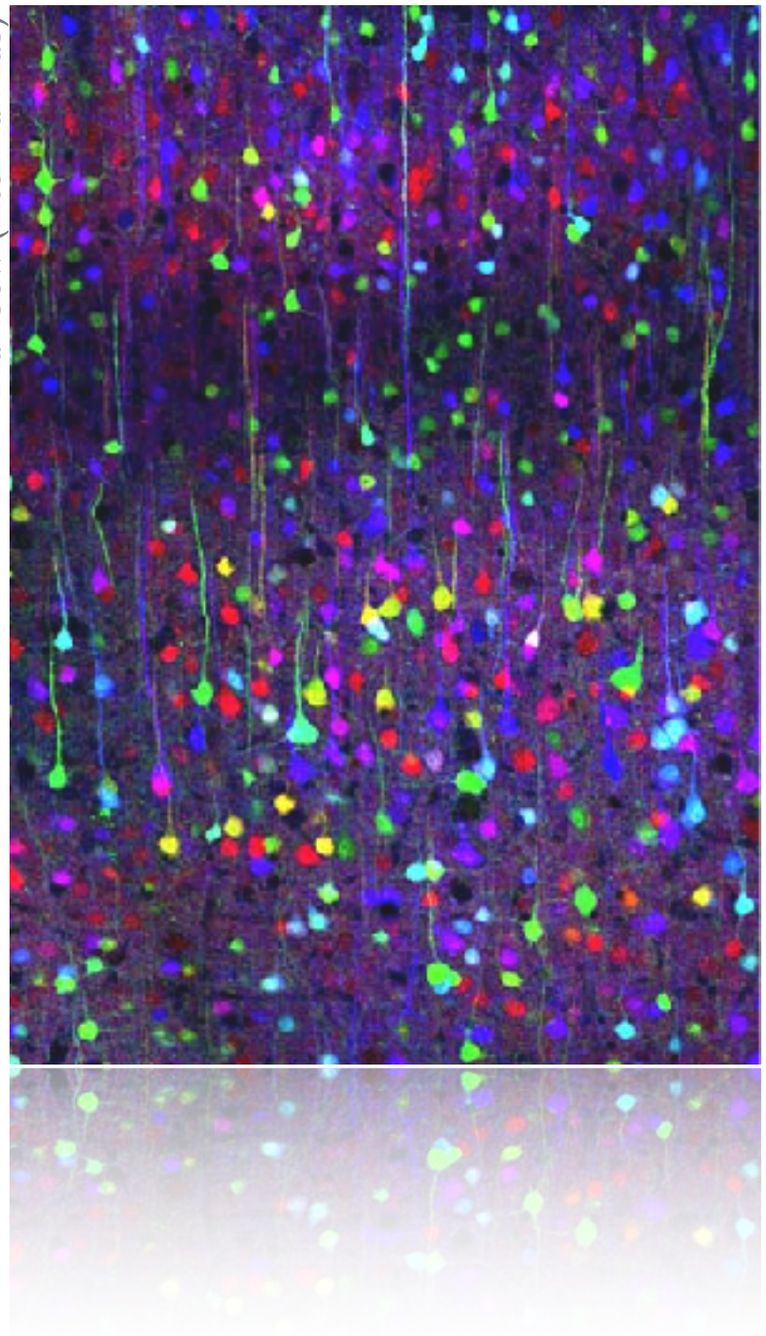
Information Processing & the Brain 2019/2020

University of BRISTOL



Brainbow (Litchman Lab)

Lecture 14

Neural circuits and learning:

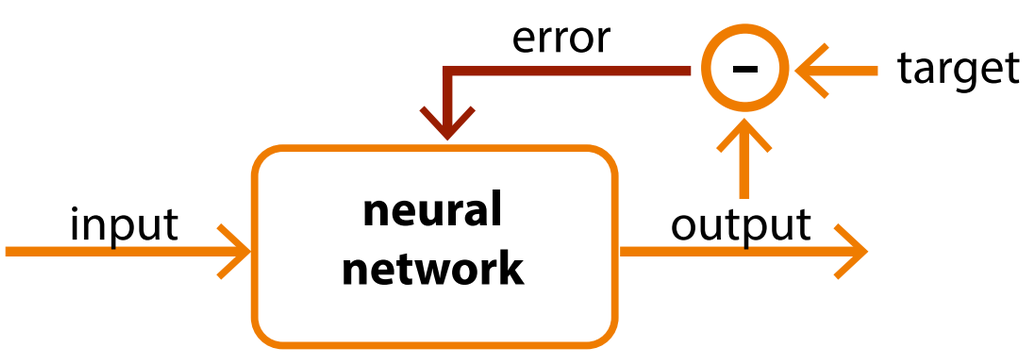
Unsupervised learning:

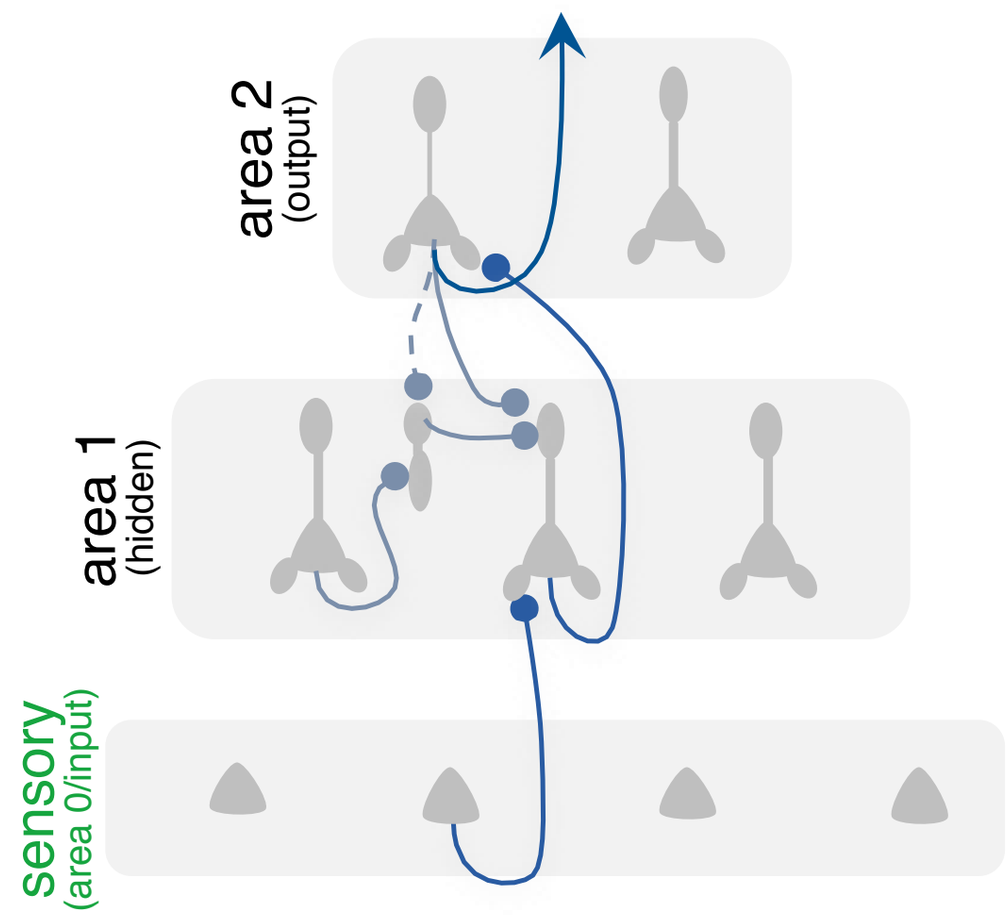
Sparse coding and autoencoders

Previously on *IPB*…

Supervised learning:

Relies on a teaching signal





But, most sensory data is unsupervised

Every few milliseconds we experience a new input!

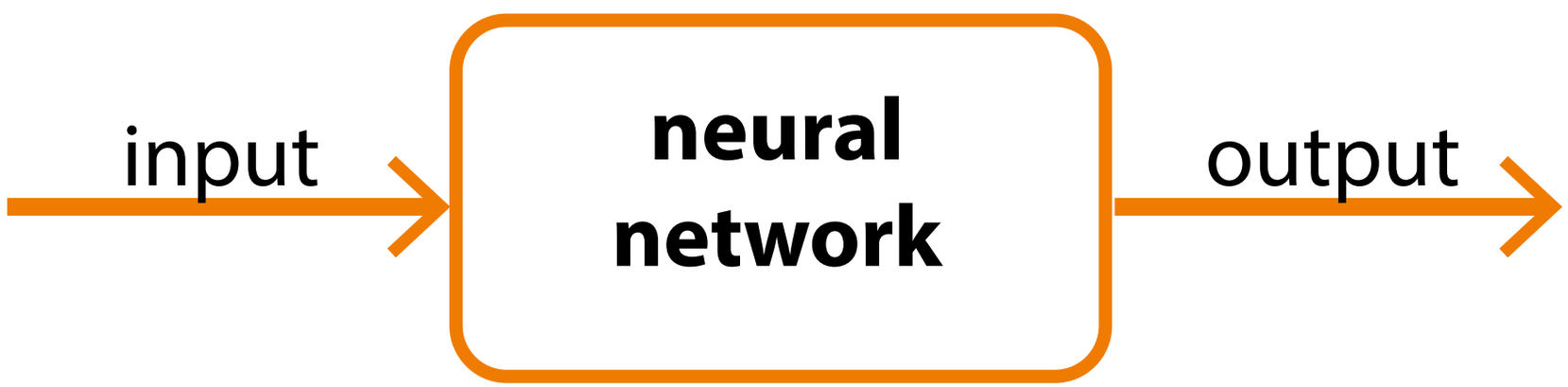
https://youtu.be/Plx9IJ1KOsg

We sense 11 million bits per second! [britannica.com]

But, most data animals experience is unsupervised

Unsupervised Learning:

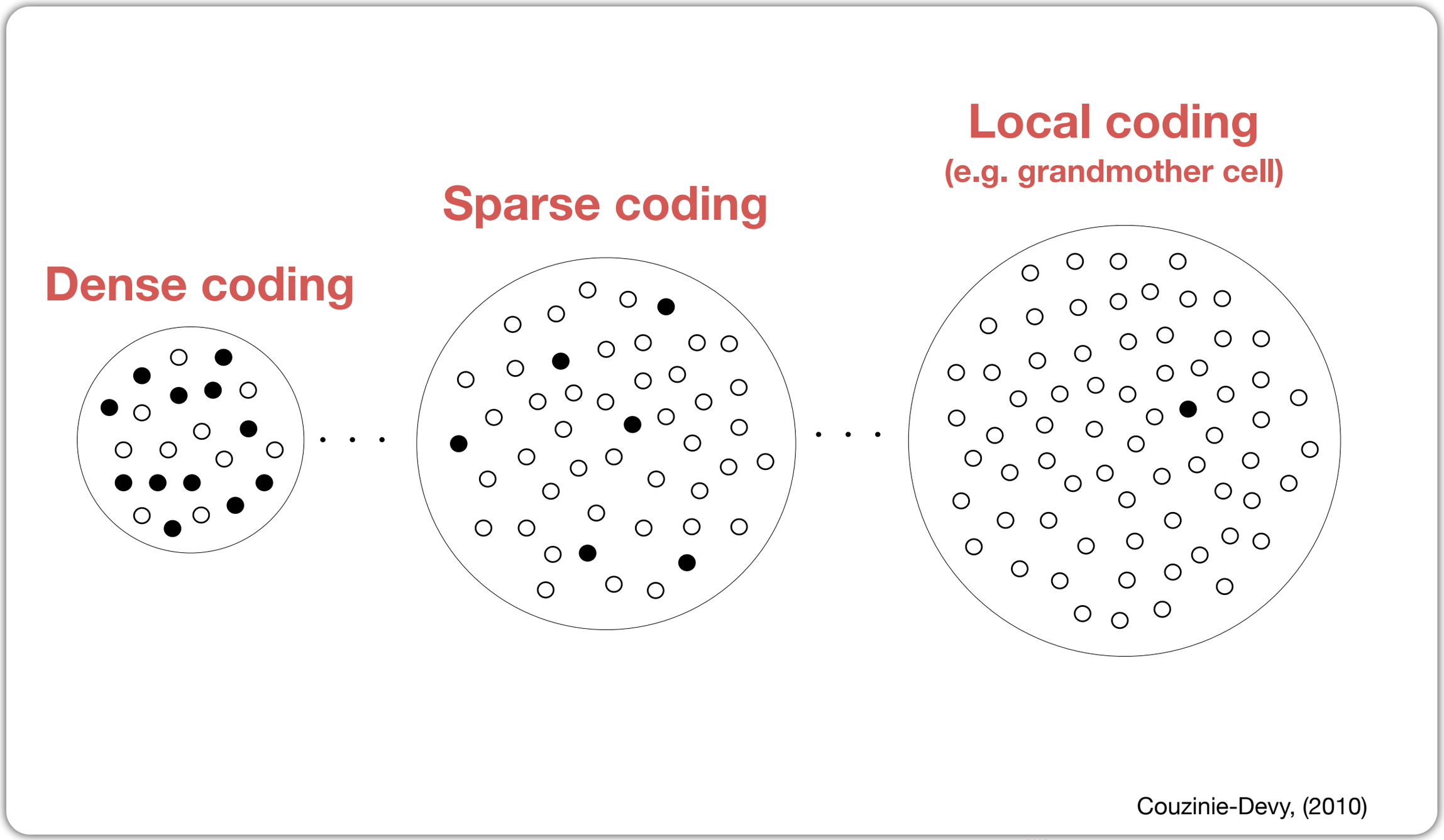
Extracts useful representations of input



Outline

1. **Sparse coding**
2. **Sparse coding in the brain**
3. **Autoencoders**

Sparse coding



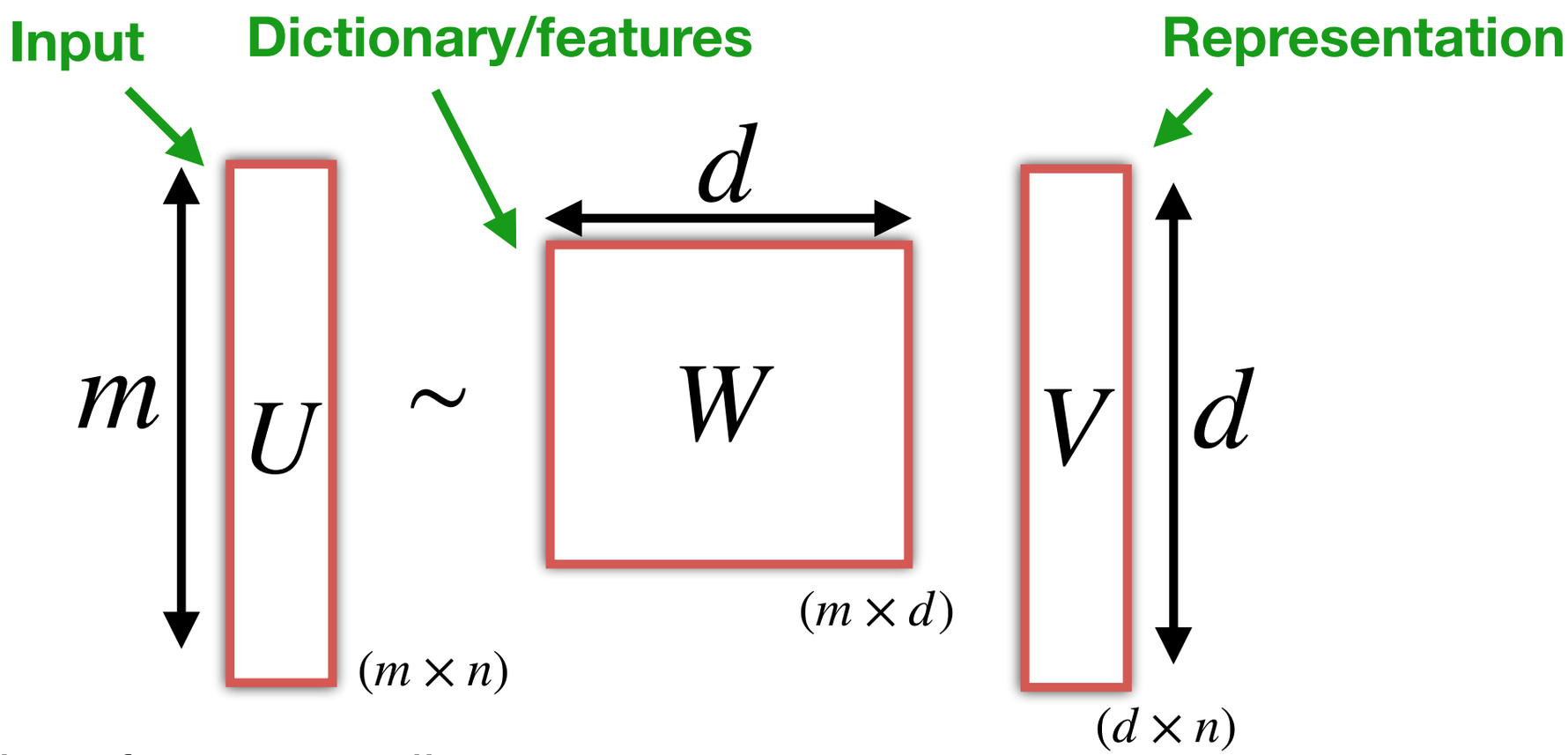
Sparse coding

- **Sparse coding** provides a framework for learning sparse representations of the input, typically through unsupervised learning.

- *Definition*: "**Sparse coding** is the representation of items by the strong activation of a relatively small set of neurons. For each stimulus, this is a different subset of all available neurons."

Foldiak and Dominik, Scholarpedia (2008)

Sparse coding



Key properties of <u>sparse coding</u>:

- **Overcomplete dictionary *W*: d>m** (i.e. more dictionary elements than input elements)
- **Sparseness**: # of zero elements in ***v*** >> # non-zero elements in ***v***
- Overcomplete and sparse representation leads to **<u>richer features</u>**

Notes:

$$\|X\|_2^2 = \sum_{i=1}^{n} |x_i|^2$$

$$\|X\|_1 = \sum_{i=1}^{n} |x_i|$$

Sparse coding
L1-norm

$$\text{cost} = [\text{preserve information}] + \lambda[\text{sparseness}]$$

reconstruction cost

enforces sparseness

Sparse coding with L1-norm:

$$\text{argmin}_{W,V} = \|U - WV\|_2^2 + \lambda\|V\|_1$$

$$\text{subject to } \|W_i\|_2^2 \leq 1 \;\; \forall_i = 1,\ldots,m$$

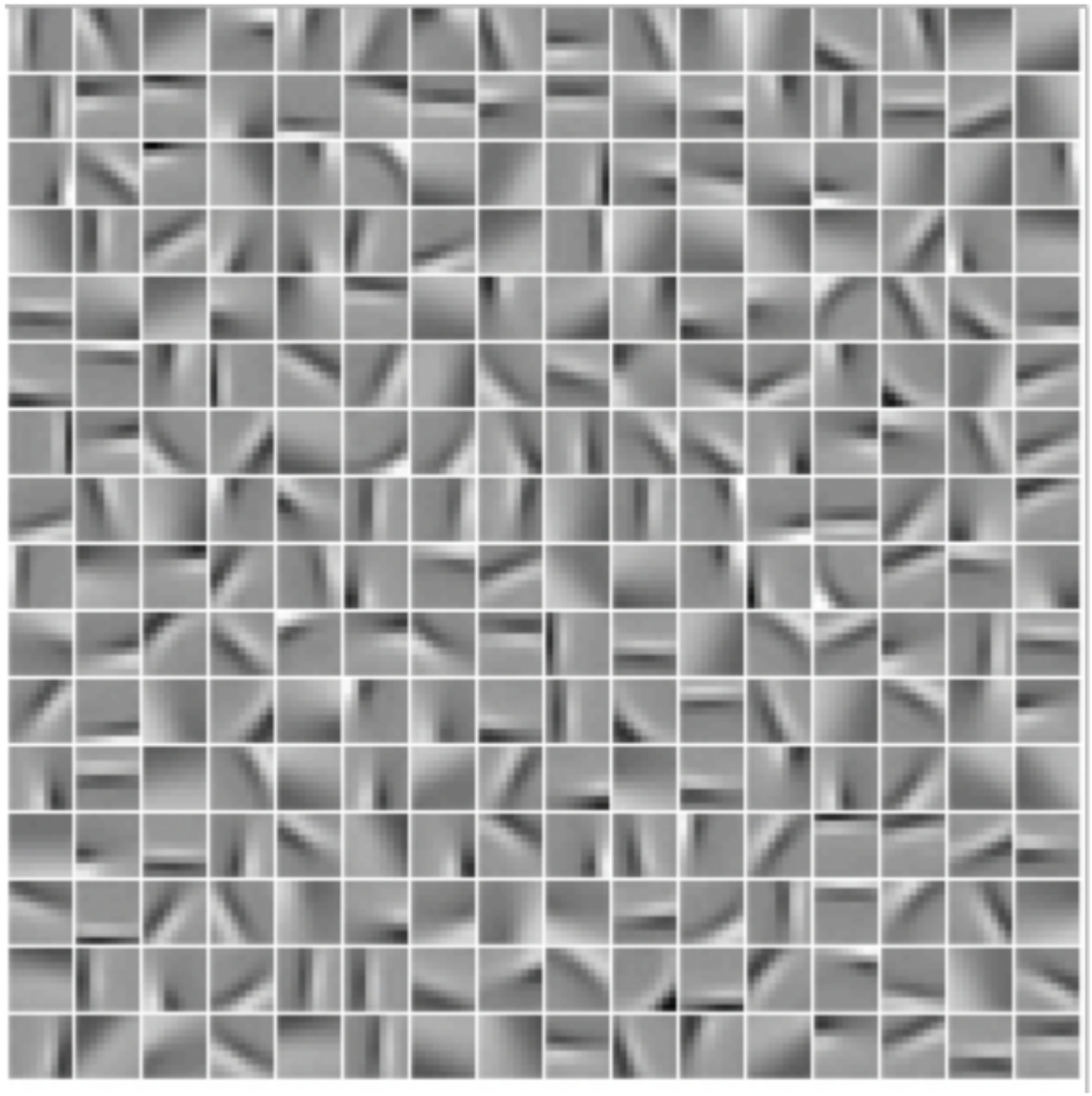
avoids arbitrarily high ***W***

- To find a good sparse code we need to optimise both the dictionary, ***W*** and the representation ***V***
- This optimisation problem is convex in ***V*** and ***W*** separately, but not both. Solution: interleaved optimisation of ***V*** and ***W***
- Several methods exist to optimise **V** and **W** (we will discuss one)
- λ controls the tradeoff between sparseness and reconstruction

Sparse coding

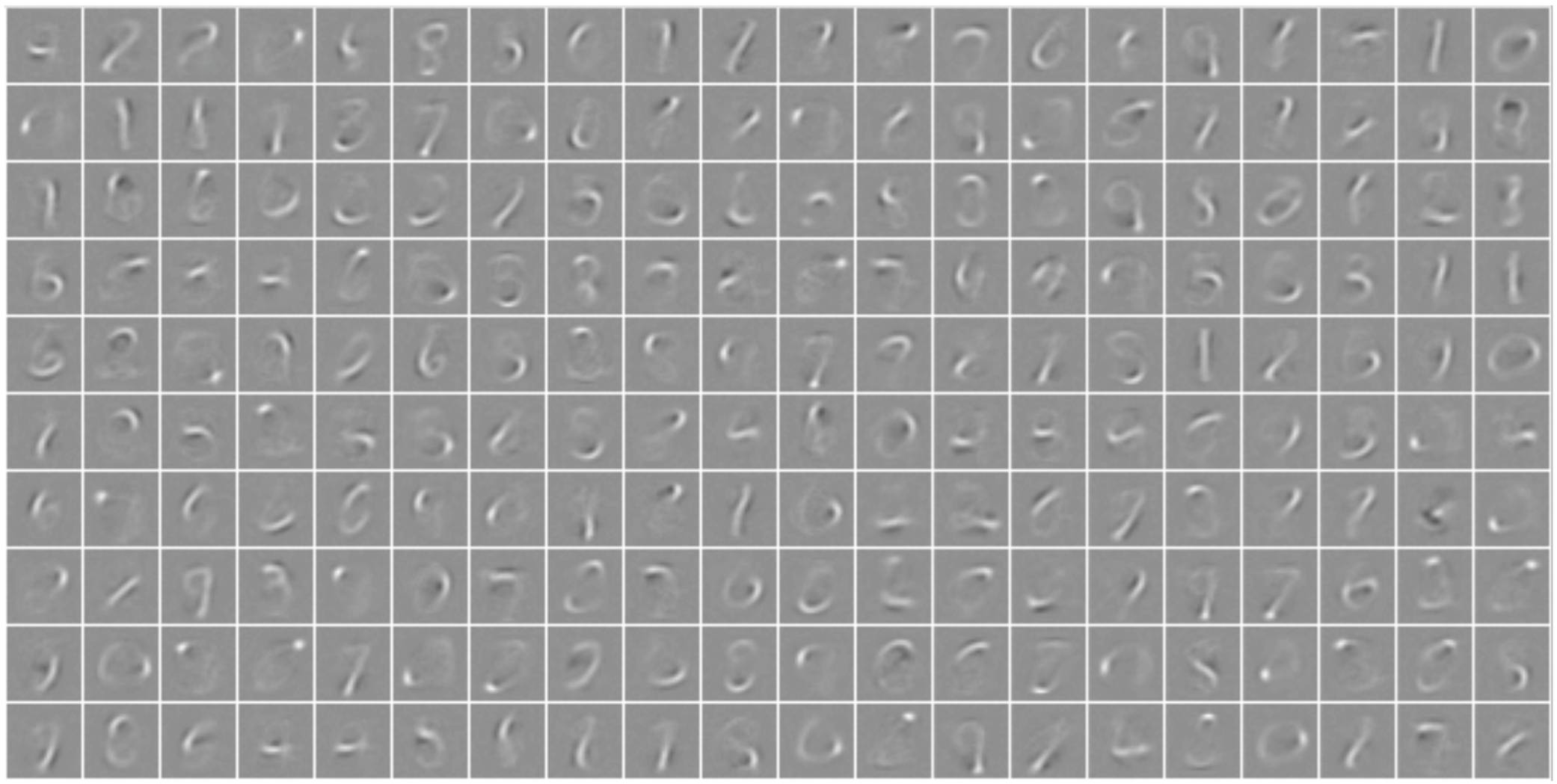
Input: Natural images

Simple cell like features:



Sparse coding

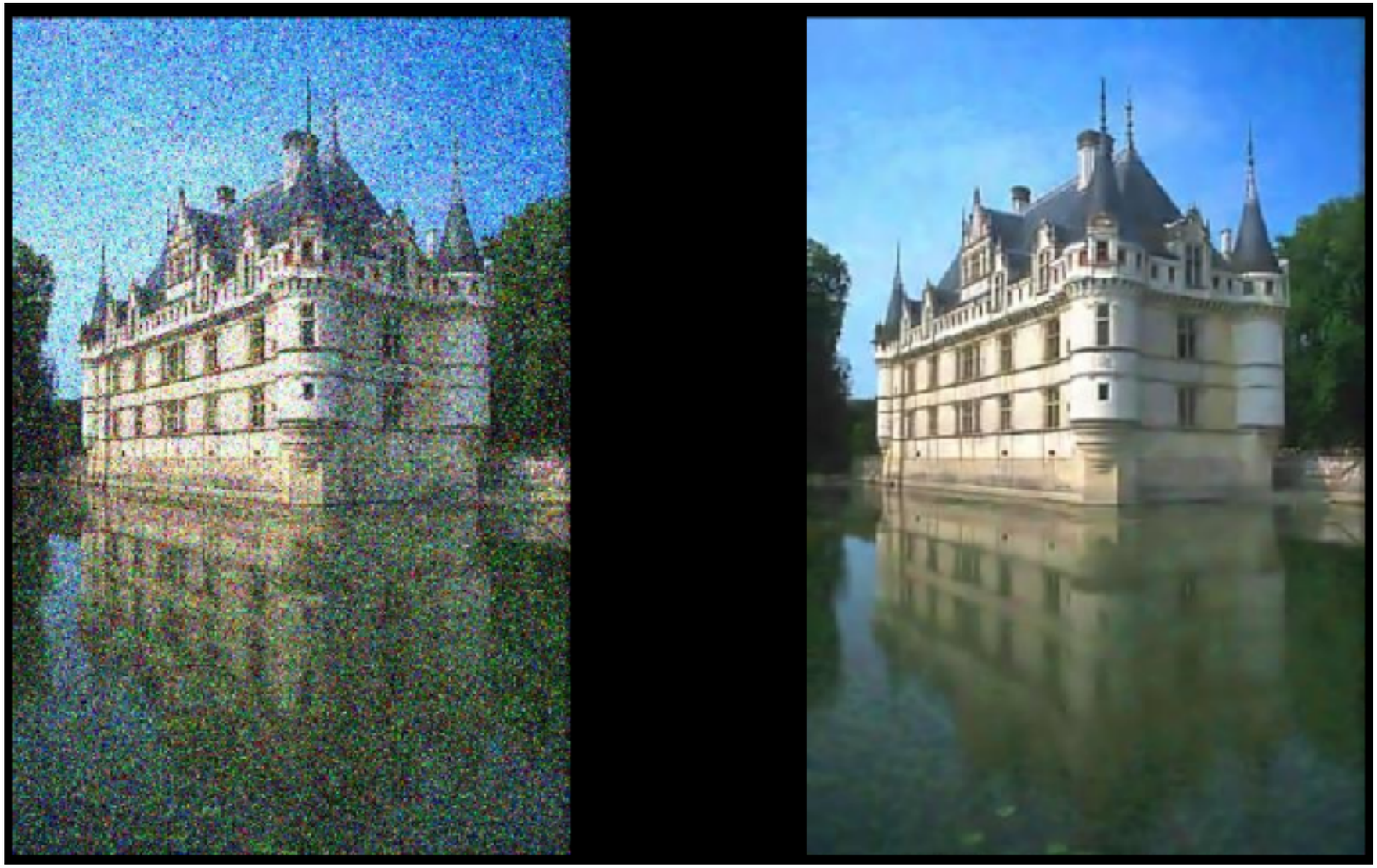
Input: MNIST dataset (handwritten digits)



Sparse coding
Applications: denoising

Original

Sparse coding method



Sparse coding
Applications: image restoration

Original

Sparse coding method

Fig. 8. Underwater. Left: Image containing many high contrast edges, with text covering 16.19% of its area. Right: restored

Richard et al. VIIP 2001

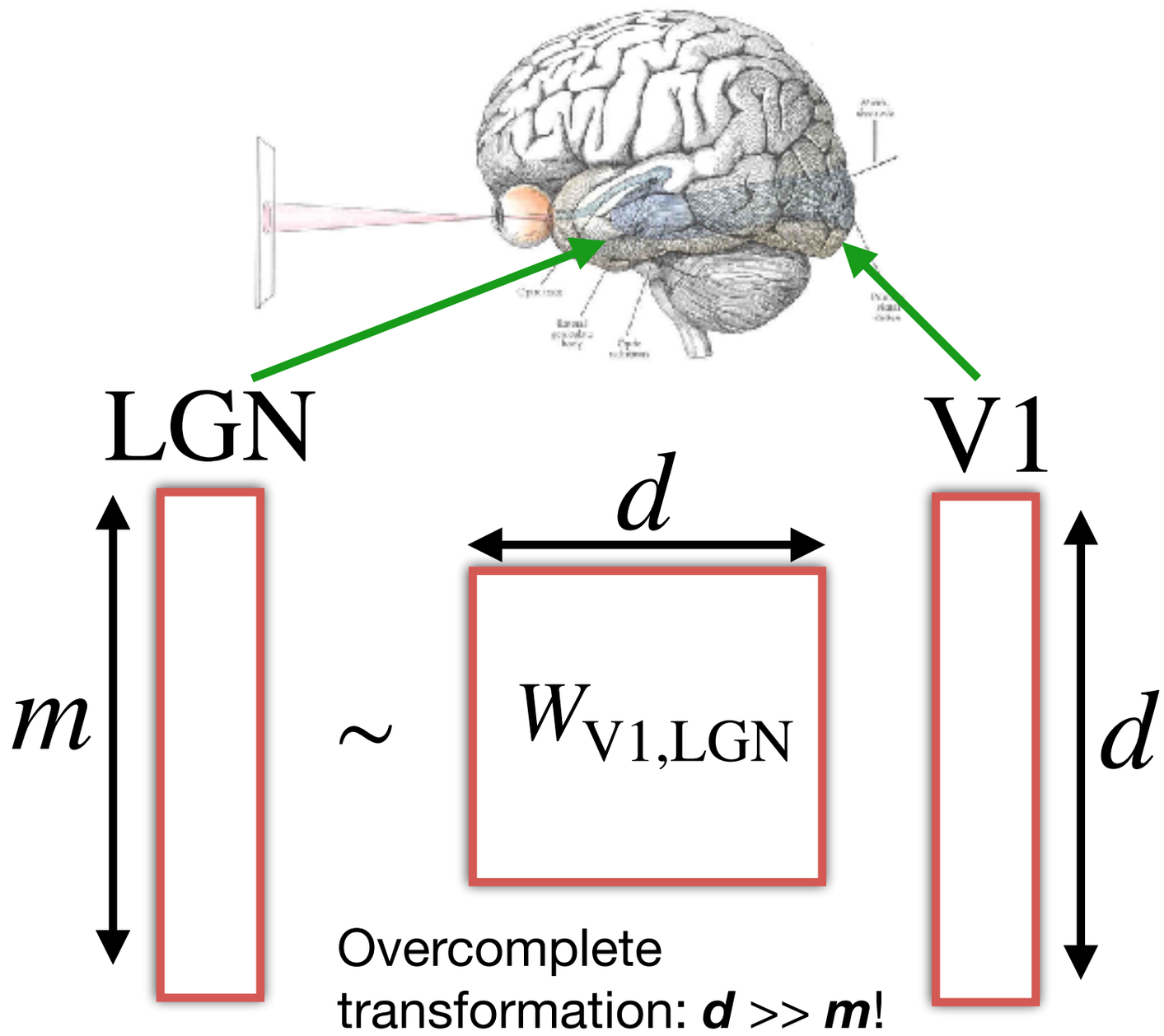
Sparse coding
Applications: digital zooming

Original

Sparse coding method

Couzinie-Devy, 2010

Sparse coding in the visual system



Barlow (1981)

Sparse coding in the brain

Olshausen and Field 1996

New cost function:

new sparseness cost

$$\text{argmin}_{W,V} = \overbrace{\|U - WV\|_2^2 + \lambda \sum_i S(v_i)}^{\text{Cost}}$$

$$S(v_i) = \log(1 + v_i^2)$$

Notation in Olshausen and Field paper:

$$\text{argmin}_{\Phi,A} = \|I - \Phi A\|_2^2 + \lambda \sum_i C(a_i)$$

Olshausen and Field, Nature (1996)

Sparse coding in the brain

Olshausen and Field 1996

First phase,

compute *v* via gradient descent:

$$\dot{v}_i = -\frac{\partial \text{Cost}}{\partial v_i}$$

$$= w_i U - \sum_{j \neq i} G_{ij} v_j - f_\lambda(v_i)$$

where

$$G_{ij} = w_i w_j$$

$$f_\lambda(v_i) = v_i + \lambda S'(v_i)$$

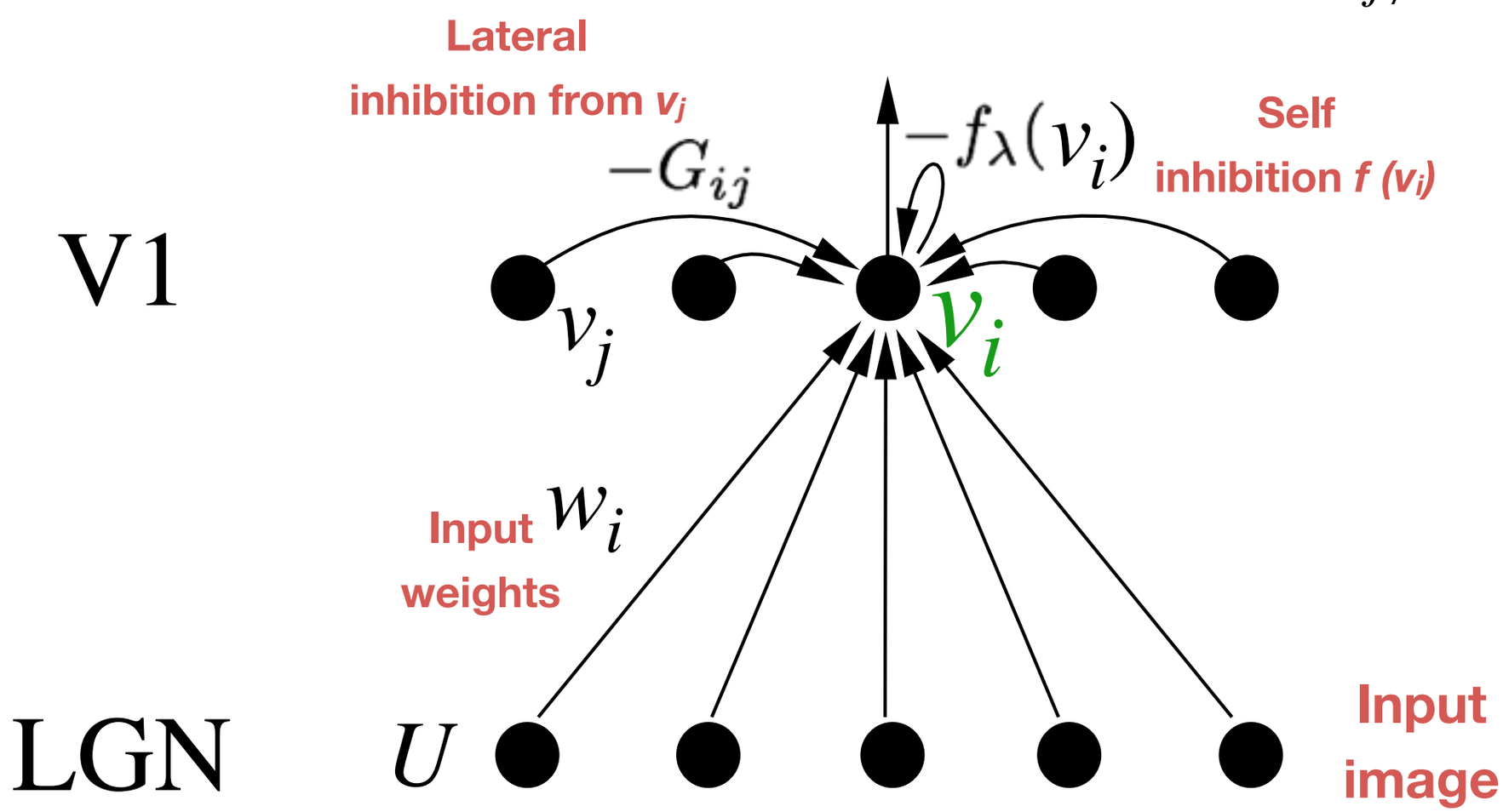
Olshausen and Field, Nature (1996)

Sparse coding in the brain

Olshausen and Field 1996

Potential network implementation of:

$$\dot{v}_i = w_i U - \sum_{j \neq i} G_{ij} v_j - f_\lambda(v_i)$$



Olshausen and Field, Nature (1996)

Sparse coding in the brain

Olshausen and Field 1996

Second phase: we need to optimise the weights as:

$$\Delta w_i = -\eta \frac{\partial \text{Cost}}{\partial w_i}$$

reconstructed image

$$= \eta < [U - \hat{U}]v_i >$$

$$= \eta < [U - \sum_{i}^{m} w_i v_i]v_i >$$

Learning rule is local, and Hebbian:

$$\Delta w \propto \underbrace{UV}_{\text{Hebbian term}} - \underbrace{WV^2}_{\text{postsynaptic term}}$$

Note: Hebbian learning rules depend on presynaptic and postsynaptic activity, e.g.:

$$\Delta w = UV$$

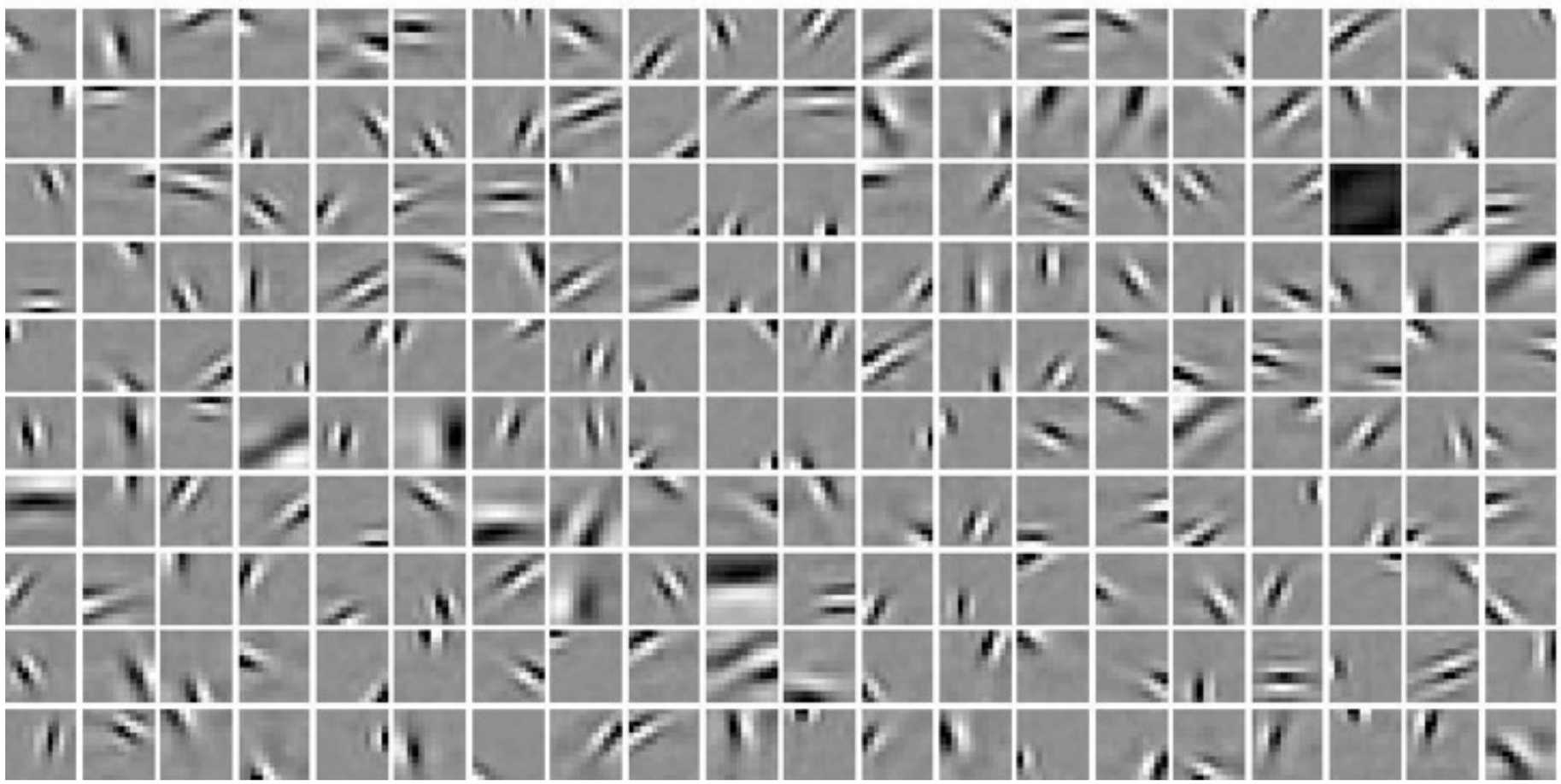
Hebbian plasticity: Cells that fire together, wire together

Olshausen and Field, Nature (1996)

Sparse coding in the brain

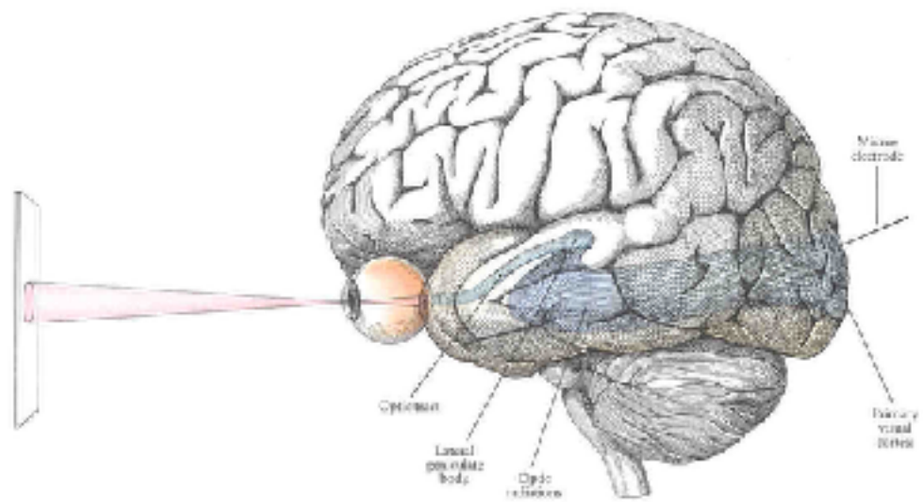
Olshausen and Field 1996

Learned dictionaries/weights (simple cell-like)**:**



Olshausen and Field, Nature (1996)

Group discussion
groups of 2-3 (5min)



Can you think of advantages of using **sparse coding** in the brain?

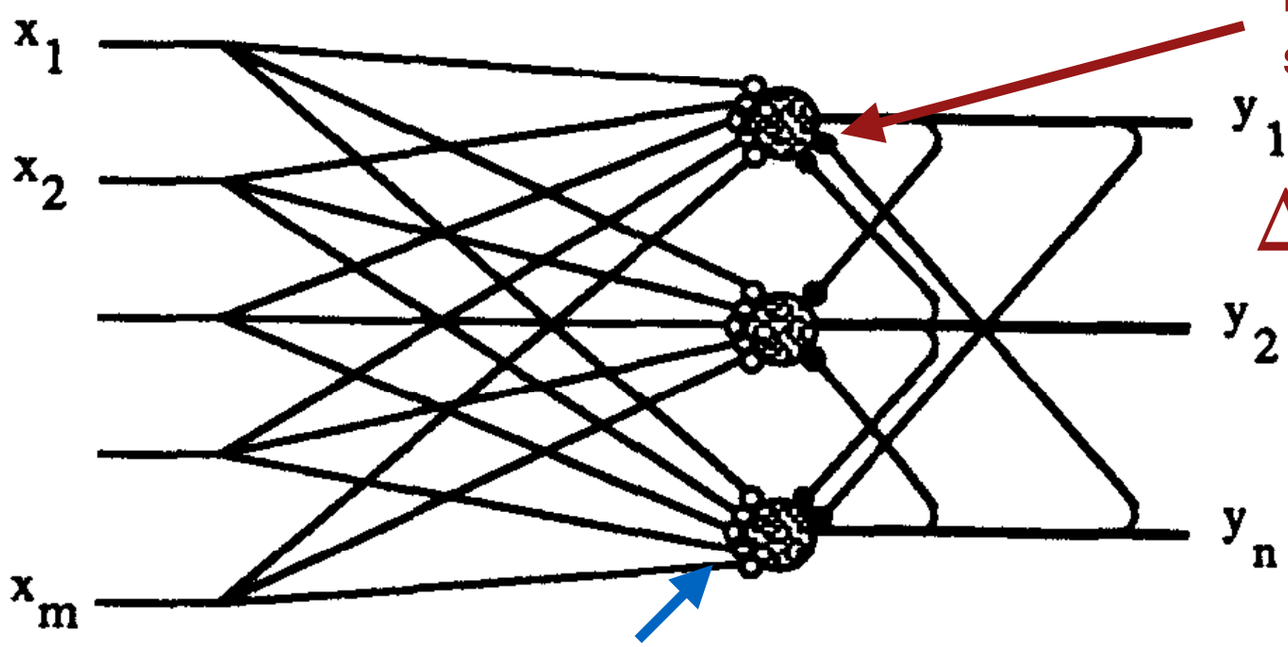
Group discussion
groups of 2-3 (5min)

Can you think of other advantages of using **sparse coding** in the brain?

- **Good representations** that can be used for decision making/ discrimination
- Sparse coding is **energy efficient** (only a few neurons active at a time)
- **More robust** to perturbations than purely localist codes

Sparse coding

With learning rules at inhibitory weights/synapses



inhibitory synapses, *w*

$$\Delta w_{ij} = -\underline{\alpha(y_i y_j} - p^2)$$

Anti-Hebbian term
(because of the minus sign)

p is the input bit probability

excitatory synapses, *q*

$$\Delta q_{ij} = \beta \underline{y_i(x_j} - q_{ij})$$

Hebbian term

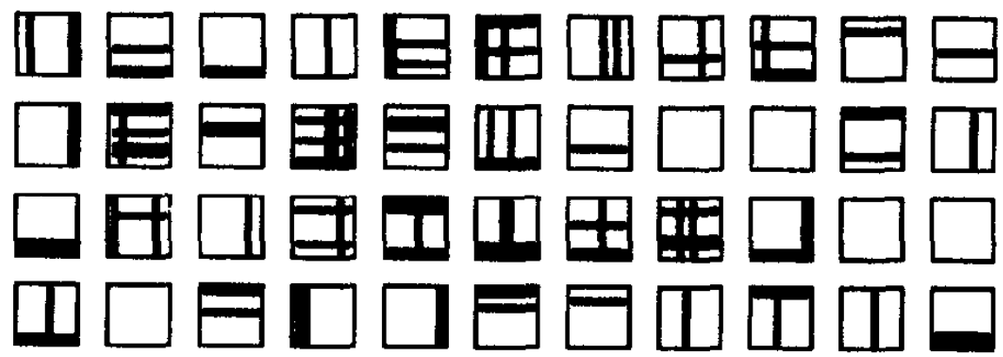
Hebbian plasticity: Cells that fire together, wire together

Foldiak, Biological Cybernetics (1990)

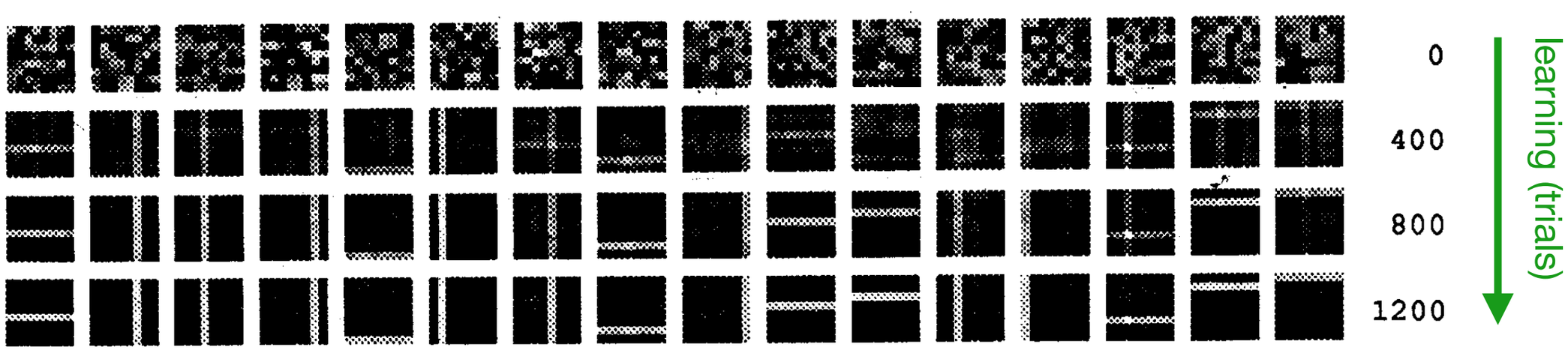
Sparse coding

With learning rules at inhibitory weights/synapses

Input defined as random patterns of vertical and horizontal lines:



Learned features:

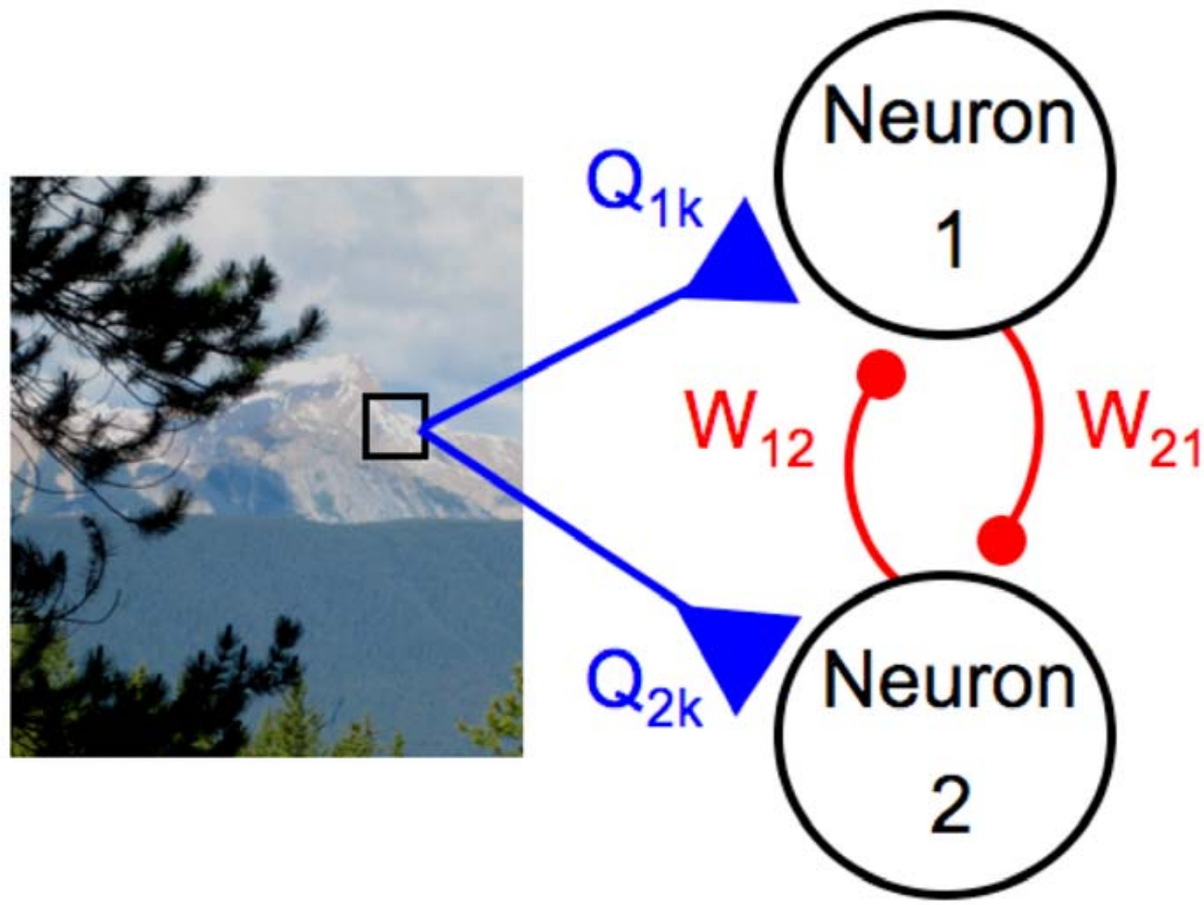


Foldiak, Biological Cybernetics (1990)

Sparse coding

Spiking neural network: SAILnet

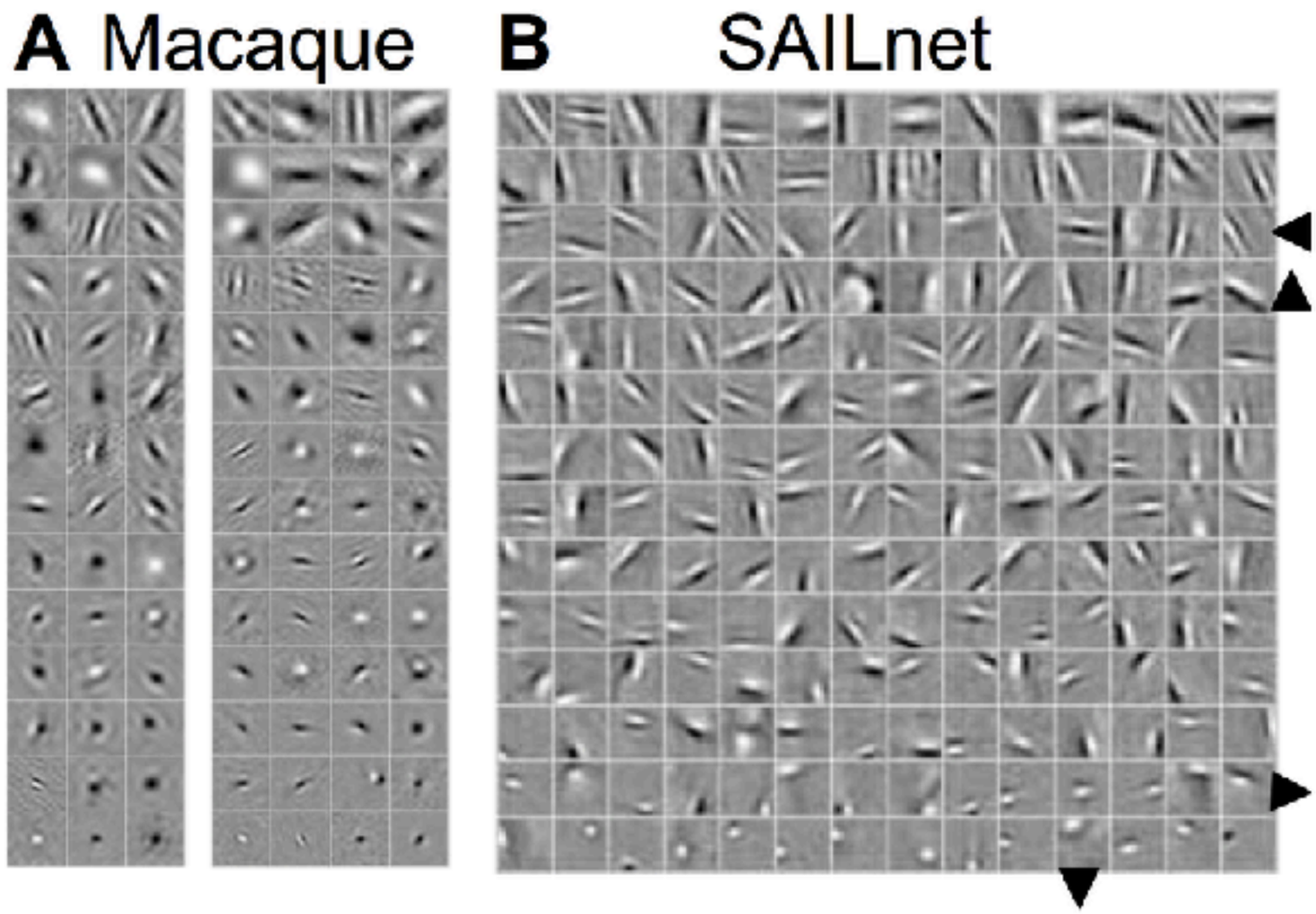
SAILnet:
A spiking version of Foldiak 1990 network with **inhibitory** and **excitatory** synapses and local learning rules:



Zylberberg et al. PLoS CompBIo (2011)

Sparse coding
SAILnet

Captures the diversity of receptive fields observed in animals:

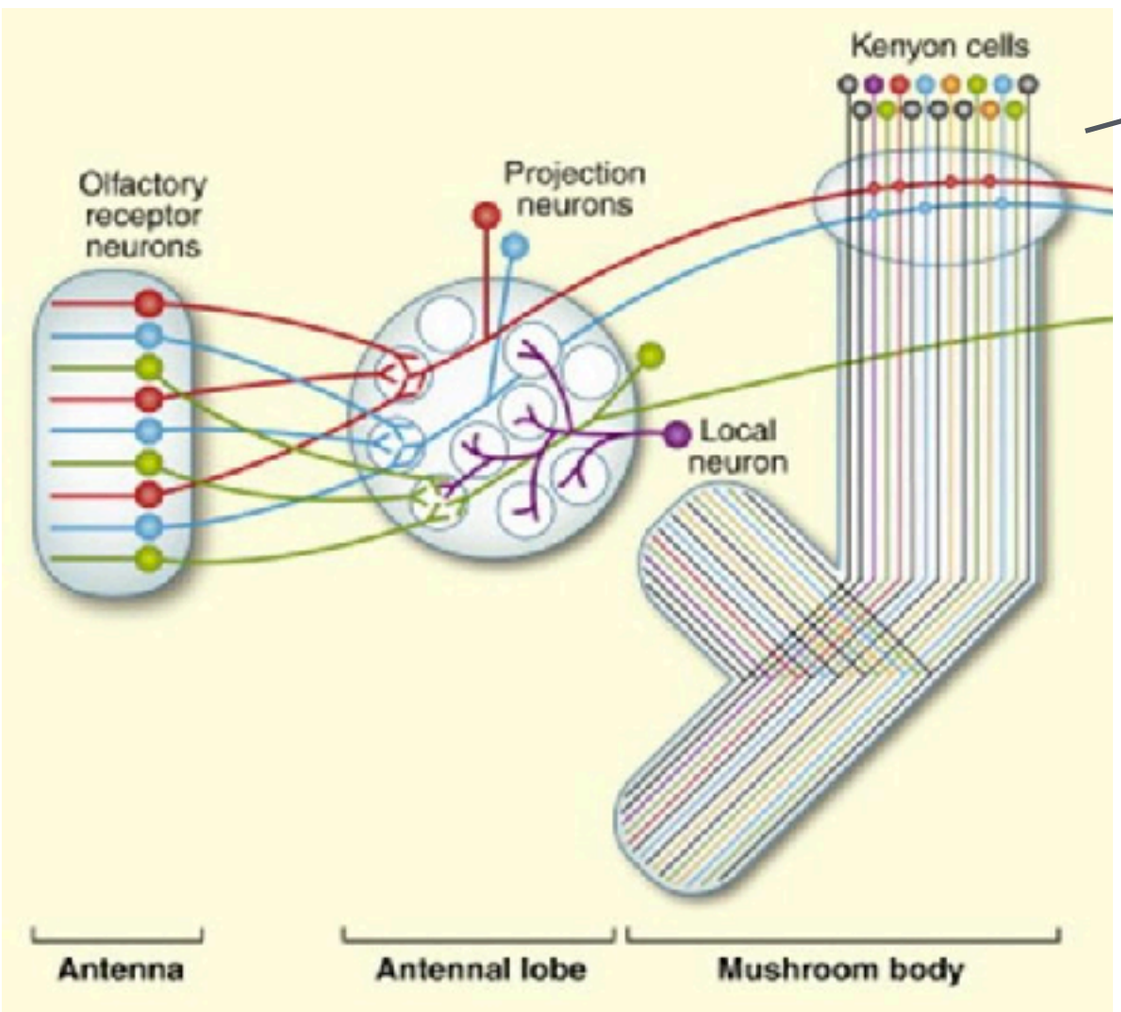


Zylberberg et al. PLoS CompBIo (2011)

Sparse coding across the animal kingdom:
from mammals to insets..

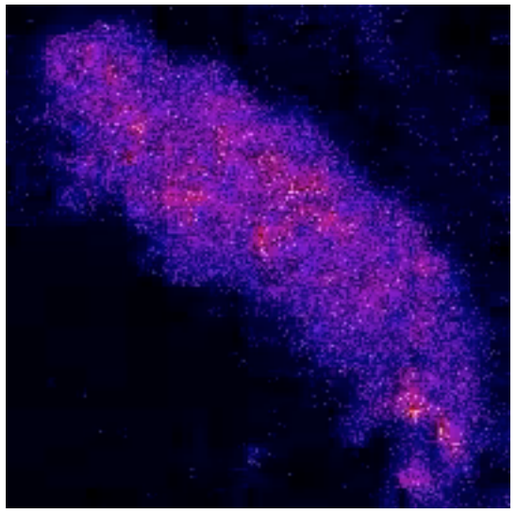
Sparse coding in fruit flies

Smell (odor) system in *Drosophila*

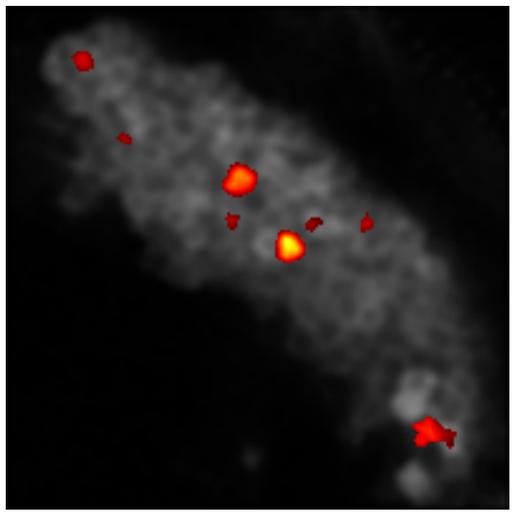


Masse et al. Current Biology (2009)

Sparse coding in Kenyon cells

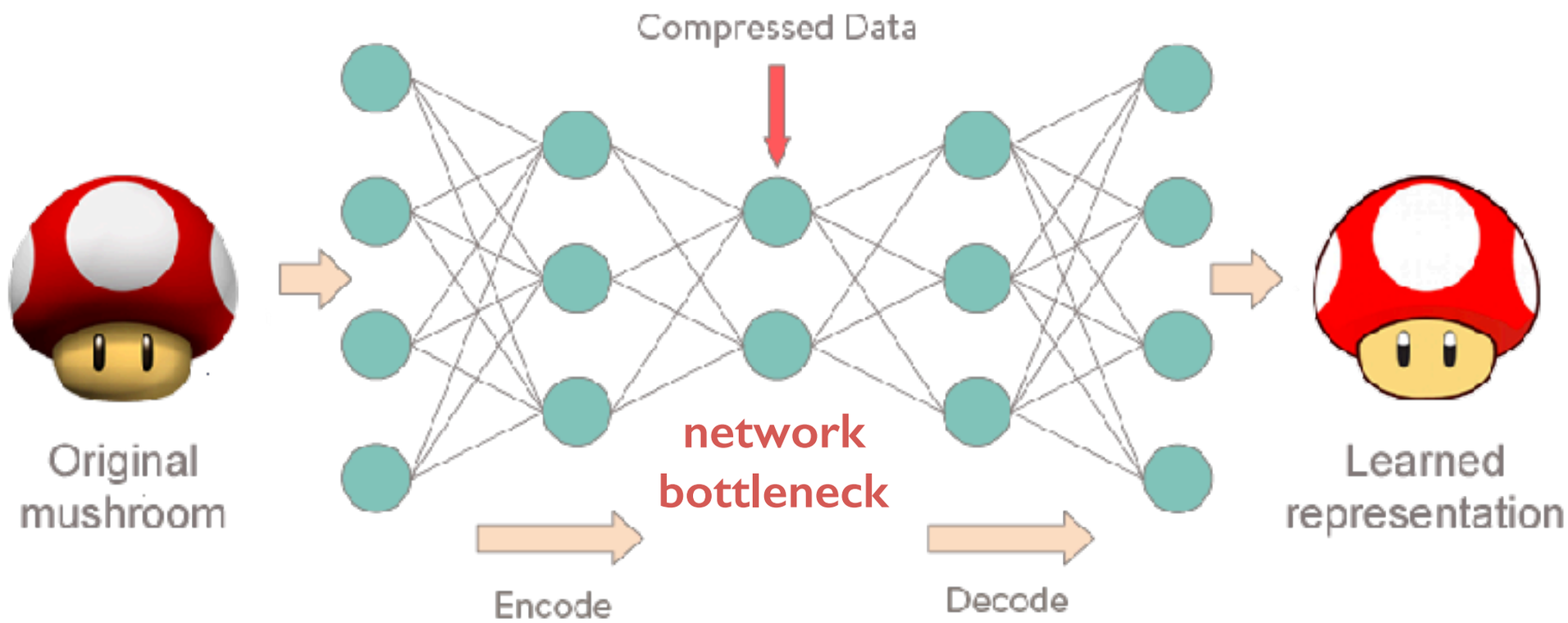


[video of neurons getting active in response to presentation of an odor]



Lin et al. Nature Neuroscience (2014)

Autoencoders



curiously.com

$$\text{argmin}_W = \|V - U\|_2^2$$

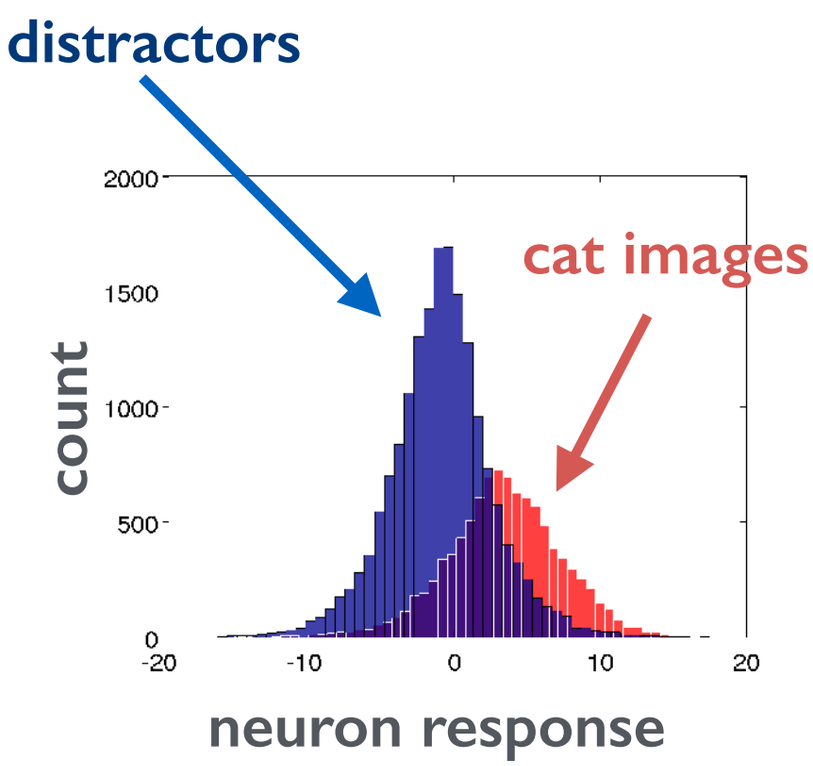
- **Autoencoders** also aim to reconstruct the input, but without an explicit constrain on sparsity (instead uses other forms of regularization)
- **Autoencoders** rely on a specific model *f* (i.e. neural network with bottleneck)
- An **autoencoders** using a sparsity constrain is doing **sparse coding**

Autoencoders

Cat neuron

Autoencoder trained on **10 million images** from **YouTube** movies.

Showing that it is possible to learn high level feature detectors from unlabelled data (akin to grandmother cells).



Le et al. ICML (2012)

Combining unsupervised with supervised learning?

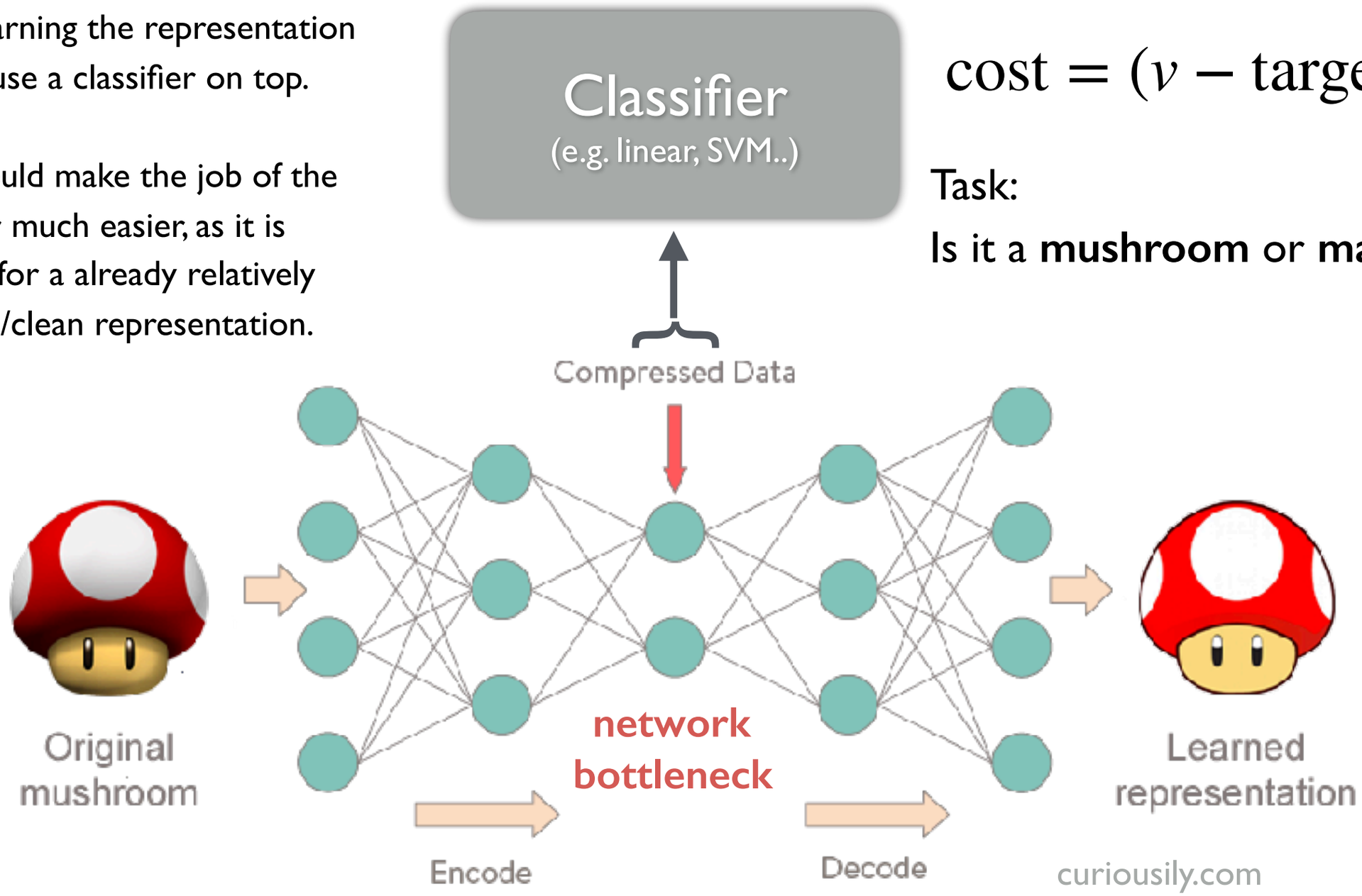
After learning the representation we can use a classifier on top.

This should make the job of the classifier much easier, as it is starting for a already relatively separate/clean representation.

$$\text{cost} = (v - \text{target})^2$$

Task:

Is it a **mushroom** or **mario**?



Summary

1. **Sparse coding** is an **optimisation** method that develops useful representations of the input
2. Olshausen and Field introduced a sparse coding framework that can be mapped onto the brain
3. **Sparse coding** develops **representations akin** to the brain
4. Autoencoders can also be used to extract useful representations, but do not (typically) rely on a sparsity cost and assume a specific neural network

References

Text books:

Natural Image Statistics, by Hyvarinen et al. 2009 [good reference on unsupervised learning from natural images]

Theoretical neuroscience: Dayan and Abbott 2001 [Unsupervised learning: chapter 3]

Deep Learning by Courville et al. 2015

Relevant papers:

- Foldiak, Biological Cybernetics (1990) [classical work on developing sparse representations with local Hebbian learning rules]
- Olshausen and Field, Nature (1996) [seminal work on sparse coding using natural images]
- Zylberberg et al., PLoS Computational Biology (2001) [more recent work on sparse coding in spiking networks with local learning rules]

Upcoming lectures

- L11[6]: Neural circuits and learning: introduction
- L12[6]: Visual system: deep learning?
- L13[7]: Supervised learning & backprop
- L14[7]: Unsupervised learning
- **L15[9]: Reinforcement learning**
- L16[9]: Temporal processing & recurrent neural networks
- L17+18[10]: RNNs and brain vs machine
- L19-20[11]: Neural data analysis (by Cian)
- L21-22[12]: Revision week