No Backprop, please!
Learning of ‘deep’ representations

Wulfram Gerstner

EPFL, Lausanne, Switzerland

Memories

are available
because of

Learning

Learning actions:

→ riding a bicycle

Remembering facts

→ previous president of the US

Remembering episodes

→ first day at college/university

Remembering ‘objects’

Close your eyes: imagine a tree!

Literature:

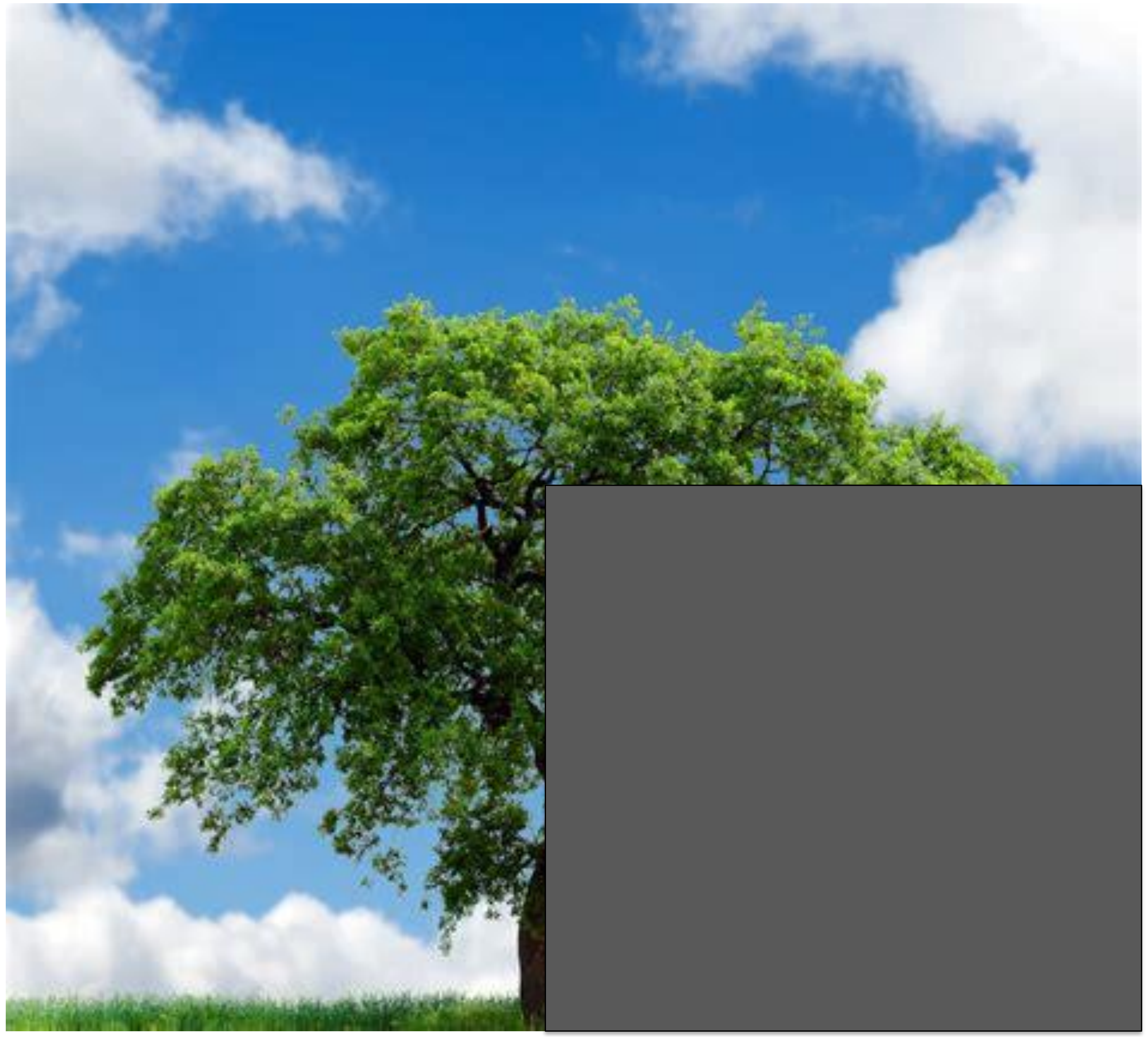
Timothy P. Lillicrap et al., Backpropagation and the brain, Nature Reviews Neurosci. 21: 335-346 (2020)

Bernd Illing et al., NeurIPS (2021), Local Plasticity rules can learn deep representations, 35th NeurIPS (2021)

Previous slide.
The first two section formulate the question in the context of what we have seen in earlier weeks. The question is how to learn deep representations with biologically plausible learning rules. An important novel aspect is the capacity of the brain to predict.

Humans can learn to predict a ‘tree image’

https://www.publicdomainpictures.net/pictures/90000/velka/crab-apple-blossom-tree.jpg



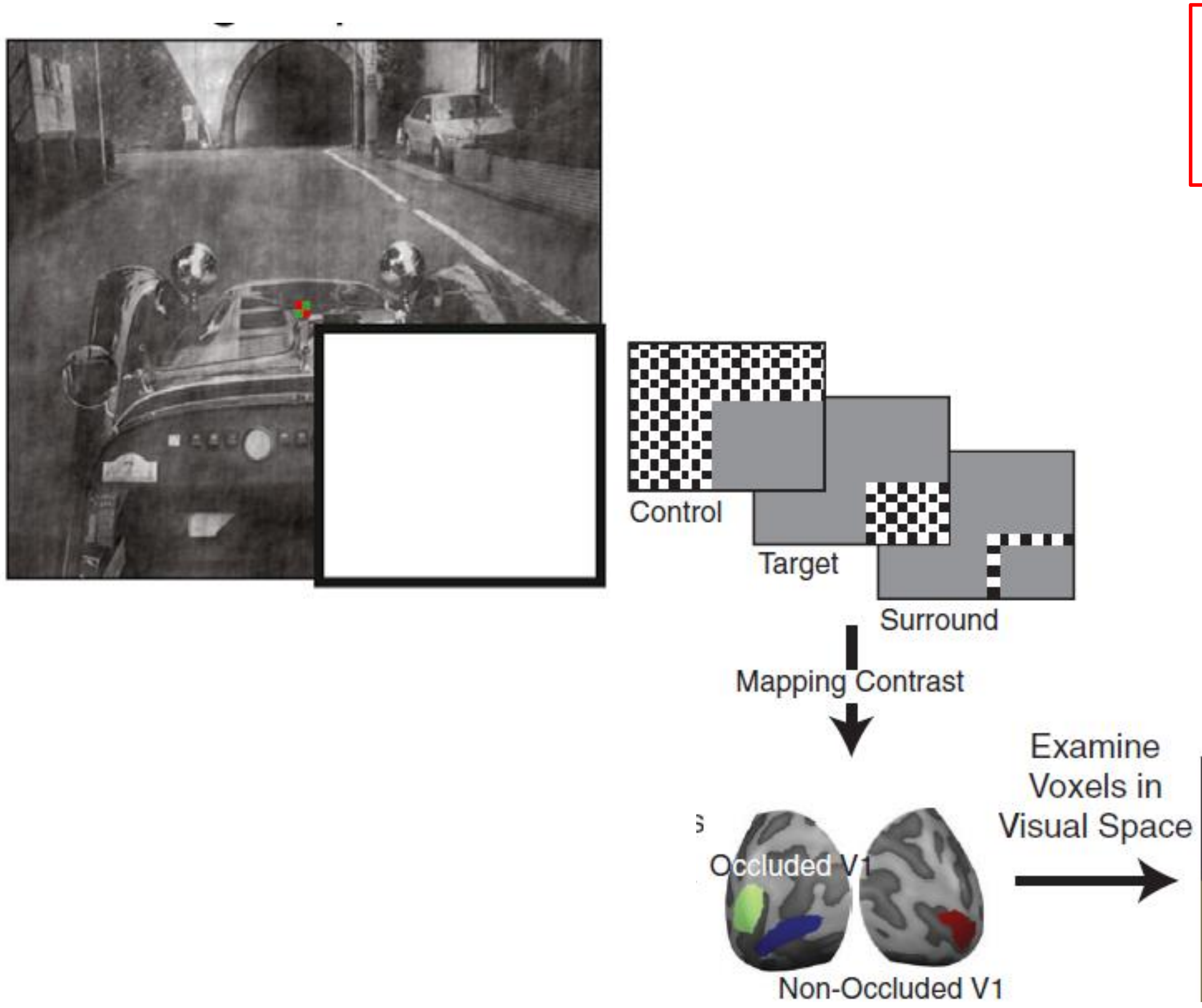
https://www.publicdomainpictures.net/pictures/250000/velka/tree-1524116693MIN.jpg

Previous slide.
On the left: we are able to imagine a tree ‘from scratch’ just triggered by the word tree.
The resulting image is not unique.

On the right: it is easy to imagine the covered parts of the tree.

fMRI experiments: spatial prediction

→Brain is able to predict missing parts



V1 receives feedback input when not directly stimulated

Contextual feedback to superficial layers of V1
L. Muckli et al., Current Biol. 25: 2690–2695 (2015)

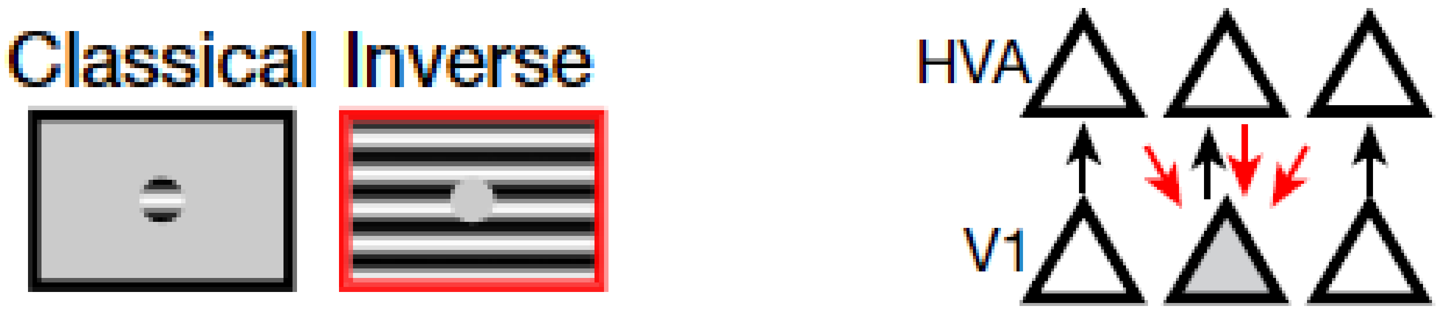
Morgan, Petro, Muckli, J. Neurosci. (2019)
Savanera, … Muckli, J. Vision (2021)

Previous slide.
fMRI (functional magnetic resonance imaging) shows brain regions that are active during a given task.

An image is represented in several brain areas, but the area V1 is the one where we understand the representation very well: First, it responds strongly to high contrast stripes or checkerboards. Second, it shows spatial organization:
The experimentalist Lars Muckli and colleagues first showed a checkerboard pattern with a grey rectangle – since neurons in area V1 respond to the checkerboard pattern it is possible to find out which neurons respond to the region of the checkerboard. And the he inversed the pattern to find out which neurons respond to the lower right corner.

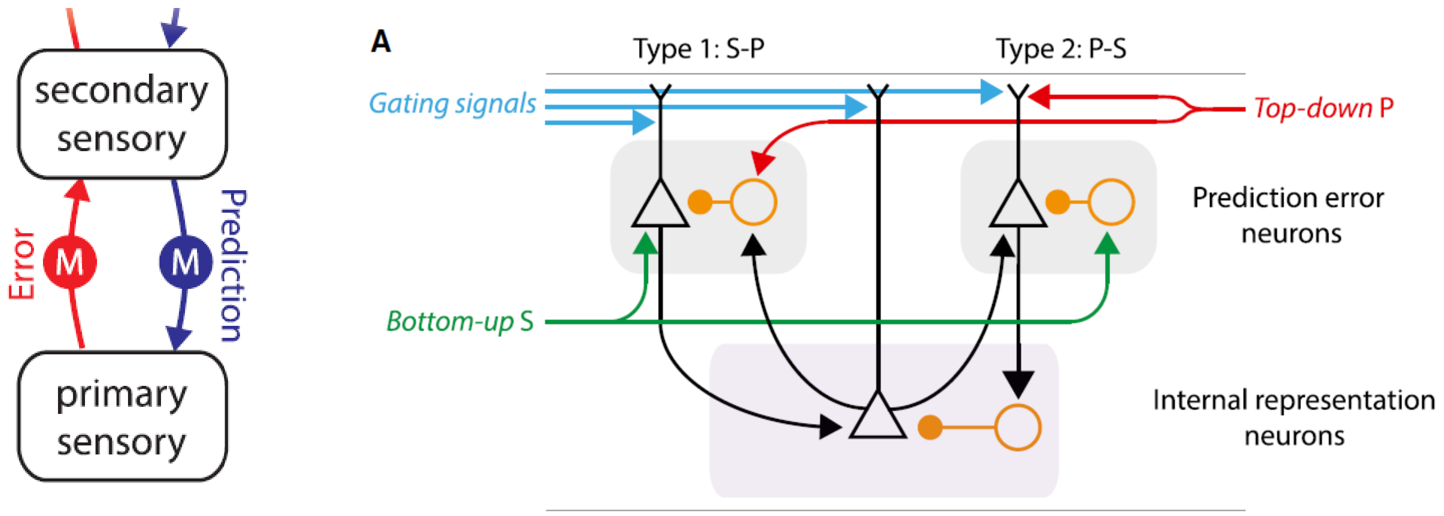
He then used a series of images (like the one with a car) where always the lower right corner was covered. I found that neurons in area V1 that belong to the covered region are active – and an obvious explanation is that these neurons receive either later input from the same area or feedback input from higher areas.

Predictions via Lateral and Feedback Connections



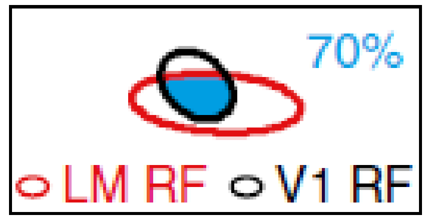
Feedback generates a second receptive field
A. Keller, Roth, Scanziani**.** Nature **582**: 545–549 (2020)

Feedback exists!
used for predictions



Predictive Processing: a canonical cortical computation.
GB Keller, TD Mrsic-Flogel, Neuron:424-435 (2018)

Predictions can
be observed!



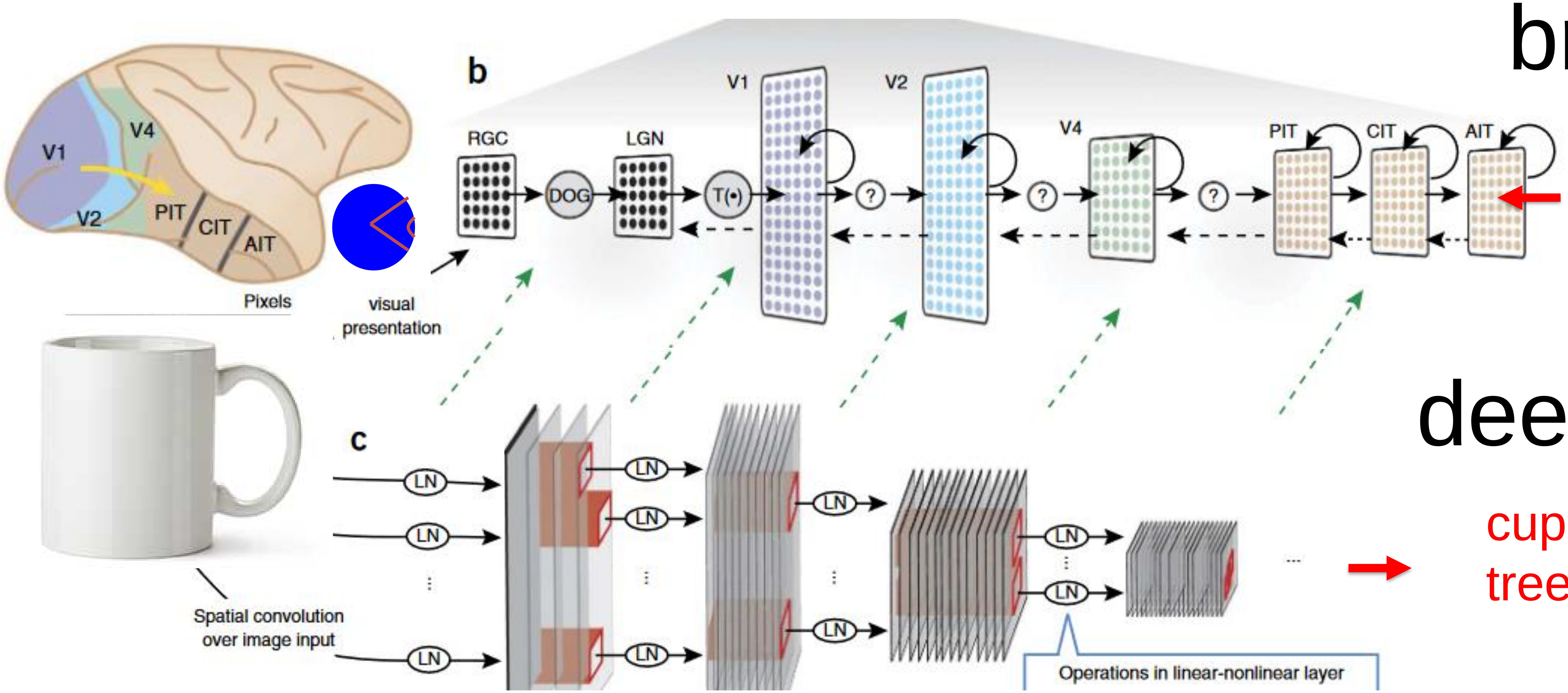
The functional organization of cortical feedback
Marques … Petreanu, Nat. Neurosci. 21:757-764 (2018)

Spatial context important!
→Feedback from wider area

Previous slide.
In animals, it is possible to see observe similar phenomena on a neuron-by-neuron basis. Feedback connections from higher areas (top) or from lateral neurons in the same area (bottom) send input, and such input has predictive properties (middle).

Deep representation in brain models

brain model



“deep”
representation

deep network

cup, mug, bird,
tree, car, airplane …

1 Cortical Area → 1 layer of a Deep Convolutional Neural Network

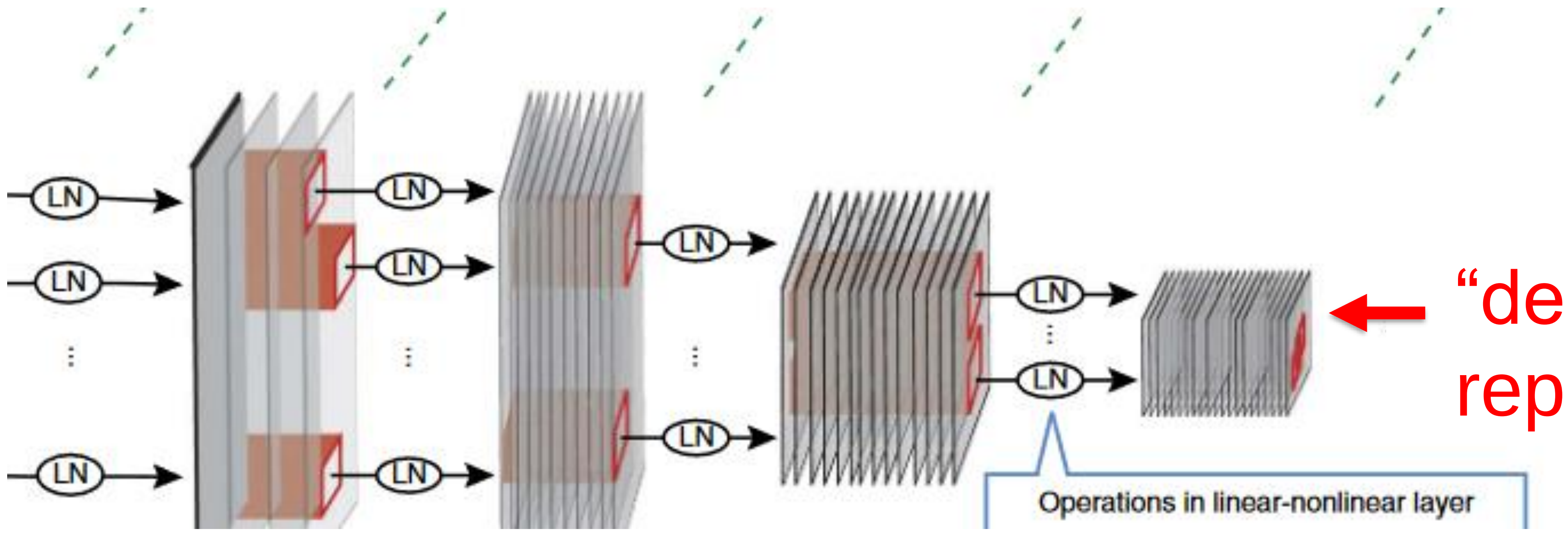
Train on classification of Mio of images with BackProp (**supervised learning**)
BUT: - Real networks do not use BackProp

Using goal-driven deep learning to understand sensory cortex
D Yamins, JH DiCarlo Nat. Neurosci.19: 356-365 (2016)

Previous slide.
This is a repetition from week 1: information (such as the image of a coffee mug) enters through the retina. Its first cortical processing state is V1, then V2, V4, and three areas within inferotemporal cortex (IT).

Today, on of the best models of the neuronal activity in these areas is a deep neural network trained with BackProp on a supervised learning task (ImageNet)

Artificial Neural Networks: self-supervised learning



"deep" representation

- **masked auto-encoders**:
  learn to fill in missing parts
- **contrastive learning:**
  image patch → image patch
  prediction possible for 'same' image,
  but impossible for 'different' image

Contrastive Predictive Coding (CPC), Van den Oordt et al. 2018

Previous slide.
Alternatively, in the absence of image labels, there are self-supervised learning paradigms that achieve the same performance.
For example, a partially covered image is presented and the task is to predict a full image at the output: this is the example of self-supervised learning of an auto-encoder.

Other variants of self-supervised learning exist and we will talk about these later today.

Learning in Artificial Neural Networks

Deep Networks for Vision
(e.g. AlexNet …)

Sutskever and Hinton, 2012

→ trained with BackProp

Deep Networks for Chess and Go
(alpha-go, alpha-zero)

Silver et al. (2017) , Deep Mind

→ trained with BackProp

Foundation Models
(LLMs, ChatGPT, Bert)

→ trained with BackProp

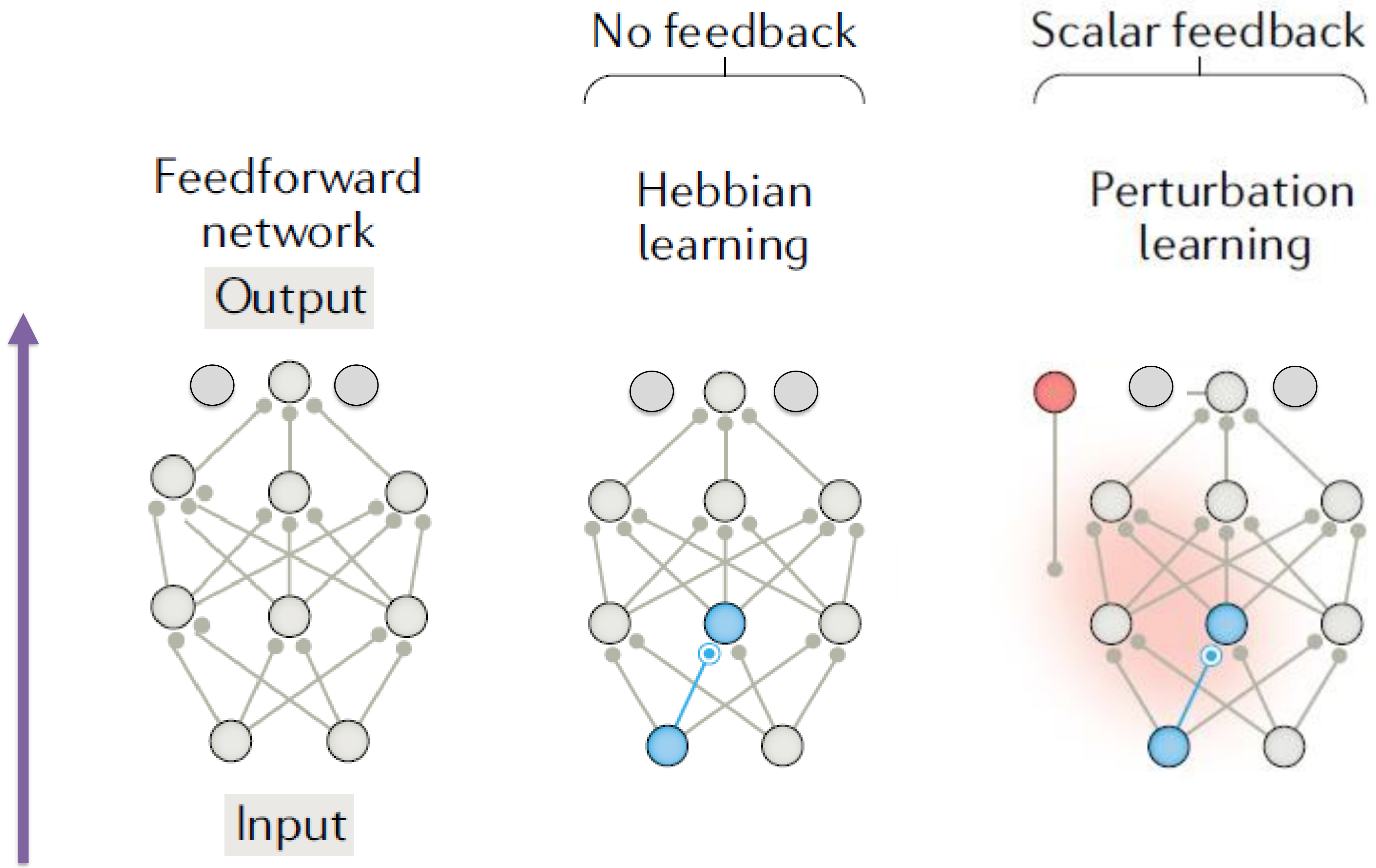
Autoencoders/contrastive learning/self-supervised learning

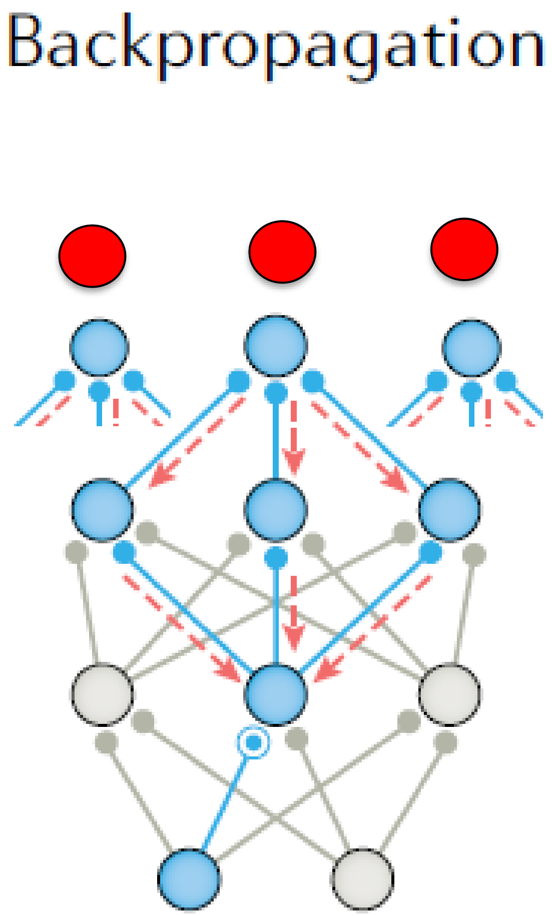
→ trained with BackProp

Previous slide.

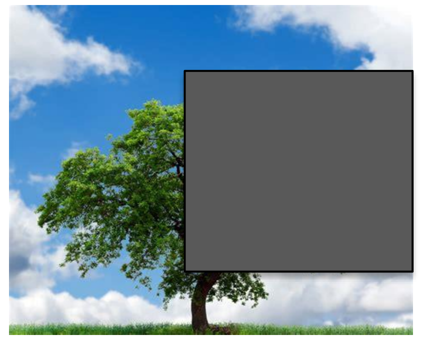
All the famous models in AI/Deep Learning are trained with Backprop.

A Spectrum of Learning Algorithms: connections change based on …





Adapted from Lillicrap et al. 2020 Nat. Rev. Neurosci.



2-factor rules:
unsupervised,
no feedback

3-factor rules:
reward-based,
feedback

BackProp rules:
vector feedback
(many local 'errors')

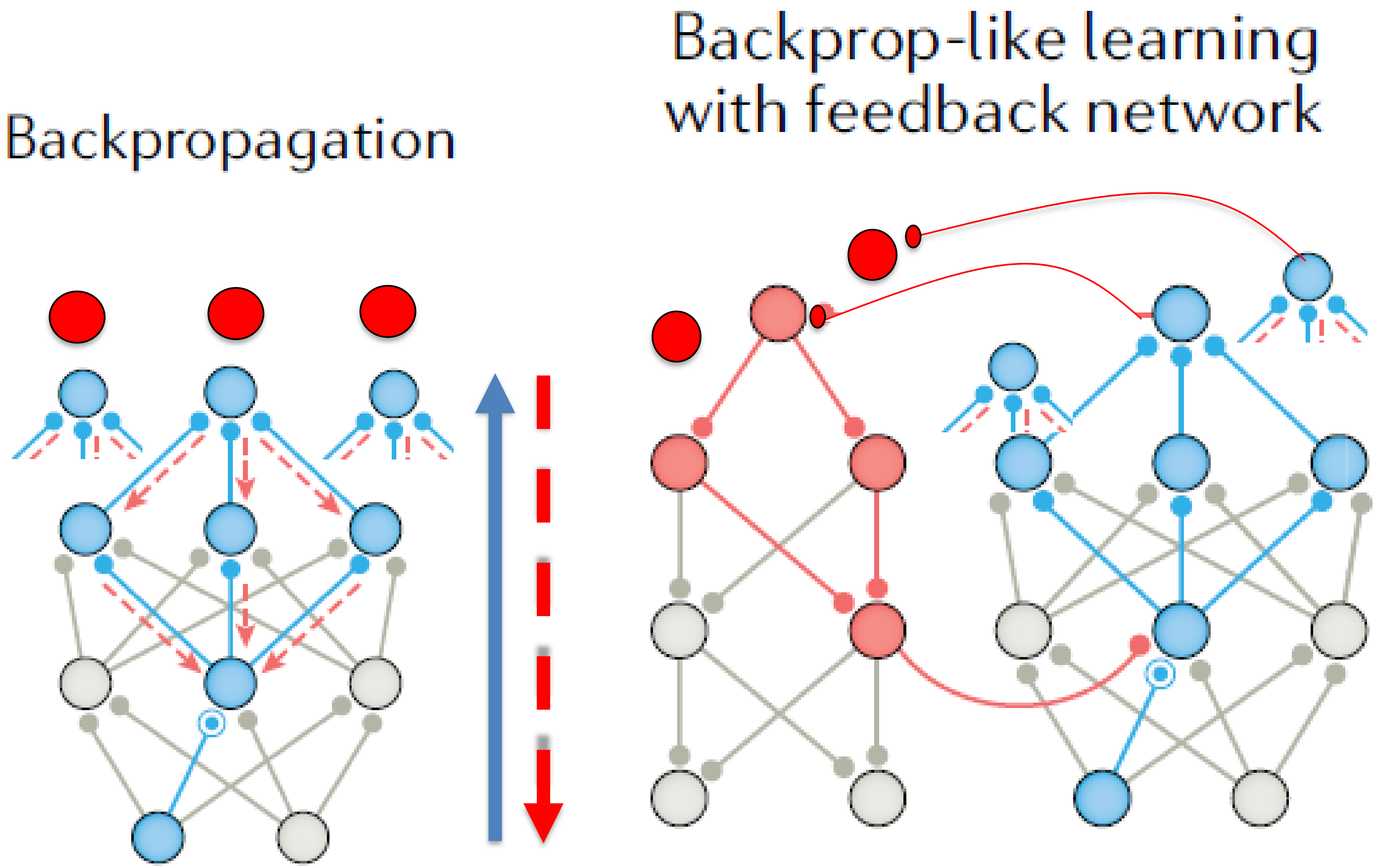
Previous slide.
Slide was already shown in week 1 and not shown again.

Backprop needs precise error feedback

Vector feedback:
- multiple outputs,
- one ‘signed error per output’
- error vector transmitted back
- precise neuron-specific errors

BackProp Algo has 4 phases:

1) Forward pass and freeze
2) Calculate local output errors
3) Backprop pass, using 2)
4) Update connections, using 1) +3)



BackProp rules: vector feedback

Adapted from Lillicrap et al. 2020 Nat. Rev. Neurosci.

Previous slide.
Slide was already shown in week 1 and not shown again.
.

How does BackProp work? Minimize errors!

- BackProp needs four separate phases:

  forward pass, output mismatch, backward pass, weight update.

- Backward pass needs specific feedback architecture

  (e.g., feedback weights = feedforward weights;
  backward multipliers depend on state
  of feedforward network).

→ Not implementable in biology!

F. Crick, The recent excitement about Neural Networks, Nature 337:129-132 (1989)
T.P. Lillicrap et al., Backpropagation and the brain, Nature Reviews Neurosci. 21: 335-346 (2020)

Previous slide.
Slide was already shown in week 1 and not shown again.

How does BackProp work? Minimize errors!

- BackProp needs four separate phases:

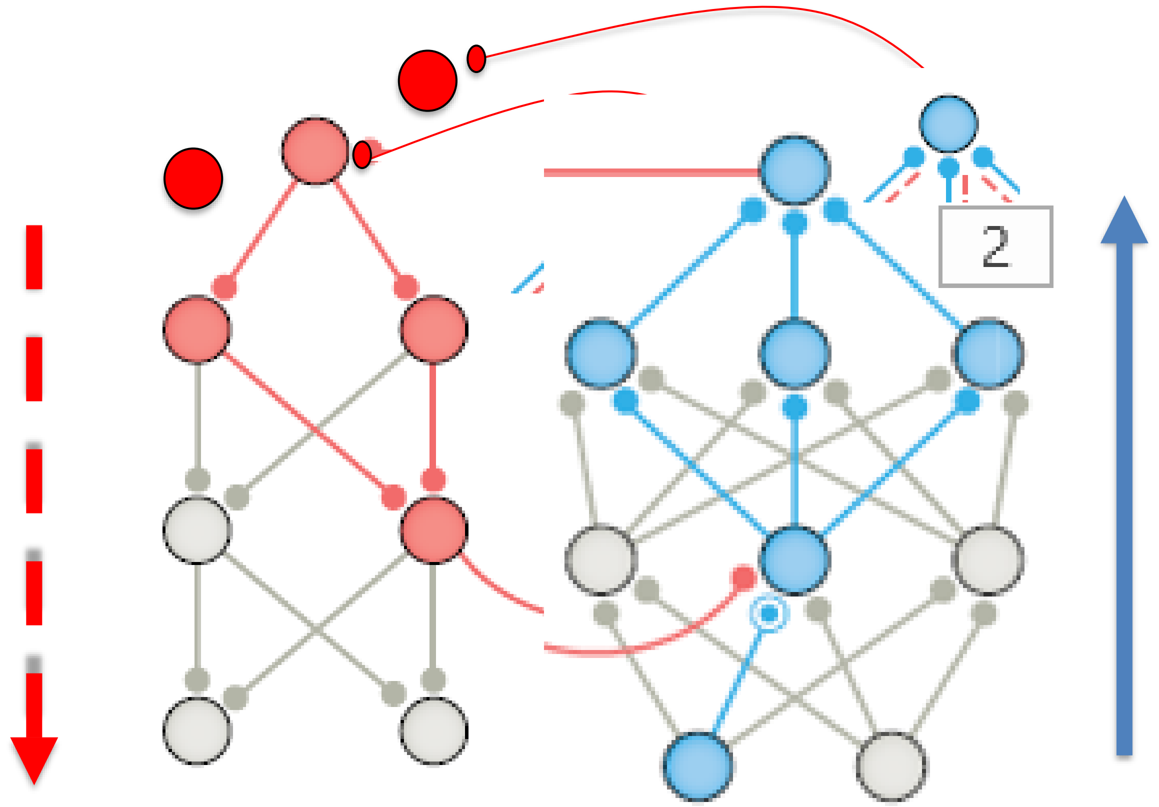
  forward pass, output mismatch, backward pass, weight update.

- Backward pass needs specific feedback architecture

  (e.g., feedback weights = feedforward weights;
  backward multipliers depend on state
  of feedforward network).

→ Not implementable in biology!

Backprop-like learning
with feedback network



F. Crick, The recent excitement about Neural Networks, Nature 337:129-132 (1989)
T.P. Lillicrap et al., Backpropagation and the brain, Nature Reviews Neurosci. 21: 335-346 (2020)

Previous slide.
Slide was already shown in week 1 and not shown again.

Summary of Introduction

1. Brain has learnt to predict missing parts
2. Analogy in machine learning is ‘self-supervised learning’
3. Backprop has several problems as model for neuroscience

- 4 phases for update	→ online, continuous time
- precise feedback architecture	→ robust, plausible feedback
- forward=backward weights	→ learning rules for all weights

BackProp is not implementable in biology

→ No BackProp, please!!!

Can we use a biologically plausible learning rule instead?

What are good candidates of learning rules?

Reading:

F. Crick, The recent excitement about Neural Networks, Nature 337:129-132 (1989)
T.P. Lillicrap et al., Backpropagation and the brain, Nature Reviews Neurosci. 21: 335-346 (2020)

Previous slide.

Similar to week 1. But this week we focus on selfsupervised learning without BackProp.

No Backprop, please!
Learning of ‘deep’ representations

Wulfram Gerstner

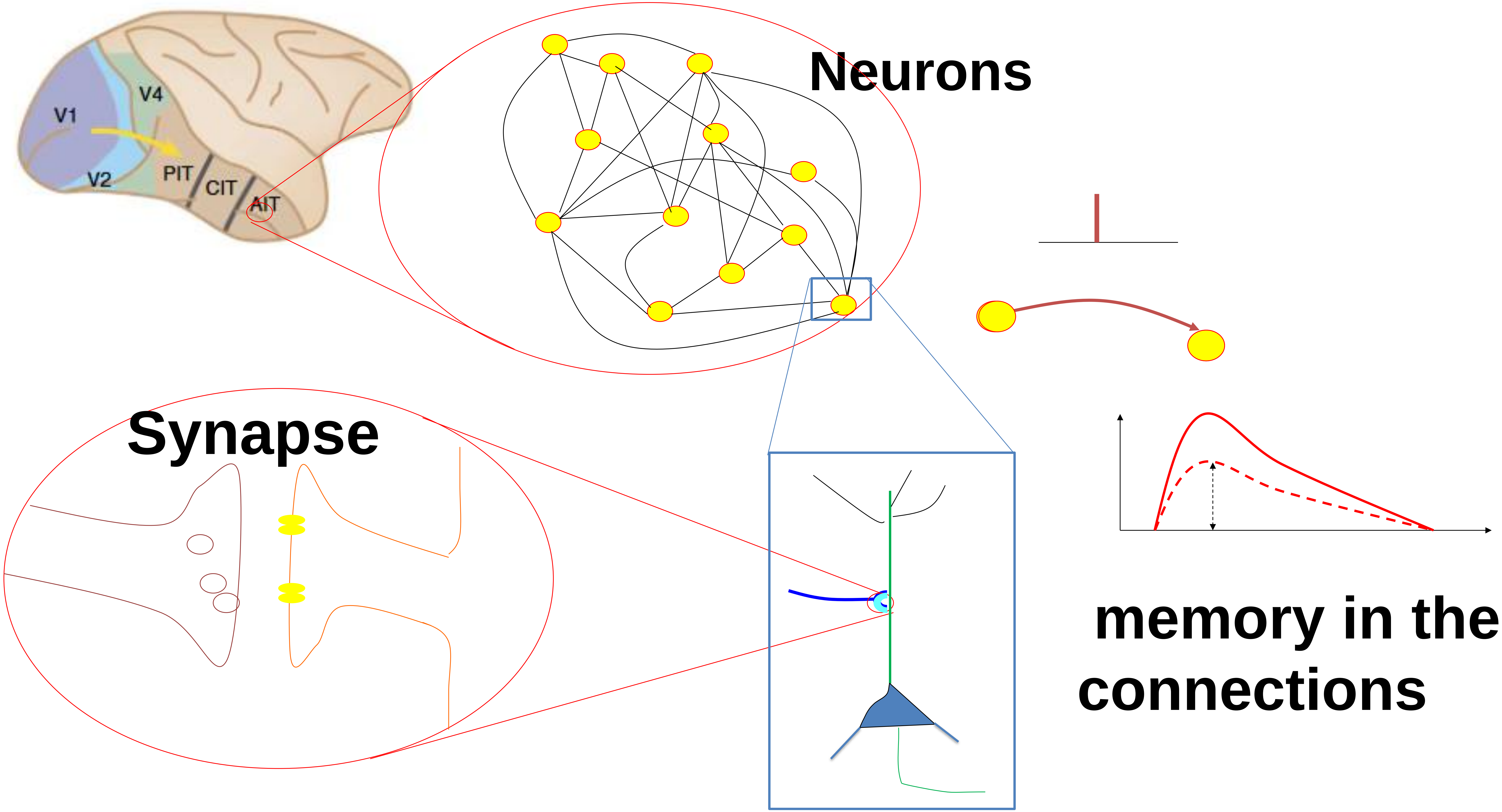
EPFL, Lausanne, Switzerland

1) Introduction (review)
2) **Plasticity and local learning rules (review)**
3) Contrastive Selfsupervised Learning
4) Representation Learning with CLAPP:
   “Contrastive, Local And Predictive Plasticity”
5) Feedback Alignment

Previous slide.

Similar to week 1 and last week.

Learning in the brain: changes between connections



Neurons

Synapse

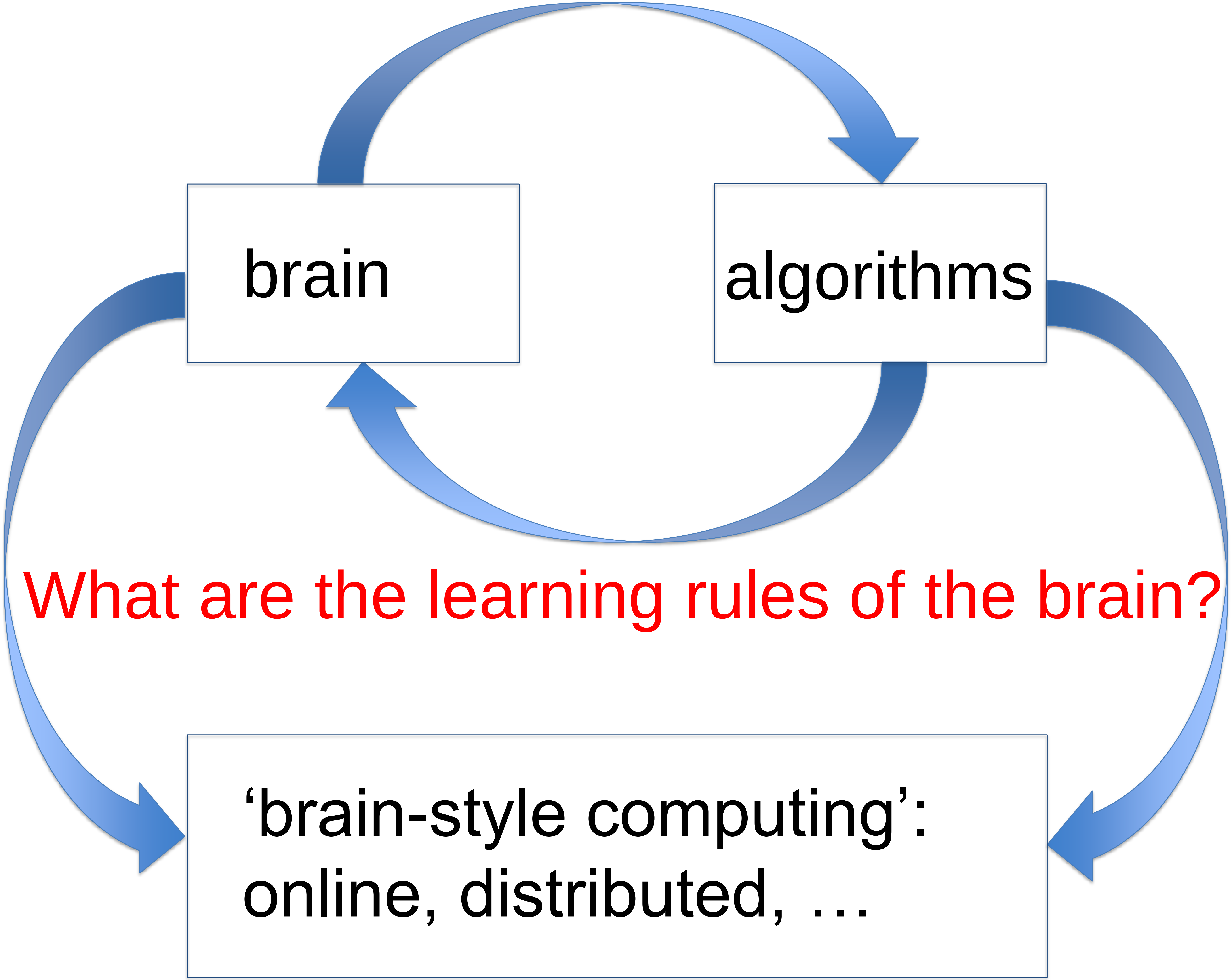
memory in the connections

learning = change of connection

Previous slide.

Similar to week 1 and last week.

Learning Rules: what makes connections change?



brain

algorithms

What are the learning rules of the brain?

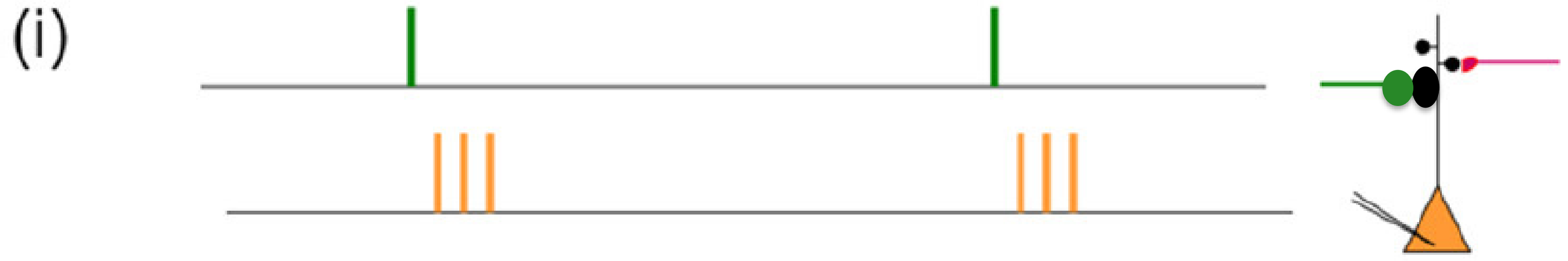
‘brain-style computing’:
online, distributed, …

Previous slide.

Similar to week 1 and last week.

Hebbian Learning (LTP)

Hebbian co-activation:



> "if two neurons are active together, the connection between those two neurons gets stronger."

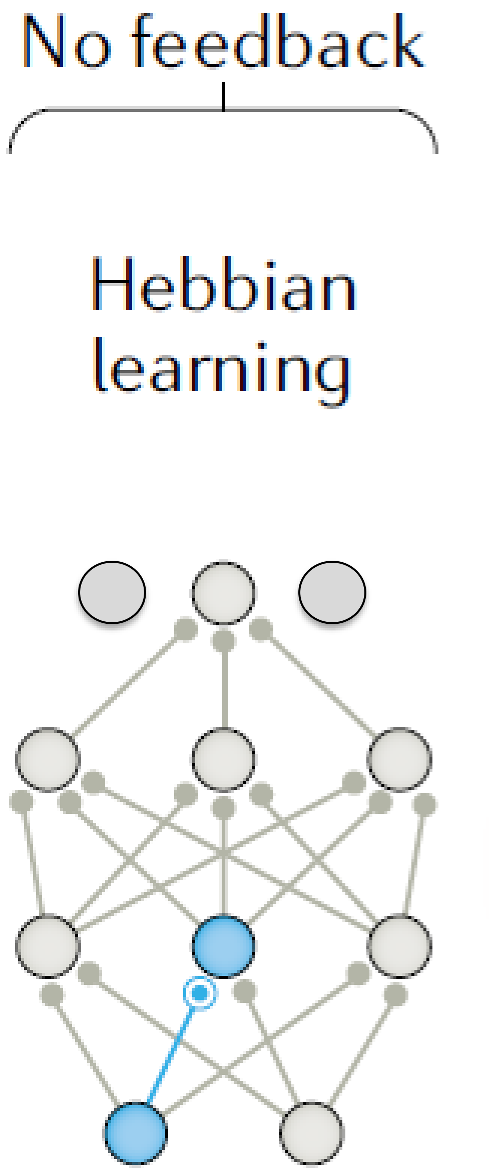
Hebb postulate (1949)

Many classic experiments: 1970-2005 *Bliss and Lomo, J. Physiol., 1973; ..*

Previous slide.

Similar to week 1 and last week.

2-factor rules use information locally available at the synapse



2-factor rules:
unsupervised,
no feedback

Big question:
Can we learn anything
at all without feedback?

Standard Answer:
Development of Receptive fields,
… but not much more!

image: Lillicrap et al. 2020

Previous slide.

In the first part of the class when we discussed 2-factor rules we saw already examples of what can be achieved with this context

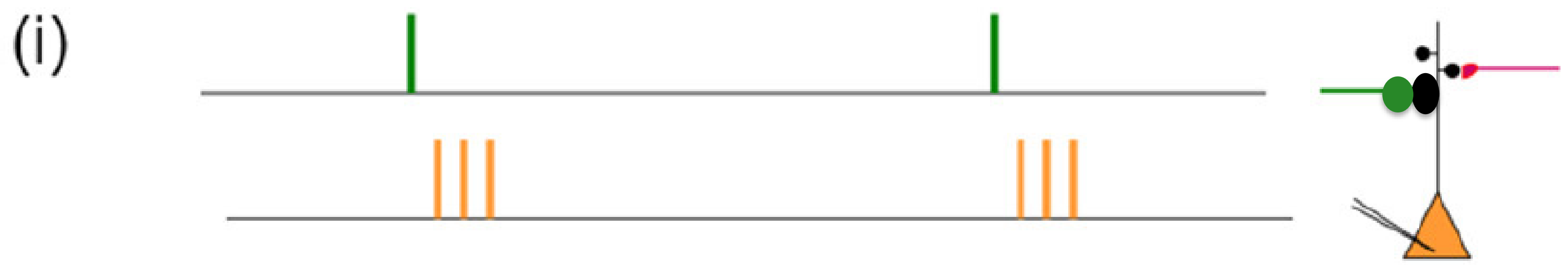
- PCA or ICA
- Receptive Field Development
- Winner-Take-All or k-means-clustering
- Soft-winner-take all/competitive learning.

Note that all of these methods have been shown in a SINGLE layer of neurons.

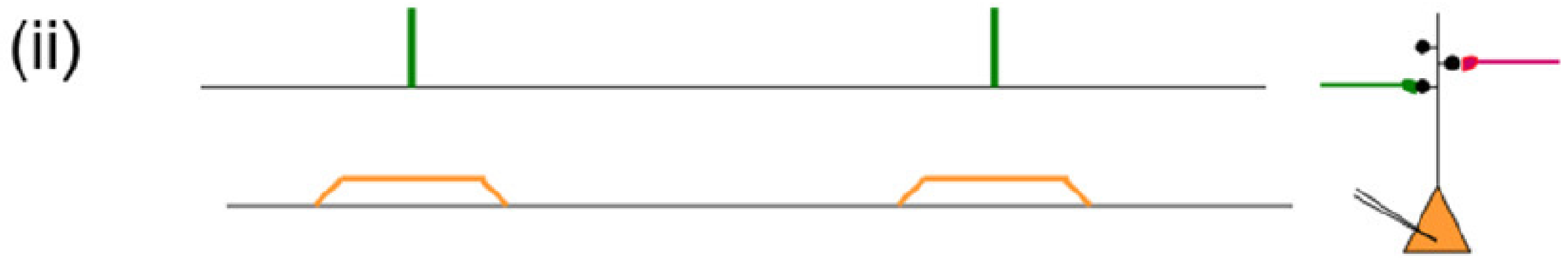
Many people have tried over decades to build multi-layer versions of these methods, but failed! We will see that we need just a bit more!

Voltage-dependence of Hebbian Learning (LTP and LTD)

Hebbian co-activation:



Hebbian coactivation
without postsyn.-spikes



Classic voltage dependent experiments: 1990-2005
Clopath model of voltage-dependent plasticity: 2010
→ synaptic changes depend on voltage and spikes

A.Artola, S.Bröcher and W. Singer (1990). Nature 347, pp. 69–72. (1990)
A.Ngezahayo, M.Schachner and A.Artola (2000). J. Neuroscience 20, pp. 2451–2458. (2000)
P.J. Sjöström, G.G. Turrigiano and S.B. Nelson (2001) Neuron 32, pp. 1149–1164.
C.Clopath, L. Busing, E. Vasilaki and W. Gerstner (2010) Nature Neuroscience 13, pp. 344–352 (2010)

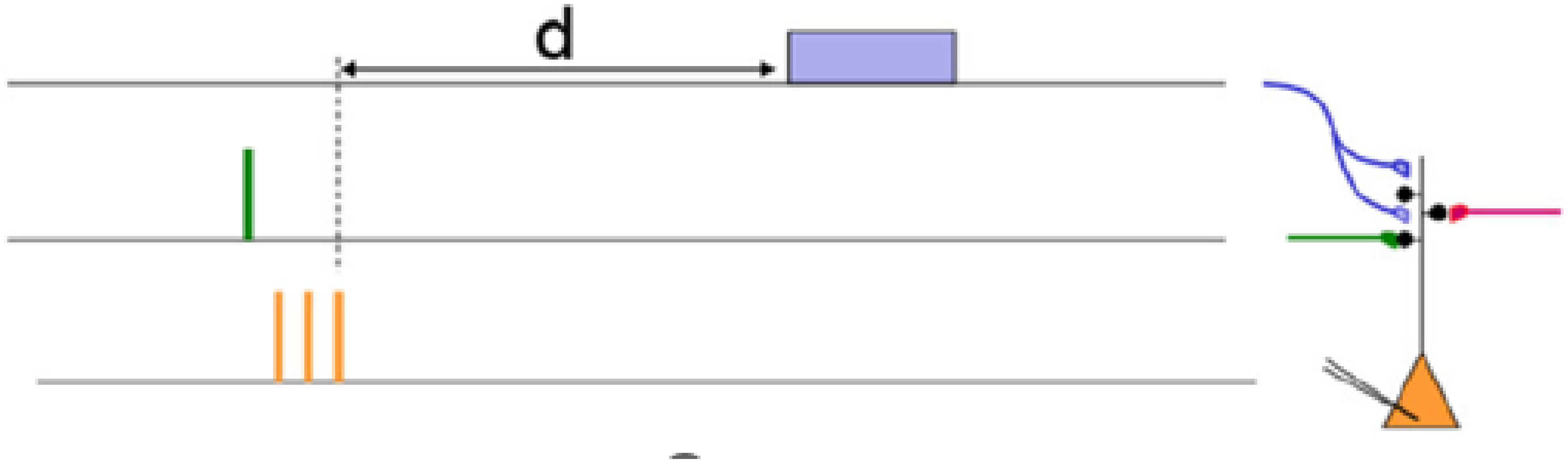
Previous slide.

Discussed last week.

Hebbian rules are not strong enough: need 3-factor rules!

Three-factor rules are Hebb + neuromodulator

Dopamine/Serotonin/Ach → reward/surprise/alert



important for action learning: to ski/to ride bicycle

→ Dopamine (even if delayed by 1s) helps learning

Experiments: 2014-2025

Dopamine: 1997-2025

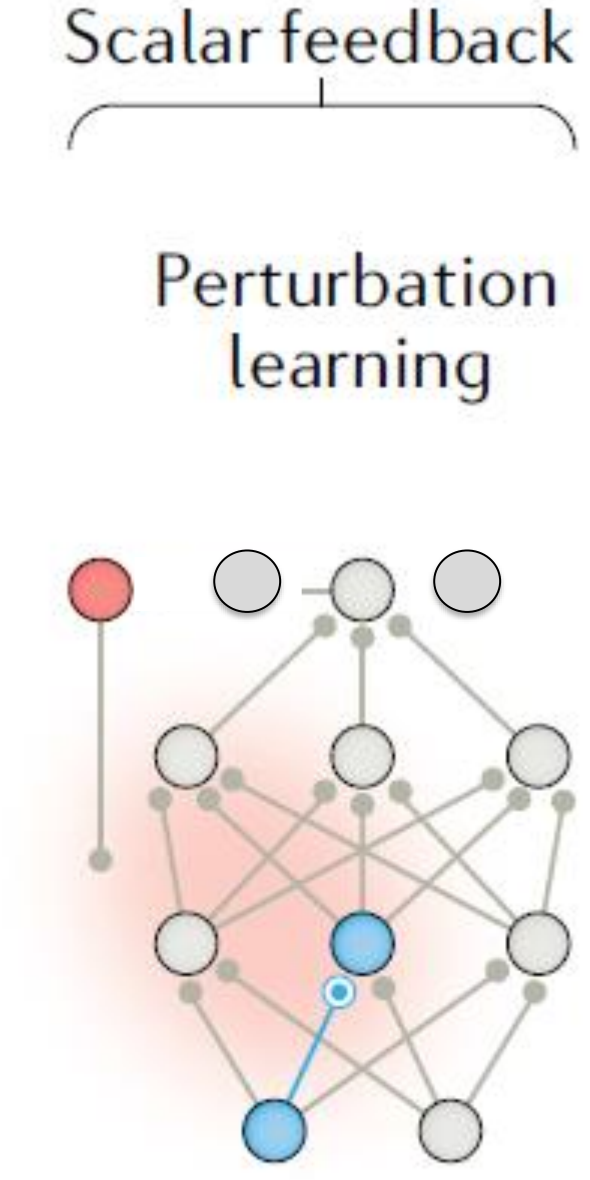
Yagishita et al. Science, 2014; …

W. Schultz, P. Dayan and R.R. Montague, 1997; …

Previous slide.

Discussed last week and in the context of 3-factor rules.

3-factor rules use information locally available at the synapse combined with one global feedback signal



3-factor rules:
reward-based,
feedback

Big question:
What can be learned with these rules?

Answer:
→ action learning (ski, bicyle, tennis)
→ rapid decision making (chess, go, buy/not buy)

BUT: 3-factor rules only work if you have a 'good representation'

image: Lillicrap et al. 2020

Previous slide.

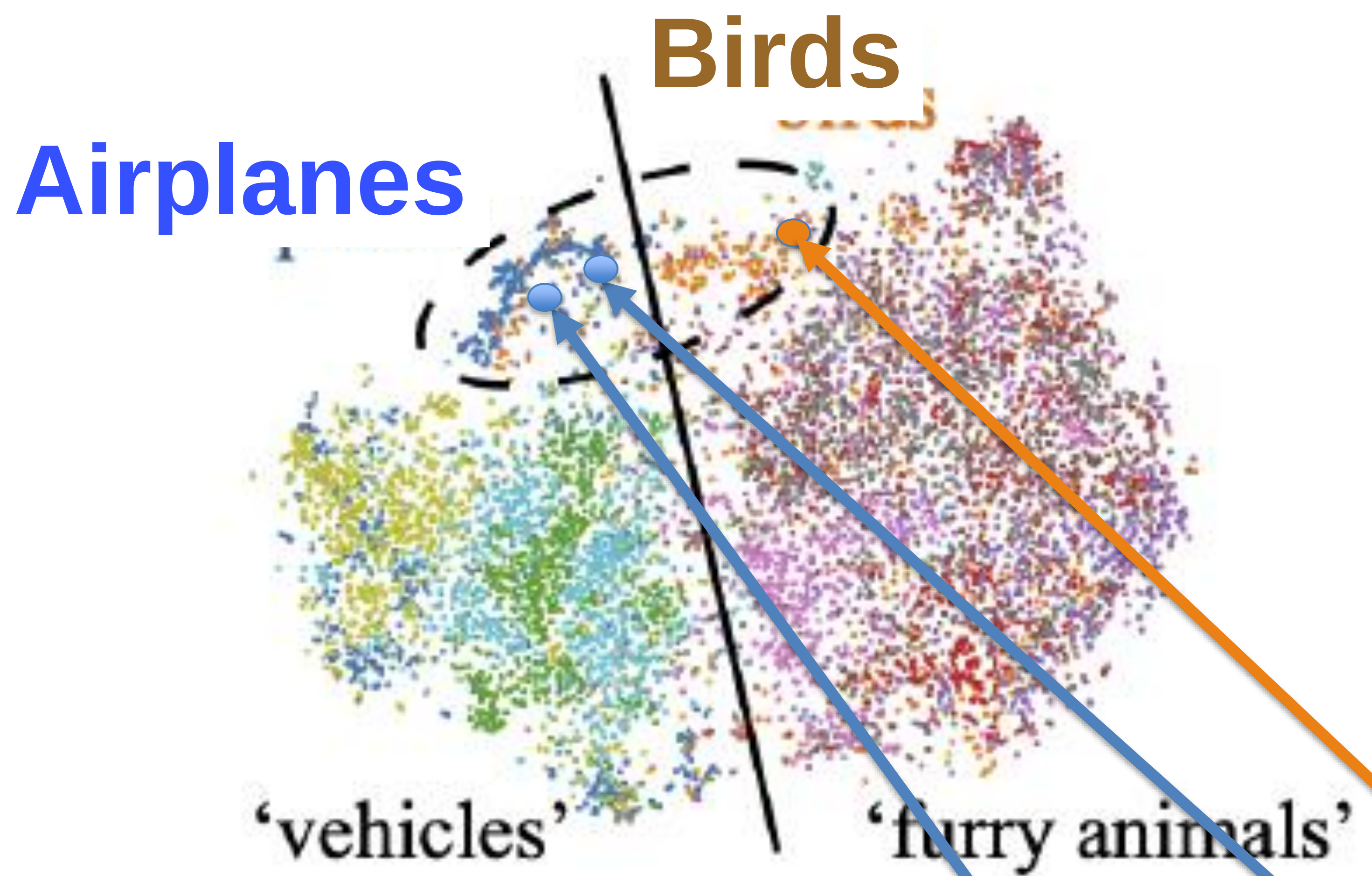
We have seen that nearly all algorithms of reinforcement learning can be implemented as 3-factor rules. Examples are SARSA, Q-learning, REINFORCE with baseline, Actor-Critic models.

However, all these 3-factor rules require a single layer of weights to be learned, i.e., the layer from state representation to action output.

Deep reinforcement learning is not compatible with 3-factor rules.

Therefore, we need a ‘good state representation’.

Good representation → needs deep network

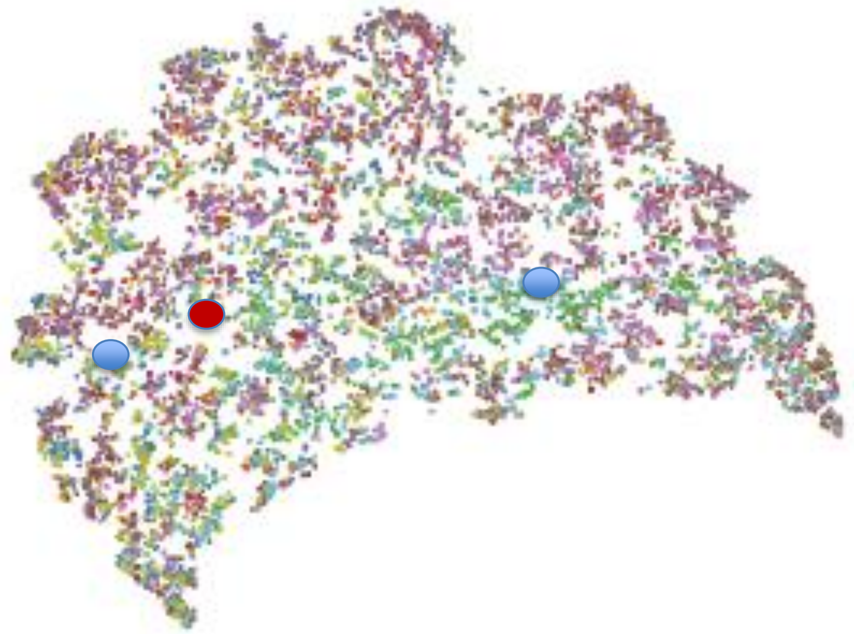


→ Objects of same class are neighbors

→ Raise arm if airplane →

→ could be learned with 3-factor rule

bad representation



Previous slide.
What would be a good state representation?

Suppose we have many unlabeled images, some of these with airplanes others with birds. At the end of the visual processing stream (say in IT), we want a representation such that two images of airplanes are represented similarly, but different from two images of birds.

This idea requires a deep network since the pixel images of an airplane from below and a bird from below may be more similar, that the image of a black airplane from below and a white airplane from above.

All single-layer methods such as PCA, ICA, or clustering would therefore not work!

The similarity in pixel space is not always a good predictor for similarity in the space of ‘meaning’ that is developed in deep areas such as IT.

However, if we have a good representation in a deep area, then it will be easy to learn a reward-based task such as raise your arm if you see an airplane.

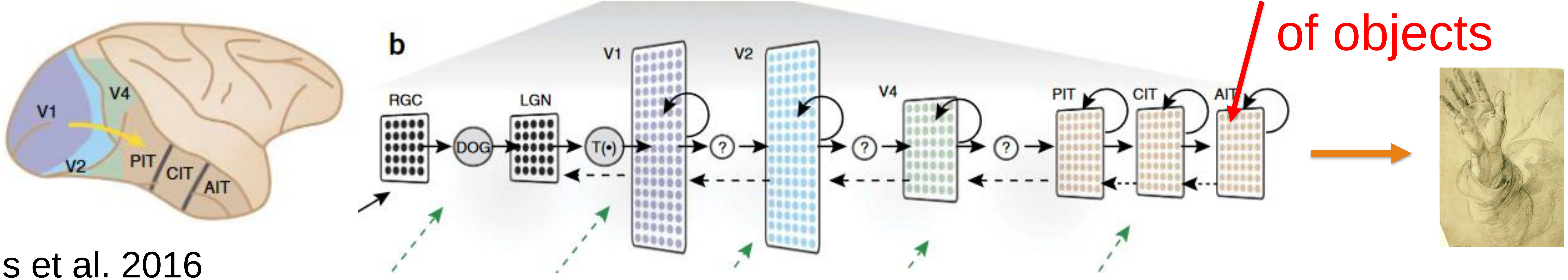
Learn a ‘good representation’!

Learning rules???

Network architecture???

What kind of feedback???

Big question:
Can we have local learning rules
(with several global signals)
that yield **good representations**
in multi-layer networks?



representation of objects

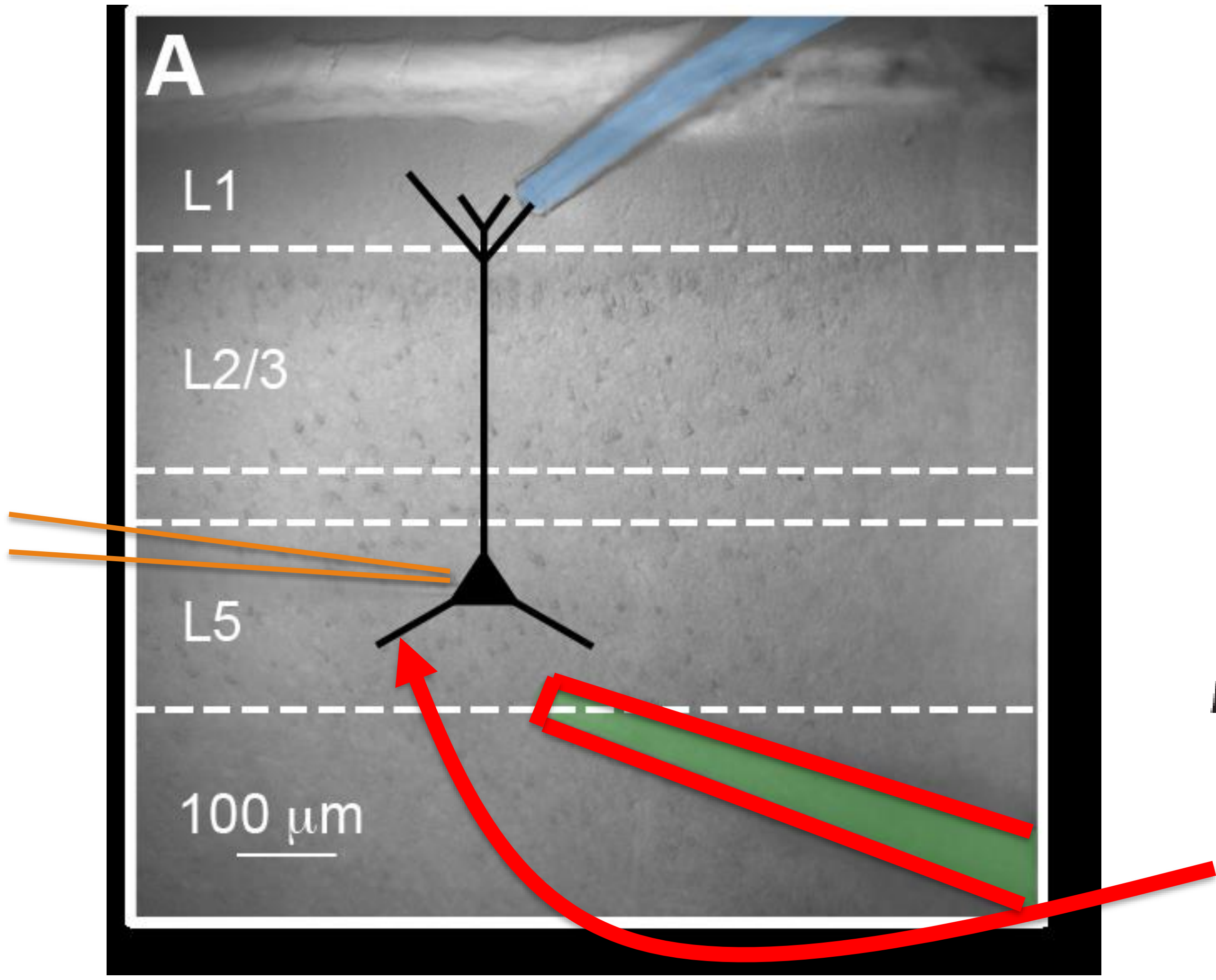
image: Yamins et al. 2016

Previous slide.

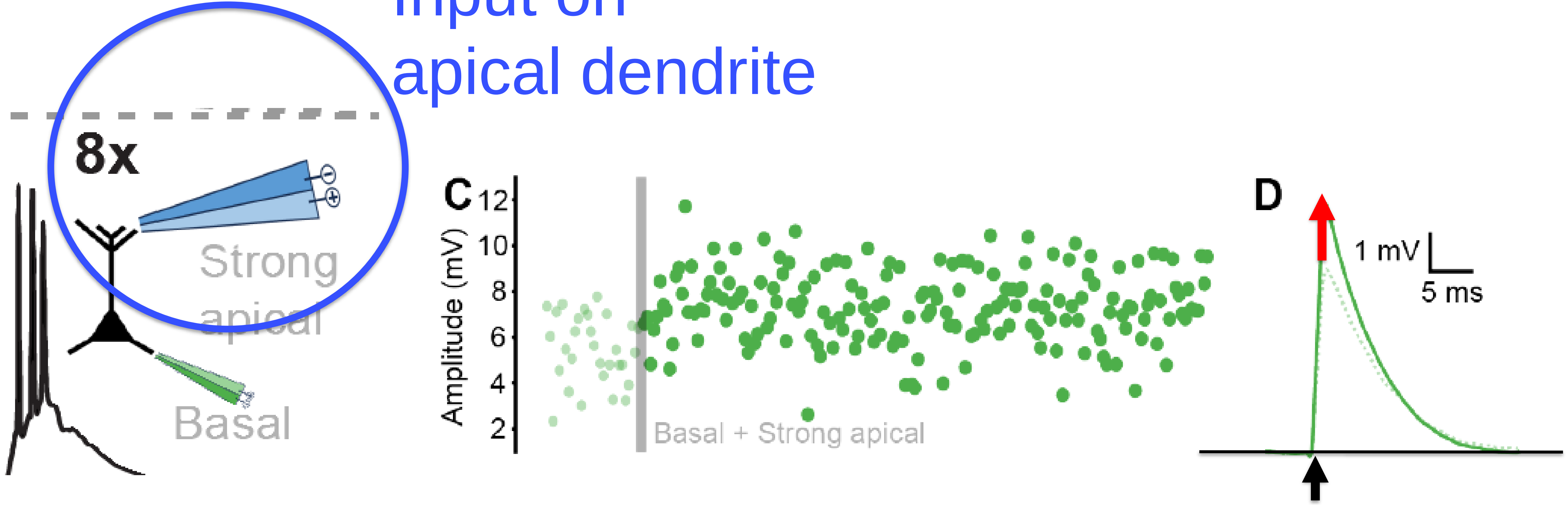
So, how can we learn a good representation in multi-layer networks?

We exploit (next slide) a phenomenon of spike-based and voltage-based learning that we already discussed last week.

2024: **Learning rule**: Feedforward synapse on basal dendrite depend on lateral/feedback input (on apical dendrite)



Input on
apical dendrite



synapse on basal dendrite → LTP

Aceituno, …, Grewe, bioRxiv (2024)
https://doi.org/10.1101/2024.04.10.588837;

Recent experiments in L5: Grewe group (2024)
Experiments in L2/3: Williams and Holtmaat (2019)

Consistent with voltage-dependent plasticity (*Sjostrom et al 2001*)
and Clopath model (*Clopath et al. 2010)*

Such a rule useful to learn ‘good’ representation!

Experiments in Mouse Frontal Cortex, L5 cells, slice, from the Grewe lab.
Two electrodes are used for extracellular stimulation at the basal dendrite (red-green) and apical dendrite (blue). Voltage is recorded with the brown electrode (A).
Initially, EPSPs are evoked by small-amplitude pulse stimuli (strength s1) with the red-green electrode yielding an EPSP of a few mV. Then the stimulation amplitude is increased (strength s2) so that the firing threshold is reached, and the postsynaptic neuron fires an isolated spike. After 8 repetitions (at 0.1Hz) no change in the EPSP amplitude is found. Thereafter the stimulation of basal synapses (with strength s2) is paired with stimulation of the apical dendrite, causing a short burst of spikes and a prolonged voltage response. After 8 repetitions (at 0.1Hz) the EPSP amplitude in response to stimulus s1 is increased (C and D).

These findings are consistent with experiments of J. Sjostrom (2001) and the voltage-dependent plasticity model of C. Clopath (2010): synaptic changes require either multiple postsynaptic spikes are a prolonged depolarization of the postsynaptic neuron, or a combination of both.

No Backprop, please!
Learning of ‘deep’ representations

Wulfram Gerstner

EPFL, Lausanne, Switzerland

1) Introduction (review)
2) Plasticity and local learning rules (review)
3) **Contrastive Selfsupervised Learning**
4) Representation Learning with CLAPP:
   “Contrastive, Local And Predictive Plasticity”
5) Feedback Alignment

Previous slide.
This section gives the general background of contrastive self-supervised learning.

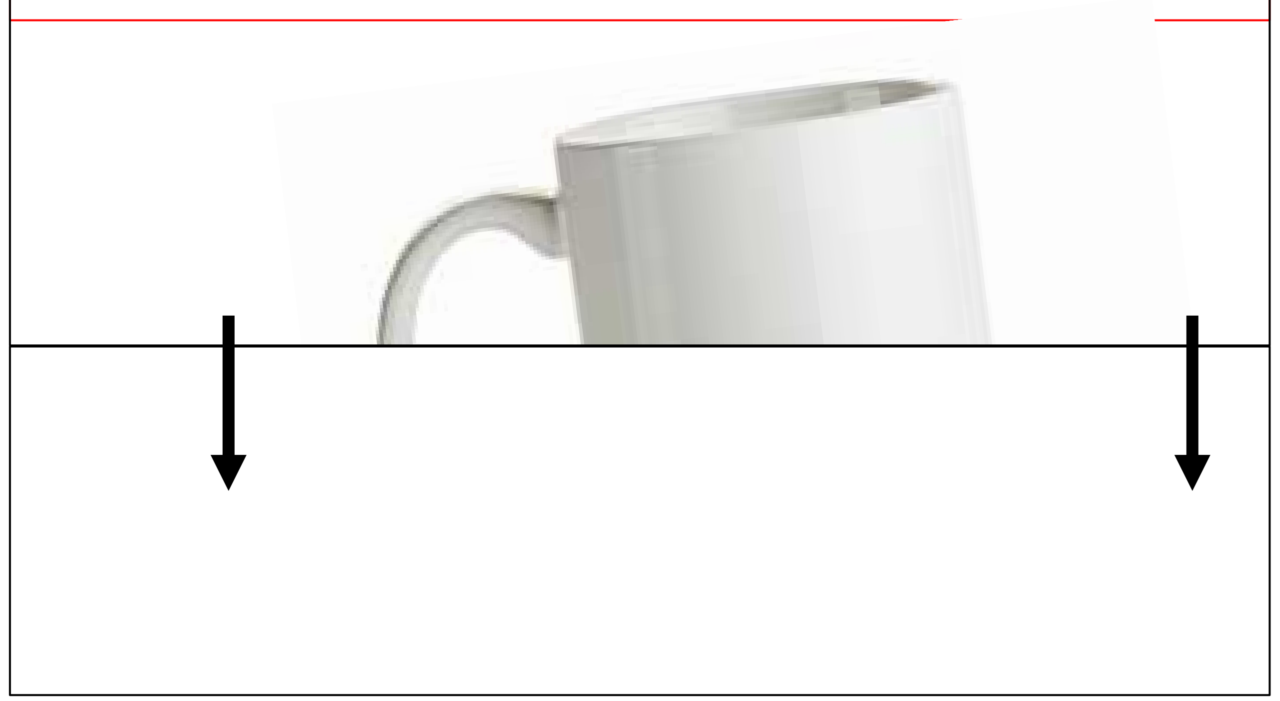
Self-supervision and prediction

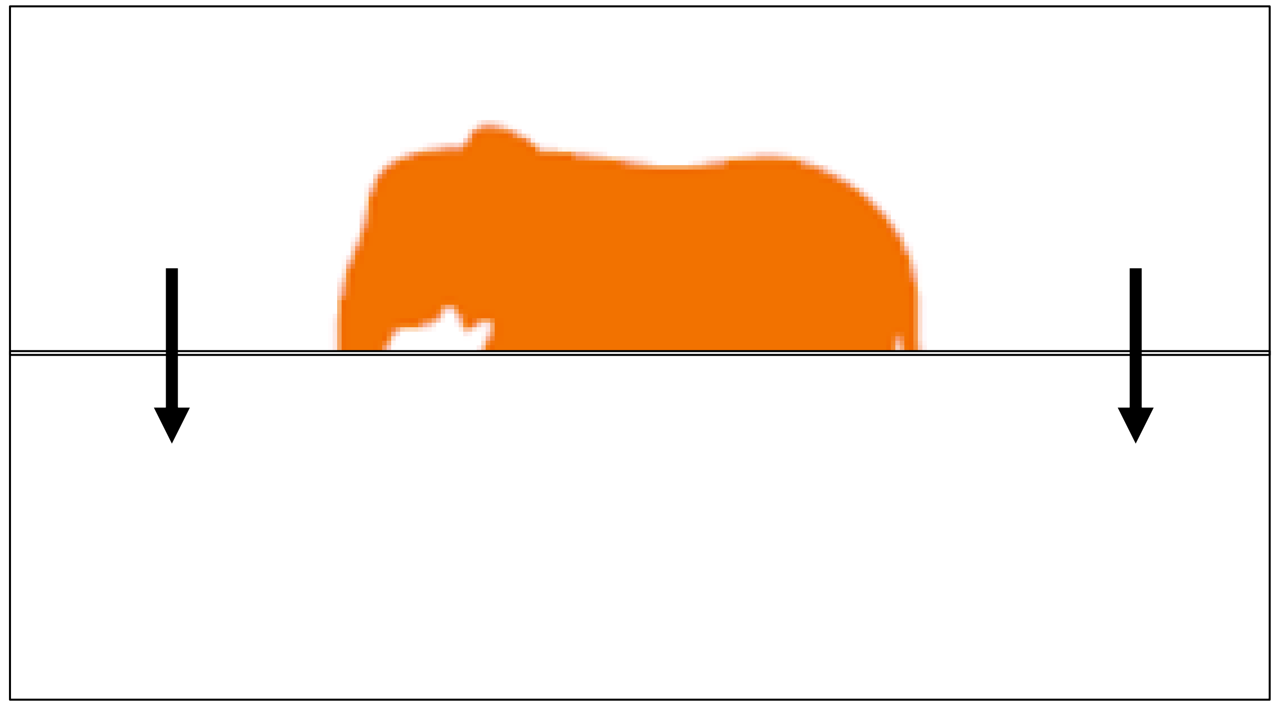
Predict across saccade far away!

Prediction impossible if not the same object!

Views of same object!

‘sameness’ info





Predict lower half!
Possible if same object!

Predict lower half!
Possible if same object!

Previous slide.

Self-supervised learning is based on predictions.
It should be possible to predict the lower half of an image from the upper half.

But it should not be possible to predict the presence of an elephant from the image of a mug.

In the first case, it is the SAME object. In the second case it is a DIFFERENT object.

Exploiting sameness versus difference is called contrastive learning.

Selfsupervised Learning: Contrastive Learning

Examples:

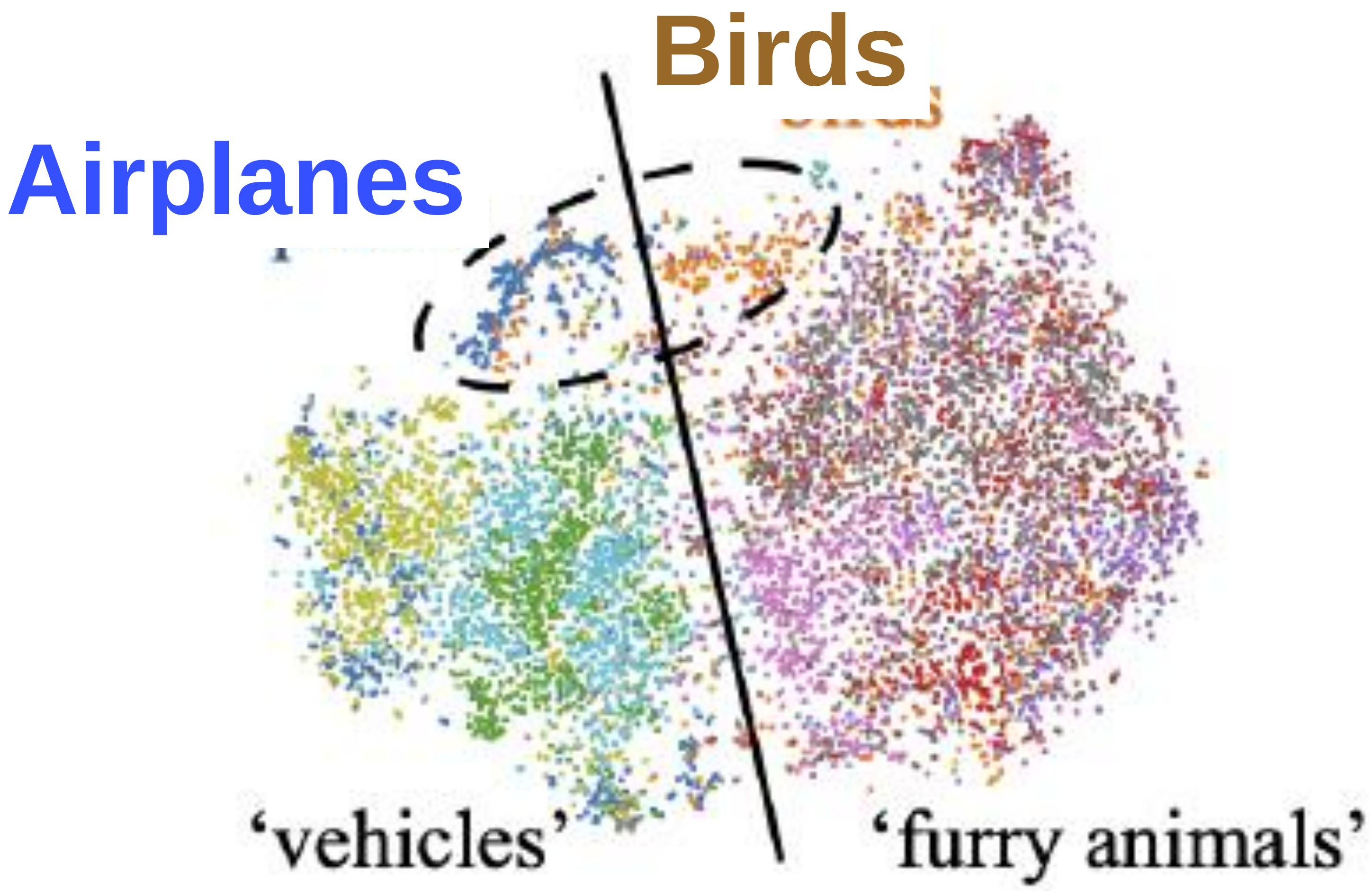
- Predict left part of image from right part
- Predict original image from augmented image

→ Align representation in representation layer

BUT:

- Avoid collapse of representation
- Use negative samples that the network should not predict

→ Move different ‘objects’ far from each other



Previous slide.

Note that in an autoencoder we would predict the pixels of the image. In self-supervised learning we only predict the representation of the image in a deep layer.

Prediction would be trivial if all objects lead to the same set of activated neurons. This situation is sometimes called a collapse of representation. That is not what we want.

We aim for a representation where different views of the same objects leads to very similar activation patterns of neurons whereas views of two different objects lead to very different activation patterns. The second aspects gives rise to the term ‘contrastive’.

Equivalently, instead of different views of the same object, we can say: if we look at one and the same image then the activity of representation neurons that respond to the LEFT part of should be predictable from the CONTEXT, i.e., the RIGHT part of the image.

CLAPP Loss = Hinge Loss

Illing, et al. NeurIPS 2021

$$L_{CLAPP}^{t,l} = \max(0, 1 - y^{t} \cdot u_{t}^{t+\delta t,l})$$

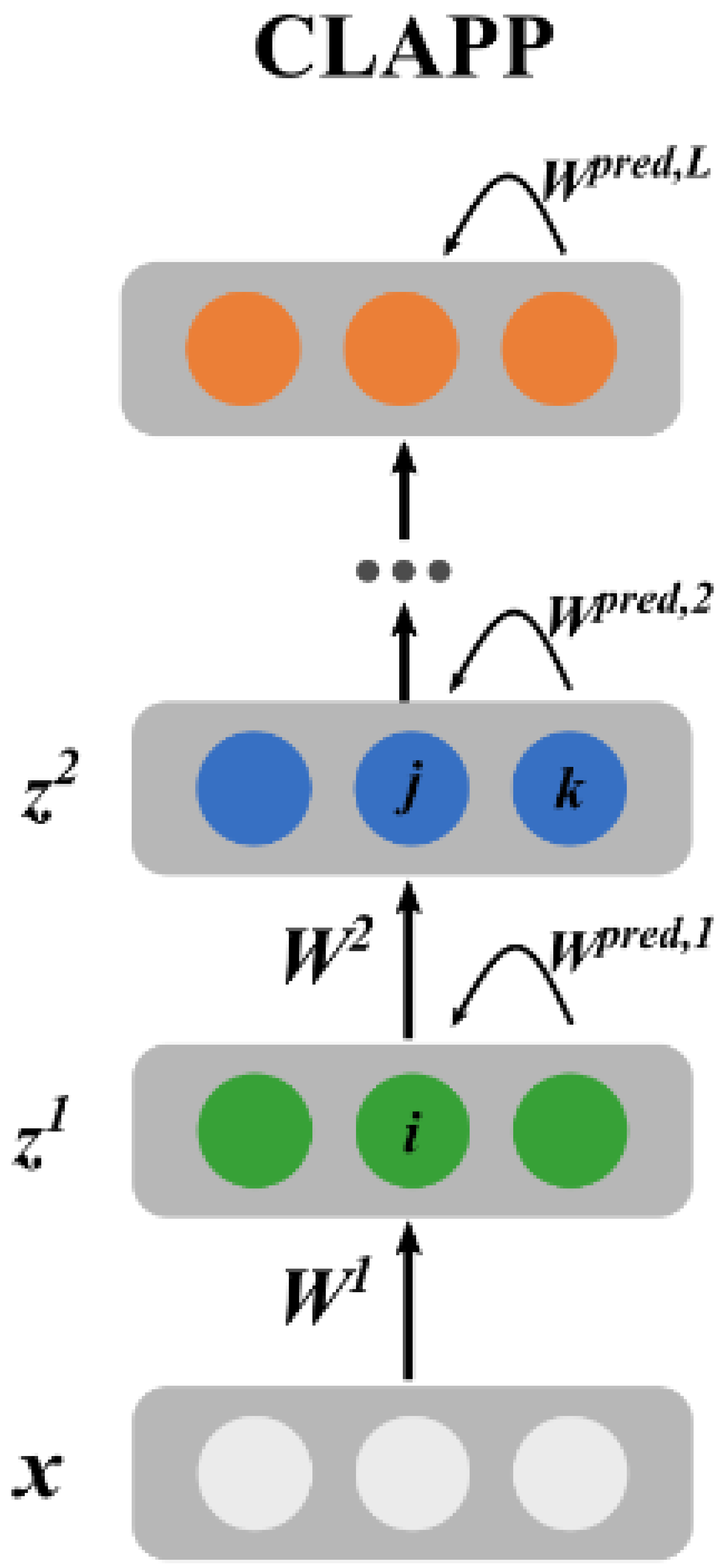
$u_{t}^{t+\delta t,l}$ = similarity:

$$u_{t}^{t+\delta t,l} = \boldsymbol{z}^{t+\delta t,l} \underbrace{\boldsymbol{W}^{pred,l} \boldsymbol{c}^{t,l}}$$

feedforward vs lateral prediction

y^{t} = sameness signal/contrastive signal

$$y^{t} = \begin{matrix} 1 \ \mathit{if\ same\ sample} \\ -1 \ \mathit{if\ next\ sample} \end{matrix}$$



Previous slide.
The small index l is the layer index. The last layer is l=L.

Let us look at the loss in the last layer. It is a hinge-loss (picture on the next slide): either zero or linear in u.

The variable u is a measure of the similarity between the activity state vector **z** in layer l and the lateral prediction from OTHER neurons **c = z** in the same layer.

If variable y tells whether the prediction comes from the SAME object (y=1) or a different object (y=-1).

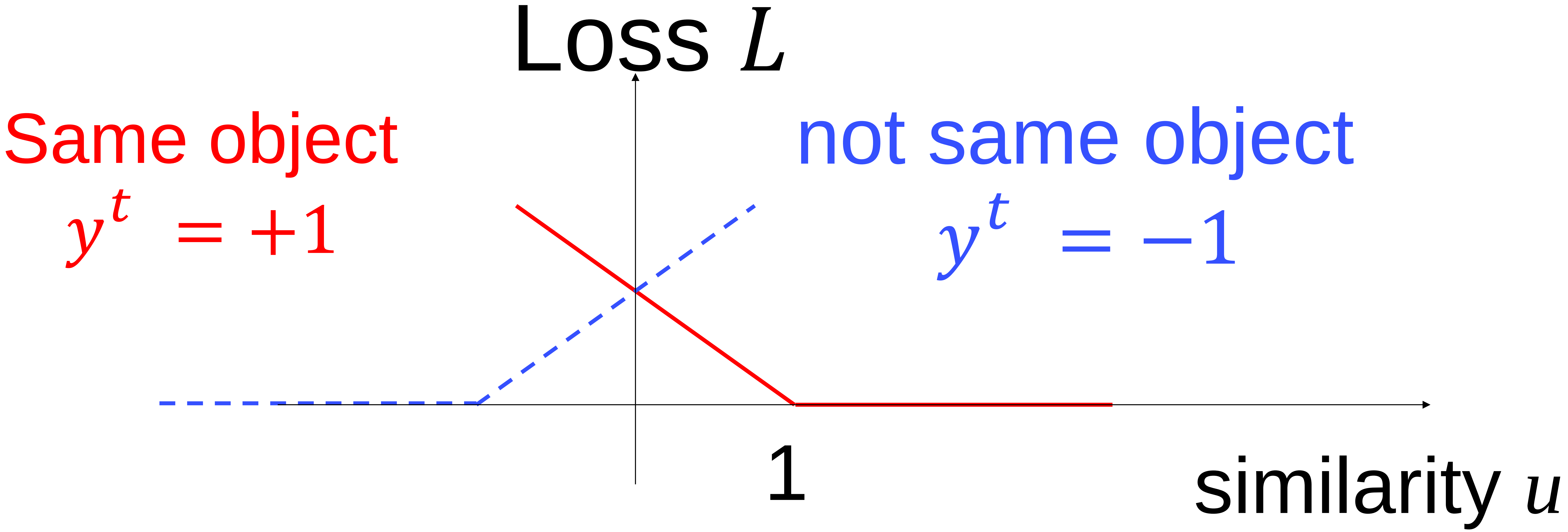
The boldface **z** refers to all neuron in a layer. For an interpretation it is easier to look at individual neurons such as neuron i in layer l. Its activity depends ONLY on the feedforward pathway

$$z_i^{t+\delta t,l} = g(\sum_j w_{ij}^l \, z_j^{t,l-1})$$

CLAPP Loss = Hinge Loss

Illing, et al. NeurIPS 2021

$$L_{CLAPP}^{t,l} = \max(0,1 - y^t \cdot u_t^{t+\delta t,l})$$



Previous slide.

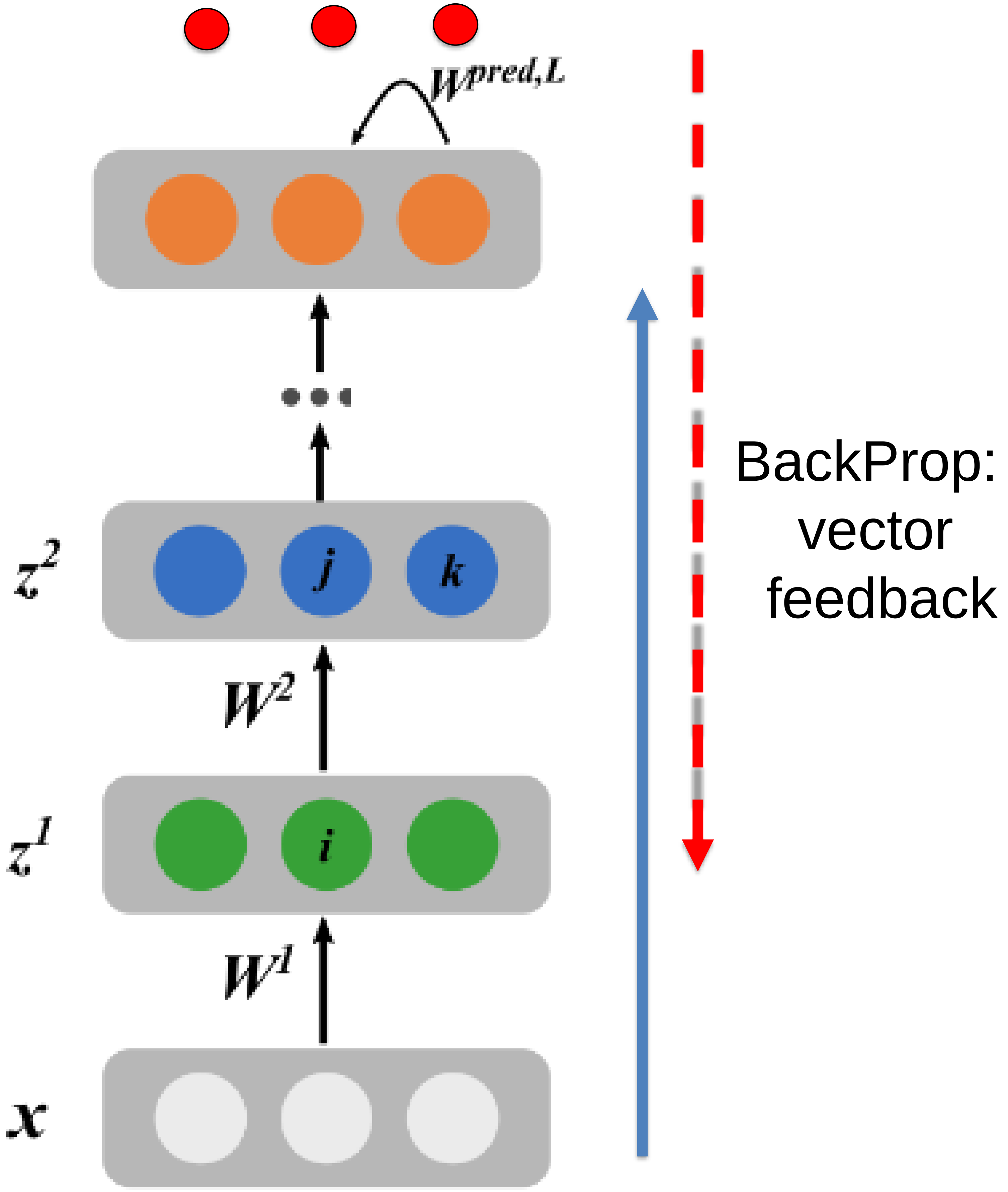
Hinge loss means in our case:
If the similarity u between the prediction based on the context **c** and the actual representation state **z** is large, and it is a valid context (i.e. same object) so that predictions should be possible (y=1), then the loss is zero.
The notion of ‘large’ is defined by a margin of unity.

Similarly, if the similarity is below zero and the context has changes (i.e. different object), then the loss is also zero.

CLAPP Loss = Hinge Loss

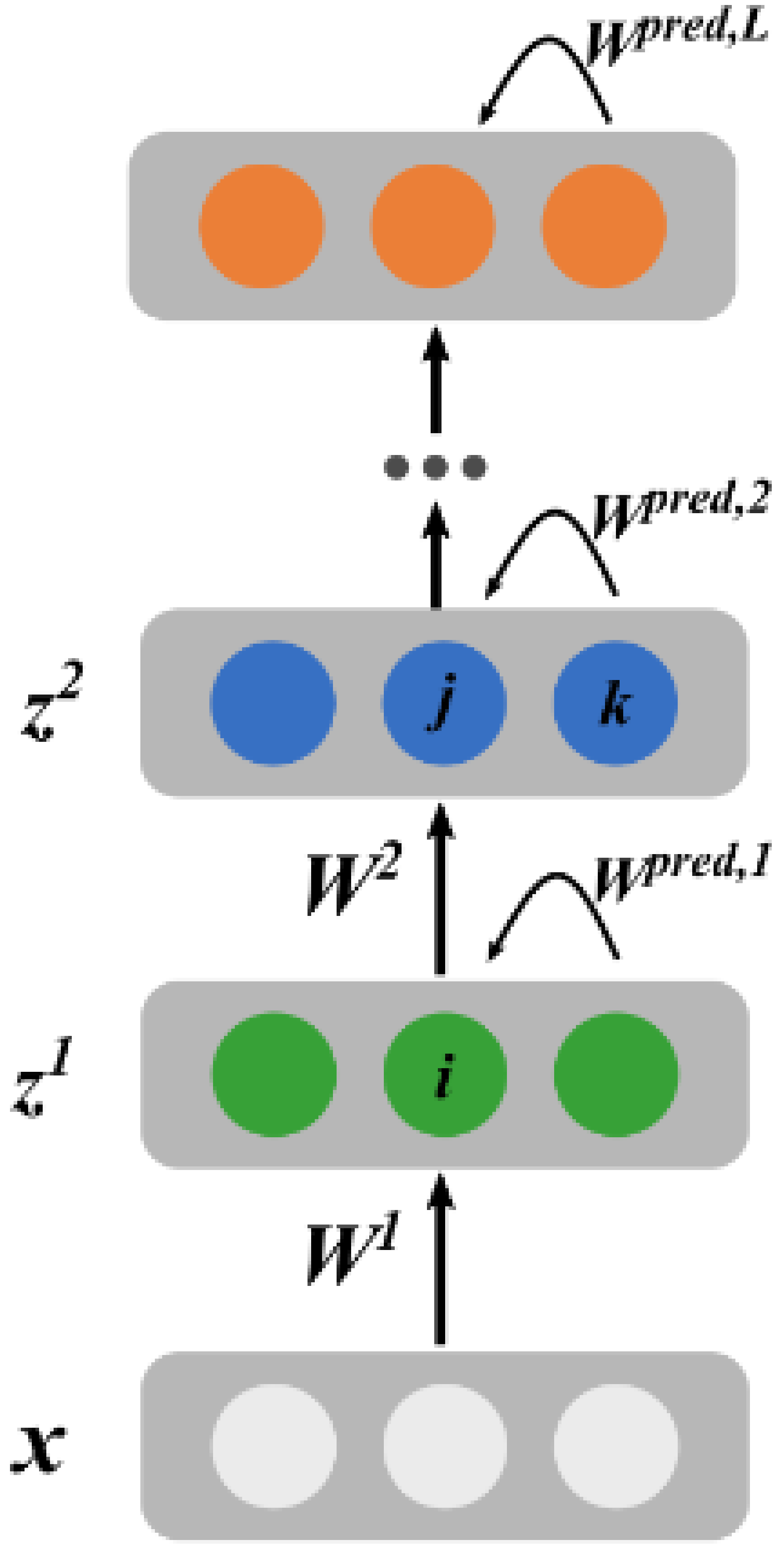
$$L_{CLAPP}^{t,l} = \max(0, 1 - y^t \cdot u_t^{t+\delta t,l})$$

Hinge Loss with Backprop: self-supervised learning



BackProp: vector feedback

Hinge Loss layerwise: CLAPP



Previous slide.

LEFT: The Hinge Loss is used only at the output layer l=L. If the loss is non-zero then weights in the whole network are adjusted using BackProp.

RIGHT: The Hinge Loss is applied separately in each layer. The resulting algorithm is called CLAPP

CLAPP Loss = Hinge Loss

Illing, et al. NeurIPS 2021

$$L_{CLAPP}^{t,l} = \max(0,1 - y^t \cdot u_t^{t+\delta t,l}) \qquad (1)$$

$$u_t^{t+\delta t,l} = \boldsymbol{z}^{t+\delta t,l}\,\boldsymbol{W}^{pred,l}\boldsymbol{c}^{t,l} \qquad (2)$$

If Loss=0, then $deriv^{t,l}$=0; else $deriv^{t,l}$=1

$\Delta w_{jk}^{pred} \propto (\mathrm{deriv}^{t,l})(y^t)\; z_j^t\; c_k^{t-\delta t}$ $\qquad$ $\Delta W_{jk}^{\text{pred}} = \gamma_t \cdot z_j^\tau \cdot c_k^{t-\delta t}$

$$\Delta w_{ij} \propto (deriv^{t,l})(y^t)(W^{pred}\, c^{t-\delta t})_i \cdot \rho'(a_i^t) \cdot x_j^t$$

$$\Delta w_{ij} \propto (NeurMod1)(NeurMod)\;\boldsymbol{W}^{\text{pred}}\boldsymbol{c}^{t_1})_{\text{i}} \cdot \text{post}_{\text{i}}^{t_2} \cdot \text{pre}_{\text{j}}^{t_2} \;.$$

2 broadcast factors $\qquad$ Hebbian

1. Sameness-label (y^t=+1=same; y^t= -1 = saccade)
2. Prediction was good (zero-loss) or not $deriv^{t,l}$

Previous slide. The calculation was done on the blackboard.

Importantly: the resulting learning rule is biologically interpretable with the following terms

For a feedforward synapse (typically located on the basal dendrite):

- Presynaptic activity
- State g'(u) of the postsynaptic neuron
- The lateral predictive input into neuron i: $lat_i^l = \sum_k w_i^{pred,l} \, z_k^{l-\delta t}$ which would arrive in the apical dendrite
- A broadcast factor that indicates 'same object' or not. The signal 'not same object' could be caused by a saccade.
- If it is the 'same object' learning only happens if the prediction is not yet good enough.

No Backprop, please! Learning of ‘deep’ representations

Wulfram Gerstner

EPFL, Lausanne, Switzerland

1) Introduction (review)
2) Plasticity and local learning rules (review)
3) Contrastive Selfsupervised Learning
4) **Representation Learning with CLAPP: “Contrastive, Local And Predictive Plasticity”**
5) Feedback Alignment

Previous slide.

Now we explore an implementation of the CLAPP rule

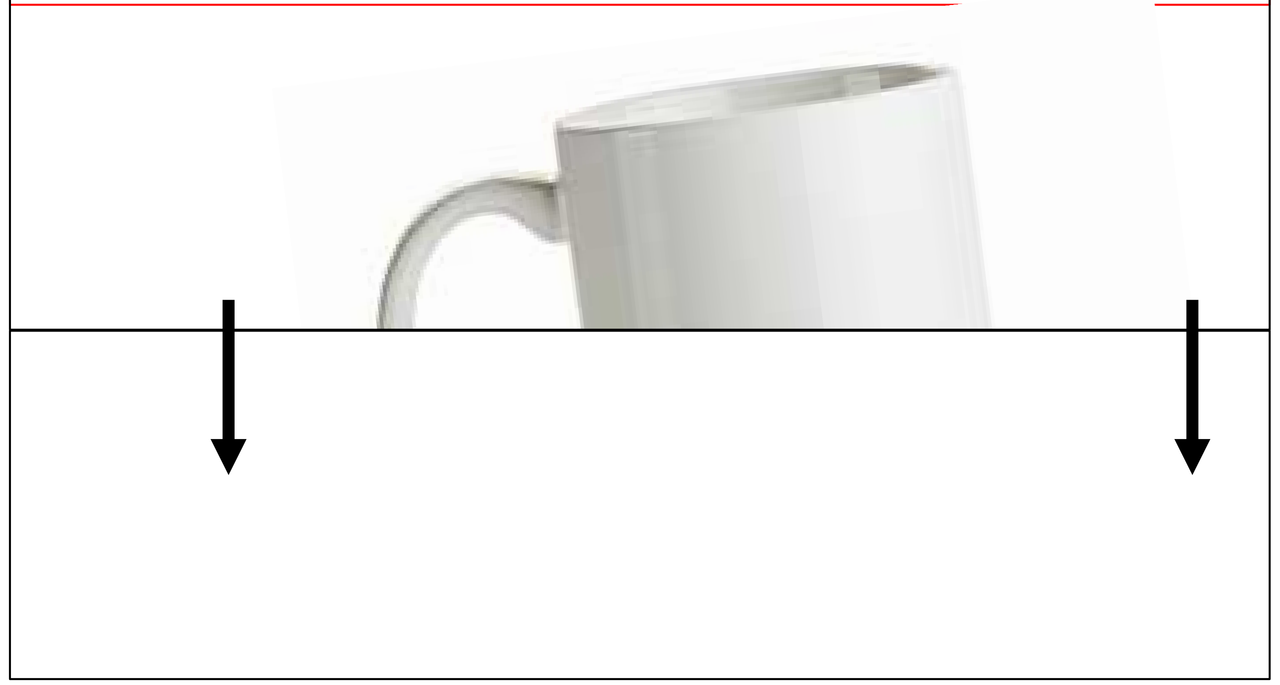
Self-supervision and prediction

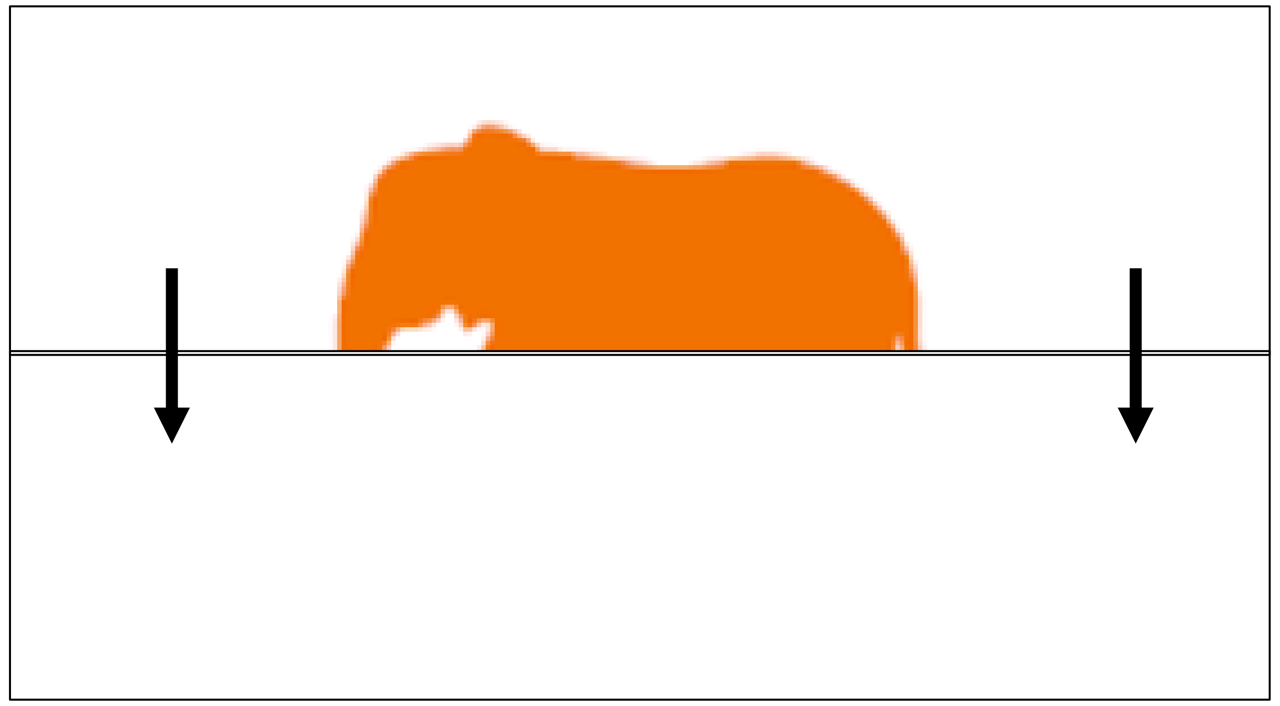
Predict across saccade far away!

Prediction impossible if not the same object!

Views of same object!

‘sameness’ info





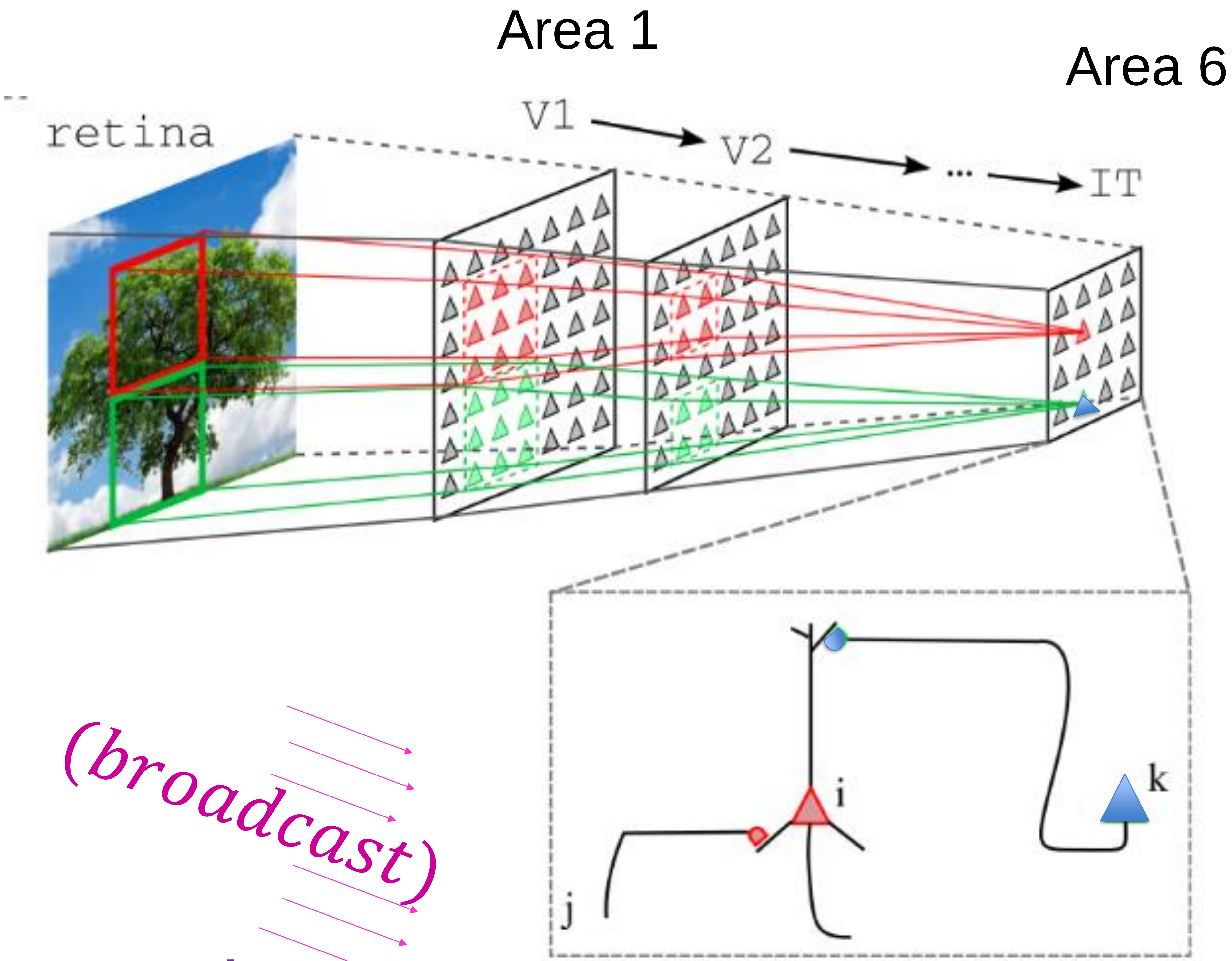
Predict lower half!
Possible if same object!

Predict lower half!
Possible if same object!

Previous slide.

Now we explore an implementation of the CLAPP rule using the scenario of contrastive learning as explained before

Network architecture: lateral feedback provides context



(broadcast)

saccade *or not*!

lateral input influences
Changes of feedforward
synapse on basal dendrite

Aceituno et al. (2024)

Previous slide.

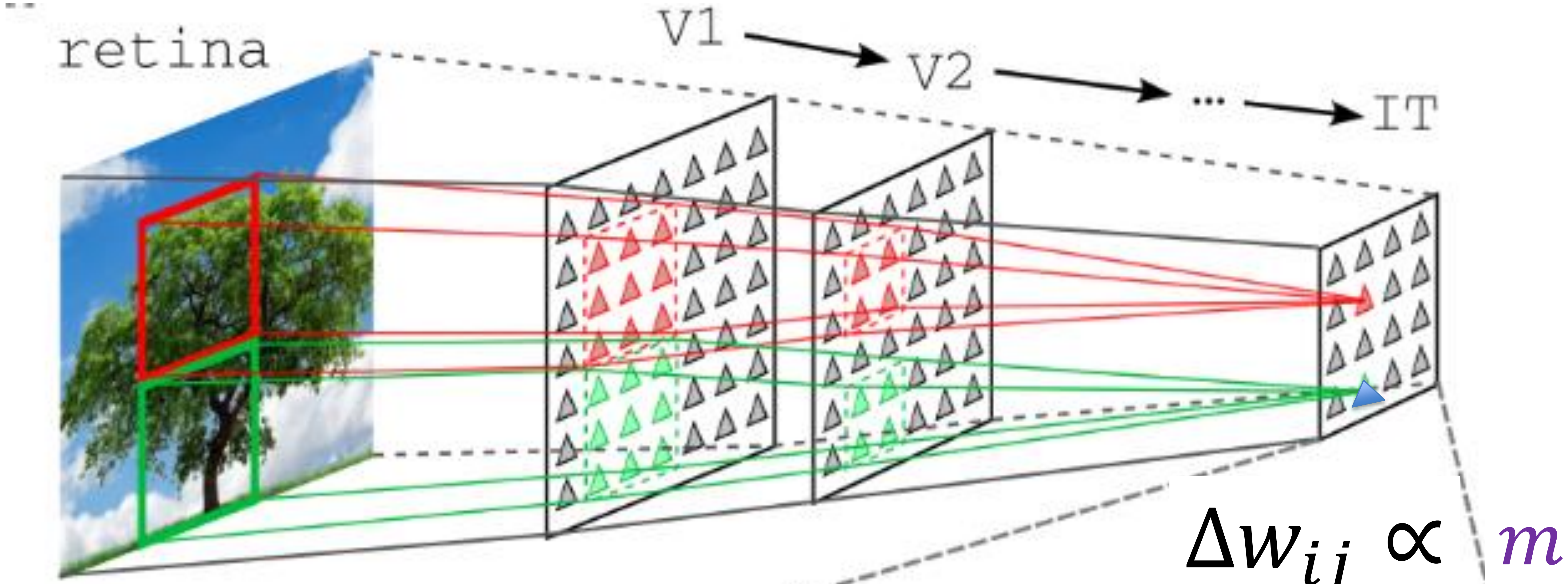
The aim is to predict in a given area (say IT) the activity of the red neuron from the activity of other neurons (e.g., the blue neuron) in the same area

We zoom in onto the red neuron. The text describes the learning rule that we just derived.

Local Learning Rule

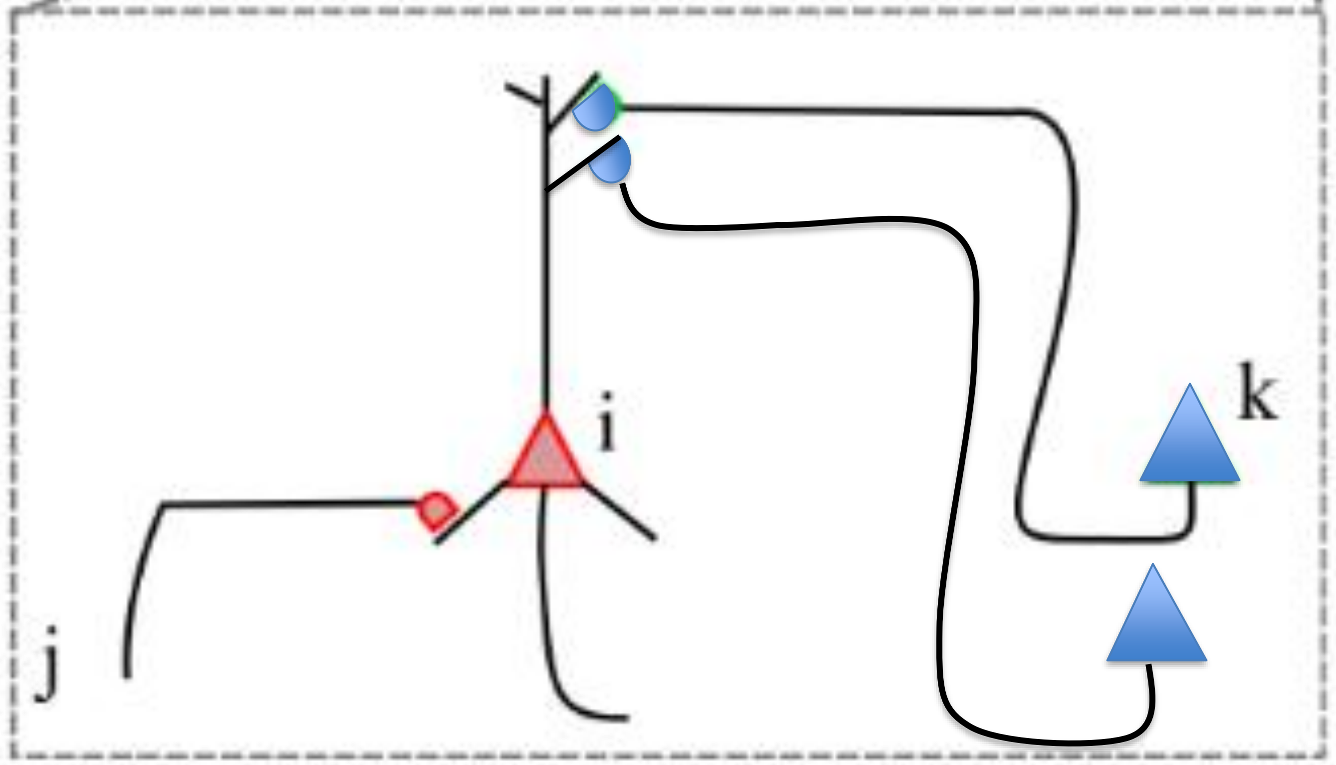
Area 1

Area 6



similar learning rule for lateral weights w_{ik}^{lat}

$$\Delta w_{ij} \propto mod(t)\; lat_i(t)\; post_i(t) pre_j(t)$$



$$lat_i(t) = \sum_k w_{ik}^{lat} post_k(t-\Delta)$$

lateral input influences **changes of** feedforward synapse on basal dendrite

(broadcast)

saccade *or not*!

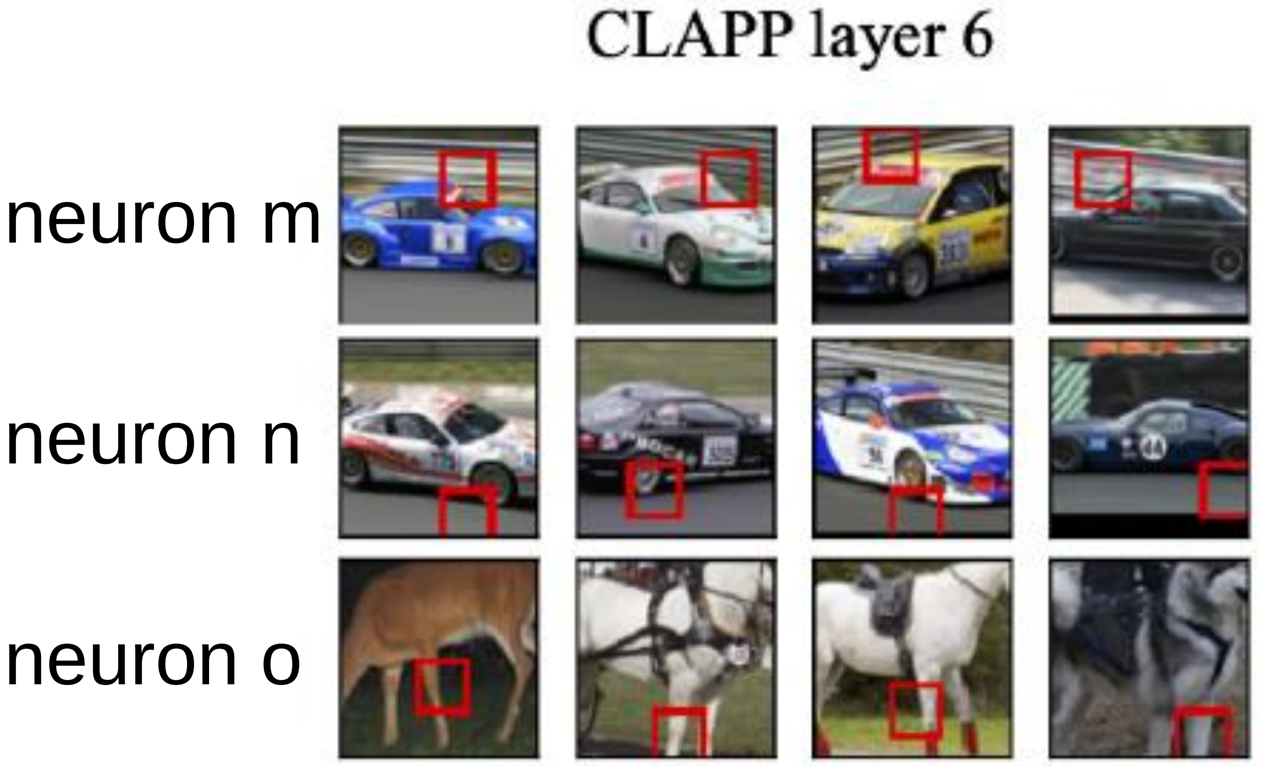
Previous slide.
Same slide again, but now with the formula for the weight change that we derived.

Performance

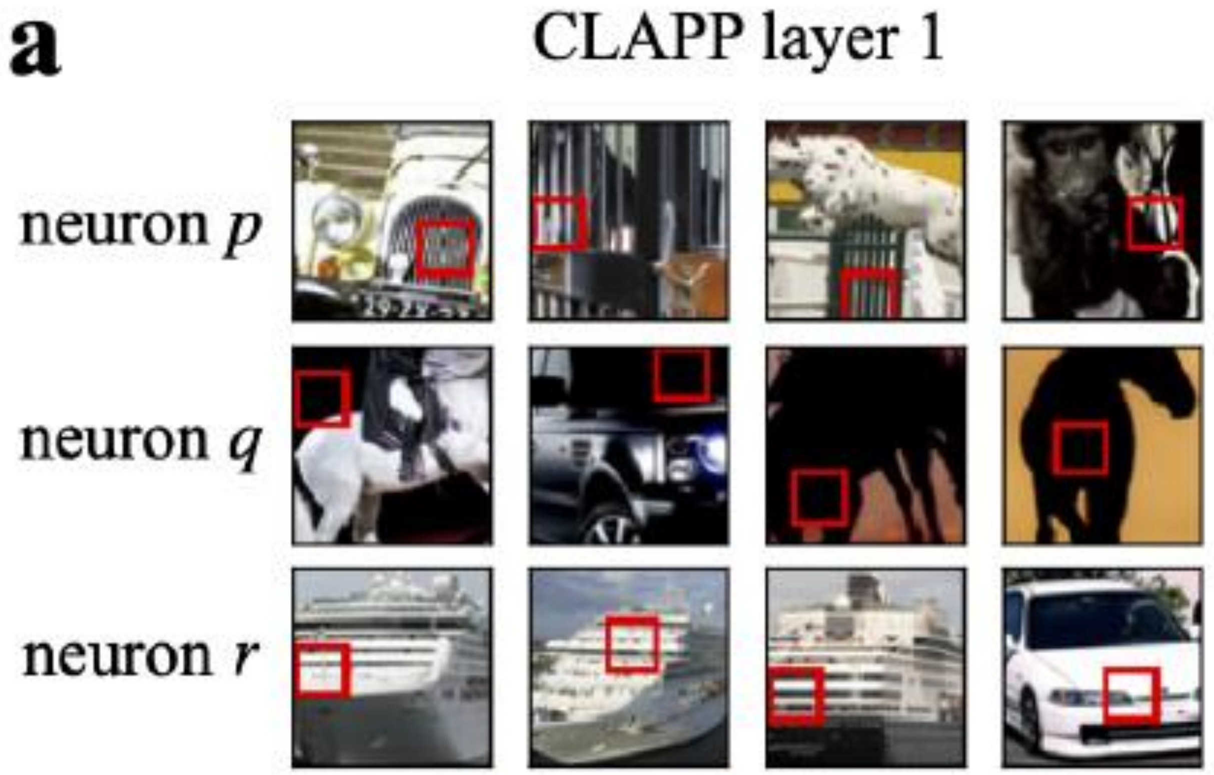
Train (STL 10 data base):
100 000 **unlabeled** images 96x96
6 convolution layers (3x3)
4 maxpool layers (2x2)

Representation Test:
data from 10 classes.
800 test images per class
(used to color the clusters here)

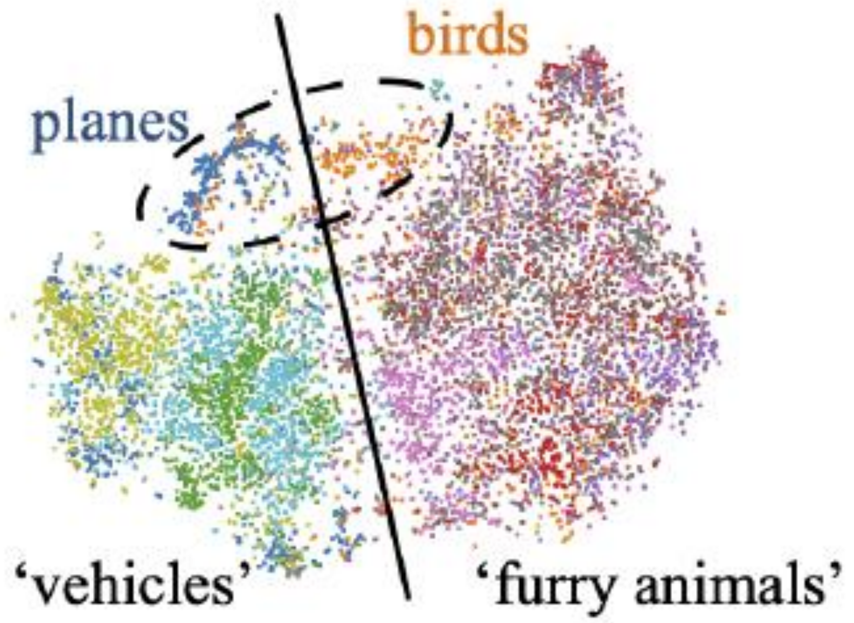
area 6



area 1



Area 6



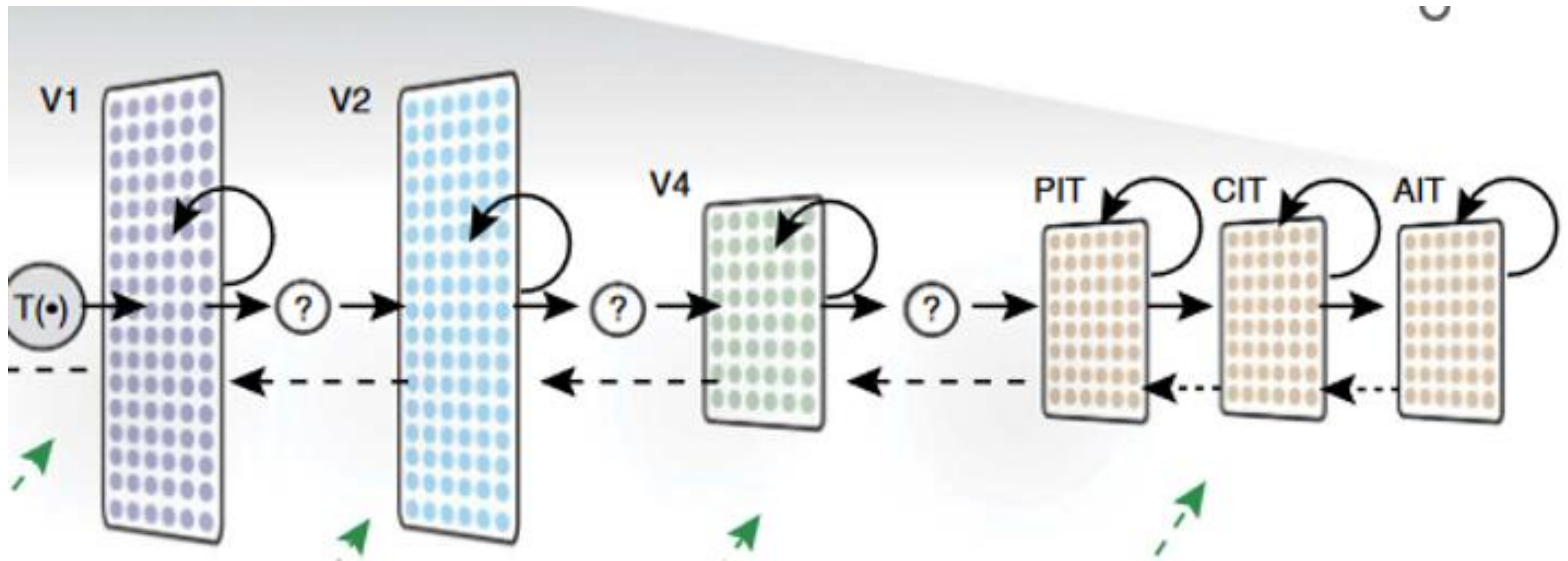
Bernd Illing et al. NeurIPS 2021

Previous slide.
Performance of the CLAPP learning rule in a 6-layer network using the STL10 database.

STL was constructed for self-supervised learning. The training set consists of images without label. After layerwise training of a convolutional network with the CLAPP rule, Neurons in area 1 (model of V1) respond to horizontal or vertical stripes (3 example neurons shown).
However, neurons in area 6 (last step of IT) respond to more abstract concepts like leg of an animal or bottom of a car close to a tire.

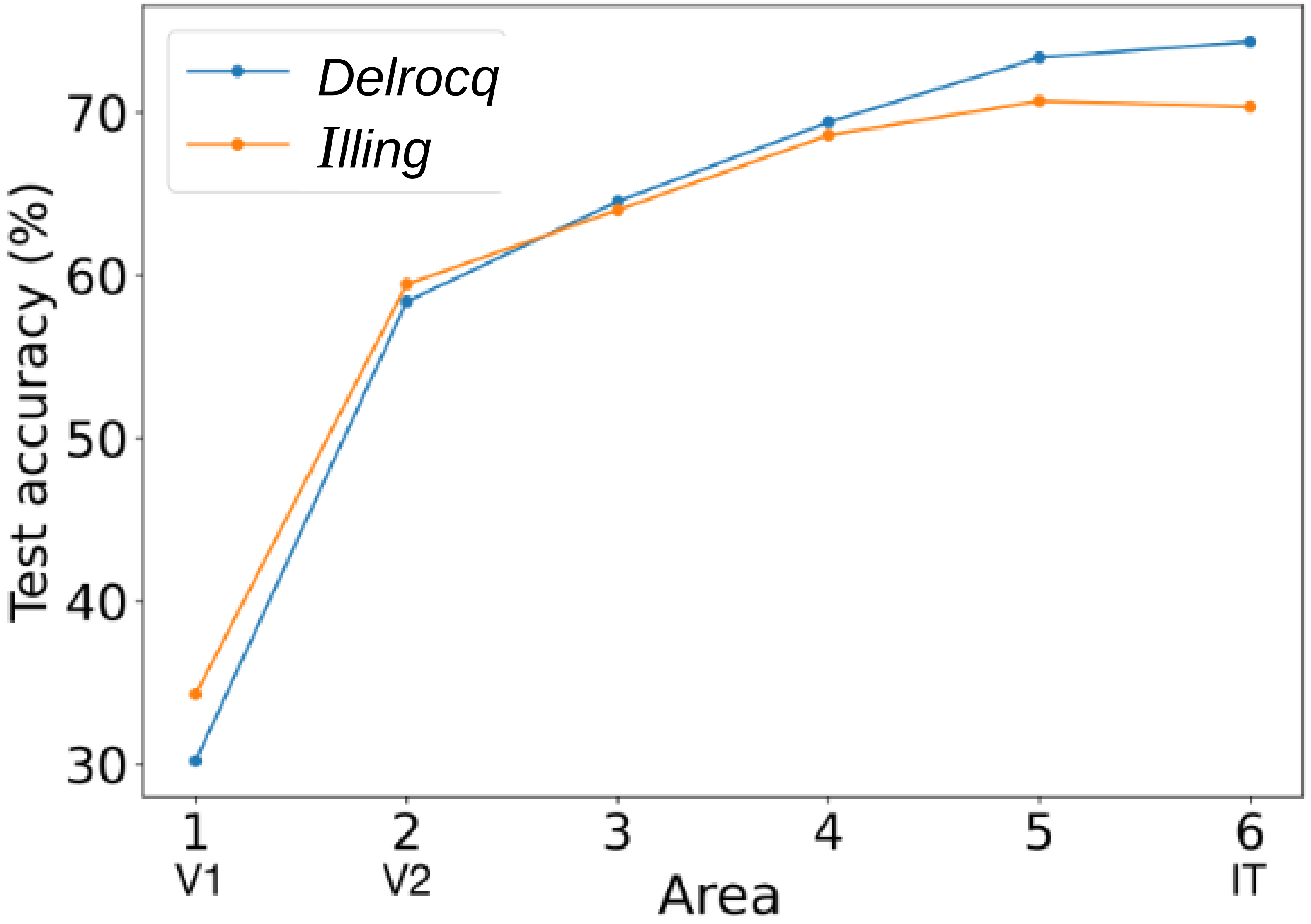
Area 6 contains thousands of neurons each responding with an high or low activity to a new image. We now take the test set and project the activation state of all neurons in Area 6 down to two dimensions. Points are colored according to the class label. We observe that images of planes (blue points) have a representation close to each other but somewhat separated from those of birds (brownish points).

Test on STL-10 image base



→ Raise arm if airplane

Usefulness of representation increases



Train:
no labels

Test:
Labeled data from 10 classes, used to train linear classifier

Bernd Illing et al. NeurIPS 2021
Delrocq et al, bioRxiv, 2024

Previous slide.
The usefulness of the representation is measured by the quality of linear readout (classification).
We find that in Area 1 the representation is not useful, but usefulness increases up to Area 6.
Hence with the representation of Area 6 it would now be possible to use a three-factor rule and reward-based learning to acquire a skill such as raising the arm each time you see an airplane.

Summary:

- AI is extremely powerful today, but trained with BackProp
- Human brain is extremely powerful, but without BackProp
- ‘Learning rules’ of the brain still largely unknown
- Learning rules are important research topic
- **Representation learning is possible with local rules**
- **Predictions are important for learning**

→ **Good representations starting point for many things!**

Literature:

- *Timothy P. Lillicrap et al.,* Backpropagation and the brain, *Nature Reviews Neurosci. 21: 335-346 (2020)*
- *Bernd Illing et al.,* Local Plasticity rules can learn deep representations, *35th NeurIPS (2021)*
- *Ariane Delrocq et al,* Critical periods support representation learning in a model of cortical processing, *bioRxiv, 2024.12. 20.629674 (2024)*
- *Pau Aceituno et al*., Target learning rather than backpropagatin explains learning in the mammalian neocotex, *bioRxiv, 2024.04.10.588837 (2024)*
- *C. Clopath, L. Busing, E. Vasilaki and W. Gerstner (2010)* Connectivity reflects coding: a model of voltage-based spike-timing-dependent-plasticity with homeostasis.. *Nature Neuroscience 13, pp. 344–352 (2010)*
- *A. Van den Oordt, Y. Li, O. Vinyals,* Representation Learning with Contrastive Predictive Coding, *ArXiv (2018)*

Previous slide.
Summary.

No Backprop, please!
Learning of ‘deep’ representations

Wulfram Gerstner

EPFL, Lausanne, Switzerland

1) Introduction (review)
2) Plasticity and local learning rules (review)
3) Contrastive Selfsupervised Learning
4) Representation Learning with CLAPP:
   “Contrastive, Local And Predictive Plasticity”
5) **Feedback Alignment**

Previous slide.

No Backprop, please!
Learning of ‘deep’ representations

Wulfram Gerstner

EPFL, Lausanne, Switzerland

Alternatives to CLAPP?

- Feedback Alignment/Deep Feedback Alignment
- Predictive Coding

Literature:

Timothy P. Lillicrap et al., Backpropagation and the brain, Nature Reviews Neurosci. 21: 335-346 (2020)
Bernd Illing et al., NeurIPS (2021), Local Plasticity rules can learn deep representations, 35th NeurIPS (2021)
T. P. Lillicrap, D. Cownden, D. B. Tweed, and C. J. Akerman. Random synaptic feedback weights support error backpropagation for deep learning. Nature communications, 7(1):13276, 2016.
J. C. Whittington and R. Bogacz. An approximation of the error backpropagation algorithm in a predictive coding network with local hebbian synaptic plasticity. Neural computation, 29(5): 1229–1262, 2017
A. Nøkland. Direct feedback alignment provides learning in deep neural networks. Advances in neural information processing systems, 29, 2016.

Previous slide.

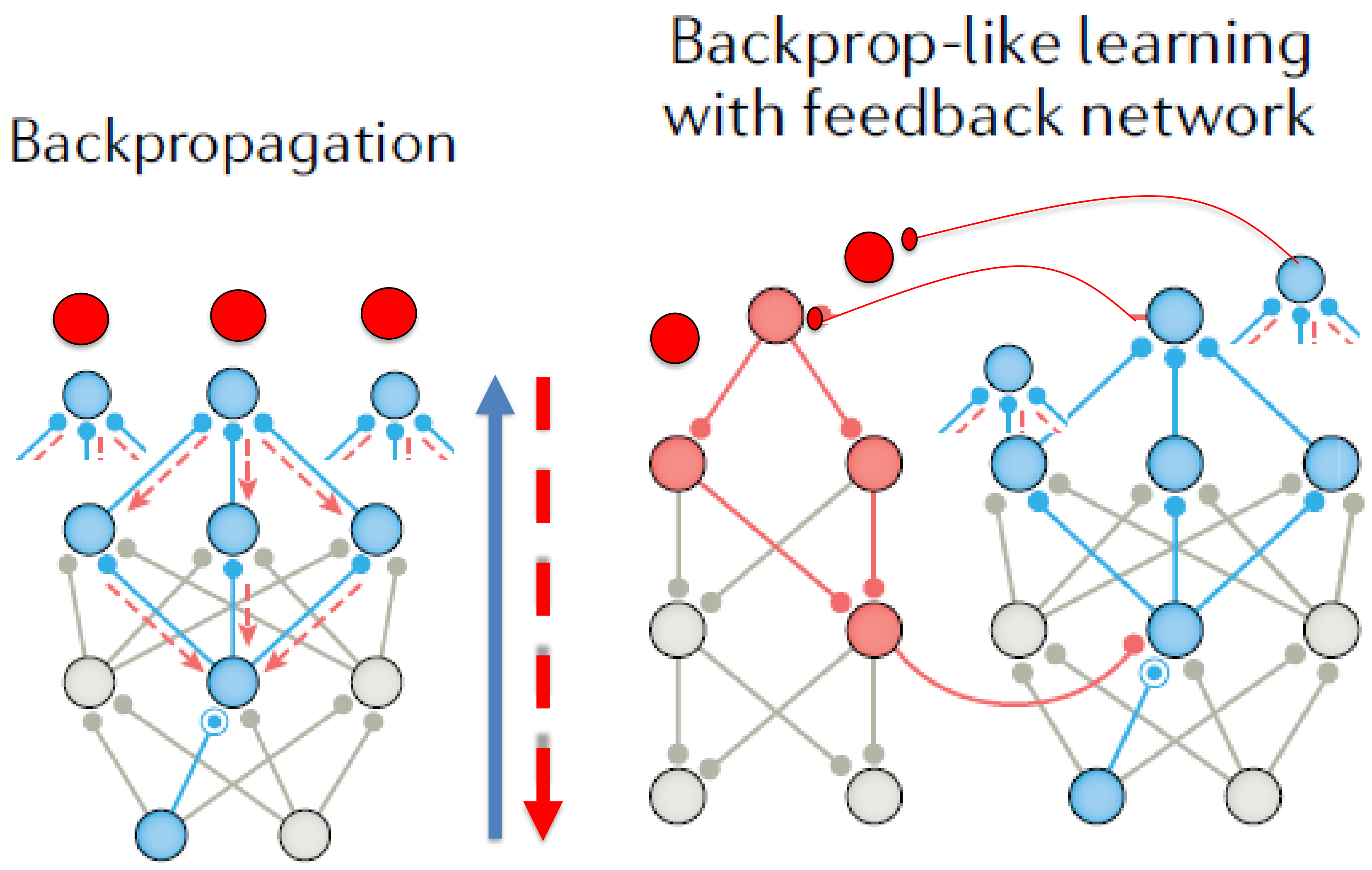
Backprop needs precise error feedback

Vector feedback:

- multiple outputs,
- one 'signed error per output'
- error vector transmitted back
- precise neuron-specific errors

BackProp Algo has 4 phases:

1) Forward pass and freeze
2) Calculate local output errors
3) Backprop pass, using 2)
4) Update connections, using 1) +3)



BackProp rules:
vector
feedback

Adapted from Lillicrap et al. 2020 Nat. Rev. Neurosci.

Previous slide.

How does BackProp work? Minimize errors!

- BackProp needs four separate phases:

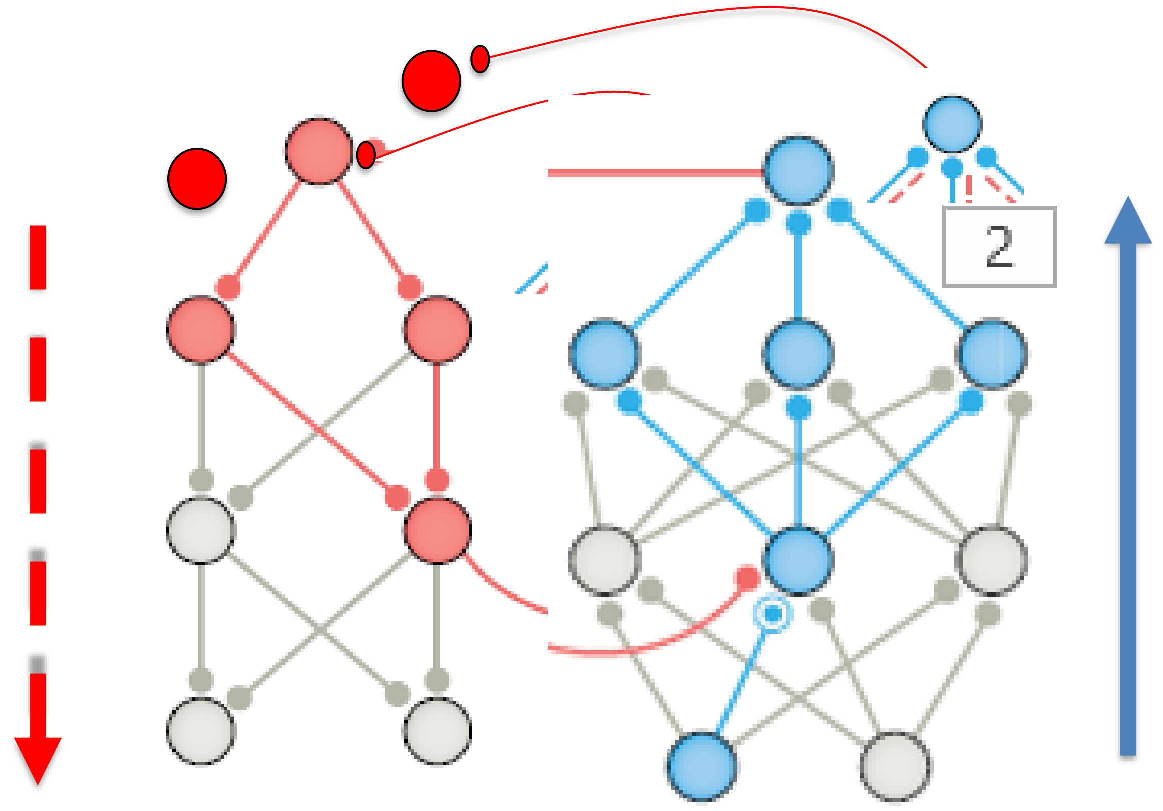
  forward pass, output mismatch, backward pass, weight update.

- Backward pass needs specific feedback architecture

  1) feedback weights = feedforward weights: ‘weight transport;

  2) backward multipliers depend on state
  of feedforward network (

→ Not implementable in biology!



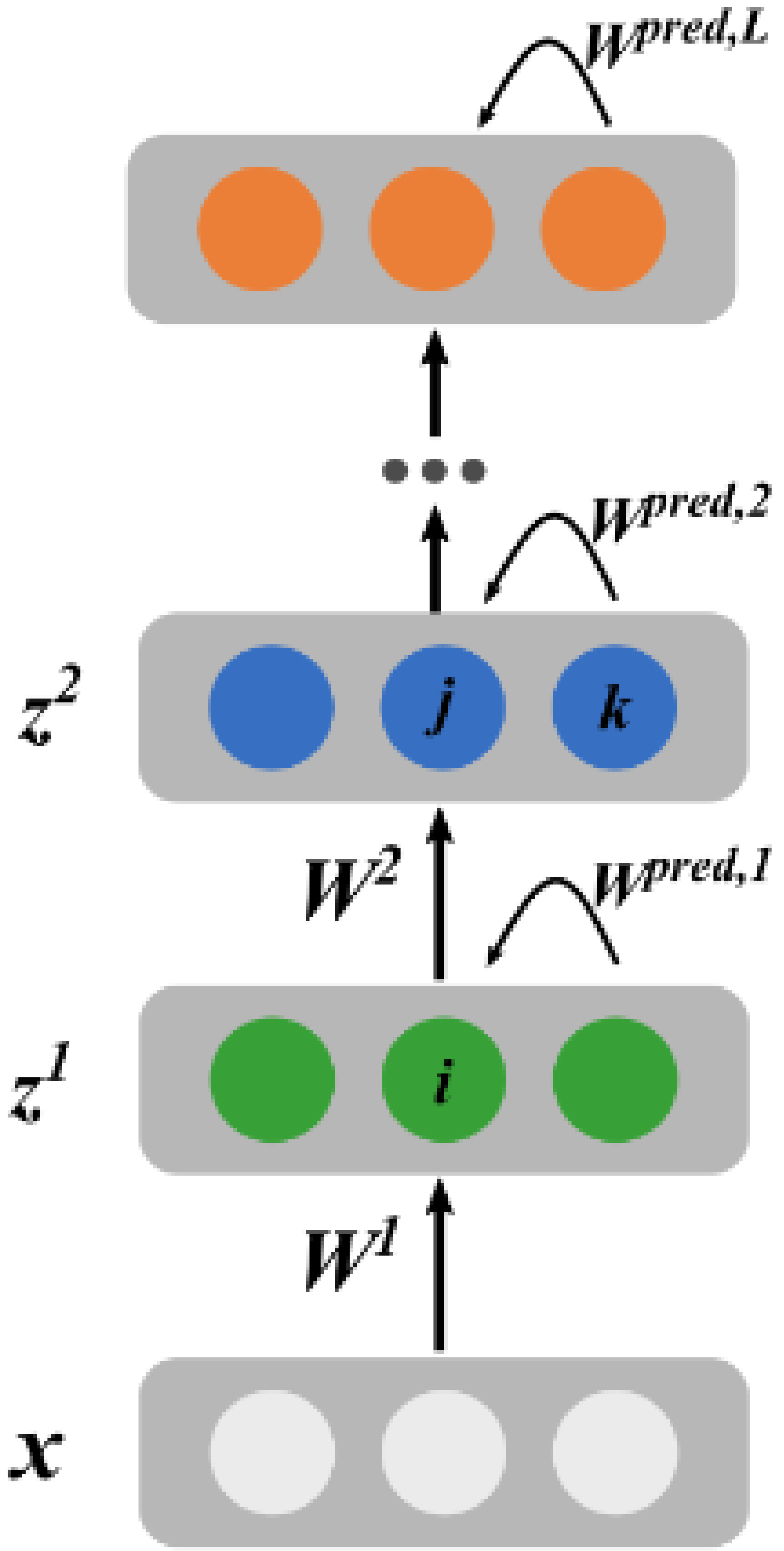
Backprop-like learning
with feedback network

F. Crick, The recent excitement about Neural Networks, Nature 337:129-132 (1989)
T.P. Lillicrap et al., Backpropagation and the brain, Nature Reviews Neurosci. 21: 335-346 (2020)

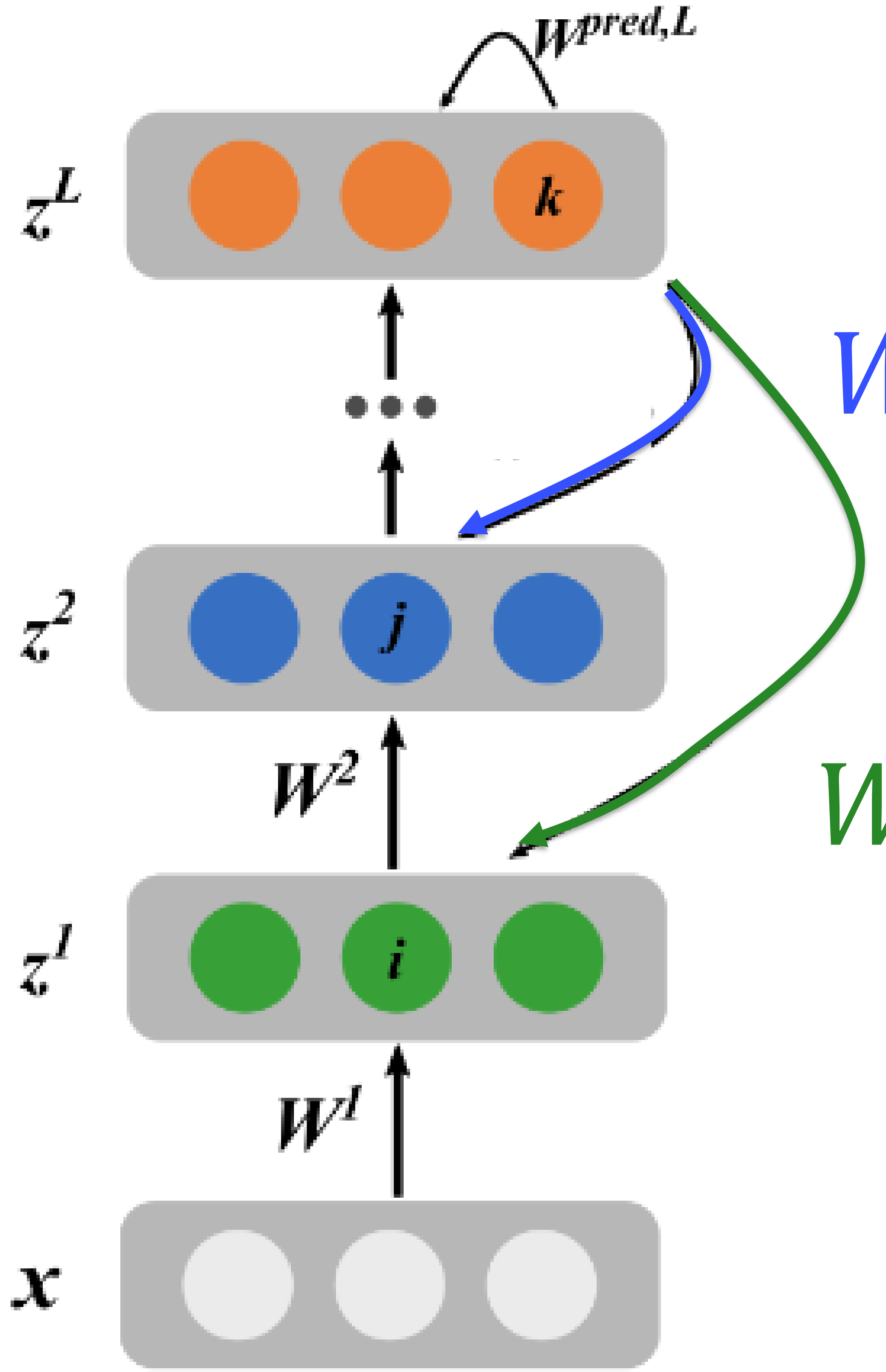
Previous slide.

Idea of feedback alignment: use random feedback weights

hinge-loss, layerwise:
CLAPP



hinge-loss with FA,
Feedback Alignment



$$W_{FA}^{(2)} = 'random'$$

$$W_{FA}^{(1)} = 'random'$$

Previous slide.
Direct feedback alignment is simple:
We measure the local mismatch in each output neuron, and then we replace the exact BackProp signal by a fixed random matrix.

The term feedback alignment arises from the observation that if you apply this idea for the weights leading up to the output layer, then the forward weights learn to become similar (‘aligned’) to the fixed random feedback weights.

Note that in backprop the forward and the backward weights should be identical.

However, for multiple layers (here 6 layers) feedback alignment does not work.

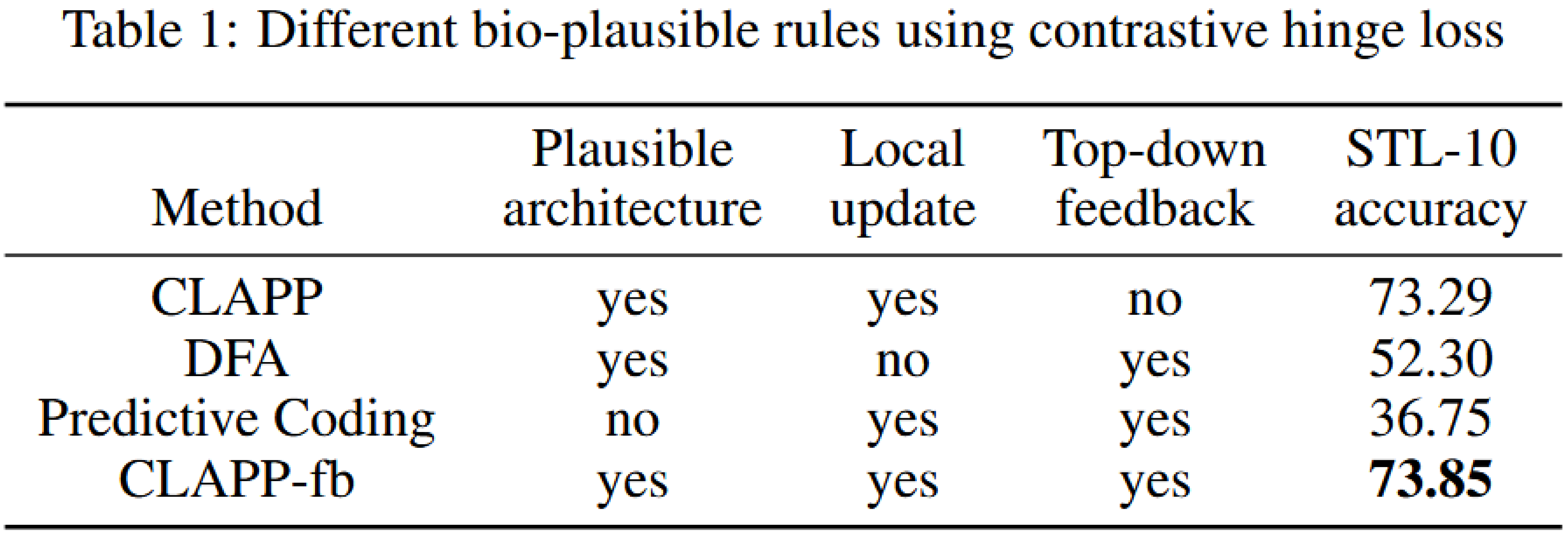


Table 1: Different bio-plausible rules using contrastive hinge loss

Method	Plausible architecture	Local update	Top-down feedback	STL-10 accuracy
CLAPP	yes	yes	no	73.29
DFA	yes	no	yes	52.30
Predictive Coding	no	yes	yes	36.75
CLAPP-fb	yes	yes	yes	**73.85**

DFA: Direct Feedback alignment, a variant of FA.

A. Nøkland. Direct feedback alignment provides learning in deep neural networks. Advances in neural information processing systems, 29, 2016.

Previous slide.

Two version of CLAPP (the second one learns feedback weights instead of lateral weights) perform much better on the STP10 task, then Direct Feedback Alignment.

Summary: Selfsupervised Learning

Contrastive Learning:

- Predict left part of image from right part
- Predict original image from augmented image

→ Align representation in representation layer

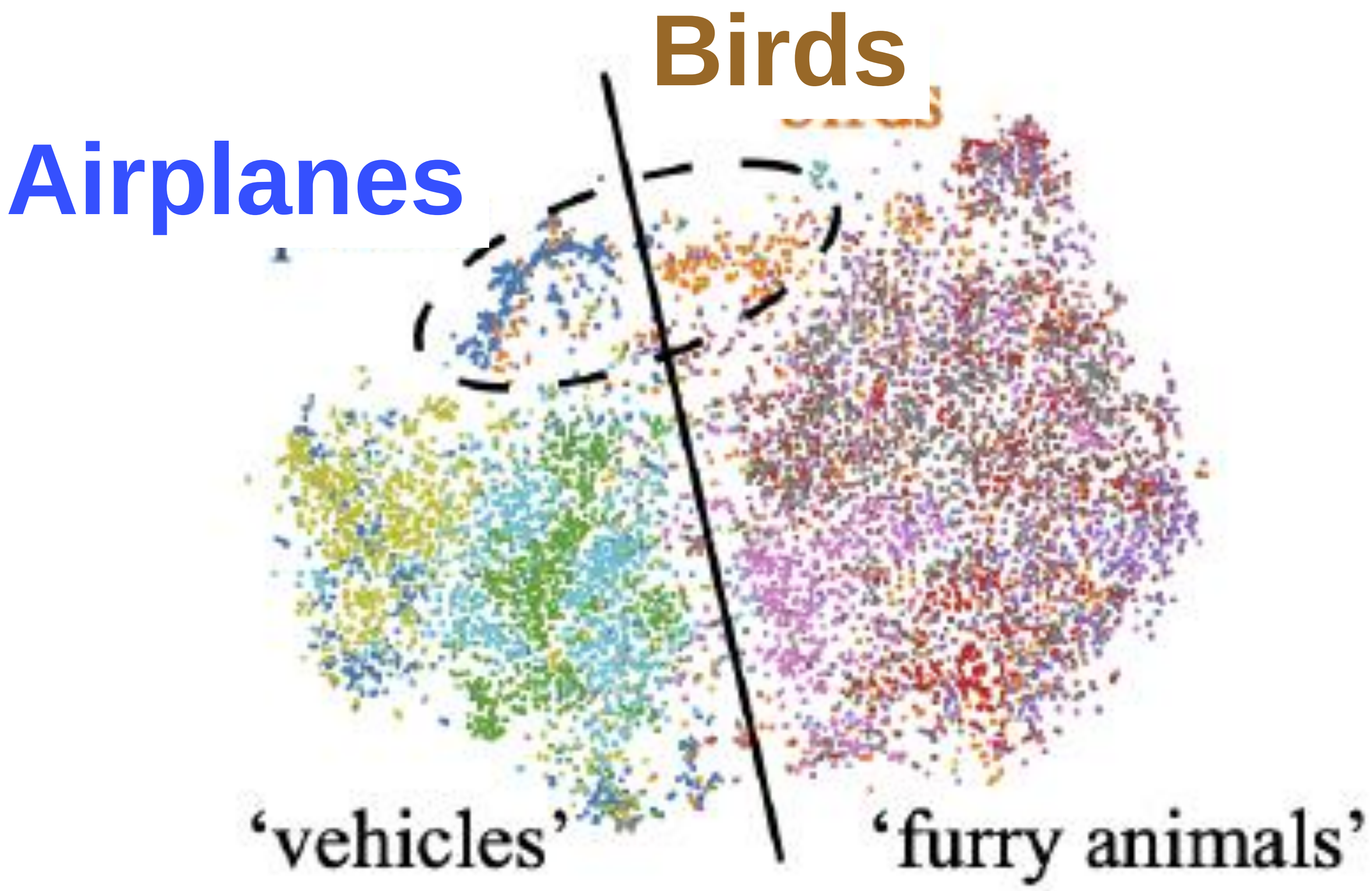
BUT:

- **Avoid collapse of representation by ‘negative samples’**
- Use negative samples that the network should not predict

→ Move different ‘objects’ far from each other

Non-contrastive Learning:

- **Avoid collapse by normalization**
- All neurons should be used
- Neurons do different things



Previous slide.

Additional background literature:

A.Artola, S.Bröcher and W. Singer (1990) Different voltage dependent thresholds for inducing long-term depression and long-term potentiation in slices of rat visual cortex*. Nature 347, pp. 69–72.*

A.Ngezahayo, M.Schachner and A.Artola (2000) Synaptic activation modulates the induction of bidirectional synaptic changes in adult mouse hippocamus*. J. Neuroscience 20, pp. 2451–2458.*

P.J. Sjöström, G.G. Turrigiano and S.B. Nelson (2001) Rate, timing, and cooperativity jointly determine cortical synaptic plasticity. Neuron 32, pp. 1149–1164.

C.Clopath, L. Busing, E. Vasilaki and W. Gerstner (2010) Connectivity reflects coding: a model of voltage-based spike-timing-dependent-plasticity with homeostasis. *Nature Neuroscience 13, pp. 344–352 (2010)*

L. Muckli et al. (2015) Contextual feedback to superficial layers of V1. *Current Biol. 25: 2690–2695*

G.B. Keller and T.D. Mrsic-Flogel (2018 Predictive Processing: a canonical cortical computation. *Neuron:424-435*

*A. Keller, Roth, Scanziani**.*** Feedback generates a second receptive field. *Nature **582**: 545–549 (2020)*

J. Homann … M.J. Berry, (2022) Novel stimuli evoke excess activity in the mouse primary visual cortex. *Proc. Natl. Acad. Sci (USA) 119:e2108882119*

M.S. Halvagal and F. Zenke, (2023) The combination of Hebbian and predictive plasticity learns invariant object representation … *Nat. Neurosci. 26:1906-1915*