

Machine Learning for Human Vision and Language

Lecture 2: **(Deep) Learning and processing in biological neurons and networks**

Ben Harvey

1

In today's class, we will move away from artificial networks for machine learning to look at the structures that inspired them: networks of biological neurons learning about the world to guide our behaviour.

This is quite different from the content covered in our last class: it is the most important topic in biology for AI, but we know many of you have no background in biology.

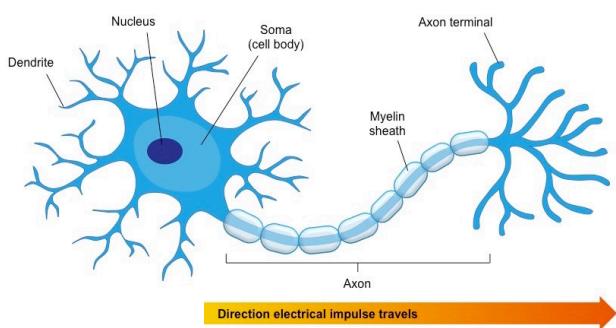
For those with CS backgrounds, who may have found the last class basic knowledge, the biological content in the next 45 minutes is probably the most challenging aspect of the course. So...

Why study biological networks?

- Inspiration for many AI networks, including DCNNs
- AI networks simplify processes involved for efficiency
 - AI deep networks not yet as advanced as human brain
 - Where the goal is to imitate human behaviour, following the neural mechanisms more closely may help
- Major link between AI and other sciences
 - Links AI to biological sciences, not only math/CS
 - Potential to link social sciences to biological sciences
- DCNNs are leading model of neural computations

2

A biological neuron



3

A neuron is a specialised type of animal cell. Most aspects work like other cells and are not important for performing computations. But the cell membrane, that is the outer surface, of neurons is specialised for performing simple computations using electrical activity.

First, the dendrites integrate electrical signals coming in from other neurons that this neuron is connected to.

These connections, or synapses, between dendrites and other neurons vary in strength, and change strength depending on past activity. This may sound familiar from the last class.

Changes in the strength of these synapse connections allows learning: changes in how the neuron's responds to its inputs as a result of its experience of previous inputs. So, the strengths of these synaptic connections are the biological equivalent of

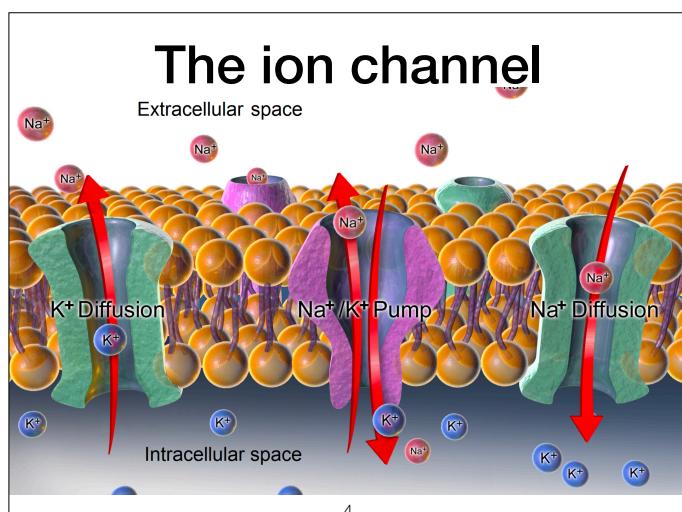
the weights in a filter kernel in artificial neural network.

We will see that the biological equivalent of a convolutional filter is a tree of dendrites working together to sample and integrate inputs from other neurons.

If these combined incoming electrical signals are strong enough to pass a THRESHOLD, the neuron will then send an electrical impulse down the axon.

On reaching the axon terminals, this is passed on to the dendrites of other cells, in the next LAYER of neurons.

So, let's look more closely at this unusual cell membrane.



The basic computational component of a biological neuron is a protein called an ion channel, which sits in the cell membrane.

-A protein is a large complex molecule, which can change shape under certain conditions.

-An ion channel can be in an open shape that allows charged atoms, or ions, of sodium and potassium to pass into and out of a neuron,

-Sodium and potassium are called natrium and kalium in Dutch and German, which are more like their chemical symbols.

-These ion channels they are normally in a closed shape, blocking these ions from moving through.

-The membrane that the channels are embedded in also does not allow ions to pass through.

-Normally-when the cell is not active-the cell membrane has an imbalance of ions on inside and out: far more sodium ions outside the cell (top), a few more potassium ions inside (bottom).

-This imbalance of electrically-charged ions causes a voltage difference across the cell membrane.

-But the ion channel protein can change shape and open, allowing a particular type of ion to pass through: there are different ion channels for sodium and potassium ions.

-When these ions cross the cell membrane, this changes the membrane voltage, also called the membrane potential.

-Note that the ion channel is a passive mechanism, it does not pump the ions from areas of low concentration to areas of high concentration. It only stays closed to keep the concentration unbalanced, or opens to allow ions to diffuse towards an equal concentration.

-This is like pulling a plug out of a bath tub, allowing the water to flow out as quickly as the hole allows.

-This is very fast, much faster than pumping ions around to change voltages. That would be like pumping water in to or out of the bath, much slower.

-But the ion imbalance must be maintained by an active mechanism (requiring energy). The bath must be filled before the plug hole can let water through.

-This pump is another protein imbedded in the cell membrane, which is always actively moving sodium ions out of the cell and potassium ions into the cell to create the imbalance.

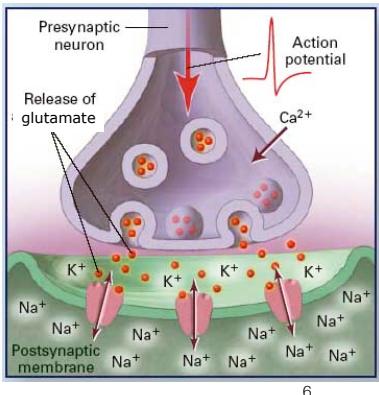
So...

- The cell membranes of neurons are specialised for electrical computations.
- Ions are atoms with electrical charges
- Cell membranes do not allow ions through
- Sodium/potassium pumps establish an imbalance of ions across the membrane
 - Much more sodium outside
 - A little more potassium inside
- Opening ion channels allows these ions to pass through, in the opposite direction, towards balance

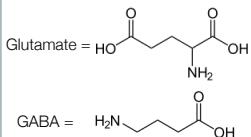
5

If you are having trouble with understanding this, there is a great introductory video here:
<https://www.youtube.com/watch?v=oa6rvUJlg7o>
This explains the basics really slowly.
Then come back to studying this class.

Opening the ion channel



6



So opening sodium ion channels causes sodium ions to enter the cell through the cell membrane, changing the membrane voltage or membrane potential.

-This may lead the cell to activate (or 'fire'), and the amount of this activation is what is simulated by the activation values in the feature maps of an artificial convolutional neural network.

-For today's class, there are two important ways an ion channel can open.

-First, it can open because a neurotransmitter binds to the ion channel. In that case, the ion channel is a 'receptor' for the neurotransmitter.

-When the neurotransmitter binds, the ion channel protein changes shape, opening the ion channel for ions to cross the membrane.

-This neurotransmitter is released by another neuron firing. The two neurons are separated by a small gap called a synapse, so the neuron releasing the neurotransmitter is called the presynaptic neuron (top), and the neuron that the neurotransmitter binds to is called the postsynaptic neuron (bottom).

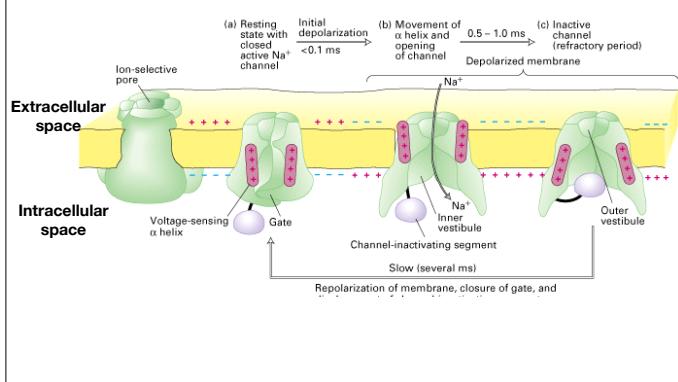
-So the neurotransmitter is a signal released by activation of a neuron in one layer of a neural network (the presynaptic neuron at the top) and causes activation of a neuron the next layer (the postsynaptic neuron)

-Here we will look at excitation of the postsynaptic neuron by binding of the neurotransmitter glutamate, and inhibition by binding of gamma-amino butyric acid (GABA). Excitation makes the neuron more likely to become active, while inhibition makes the neuron less likely to become active.

-Using both excitation and inhibition allows connections with both positive and negative weights, as convolutional filters have.

-Other neurotransmission works by similar mechanisms, but glutamate and GABA are by far the most common neurotransmitters.

Opening the ion channel



The second way an ion channel can open is due to a change in membrane voltage. This is called a voltage-gated ion channel.

-Here, the voltage across the membrane has been changed by other ion channels opening and allowing electrically charged ions through.

-This change in membrane voltage changes the shape of the voltage-gated ion channel, opening it and allowing MORE ions to pass and the membrane voltage to change MORE.

-After a lot of ions have passed and the voltage has changed a lot, the ion channel is closed in a different way: this inner segment moves into the channel's gate.

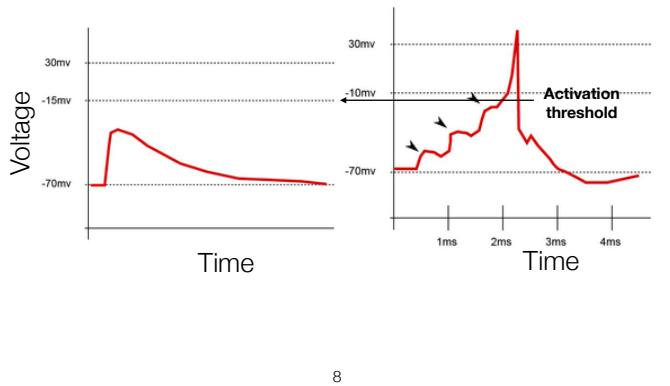
-Then, over a longer time, pumping of sodium and potassium returns the membrane voltage to its resting potential, the ion channel returns to its original closed state, and this process can begin again.

The voltage-gated ion channel is particularly important because opening the ion channel also changes the membrane voltage.

-So changing membrane voltage can lead to further changes in membrane voltage.

-In this way, the voltage-gated ion channel acts very much like an electronic transistor, that is an electrical switch that opens and closes because of an electrical input. Miniaturised transistor circuits are the basis of computer processors.

Excitatory post-synaptic potentials (EPSPs)



-When glutamate binds to a post-synaptic receptor, it first causes the receptor's ion channel to open, and sodium to enter the cell, as we have seen.

-This causes a change in the membrane voltage, called an excitatory post-synaptic potential.

-The ions become more balanced across the cell membrane. The size of the voltage difference across the cell membrane decreases, so we say the cell membrane is depolarised (less negative).

-But over time, this depolarisation decreases as sodium and potassium are pumped around to return the membrane to its resting potential (-70mV), the original imbalance of ions across the membrane.

-So this one neurotransmitter molecule binding causes no activity on the postsynaptic neuron, because it doesn't reach a threshold that activates the second step: the voltage-gated ion channels. Then, the neuron doesn't pass this event on by firing, and it has no further effect.

-However, if several neurotransmitters molecules bind over a short period, this can reach the **threshold voltage** for the voltage-gated sodium channels.

-This causes a much more extreme depolarization, a spike of voltage called an action potential, firing or spike. This firing will be passed down the axon to release neurotransmitters that bind to the next layer of

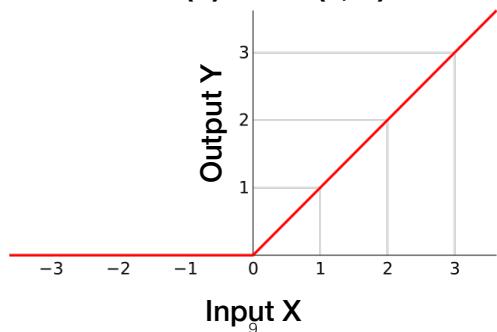
neurons.

-So this threshold for activation of voltage-gated sodium channels is the biological equivalent of the threshold/rectification operation we saw in artificial networks.

The threshold/rectification operation

an activation function using a rectified linear unit (ReLU)

$$Y = f(X) = \max(0, X)$$



-Until we reach a certain threshold voltage, there will be no response.

-This threshold is not zero in biological neurons, it's around -55mV because an imbalance of ions is maintained at rest to speed up neuron responses.

-Then the firing rate (output) of the neuron will increase as the strength of the inputs increases.

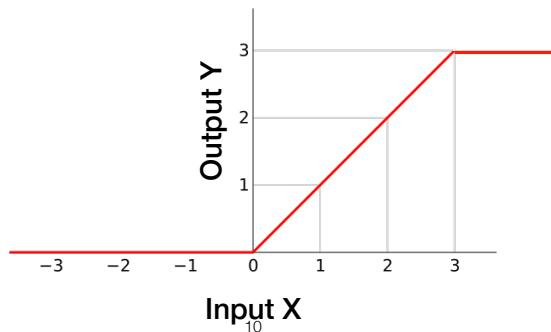
-The response of a neuron may be considered a binary event: the neuron either fires or not.

-But the rate at which the neuron fires will increase as the strength of its inputs increases (either through stronger synapse or more presynaptic activity). This firing rate is a continuous property of neural activity, not simply binary.

The threshold/rectification operation

an activation function using a rectified linear unit (ReLU)

$$Y = f(X) = \min(\max(0, X), \text{MaxActivation})$$

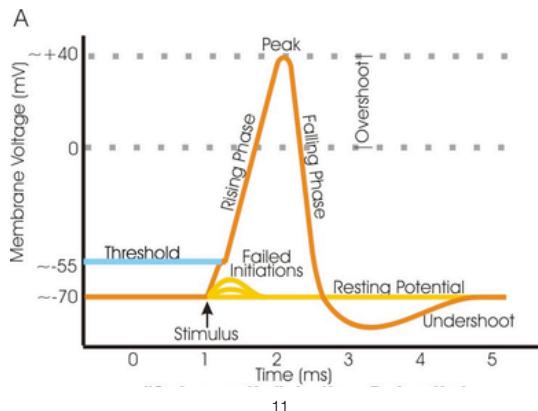


And at some point this output will reach a maximum firing rate.

This maximum is often not implemented in artificial neural networks.

Biological neurons rarely reach their maximum firing rate, so it isn't usually necessary to include. It's more important when we aim to simulate biological activity more closely.

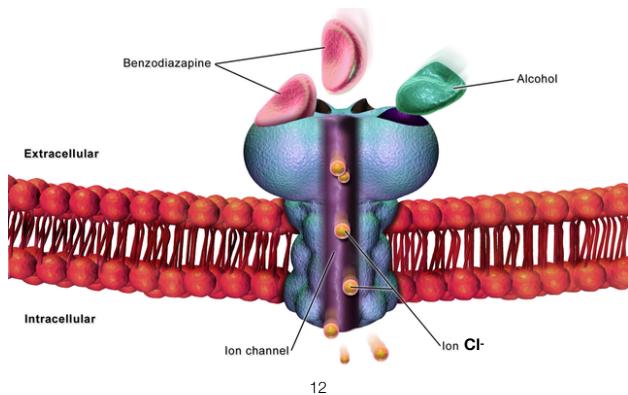
Action potentials



The reason for this maximum firing rate is that an action potential takes time, although only a few milliseconds.

- At the peak, Na^+ concentration is almost equal across the cell membrane
- The high voltage then causes the Na^+ channels to close. Again, this is a different closed state than before, as the channel protein is a different shape.
- Here, the Na^+ channel cannot open again until the cell is repolarised (more negative).
- The initial repolarisation (falling phase) relies on K^+ channels opening, which repolarises faster than pumping, but the Na^+ concentration is too high inside for further diffusion through ion channels, and the K^+ concentration is too low inside. So opening the Na^+ channels again will do nothing. The Na^+/K^+ pump needs to return the cell to resting potential.

Inhibitory post-synaptic potentials (IPSPs)



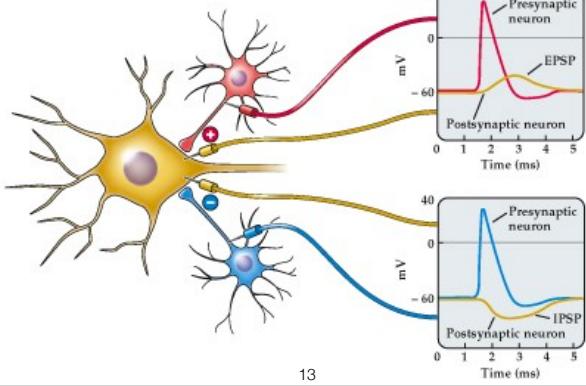
Activity in a presynaptic neuron can also inhibit the postsynaptic cell firing, by causing the membrane to polarise further, making the membrane voltage more negative. We call this state **hyperpolarized**: the membrane potential difference is greater, specifically **more negative**, than its usual resting polarised state.

-The GABA receptor is also an ion channel, but for negative chloride (Cl^-) ions. When GABA binds, Cl^- enters and the cell membrane potential becomes more negative than the resting potential.

-This pushes the cell further away from activating.

-Alcohol and benzodiazepine drugs like valium also bind to the GABA receptor, reducing neural firing.

Inhibitory post-synaptic potentials (IPSPs)

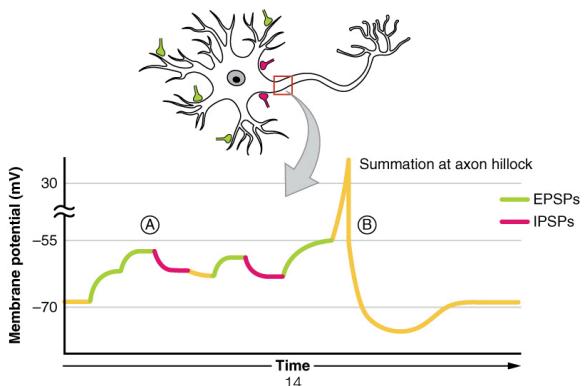


So we have both positive and negative inputs onto the same postsynaptic neuron.

If we have two presynaptic neurons (red and blue), when the red one is active (red line) it makes the postsynaptic neuron (yellow line in the top graph) depolarised through excitatory neurotransmitters like glutamate.

When the blue one is active in the same way (blue line) it makes postsynaptic neuron hyperpolarised (yellow line in the lower graph) through inhibitory neurotransmitters like GABA.

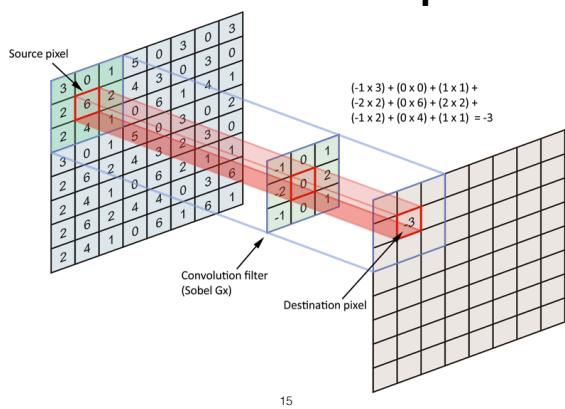
Inhibitory post-synaptic potentials (IPSPs)



These inputs can be from a range of different places, and from a number of different neurons.

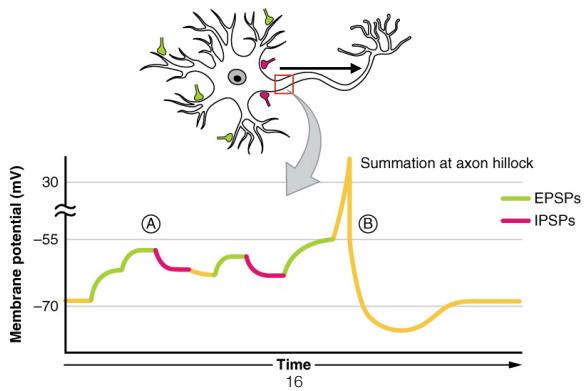
However, there is a limited spatial distribution of inputs because the dendritic tree has a limited size. Which may sound very familiar...

The filter/convolve operation

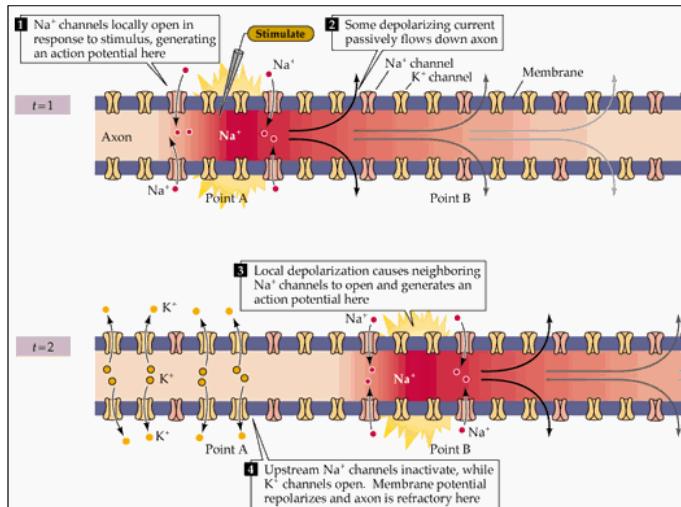


Because convolutional network filters imitate this structure: they have positive and negative weights (convolutional filter). These are multiplied by the (always positive) activity in a spatially limited group of units in the previous layer (source input) to determine the activity of a single unit in the next layer (destination output).

Action potentials

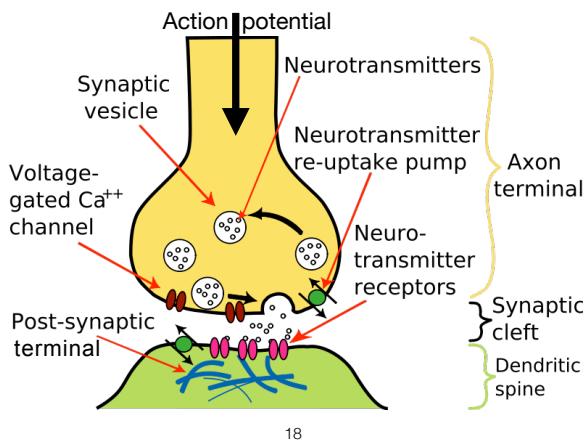


Once an action potential has been triggered, it travels from the input end of the neuron, the postsynaptic dendrites, along the length of the the neuron's main fibre, the axon, to provide inputs to the next layer, which is often in a different brain area some distance away.



- This process of carrying the neural response down the axon also relies on voltage-gated sodium channels.
- The depolarisation at Point A spreads to neighbouring locations to push their membrane potential above threshold. This causes voltage gated ion channels at this neighbouring location to open, leading to Na^+ entry and depolarisation at point B. This opens further voltage gated ion channels even further along (further right), causing Na^+ enter here.
- So the depolarisation spreads down the axon like a wire.
- Note this can only go in one direction. After the action potential wave has passed, the membrane will be depolarised and there won't be a suitable concentration difference in Na^+ ions across the membrane. Also the ion channels will be closed by their inner segment first, rather than having their gate close by the resting membrane potential.

Neurotransmitter release



18

- On reaching the presynaptic terminal for the synapse to the next layer of processing, the action potential leads to neurotransmitter release.
- At rest, neurotransmitters are stored ready for use, in membrane bubbles called vesicles.
- The arriving action potential causes voltage gated calcium ions (Ca^{++}) channels to open, allowing calcium ions to flow into the cell.
- Ca^{++} binds to proteins in the axon terminal to change their shape, bringing the vesicle bubbles to the synaptic surface and releasing the neurotransmitter to activate the next postsynaptic neuron.

Changing membrane potential (voltage), firing neuron

- Neurotransmitters (e.g. glutamate and GABA) released from a pre-synaptic cell can excite (depolarise, EPSP) or inhibit (hyperpolarise, IPSP) activity in the post-synaptic neuron
 - This relies on ligand-gated (i.e. neurotransmitter activated) ion channels
- If membrane polarisation reaches a threshold, voltage-gated ion channels open
 - Results from many excitatory inputs
 - Strongly depolarises (fires) the neuron: Action potential
- Those opens voltage-gated sodium ion channels in nearby sections of the membrane
 - Causes voltage to change in these sections too
 - Causing further voltage-gated ion channels to open in further locations
 - A chain reaction travels down the long axon
- At the far end of the axon, voltage-gated calcium ion channels open
 - Calcium binds to web of cellular proteins, changing their shape
 - Draws vesicles (bubbles) of stored neurotransmitters to pre-synaptic surface
 - Released neurotransmitters bind to NEXT neuron, leading to EPSPs and IPSPs there

19

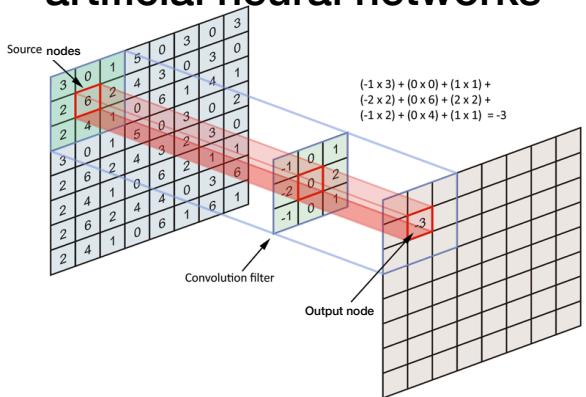
At end:

So, now we have seen how both artificial neural network nodes and biological neurons combine the responses from multiple inputs to determine their activity.

In both cases we have seen that the resulting activity provides the input to the next stage of processing.

In both cases, the responses of the input nodes or neurons affect the current node or neuron by different amounts: the weight or strength of connections between neurons differs...

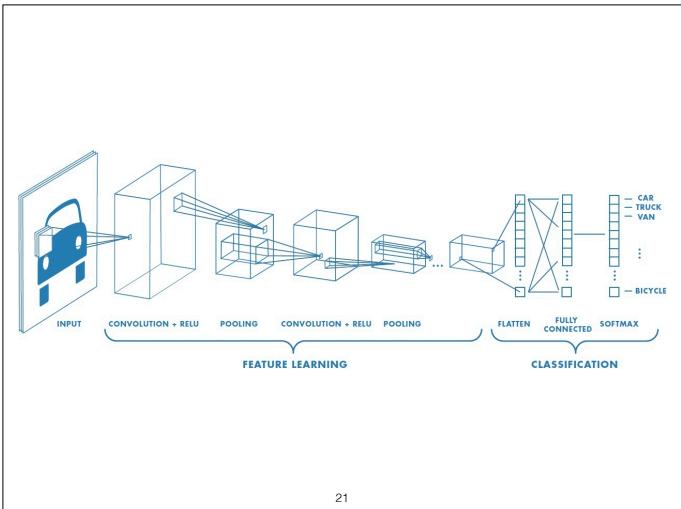
Backpropagation of error in artificial neural networks



Like we see here for an artificial network. The response in the output node is affected by different extents by the activity of different source nodes.

Responses in some source nodes increase the response of the output node, because the convolutional filter weights by which these source nodes' responses are multiplied are positive (right side of this filter).

Where the convolutional filter values are negative, responses in the corresponding source nodes will inhibit (or reduce) the response in the output node. And where the weights are zero, and response in the source nodes will have no effect.



21

Something here I skipped in the last class.
The last stage of the network, after several convolution layers, uses the activity pattern after the final convolutional layer for classification.

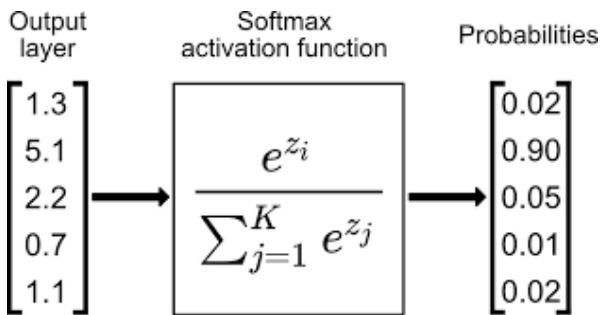
To compare this activity pattern with previously-seen patterns, the spatial relationships are first discarded, ‘flattening’ the last feature map into a line of independent units.

Each of these is connected to units that represent the possible candidate classifications, the labels that describe the input image.

Here we use a fully-connected layer to link every unit to every possible classification with different learned weights.

Essentially this tests, for each possible classification, how much the top-layer patterns the network was trained on is similar to the current top-layer response pattern.

The softmax operation



22

So, the weights through our network will transform each input image into some ‘score’, reflecting the match between the top layer’s pattern of activation and the pattern of activation by previous examples of each category.

This SCORE must then be converted to a PROBABILITY that this input image falls into each category.

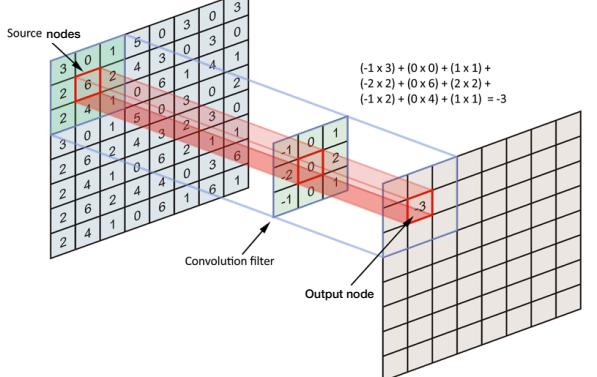
This should take into account not just the score for one category, but also the scores for all other categories: the relative scores determine the probabilities.

This is almost always done with the normalised exponential function, or ‘softmax’.

That is, a constant e raised to the power of the score, divided by the sum of this exponent over all classification scores. As a result, the probabilities sum up to one.

The math is not particularly important to know, but note that, following an exponential function, an output layer score that is only slightly higher than another leads to a probability that is MUCH higher.

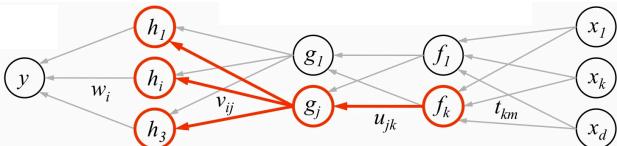
Backpropagation of error in artificial neural networks



The magnitude of the response in the source nodes also affects the output node. But this magnitude of the source node's response also depends on the weights in convolutional filters in the previous processing stage.

So, if we want to change the responses of the network, we need to change the weights in the convolutional filter. Changing these weights to produce the desired responses is the basis of machine learning in deep networks. And importantly, if we change the weights at one processing stage this changes the activity in the source nodes for the NEXT processing stage.

Feedforward processing

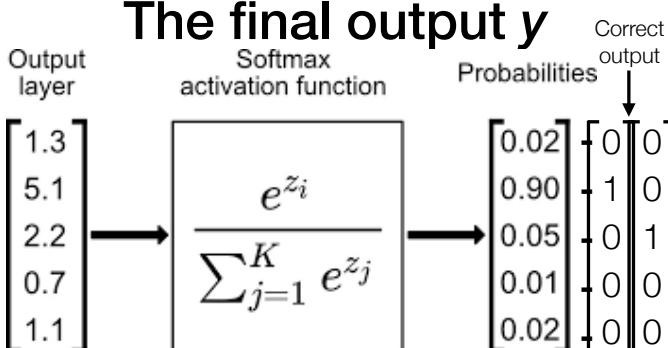


- Inputs x multiplied by corresponding weights in t and summed to give responses f
- Responses f multiplied by corresponding weights in u and summed to give responses g
- Responses g multiplied by corresponding weights in v and summed to give responses h
- Responses h multiplied by corresponding weights in w and summed to give final output y

So, to understand how we need to change these weights to get the output we want from our network, let's first consider how the output of a very simple network is computed.
(GO THROUGH STEPS)

We know the output pattern that the network currently produces, y , and the output we want if our network has correctly learned the wanted input-output mapping (y^*).

The final output y



$$\text{RMS} = \sqrt{\text{mean}(0.02^2+0.1^2+0.05^2+0.01^2+0.02^2)} = 0.052$$

$$\text{RMS} = \sqrt{\text{mean}(0.02^2+0.9^2+0.95^2+0.01^2+0.02^2)} = 0.584$$

This output is in the form of the probability of each possible output, in the case of a classification task.

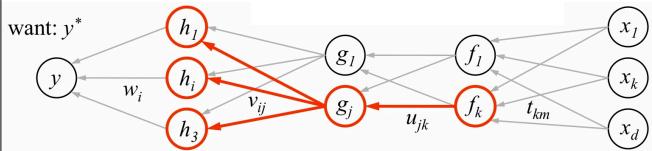
We also have the ideal output, where the probability is one for the correct classification.

We then use some 'cost function' to decide how we relate the difference between the seen and expected outputs to how much we change the network.

In this case we will use the root mean square difference, or RMS, a common cost function.

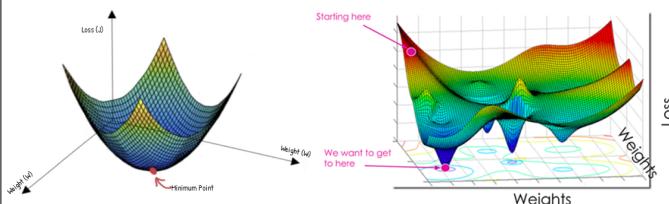
We want to minimise this cost function. This is the error that will be used for backpropagation.

Backpropagation



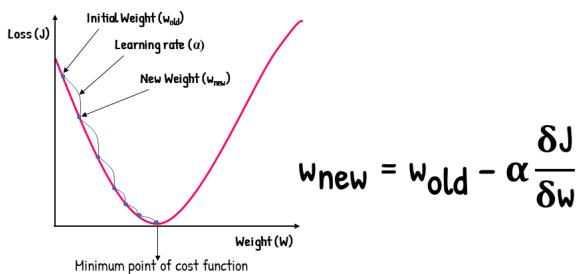
- Responses h multiplied by corresponding weights in w and summed to give final output y
- How should the weights in w change to minimise the cost function?**
 - (Minimise the difference between y and y^*)

Gradient descent



- The error (value of the cost function) can be calculated from the weights
 - But the relationship is a complex function
 - And there are too many possible weights to try all of them (hundreds of weights in each layer, infinite values)

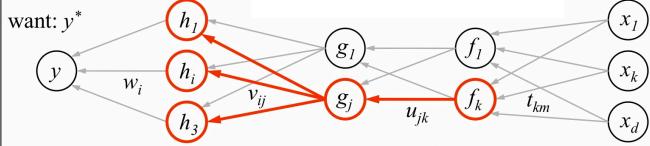
Gradient descent



- We know the current value of the cost function, and the equations that gave this value
- We can take the derivative of the equations for the cost function to determine whether increasing or decreasing each weight would decrease the loss
 - This gives the direction that each weight should change
- The step size is given by the learning rate (which we set)
 - Too small rate \rightarrow takes too many iterations
 - Too large rate \rightarrow may jump over the minimum

As a nonlinear function of the inputs, there is generally some state of the inputs that produces a minimum cost. (This would not be true for linear inputs).

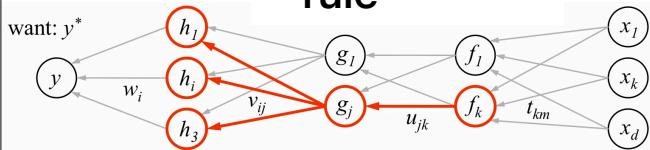
Backpropagation



- Responses h multiplied by corresponding weights in w and summed to give final output y
- Responses g multiplied by corresponding weights in v and summed to give responses h
- Responses f multiplied by corresponding weights in u and summed to give responses g
- Inputs x multiplied by corresponding weights in t and summed to give responses f
- How should the weights in v , u and t change to minimise the cost function?**
 - This changes the responses h (and g and f) to minimise the difference between y and y^*

So, see here that there is a chain of effects working through the network

Backpropagation: The chain rule



$$\frac{dh}{dx} = \frac{dh}{du} \cdot \frac{du}{dx} \quad t_{\text{new}} = t_{\text{old}} - \alpha \frac{\delta J}{\delta t}$$

A useful feature of derivatives is that they can also be linked together in a chain.

If the effect on a final result depends on two steps that have a factor in common, the resulting equation simplifies to jump over the effect of that factor.

So, when we have already changed the weights in w to minimise the cost function, then we can calculate the derivative of the weights in v with respect to y to see which direction v must change to minimise the cost function j , considering the new values in w .

And when we have done that, we can calculate the derivative of the weights in u with respect to y to see which direction u must change to minimise the cost function j .

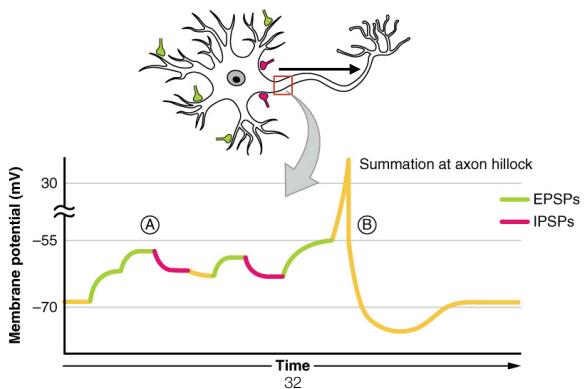
This is all exceptionally computationally intensive. It is why training a deep neural network on a large data set is so expensive, compared to predicting an output for a new input, which is far less computationally intensive.

Backpropagation

- The responses in each layer depend on the weights linking each node to the previous layer
 - And the responses in the previous layer, which in turn depend on weights linking that layer to the last
- We compare the network's output to the ideal output state (i.e. the labels for each input)
- The difference is used in a cost function, describing how wrong the output is and so how much the weights should change (i.e. loss)
- Using derivatives of the equations for the cost function, backpropagation determines which direction each weight should change to reduce the loss
 - This direction is multiplied by the learning rate to determine how much the weights should change
- Weights at earlier layers also affect the cost function
 - The chain rule is used to calculate their derivatives, and so which direction the weights at earlier layers should change

31

Variable connection strengths between neurons



32

In biological neurons, just like in a convolution operation, different synaptic inputs (coloured ends here) have different effects on the response of the neuron they are linked to. These inputs again differ in the strength of the synapse (synaptic weight) and also the strength of the response in the presynaptic neurons.

So here we see that the different EPSPs and IPSPs are of different strengths. Though it is much harder here to measure the responses and synaptic strengths of all of these neurons, so it is harder to tell whether a stronger EPSP results from a stronger synapse or stronger responses in the presynaptic neuron.

But again here, assuming the same input, any change in the response of the presynaptic neuron must result from changes in synaptic strengths at earlier stages.

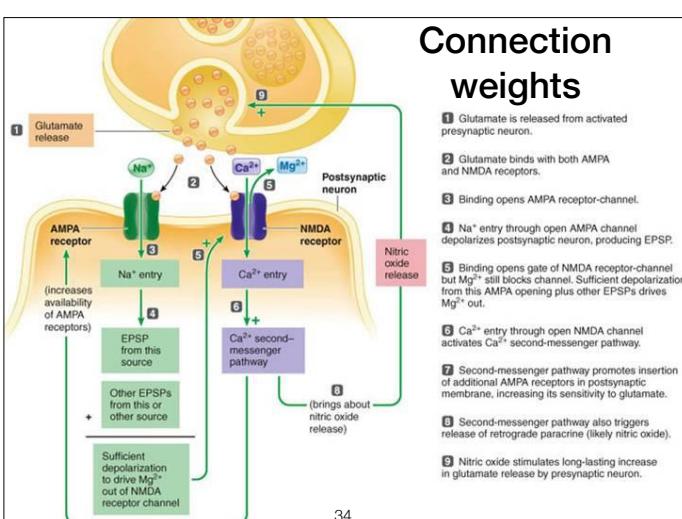
So again here, biological learning results from changes in synaptic strengths.

Weights in biological neurons

- Connection weights in biological systems don't depend on backpropagation of error
 - We are not trained by a supervised process
 - There is no 'correct' response in how we learn about the world
 - However, there can be adaptive and maladaptive responses/ behaviours
- Instead, we learn mainly by unsupervised processes
 - To recognise patterns of activity we have seen before
 - To learn the statistics of the world we live in
- Long-term potentiation
 - Lasting enhancement of synaptic connections by co-activation of presynaptic and postsynaptic neurons
 - Hebb's postulate: 'Cells that fire together, wire together'

33

- The Skinner box is an apparatus used in animal experiments to deliver rewards (food) and punishments (electric shocks through the floor) when an animal performs (or fails to perform) certain behaviours. This trains the animal and changes its behaviour.
- Conditioned responses like these those learned from rewards and punishments are unlikely to work through backpropagation-like mechanisms affecting the whole network.
- They are more likely to work at a single decision stage, where the final decision is associated with a reward or fear response. This is enough to modify behaviour, there is no need to modify the animal's entire visual processing: the rat doesn't see the lights differently, only its decisions change.



34

So, how does firing together lead to wiring together?

- Postsynaptic neural activity modifies synapse structure at both the presynaptic and postsynaptic terminal through a variety of mechanisms.

- You should know that these processes increase presynaptic neurotransmitter concentration and release, and the density of (active) postsynaptic receptors for the neurotransmitter.

You do not need to know how that happens, but briefly, here are some example mechanisms.

AT END: Another important mechanism is an increase in the transcription of receptor genes in the cell nucleus, as cell polarisation affects the activity of gene transcription enzymes.

- Mechanisms like this seem very important in learning and memory, and seem to occur at all synapses.

- Several common drugs interfere with these processes. Alcohol and benzodiazepines like valium reduce spiking activity by activating GABA receptors and reducing the probability of depolarisation. This reduces memory formation, so these sedative drugs have a tendency to cause amnesia.

-Ketamine blocks the NMDA glutamate receptor, a vital step in strengthening weights. Ketamine is therefore very effective at causing amnesia.

The story so far

- Filter: integration of EPSPs and IPSPs across the dendrite tree
- Threshold: activation of voltage-gated Na^+ channels by above-threshold depolarisation by EPSPs
- Pool: ?
- Normalise: ?
- Network layers: layers of neurons after different levels of synapses
- Feature map: ?
- Learning mechanism:
 - Unsupervised: 'cells that fire together, wire together' (Hebbian learning)
 - Activity-dependent changes in synapse structure/strength (weights)

35

AT END:

The feature maps and pooling and normalisation functions rely not just on one neuron, but analysing relationship among many neurons.

After the break, we will start looking at how larger groups of neurons represent information and interact. This will allow us to think about how these are implemented in biological systems.

The story so far

- Filter: integration of EPSPs and IPSPs across the dendrite tree
- Threshold: activation of voltage-gated Na^+ channels by above-threshold depolarisation by EPSPs
- Pool: ?
- Normalise: ?
- Network layers: layers of neurons after different levels of synapses
- Feature map: ?
- Learning mechanism:
 - Unsupervised: 'cells that fire together, wire together' (Hebbian learning)
 - Activity-dependent changes in synapse structure/strength (weights)

36

So the weights on filters result primarily from learning.

-However, some filters appear hard-wired in early stages of vision. These seem to arise from evolution rather than learning, suggesting they provide a useful structure that subsequent layers can build on.

-In artificial deep networks for vision, the filters before the first hidden layer can also be hard-wired as edge detectors, and otherwise edge detectors easily arise through learning.

-As we saw in the last class, neural network processing can be easier to understand by looking at specific examples of filters like this.

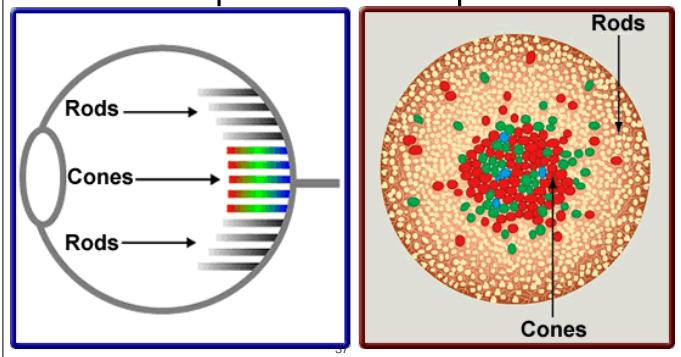
-Understanding these early filters gives an idea of which feature maps arise in biological vision, and how they are used by subsequent stages.

-Understanding this is important because normalisation depends on the structure of these feature maps.

-So, let's look at some biological feature maps, and how those extract features and normalise responses.

-All of these structures and operations don't rely on a single neuron, but a larger group of neurons. So now we will look at larger structures containing large numbers of neurons.

Input feature maps: The photoreceptors



The eye has four different sensors for light: three types of colour-sensitive cones sensor and one type of rod sensor.

-Rods don't carry colour information, but respond quickly and under low light conditions. Their output also gives good information about fast events like moving objects.

-Together, these form the input image to the visual processing network, and are essentially already four feature maps.

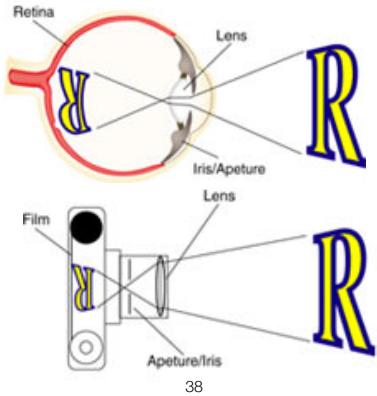
-Unlike in a camera that usually provides the input to an artificial network, these light sensors are not evenly distributed across the image.

-In central vision ('where we are looking') we only find cones, while further out we find mostly rods.

-The total density of photoreceptors drops off with the distance from central vision. So vision strongly over-emphasises the centre, where we are looking.

-This greatly reduces the computational load on the brain, while still giving high detail in central vision.

Image inversion



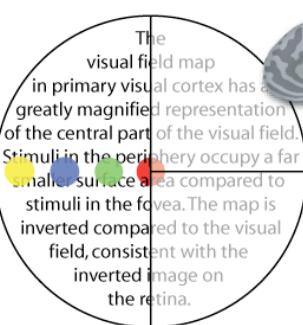
38

The brain's representation of visual space is also flipped up-down and left-right.

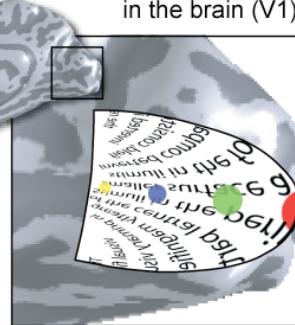
-This begins in the retinal projection of the image because rays of light are passing through a small hole at the pupil, much like an image is inverted in a camera sensor.

Cortical visual field representation

Visual field



Visual field representation in the brain (V1)



39

This over-representation of central vision and this image inversion continue into the cortex, where the visual image is 'mapped' onto the cortical surface.

-You may think that this strange representation of vision would cause problems for perception.

-However, the location of information on the retina or cortex is not important when we think of the visual cortex as a set of feature maps, neural representations of image transformations, rather than thinking of this as an image. As I mentioned last time, the activity in a feature map can be visualised into an image, but it is very misleading to think of a network layer's activity as an image.

Cortical map representations

- Spatially map the structure of the sensory organ onto the cortex
 - Maintain spatial relationships
 - Like DCNNs, leads to analysis of restricted patterns in local areas of the visual image with spatially-restricted dendritic trees
- Magnify the representation of important parts of sensory space
 - Central vision, hands and face, vocal auditory frequencies
 - Balances detailed analysis with computational load
- We can change what we sample in detail
 - Move our hands and eyes
 - Attention increases processing of attended parts
- Artificial networks for vision process the whole image in detail
 - Normally, a human has decided where to point the camera

-But it is very important that neighbouring locations are represented next to each other so that the spatial extent of a filter represents a continuous piece of the input image.

-This is particularly important when we consider that the dendrites of the next neuron will sample from this brain area, and have a limited extent too. This seems to be the main reason why the brain maintains spatial relationships at each level of processing: the mapping of the visual image onto the brain's surface is found throughout visual processing.

CLICK

-So we process central vision in more detail. Processing the whole image input in similar detail would be computationally intensive for the brain.

-Artificial sensory systems generally don't use such distorted inputs, aiming instead to process everything in great detail.

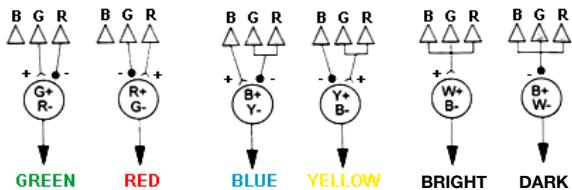
CLICK

-The main advantage of processing the whole image in the same detail is that the cameras that provide the inputs to artificial networks don't

need to move to sample the important parts of the image in detail, like our eyes do. Normally, the inputs are static images, so this isn't an option anyway.

-It will take considerable advances in artificial deep networks before they can process input images immediately, decide where the important details are, and move the sensors to sample those details.

Colour map comparison filters: colour opponency



41

Now let's look at the first filtering stages in the eye: how is incoming image first transformed? First there is a comparison over the different colour cones. The colour cones effectively form 3 feature maps in the input image.

-Comparing between them forms six feature maps in the next layer of processing, one each for greenness, redness, blueness, yellowness, brightness and darkness.

-This is very distinct from the cone activations, though this is not obvious for most students. For example, white light contains all colours and so it activates all cones strongly. But that light isn't coloured, it's white. So the activation of a single cones does not tell us the light's colour, the relative activations of different cones must be compared to determine colour.

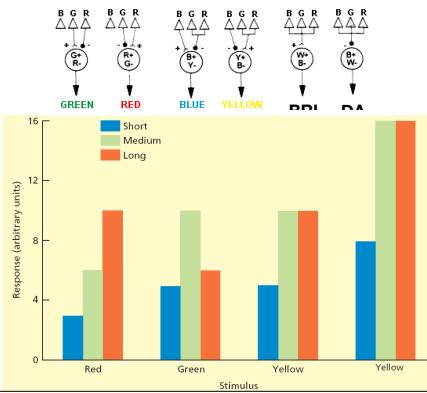
-There are relatively few blue cones, so redness and greenness are each determined by the difference between their intensity and the intensity of the other. Where both red and green are present at equal intensity, for example in white light, there is no response in these channels.

-Blueness is determined by the difference between the intensity of blue cone activation and the intensity of other cone's activations.

-Yellow light activates both green and red cones, but not blue. So yellowness is the average of red and green cone activations minus blue cone activation. Again, white light will produce no response.

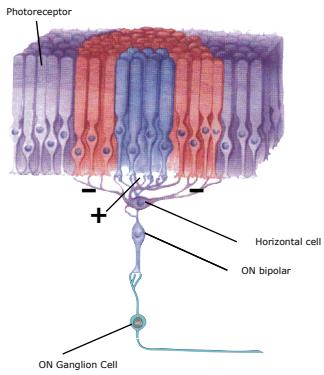
-To detect brightness and darkness, all three cones activate or inhibit the response in the output channel.

Colour map comparison filters: colour opponency



- This is approximately optimal for representing colour information
- Beyond the photoreceptor, we have three opponent channels for the **ratios** of R vs G, Y vs B and light vs dark
- Using these ratios, each individual colour representation gives an accurate description of the colour of light falling on the retina, not its intensity.
- To get this colour-based description, there would always need to be a stage to **COMPARE** the ratio of activation of the three cone types
- Doing the comparison early (in the retina) produces that representation as early as possible, so it can be used for all later processing.

Spatial comparison filters: surround suppression



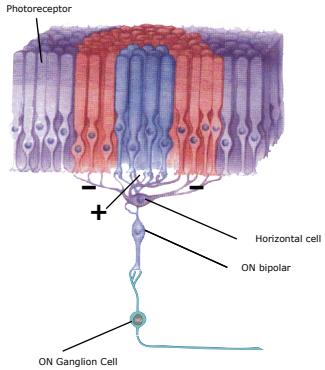
43

0	-1	-1	-1	-1	-1	0
-1	-1	-1	-1	-1	-1	-1
-1	-1	7	7	-1	-1	
-1	-1	7	7	-1	-1	
-1	-1	-1	-1	-1	-1	-1
0	-1	-1	-1	-1	-1	0

- As well as these colour comparisons, the retina makes spatial comparisons.
- The first stages begin analysing the relationships between nearby locations, comparing the responses of a group of neighbouring photoreceptor cells, all linked in to its tree of dendrites.
- Those in the centre of the tree produce strong EPSPs, while those over the whole tree produce weaker IPSPs at the output 'bipolar' cell because they activate the large, inhibitory horizontal cell. So activation from the centre is inhibited by activation over the surrounding region.
- As a result, a point of light at the centre produces a stronger response in this bipolar cell than a field of light covering both the centre and the surround. CLICK—Expressed as a convolutional filter, it might look something like this. If we have a bar of equal intensity (+1) running through the centre of the filter, we get a positive response (+20: $4*7+8*-1$). A point of light will at the centre only will produce a larger positive response (+28). A field of light everywhere will produce zero response ($4*7+28*-1$).

- Importantly, just like a convolutional filter, there are similar cells with overlapping connections working in parallel throughout the retina. So a photoreceptor that falls in the inhibitory surrounding zone of this one horizontal cell filter, also falls in the excitatory centre zone of another.

Spatial comparison filters: surround suppression



44

0	-1	-1	-1	-1	-1	0
-1	-1	-1	-1	-1	-1	-1
-1	-1	7	7	-1	-1	
-1	-1	7	7	-1	-1	
-1	-1	-1	-1	-1	-1	
0	-1	-1	-1	-1	-1	0

-This analysis effectively converts the amount of light at each location to the amount of contrast: how much the amount of light **changes** around this location.

-It represents contrast because a full field of light or darkness produces no response.

-Then, responses to most of the image, which contain no changes, can stop. This saves energy as neural responses take energy.

-But the CHANGES in the image are maintained. This keeps all the useful information, but greatly compresses it for transmission down the small optic nerve to the brain.

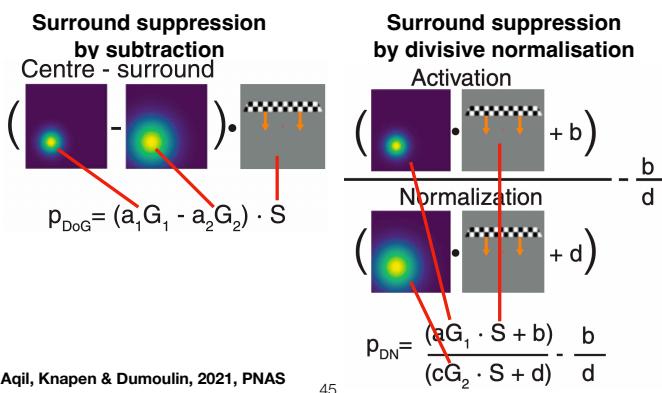
-More generally, the visual system responds to changes rather than constant inputs. These spatially-specific 'changes' can be thought of as 'features'.

-An artificial convolutional network filter is also detecting contrast, or local CHANGE, when the mean of the convolutional filter is zero.

-Any area of the input feature map that has no change in activation (i.e. no contrast) will activate all the weights in the filter by the same amount, making zero activation regardless of the level of activation of the input feature map.

-So activation of an output feature map reflects contrast or structure in the input, specifically structure that is consistent with the structure of the filter weights.

Surround suppression and normalisation



Aqil, Knapen & Dumoulin, 2021, PNAS

45

So, this centre-surround organisation is typically modelled as 2 components.

-First is the response of the centre region (G1) with a particular amplitude (a1) multiplied (.) by the stimulus positions (S).

Here, the checkered bar on the right is just an example of a visual input with a lot of contrast (brightness changes) in a specific set of image positions.

-Second is the response over a larger region, which is called the 'suppressive surround' even though it includes the centre (G2). This has a different amplitude (a2) to the same stimulus (S).

-Typically, the surround response is simply subtracted from the response to the centre, as shown in the equation here on the left.

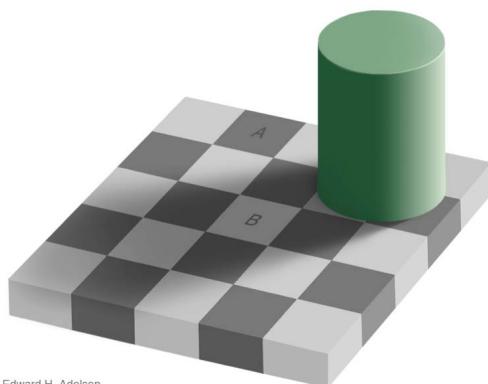
-But recently it has been shown that this is better modelled as the centre response DIVIDED by the surround response. (These equations are more complex because division operations don't cancel out components in the same way as subtractions).

-This division is a form of normalisation, where the centre region activity is normalised by the

mean activity over the larger ‘surround’ region.
-But the mean surround activity used here is not taken over the whole image, like in the global normalisation in artificial deep networks.

- Each is suited to the properties of biological and artificial networks. In the global normalisation of artificial networks, it is faster to take the mean and standard deviation of each feature map’s activity only once and apply this by subtracting and dividing the mean and standard deviation everywhere.
- In biological networks, there is a limit to the spatial extent of nearby neurons the dendritic tree can interact with.
- This allow more complex patterns of normalisation, which might be useful if an image has light parts and dark parts.

Perception of likely object brightness, not retinal light

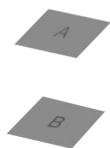


Edward H. Adelson

46

Our perception of brightness is therefore strongly affected by the brightness of the immediately surrounding area.
One major benefit for our perception is that we perceive the object’s likely real-world colour regardless of how it is illuminated: neural responses take surround colours into account, so normalise responses by surrounding colours.

Perception of likely object brightness, not retinal light



47

Removing the context makes this clear, but it is very hard to see in the original image

The receptive field

- The part of the input IMAGE that produces a positive response in a biological neuron is the 'receptive field'
 - The surrounding area inhibiting that response is the 'suppressive surround'
- In artificial networks, 'receptive field' often means the spatial extent of a convolutional filter
 - Often in its INPUT FEATURE MAP, not the input image
 - This is a programmed parameter (filter size)
 - It's harder to determine the spread of the inputs in the input image
- In biological neurons, it's really hard to determine the spatial spread of a neurons input
 - This is sometimes called the 'connective field'

-The area producing a positive response is called the receptive field of the cell. The larger area suppressing or normalising that response is called the suppressive surround, though it is better described as normalisation than suppression and the 'surround' also includes the receptive field itself.

-In an artificial neural network, the term 'receptive field' is often used for the spatial extent of a filter. Properly, this should refer to its spatial extent in the input image, though it is often used to mean the extent in the previous feature map.

-In biological systems, 'receptive field' always refers to the extent in the retinal image, partly because it's really hard to determine the spatial spread of a cell's inputs. We'll look at some methods to do this in the next class.

Deep learning in biological neurons and networks

- Shares most operations with artificial networks (except pooling), but typically with some differences
- Filter/convolve:
 - Dendritic tree synapses with many neurons
 - Different synapses have different strengths (weights)
 - Filters cross multiple feature maps, even at earliest stages
 - Similar dendritic trees across the whole image, like filters tested at every location
- Threshold:
 - Activation of voltage-gated ion channels
 - Following summation on EPSPs and IPSPs
 - Leads to action potentials
- Normalise:
 - Inhibition of activity by local mean activity
- Learning
 - Hebbian learning: 'cells that fire together wire together'
 - Strengthens responses to patterns of activity previously causing with responses