

Supervised learning in deep neural networks

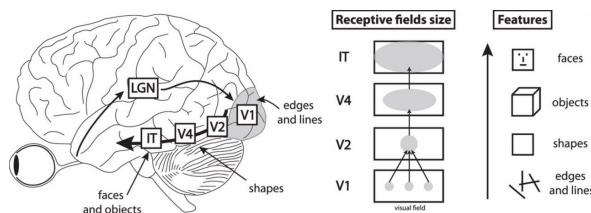
Seán Froudist-Walsh

Lecturer in Computational Neuroscience

1

1

How can we understand a neural system?



- A neuron in visual cortex responds to inputs in a particular part of space (it's receptive field), because that its connections can be traced back to the precisely that position on the retina of the eye.
- It is therefore the patterns of connectivity that determine the what each brain cell represents.
- To understand a neural system we must ask: what is the principle by which the connections are learned?

Summerfield, 2018; Manassi et al., *J. Vision*, 2013

2

1

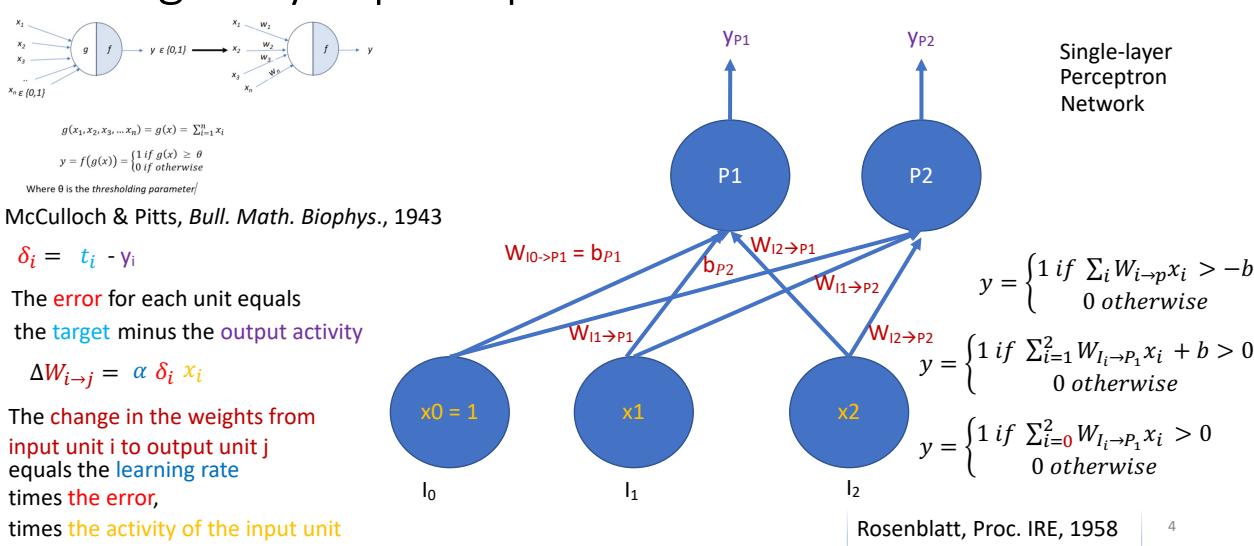
Intended Learning Outcomes

- By the end of this video you will be able to:
 - describe historical and modern approaches to supervised learning
 - update weights in a neural network using the backpropagation of errors algorithm
 - critically assess the success and failures of deep neural networks trained with supervised learning as models of human vision
 - describe two biologically-inspired variants on the backpropagation algorithm

3

3

Learning in a simple neural network single-layer perceptron



4

Supervised learning

$$\delta_i = t_i - y_i$$

The **error** for each unit equals
the **target** minus the **output activity**

Some all-knowing teacher informs the network of the correct response

5

5

The loss function – how wrong am I
for this training example?

The **cost** function – how wrong am I
on average over all training examples?

$$\frac{1}{m} \sum_{i=1}^m (t_i - y_i)^2 = J$$

We usually want to minimise the average loss over all training examples (=cost)

6

6

Gradient descent – change the weights and biases to become less wrong

The gradient tells you how the cost changes with little changes to all of the weights and biases. The magnitude of each element in the gradient tells you how sensitive the cost function is to change in each weight and bias.

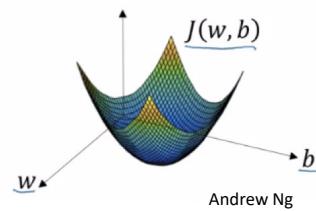
$$w_{\text{new}}^{(L)} = w_{\text{old}}^{(L)} - \alpha \frac{\delta J}{\delta w_{\text{old}}^{(L)}}$$

With a little change in the weight how much does the cost go up or down?

We change the weight in proportion to the learning rate so that we move down (towards lower cost)

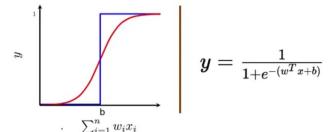
$$\nabla J = \begin{pmatrix} \frac{\delta J}{\delta w^{(1)}} \\ \frac{\delta J}{\delta b^{(1)}} \\ \vdots \\ \frac{\delta J}{\delta w^{(L)}} \\ \frac{\delta J}{\delta b^{(L)}} \end{pmatrix}$$

We want to find w, b (weights, biases) that minimize J – the cost



Andrew Ng

Learning by gradient descent is only possible when a small change in the weights leads to a small change in the output value.



This is not possible when the output changes abruptly between 1 and 0, as in the original version of the perceptron

It is possible with activation functions like the sigmoid and ReLU ⁷

7

Quiz - blackboard

- Gradient descent is not possible for the original perceptron model because the _____ is not _____

8

8

Backpropagation – how to figure out the gradient

$w^{(L)}$ $a^{(L-1)}$ $b^{(L)}$

y

$J_0 = (a^L - y)^2$

$a^{(L)} = f(w^{(L)} a^{(L-1)} + b^{(L)}) = f(z^{(L)})$

f = activation fn (like ReLU or sigmoid)

Our goal is to find how sensitive the loss function is to changes in each of the weights and biases (parameters)
Which changes to these parameters will cause the biggest decrease in the loss?

$$\frac{\delta J}{\delta w^{(L)}} = \frac{\delta z^{(L)}}{\delta w^{(L)}} \frac{\delta a^{(L)}}{\delta z^{(L)}} \frac{\delta J}{\delta a^{(L)}} \quad \text{Chain rule}$$

Goal 1: what is the change to the loss in response to small changes to the weight?

9

9

Backpropagation – example

$\frac{\delta J_0}{\delta a^{(L-1)}}$

$w^{(L)}$ $a^{(L-1)}$ $b^{(L)}$

y

$\nabla J = \begin{pmatrix} \frac{\delta J}{\delta w^{(1)}} \\ \frac{\delta J}{\delta b^{(1)}} \\ \vdots \\ \frac{\delta J}{\delta w^{(L)}} \\ \frac{\delta J}{\delta b^{(L)}} \end{pmatrix}$

$\frac{\delta J_0}{\delta w^{(L)}} = \frac{\delta z^{(L)}}{\delta w^{(L)}} \frac{\delta a^{(L)}}{\delta z^{(L)}} \frac{\delta J_0}{\delta a^{(L)}}$

$\frac{\delta J_0}{\delta a^{(L)}} = 2(a^L - y)$ partial derivatives

$\frac{\delta a^{(L)}}{\delta z^{(L)}} = f'(z^{(L)})$ things to differentiate

$\frac{\delta z^{(L)}}{\delta w^{(L)}} = a^{(L-1)}$

f = activation fn (like ReLU or sigmoid)

$a^{(L)} = f(z^{(L)})$

$z^{(L)} = w^{(L)} a^{(L-1)} + b^{(L)}$

$\frac{\delta J_0}{\delta w^{(L)}} = a^{(L-1)} f'(z^{(L)}) 2(a^L - y)$

$\frac{\delta J_0}{\delta b^{(L)}} = 1 f'(z^{(L)}) 2(a^L - y)$

$\frac{\delta J_0}{\delta a^{(L-1)}} = w^{(L)} f'(z^{(L)}) 2(a^L - y)$

How much a change to the weight influences the loss depends on the activity of the previous layer

10

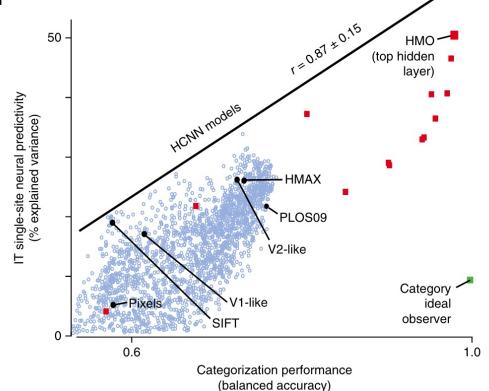
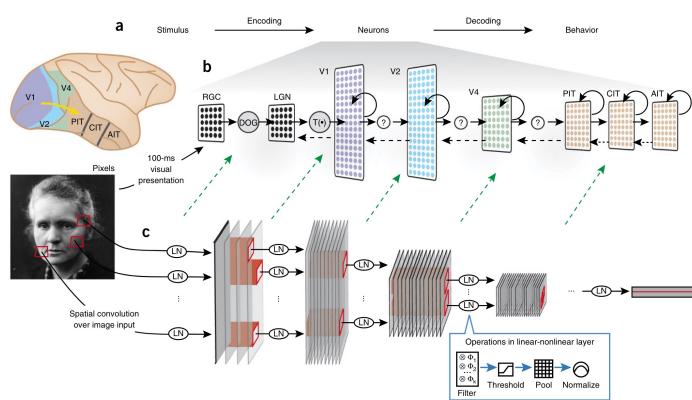
Quiz - blackboard

- The objective of backpropagation is to find the _____ that minimize the error between the predicted and target outputs. This is done using the _____ rule.

11

11

Deep neural networks trained with backprop
that perform better on object recognition tasks
also better predict cortical spiking data



Yamins & DiCarlo, *Nat. Neurosci.*, 2016

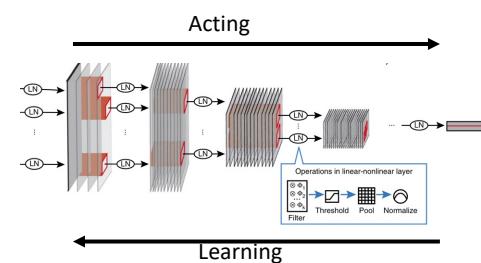
12

Criticism: the backpropagation of error algorithm (backprop) is not biologically realistic because:

1. The weights going forwards equal the weights going back
2. The weight update depends on information from distant neurons
3. The network acts (forward-propagates activity) and learns (back-propagates errors) in two separate phases

$$\frac{\delta J_0}{\delta a^{(L-1)}} = w^{(L)} f'(z^{(L)}) 2(a^{(L)} - y)$$

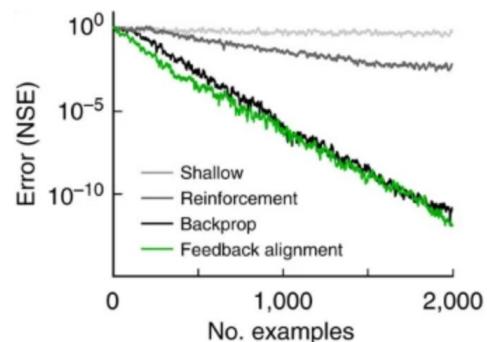
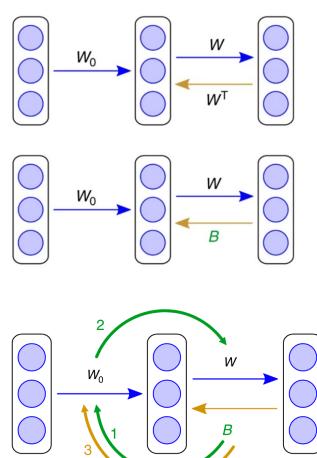
$$\frac{\delta J_0}{\delta w^{(L-1)}} = \frac{\delta z^{(L-1)}}{\delta w^{(L-1)}} \frac{\delta a^{(L-1)}}{\delta z^{(L-1)}} \frac{\delta J_0}{\delta a^{(L-1)}}$$



13

13

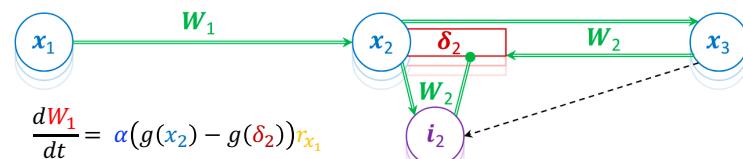
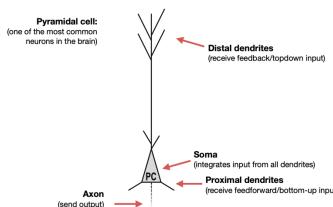
Ways the brain could do something like backprop – 1 – feedback alignment

Lillicrap et al., *Nat. Comms.*, 2016

14

14

Ways the brain could do something like backprop – 2 – dendritic errors



When this network converges to the equilibrium, the neurons encode their corresponding **error terms** in their **dendrites**.

A single neuron is used simultaneously for activity propagation (at the cell body), error encoding (at dendrites) and error propagation to the cell-body without the need for separate phases.

Sacramento et al., *NeurIPS*, 2018
Whittington & Bogacz, *TiCS*, 2019

The weights are updated according to a **learning rate**, **firing rate input from the lower area**, and the difference in voltage between the **dendrite** and **cell body**



Rui Ponte Costa, Maija Filipovica, Ellen Boven, Joe Pemberton, Dabal Pedamonti, Will Greedy, Kevin Nejad & others

15

Quiz- blackboard

- The dendritic error model provides a solution to the problem of non-local learning in backpropagation because the plasticity rule depends three types of activity in the same _____ .

16

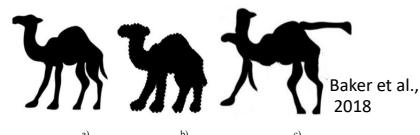
16

Deep Neural Networks do not solve image recognition tasks the way humans do

- DNNs do the best job in predicting brain signals in response to images taken from various brain datasets
- However, these behavioral and brain datasets do not test hypotheses regarding what features are contributing to good predictions
- “Deep Neural Networks account for almost no results from psychological research.”



Geirhos et al., ICLR, 2019



Baker et al., 2018

Bowers et al., Behavioral and Brain Sciences, 2022

17

Recap

- Supervised learning – learning from feedback, what exactly the response should have been, from a “teacher”
- Single layer perceptron networks were the first learning neural networks
- The gradient is how sensitive the cost is to changes to individual weights and biases (the direction and rate of fastest increase of the cost function)
- Gradient descent – a method to find the local minimum of the cost function – only works if the activation function can be differentiated (doesn’t work for the step function in the first perceptron model)
- Backpropagation of error algorithm (backprop) – use the Chain Rule of calculus to calculate the gradient with respect to all the weights and biases in the network, and use this to update the weights
- Deep neural networks trained with backprop that perform better on object recognition tasks also better predict cortical spiking data
- Backprop is usually considered not biologically realistic for several reasons
- Biologically-inspired variants of backprop have been proposed and quite successful
- Deep Neural Networks do not solve image recognition tasks the way humans do

18

18

Still curious? You can dive in deeper to any of today's topics:

- An early criticism of the biological realism of backpropagation of errors in the brain by Nobel prizewinner Francis Crick
 - Crick, Francis. "The recent excitement about neural networks." *Nature* 337, no. 6203 (1989): 129-132.
- Comparing deep ConvNets to brains
 - Yamins, Daniel LK, and James J. DiCarlo. "Using goal-driven deep learning models to understand sensory cortex." *Nature neuroscience* 19, no. 3 (2016): 356-365.
- Feedback alignment
 - Lillicrap, Timothy P., Daniel Cownden, Douglas B. Tweed, and Colin J. Akerman. "Random synaptic feedback weights support error backpropagation for deep learning." *Nature communications* 7, no. 1 (2016): 13276.
- Dendritic error model
 - Sacramento, João, Rui Ponte Costa, Yoshua Bengio, and Walter Senn. "Dendritic cortical microcircuits approximate the backpropagation algorithm." *Advances in neural information processing systems* 31 (2018).
- Problems with neural network models of human vision
 - Bowers, Jeffrey S., Gaurav Malhotra, Marin Dujmović, Milton Llera Montero, Christian Tsvetkov, Valerio Biscione, Guillermo Puebla et al. "Deep problems with neural network models of human vision." *Behavioral and Brain Sciences* (2022): 1-74.

19

19

10