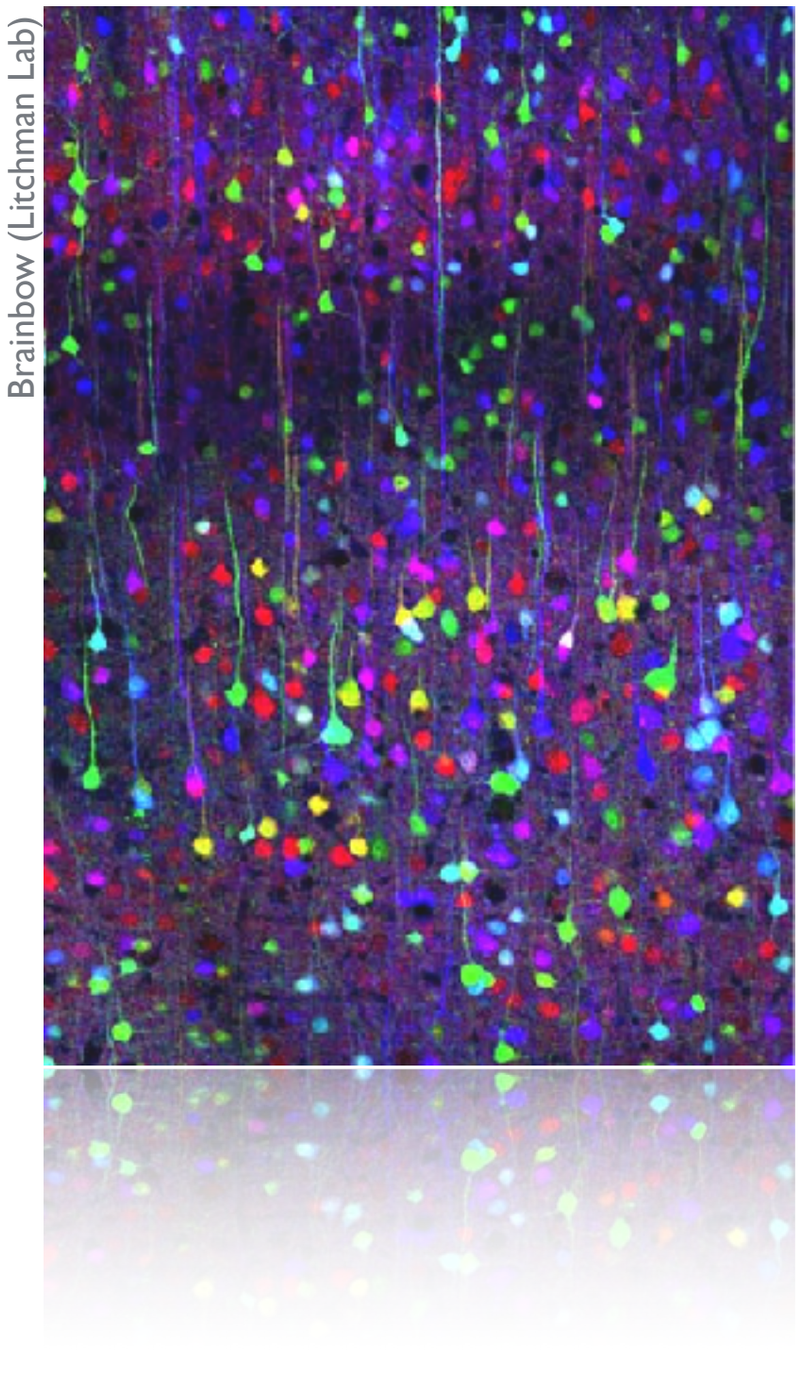
Neural Information Processing 2018/2019

University of BRISTOL



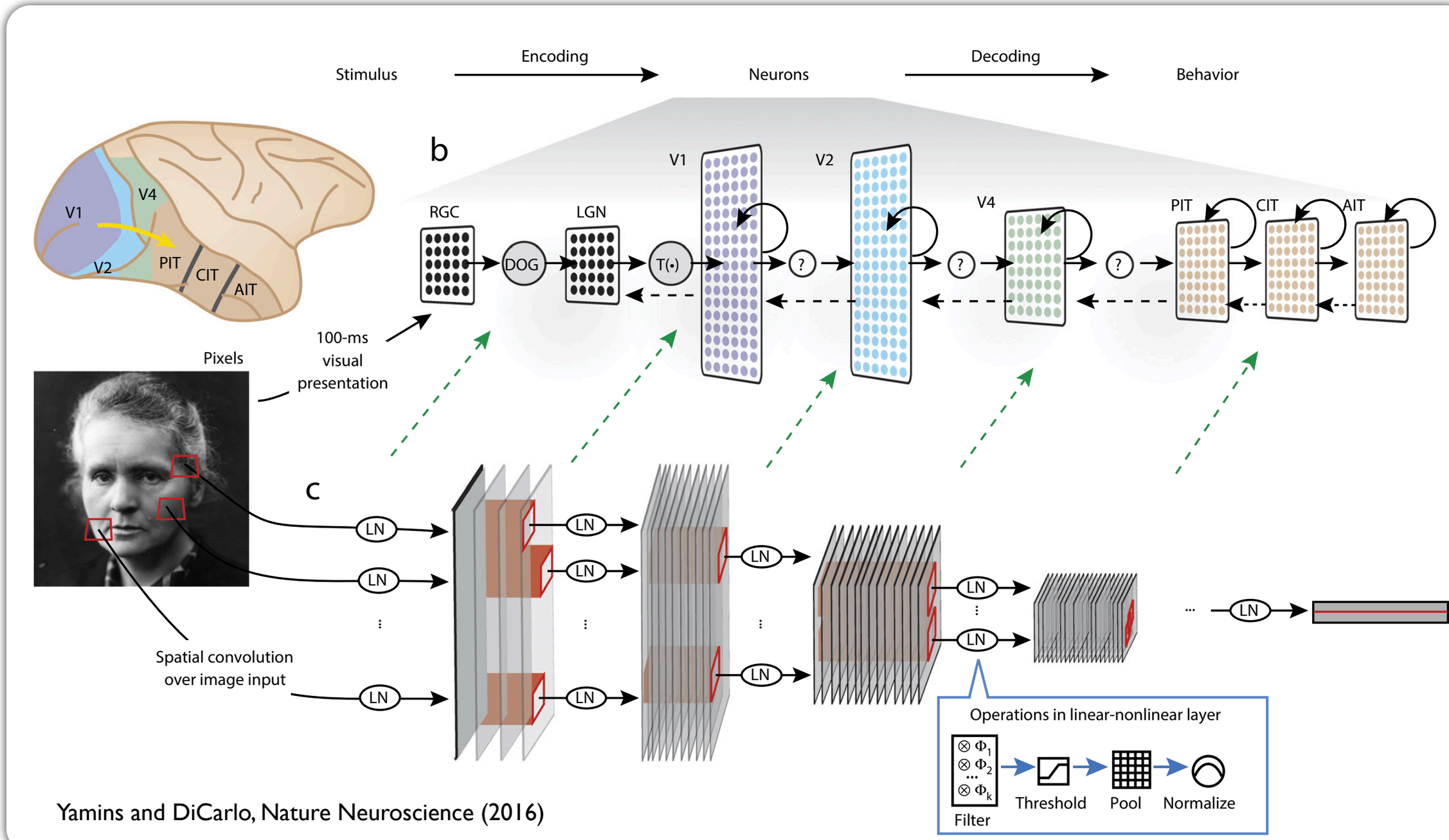
Lecture 12
Neural circuits and learning:
Supervised learning:
backprop & cerebellum

Extra lab session

Next Wednesday (5th of December)!

Sign up for help with course work:
https://doodle.com/poll/tbky2r8st4xxc3kb

Previously on *Neural Information Processing*...

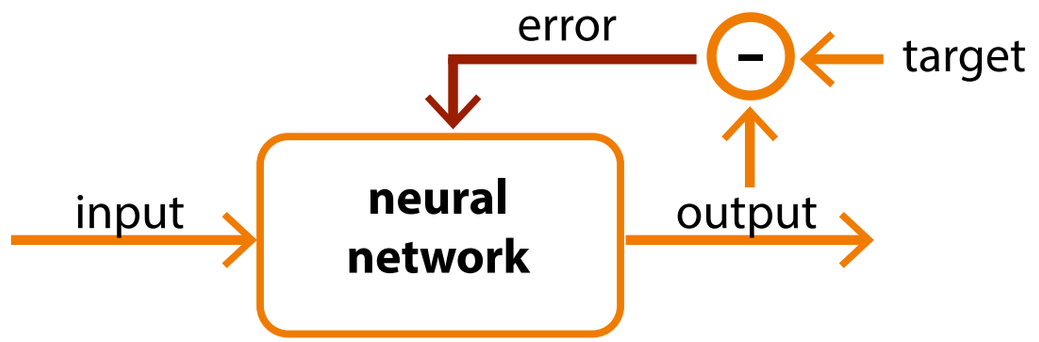


Yamins and DiCarlo, Nature Neuroscience (2016)

But, how to exactly train neural networks?

Supervised Learning:

Relies on a teaching signal



Solution: The backprop algorithm which is the workhorse of deep learning!

Outline

1. **The backpropagation algorithm**

2. **Backprop in the brain**

3. **Cerebellum, the brain supervisor?**

The backpropagation of errors algorithm

- **Backpropagation** is an algorithm used to train artificial neural networks. It calculates a gradient from the error function with respect to a given parameter (i.e. weight). With this gradient errors are effectively back propagated during training throughout the network.

$$\Delta w = \eta(v - \text{target})f'(wu)u$$

- It is a generalisation of the **delta rule** (above) to deep feedforward networks, which relies on the use of the chain rule to compute gradients for each layer (see next slides).

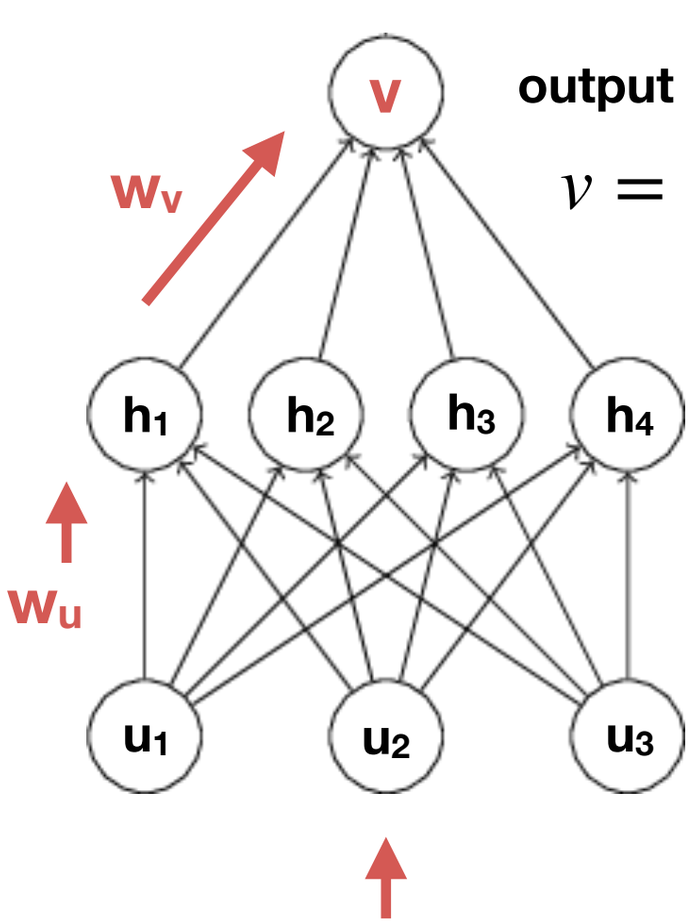
- **Backpropagation** is typically used to perform gradient descent optimisation, adjusting the weights of neurons while optimising a desired loss function (error).

Rumelhart et al. 1986 PDP

The backpropagation algorithm

The backpropagation algorithm:

1. Calculate forward activity



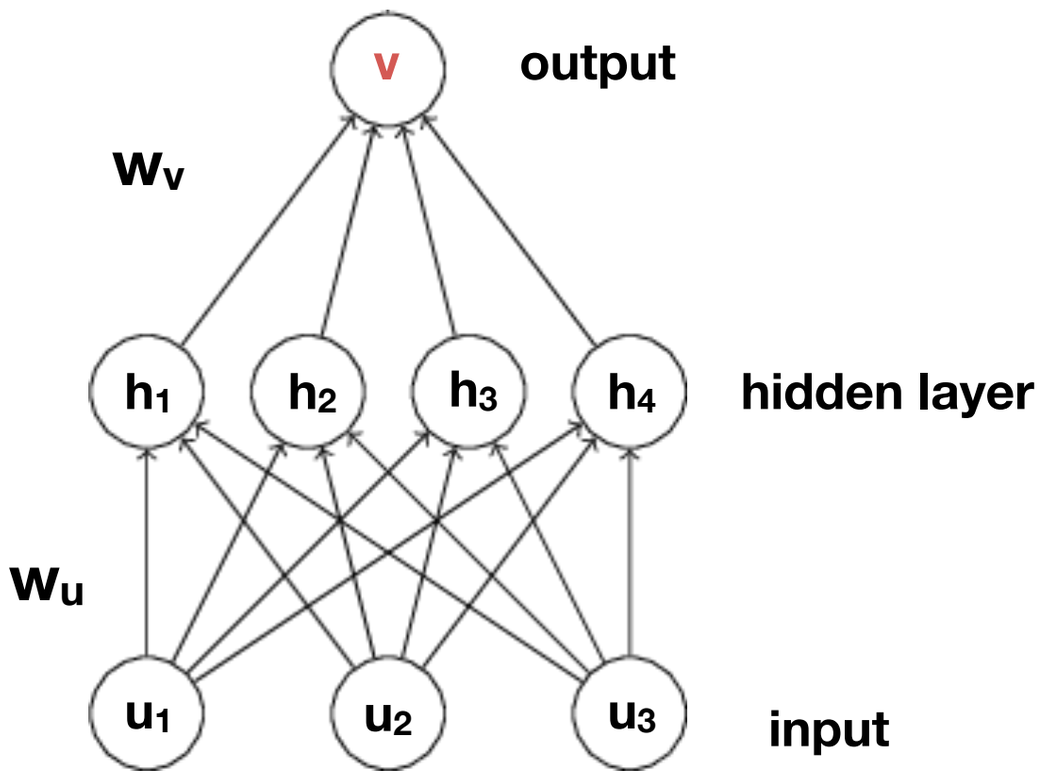
Note: Here, we are assuming a linear activation for ***v***

f denotes the activation function (i.e. input-output function), which is typically modelled as a sigmoid or rectifying linear unit function (ReLU)

Rumelhart et al. 1986 PDP

The backpropagation algorithm

$$\text{error} = \frac{1}{2}(v - \text{target})^2$$

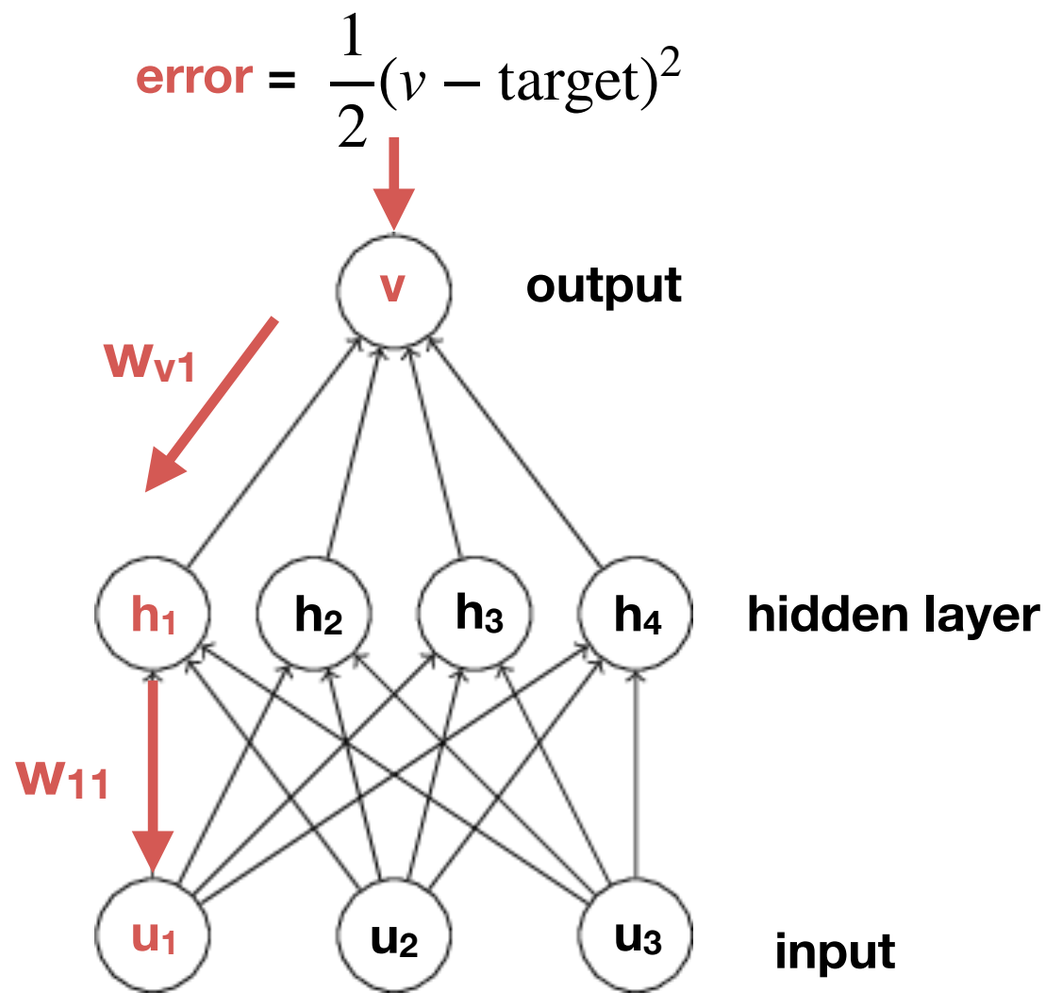


The backpropagation algorithm:

1. Calculate forward activity
2. Calculate **error**

Rumelhart et al. 1986 PDP

The backpropagation algorithm



The backpropagation algorithm:

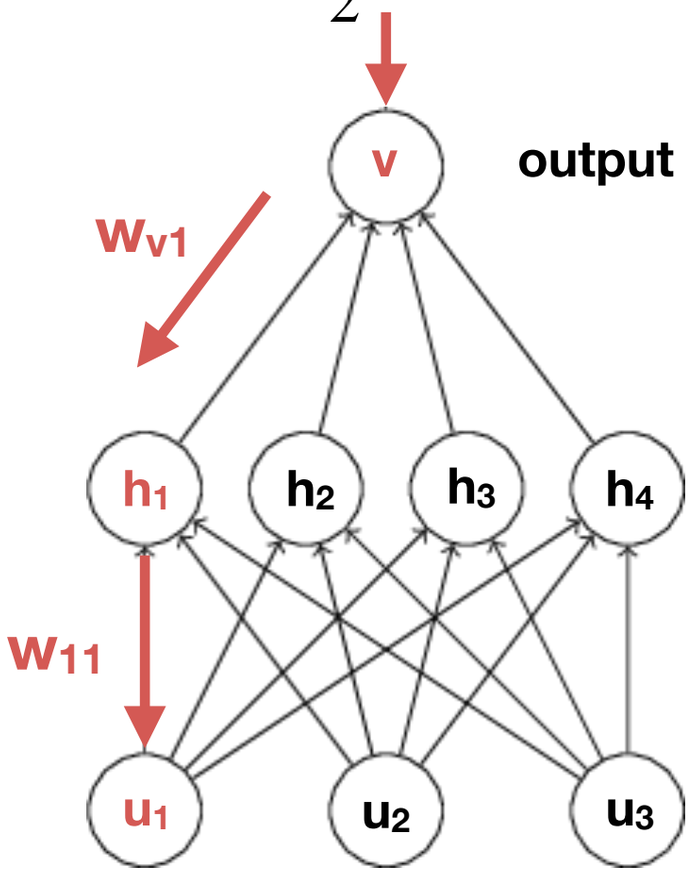
1. Calculate forward activity
2. Calculate error
3. **Backpropagate** error, e.g. how to calculate the gradient wrt w_{11}:

$$\frac{\partial \text{Error}}{\partial w_{11}} = \underbrace{\frac{\partial \text{Error}}{\partial v}\frac{\partial v}{\partial h_1}\frac{\partial h_1}{\partial w_{11}}}_{\text{chain rule}}$$

Rumelhart et al. 1986 PDP

The backpropagation algorithm

$$\text{error} = \frac{1}{2}(v - \text{target})^2$$



Note: Write down the gradient wrt a given weight for practice!

The backpropagation algorithm:

1. Calculate forward activity
2. Calculate error
3. **Backpropagate** error, e.g. how to calculate the gradient wrt w_{11}:

$$\frac{\partial \text{Error}}{\partial w_{11}} = \underbrace{\frac{\partial \text{Error}}{\partial v}\frac{\partial v}{\partial h_1}\frac{\partial h_1}{\partial w_{11}}}_{\text{chain rule}}$$

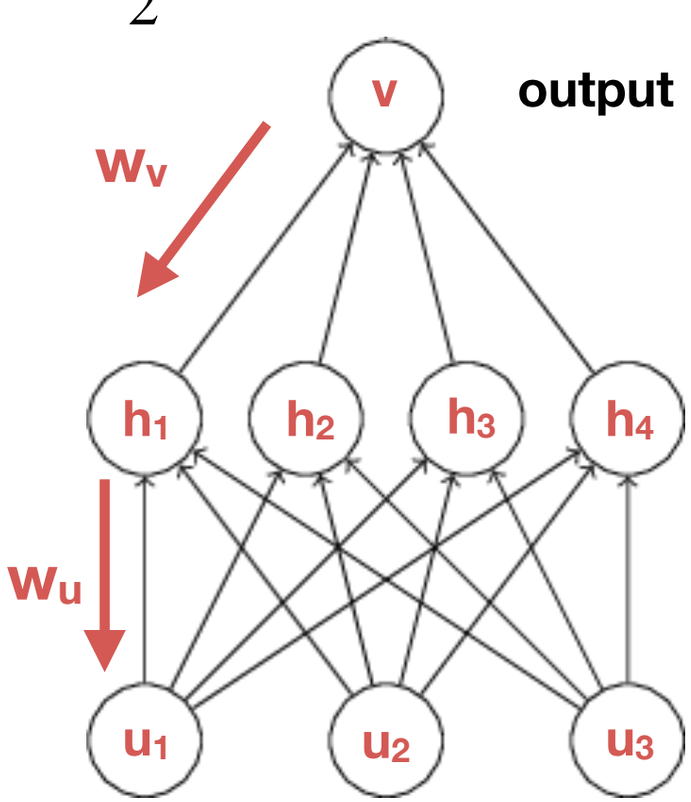
weight update:

$$\Delta w_{11} = -\eta \frac{\partial \text{Error}}{\partial w_{11}} = \underbrace{(\text{v} - \text{target})}_{\text{error}} w_{v1} f'_{h_1} u_1$$

Rumelhart et al. 1986 PDP

The backpropagation algorithm general weight update

error = $\frac{1}{2}(v - \text{target})^2$



The general form for the weight update:

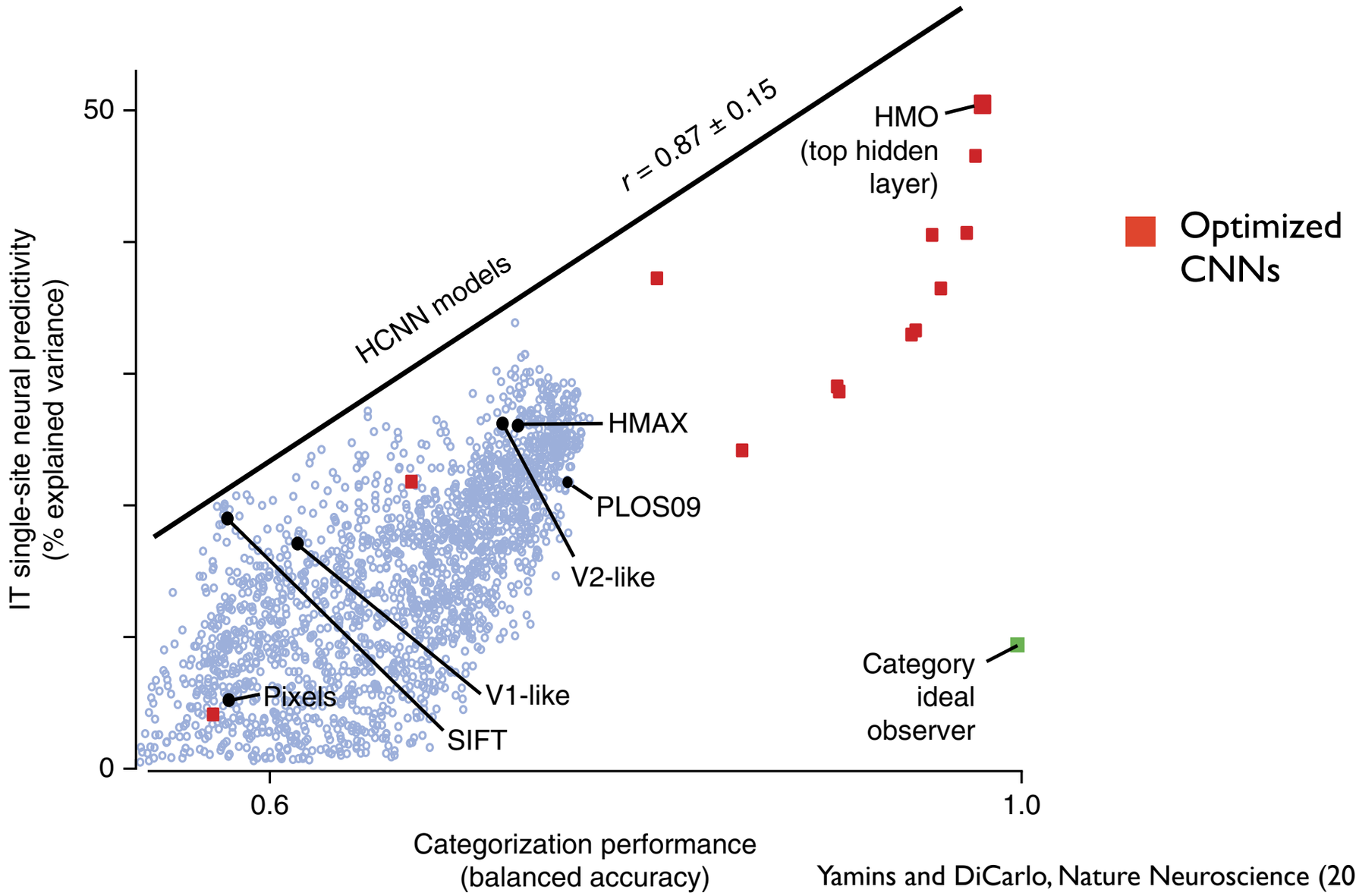
$$\Delta w_u = -\eta \underbrace{(\text{v} - \text{target})}_{\text{error}} \; w_v^T f_h' u$$

Note: Write down the gradient wrt a weight for practice!

Rumelhart et al. 1986 PDP

The backpropagation algorithm in the brain

Reminder: CNNs trained with backprop can predict neural data to a high degree!



Yamins and DiCarlo, Nature Neuroscience (2016)

The backpropagation algorithm
is **NOT** = backpropagating action potential

In biology there is a phenomenon known as action potential backpropagation.
This does **NOT** directly relate to the backpropagation algorithm!

Backpropagating action potential ≠ Backpropagation algorithm

see Stuart et al. TiNS (1997) for more info

The backpropagation algorithm in the brain

However, at a first glance the backpropagation algorithm seems to not be biologically plausible.

Key issues:

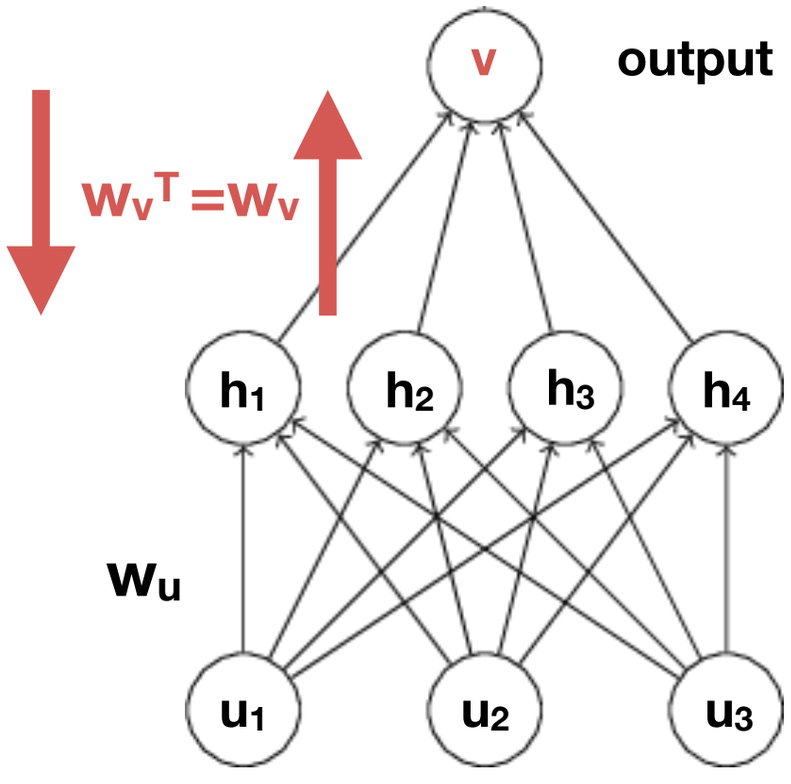
1. **Weight transport problem**: It proposes feedback weights equal to feedforward
2. **Derivative of activation functions**: Relies on calculating derivatives of the activation functions
3. **Two phase learning**: (1) feedforward propagation of activity and (2) error backpropagation
4. **Separate error network**: Suggest the need for a separate biological network computing the learning rules
5. **Non-local learning rules**: The weight update depends on non-local information
6. **Target**: The target label guides supervised learning

Useful references:
Roelfsema and Holtmaat, Nature Neuroscience Rev (2018)
Richards and Lillicrap, Current Opinion in Neurobiology (2019)

The backpropagation algorithm
and why it is at odds with biology
1. Weight transport problem

Backprop: The chain rule predicts feedback weights that are equal to feedforward weights: $W_v^T = W_v$

Biology: No evidence that feedback connections exactly mirror feedforward ones.



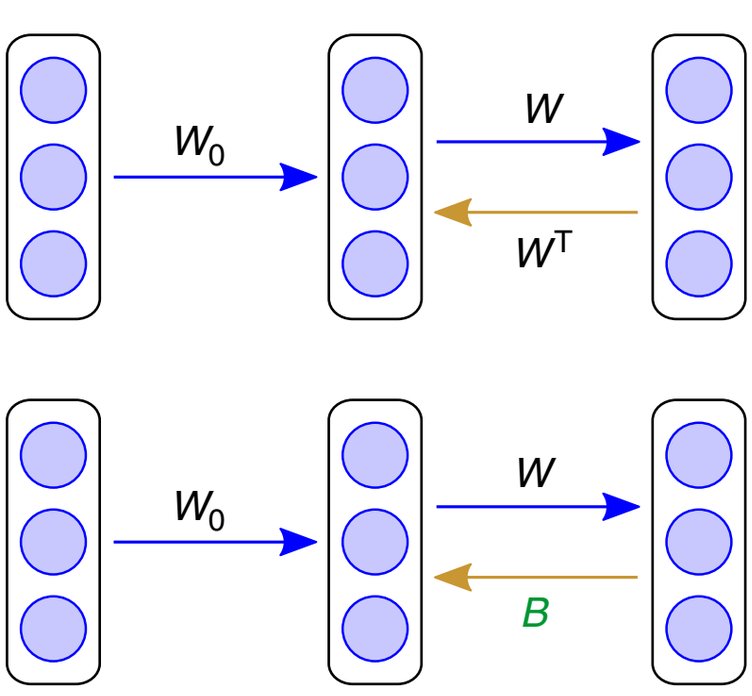
$$\Delta w_u = -\eta \underbrace{(\mathrm{v} - \mathrm{target})}_{\text{error}} \; w_v^T f_h' u$$

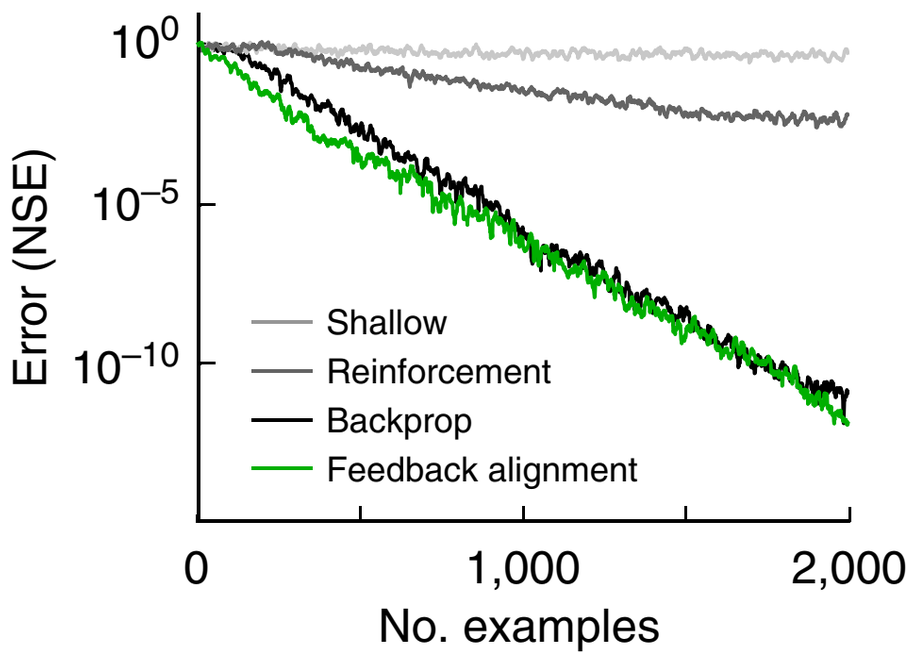
Feedback weights = Feedforward weights

$$v = W_v h$$

The backpropagation algorithm
and why it is at odds with biology
1. Weight transport problem

Solution: Unclear, but Lillicrap et al. found that exact feedback weights can be replaced by random weights, without a major impact on performance. This method is known as **feedback alignment**, and suggests that the brain may not need an exact feedback signal to implement 'backprop'.





B is a random matrix, replacing W^T

Lillicrap et al., Nature Comms (2016)

The backpropagation algorithm
and why it is at odds with biology
2. Derivative of activation function

Backprop: Relies on the derivative of the activation function.

Biology: Not clear how neurons could calculate their own derivative.

Solution: Unclear, but recent results suggest that backpropagating this derivative is not critical for performance (similar to feedback alignment).

$$\Delta w_u = -\eta \underbrace{(\mathrm{v} - \mathrm{target})}_{\text{error}} \; w_v^T f_h' u$$

Derivative of activation function

The backpropagation algorithm
and why it is at odds with biology
3. Two-phase learning

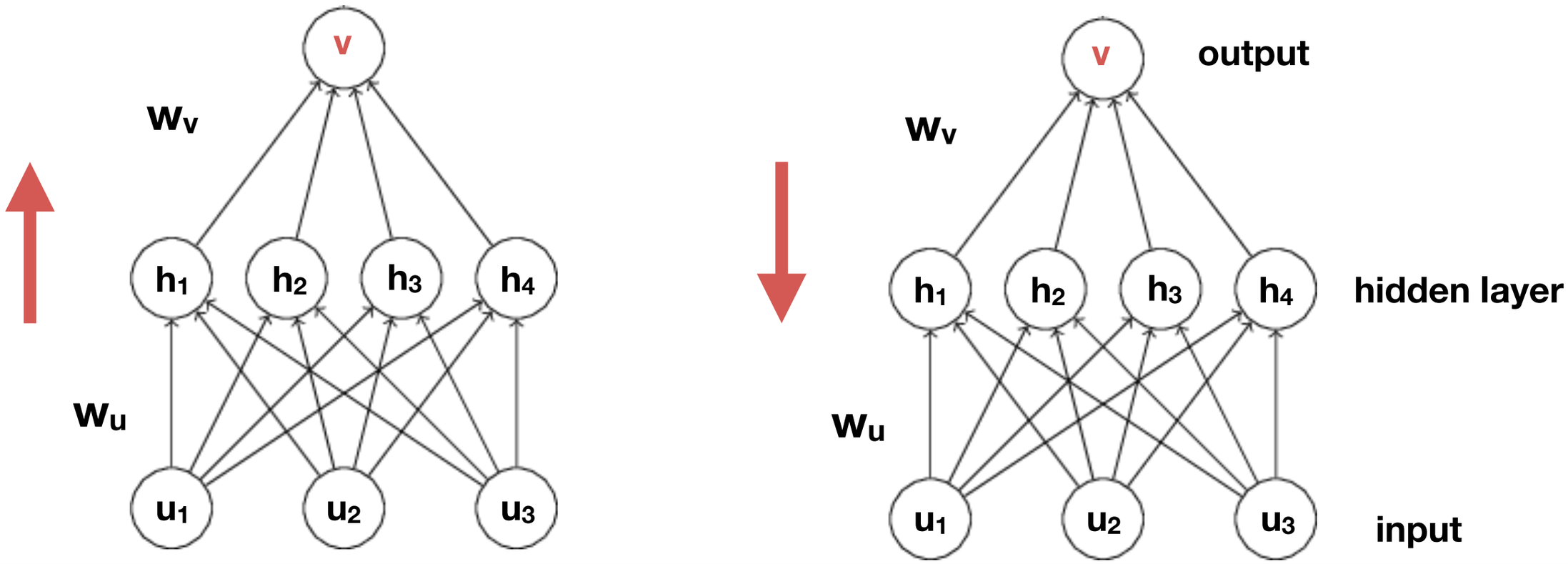
Backprop: The backprop relies on **two phases**, first a feedforward pass of the activity and then a backward pass of errors.

Biology: In the brain there is no clear separation between perception and learning.

Forward phase
(perception)

Backward phase (learning)

$$\text{error} = \frac{1}{2}(v - \text{target})^2$$



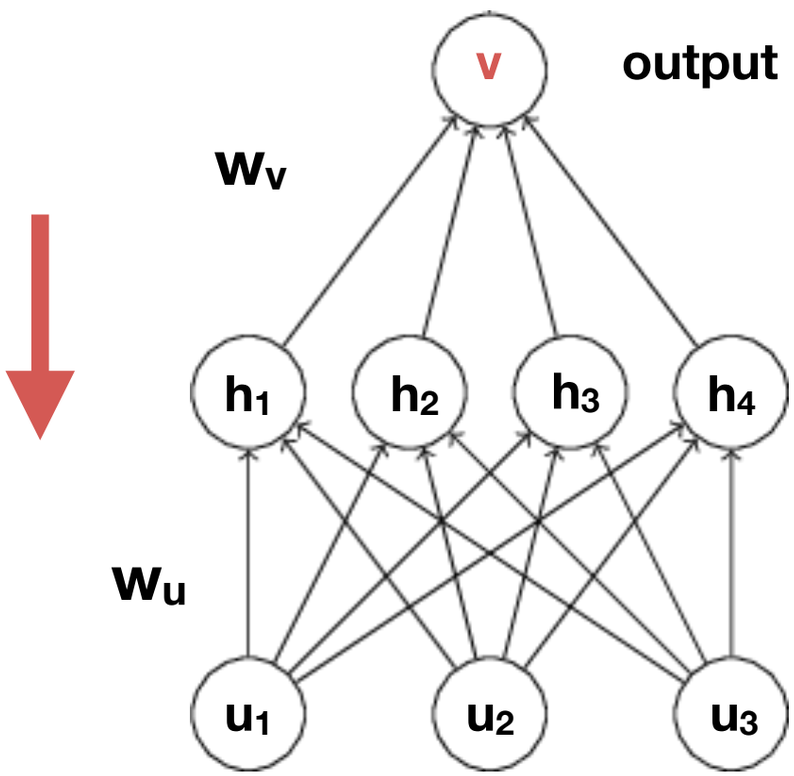
The backpropagation algorithm
and why it is at odds with biology
4. Separate error network

Backprop: The chain rule can be seen as a separate (error/gradients) network.

Biology: Minimal evidence for the existence of such gradient or error networks.

Backward phase ~ error network (EN)

error = $\frac{1}{2}(v - \text{target})^2$



EN weights

$$\Delta w_u = -\eta \underbrace{(\text{v} - \text{target})}_{\text{error}} w_v^T f_h' u$$

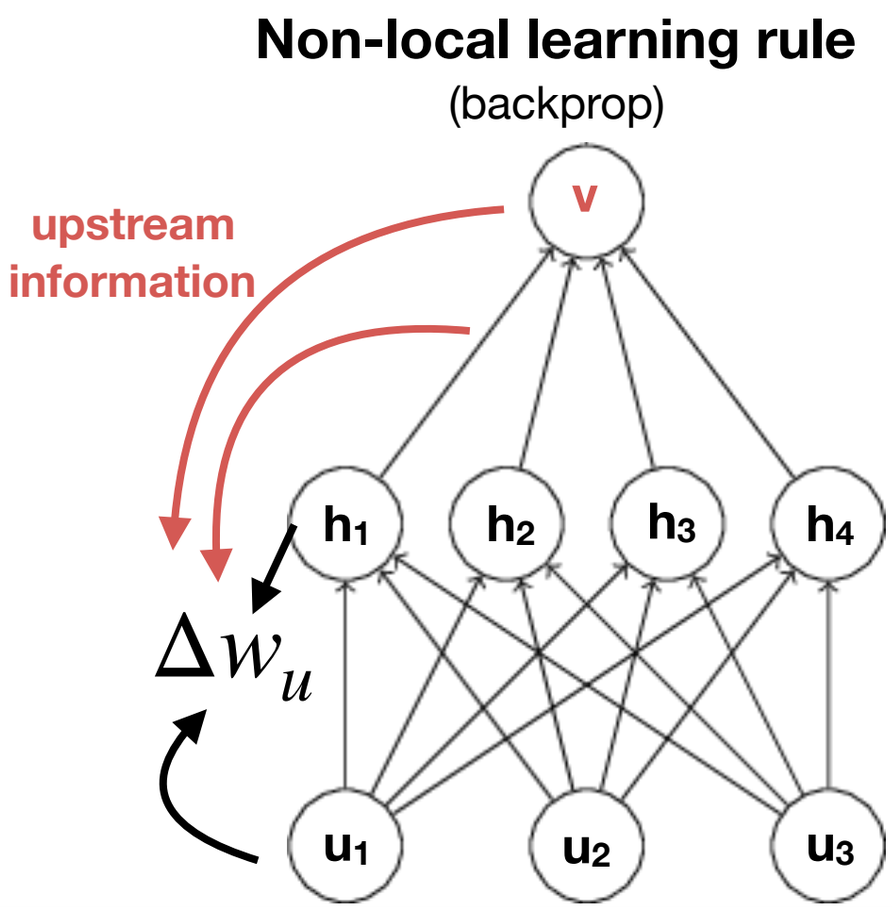
EN activation function

The backpropagation algorithm
and why it is at odds with biology
5. Non-local learning rules

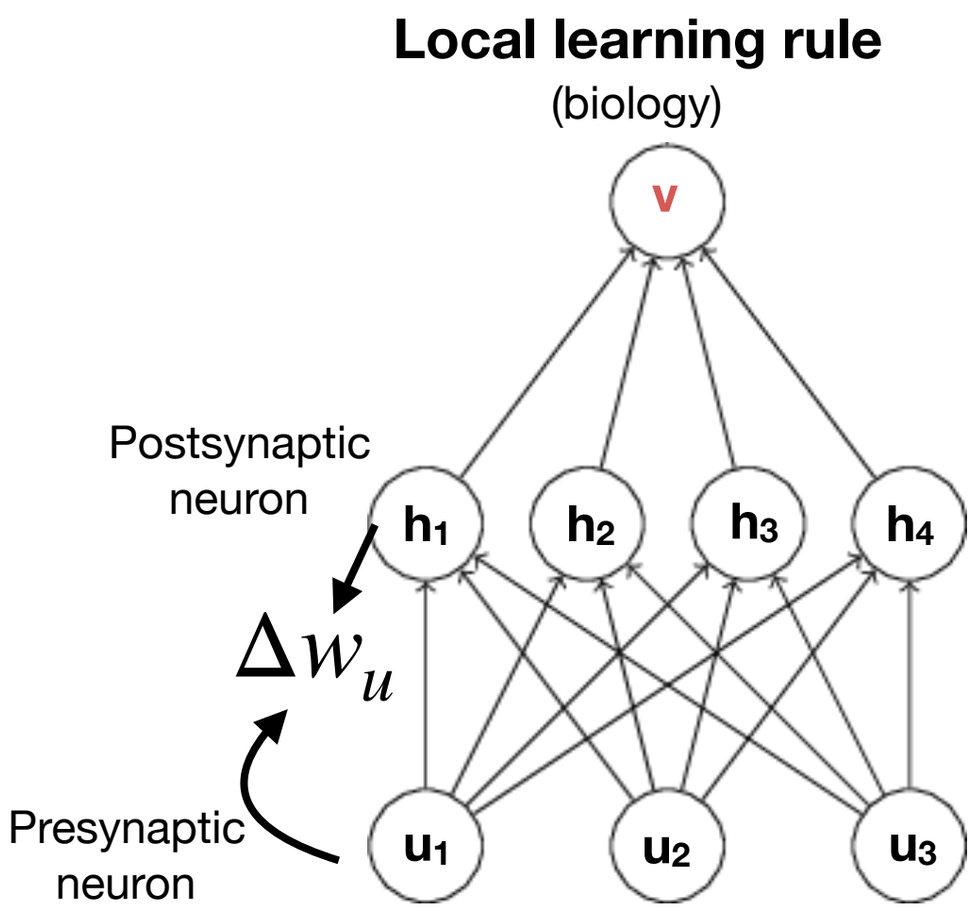
Backprop: Relies on non-local learning rules (i.e. a given neuron needs information about other neurons and layers).

Biology: Learning is believed to occur through local changes at synapses, a process known as synaptic plasticity which can only access locally available information (i.e. within the pre and postsynaptic neuron).

Non-local learning rule
(backprop)



Local learning rule
(biology)



Group discussion
groups of 2-3 (5min)

Which of the following points do you think is the **least biologically plausible**?

Key issues:

1. **Weight transport problem**: It proposes feedback weights equal to feedforward
2. **Derivative of activation functions**: Relies on calculating derivatives of the activation functions
3. **Two phase learning**: (1) feedforward propagation of activity and (2) error backpropagation
4. **Separate error network**: Suggest the need for a separate biological network computing the learning rules
5. **Non-local learning rules**: The weight update depends on non-local information
6. **Target**: The target label guides supervised learning

$$\Delta w_u = -\eta \underbrace{(\mathrm{v} - \mathrm{target})}_{\text{error}} \; w_v^T f_h' u$$

The backpropagation algorithm in the brain

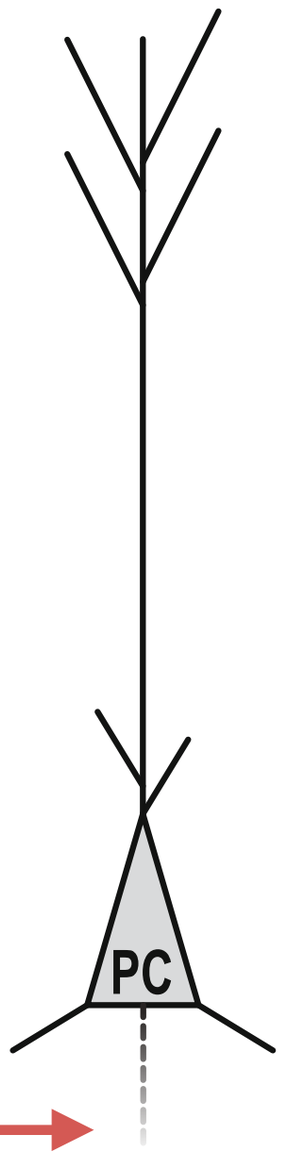
How to solve:

1. **Weight transport problem**: It proposes feedback weights equal to feedforward
2. **Derivative of activation functions**: Relies on calculating derivatives of the activation functions
3. **Two phase learning**: (1) feedforward propagation of activity and (2) error backpropagation
4. **Separate error network**: Suggest the need for a separate biological network computing the learning rules
5. **Non-local learning rules**: The weight update depends on non-local information
6. **Target**: The target label guides supervised learning

The backpropagation algorithm in the brain

Inspiration from real neurons: pyramidal cells

Pyramidal cell:
(one of the most common neurons in the brain)



Distal dendrites
(receive feedback input)

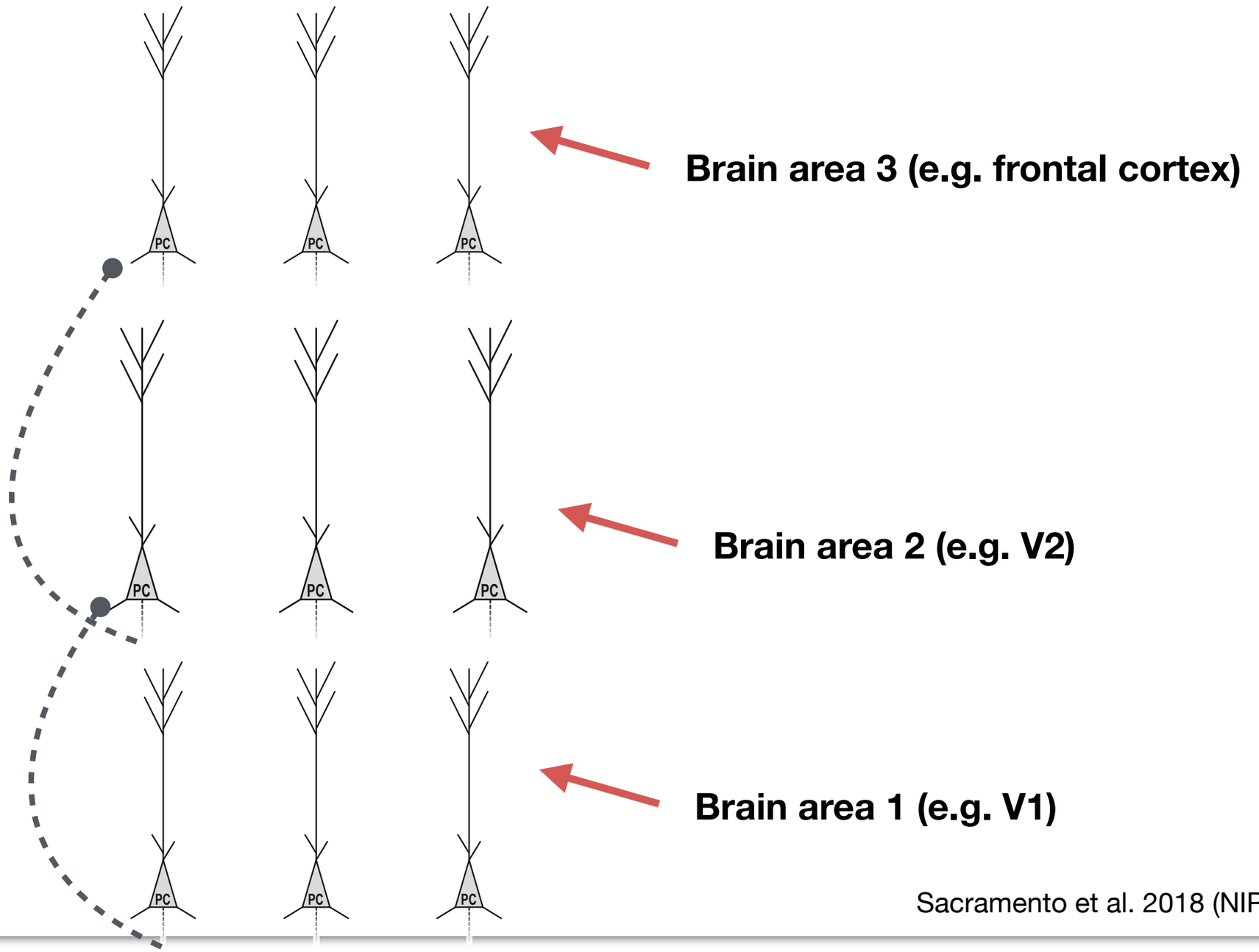
Proximal dendrites
(receive feedforward input)

Axon
(send output of this cell to others)

Sacramento et al. 2018 (NIPS)

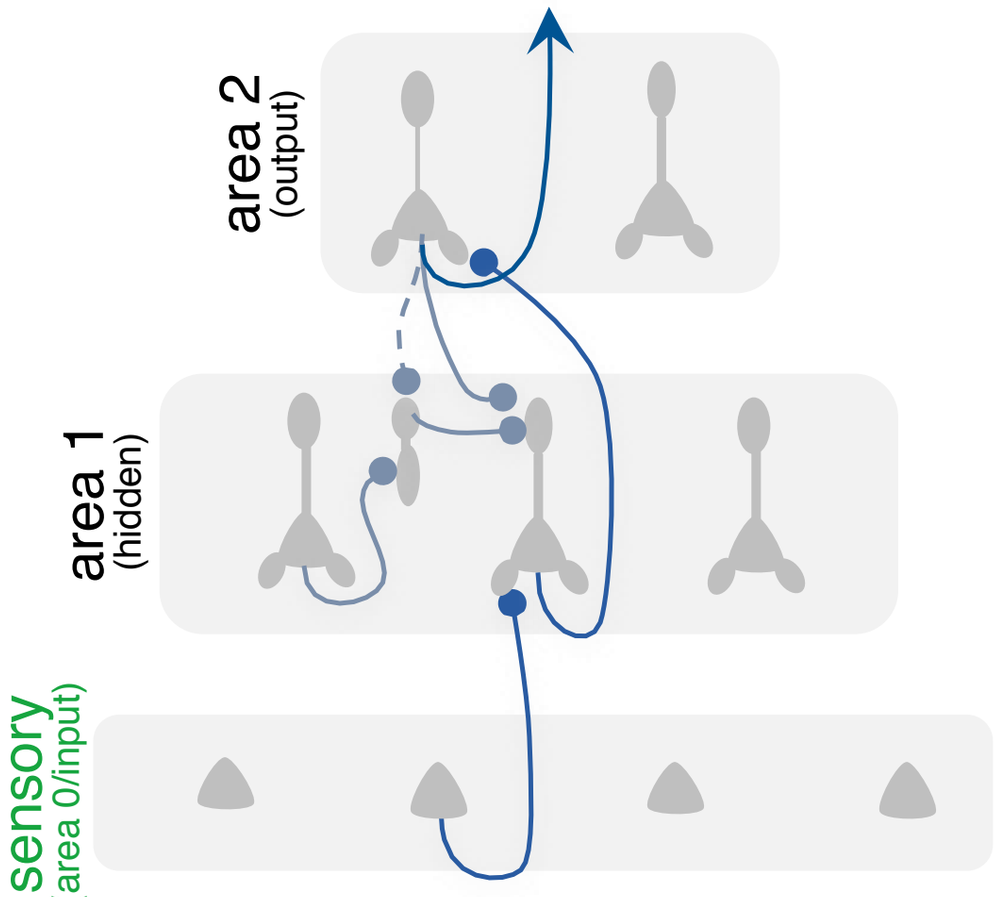
The backpropagation algorithm in the brain

Inspiration from real neurons: pyramidal cells



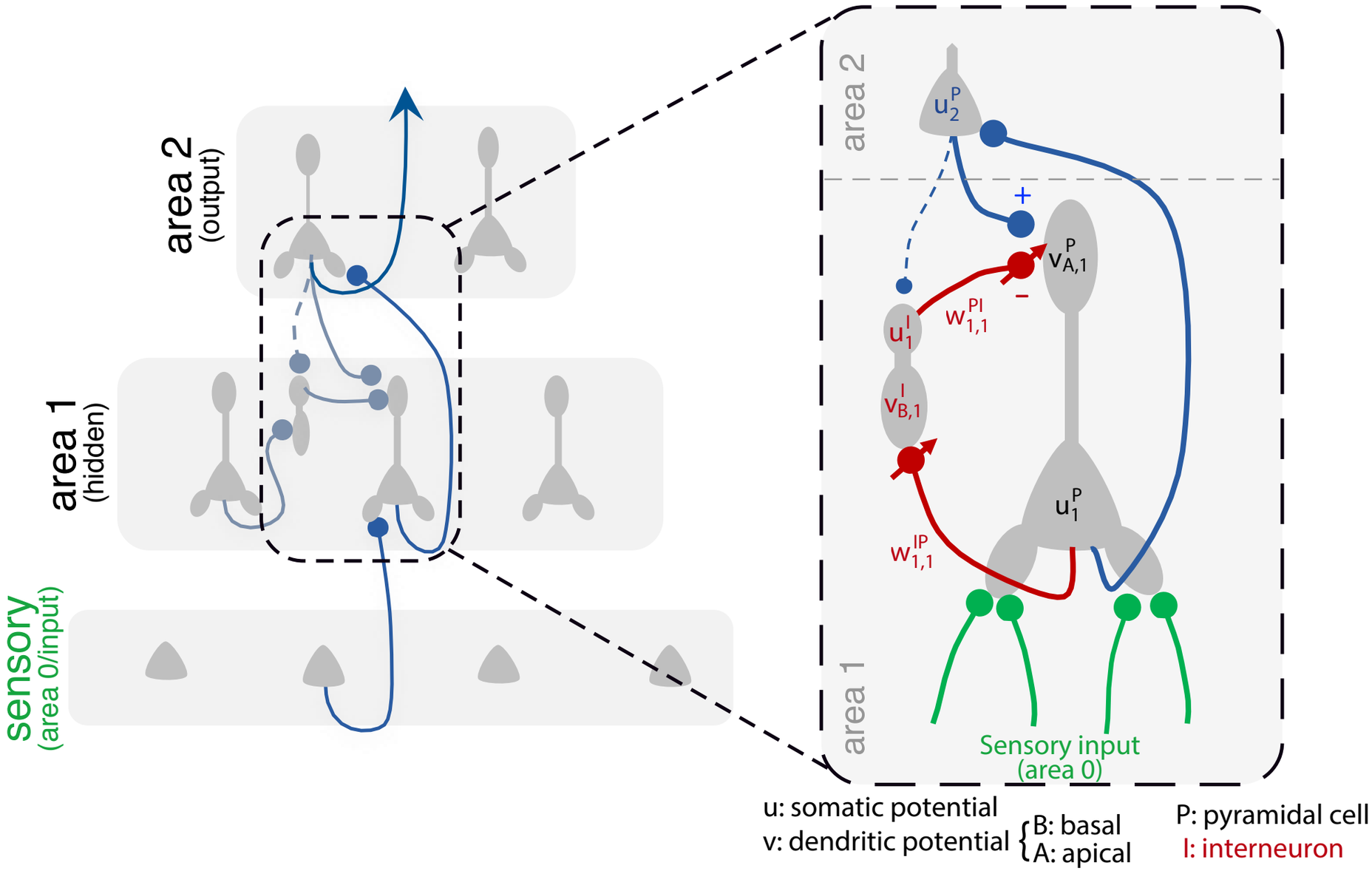
Sacramento et al. 2018 (NIPS)

Dendritic microcircuits approximate backprop



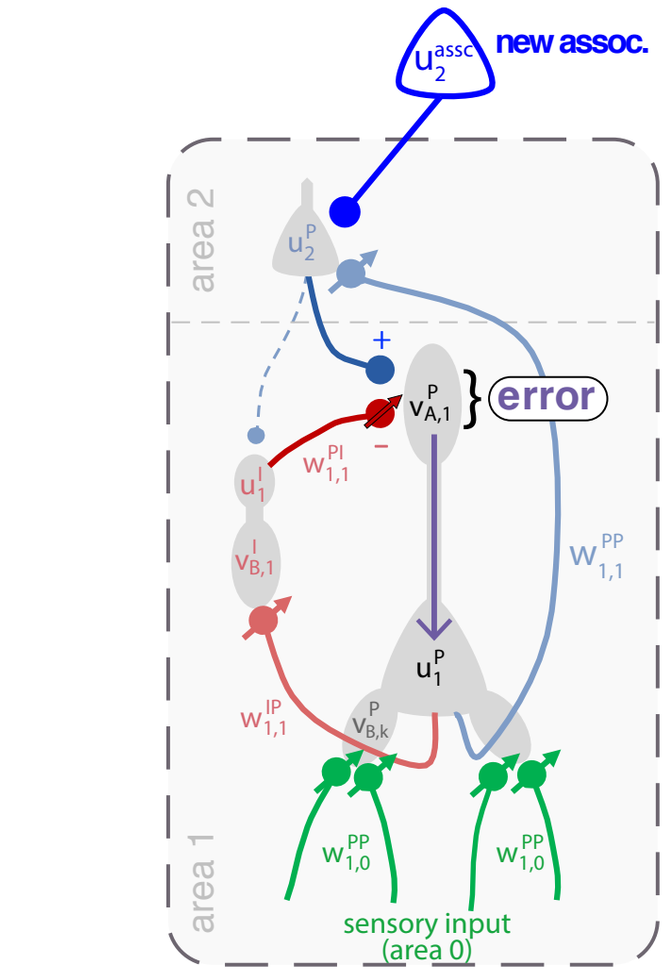
Sacramento et al. 2018 (NIPS)

Dendritic error microcircuits approximate backprop



Sacramento et al. 2018 (NIPS)

Dendritic error microcircuits approximate backprop

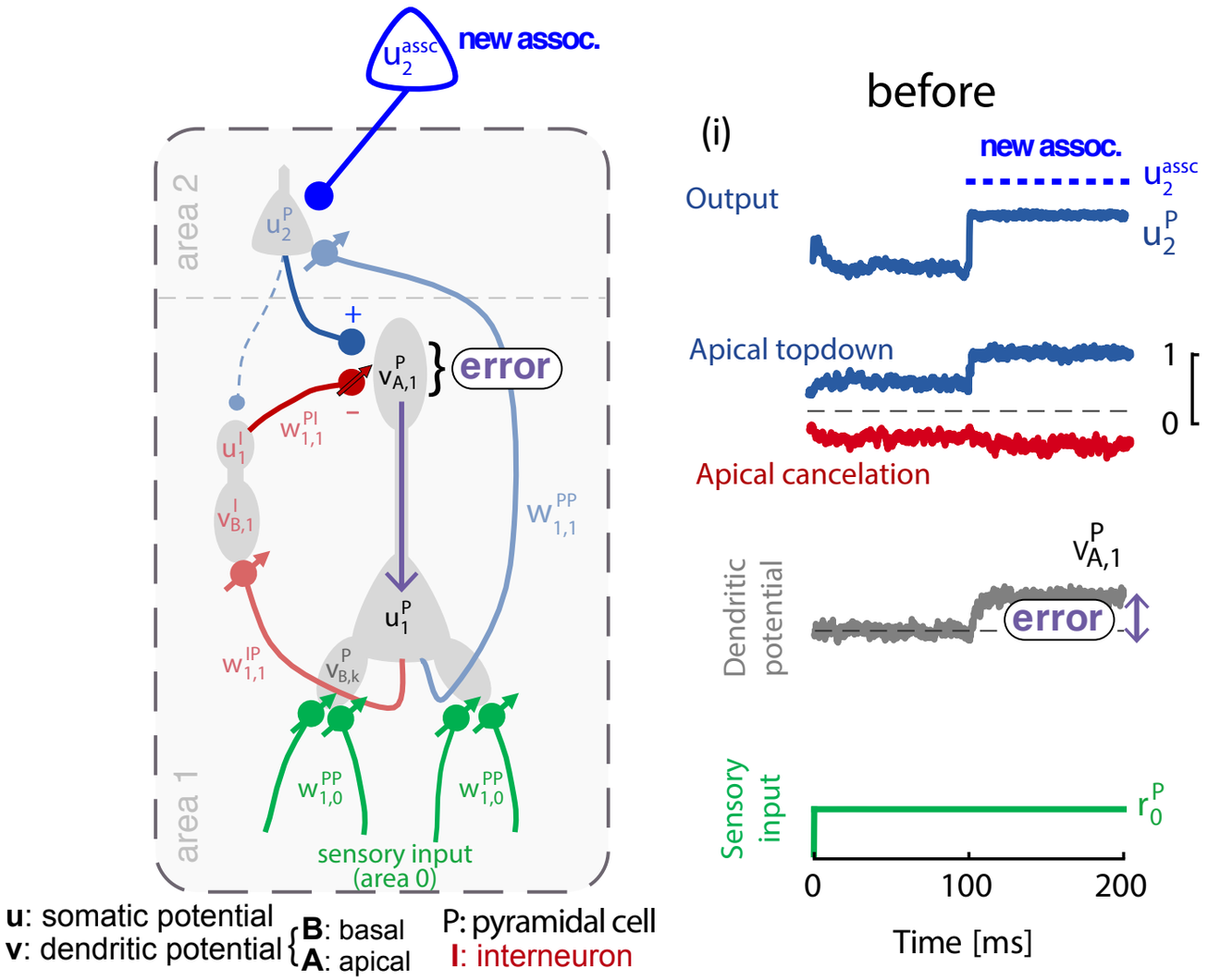


u: somatic potential
v: dendritic potential { **B**: basal, **A**: apical

P: pyramidal cell
I: interneuron

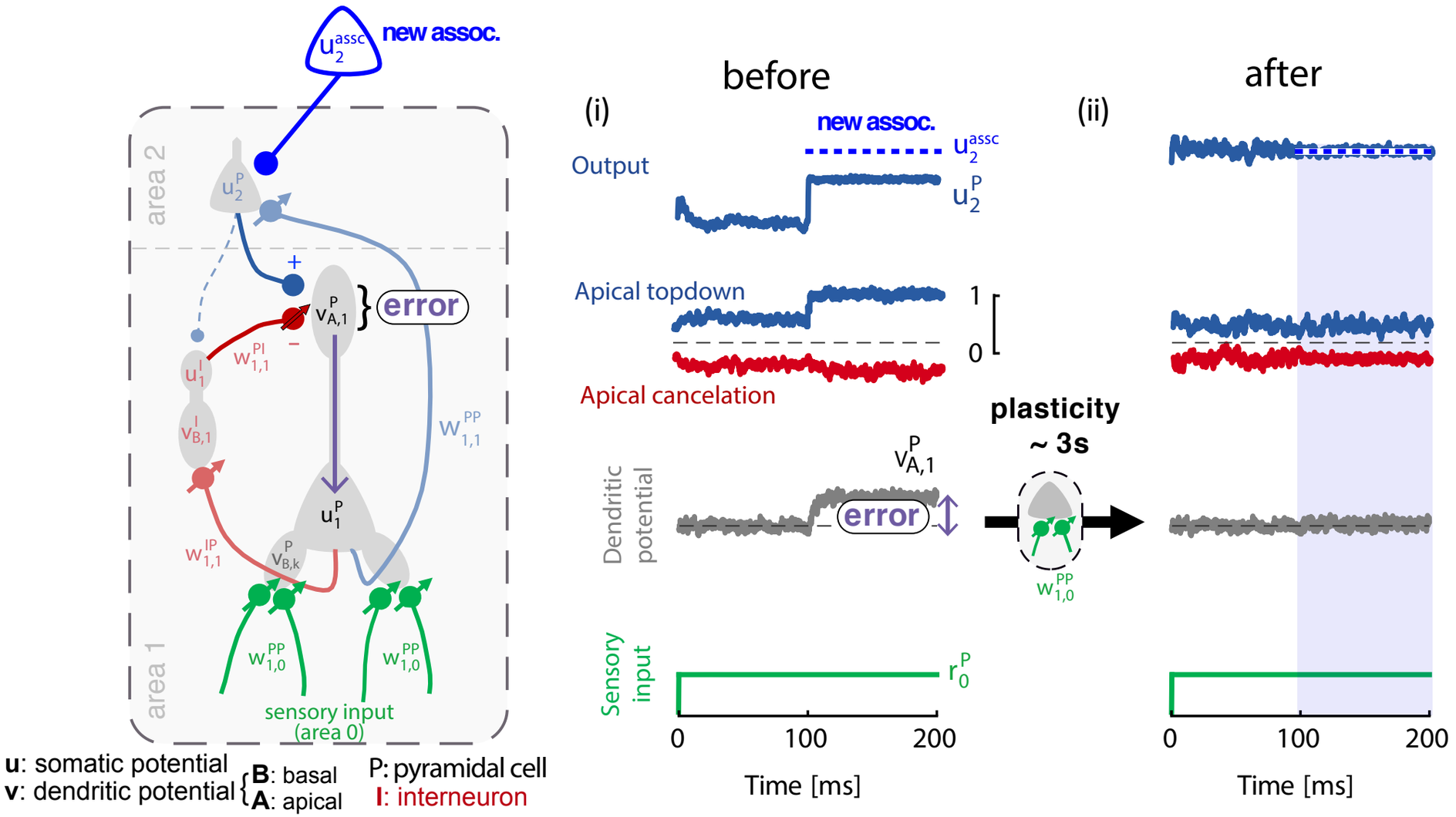
Sacramento et al. 2018 (NIPS)

Dendritic error microcircuits approximate backprop



Sacramento et al. 2018 (NIPS)

Dendritic error microcircuits approximate backprop

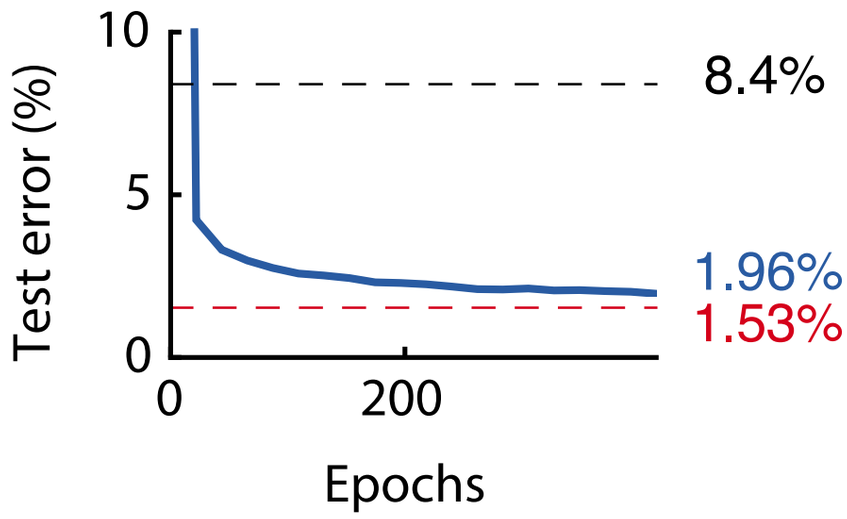


u: somatic potential
v: dendritic potential
{**B**: basal, **A**: apical}
P: pyramidal cell
I: interneuron

Sacramento et al. 2018 (NIPS)

Dendritic error microcircuits learn to classify hand-written digits

MNIST Dataset



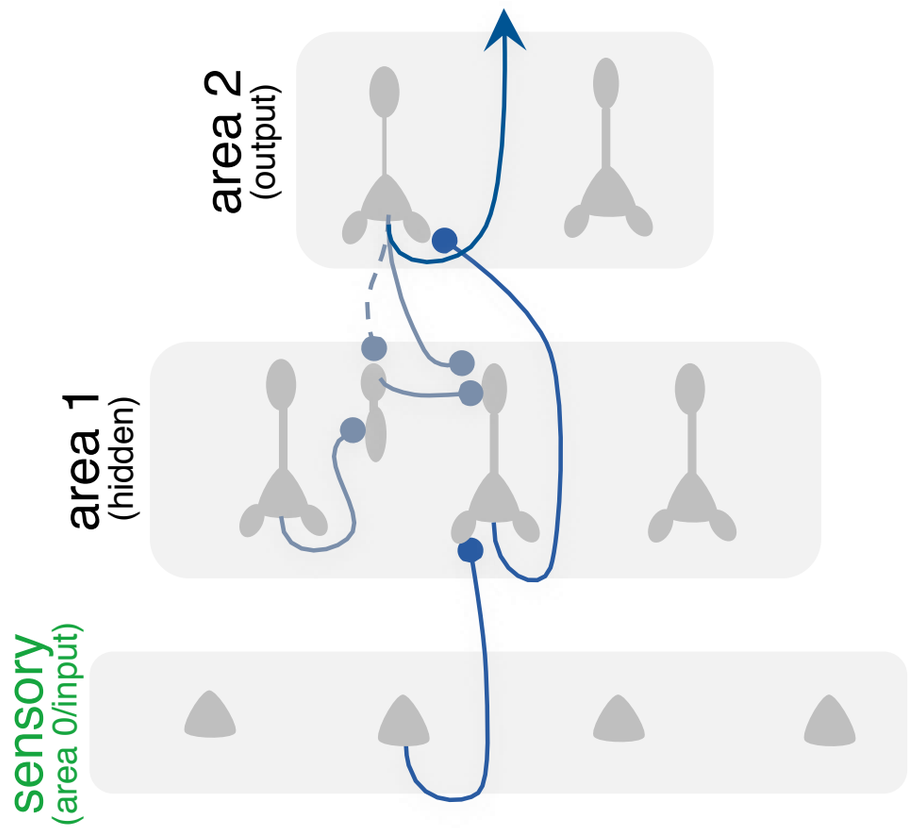
single-layer

500+500+10 microcircuit

500+300+10 backprop

Sacramento et al. 2018 (NIPS)

Dendritic error microcircuits approximate backprop



By being closer to biology neural networks ‘solve’ three key problems:

3. **Two phase learning**: No need to separate into two phases
4. **Separate error network**: Distal dendrites encode errors
5. **Non-local learning rules**: Learning rules are local

see also Guerguiev et al. eLife 2017

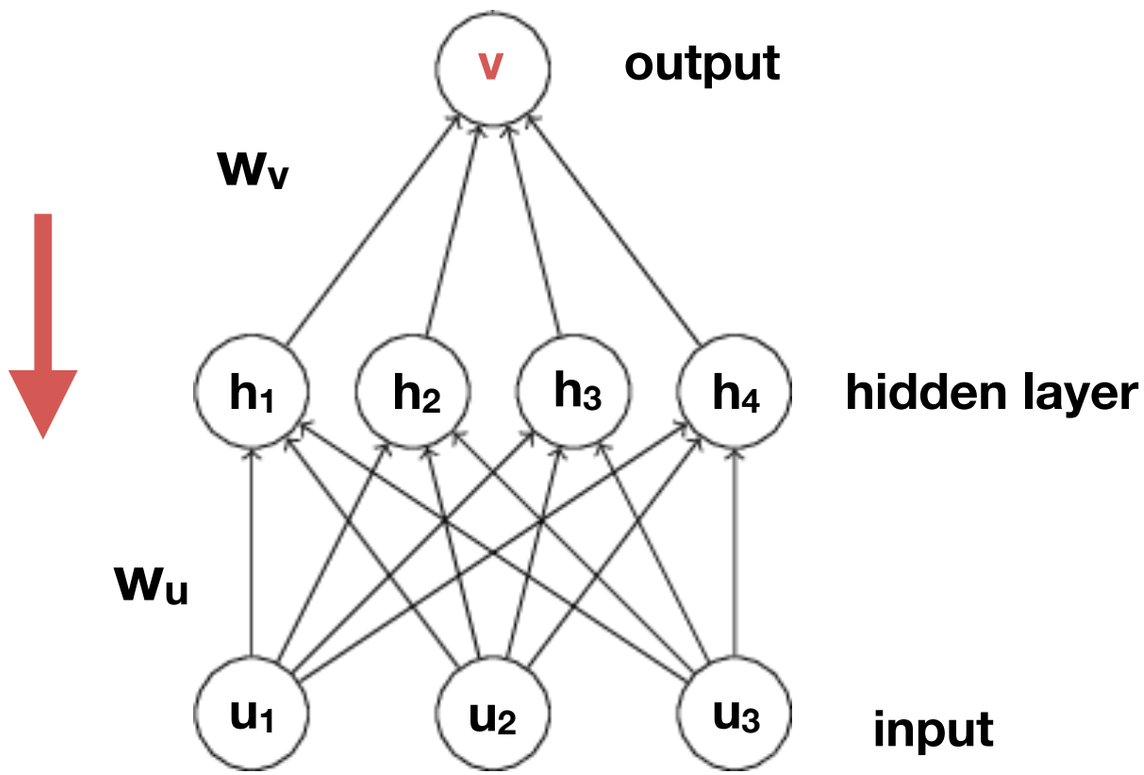
Sacramento et al. 2018 (NIPS)

The backpropagation algorithm
and why it is at odds with biology
6. Target

Backprop: When doing supervised learning it needs a specific target (teaching signal).

Biology: It is far from clear how such target signals could be generated.

$$\textbf{error} = \frac{1}{2}(v - \text{target})^2$$



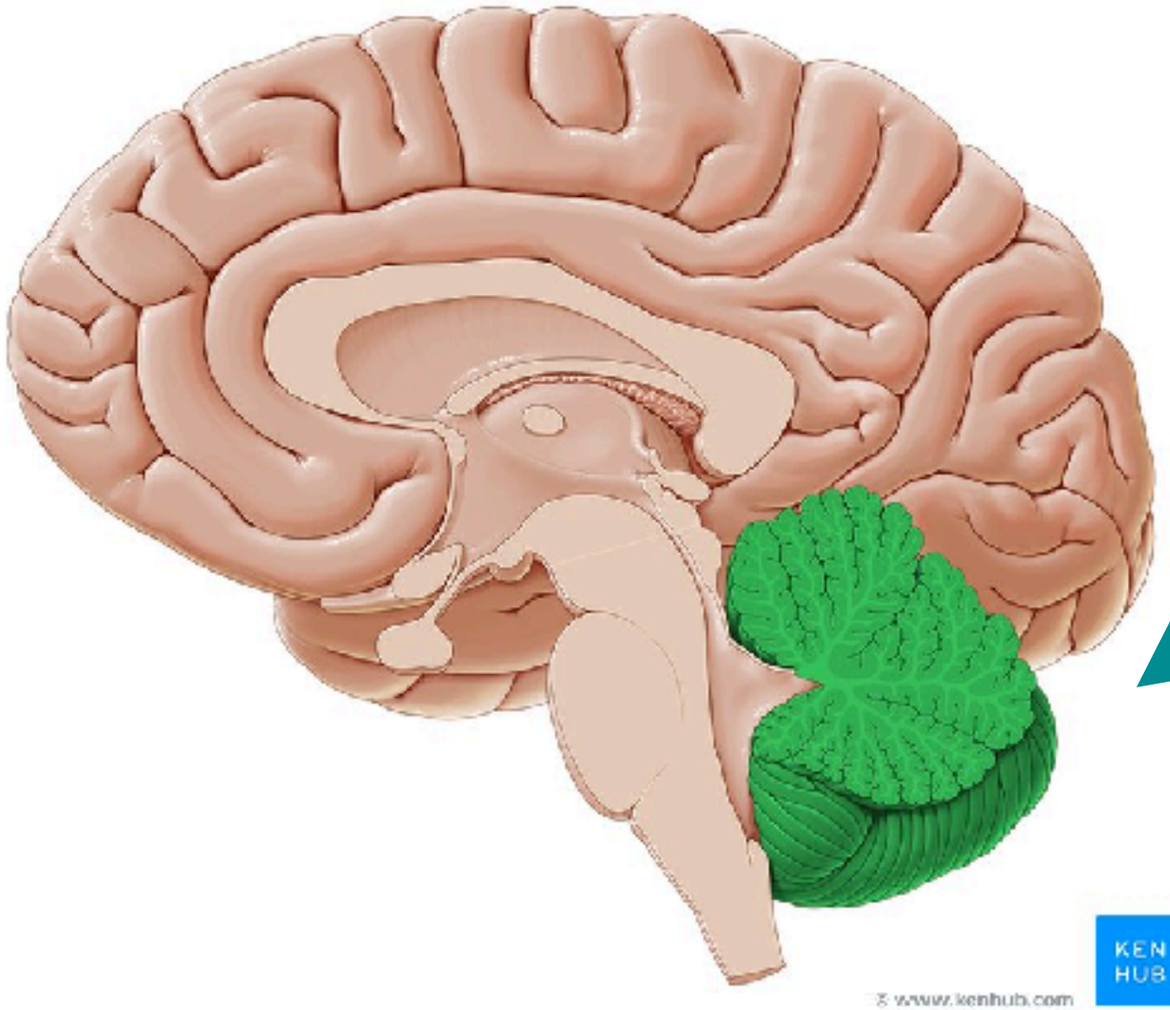
The backpropagation algorithm
and why it is at odds with biology
6. Target

Solution 1: Backprop also works in the context of unsupervised (e.g. autoencoders) and reinforcement learning (deep Q-learning). We are going to cover some of this in the coming lectures.

Solution 2: Specific brain areas (e.g. cerebellum) may be calculating teaching signals, which are then used for internal supervised learning at other brain areas (see end of this lecture).

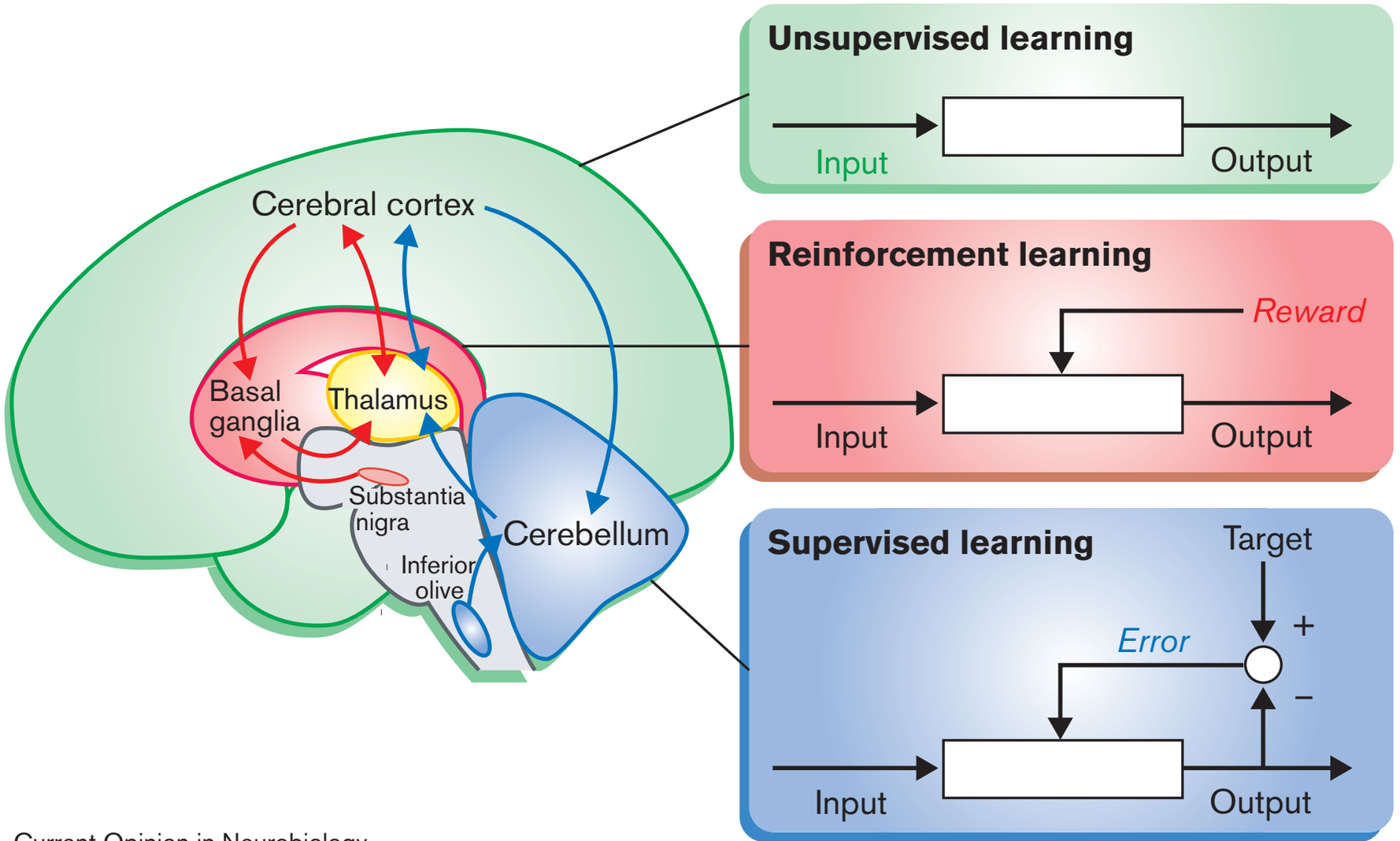
Cerebellum, the brain supervisor?

The **cerebellum** (small brain in Latin) has a stereotypical structure, and the most numerous neuron in the brain (Purkinje cells). It is classically involved in motor error correction, but growing evidence suggests to be also involved in regulating many other aspects of behaviour.



Cerebellum, the brain supervisor?

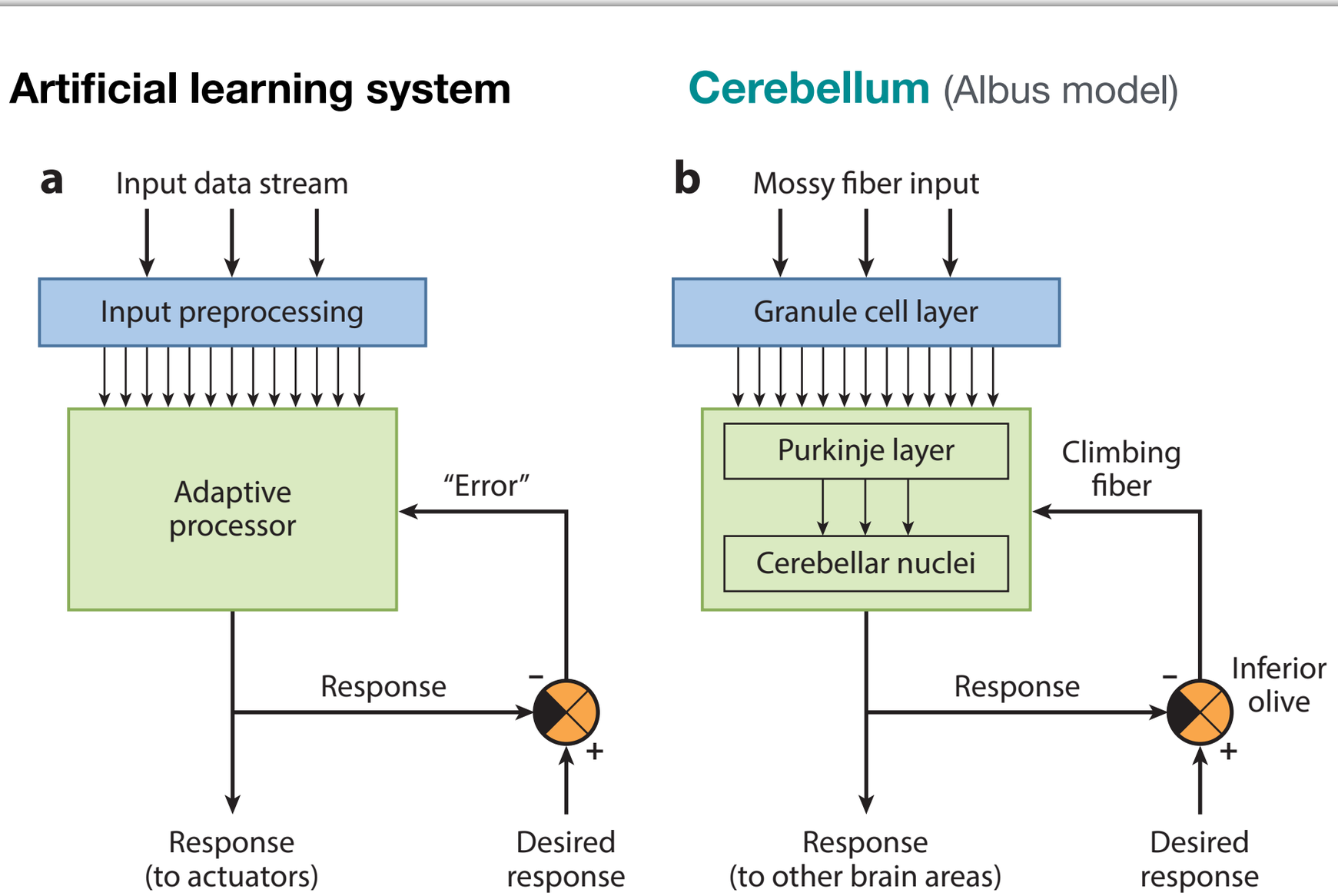
The **cerebellum** itself seems to use a form of supervised learning, but may also provide teaching signals to the cerebral cortex.



Current Opinion in Neurobiology

Doya, Curr Opin Neurobiol (2000)

Cerebellum may rely on supervised learning



Raymond and Medina, Annual Rev Neurosci (2018)

Summary

1. **Deep learning successes relies on the backprop algorithm**

2. **Backprop is slowly becoming more accepted as a model of learning in the brain, but several issues remain unsolved**

4. **Dendritic microcircuits provide a powerful model of learning in the brain that approximates backprop**

6. **Cerebellum is involved in supervised learning**

Note: Practice writing down the gradient wrt a given weight in a simple network!

References

Text books:

Parallel Distributed Processing, Rumelhart et al. 1986 (classical book on neural networks)

Deep Learning by Courville et al. 2015

Relevant papers:

- Roelfsema and Holtmaat, Nature Neuroscience Rev (2018) (recent review on the credit assignment problem in the brain)
- Doya, Curr Opin Neurobiol (2000) (review on how different brain areas learn)
- Richards and Lillicrap, Current Opinion in Neurobiology (2019) (recent review on dendritic credit assignment)
- Sacramento et al., NIPS (2018) (shows how dendritic microcircuits may approximate backprop)

Upcoming lectures

- Neural circuits and learning (L10)
- **Neural circuits: sensory processing**
  - L11: Visual cortex
  - L11: Convolutional neural networks
  - L12: **Supervised learning:** The backpropagation algorithm/cerebellum
  - **L13: Unsupervised learning and Boltzmann Machines**
- Temporal processing in the brain
  - L14: Auditory cortex and recurrent neural networks
  - L15: Gated recurrent neural networks
- Reinforcement learning in the brain
  - L16: Temporal difference learning/dopamine
  - L16: Q-learning and deep RL