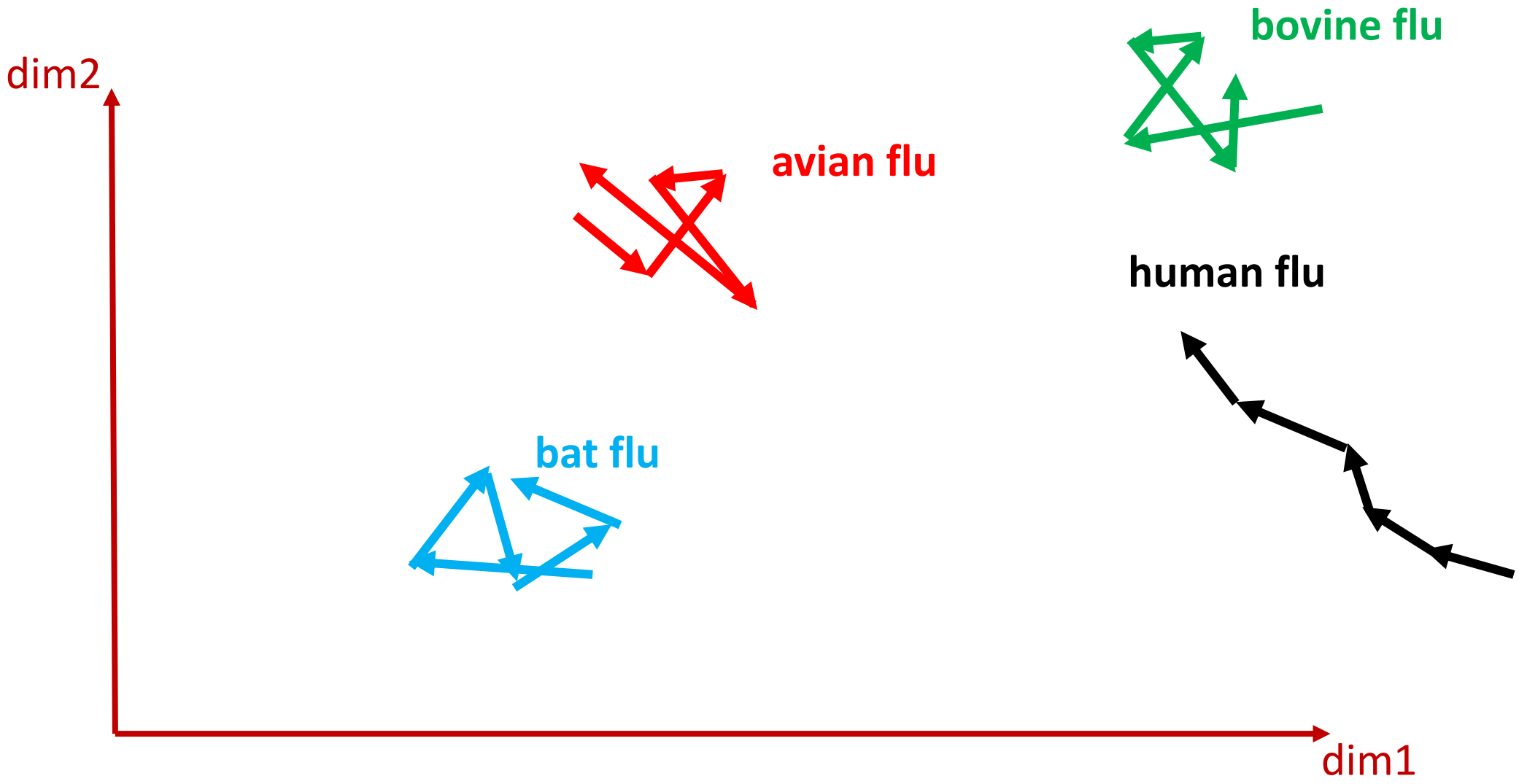
DNA of Europeans: n = 3,000 d = 500,000



Latent semantic indexing of documents [Gerard Salton 1975]

- Every book is a 10,000 (say) dimensional vector, where every dimension is a word
- Preprocessing: stopwords, word forms, log of #appearances
- So, the Library of Congress is a collection of 15,000,000 vectors of dimension 10,000
- What are the principal components?
- "Topics" How many?
- cf: word embeddings and ***"Parallelogram analogies"***

PCA of the flu virus database…



What is PCA? (aka SVD)

Of all the low-dimension projections
choose the one that has
the smallest sum of quadratic distances
from the original point set
Caveats: sensitive to high variance noise;
misses nonlinear structure,
hard to interpret the higher-order dimension

PCA: the math

- PCA computes the top eigenvectors of the covariance matrix $C = A^T A$
- C_{ij} ~ How many people like both movie i and movie j
- C a positive definite matrix, maps the unit sphere to an ellipsoid E_C
- We want to find the directions of its longest axes

So, how do you find this?

- Preprocessing: Curate data, subtract to zero mean, normalize coordinates
- Maybe project to a lower subspace first

Then…

PCA: the algorithm

Pick a start vector u(0) at random

Keep multiplying it with matrix C:

$u(t+1) = u(t) \times C$

until $|u(t+1) - u(t)|/|u(t)| < \varepsilon$

The resulting u is the direction of the largest axis

Project it out and repeat for the second largest…

Sparse coding
(as in the Olshausen-Fields paper)

- You are given data vectors $y_1, \ldots, y_n$ in R^d
- find an mXd coding matrix B with column vectors $b_1, \ldots, b_m$ in R^d (expect $m > d$)
- and sparse vectors $x_1, \ldots, x_n$ in R^m
- Such that $\min_{B,\, x} \sum_i |y_i - Bx_i|^2 + \sum_l S(x_i)$ ← a sparsity penalty function

Sparse coding: Algorithms

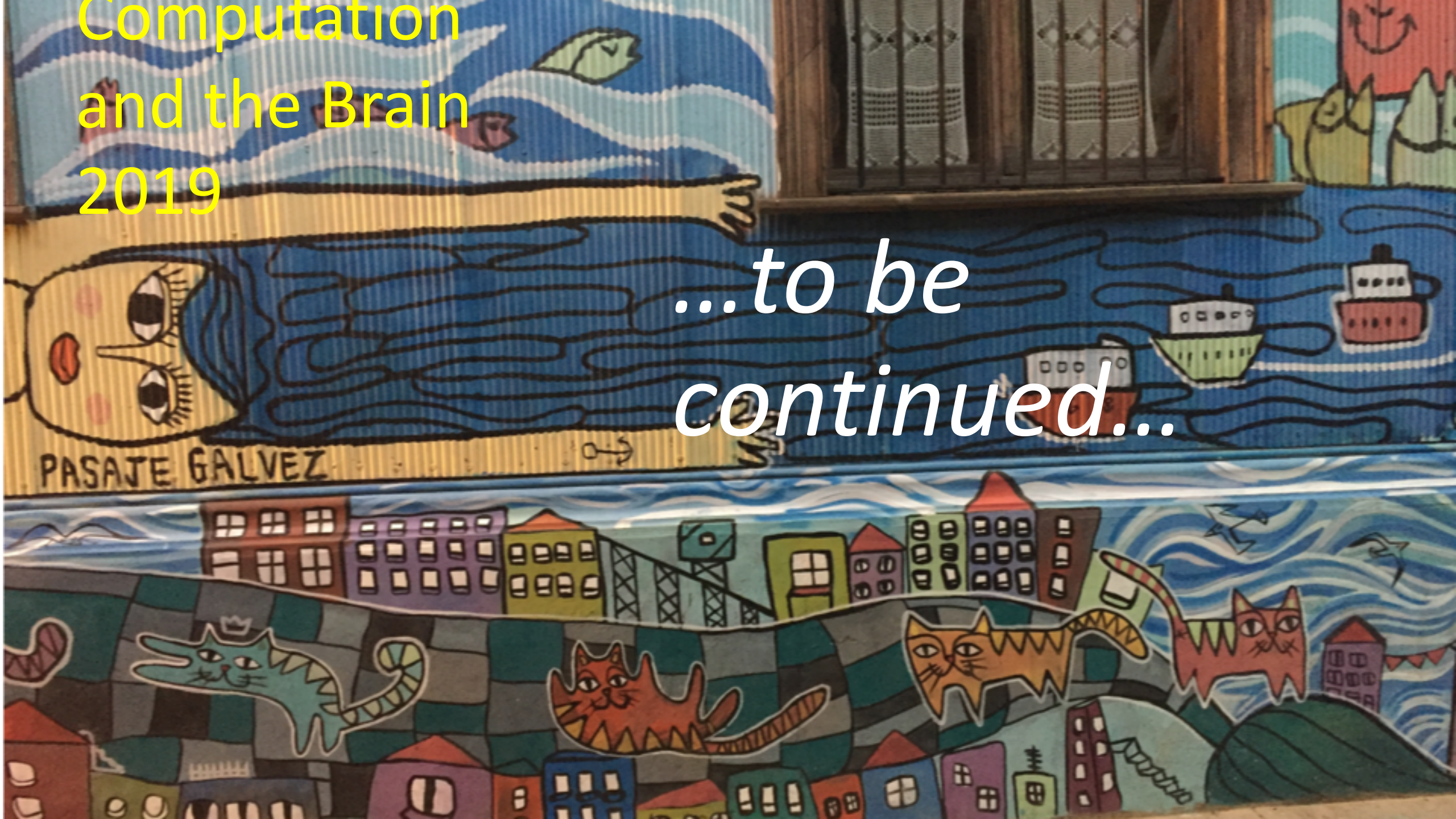
- Gradient descent [OF 1998]
- Maximum likelihood, assuming a generative model and noise [Lewicki and Sejnowski 2000]
- Alternating minimization:

Start with a basis B

Repeat: Sparse decode the data on B to get x

Gradient step on B

- Performance guarantees, [Arora et al 2015]


Computation
and the Brain
2019
...to be
continued...
PASAJE GALVEZ