

# Deep Learning Demystified



# TL;DR

Deep learning isn't magic.

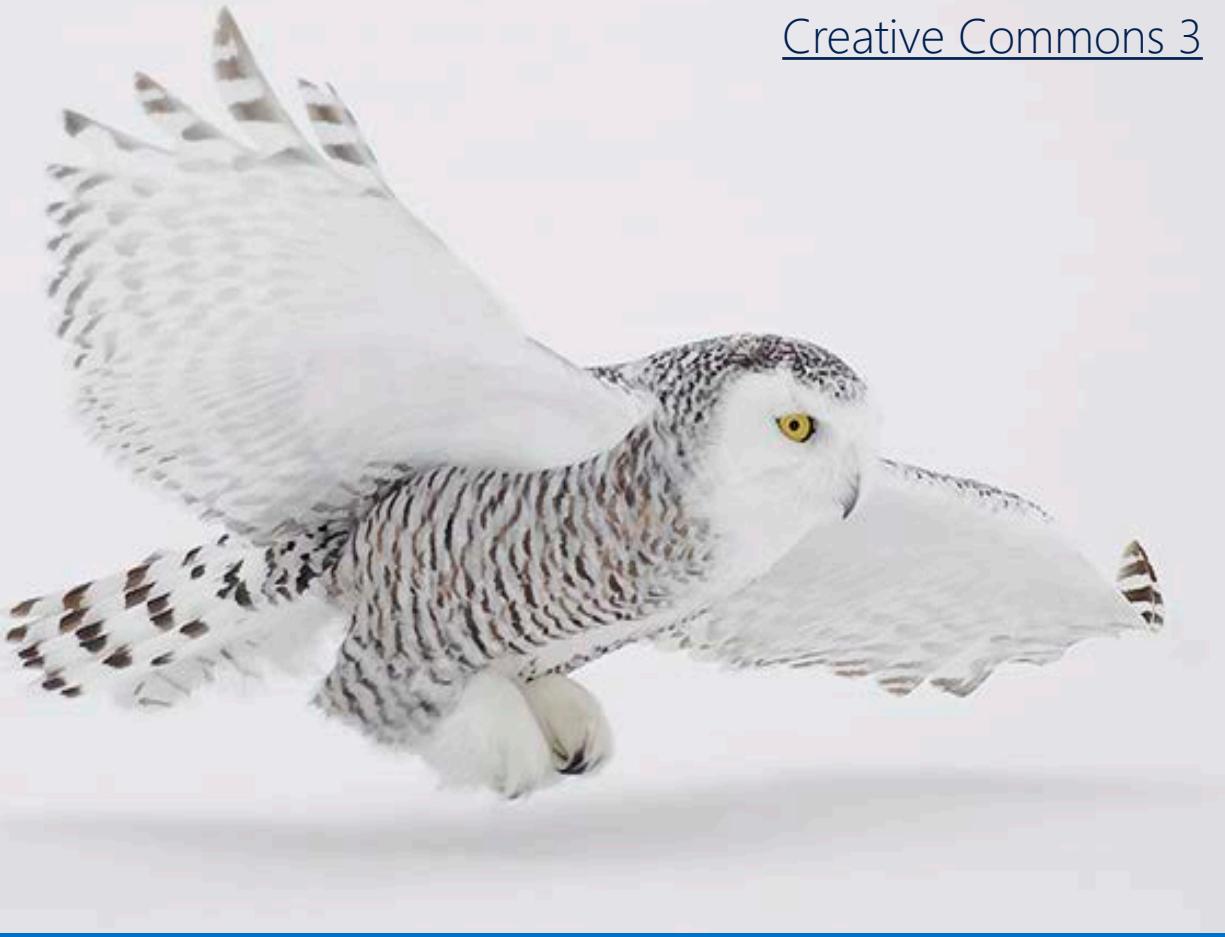
But it is very good at finding patterns.

# The brain and deep learning

Creative Commons 3



# The brain and deep learning



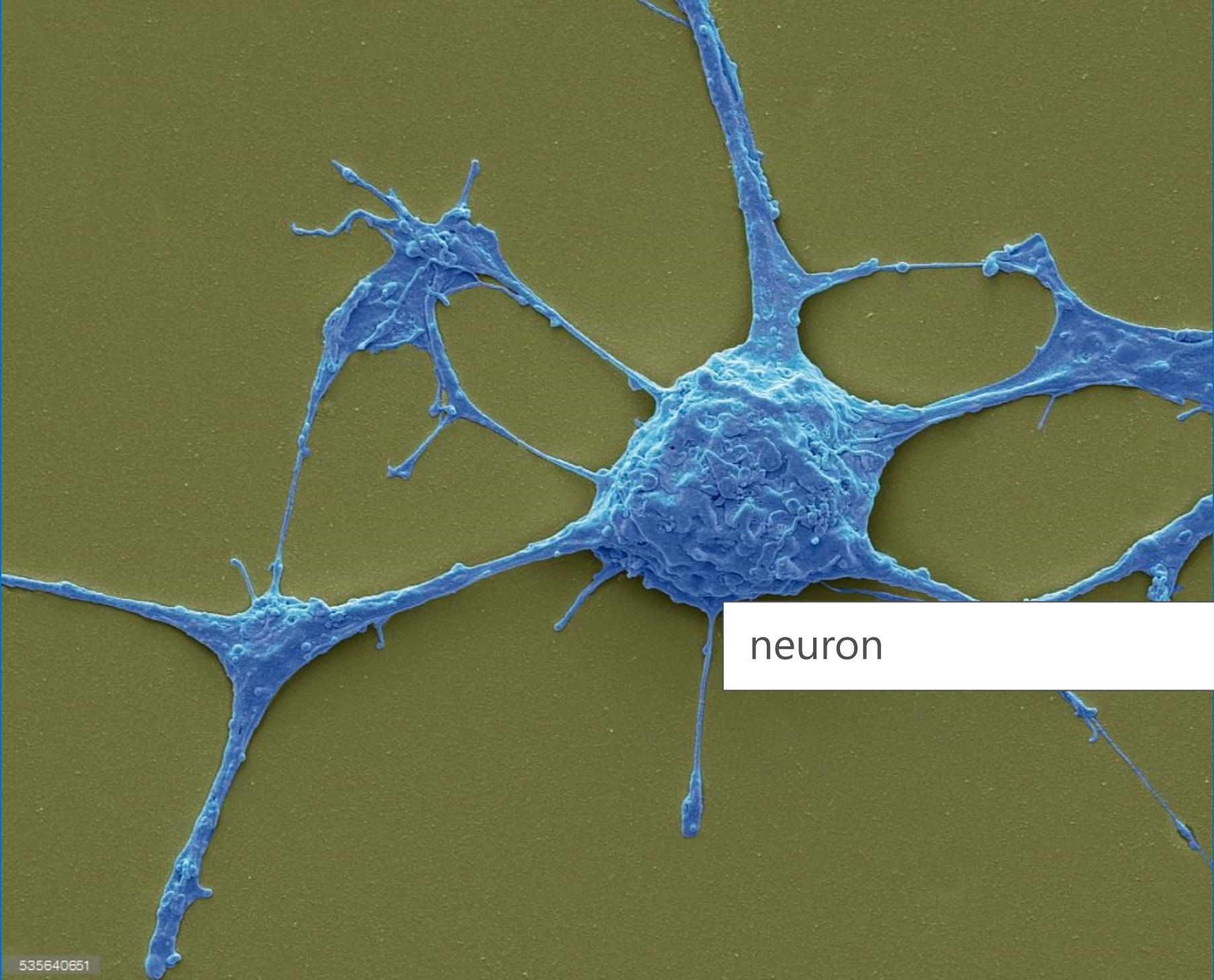
Creative Commons 3



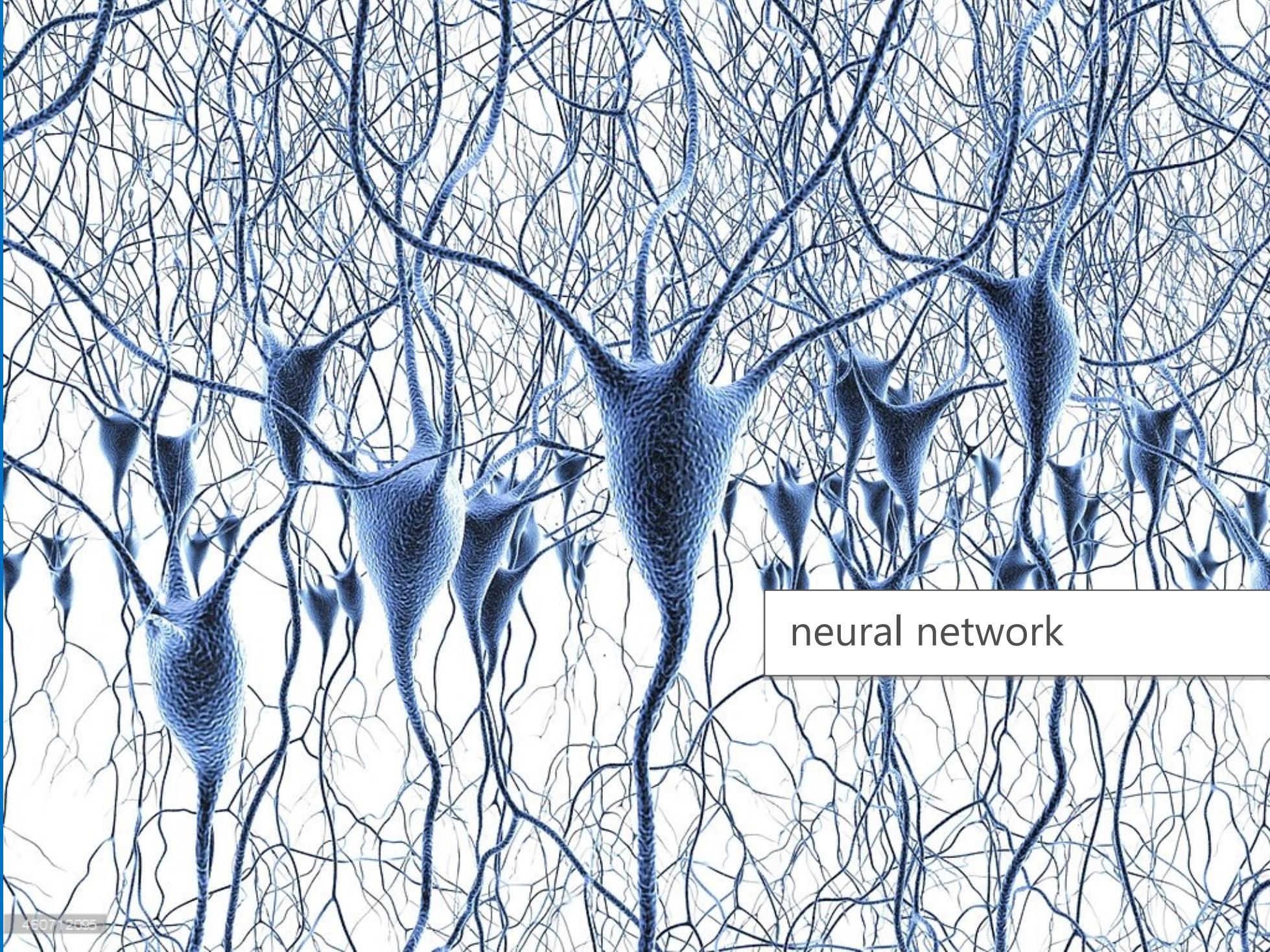
David Hemmings  
public domain

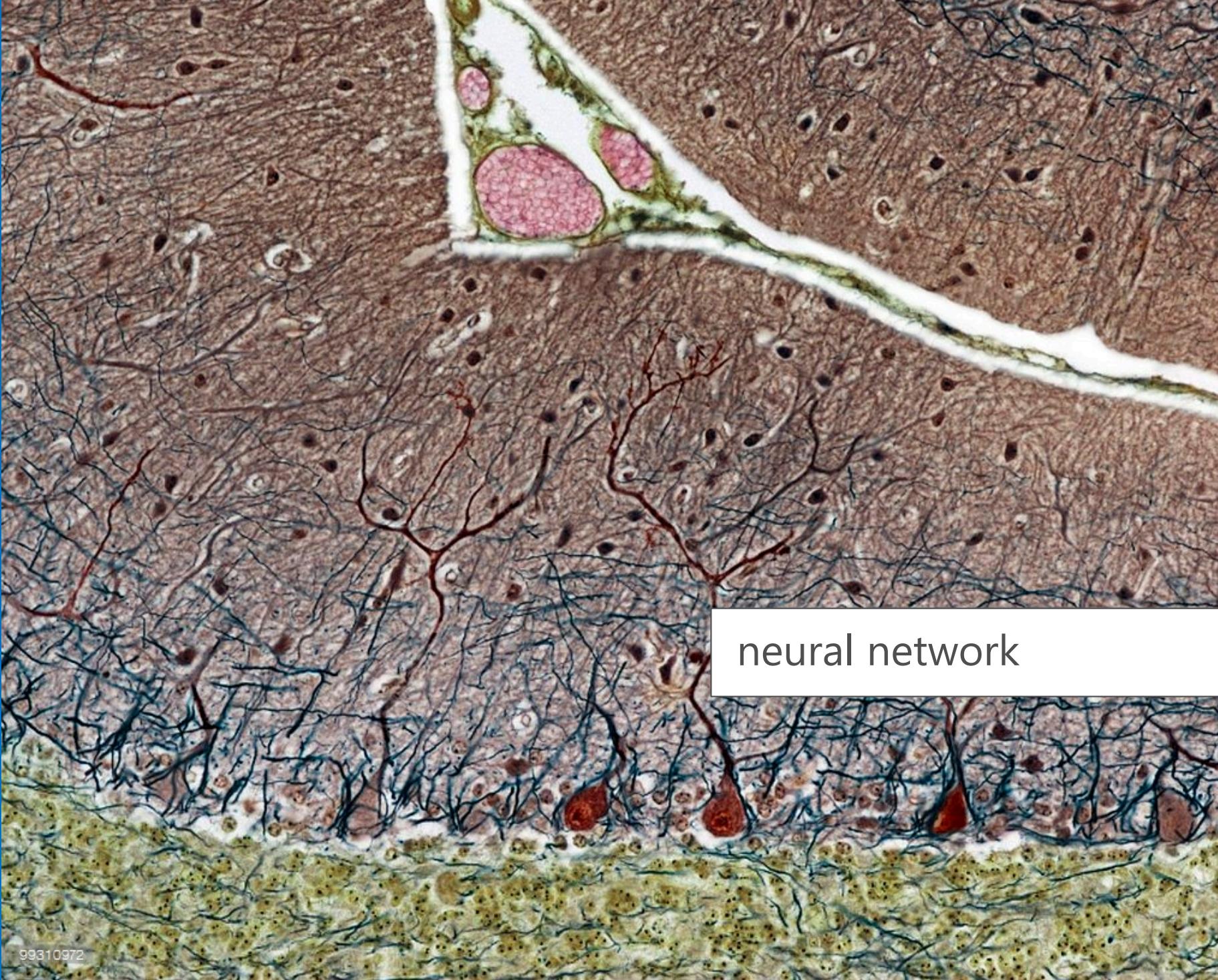
public domain





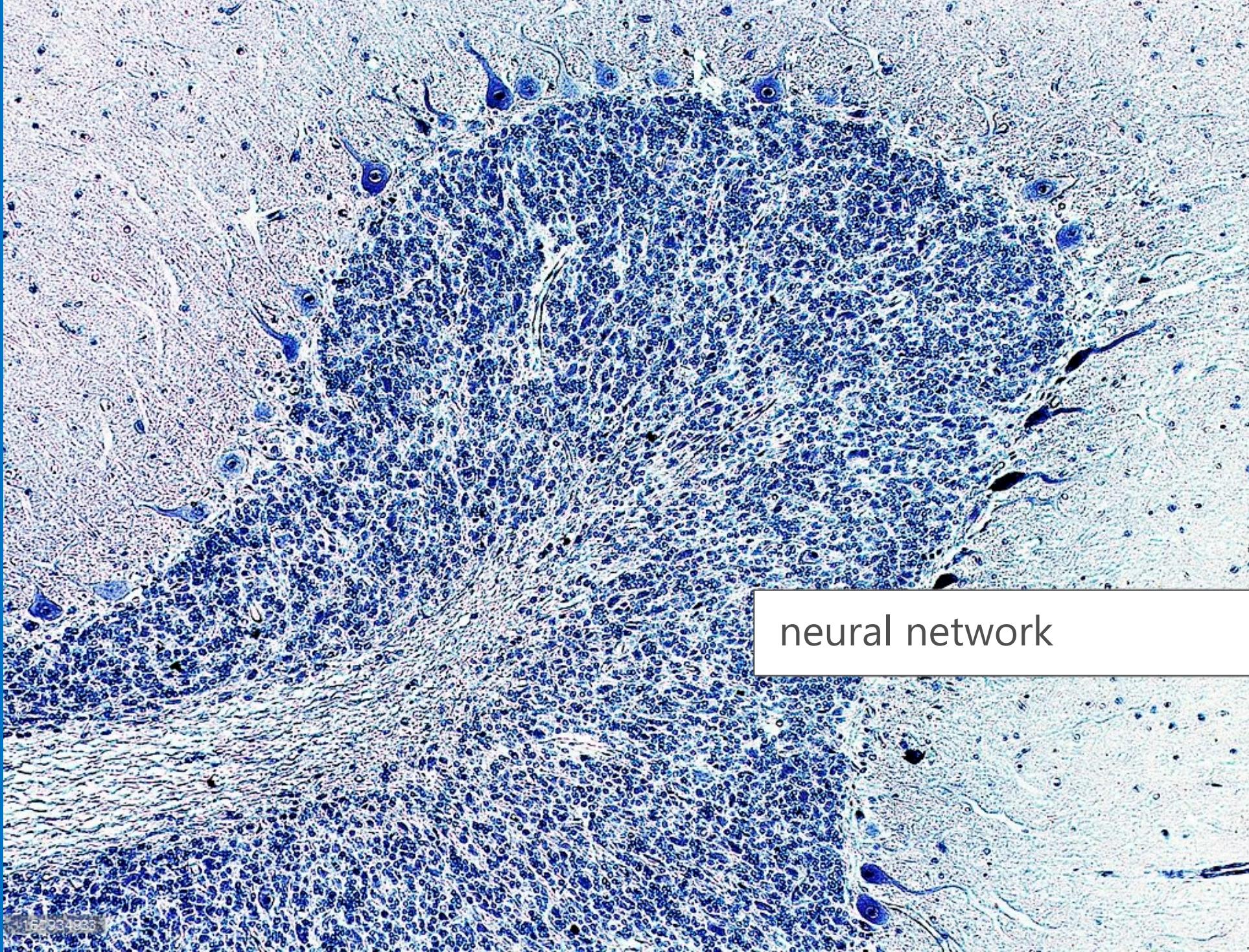
535640651





neural network

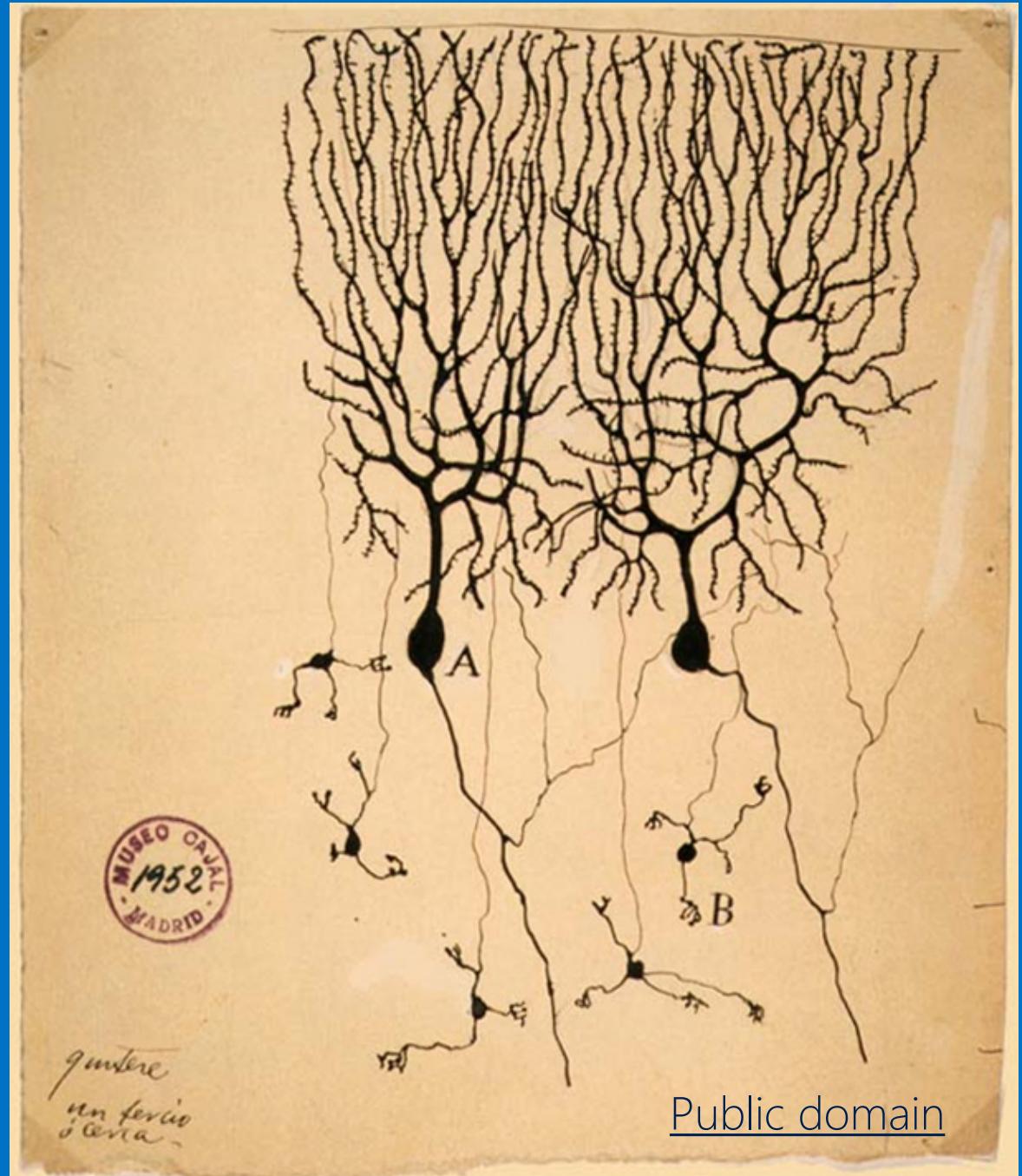
99310972

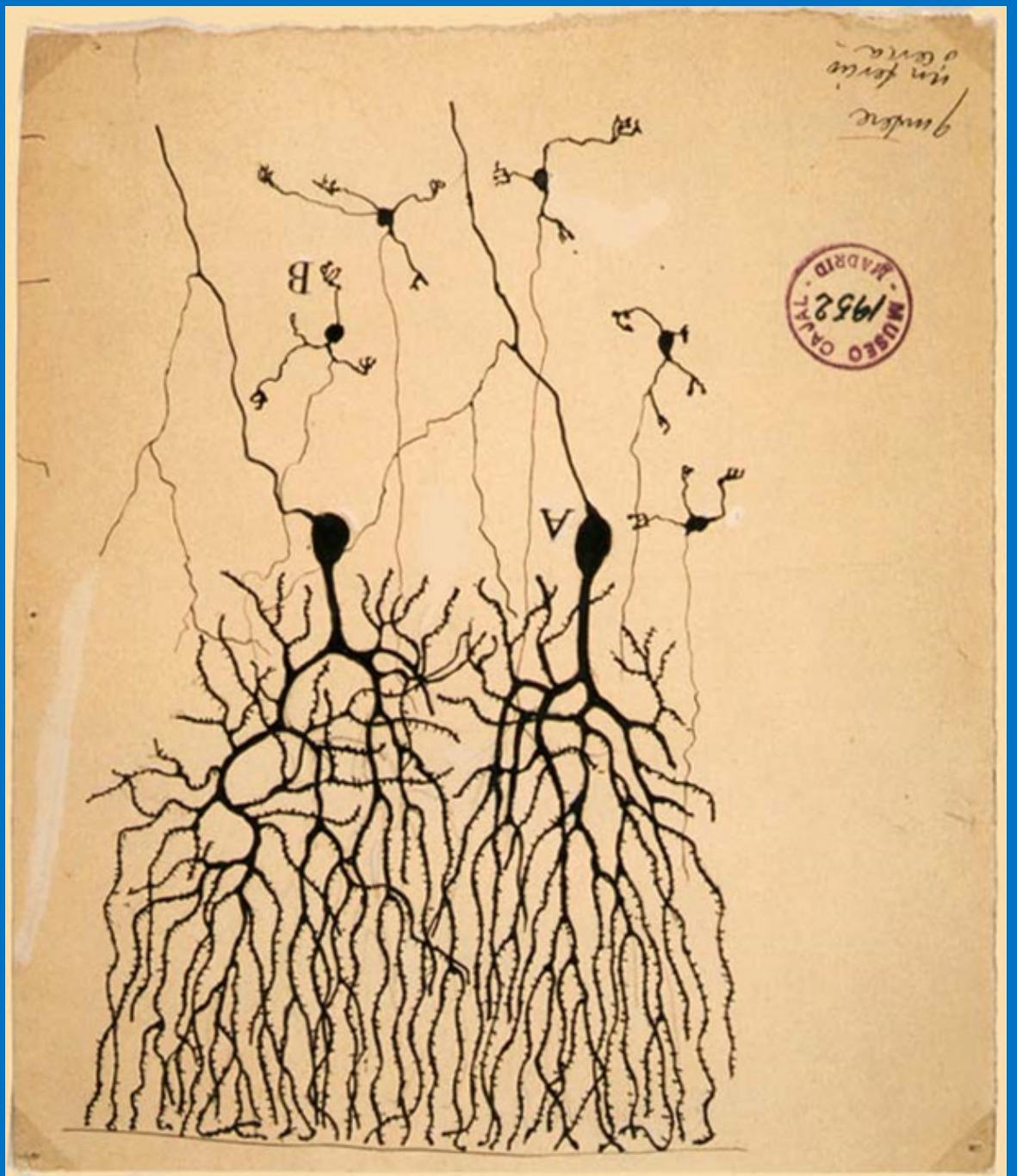


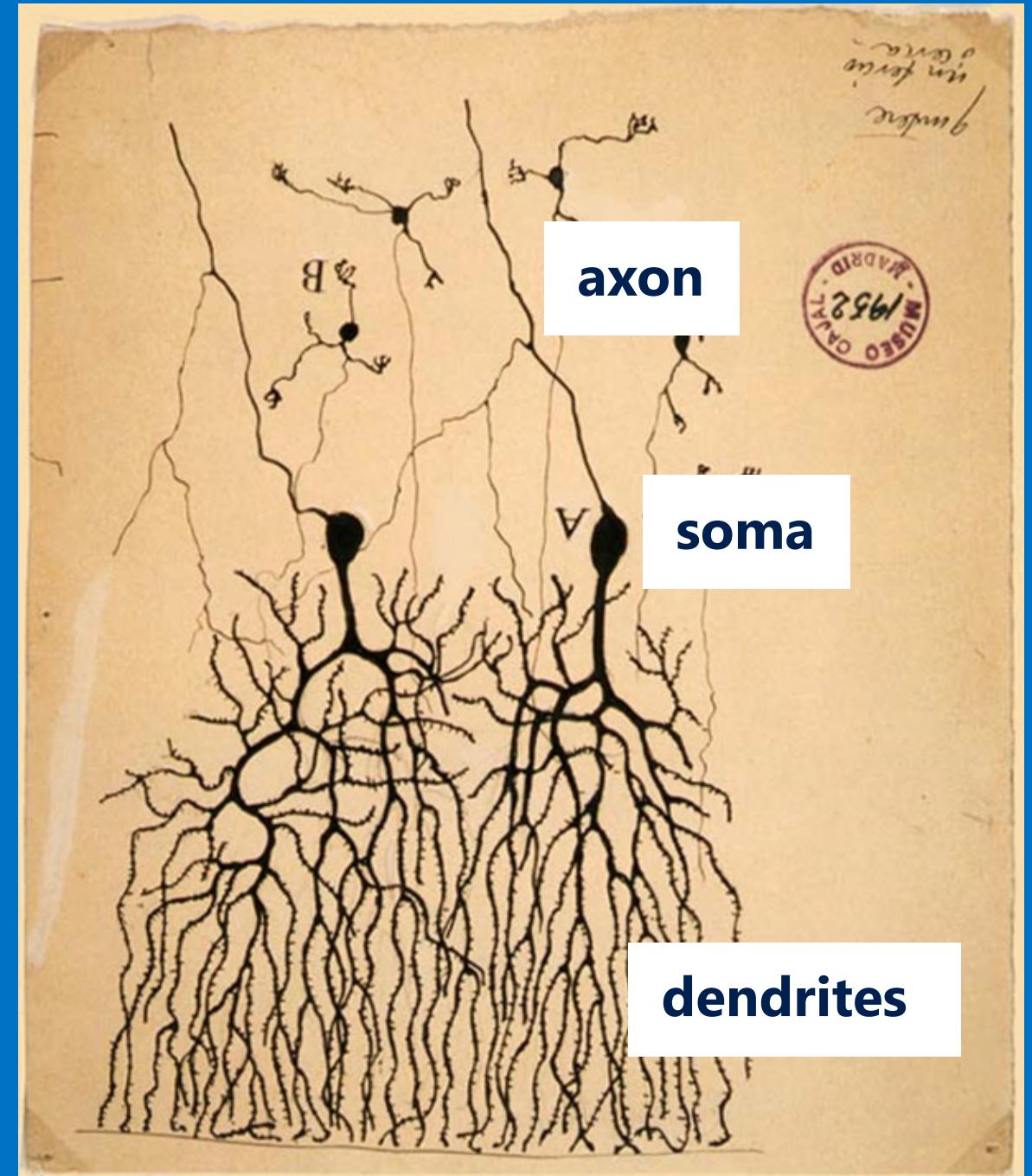
neural network

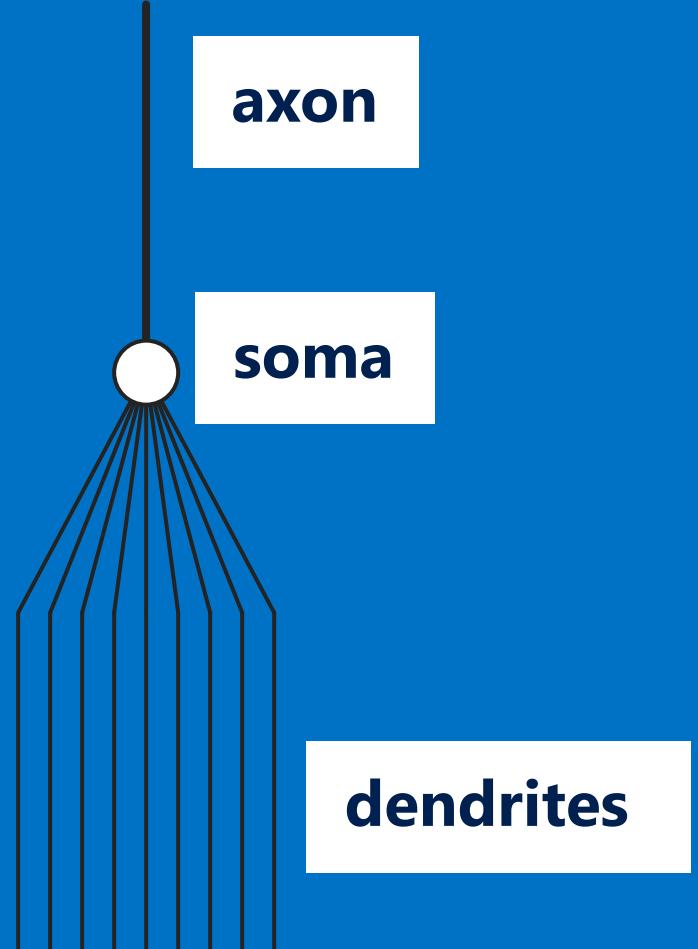
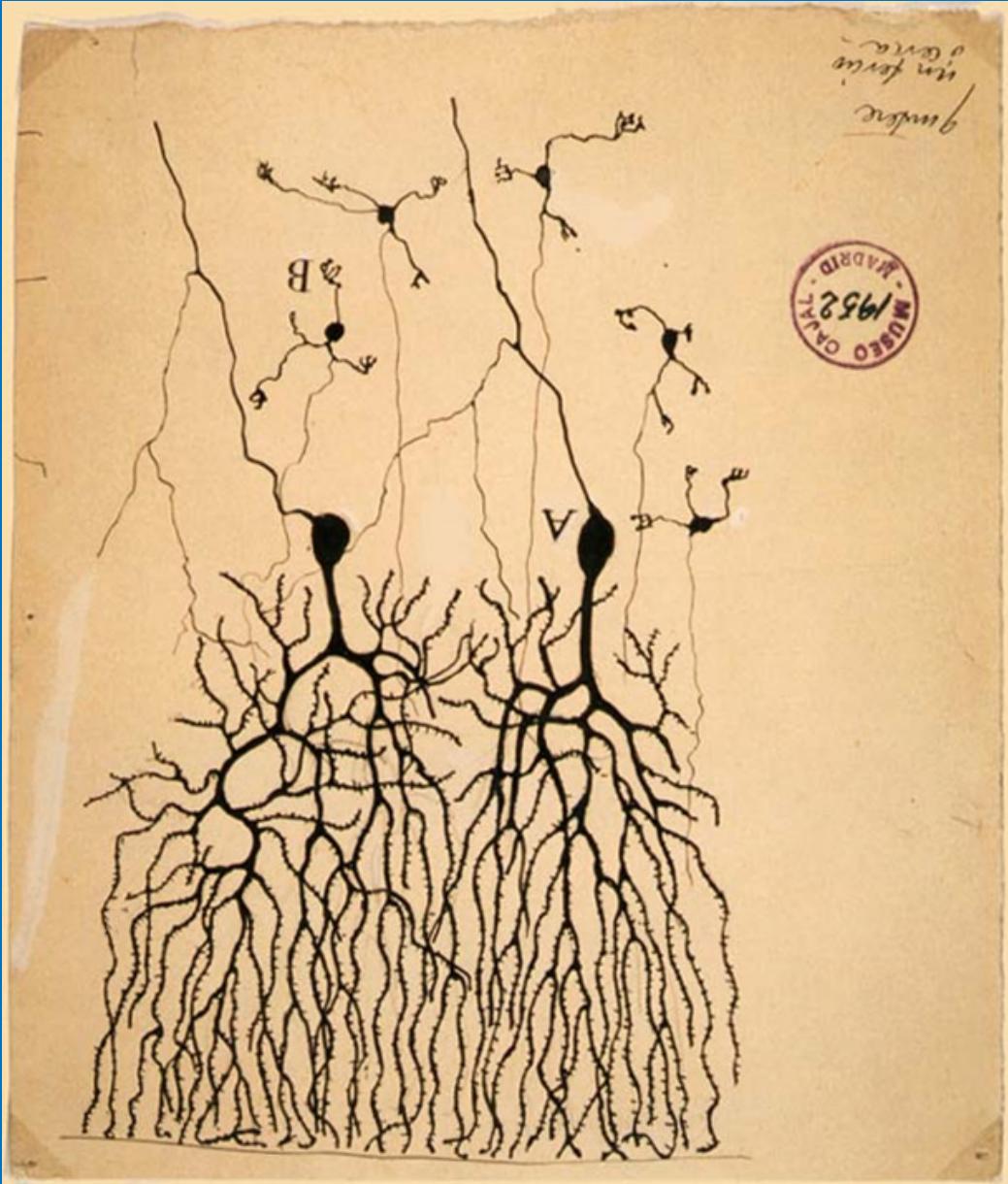
# Neurons

Drawing of  
Purkinje cells (A) and  
granule cells (B) from  
pigeon cerebellum by  
Santiago Ramón y Cajal,  
1899; *Instituto Cajal*,  
Madrid, Spain

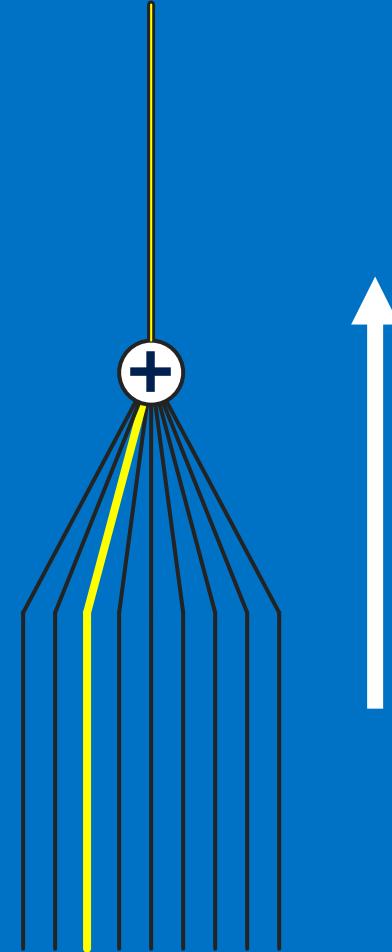




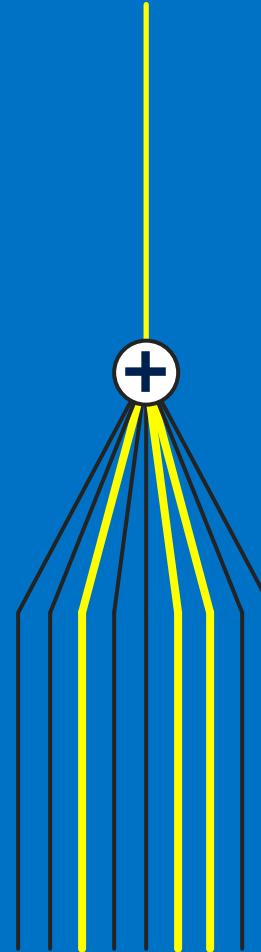


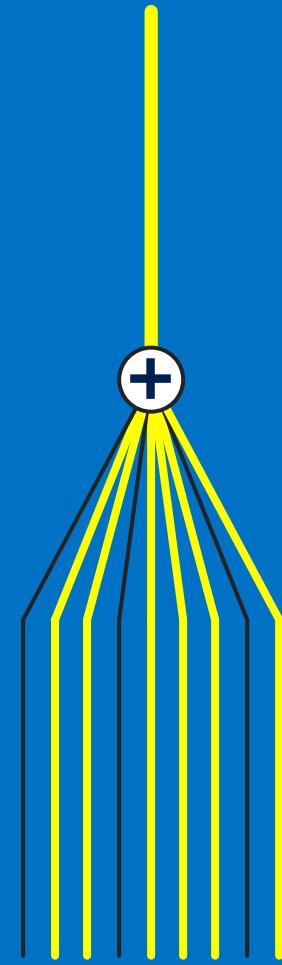


Soma adds dendrite activity together and passes it to axon.



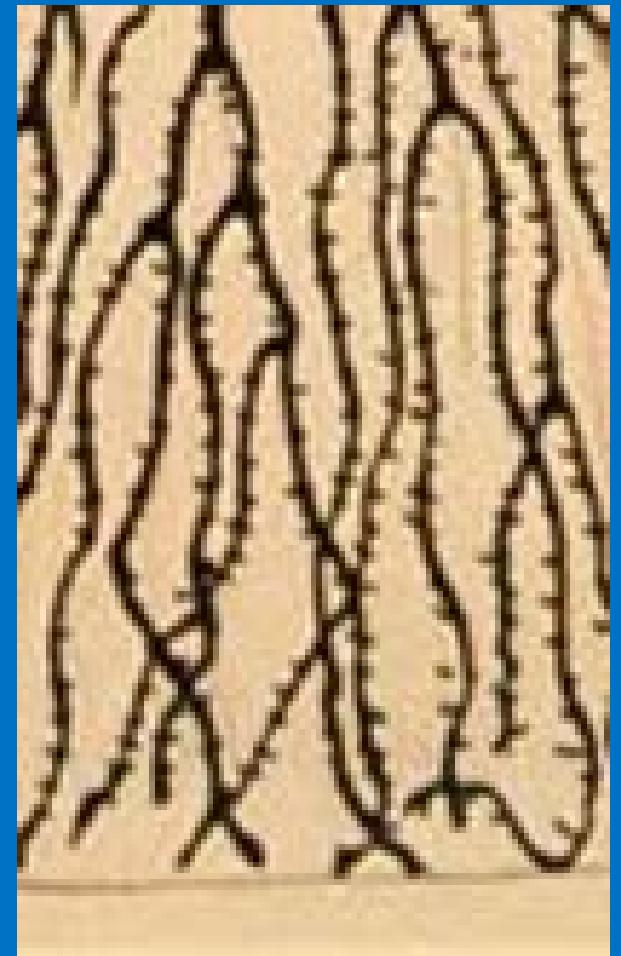
More dendrite activity  
makes more axon  
activity





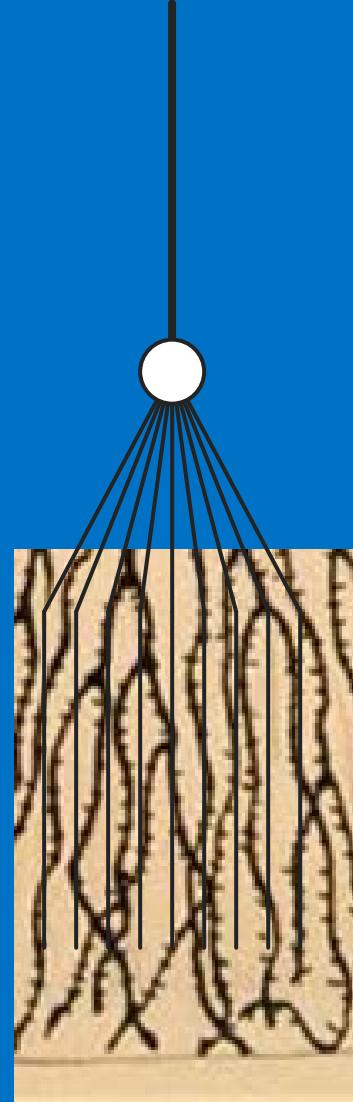
# Synapse

Connection between axon of one neuron and dendrites of another



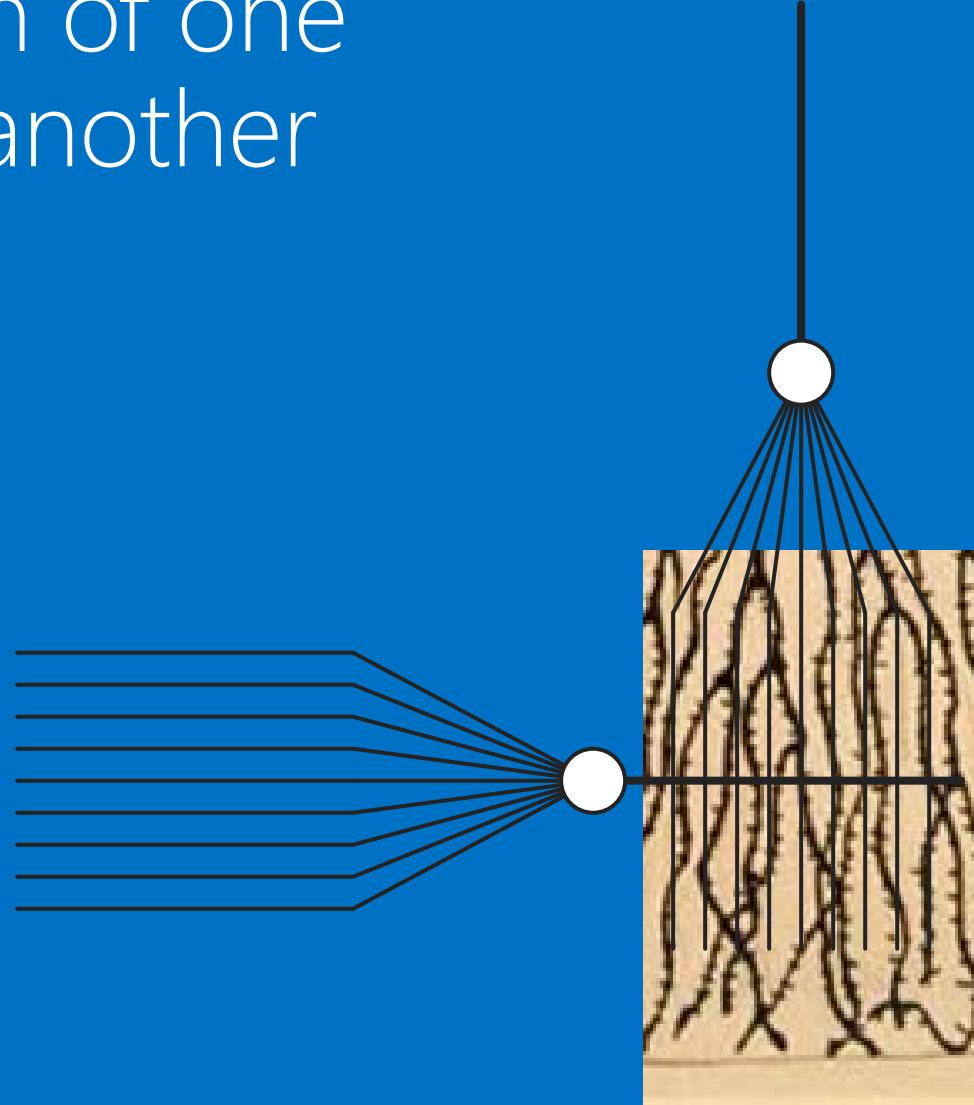
# Synapse

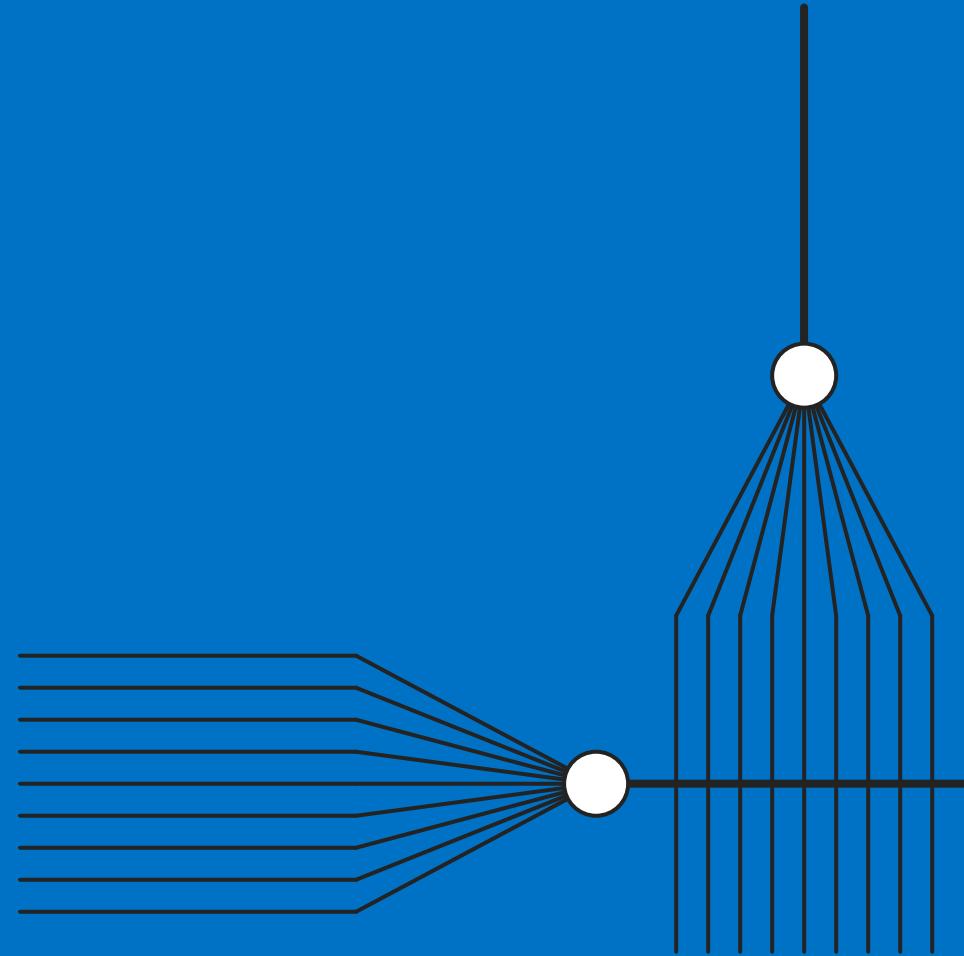
Connection between axon of one neuron and dendrites of another



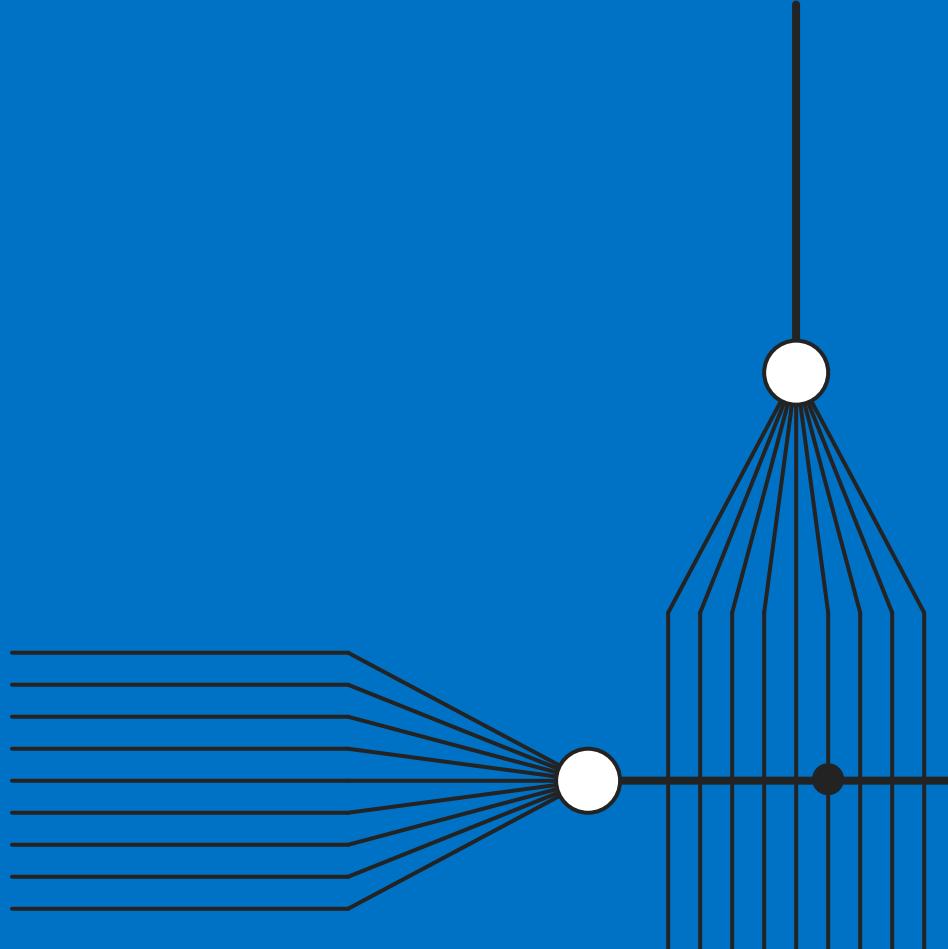
# Synapse

Connection between axon of one neuron and dendrites of another

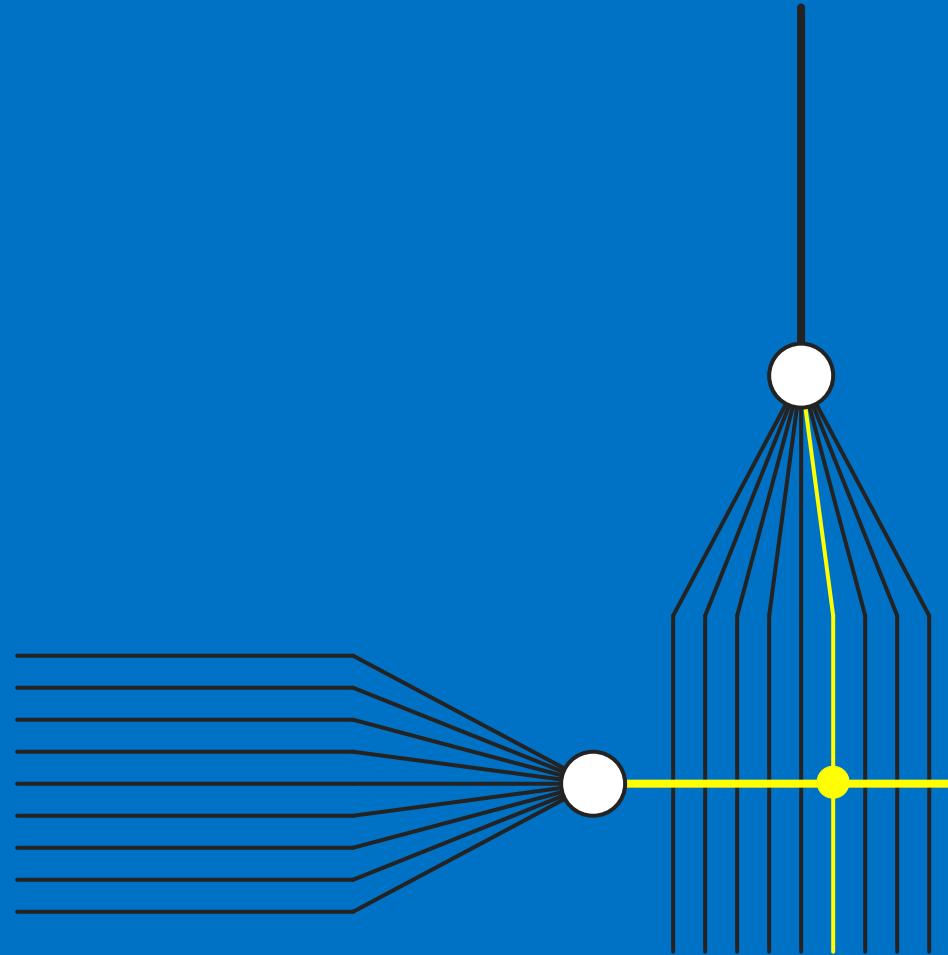




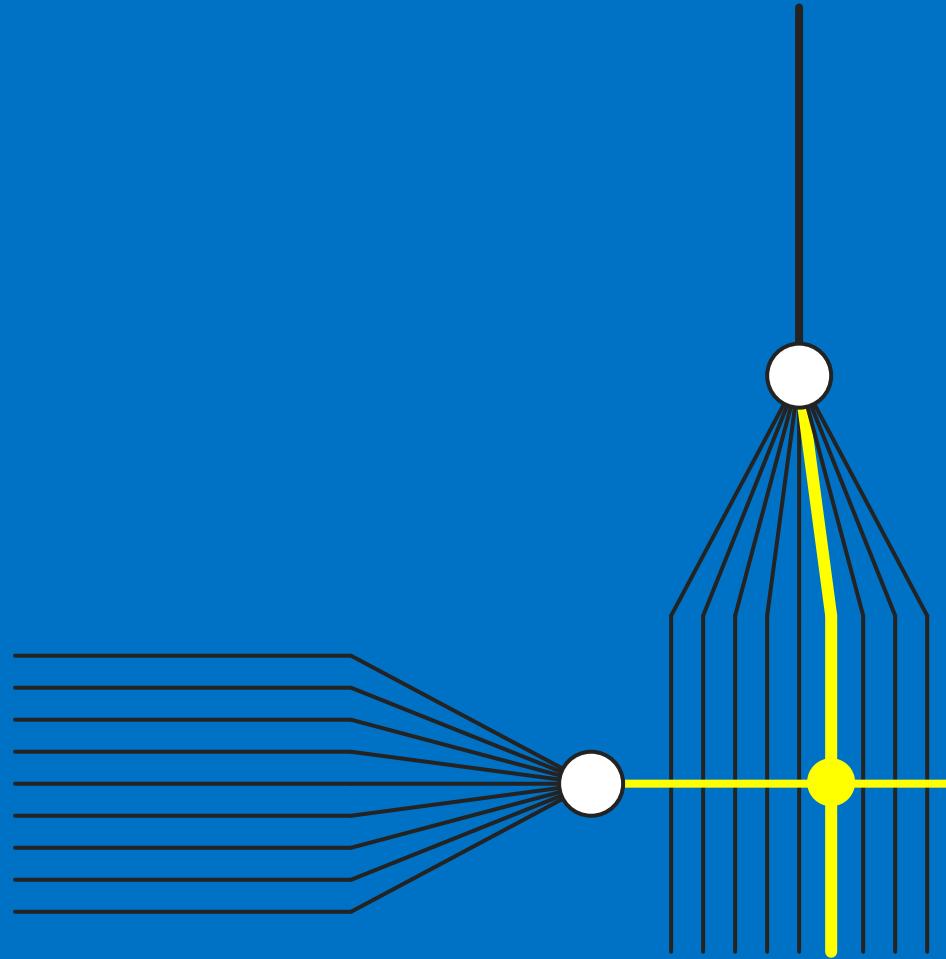
Axons can connect to dendrites strongly, weakly, or somewhere in between.



# Medium connection (.6)

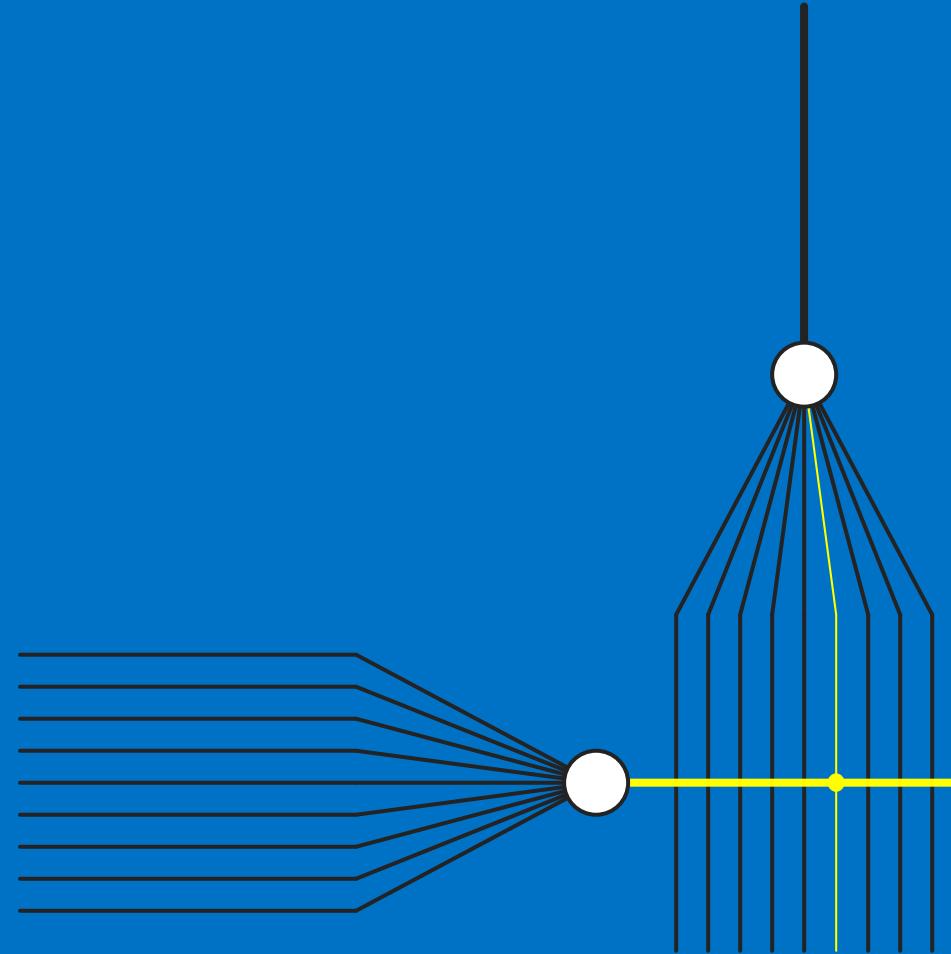


# Strong connection (1.0)

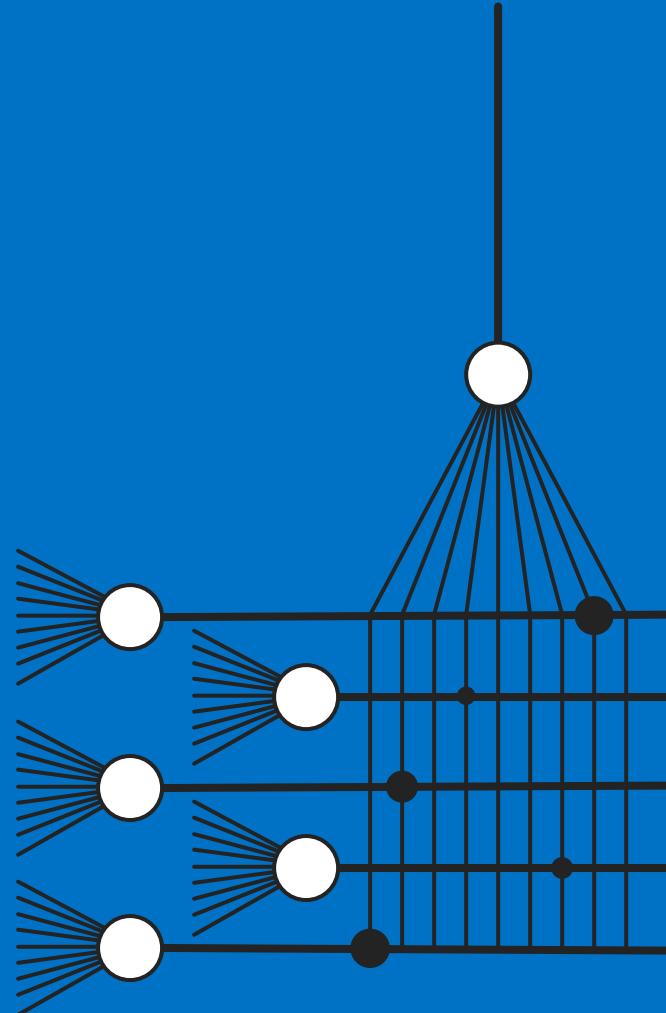


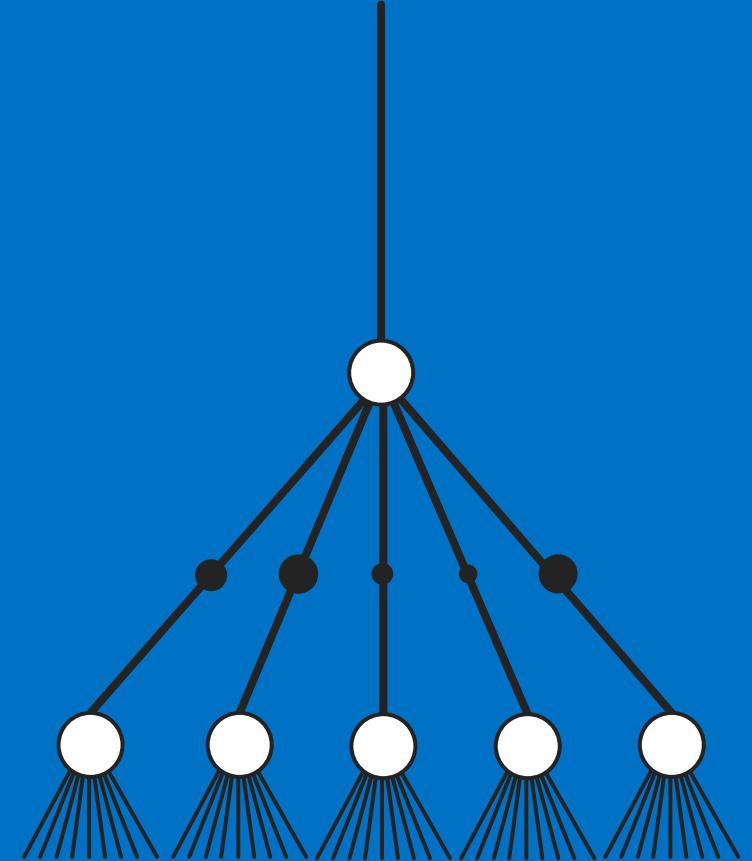
## Weak connection (.2)

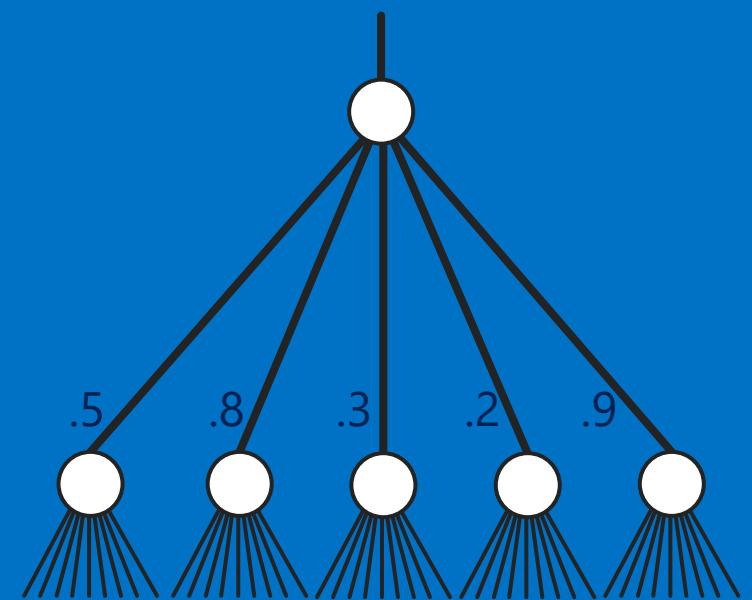
No connection is a 0.

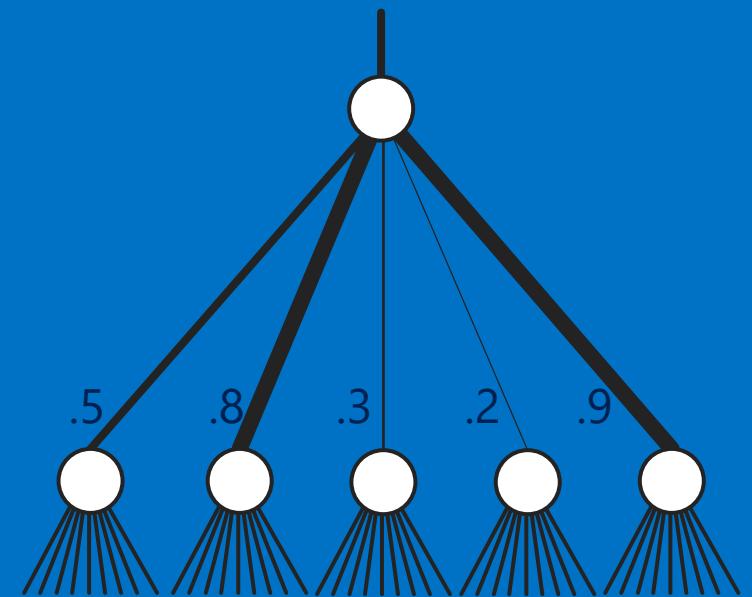


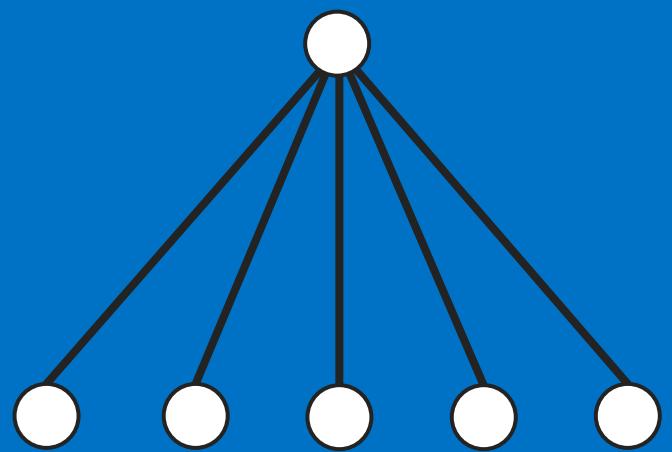
Lots of axons connect with  
the dendrites of one neuron.  
Each has its own connection  
strength.

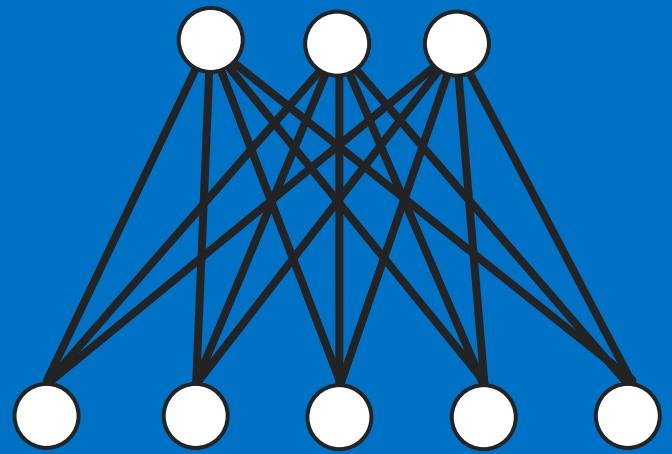


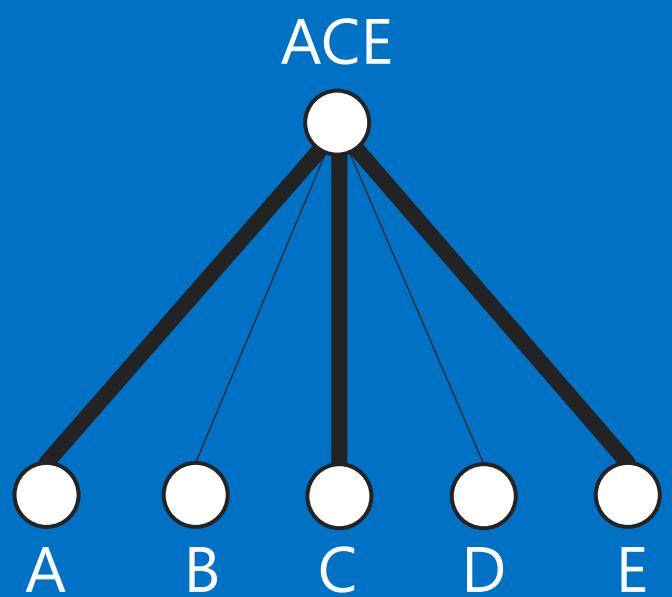


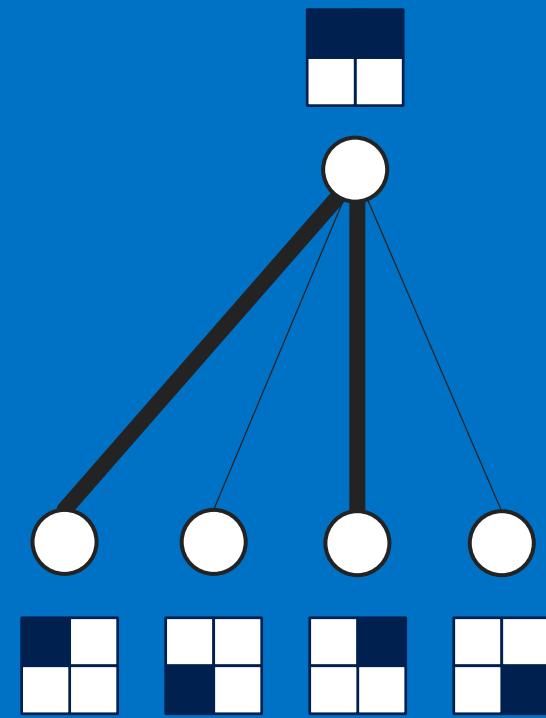


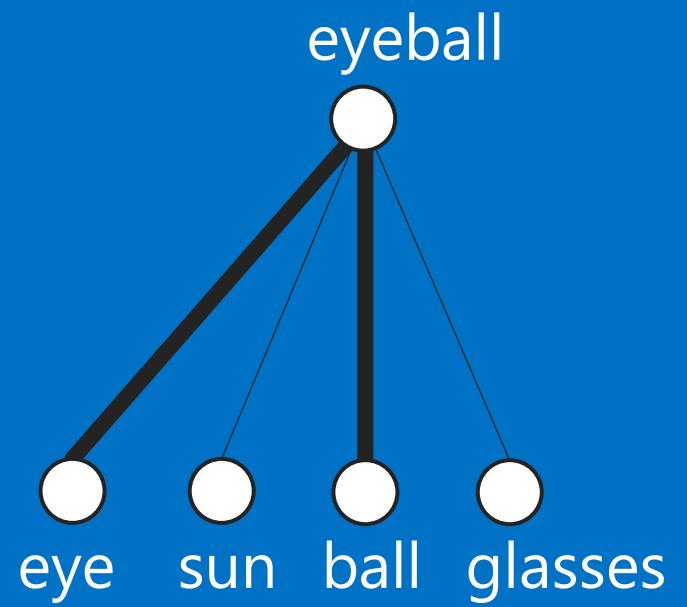




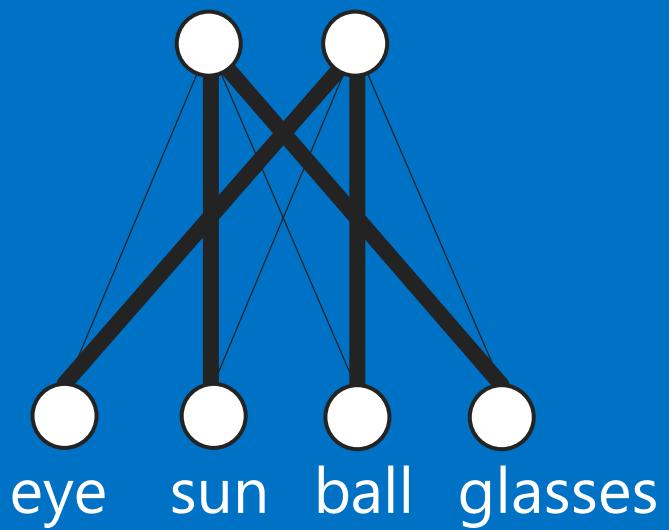




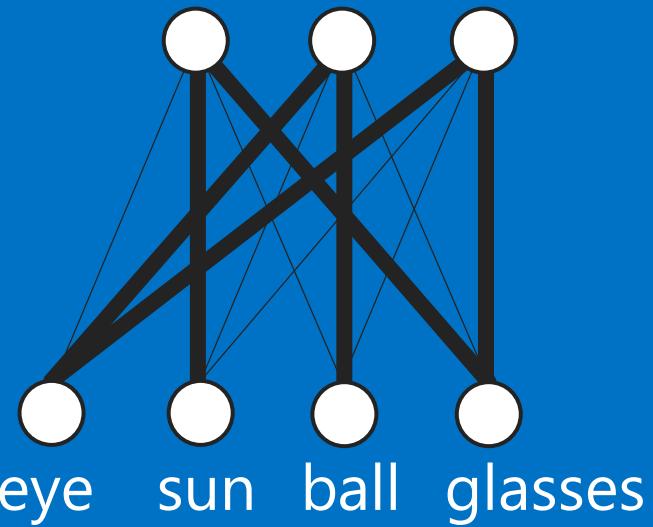




sunglasses eyeball

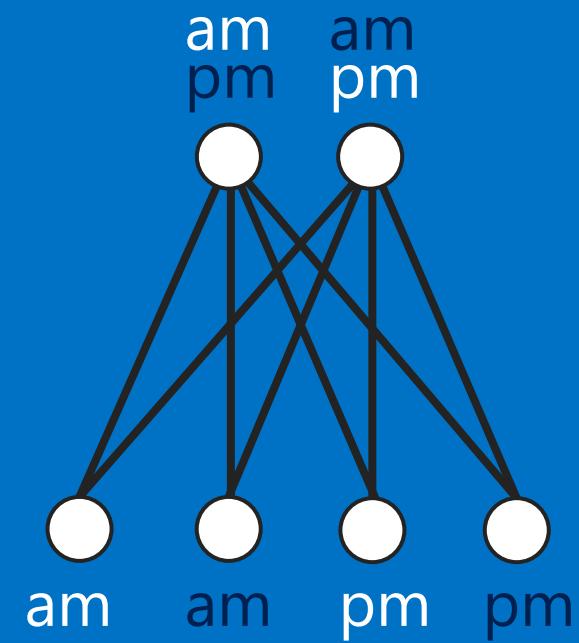


sunglasses eyeball eyeglasses

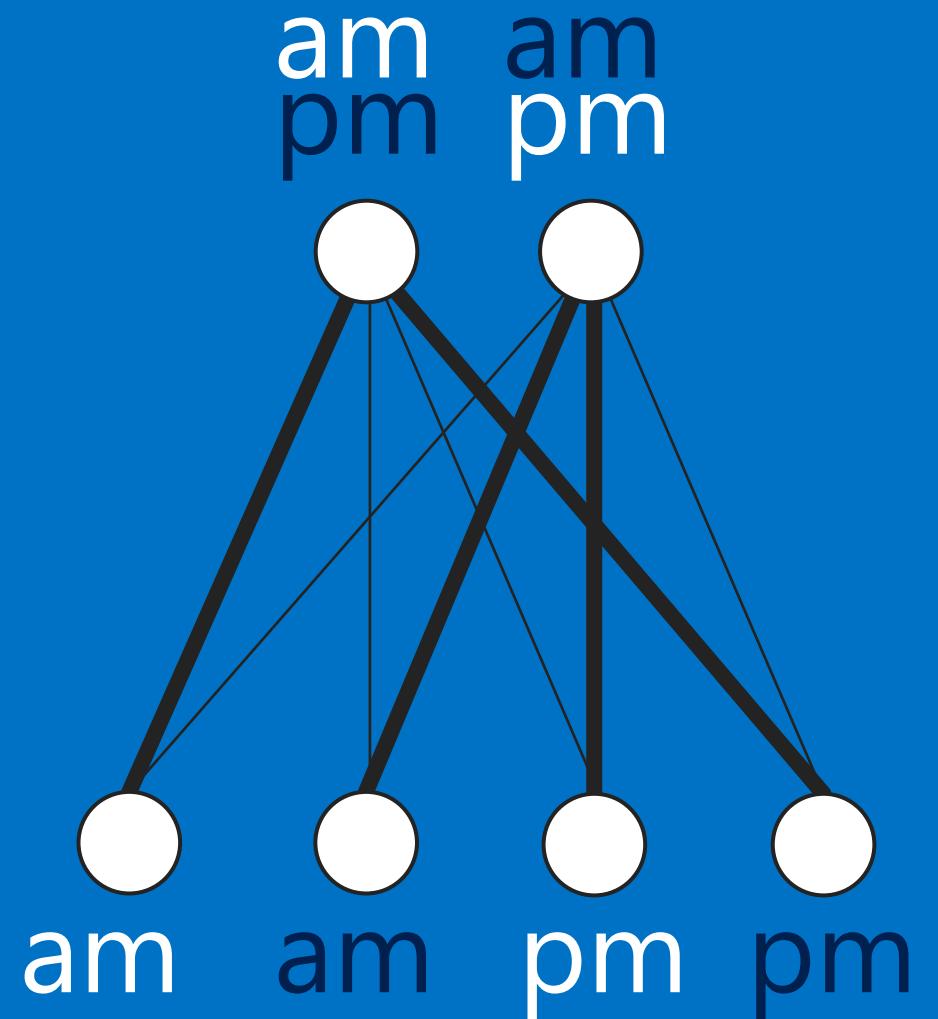


# The guy at the shawarma place

Sometimes he works  
mornings, sometimes  
evenings

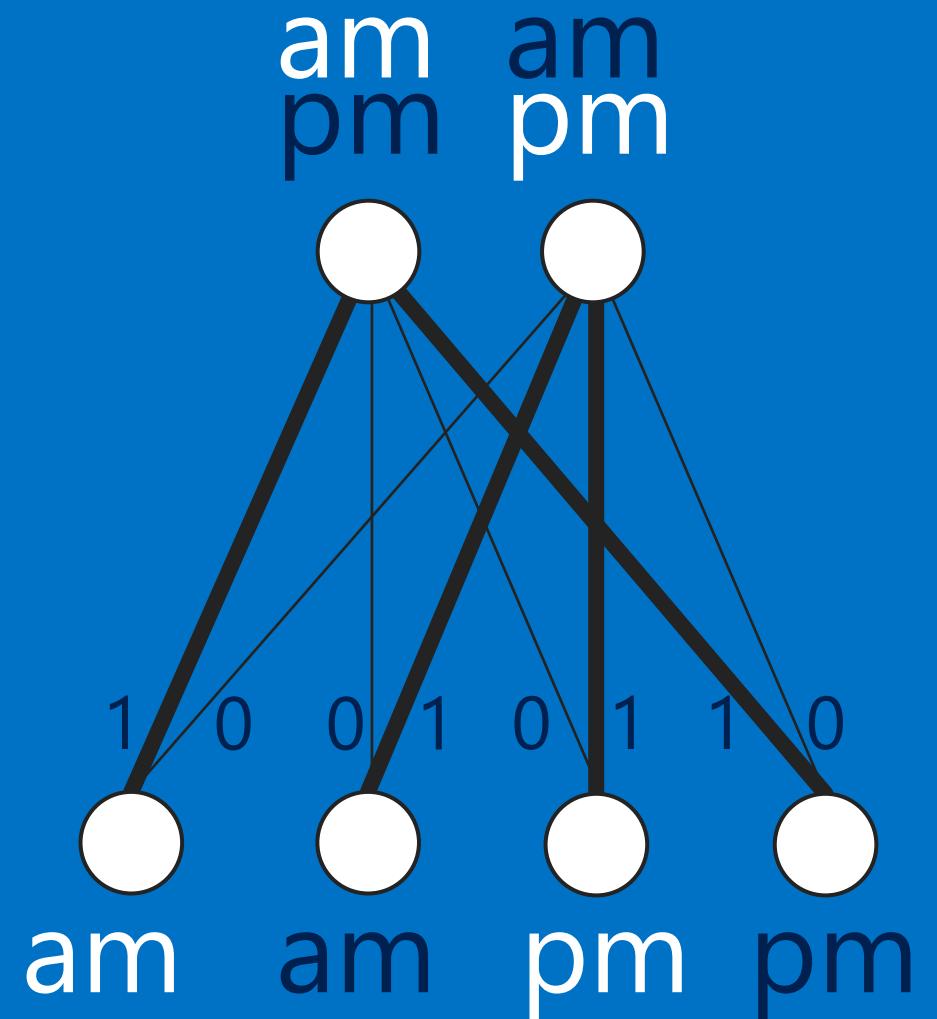


This neural network captures  
the patterns in his schedule.



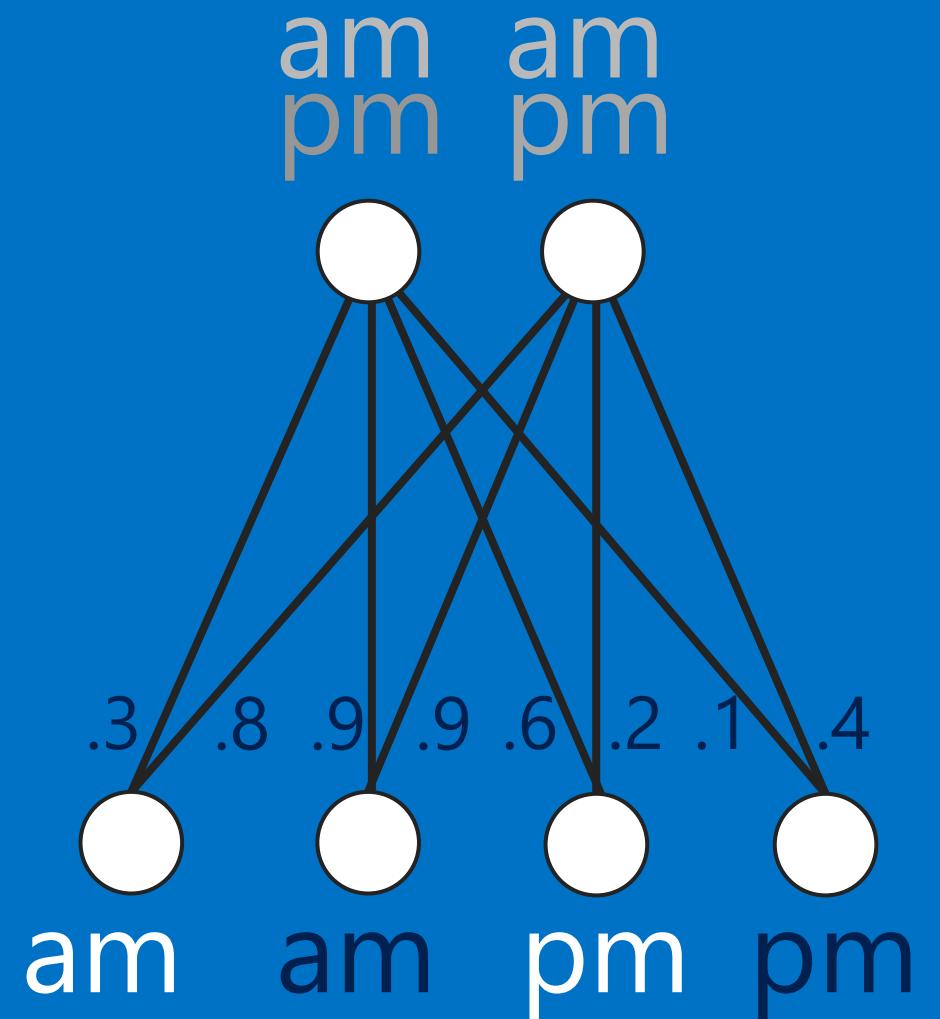
# Training

How do we learn this?

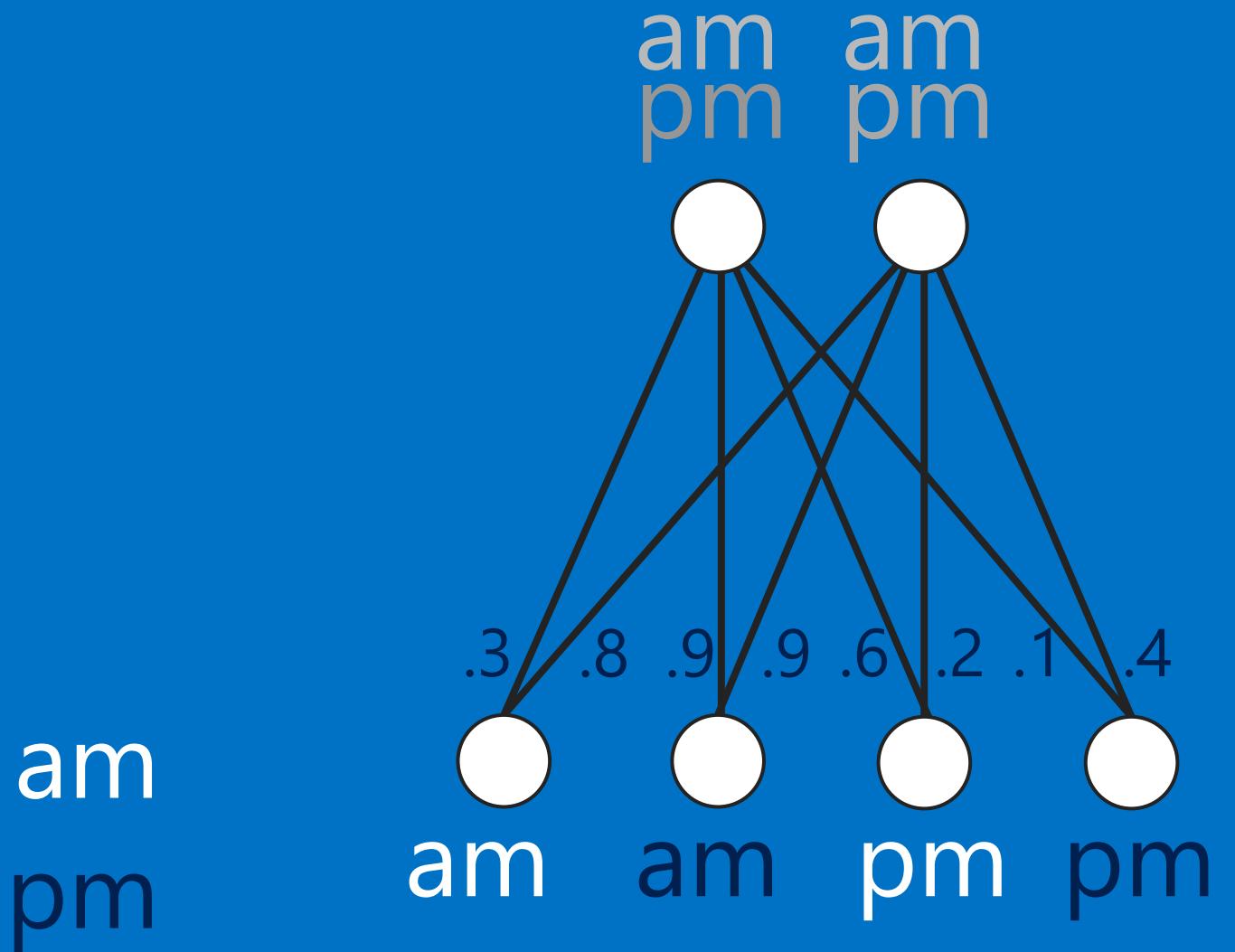


Initialize

Start random



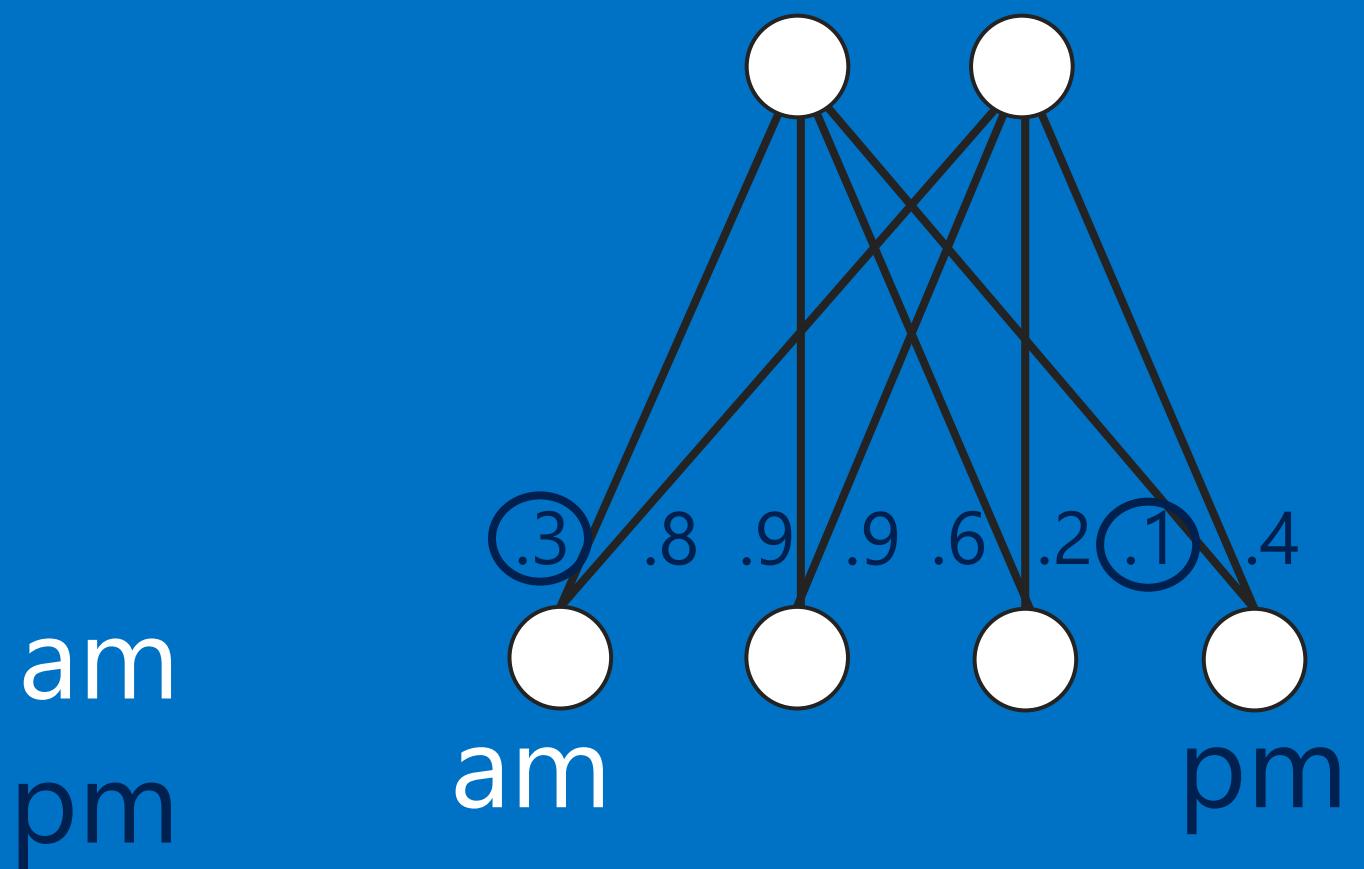
Gather data



Activate each of the neurons

$$(.3+.1)/2=.2$$

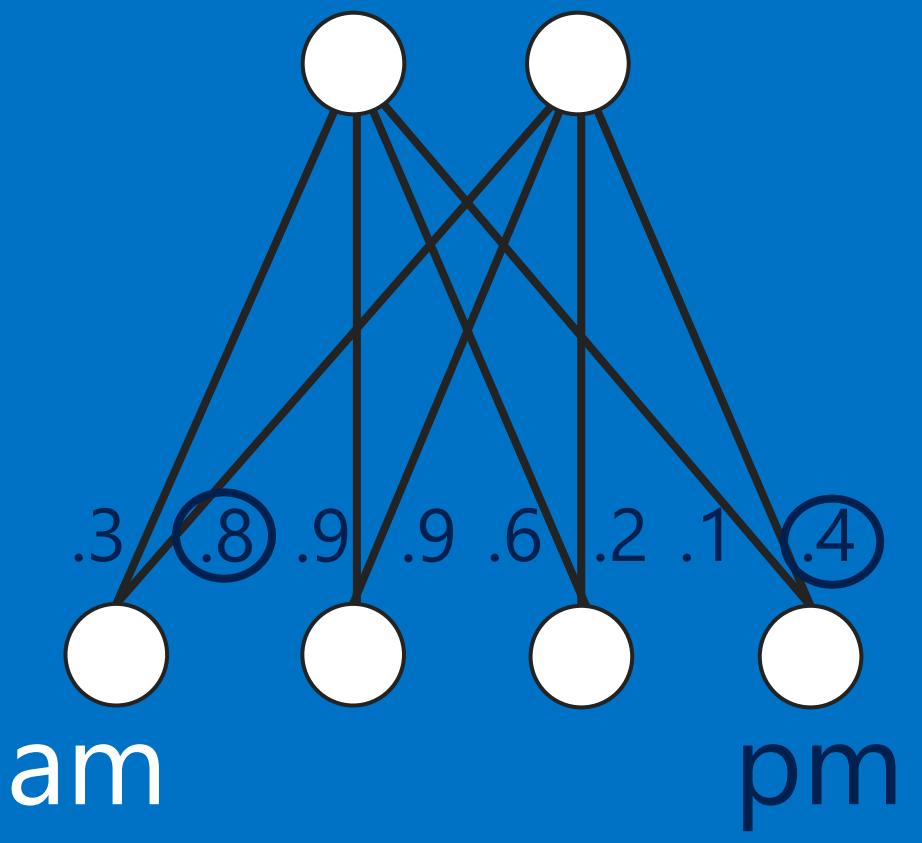
am  
pm am  
pm



Activate each of the neurons

am  
pm

$$.2 \begin{matrix} am \\ pm \end{matrix} \quad am \\ pm \quad (.8+.4)/2=.6$$



# Error

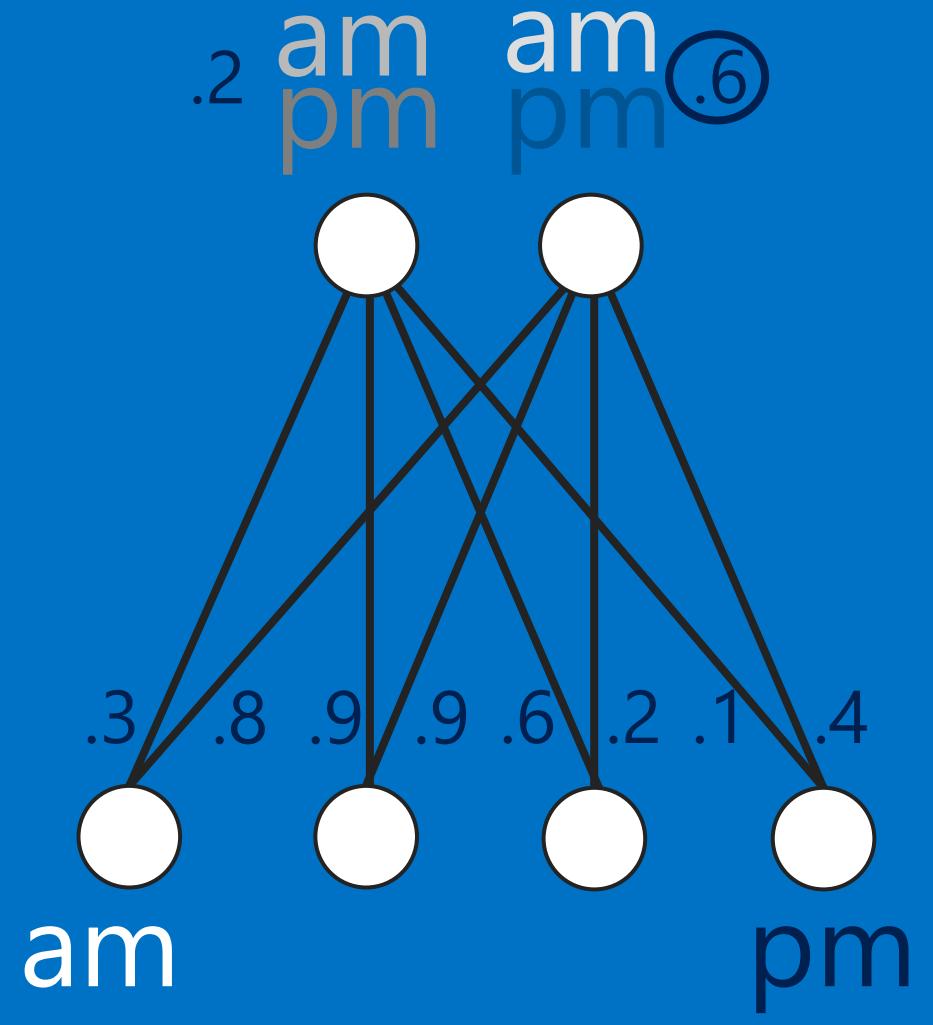
Calculate how bad  
the best fit neuron is.

If it's a perfect fit,  
there is no need to  
update the weights.

If the best fit is bad,  
the weights need a  
lot of work.

am  
pm

$$\begin{aligned} \text{Error} &= 1 - .6 \\ &= .4 \end{aligned}$$

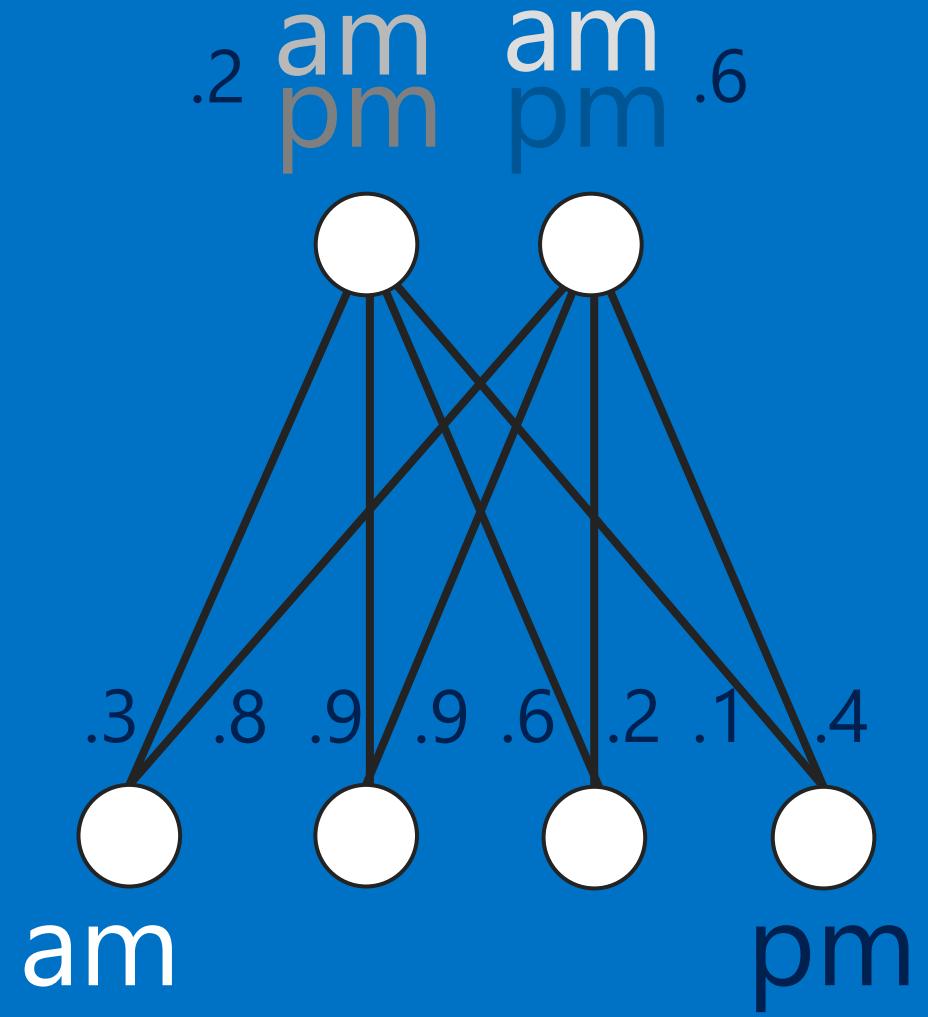


# Gradient descent

For each weight,  
adjust it up and down  
a bit and see how the  
error changes.

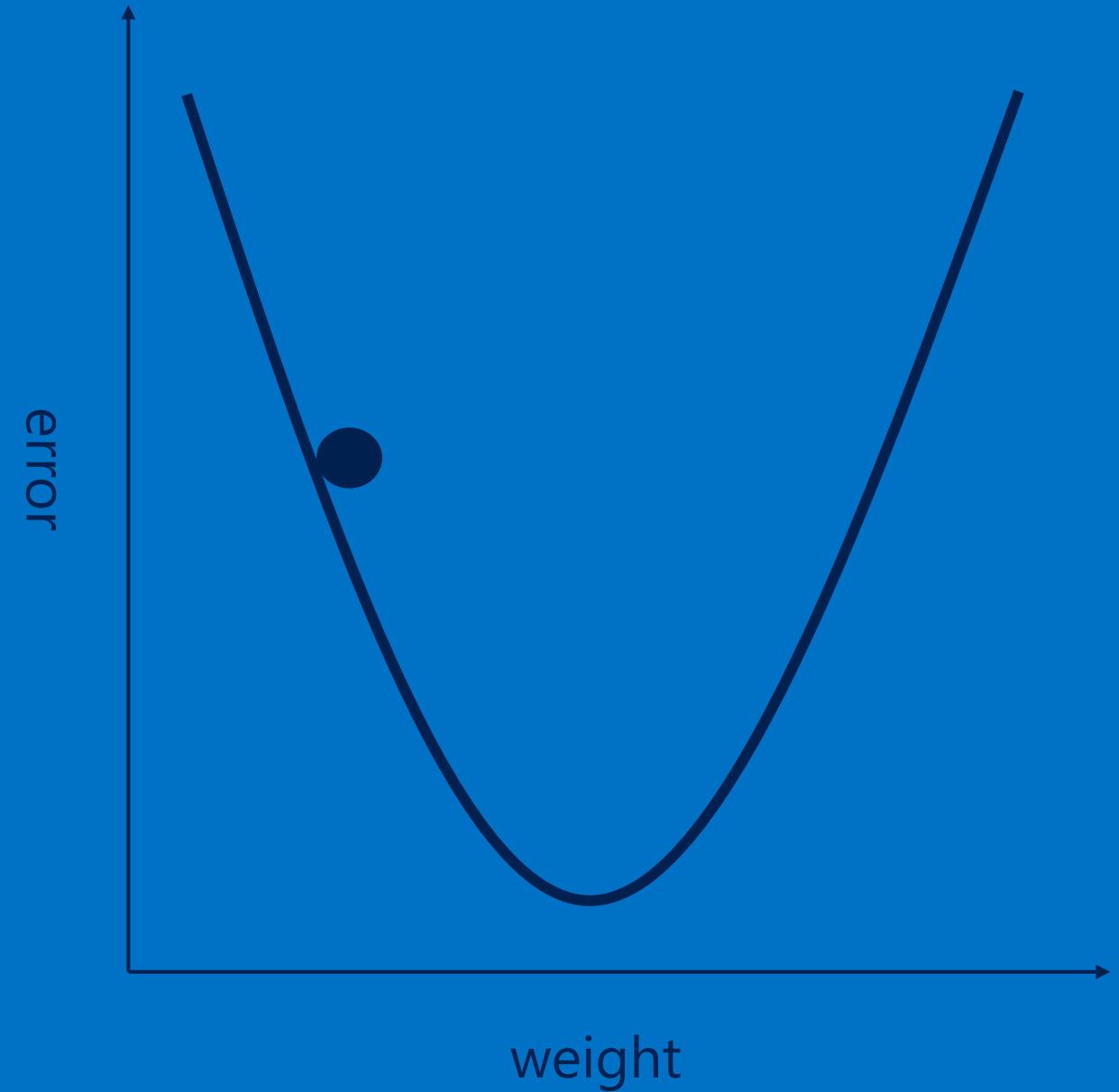
Error=.4

am  
pm



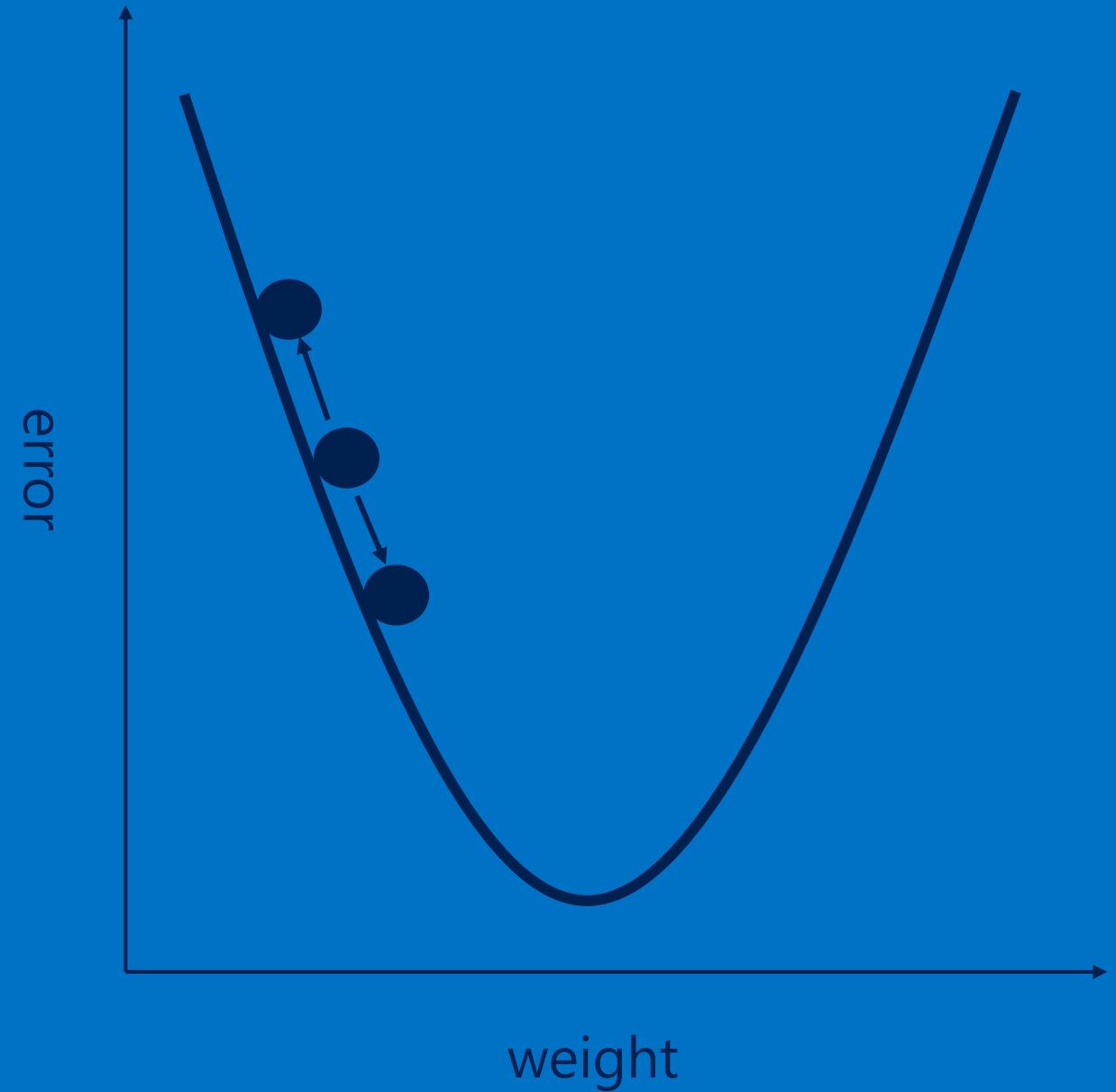
# Gradient descent

For each weight,  
adjust it up and down  
a bit and see how the  
error changes.



# Gradient descent

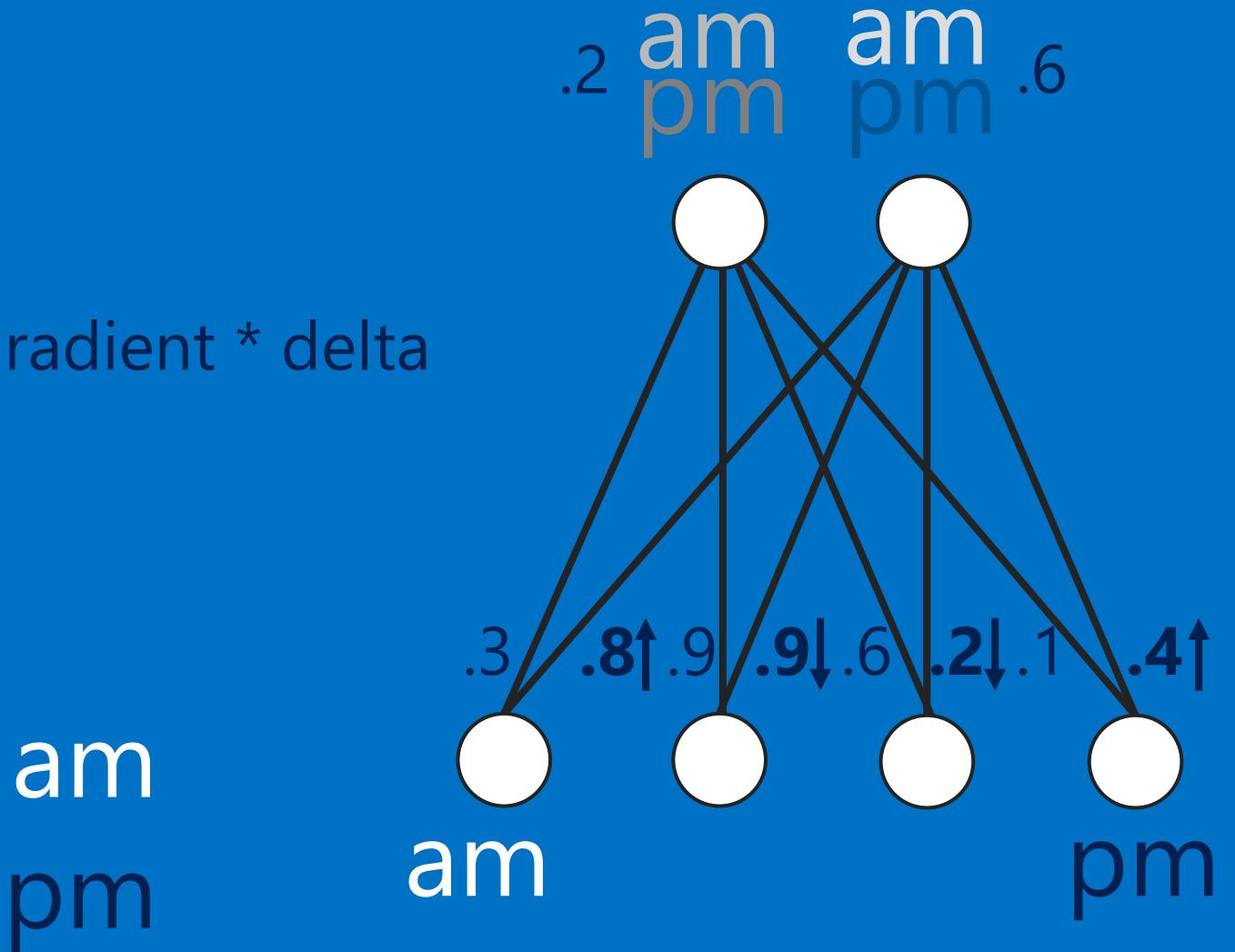
For each weight,  
adjust it up and down  
a bit and see how the  
error changes.



# Backpropagation

Adjust all the weights

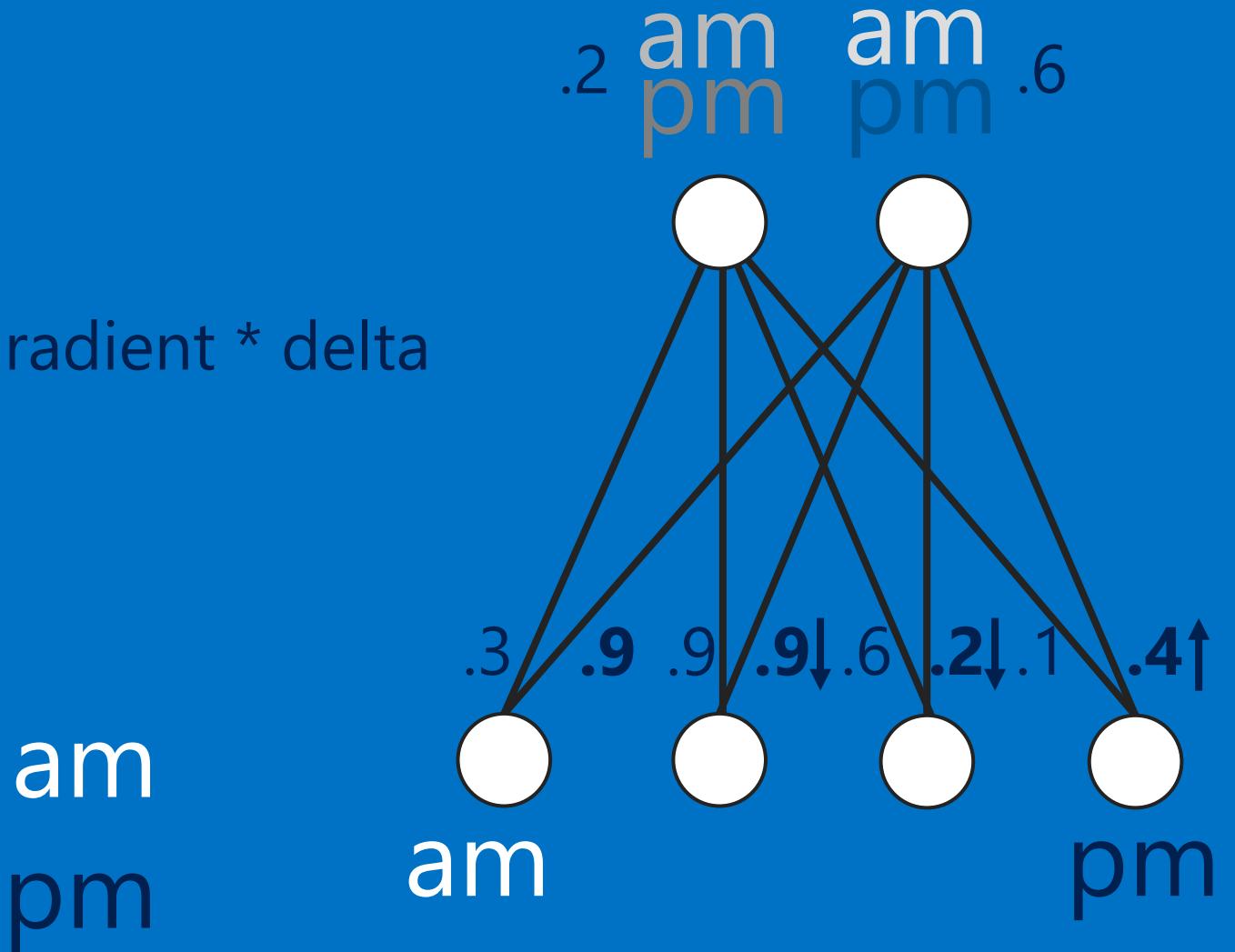
adjustment = error \* gradient \* delta



# Backpropagation

Adjust all the weights

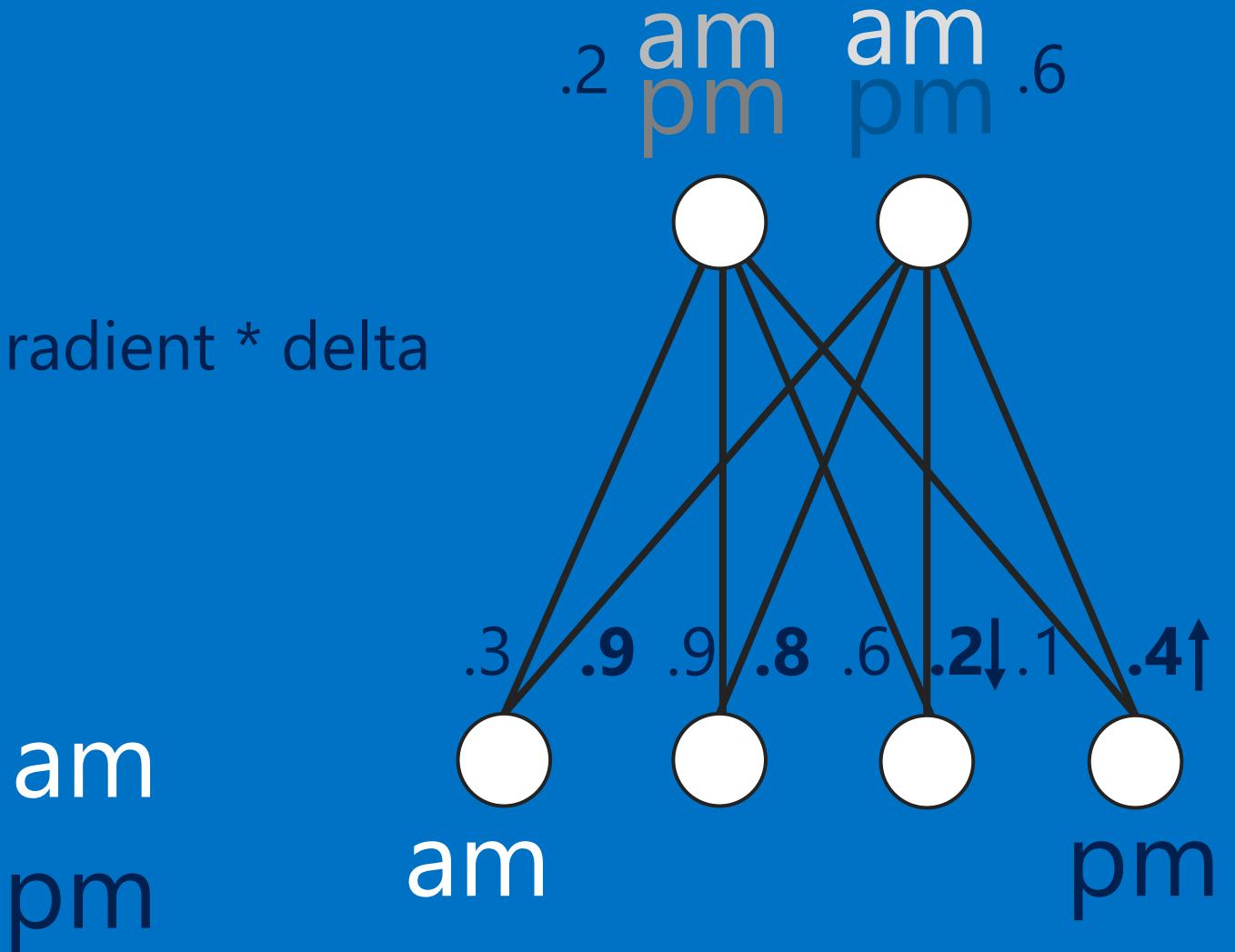
adjustment = error \* gradient \* delta



# Backpropagation

Adjust all the weights

adjustment = error \* gradient \* delta

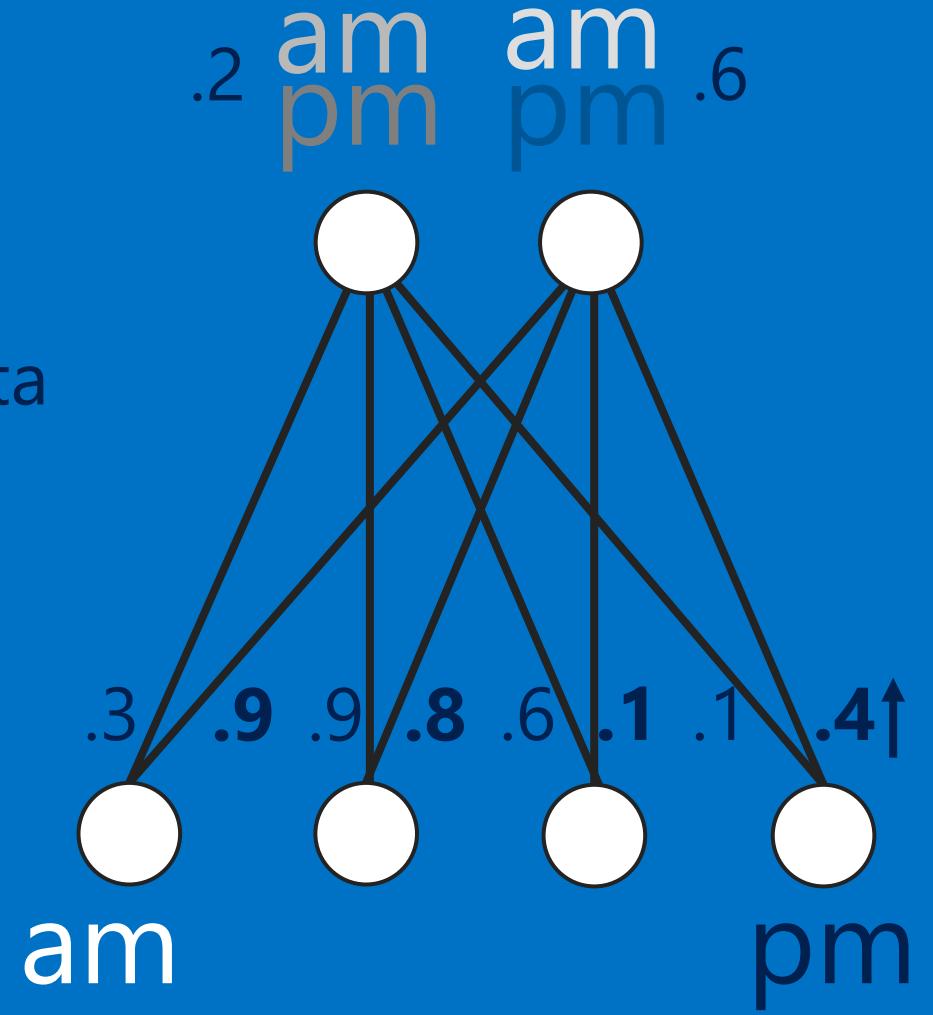


# Backpropagation

Adjust all the weights

adjustment = error \* gradient \* delta

am  
pm

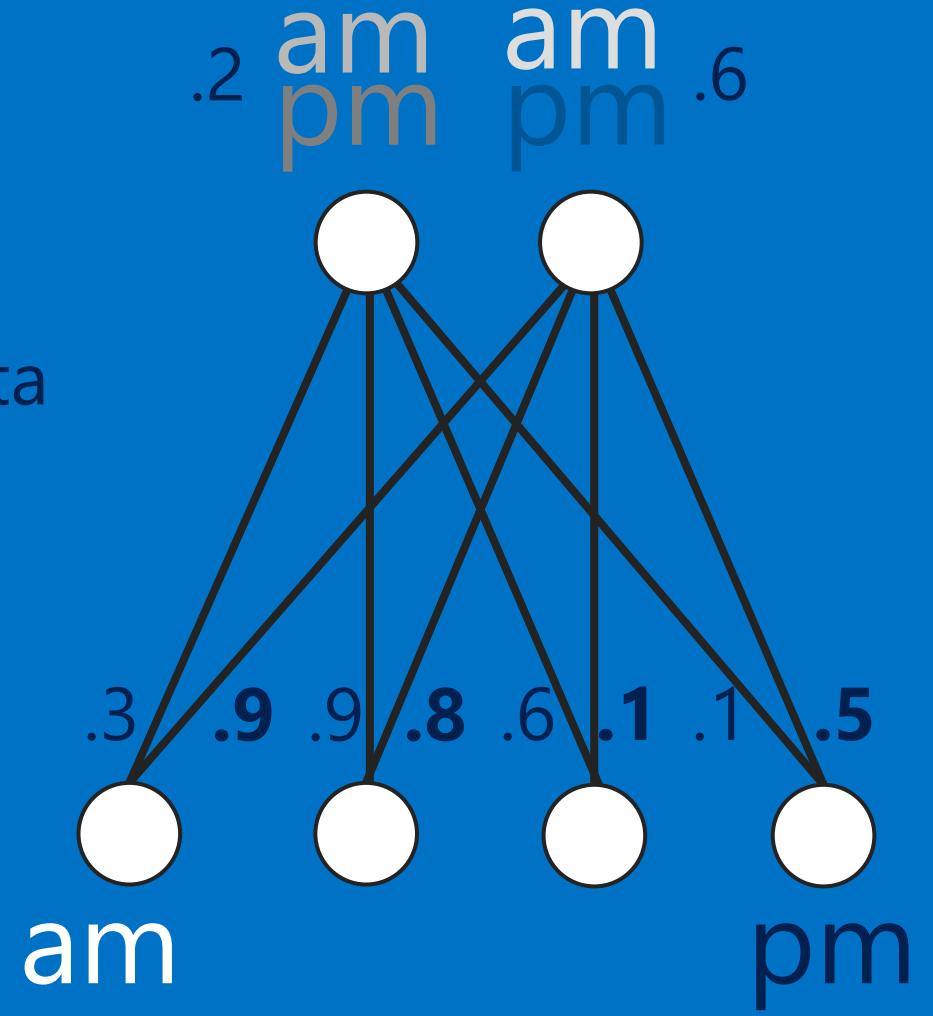


# Backpropagation

Adjust all the weights

adjustment = error \* gradient \* delta

am  
pm



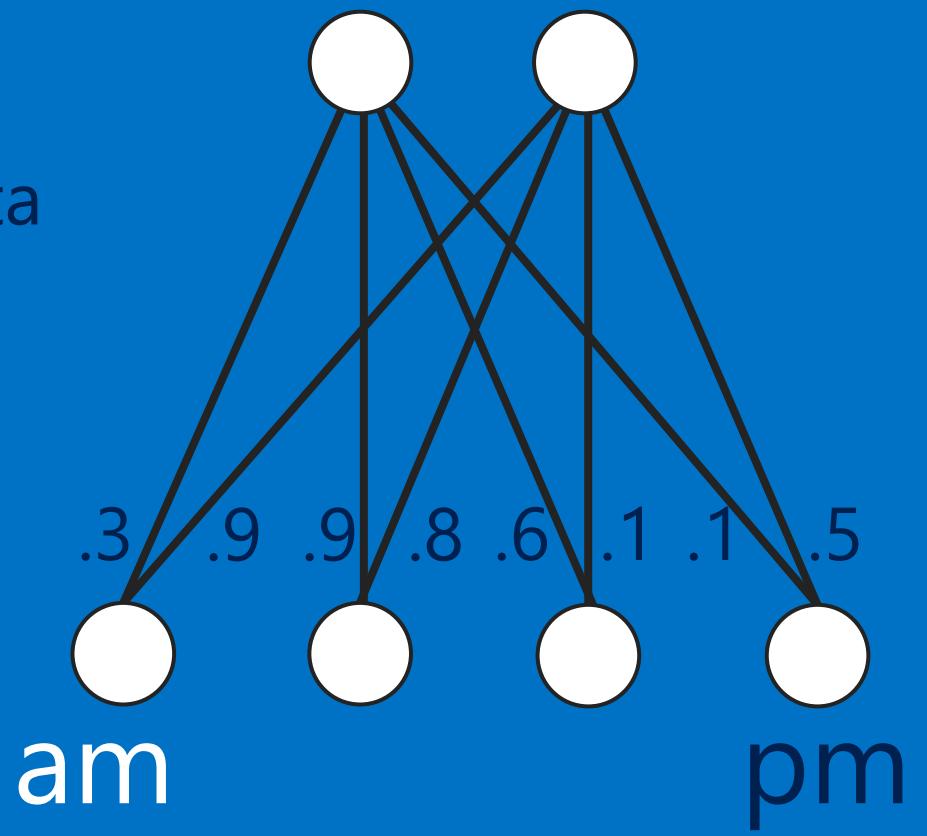
# Backpropagation

Adjust the weights

adjustment = error \* gradient \* delta

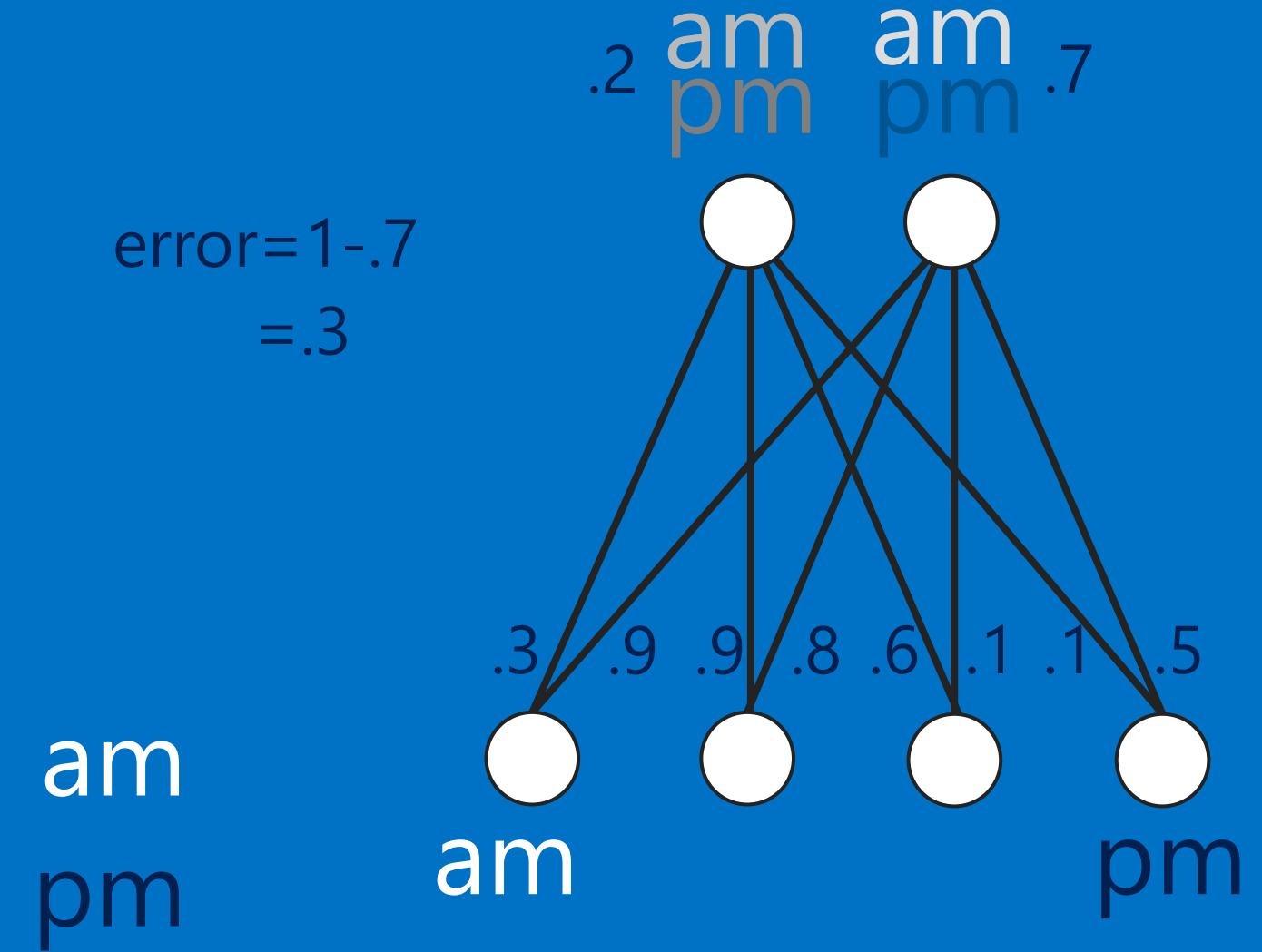
am  
pm

.2 am  
pm  $(.9 + .5)/2 = .7$



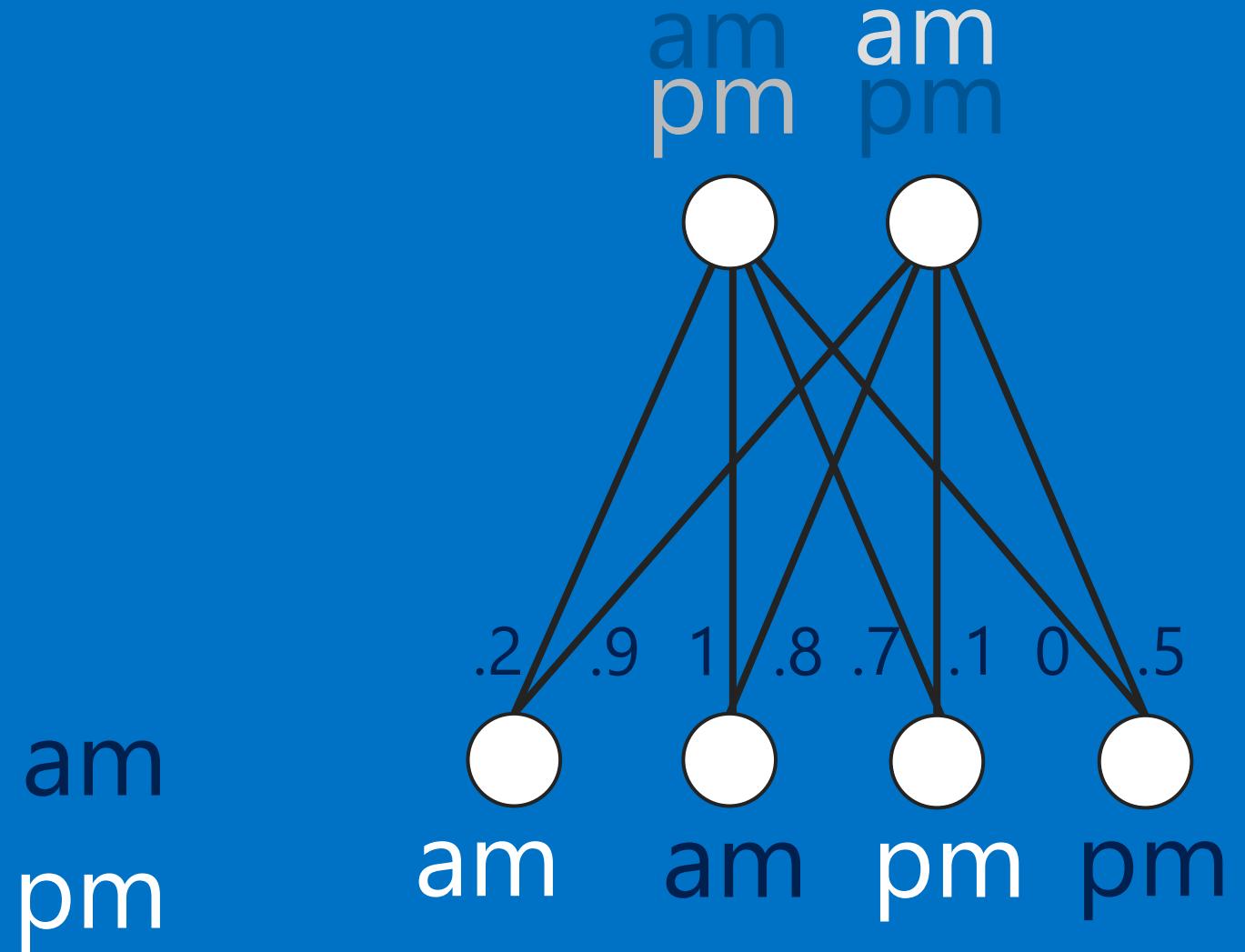
# Backpropagation

The error is now a little lower.



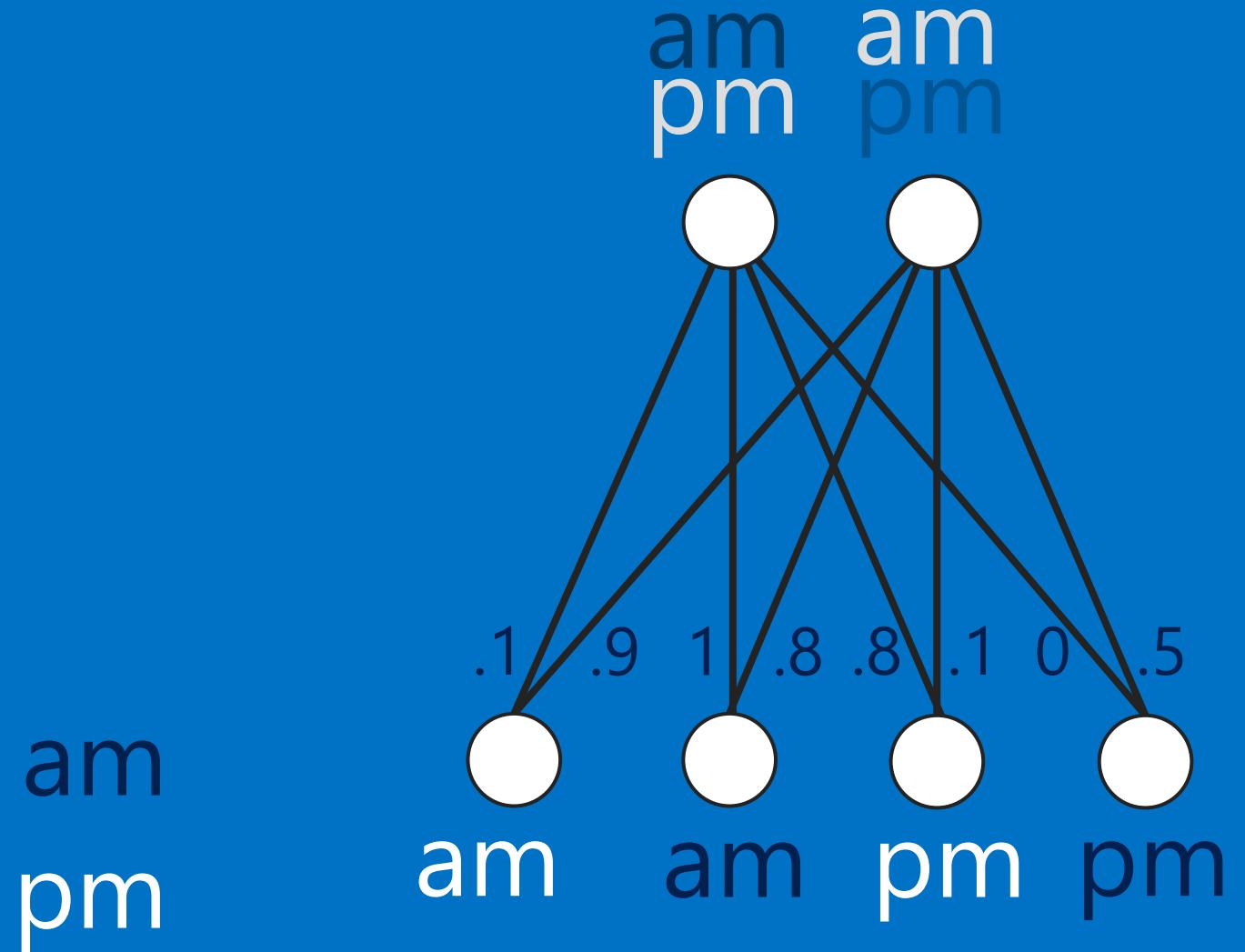
# Iterate

Repeat this process  
until the weights stop  
changing.



# Iterate

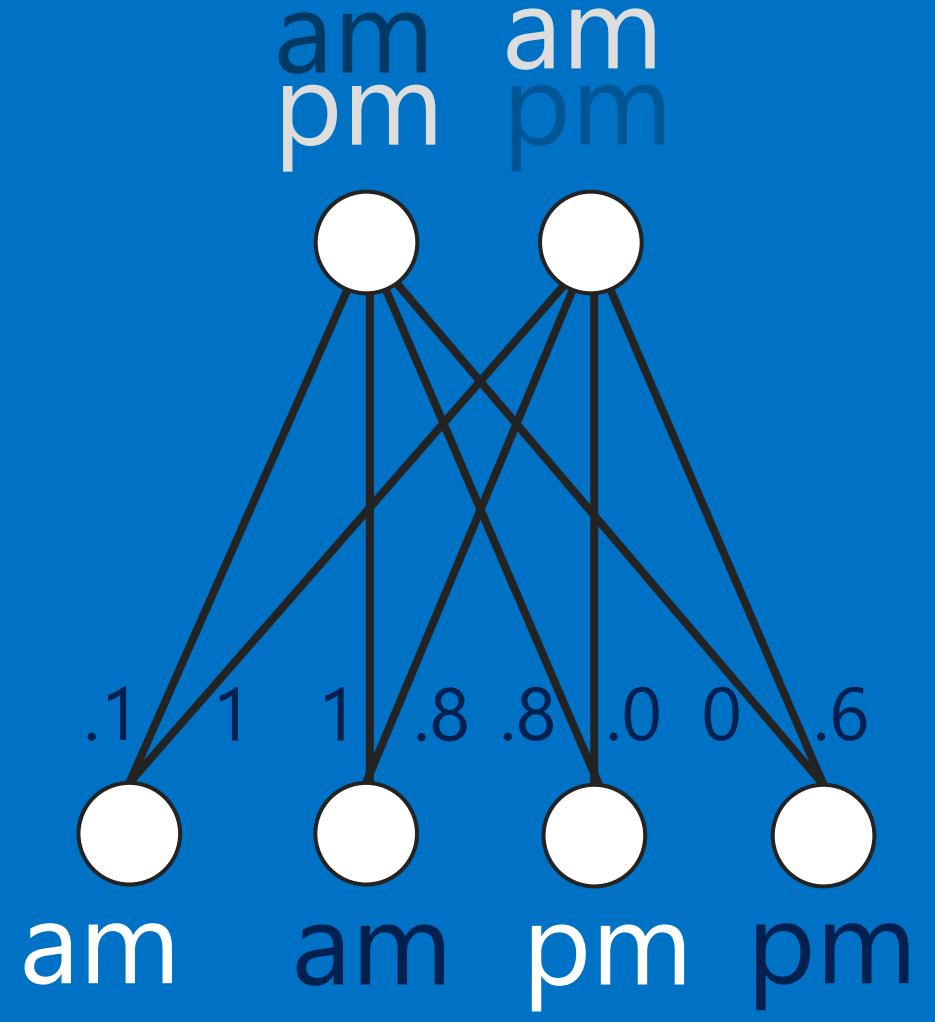
Repeat this process  
until the weights stop  
changing.



# Iterate

Repeat this process  
until the weights stop  
changing.

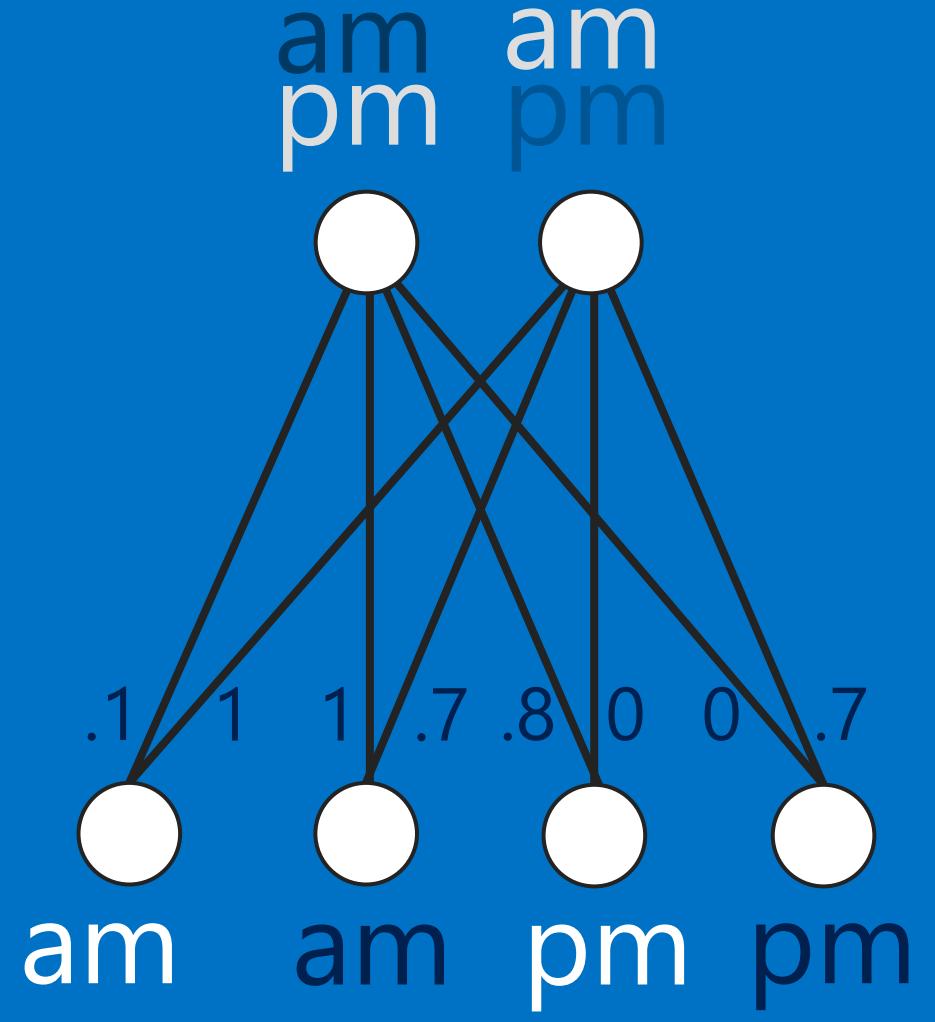
am  
pm



# Iterate

Repeat this process  
until the weights stop  
changing.

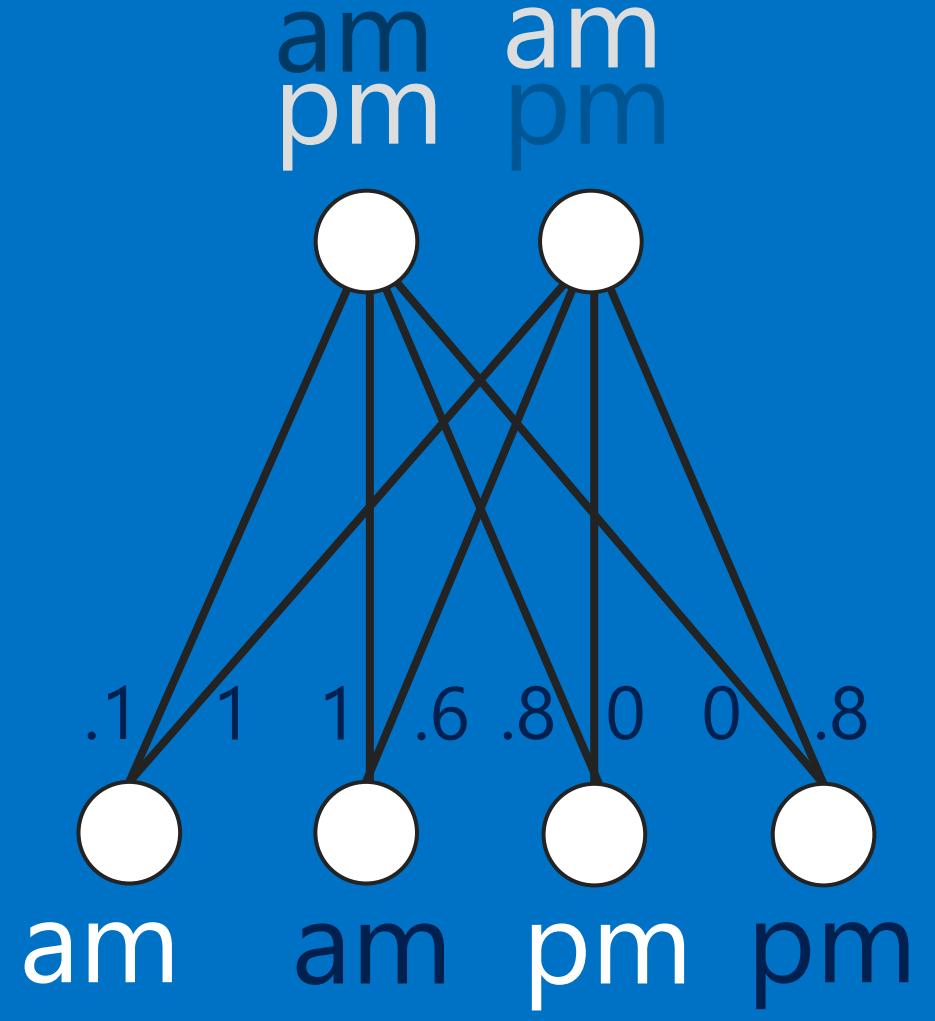
am  
pm



# Iterate

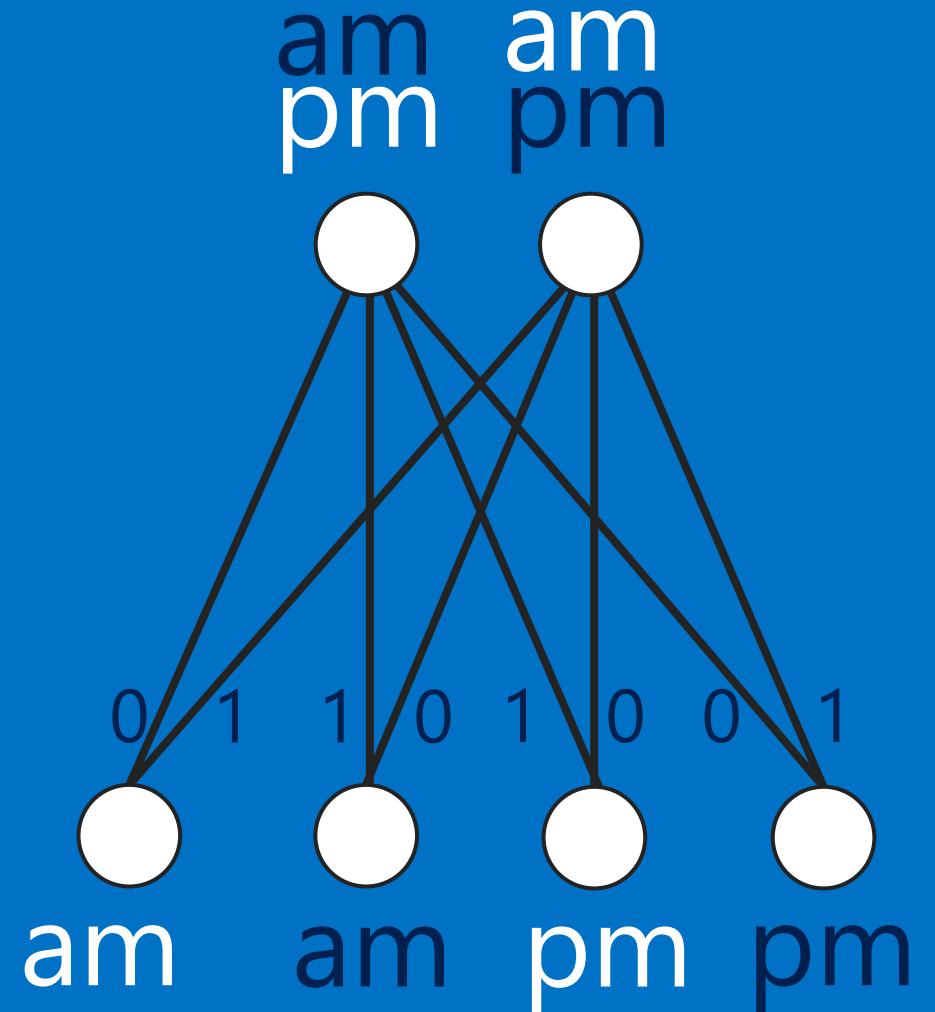
Repeat this process  
until the weights stop  
changing.

am  
pm



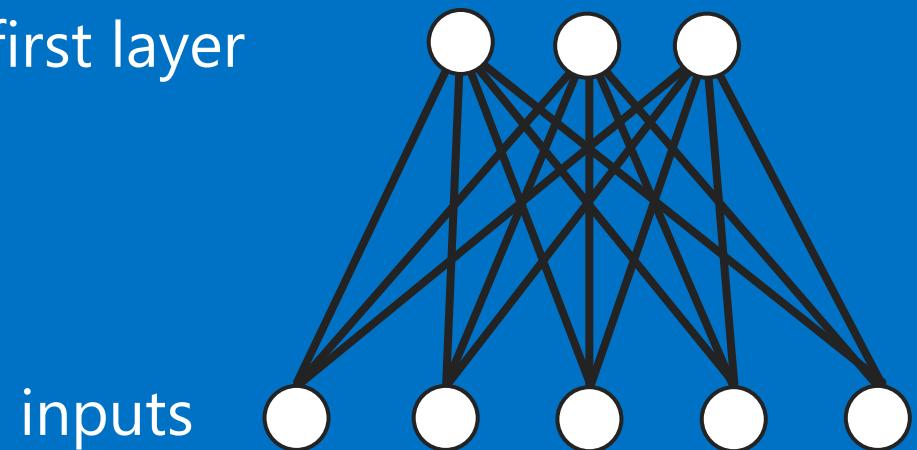
# Iterate

Repeat this process  
until the weights stop  
changing.



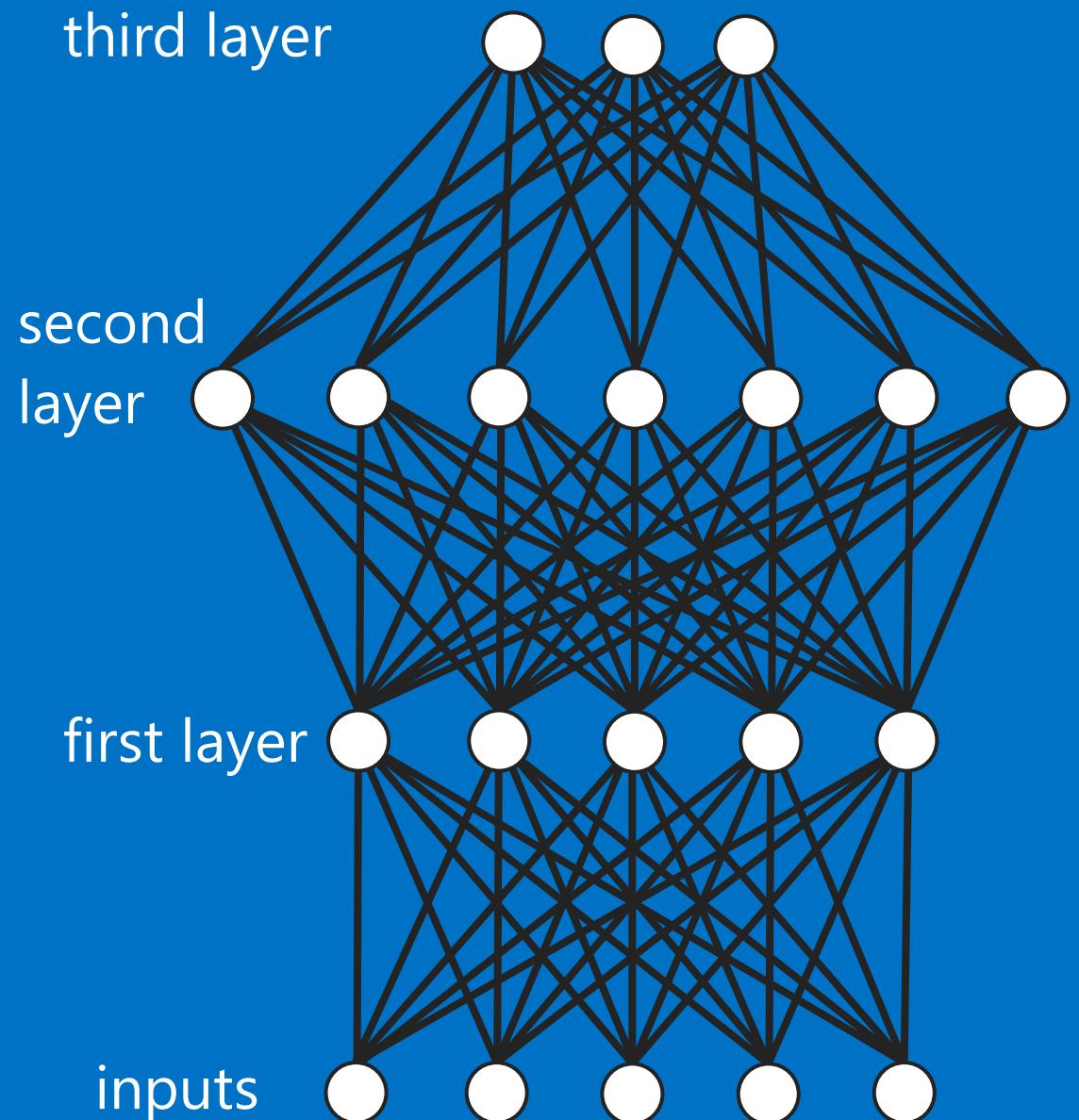
Each node represents a pattern, a combination of the neurons on the previous layer.

first layer



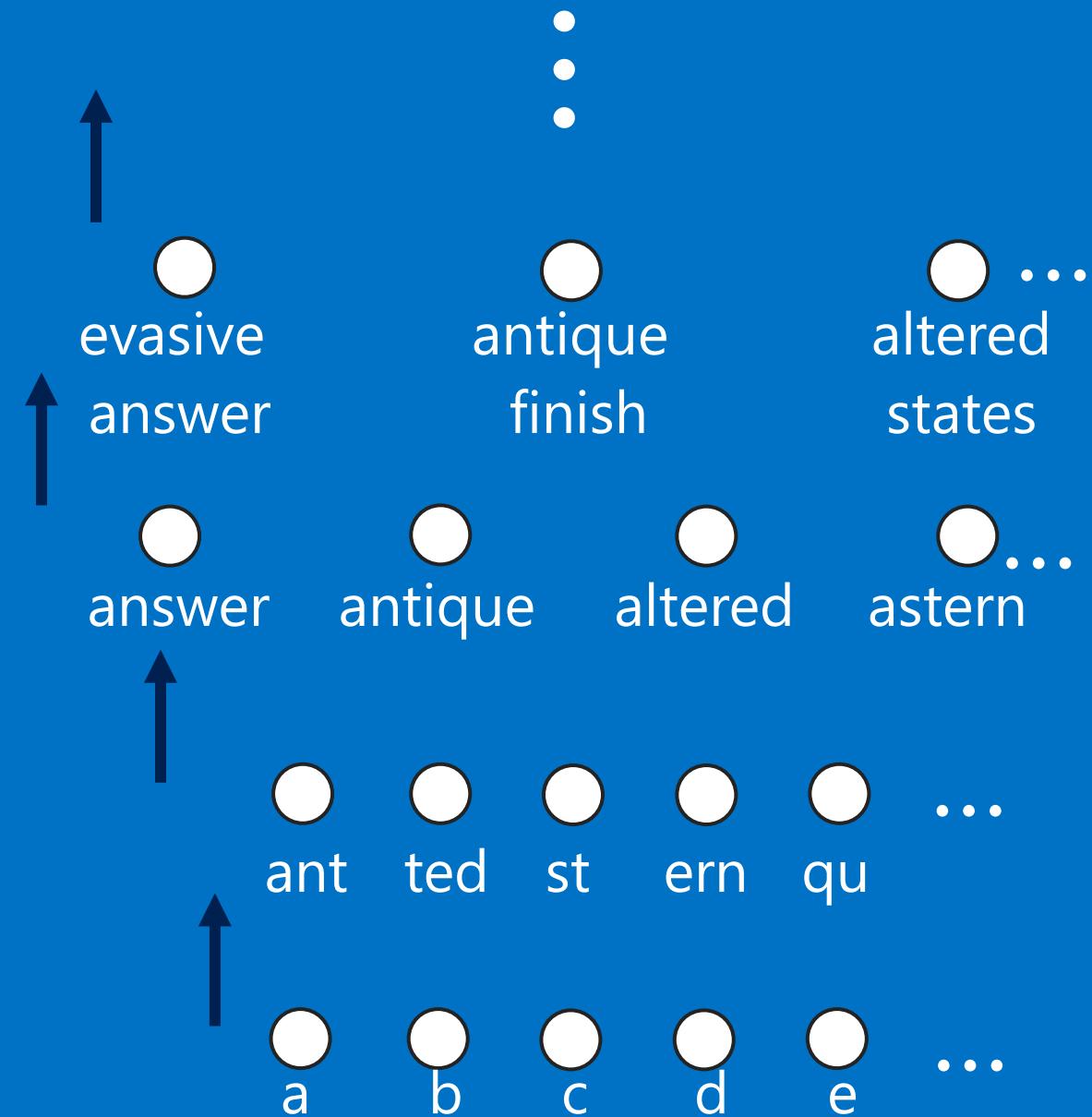
# Deep network

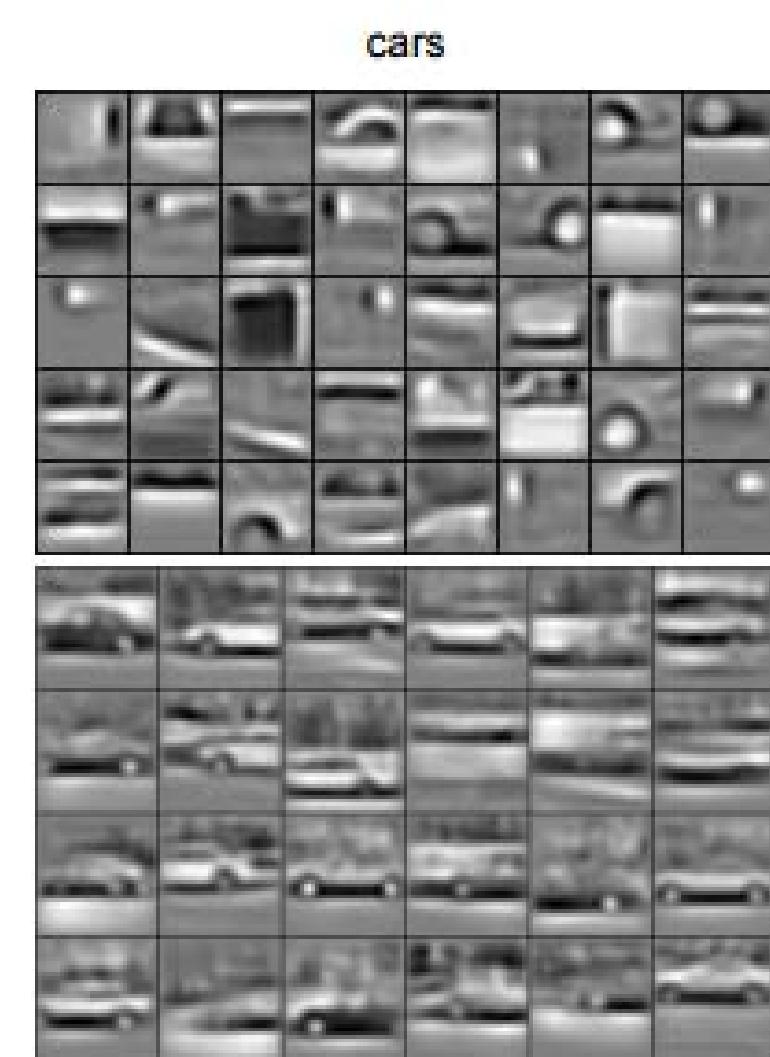
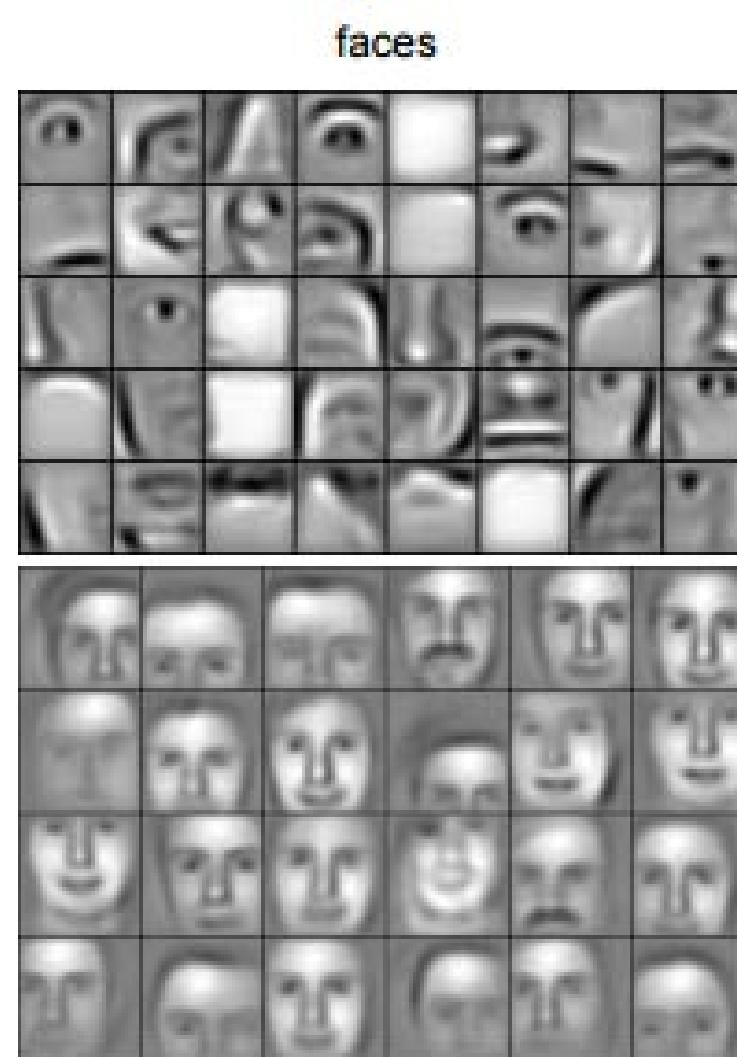
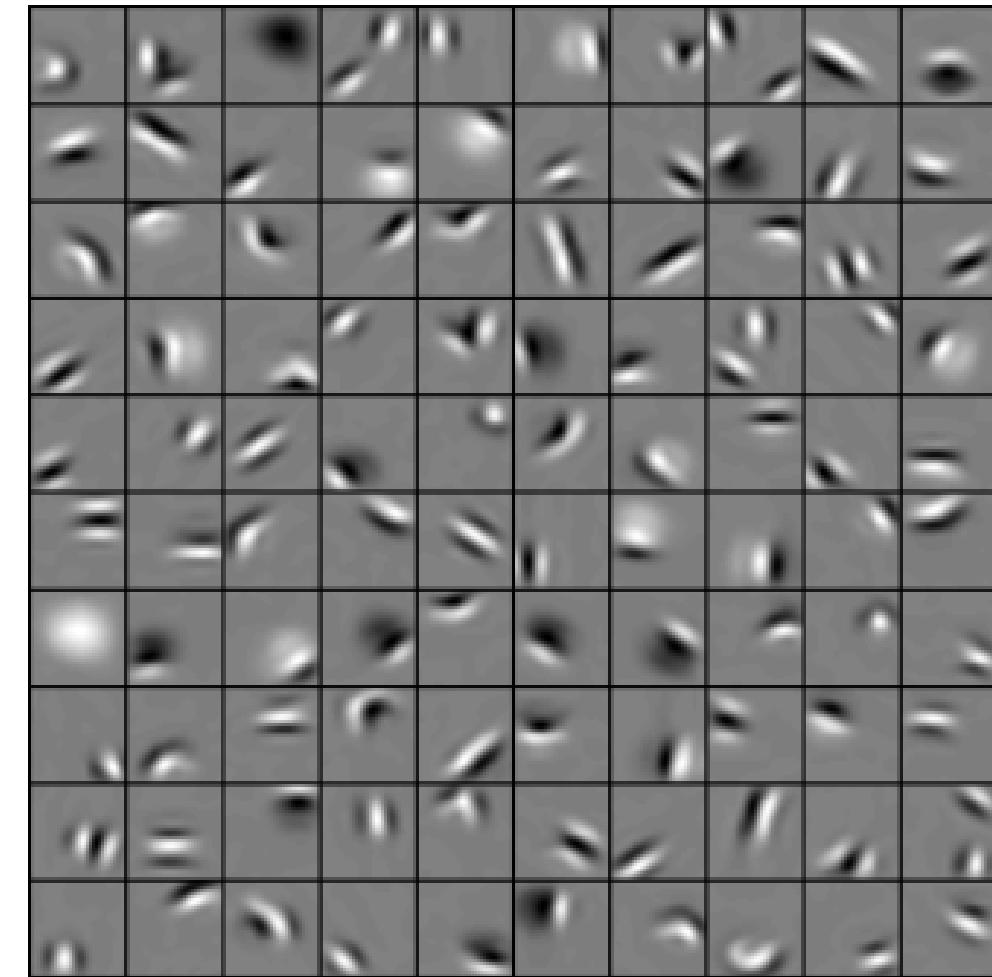
If a network has more than three layers, it's deep.  
Some 12 or more.



# Deep network

Patterns get more complex  
as the number of layers  
goes up.



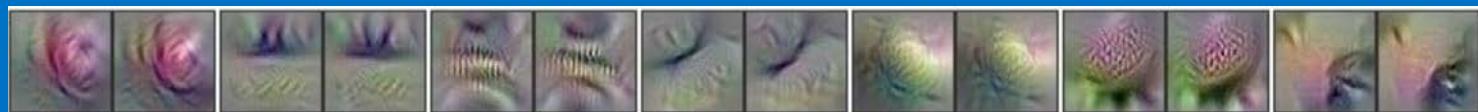
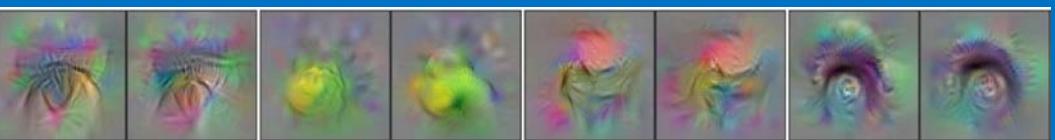
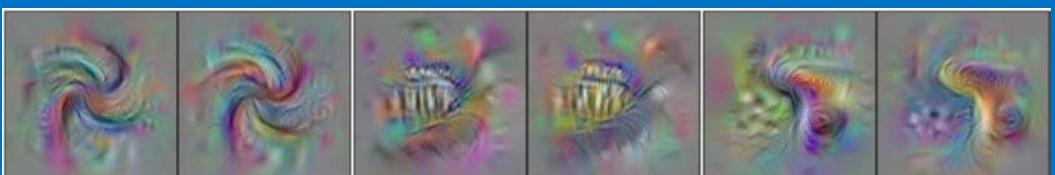
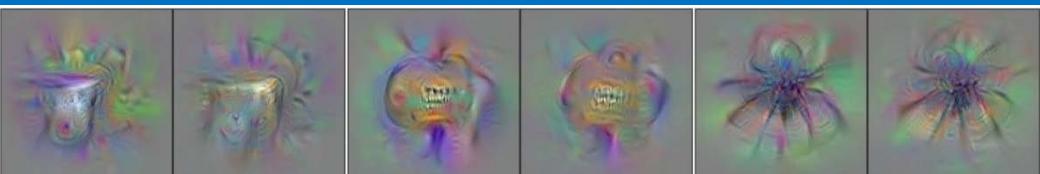


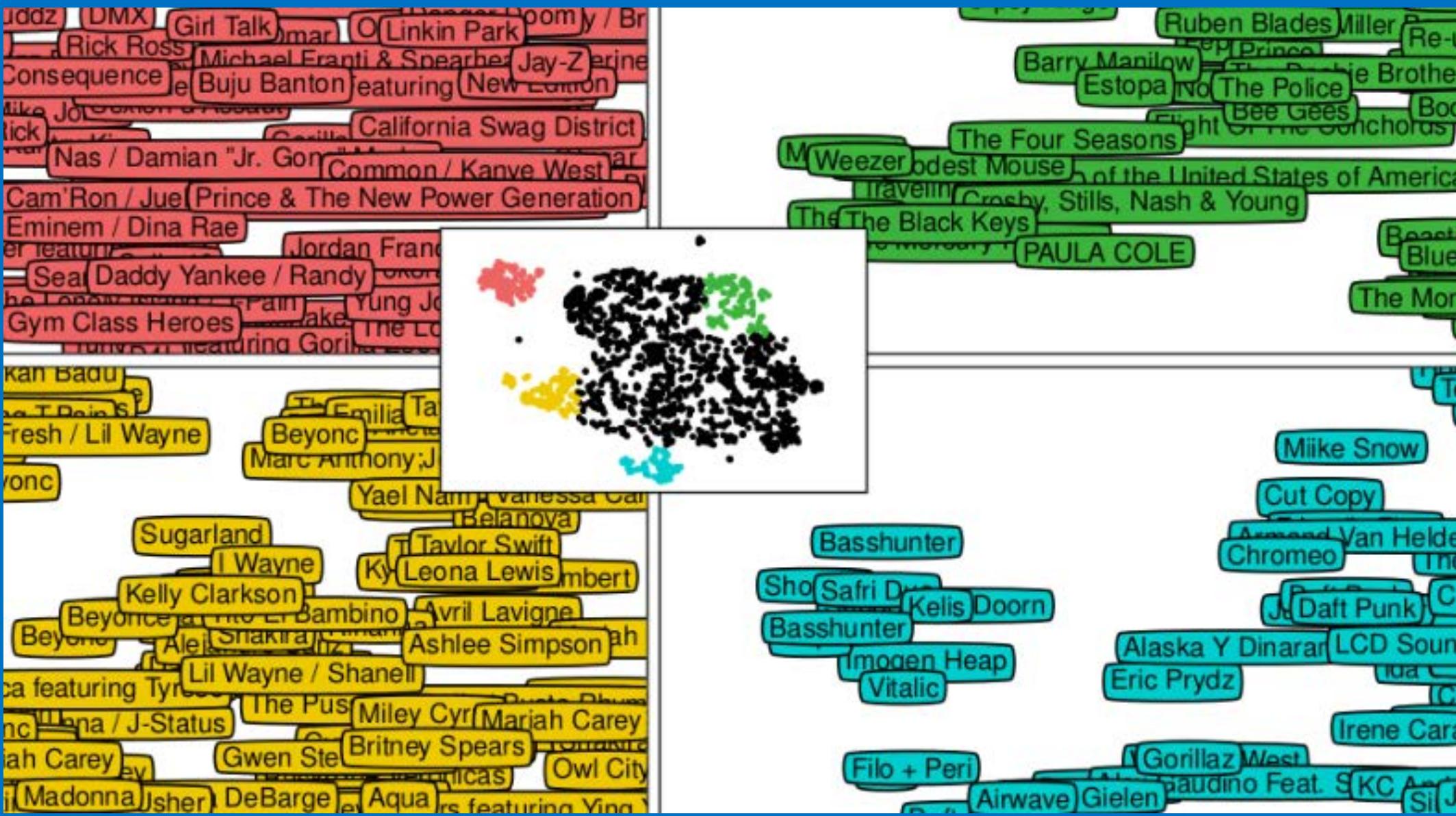
Convolutional Deep Belief Networks for Scalable  
Unsupervised Learning of Hierarchical Representations  
Honglak Lee, Roger Grosse, Rajesh Ranganath,  
Andrew Y. Ng

# Understanding Neural Networks

Through Deep Visualization.

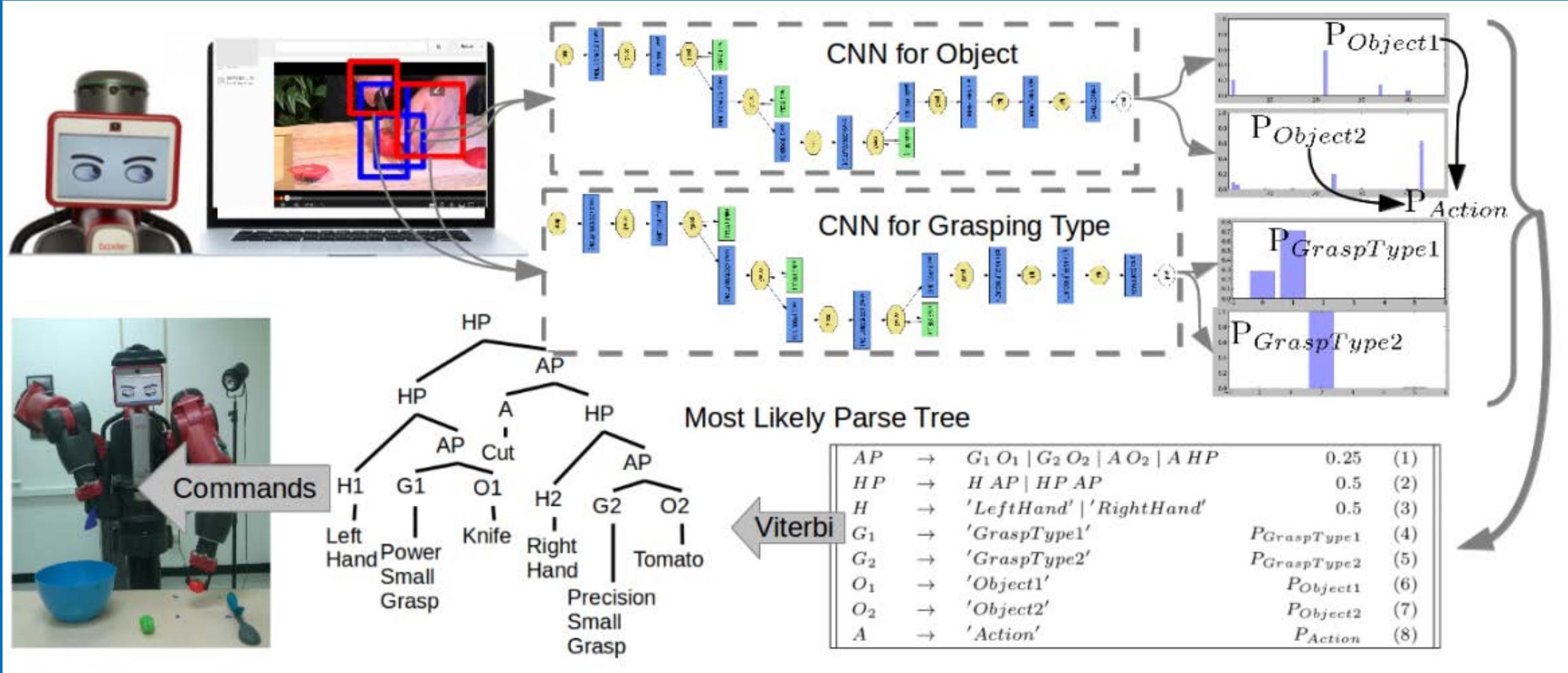
Jason Yosinski, Jeff Clune, Anh Nguyen,  
Thomas Fuchs, Hod Lipson







Playing Atari with Deep Reinforcement Learning.  
Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves,  
Ioannis Antonoglou, Daan Wierstra, Martin Riedmiller



Robot Learning Manipulation Action Plans by "Watching" Unconstrained Videos from the World Wide Web.

Yezhou Yang, Cornelia Fermüller, Yiannis Aloimonos

# Deep learning BINGO

Convolutional Neural Network

Deep Belief Network

Restricted Boltzmann Machine

Deep Reinforcement Learning

Deep Q Learning

Hierarchical Temporal Memory

Stacked Denoising Autoencoders

Bottom line

Deep learning is good at learning patterns

# Thanks for listening!

Thoughts to share? Questions?

Connect with me offline:

Brandon Rohrer on LinkedIn

@\_brohrer\_ on Twitter

[brohrer@microsoft.com](mailto:brohrer@microsoft.com)



Also, check out the writings of  
Christopher Olah

