Computational Neuroscience

Focused summary of selected topics from Coursera Computational Neuroscience course, supplemented by Dayan and Abbott[1]

Computational neuroscience provides tools and methods for characterizing what nervous systems do, determining how they function, and understanding why they operate in particular ways.

These questions are addressed by different kinds of models:

1. Descriptive - *what*. How do neurons respond to external simuli and how do we describe this with a quantitative model?
2. Mechanistic - *how*. How do neurons and nervous systems behave on basis of anatomy, physiology and circuitry?
3. Interpretive - *why*. What are the computational and information-theoretic principles underlying brains' operation?

Example of models for receptive fields

Definition. *Receptive field* Specific properties of a stimulus that generate a strong response from the cell. (NB. E.g. for retinal receptive fields, the sensory stimulus is not coming from the entire retina, but a very small region of the retina associated with that cell.)

1. Descriptive - can model retinal cell responses to stimulus as e.g. "on-center, off-surround" receptive fields, whereby cell responds only if stimulus is concentrated in the center of the region. Similarly have oriented receptive fields in the V1 cortex, which respond e.g. to oriented bars of light.
2. Mechanistic - how do we go from retinal center-surround RFs to oriented RFs?
3. Interpretive - e.g. efficient coding hypothesis, goal of the brain being to represent natural images as *faithfully* and *efficiently* as possible.

1 Basic neurobiology

Neurons A biological neuron is mainly composed of three parts:

1. Soma - cell body
2. Dendritic tree - antennae of the neuron, covered with thousands of synapses
3. Axon - structure used by a neuron for transfer of electrical signals

Action potentials The layer separating the inside of the neuron from the inside is called the neuronal membrane. The resting potential of the membrane is slightly negative ($\sim$-70mV), hence polarised. The electrical signals from other neurons are accumulated within the dendritic tree. When the signal is strong enough and the membrane is depolarised beyond a threshold level, an outgoing spike (action potential) is generated. An action potential is a $\sim$100mV fluctuation in the electrical potential across the cell membrane that lasts for about 1ms.

Synapses Action potentials travel down the axon to synapses on dendritic trees of other neurons. Most synaptic transmission is chemical. Arrival of an action potential to the axon terminal causes release of neurotransmitters in the synapse; these neurotransmitters then pass through the synaptic cleft and bind to receptors in the membrane of the postsynaptic neuron. This opens holes through which ions can pass. The resulting outcome can be either be excitatory or inhibitory. The effectiveness of the synapse is the key to learning.

Synaptic plasticity Synapses allow learning in the brain through synaptic plasticity. One particular example is Hebbian plasticity, where synaptic strength between two neurons would for example increase if the two neurons often fire together. Crucially, synaptic plasticity depends on relative spike timings. With spike-timing dependent plasticity, repeated presynaptic spike arrival a few milliseconds before postsynaptic action potentials leads to Long-Term Potentiation (LTP) of the synapses, whereas repeated spike arrival after postsynaptic spikes leads to Long-Term Depression (LTD) of the same synapse.

2 Neural Encoding

One of the best studied processes in the brain is the transformation and representation of sensory information. Our task is to discover how such information is represented, i.e. what is the neural code. Characterizing the relationship between stimulus and response is difficult because neuronal responses are complex and variable. Instead of aiming to predict each spike deterministically, we seek probabilistic models of evoked spike sequences in response to stimuli. This can both be done for single neurons and populations of neurons. The patterns of responses to stimuli in both cases define a neural code.

2.1 Firing rates

Neuronal responses are usually recorded electronically, both via intracellular and extracellular electrodes. The output then is essentially a sequence of timings of spikes. To mathematically describe such spike sequences, we idealise spikes as Dirac-delta functions. A given spike sequence of spikes occurring at times t_i can then be expressed by a neural response function

$$\rho(t) = \sum_i \delta(t - t_i) \tag{1}$$

We can get the number of spikes occurring in a given time interval from 0 to T as

$$n = \int_0^T d\tau \rho(\tau) \tag{2}$$

For a single trial we can then define the spike-count rate as

$$r = \frac{n}{T} = \frac{1}{T} \int_0^T d\tau \rho(\tau) \tag{3}$$

We denote trial averaged quantities using the same stimulus by angle brackets $\langle\rangle$. The time-dependent firing rate is then

$$r(t) = \frac{1}{\Delta t} \int_t^{t+\Delta t} d\tau \langle \rho(\tau) \rangle \tag{4}$$

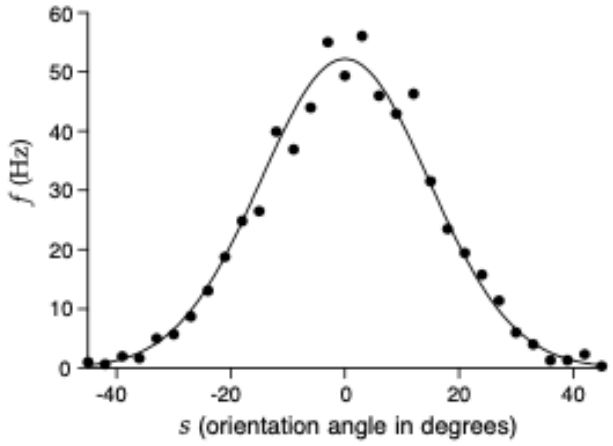
For sufficiently small Δt, any given trial would see at most a single spike in the interval $(t, t+\Delta t)$. $r(t)\Delta t$ would then be equal to the fraction of trials that saw a spike in that interval, or equivalently the probability of seeing a spike in that interval.

We can get the overall average firing rate by averaging the spike-count rate across trials

$$\langle r \rangle = \frac{\langle n \rangle}{T} = \frac{1}{T} \int_0^T d\tau \langle \rho(\tau) \rangle = \frac{1}{T} \int_0^T dt r(t) \tag{5}$$

In practice, however, the firing rate cannot be determined exactly from raw spike trains, and we resort to various approximations. Two example approximations are binning and counting spikes, or using a linear filter.

Tuning curve example Neuronal responses typically depend on many different properties of a stimulus. For now we assume dependence of responses on a single attribute of a stimulus. As a measure of response, we consider the average firing rate. A tuning curve is then this average firing rate as a function of the stimulus attribute.



Above tuning curve is of a neuron in monkey primary visual cortex, where the stimulus was a bar of light presented at varying angles of orientation.

2.2 Basic response models

We present some basic models of increasing complexity linking stimulus to response.

Linear response Response is some linear, potentially time-delayed transformation of the stimulus.

$$r(t) = \phi s(t - \tau) \tag{6}$$

Temporal (linear) filtering We effectively assign a weight to stimuli in the past, and sum them to get the current response.

$$r(t) = \int_{-\infty}^{t} d\tau s(t - \tau) f(\tau) \tag{7}$$

Spatial (linear) filtering As above, but instead of varying in time, we vary across space.

$$r(x, y) = \int \int dx' dy' s(x - x', y - y') f(x', y') \tag{8}$$

Key property of the linear filter is that it searches for portion of the input similar to the filter, i.e. it is a feature! The filtering process is also a projection.

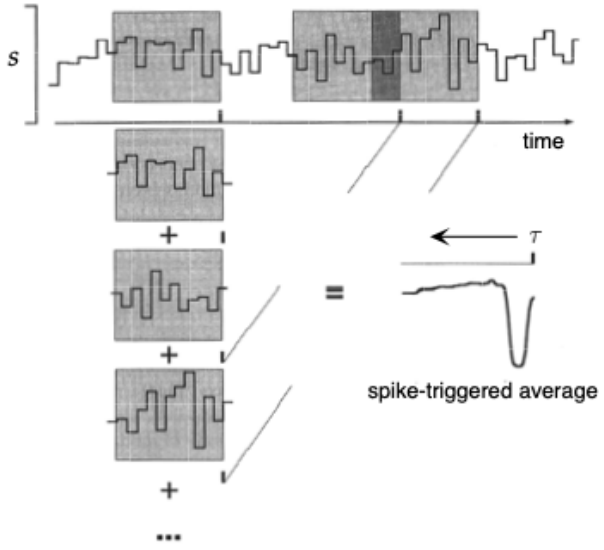
Non-linearity Ensuring proper behaviour of the response, such as positive firing rates and proper saturation can be achieved by adding a static non-linearity g to the linear filter output.

$$r(t) = g(\int_{-\infty}^{t} d\tau s(t - \tau) f(\tau)) \tag{9}$$

2.3 Determining features

We now seek to characterise neuronal selectivity, which is the same as finding the features of the stimulus producing given responses. A useful quantity for this goal is the spike-triggered average. It is the average value of the stimulus at time τ before a spike is fired (averaged over trials).

$$C(\tau) = \langle \frac{1}{n} \sum_i s(t_i - \tau) \rangle \tag{10}$$



Of course what remains is the choice of the stimulus we present. A useful stimulus is the white-noise stimulus, uncorrelated across time. The STA will then give us a good approximation to a single feature. For multiple features, we could use PCA on the stimuli that triggered a spike.

2.4 Poisson process

So now we have some model of $r(t)$, which recalling is proportional to the probability of seeing a spike, and want to relate it to the actual arrival time of spikes. This stochastic process can be modelled as a Poisson process, where we assume there is no dependence on preceding events. We also assume a constant firing rate over time for the demonstration.

Consider a binomial setting of n coin tosses by performing each in a discretised time interval Δt, $n = \frac{T}{\Delta T}$. Let probability of success for a given coin toss be p. The distribution over k successes is given by the binomial distribution

$$P(k) = \binom{n}{k} p^k (1-p)^{n-k} \tag{11}$$

However, possible spike times are continuous, so we take the limit of $\Delta t \to 0$. We also bring in the rate by letting $p = r\Delta t$. Appropriate approximations then yield the Poisson distribution as

$$P_T(k) = \frac{(rT)^k e^{-rT}}{k!} \tag{12}$$

The probability density of time intervals between adjacent spikes is called the interspike interval distribution. Consider a spike occurring at time t_i. Probability of a spike occurring during interval $(\tau, \tau + \Delta t$ which falls between t_i and t_{i+1}, is probability of no spike occurring for time τ, followed by probability of a single spike occuring in time Δt.

$$P_\tau(0) \times r\Delta t = e^{-r\tau} r\Delta t \tag{13}$$

so density is $re^{-r\tau}$.

3 Information theory and neural coding

Information theory allows us to quantitatively answer the question "how much does a neural response tell us about a stimulus?".

Different neuronal responses form a symbolic neural code, e.g. discrete spike-count firing rates. We then consider two quantities: *entropy*, and *mutual information*.

Entropy measures the theoretical capacity of a code to convey information; on average, how interesting/surprising a set of responses is. Put yet differently, it measures variability of responses, but does not tell us anything about the source of that variability. It is defined as

$$H[R] = -\sum_r P(r) \log P(r) \tag{14}$$

Mutual information measures how much of that capacity is actually used when the code is employed to describe a particular set of data. To convey information about a set of stimuli, neural responses must be different for different stimuli. A neuron can provide information about a stimulus only if its response variability is correlated with changes in that stimulus. One way of determining this correlation is to compare responses using a different set of stimulus on every trial - total response entropy - to responses in trials involving repeated presentation of the same stimulus - so called noise entropy.

The intuition is that if a response tells us something about the stimulus, it should vary less if that stimulus is repeatedly presented than in the case random stimulus is presented.

$$H[R \mid S] = \sum_s P(s) H[R \mid s] = -\sum_s P(s) \sum_r P(r \mid s) \log P(r \mid s) \tag{15}$$

This is the entropy associated with response variability not due to changes in the stimulus (stimulus is fixed for calculation of each $P(r \mid s)$). Mutual information then is the subtraction of this quantity from the total response entropy:

$$I(R; S) = H[R] - H[R \mid S] \tag{16}$$

If our response measure is a continuous quantity, we have to include a limit Δr on the measurement accuracy, as otherwise the infinite accuracy of the variable would allow transmission of infinite amount of information. The probability of falling into a discrete bin is then $P(r)\Delta r$.

$$H[R] = -\sum P(r)\Delta r \log P(r)\Delta r = -\sum P(r)\Delta r \log P(r) - \log \Delta r \tag{17}$$

In the limit we can thus only determine the entropy up to an additive constant, yielding the differential entropy

$$\lim_{\Delta r \to 0} (H + \log \Delta r) = -\int dr P(r) \log P(r) \tag{18}$$

Note that since mutual information is the difference of entropies, the $\log \Delta r$ term cancels, so we can evaluate it exactly as

$$I(R, S) = \int ds \int dr P(s) P(r \mid s) \log \frac{P(r \mid s)}{P(r)} \tag{19}$$

3.1 Information in spike trains

A spike train is characterised by a sequence of continuous spike times. To obtain discrete symbols, we discretise time into bins of length Δt, which is fine enough that at most a single spike falls into each bin. We can thus binarise the entire sequence, indicating a spike with a 1, and 0 otherwise. The sequence is then chopped into blocks b (can be overlapping) of duration T_s, meaning each block, now a symbol, constitutes $T_s/\Delta t$ binary values - e.g. "001010".

Probability of a given block b can then be estimated by counting how many times b has been seen within spike trains (overlaps included), across trials. The full spike train entropy is then simply

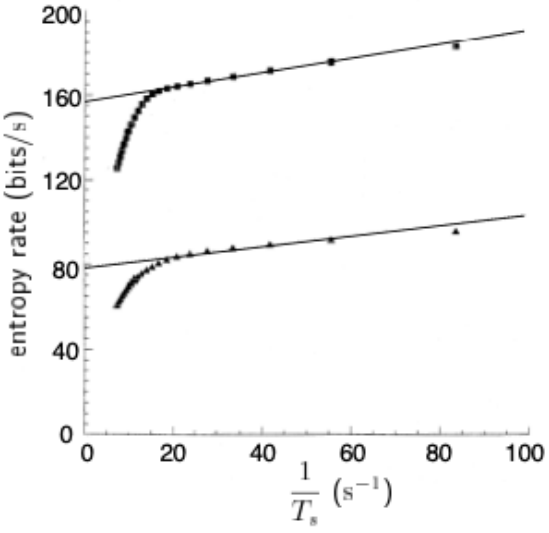
$$H[B] = -\sum_b P(b) \log P(b) \tag{20}$$

The noise entropy requires us conditional probabilities of blocks given fixed stimuli. This is done by taking a long random stimulus sequence, and repeating it across trials. Blocks starting at time t were then evoked by the same stimulus. The conditional probability of blocks given stimulus is then given by distribution across blocks starting at time t in this setup, which will be proportional to number of times b was seen starting at time t across trials. The full noise entropy is then given by averaging across times t.

$$H[B \mid S] = -\frac{\Delta t}{T} \sum_t \sum_b P(b(t)) \log P(b(t)) \tag{21}$$

since we have a total of $\frac{T}{\Delta t}$ starting times.

However, finite length sequences of duration T_s imply adjacent blocks will invariably be correlated, and the given entropies are actually upper bounds as we will be overestimating the true entropies. As T_s increases, this correlation goes down; however, it also practically makes it much harder to estimate probabilities accurately as we are much less likely to see all possible sequences of this duration. We extrapolate to $T_s \to \infty$ by extrapolating the entropy plots as functions of $1/T_s$ towards 0.



Another key factor in estimating these entropies is the choice of bin discretisation Δt. Intuitively, the finer the discretisation, the more information we can extract from spike sequences. This trend should uphold to some lower bound on Δt which matches the natural degree of noise in spike timings.

The figure above shows such measurements for a fly H1 visual neuron. Since the entropy/mutual information tend to grow linearly with length of blocks considered, we instead report entropy rates by dividing entropies by T_s. For a $\Delta t = 3ms$, the information rate is approximately $159 - 79 = 78$ bits/s.

3.2 Entropy maximisation

Information theory also provides a lens through which we could characterise computational principles underlying observed response selectivity. For example, we might ask whether neural responses to natural stimuli are optimized to convey as much information as possible. This hypothesis can be tested by computing the response characteristics that maximize the mutual information conveyed about naturally occurring stimuli and comparing the results with responses observed experimentally.

Given mutual information is proportional to full response entropy, we will first consider how this response entropy can be maximised. Our analysis will crucially depend on choice of constraint.

Single neuron - fixed maximum firing rate We are maximising

$$-\int_0^{r_{\max}} dr P(r) \log P(r) \tag{22}$$

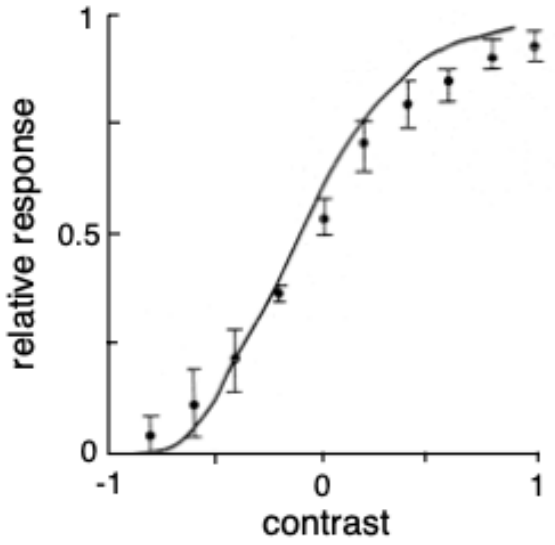
subject to

$$\int_0^{r_{\max}} dr P(r) = 1 \tag{23}$$

which leads to the solution

$$P(r) = \frac{1}{r_{\max}} \tag{24}$$

Probability theory then dictates that this is achieved if we have $r = f(s)$, where f is the CDF of $P(s)$.



As an experimental example, it was found that the large monopolar cell in the visual system of the fly satisfies this condition. Plotted is comparison of contrast response to CDF of natural contrast.

If our constraint is instead a desired mean firing rate, we obtain the exponential distribution. If both mean and variance are given, we obtain the Gaussian distribution.

Populations of neurons When a population of neurons encodes a stimulus, optimising their individual response properties will likely lead to different neurons encoding the same information, thus leading to a suboptimal population encoding.

Consider a vector of N firing rates. We have

$$H = -\int d\mathbf{r} P(\mathbf{r}) \log P(\mathbf{r}) - N \log \Delta r \tag{25}$$

This full response entropy is upper bounded by the sum of individual neuron entropies. To see this, consider the following KL divergence, which is one measure of redundancy:

$$D_{KL}[P(\mathbf{r})\|\prod_i P(r_i)] = \int d\mathbf{r} P(\mathbf{r}) \log \frac{P(\mathbf{r})}{\prod_i P(r_i)} = \sum_i H_i - H \geq 0 \tag{26}$$

Equality is only achieved if $P(\mathbf{r}) = \prod_i P(r_i)$, i.e. responses are statistically independent. Such a code is called a *factorial code*. Thus theoretically optimal population encoding is achieved if responses are independent, and each response probability is optimal for given constraint (c.f. single neuron entropy maximisation). Practically, these may be impossible achieve depending on the form of the neural response (e.g. linear function of stimulus).

Instead we desire that the lowest-order moments of our population response distribution match those of a fully factorised and equalised distribution. In the first order, the means will be the same, μ_r. In the second order, the covariance matrix will be proportional to the identity matrix, i.e.

$$Q_{ij} = \int d\mathbf{r} P(\mathbf{r})(r_i - \mu_r)(r_j - \mu_r) = \sigma^2 \delta_{ij} \tag{27}$$

Application to Retinal Ganglion Cell Receptive Fields

We now look at evidence of this principle in receptive fields early in the visual pathway. Since information theoretical analyses are sensitive to the statistical properties of the stimuli being represented, the statistics of natural scenes play an important role.

We describe a visual stimulus using a contrast function $s(x, y, t)$, which is proportional to difference between luminance at a point and average luminance, potentially also further divided by average luminance. We also combine (x, y) into a single vector $\mathbf{v}$.

First we analyse the spatial receptive field. The linear estimate of the response of a visual neuron is then

$$L_s = \int d\mathbf{v} D_s(\mathbf{v}) s_s(\mathbf{v}) \tag{28}$$

An important assumption we make about the statistics of the natural input is that they are spatially translation-invariant, i.e. locations we consider are equivalent. This motivates giving all the receptive fields the same spatial structure, so the spatial kernel describing a retinal ganglion cell with receptive field centered at the point $\mathbf{v}$ can be written as $D_s(\mathbf{v} - \mathbf{a})$.

$$L_s(\mathbf{a}) = \int d\mathbf{v} D_s(\mathbf{v} - \mathbf{a}) s_s(\mathbf{v}) \tag{29}$$

In this way we are labelling neurons by the centers $\mathbf{a}$ of their receptive fields. It is also reasonable to assume that these receptive fields are dense enough such that the responses can be deemed continuous functions of $\mathbf{a}$.

Whitening filter

We now show how the above moment-matching objective leads to a whitening filter. Covariance of interest is between spatial linear responses with different receptive field centers, averaged across natural stimuli

$$Q_{LL}(\mathbf{a}, \mathbf{b}) = \langle L_s(\mathbf{a}) L_s(\mathbf{b}) \rangle = \int \int d\mathbf{v} d\mathbf{w} D_s(\mathbf{v} - \mathbf{a}) D_s(\mathbf{w} - \mathbf{b}) \langle s_s(\mathbf{v}) s_s(\mathbf{w}) \rangle \tag{30}$$

We wish to solve the following for D_s:

$$Q_{LL}(\mathbf{a}, \mathbf{b}) = \sigma^2 \delta(\mathbf{a} - \mathbf{b}) \tag{31}$$

Given our assumption about the spatial homogeneity, the covariance between stimuli $\langle s_s(\mathbf{v}) s_s(\mathbf{w}) \rangle$ is in-fact only a function of the distance between the two points. We denote this covariance as

$$Q_{ss}(\mathbf{v}-\mathbf{w}) = \langle s_s(\mathbf{v}) s_s(\mathbf{w}) \rangle \tag{32}$$

We can solve for D_s by expressing D_s and Q_{ss} in terms of their Fourier transforms

$$D_s(\mathbf{v}-\mathbf{a}) = \frac{1}{4\pi^2} \int d\hat{\kappa} \exp(-i\hat{\kappa}(\mathbf{v}-\mathbf{a}))\hat{D}_s(\hat{\kappa}) \tag{33}$$

$$Q_{ss}(\mathbf{v}-\mathbf{w}) = \frac{1}{4\pi^2} \int d\hat{\kappa} \exp(-i\hat{\kappa}(\mathbf{v}-\mathbf{w}))\hat{Q}_{ss}(\hat{\kappa}) \tag{34}$$

From which we obtain only the amplitude of the linear kernel (we are free to choose the form).

$$|\hat{D}_s(\hat{\kappa})|^2 \hat{Q}_{ss}(\hat{\kappa}) = \sigma^2 \tag{35}$$

The product on the left side is the power spectrum of L. It is independent of the spatial frequency $\hat{\kappa}$, and therefore has the same characteristics as white noise. This kernel is thus called a whitening filter.

Noise filter

A notable consequence of the whitening filter maximising entropy is that they equalise distribution of response power over the entire spatial frequency range. High spatial frequencies usually correspond to noise, especially in low-light conditions, thus amplifying them is undesirable. This arises because pure entropy maximisation makes no distinction between entropy from natural signals, and noise. An approximate solution to this is to filter the input first to remove noise, and maximise the entropy of the resulting signal.

The approach taken is to express the Fourier transform of the linear kernel in terms of this additional noise filter, and previous whitening filter. The desired property of the noise filter is that when applied to the overall input, consisting of signal and noise, the output is as close as possible to the signal.

The resulting linear kernel is radially symmetric with filtering properties depending on noise condition. In the low noise regime, it has a bandpass character (filters for only select range of frequencies), with a center-surround structure. In the high noise regime, it is low-pass and loses its surround, instead averaging inputs to reduce noise.

4 Plasticity and Learning

Here we discuss various synaptic plasticity rules based on Hebbian learning. General forms of the Hebb rule state that synapses change in proportion to the correlation or covariance of the activities of the pre- and postsynaptic neurons.

4.1 Single postsynaptic neuron

We take x and y to denote pre- and postsynaptic neurons respectively, connected by synapses of strength denoted by weights w. The postsynaptic activity we take as

$$y = \mathbf{w}^T \mathbf{x} \tag{36}$$

Basic Hebb rule (correlation) Simultaneous pre- and postsynaptic activity increases synaptic strength.

$$\tau \frac{d\mathbf{w}}{dt} = y\mathbf{x} \tag{37}$$

Synaptic plasticity is generally modelled as a slow process relative to presentation of inputs. We can approximate changes induced by a set of inputs by averaging across the updates across them. This gives us the correlation rule as

$$\tau \frac{d\mathbf{w}}{dt} = \langle y\mathbf{x} \rangle = \langle \mathbf{x}\mathbf{x}^T \rangle \mathbf{w} = Q\mathbf{w} \tag{38}$$

Both these rules are unstable. The rate of change of the length of the weight vector is

$$\frac{d}{dt}\|\mathbf{w}\|^2 = 2\mathbf{w}^T \frac{d\mathbf{w}}{dt} = 2y\mathbf{w}^T\mathbf{x} = 2y^2 > 0 \tag{39}$$

Covariance rule If $\mathbf{x}$ and y represent neuron activities, they should be positive. Correlation-based rules then only capture LTP. Covariance rule accounts for both LTP and LTD.

$$\tau \frac{d\mathbf{w}}{dt} = y(\mathbf{x} - \langle \mathbf{x} \rangle) \tag{40}$$

$$\Rightarrow \langle y(\mathbf{x} - \langle \mathbf{x} \rangle) \rangle = \langle (\mathbf{x} - \langle \mathbf{x} \rangle)(\mathbf{x} - \langle \mathbf{x} \rangle)^T \rangle \mathbf{w} = C\mathbf{w} \tag{41}$$

Covariance rule is similarly unstable.

$$\frac{d}{dt}\|\mathbf{w}\|^2 = 2\mathbf{w}^T \frac{d\mathbf{w}}{dt} = 2y\mathbf{w}^T(\mathbf{x} - \langle \mathbf{x} \rangle) = 2y(y - \langle y \rangle) \tag{42}$$

Averaging above gives variance of y which for non-trivial behaviour will be greater than 0.

BCM rule The covariance rule can lead to LDP even if there is no presynaptic activity. Experimental evidence supports the requirement that both pre- and postsynaptic activity is required for synaptic weight updates. This motivates the following modification

$$\tau \frac{d\mathbf{w}}{dt} = y\mathbf{x}(y - \theta_y) \tag{43}$$

where θ_y is a threshold on the postsynaptic activity that determines whether synapses are strengthened or weakened. If θ_y is kept fixed, BCM is also unstable.

Weight constraints Rules encountered so far were all unstable, leading to unbounded growth in the weights. A direct way of limiting this growth is through *synaptic saturation*, where the size of the weights is upper bounded (and potentially lower bounded too).

Another approach to stabilisation is to normalise weights, which involves imposing some sort of global constraint. For weights that can be positive and negative, this could come in the form of constraining the sum of the squares of the weights.

Oja's rule A dynamic rule which ensures the constancy of the squared norm of the weight asymptotically is

$$\tau \frac{d\mathbf{w}}{dt} = y\mathbf{x} - \alpha y^2 \mathbf{w}, \quad \alpha > 0 \tag{44}$$

The norm of the weight vector in this case settles to $1/\sqrt{\alpha}$

$$\frac{d}{dt}\|\mathbf{w}\|^2 = 2\mathbf{w}^T \frac{d\mathbf{w}}{dt} = y^2(1 - \alpha\|\mathbf{w}\|^2) \tag{45}$$

Competition Hebbian learning also suffers from the fact that synapses are modified *independently*. This can, for example, drive all the weights to their imposed maximum values. More generally, some form of *synaptic competition* is required for a neuron to develop input selectivity. In some cases, the same mechanism that leads to competition also stabilizes growth of the synaptic weights. In other cases, it does not, and saturation constraints must also be imposed.

Of the learning rules covered so far, only Oja's rule induces competition. Since the norm is constrained, increases in some weights must be accompanied by decreases in other weights.

PCA We discuss one computational property of rules discussed so far. Consider the basic correlation/covariance rule. Ignoring constraints on $\mathbf{w}$ allows us to solve for $\mathbf{w}$ with matrix diagonalisation. We express $\mathbf{w}$ as a linear combination of the eigenvectors of C:

$$\mathbf{w}(t) = \sum_i^D c_i(t)\mathbf{e}_i \tag{46}$$

$$\Rightarrow \tau \frac{d\mathbf{w}}{dt} = \tau \sum_i^D \frac{d}{dt} c_i(t)\mathbf{e}_i = \sum_i^D c_i(t)\lambda_i \mathbf{e}_i \tag{47}$$

$$\frac{d}{dt}c_i(t) = \frac{\lambda_i}{\tau}c_i(t) \Rightarrow c_i(t) = c_i(0)\exp\left(\frac{\lambda_i}{\tau}t\right)\mathbf{e}_i \tag{48}$$

Correlation/covariance matrices are positive semi-definite, so eigenvalues are non-zero and the exponential grows with time. Further, the sum will be dominated by the component with the largest eigenvalue, here taken to be $\mathbf{e}_1$, so $\mathbf{w}$ will be aligned with the first principal eigenvector!

Same alignment is found with Oja's rule, but with length constrained to $1/\sqrt{\alpha}$.

4.2 Multiple postsynaptic neurons

This is a short discussion on another aspect of competition that becomes relevant with multiple outputs. To ensure output units don't all learn the same selectivity, competition between them can be introduced through a separate set of recurrent synapses.

One way to reduce redundancy in a linear model is to make linear recurrent interactions plastic rather than fixed, using an *anti-Hebbian* modification rule. The recurrent synapses tend to make the output units less correlated. Importantly, being only linear models, they are capable of removing only second-order redundancy, i.e. redundancy characterized by the covariance matrix.

Generally, competition can be made stronger through non-linear interactions.