INTRODUCTION

**Course Objectives.** The main objective of this course is to present some major concepts in the field of computational neuroscience and to give the students some idea of common approaches taken by computational neuroscientists. The basic thinking in the presentation of the field given here is that the key contributions of computational neuroscience are conceptual, and do not rely on a deep understanding of the underlying mathematics, but rather on an understanding of "systems neuroscience". However, some mathematical concepts will be presented, because in my opinion, some of the major insights gained from computational neuroscience to understanding the nervous system result from translating biological facts into mathematical concepts and vice-versa. This course is addressed to a wide audience with a variety of backgrounds; hence basic neurobiology as well as basic mathematical concepts will be presented as needed. If at any point during this class, your understanding of the mathematics or neuroscience presented is less than rock-solid, don't worry. I will not expect that students taking this class can pass an exam in linear algebra, differential equations or cellular biophysics. I will only expect you to understand what the mathematical concepts represent, not to be able to do the math yourselves. In the same spirit, if you don't understand some of the readings, don't worry! Readings partially consist of Journal papers, which are dense and difficult. I will expect you to make an effort, but will not expect you to grasp everything you read without help and explanations. The readings are necessary in order to help the students to get a feeling for the field, but I am well aware of the fact that papers are rarely written clearly enough to be easily understood by a student in a 300-level class. The most important thing to remember is: ask questions! If you do not feel comfortable asking questions in class, email me or come talk to me. The only means at the disposition of an instructor to know how well the material is understood by the class is feedback from the students! I am more than willing to repeat anything that is difficult, to explain concepts and to spend time with students. However, I can only do that if you let me know when difficulties arise!

**What is Computational Neuroscience? and what does this course attempt to do?**

Good question …

* Attempts to understand how brains "compute"
* Using computer to simulate and model brain function
* Applying techniques from computational fields (math, physics) to study brain function

The term Computational Neuroscience covers a lot of different approaches and is used by many scientist in many different ways. This class will obviously cover no more than a small fraction of the approaches, questions and tools commonly considered to fall under the term “Computational Neuroscience”.

The following is an extract of a text written by my colleague Todd Troyer from the University of Texas San Antonio. This text, which is part of the notes for his class, clearly describes some ways to classify Computational Neuroscience: “The term computational neuroscience covers a dizzying array of approaches to understanding the nervous system, and to achieve any coherence in constructing a course called “computational neuroscience” requires vast subsections of the field to be excluded. A number of different schemes have been used to divide up the field. Many of these fall under the phrase “levels of analysis.” One division according to level of analysis is biological: the brain can be studied at a hierarchy of scales ranging from the cellular and molecular level to the level of small localized circuits in the brain to the level of large-scale brain circuits involving multiple neural subsystems. A second class of scheme is that proposed by David Marr (1982). Marr was making computer models of visual perception, and made the distinction between three levels of analysis: the computational, the algorithmic and the implementational levels of analysis. Roughly, the computational level is the most abstract and concerns itself with a description of the problem to be solved, i.e. what is the computation that is being performed. Here the object of study are the high-level computational principles involved such as optimality, modularity, etc. The algorithmic level of description concerns itself with the structure of the solution, e.g. the nature of the subroutines used to perform the calculation. Finally, the implementational level concerns how this algorithm is actually implemented in a machine. The difference between the algorithmic and implementational levels is often described as analogous to the difference between a high level programming language like C or Lisp and it’s platform specific implementation is machine or assembly language.
Yet another scheme that falls under the “levels of analysis” rubric is the distinction between “top-down” and “bottom-up” approaches to understanding the brain. A top-down approach starts at the level of cognitive phenomena and tries to reach “down” to connect these phenomena to specific events taking place in the brain. The bottom-up approach starts with biological knowledge about brain cells and circuits and tries to determine how these mechanisms support complex mental phenomena.

Dayan and Abbott (2001) have made yet another tripartite division, dividing computational and theoretical models according to three basic purposes for which they can be used. Mechanistic models concern themselves with how nervous systems operate based on known anatomy and physiology. Descriptive models summarize large amounts of experimental data, accurately describing and quantifying the behavior of neurons and neural circuits. Finally, interpretive models explore the behavioral and cognitive significance of nervous system function, often connecting explaining experimental data in terms of certain theoretical principles.

Each of these schemes for subdividing the field has advantages and disadvantages. In many cases, they can be brought into rough alignment. The topics addressed by these notes, can be best localized in terms of the biology hierarchy - almost all the topics explored fall at the level of neurons or local circuits. The notes are generally “middle-out” although they probably fall closer to the bottom-up rather than the top-down approaches. A truly complete survey of computational neuroscience would probably treat two separate clusters of ideas that are dealt with here in only a cursory manner. The first tradition traces its roots back to by far the most successful model in all of neuroscience: the Hodgkin-Huxley model. The formalism embodied in their model laid the groundwork for how we understand the electro-chemical events at the heart of how neurons transmit and transform information. So-called compartmental modeling belongs to this tradition. “Todd Troyer, Class Notes for NACS 643, 2005.

Learning objectives: The present course, and with it its notes, present Computational Neuroscience from a Biological rather than Mathematical point of view. The goal is to understand what types of computations brains are faced with and what solutions scientists think exist. Students will read equal numbers of experimental and computational/theoretical papers. A major focus is the “translation of data into computational terms and equations. Students will gain a basic understanding of:

A. “Neural computation” – what can neurons and networks of neurons compute
B. Current models of learning and memory
C. Translation of experimental data into computational frameworks
D. Reading computational papers
E. Appreciation of mathematical concepts

Outline. The course will focus on how the brain (1) encodes, (2) performs computations on, (3) memorizes, (4) integrates with previously gathered information and (4) decodes information. Theoretical concepts taught include - but are not limited to - neural coding, population vector representations, neural assemblies, oscillations and synchrony, hebbian plasticity, linear associators, associative memory function, neuromodulation, models of human memory. Additional concepts and approaches briefly introduced include: cable theory, compartmental modeling, information theory, statistical approaches, Bayesian approaches ... Students will be expected to actively participate, problem solve and design, either using code or just pen and paper, models to solve particular problems. The course includes lectures, discussion, hands on computer exercises, homework, quizzes, readings of original literature and writing of two papers.

Historical notes on Neural Nets and Computational Neuroscience (incomplete)

1890: William James Detailed, mechanistical model on association that is almost identical in structure to later (1970) associative memory networks. “The amount of activity at any given point in the brain cortex is the sum of the tendencies of all other points to discharge into it, such tendencies being proportionate (1) to the number of times the excitement of each other point may have accompanied that of the point in question; (2) to the intensity of such兴奋ments and (3) to the absence of any rival point functionally disconnected with the first point, into which the discharge might be diverted.”
1907: Louis Lapicque published a paper on the excitability of nerves that is often cited in the context of integrate-and-fire neurons.

1943: Warren McCulloch and Walter Pitts Networks of logical threshold units (all or nothing responses) can perform logic calculations. Any finite logical expression can be realized by these McCulloch-Pitts neurons. Describes a true connectionist model, with simple computing elements, arranged largely in parallel, doing powerful computations with appropriately constructed connections. (see lecture 3).

1949: Donald O. Hebb The organization of behavior was the first explicit statement of a physiological learning rule for synaptic modification (since become known as the Hebb rule). “When an axon of cell A is near enough to excite a cell B and repeatedly or persistently takes part in firing it, some growth process or metabolic change takes place in one or both cells such that A’s efficiency, as one of the cells firing B, is increased.” (lectures 16-18)

1952: Hodgkin and Huxley developed mathematical model of squid giant action potential.

1956: Rochester, Holland, Haibt and Duda Probably the first attempt to use computer simulations to test a well formulated theory based on Hebb’s postulate of learning. Discovered the nearly universal finding for computer simulations designed to check brain models: the first attempt did not work. The results showed clearly that inhibition needed to be added to the theory.


1956: F. Rosenblatt The perceptron model and the perceptron convergence algorithm Described a learning machine with simple computing elements that was potentially capable of complex adaptive behaviors.

1959: W. Rall develops cable theory, a mathematical means to describe signal propagation in neurons.

1969: Minsky and Papert Used elegant mathematics to demonstrate that there are fundamental limits on what a one-layer perceptron can compute. “In the popular history of neural networks, first came the classical period of the perceptron, when it seemed as if neural networks could do anything. A hundred algorithms bloomed, a hundred schools of learning machines contended. Then cam the onset of the dark ages, where, suddenly, research on neural networks was unlivied, unwanted, and most important, unfunded.”

1973; 1976: Christoph van der Malsburg Demonstrated self-organization in computer simulations motivated by topologically ordered maps in the brain. Also first to develop a theory about coding in Neural Assemblies.

1977-1980: Traub, Llinas and others developed first detailed models of neural function.

1982: John Hopfield Used the idea of an energy function to formulate a new way of understanding the computation performed by recurrent networks with symmetric synaptic connections. He established the relation between such recurrent networks and an Ising Model used in statistical physics. Introduced the notion of "attractor models" to brain-science. (Lecture 21, 22).

1983: Sutton, Barto and Anderson Introduced reinforcement learning and showed that a reinforcement learning system could learn to balance a broomstick in the absence of a helpful teacher.

1986: Rumelhart, Hinton and Williams Developed the backpropagation algorithm which solved the credit assignment problem for multi-layer networks, which emerged as the most popular algorithm for the training of neural networks. It was discovered independently also by Parker and LeCun.

1980’s: Term Computational Neuroscience is created. Neural simulators (GENESIS, NEURON) are developed. First Computational Neuroscience meeting (CNS) and Journal appear. Scientists like Jim Bower, Terry Sejnowski, Larry Abbott identify as Computational Neuroscientists.

1990: Computational Neuroscience becomes a distinct field. Training programs, Journals, Meetings are available.

Why model?
Important insights can be gained from applying computational techniques to understanding the nervous system result from the process of translating biological facts into mathematical facts and vice versa.

* Models help organize the data and motivate experiments.
* Models suggest how data might fit together to yield an explanation of a phenomena.
* Models bring assumptions to the light of the day
* Models are essential for ruling out hypotheses.
Basic Computation

Exercise: (a) Consider the following scenario: A child sees a balloon for the first time. The balloon pops and makes a loud noise. The child is frightened. Two days later, the child sees another balloon and starts crying.
(b) Think through all the computations that may happen in the brain of this child. What information has to be processed? What information has to be stored?
(c) Pretend to write a computer program that would reproduce this scenario. Write a flow diagram for this program.
(d) Look up brain structures and how they connect to each other. Make a diagram of the information flow through such a diagram.

Computing units

The fundamental computing unit is the neuron. Neurons have many different functions, shapes and characteristics. Some types include sensory neurons, pyramidal cells, local interneurons and many more. Neurons communicate with each other and transmit information to each other chemically or electrically, they can excite or inhibit each other. [Appendix: Cells, Neuron]. For the purposes of this class we will use mostly simplified descriptions of neurons to model or simulate their basic characteristics.

Consider the schematic of information flow in a neuron shown below:

In this simplified, one-directional view of neural function, signals travel from other neurons to the dendrites, are integrated in the cell body and then transmitted to yet other neurons via the axons. Most mathematical simplifications of neurons include these three separate functions: Input, transfer function and output. The input function can include the details of synaptic transmission (conductance, transmitter release, Nernst potential), the transfer function describes the transformation of the integrated
input activation into action potentials or some other form of output activation and the output to other neurons consists of whatever activity is transmitted. [Note that this uni-directional depiction of neural function of oversimplified: many neurons can backpropagate action potentials into their dendrites, and many neurons communicate their activation to other neurons via dendritic interactions, and some neurons don't even have axons or don't generate action potentials.]

Here is the simple explanation of how it works:

1) An action potential triggers transmitter release from the axon terminal of the pre-synaptic neuron.
2) The neurotransmitter binds to specific receptors at the postsynaptic neuron.
3) Binding of neurotransmitter leads to a state change in the synaptic channels in such a manner that charged ions can enter or leave the cell more easily. If one considers the flow of charged ions from the outside to the inside (or vice versa) to be a current (synaptic current $i$), then channel itself represents a way for that current to flow. How easy it is for that particular ion to flow through a channel is defined by the channels conductance $g$ and the cells membrane voltage $v$ ($g = 1/resistance$, if you remember $V=IR$ then you can see that $I=VG$).

Glutamate for example binds to channels that when open (high conductance), let Na$^+$ ions flow into the cell. The cell then depolarizes (voltage inside becomes more positive).
4) Because of charged ions flowing in or out of the cell, the cells voltage with respect to the outside changes. The cell depolarizes (becomes more positive) or hyperpolarizes (becomes more negative).
5) This change in voltage triggers a number of changes inside a cell and it can, in some cases lead to an action potential or spike.

Sometimes a neuron is simply considered an all-or nothing device. It adds up its inputs, if they are above a threshold it creates an output, if they are below a threshold, it stays inactive. A formal description of this type of neuron is called a **McCulloch Pitts neuron (or logic threshold unit)**.
This type of formal neuron represents the summation of input at the synaptic site as well as the “all or nothing” nature of the action potential generation. It is a binary device; that is, it can be in only one of two possible states. Each neuron has a fixed threshold \( \Theta \). When then the \( i(t) \) sum of its inputs \( (i_1(t),i_2(t)...) \) is equal or greater than the threshold \( \Theta \), the neuron's output \( x(t) \) becomes active \( (x=1) \), if not it is inactive \( (x=0) \).

\[
\Sigma \quad \quad x(t) = F(I(t), \Theta) \quad \text{output}
\]

\[
I(t) = \sum_{j=1}^{N} i_j(t) \quad \text{total input}
\]

Connected networks of McCulloch-Pitts neurons can perform finite logical expressions.

Exercise: Use networks of McCulloch Pitts neurons to create a NOT, AND, OR and XOR device.
Connections or synaptic weights between neurons can be positive or negative. Define threshold and synaptic weights and assume that a neuron's output can be either 0 or 1.

The most classically used simplified neuron model is probably the **integrate and fire neuron**. An integrate and fire neuron captures one aspect that a threshold unit does not: summation over time. An integrate and fire neurons sums its inputs not only at a given time step, but over time (thus the name integrator). An internal variable \( v(t) \) corresponds to the sum of inputs \( I(t) \) between timestep zero \( (t=0) \) and the time step under consideration \( t \). Thus:

\[
Cm \frac{dv}{dt} = I(t) \quad \text{for } t > 0 \quad \text{or} \quad v = \frac{1}{Cm} \int_{t=0}^{t} I(t) \, dt
\]

\[
f(I) = \frac{I}{CmVth} \quad \text{is the firing rate in response to a constant input } I.
\]

\( Cm \) represents the capacity and defines the slope of integration.
When the internal variable $v(t)$ reaches a threshold $\Theta$, the neurons output $x$ becomes active ($x(t) = 1$), and $v(t)$ is reset to zero. As long as $v(t)$ is smaller than threshold, the output is inactive ($x(t) = 0$). Because the internal variable is reset to zero after the threshold has been reached, the output will become active only for one time step at a time. In this model, the internal variable $v$ is comparable to a neuron’s membrane voltage: when the voltage reaches threshold, the neuron fires an action potential (output = 1) and the membrane voltage is reset to resting potential.

If the input $I$ is a constant in time, the firing frequency $f$ of the neuron is determined by the amplitude of the input and the threshold: $f = I/\Theta$.

In contrast to the simple integrate-and-fire neuron, a leaky integrate and fire neuron assumes a decay of the internal variable $v(t)$ [Appendix: Low pass filter]. Incorporating this assumption leads to a differential equation relating stimulus magnitude, $I$, and the internal variable, $v$: $\tau \frac{dv}{dt} = -v(t) + RI(t)$ . As before, the output becomes active when $v(t)$ exceeds threshold and $v(t)$ is then reset to zero. In this case the relation between the firing frequency $f$ and the amplitude of the input $I$ looks linear except at low values of $I$, where it shows threshold behavior. This means that for the leaky integrate and fire neuron, an input threshold exists below which the neuron’s output stays inactive (this is not true for the integrate and fire neuron!).
The basic electrical circuit this idea is based on is the following:

\[ I(t) = I_R(t) + I_C(t) = \frac{v(t)}{R} + \frac{q(t)}{C} = \frac{v(t)}{R} + C \frac{dV}{dt} \]

\[ \tau_m \frac{dv(t)}{dt} = -v(t) + RI \quad \text{with} \quad \tau_m = RC \text{ (time constant)} \]

**Exercise:** Using firing rate as the variable that defines ON and OFF, create an NOT, AND, OR and XOR gate using leaky integrate and fire neurons. What happens of you lower or raise the firing threshold?

**Exercise:** make a list of characteristics of biological neurons that are captured by the leaky integrate and fire neuron and a list of characteristics that are not captured by it. For these latter ones, suggest changes to make that would capture these.

**Exercise:** (a) You want to create a sensory neuron that responds to sensory stimuli ABOVE a certain threshold with a close to linear input-output function. The response should be in spikes/second. You can assume that the input changes on a very slow time scale (minutes). Choose one of the three types of neurons above to implement this sensory neuron and defend your choices. (b) You want to create a sensory neuron that responds to ANY sensory input and has a linear input-output function. The response should be expressed in spikes/seconds. You can assume that the input changes on a very slow time scale (minutes). Choose one of the three types of neurons above to implement this sensory neuron and defend your choices.

**Synapses**

Neurons communicate information to one another. This is usually done via chemical or electrical synapses. The presynaptic neuron if the neuron sending the signal, the postsynaptic neuron is the neuron receiving the signal. In the case of chemical synapses, the presynaptic neuron releases a substance called a neurotransmitter which binds to receptors on the postsynaptic neuron, and opens channels for charged ions to enter the cell; this results in a number of processes that change the cell’s membrane voltage. Electrical synapses consist in electrical coupling between the two neurons. This works like a resistor through which current can flow back and forth depending on which neuron is more depolarized.

In simulations, or neural networks, communication between neurons can be represented or implemented in many different ways. The simplest is to assign a “weight”, or “effect” to the communication. This describes by how much a pre-synaptic neuron affects a postsynaptic neuron. For a network of McCulloch Pitts neurons for example, each synaptic weight would be a parameter that describes how much input a neuron gets from another neuron.

**Exercise:** You have three McCulloch Pitts neurons. All three have the possible states 0 and 1, and their thresholds are 1.0. Two neurons receive outside inputs (in1, in2), and these two make synapses with the third who is considered the output (o). You want the output to be 1 when the sum of the inputs > 4, 0 otherwise. How do you choose your synaptic weights?

Many times, synaptic weights are called w. Classically, wij designates the synaptic weight between presynaptic neuron j and postsynaptic neuron i. If xj is the output of neuron j, and in is the input of neuron i, then the following can be written: \( in = w_{ij} * x_j \). This basically just says that the input to neuron i from
neuron \( j \) is the weighted output from neuron \( j \). If a neuron receives inputs from many (\( n \)) presynaptic neurons, one can write: 
\[
ini = \sum_{j=1}^{n} w_{ij} \times x_j
\]

These types of linear synaptic interactions are rarely implemented, often, more complex equations are used for synaptic interactions. These take into account the time course of synaptic interactions as well as the probabilistic nature of the transmitter release. We will discuss those details when reading papers employing those methods [Appendix: Mathematical aside].

**Neural coding, Or How the brain represents and transforms information**

**Can a neural code be defined?**

We (neuroscientists) place ourselves in the position of the homunculus, monitoring neural activity in the brain as stimuli vary in time along an unknown trajectory.

Neurons signal information in various manners. For the time being, we will restrict our discussion to information signaling via *spikes*, or *action potentials*.

Lord Ardian ([The Nobel Prize in Physiology or Medicine 1932](https://www.nobelprize.org/nobel_prizes/medicine/laureates/1932/ardian-bio.html)) discovered the following principles of neuronal signaling:

1. Individual neurons produce stereotyped action potentials ("all-or-nothing" law). (see readings on experimental methods for details on action potential recordings).
2. In response to a static stimulus, the rate of spiking increases as the stimulus becomes larger.

*from Rieke et al. (1997) Spikes, MIT Press, Cambridge MA*
In this schematized illustration, the number of action potentials fired (Response) is proportional to the amplitude of the stimulus (Stimulus). If the number of action potentials is reported as a function of the stimulus amplitude (right hand graph), a linear relationship is depicted. Examples of this type of neuronal response include olfactory receptor neurons, cold receptors and stretch receptor neurons.

**Responses of an olfactory receptor neuron to odor stimuli of increasing concentration.**

**Responses of cold receptors to progressively colder stimuli**

\[\text{Amplitude - to - frequency transformation} \]

or "rate coding"

"Rate" or "# of action potentials"
measured over window of stimulus application

Responses of an olfactory receptor neuron to odor stimuli of increasing concentration.

**FIGURE 24-4**

Cold receptors discharge when the skin is cooled below 34°C. A. The frequency of discharge of a single cold fiber of a monkey increases with progressively greater cooling. [Adapted from Darian-Smith et al., 1975.]

Responses of cold receptors to progressively colder stimuli
Responses of stretch receptor neurons.

What's wrong with this simplified picture? For example, real neurons do not fire at arbitrarily high frequencies (the refractory period limits the neuron's ability to fire above a certain frequency).

The fact that neurons do not respond with arbitrarily high frequencies can be described as saturation. In addition, most neurons do not fire in response to arbitrarily low stimulus amplitudes. This property can be described by introducing a "firing threshold". Nociceptors (pain) for example start responding to temperatures around 42 degrees Celsius, whereas nonnociceptive thermoreceptors respond to temperatures as low as 0 degrees Celsius.

Previously, we have assumed that once a stimulus is applied and does not change (is static), the neurons response is static (number of action potentials emitted during a given time stays constant); however, for most neurons, this assumption is not correct: (3) **If a stimulus is applied for a long time, spike rate begins to decline (adaptation).**
For some of the reasons depicted below, it is difficult to establish what the relevant parameters of neural responses are; indeed, these may vary depending on the neuron, stimulus, situation etc.

Let's consider a simple case, the rate code. Rate code means that the number of action potentials fired during a given period of observation contains information about the nature or the amplitude of the stimulus in question.

Now consider the example below. Obviously, an observer of the neuron's firing can distinguish this neuron's responses to different stimulus amplitudes; however, the rate (number of spikes fired during the time of observation or during the time of stimulus application) does not vary as a function of stimulus amplitude.
However, the frequency of these action potentials does change as a function of stimulus amplitude. Similarly, if the window of observation was shortened, the number of action potentials would vary as a function of stimulus amplitude.

In the example below, three parameters of the evoked spike trains change as the stimulus amplitude changes: the number of action potentials fired, the average frequency of this firing, and the first interspike interval.

One can see that "rate" depends strongly on the definition as well as on the method of measuring it. In the example below, the rate of firing is measured using windows of observation of increasing length, which changes the manner in which rate represents the stimulus.
In some cases, the rate or the number of action potentials fired are not informative of stimulus quality or amplitude. However, other informative parameters of the neural response can often be extracted.

In the example above, neither the number, nor the frequency or first interspike interval change as a function of stimulus amplitude (left graph). However, the distribution of interspike intervals during the response changes as a function of stimulus amplitude (right graph).

When considering "neural coding", it is crucial to: (1) define your stimulus of interest; (2) define your time scale of interest and (3) define your sampling step. In some cases, codes can be proposed if a correlation between neural activity and some experimental manipulation can be measured. However, any experimental manipulation (recording, training ..) will necessarily interfere with and change the code.

Exercise: Write an equation that calculates the rate code \( r(t) \) from a spike train \( x(t) \).

Exercise: Look at the following spike train:
If you wanted to show that the spikes “encode” the stimulus variations, what type of rate code would you need? Write and equation.

Exercise. You want to measure stimulus response functions of visual, auditory, taste, olfactory and touch receptors. For each type of receptor, define the axis of variation for the stimulus you would use and the response measure you would use. Draw the hypothetical stimulus-response functions.

Receptive Fields and Tuning curves

At the most basic level, all experiments designed to uncover the neural code can be described as presenting a bunch of stimuli (more or less under the control of the experimenter) and recording the neural responses. Although we will describe some systematic methods below, most often the stimulus set is arrived by a combination of educated guesses and trial and error. The most common form of presenting the results is in the form of a response function – a graph where the mean number of spikes is represented as a function of the particular parameter value associated with each stimulus.

The receptive field of a sensory neuron is a region of space in which the presence of a stimulus will alter the firing of that neuron (from Wikipedia). Sometimes the receptive field refers to a primary sensory neuron and the term Tuning Curve is used for secondary neurons whose tuning curve may be the result of interactions with other neurons as well as the variations in stimulus.

We have spend the last few lectures talking about neurons and action potentials, how they are produced and what they could "represent". Today we will look at some examples of "tuning curves", or how neurons respond to systematic variations in a given stimulus.

As noted before, neurons are often thought to transmit information via "all-or-nothing" events called action potentials or spikes.
We talked about how different types of receptor neurons are activated by sensory stimuli when the stimuli are varied in some systematic way. In these examples, the nature, or quality, of the activating stimulus was kept constant, and the amplitude was varied (cold receptors: amplitude of coldness, olfactory receptors: concentration of stimulus).

In many cases, a relationship between a neuron's activation and the stimulus parameters can be detected.

For example, when neural activity in the olfactory mucosa, bulb or cortex is recorded in response to varying chemical stimuli, response profiles of neurons with respect to chemicals can be constructed.
In this example, the variable parameter of the stimulus is not its amplitude (concentration), but its quality (chemical structure), namely the number of carbons in straight chain aliphatic aldehydes. From these recordings, one can construct a response profile of this neuron for aliphatic aldehydes:

One might say that this neuron is "tuned" to respond to aliphatic aldehydes of chain length 6 and 7 (maximal response), or that its "receptive field" is around chain lengths 6 and 7.

In many experiments, stimulus parameters can be varied with arbitrarily low resolution (for example: wavelength for visual or auditory stimuli). In those cases, smooth response profiles can be constructed. A common example for a response profile is that of spatial orientation tuning in the visual cortex.

A simple way of characterizing the response of a neuron is to count the number of action potentials fired during the presentation of a stimulus. This approach is most appropriate if the parameter characterizing the stimulus is held constant during the trial. If one repeats a trial many times, and then averages the number of action potentials fired (over, in theory an infinite number of trials), and divides by trial duration, one obtains the average firing rate \( r \). The average firing rate is written as \( r = f(s) \) and is called the neural response tuning curve. The functional form of a tuning curve depends on the parameter \( s \) used to describe the stimulus.

The next figure shows extracellular recordings in the primary visual cortex (V1) of a monkey. While these recordings were being made, a bar of light was moved at different angles across the region of the visual field where this cell responded to light (neurons in the visual cortex respond to light stimuli in particular areas of the retina, called their receptive field). One can see in the figure that the number of action potentials depends on the angle of orientation of the bar.

The data has been fitted by a response tuning curve of the form:

\[
f(s) = r_{\max} \exp\left(-\frac{1}{2}\left(\frac{s - s_{\max}}{\sigma_f}\right)^2\right),
\]

where \( s \) is the orientation of the bar, \( s_{\max} \) is the orientation angle evoking the maximal response rate \( r_{\max} \), and \( \sigma_f \) determines the width of the tuning curve.
Exercise (a) Construct a tuning curve for the following experiment. You are recording from a visual neuron on the thalamus. The cell has a spontaneous firing rate of 10 Hz (spontaneous firing rate is how much the cell fires when NO stimulus is applied). You are moving the stimulus on a 5x5 grid and record the following average numbers of spikes at each location. It's not easy to draw such a 3-dimensional tuning curve, so be creative.

<table>
<thead>
<tr>
<th></th>
<th>5</th>
<th>10</th>
<th>10</th>
<th>10</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>10</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>10</td>
</tr>
<tr>
<td>5</td>
<td>10</td>
<td>15</td>
<td>20</td>
<td>15</td>
<td>10</td>
</tr>
<tr>
<td>5</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>5</td>
<td>10</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
</tbody>
</table>

(b) How could the stimulus create a spiking response that is less than the spontaneous rate?

Tuning curves can be used to characterize the selectivity of neurons in cortical areas to a variety of stimulus parameters. For example, tuning curves can be measured for neurons in motor areas, in which case the average firing rate is expressed as a function of one or more parameters describing motor action. The next figure shows an example of extracellular recordings in primary motor cortex in monkeys that have been trained to reach in different directions. The firing pattern of the cell, in particular spike rate, are correlated with the direction of arm movement.

The data points from these recordings have been fitted by a tuning curve of the form:

\[ f(s) = r_0 + (r_{\text{max}} - r_0) \cos (s - s_{\text{max}}) \]

where \( s \) is the reaching angle of the arm, \( s_{\text{max}} \) is the reaching angle associated with the maximal response \( r_{\text{max}} \), and \( r_0 \) is an offset or background firing rate that shifts the tuning curve up from the zero axis.
Interestingly, sometimes the neurons receptive fields or tuning curves are LESS detailed or precise – have less resolution – than the behavioral responses animals can make in response to these stimuli. We will talk more about that later.

Exercise. Give examples of tuning curves in (a) sensory, (b) cortical or (c) motor neurons you know about.

**What might be the advantage of such broad tuning curves?**

One could imagine three basic scenarios depicted in the figure below: (a) local coding, (b) scalar coding or (c) vector coding (I am using the terminology from Churchland and Sejnowski here, not a general terminology). Imagine that the neural system "wants" to encode the direction of arm movement with a resolution of 10 degrees. One possible way of doing that is depicted in (a): local coding, in which individual neurons are dedicated to respond to a given direction (also often called "labeled line coding"). A second possibility, (b) would use a single neuron which responds with different firing rates to variation in angle (scalar coding), or (c), in which several neurons respond with variations in firing rates to variations of angle over a certain range (vector coding, ensemble coding).

For example color coding at the photoreceptor level is achieved by "vector" coding via three types of neurons with broad receptive fields with respect to wave length.
Exercise. Determine a “rule” constructed from these three tuning curves that would allow you to know intermediate light wave length.

Let’s imagine we record from motor cortex from a monkey. We ask the monkey to move its arm around. We find neurons that have “tuning” curves with respect to different movement angles.

As we have seen before, the data was fitted with an equation of the form: \( f(s) = r_0 + (r_{\text{max}} - r_0) \cos (s - s_{\text{max}}) \), to describe the activation as a function of the movement angle. When we record, we find tuning curves that cover the range of movement of the animal. The graph below depicts hypothetical tuning curves of a lot of seven neurons that were recorded, each had a different maximal response angle.
For the sake of this exercise, lets imagine that there are only THREE primary neurons responding to angle of movement, and that the curves depicted above of constructed in secondary neurons from these primary responses.

How could these tuning curves arise from a few angle-sensitive neurons that project to the neurons we are recording from?

Lets assume that the three types of primary neurons have maximal responses to left (0°), up (90°) and right (180°) arm movements.

\[
\begin{align*}
x_{\text{left}} &= \text{Const} \times \cos(\theta) \\
x_{\text{up}} &= \text{Const} \times \sin(\theta) \\
x_{\text{right}} &= -\text{Const} \times \sin(\theta)
\end{align*}
\]

Now imagine that these neurons with activations \(x_{\text{left}}, x_{\text{up}}, \) and \(x_{\text{left}}\) project to postsynaptic neurons with linear, excitatory synaptic connections. These synaptic connections have variable connection strengths (synaptic weights) \(w\). Thus, the inputs to a given postsynaptic neuron would be calculated by:

\[
I_{\text{post}} = w_{\text{post,left}}x_{\text{post,left}} + w_{\text{up}}x_{\text{up}} + w_{\text{right}}x_{\text{right}}.
\]

In the linear case, the output activation \(x_{\text{post}}\) of the postsynaptic neuron equals its total input \(I_{\text{post}}\).

For example, one neuron may receive inputs from \(N_{\text{left}}\) and \(N_{\text{up}}\):

Input = \(0.6 \times x_{\text{left}} + 0.4 \times x_{\text{up}}\) with different strengths (0.4 and 0.6). Then its total input is given by: \(0.6 \times \cos(\theta) + 0.4 \times \sin(\theta)\) and its output activation would be equal to the total input:
Thus, linear combinations of 3 large receptive fields can yield additional receptive fields with arbitrarily low resolutions! If a threshold is applied to these neurons (non-linear case), the output activation is a non-linear function of the total input.

Then the receptive fields look very much like those recorded in motor cortex:
and the width of the receptive fields depend on the value of the threshold:

Let’s look back to our presynaptic neurons which respond preferentially to specific angles of arm position. One of the problems with this idea is that the postsynaptic tuning curves relies on the fact that the presynaptic neurons transmit graded "negative" activity, which is a very unrealistic assumption. In the more realistic assumption, the presynaptic neuron would have an output activation threshold, i.e., it would transmit only positive output values. $x_{\text{left}} = F(\text{const} \cdot \cos(\theta))$, $x_{\text{up}} = F(\text{const} \cdot \sin(\theta))$ and $x_{\text{right}} = F(-\text{const} \cdot \cos(\theta))$ with $F(x) = 0$ if $x \leq 0$ and $F(x) = x$ if $x > 0$. In that case:
One way around this problem would be to add an "offset" (constant positive amplitude) to the activation values of these neurons.

A second, slightly more sophisticated circuitry is to assume that neurons come in opposing pairs, and the connections from these pairs are arranged in a push-pull arrangement, i.e. if a neuron receives an excitatory connection from one neuron, it receives an inhibitory connection from the other. For example, we can recover the linear picture we saw before if we assume that our neuron receives an inhibitory input from a leftward neuron that is the same strength as the excitatory input from the rightward neuron.
There is certainly plenty of evidence for push-pull arrangements. In the visual system, retinal ganglion cells come in both "On" and "Off" subtypes. For certain types of visual cortical neurons, it has been shown that in locations where bright spots elicit excitation, darks spots elicit inhibition and vice versa. There are still plenty problems with this picture! First, truly linear responses would necessitate well balanced pairing and all the neurons in question would have to have effective thresholds near zero. Yet another issue that has been extensively discussed is "magnitude invariant" tuning. Neurons that compute with the type of equations we have described (called the "dot product" in mathematics), reflect the both the nature and the amplitude of the input in their output activation.
Exercise. (a) Create a network that can resolve 9 different colors from the three color tuning curves. Write down the equations and define what colors would be approximately resolved. (b) Now determine what would happen in your network if the blue pigments disappeared.

How can movement direction be encoded precisely by neurons that are so broadly tuned? Georgopoulos and colleagues proposed the idea that direction of movement is not determined by the activation of single neurons but by the net action of a broad population of neurons. As a simple hypothesis for how information of many directionally-tuned cells can be combined, these researchers suggested the construction of a population vector: a unit vector pointing toward the preferred direction of each cell is weighted by the firing rate of the cell, and all those vectors are summed. The resulting vector is called the population vector. As we have seen before, the data was fitted with an equation of the form: \( f(s) = r_0 + (r_{\text{max}} - r_0) \cos(s - s_{\text{max}}) \), to describe the activation as a function of the movement angle.

In this equation, \( r_{\text{max}} \) is the maximal response, and \( s_{\text{max}} \) is the angle of movement at which this maximal response occurs for a given neuron.
The activity of each cell is represented by a vector of angle $s_{\text{max}}$, the length of which is proportionate to its firing frequency [Appendix: Mathematical aside]. The figure below shows a cell whose maximal response $r_{\text{max}}$ occurs at an angle $s_{\text{max}} = 135^\circ$. The graphs show the representations of the neurons activity for 3 different movement angles. For a given neuron, different angles are represented by vectors spanning the same angle but of varying lengths:

For a given angle, the activities of different neurons are represented as vectors of varying angles and length:
The sum of the individual neurons' vectors, called the population vector, indicates the direction of movement represented by the activity of a given population of neurons. One can easily see that the angle of the population vector can be independent on the movement magnitude (assuming that the magnitude is within all the neurons' linear amplitude-response curve).

\[ \text{smax}(N_1) = 40^\circ \]
\[ \text{smax}(N_2) = 110^\circ \]
\[ \text{smax}(N_3) = 135^\circ \]
\[ \text{smax}(N_4) = 180^\circ \]
\[ \text{smax}(N_5) = 230^\circ \]

This vector has the interesting property that it rotates in anticipation of movement, inspiring us to believe that we are reading the monkey's intentions. From a quantitative point of view, the population vector will generate systematic errors if one records from neurons that do not uniformly cover the whole range of movement.

You can think of it as each neuron voting as to the most likely angle: if the neuron has a high firing rate, it gets a high vote, if it has a low firing rate, it gets a low vote.

**Exercise:** Think of possible practical issues with this type of code.

**Exercise:** You recorded from three neurons during a monkey’s movement. You have approximated each response curve with a cosine function. Each neuron’s response is normalized to its OWN maximal response.
You now use your recordings to predict what the monkey is going to do. The numbers I will give you will be in normalized firing rate.

(a) You record the following: A = 1.0, B = 0.0, C = 0.0. What is the best estimate for the angle of movement?

(b) You record A = 0.9, B = 0.6. Use three different methods to estimate the angle of the movement from these numbers. Discuss which is best and why.

(c) You record A=0, B = 0.6 and C = 0. What is your best estimate? How accurate do you think this could be?

(d) You notice that all your predictions are off by several degrees. State several reasons why this could be the case.

(e) How well would a labeled line vs population code decoding strategy work in this case?

(f) What would happen if you could increase the number of neurons by 100 fold?

If you want more god information abut colorvision, visit http://www.firelily.com/opinions/color.html

In human color vision, it is the relative activation of the three types of cones that represents different colors. Each type of cone is activated by a given wavelength, the result is a vector in the three-dimensional space (red, green and blue). The sensitivity curves for the cones are centered around wavelengths of approximately 425 (blue), 525 (green), and 625 (red)nm, respectively, but these sensitivity curves spread out and overlap. Understanding the overlap is crucial. CIE, the International Commission on Lighting, has established three wavelengths that correspond to primary colors (700 nm for red, 546 nm for green, and 436 nm for blue); these are used for color matching.

Yellow light ~ 590 nm

The population vector constructed from the firing of red green and blue centered cells can code for spectral colors in a continuous fashion.

(A spectral color is one which the eye can perceive based on a single wavelength of light. Violet is a spectral color, but purple (a combination of red and blue, whose corresponding cone sensitivity curves do not overlap) is a non-spectral color. White is also a non-spectral color; it is perceived when all three cone types are stimulated by light.)

The most commonly recognized form of "color-blindness" is the sex-linked red-green perception deficiency. Note that even the term "red-green" is misleading; the defect is always with one or the other, but with side effects, and that there are four distinctly different defects. The cone cells, which are the cells
in the retina which detect color, each contain a light-sensitive pigment which is sensitive over a range of wavelengths. The sensitivity curve is roughly bell-shaped. Genes contain the coding instructions for these pigments, and if the coding instructions are wrong, then the wrong pigments will be produced, and the cone cells will be sensitive to different wavelengths of light. The colors that we see depend on the interaction between the sensitivity ranges of those pigments and the brain's expectations of those sensitivity ranges. When those expectations don't match the realities of the cones, perception of color suffers.

When the red and green cones both contain erythrolabe, the pigment sensitive to the red portion of the visible spectrum, the only sensitivity in the green range is from the "overlap" segments of the sensitivity curves. The green-sensitive cones are active, but they respond to the red portion of the spectrum; the brain will respond to this as if the green-sensitive cones were responding to green light. The population vectors for red and yellow light now point in the same direction: red and yellow produce similar firing patterns.

As shown in the examples above, tuning curves are often very broad and unspecific. In general, sensory neurons are more broadly tuned than one would expect from the behavioral resolution one can measure in animal experiments. For example, individual olfactory receptors in rats and mice are activated by ~30% of all odorants, nevertheless, animals can clearly differentiate between these different odorants in a very specific and detailed manner. Recordings in the olfactory bulb suggest that inhibition between principal cells can shape their response curves (or tuning curves).
Tuning “tuning curves”

In many sensory systems, the tuning curves of neurons are not identical to the receptive fields projected to these neurons by sensory neurons. They may be more narrow, include inhibitory parts of the curve, they may be wider or more separated from each other. There are a number of processes that “tune” tuning curves, these include interactions between neurons such as inhibition, excitation and feedback interactions. As we noted before, sensory receptive fields are often broad and relatively non-specific, for example frequency tuning curves in the auditory nerve can span a large range of frequencies.

A common neural circuit that creates more narrowly tuned neurons and also enhances contrast is “lateral inhibition”. In this mechanism, inhibitory neurons which receive the same afferent (from sensory neurons) input that excitatory neurons do, inhibit a range of neighboring neurons. This results in a general dampening of activity but also create more narrow tuning curves.

Exercise: Assuming linear interactions and all synaptic weights being zero, construct the approximate resulting receptive fields for these neurons and this network:
Excitatory neurons N1, N2, N3

Inhibitory neurons I1, I2, I3

Exercise: Look at the recordings below. Think about what you could learn from these and what additional information you would need to get useful information from this experiment.

Contrast

Because of broad receptive fields and tuning curves, neural circuits are thought to enhance “contrast” or increase the difference between sensory stimuli in order to make them more easily recognizable, their features more salient, more distinguishable from each other.

Exercise: (a) You have a number of chairs and a number of tables. List features these have in common. Now list features that differentiate them. Write a list of yes no questions that would allow you to decide (i) if an object does belong to either category and (ii) if it is a chair or a table. Now find some examples that would not easily be classified. Create a neural network with a layer of feature detectors (respond to a specific feature), a layer of inhibitory neurons and one or two more layers of neurons including a layer of output neurons. At the output, you want to know if the object you detect is a chair or a table. Think about which features you want to suppress (inhibit) and which you want to have compete against each other.

So far we have mainly looked at receptive fields and tuning curves and we talked about how one can determine the stimulus given a limited number of broad tuning curves. In the following section, we will not consider the stimulus identity or quality the important question, but ask how well a system could discriminate between two stimuli.

Look at the spike trains below in response to different stimuli.
Each mitral cell responds to a number of aliphatic aldehydes with a range of carbons in the chain. If one were to consider only two mitral cells, then their responses to each odor would span a vector in a two-dimensional space of mitral cell activity [Appendix: Mathematical aside]. Within this space, odorant representations (as depicted by their activity-vector) can be more or less similar (overlap) or different (distance between vectors). It has been shown that distance measures computed from these types of measurements are sufficient to predict perceptual similarities between pairs of odorants.

Imagine that the inhibition between mitral cells, which is the source of the "negative" responses to odorants surrounding the responses with higher firing rates, is blocked or decreased. Each mitral cell now responds to more different odorants; as a consequence, the overlap between the odor representation increases (or the distance between the representation decreases).
Exercise: What happens to the distance measure and dot product measure if the vectors are “normalized” first (this means they all have length 1.0 and span the unit circle).

**On cortical neural circuits**

So far, we have talked about tuning curves, ways to construct tuning curves, how to decode the information represented in these tuning curves and we have seen that tuning curves are plastic and change as a function of experience and changes in the animal's environment. Cortical structures use this information to detect important signals, compare them to what has been seen and learned about before, decide on actions to take etc. While not very much is known yet about how exactly these computations are performed, we do know quite a bit about how cortex is structured and what the computational implications of these circuits could be.

Cerebral cortex in mammals has several common features:
1. Cell bodies of cortical structures are organized in layers most of which lay parallel to the surface of the brain.
2. The layer closest to the surface contains no neurons, is called the molecular layer or layer a and is an area of synaptic interactions.
3. At least one layer contains the cell bodies of the principal neurons, or pyramidal cells, which extend long dendrites (apical dendrites) into layer 1.

Hippocampus and olfactory cortex have a single cell body layer whereas neocortex has multiple cell body layers. Cell types distinguish principal cells or pyramidal cells with axons which project outside of the structure in question and intrinsic neurons, or local interneurons, which do not project outside of the layer in question. All cortical structures integrate at least the following computational elements:
1. Afferent input of information from "lower" areas (for example thalamus -> primary visual cortex, or olfactory bulb -> olfactory cortex, or primary visual cortex -> secondary visual cortex);
2. Central, associative, inputs from "higher" areas (for example hippocampus -> olfactory cortex, secondary auditory cortex -> primary auditory cortex);
3. Intrinsic interactions between pyramidal cells (association fibers);
4. Local inhibitory circuits (feedforward and feedback inhibition);
5. Neuromodulatory inputs (for example from the locus coeruleus, substantia nigra etc).

Of course, many cortical areas (association cortex) are not associated with a particular sensory system but integrate inputs from all sensory systems, from motor systems and from other cortical areas.

We will illustrate the common principles using the olfactory cortex, because of its simplified, single cell body layer structure.
What do these different elements do?

1) Afferent input. Pyramidal cells and local interneurons receive information about the sensory worlds on their dendrites in layer 1a of the molecular layer. In olfactory cortex, mitral cells in the olfactory bulb spike in response to olfactory stimuli and excite pyramidal cells in the olfactory cortex. Primary visual cortex receives information from neurons in the visual areas of the thalamus. In the simplest scenario possible, the receptive fields of these neurons are transmitted to neurons in the cortical area in question.

2) Local interneurons providing *feedforward* inhibition. These interneurons are considered to receive the same afferent information than the pyramidal cells surrounding them. They are also believed to synapse onto the apical dendrites of these pyramidal cells, as well as on some neighboring pyramidal cells. As a
consequence, these interneurons provide inhibitory input with similar information than the excitatory input provided by the sensory inputs. This inhibitory input can lead to suppression of the weaker activated pyramidal cells, resulting in what is called "contrast enhancement". This is an example of “lateral inhibition”.

![Diagram of neural network](image)

If the inhibitory networks are arranged in such a way that they receive input from the pyramidal cells as well as from the afferent input, a "winner-take-all" scheme in which only the most active pyramidal cell stays active can be implemented. This is because now each pyramidal cell receives inhibition that is proportional to other cells’ activation values; as a consequence, the most active pyramidal cell can suppress the activity in all others.

3) Pyramidal cells activate other pyramidal cells in the network via "association fibers". These fibers are thought to be the main substrate for memory formation. In the simple scheme provided here, pyramidal cells activated by the same afferent inputs can boost each others activity and thus enhance the above described contrast enhancement effect (amplification).

![Diagram of neural network](image)

4) A second layer of inhibitory neurons, often called feedback inhibitory neurons (because they do not receive feedforward input from afferent inputs) can help limit the activity of the pyramidal cells as well as change the timing of the action potentials. Once a positive feedback loop (excitatory interactions among
pyramidal cells that are active) has been added, it is important to have a layer of inhibition that keeps the activity of these neurons bound.

5) Feedback inputs from other brain regions usually interact with pyramidal cells in layer 1b, but can also synapse onto basal dendrites in layer III. The role of these feedback inputs is not completely understood. It is thought that they can mediate associations between sensory modalities, associations between sensory input and actions or reward associated with these inputs etc.

6) Neurmodulatory inputs act on cellular and synaptic properties within cortical networks. Neurmodulators can change conductances, depolarize cells, change synaptic transmission etc. They are often associated with a particular state (fear, hunger), motivation or attention.

Exercise. Let’s work through an example to get used to writing and reading equations.

1) We will start by constructing a network of neurons in motor cortex receiving inputs from presynaptic neurons $s$ with the following tuning curves as a function of the angle $\alpha$ of arm movement and $\alpha$ threshold $\theta$:

$$x_{sj} = F(\cos(\alpha - \text{offset}_i), \theta)$$

where $\text{offset}_j = 0, -30, -60, -90$ and -120.

Now, we will create 10 postsynaptic neurons (5 excitatory pyramidal cells and 5 inhibitory local interneurons), each receiving presynaptic neurons, assuming all synaptic weights $w = 1$:

$$I_{pj} = x_{sj}; I_{inj} = x_{sj}$$

with $I_{pj}$ being the input to pyramidal cell $j$ and $I_{inj}$ the input to interneuron $j$. These are linear threshold neurons with continuous output for now, so $x_{pj} = I_{pj}$ and $x_{ij} = I_{inj}$ if $I > \theta$ and $x = 0$ and $x_{ij} = 0$ if $I < \theta$. In vector notation:

$$\begin{bmatrix}
I_{p1} \\
I_{p2} \\
I_{p3} \\
I_{p4} \\
I_{p5}
\end{bmatrix} = 1.0 \cdot \begin{bmatrix}
x_{s1} \\
x_{s2} \\
x_{s3} \\
x_{s4} \\
x_{s5}
\end{bmatrix}; \begin{bmatrix}
I_{in1} \\
I_{in2} \\
I_{in3}
\end{bmatrix} = 1.0 \cdot \begin{bmatrix}
x_{s1} \\
x_{s2} \\
x_{s3}
\end{bmatrix}; x_{pi} = F(I_{pi}, \theta); x_{ini} = F(I_{ini}, \theta)$$

We obtain the following response curves as a function of the angle $\alpha$:

Now we will add “feedforward” inhibitory connections in which each interneuron inhibits the two “neighboring” pyramidal cells (i.e., those with tuning curves most similar, but not identical, to its tuning curve). For illustration purposes, only the middle three pyramidal cells will be plotted. Since all neurons are linear threshold neurons, and synapses are linear and synaptic weights equal to 1.0 or -1.0, we now obtain:

$$I_{pj} = x_{sj} - x_{(j-1)} - x_{(j+1)}.$$ The inputs and outputs of the interneurons don’t change.

---

1 Notations: $I$: input to a neuron; $x$: output from a neuron; $j$: neuron index; $\theta$: threshold; $\alpha$: movement angle; $s$: sensory neuron; $p$: pyramidal neuron; $in$: interneuron.
In vector notation:

\[
\begin{bmatrix}
I_{p1} \\
I_{p2} \\
I_{p3} \\
I_{p4} \\
I_{p5}
\end{bmatrix} = 1.0 \begin{bmatrix}
x_{s1} \\
x_{s2} \\
x_{s3} \\
x_{s4} \\
x_{s5}
\end{bmatrix} + \begin{bmatrix}
0 & -1 & 0 & 0 & 0 \\
-1 & 0 & -1 & 0 & 0 \\
0 & -1 & 0 & -1 & 0 \\
0 & 0 & -1 & 0 & -1 \\
0 & 0 & 0 & -1 & 0
\end{bmatrix} \begin{bmatrix}
x_{i1} \\
x_{i2} \\
x_{i3} \\
x_{i4} \\
x_{i5}
\end{bmatrix}; x_{pi} = f(I_{pi}, \theta)
\]

The pyramidal cell tuning curves have become much “sharper”, i.e. each pyramidal cell responds to a smaller range of angles because of the lateral inhibition. If we want to simultaneously boost the signal of the pyramidal cells, we can add feedback inhibition of each pyramidal cell onto itself (note the difference in scale):

If we change the threshold of the feedback interaction to be higher, the effect becomes more pronounced:
Even sharper response curves can be obtained if the inhibitory effects are made larger, however, the feedback boost also needs to increase in that case:

With stronger inhibition

and stronger excitatory feedback.

Exercise: Write the relationships between neurons in vector notation for the case with feedback excitation.
Pyramidal cells often receive extensive excitatory feedback from other pyramidal cells in the same layer. The possible function of that feedback will become clear in later lectures in which we talk about associative memories. If we connect the three pyramidal cells together with a low synaptic weight (leaving all other parameters as they are), the following happens: each pyramidal cell has a smaller response at the center of the neighboring pyramidal cells:

![Diagram of neural network with sensory neurons, local interneurons, and pyramidal cells]

**Exercise:** Add the input from neighboring pyramidal cells to the equations.

Feedback inhibition, in which all pyramidal cells feed into a common inhibitory interneuron and then receive inhibitory input from this interneuron can help regulate the activity across the network. So far we have looked at average response over a range of stimuli without taking into time into consideration. Now, let's look at the response to two sequential stimuli over time with the different configurations discussed. We will also make the neurons (except sensory neurons) into probabilistic linear threshold neurons: these fire action potentials with a higher and higher probability as their inputs become larger.

Sensory neurons and interneurons are described by the same equations as before. Pyramidal neurons first receive inputs from the corresponding interneurons. Their output is a probabilistic function of their input: prob (x_{jp} = 1.0) = F (I_{jp}, \theta), i.e., the probability that the neuron emits a spike increases with the input to the neuron.

![Graph of firing probability vs. input]

Choosing two stimuli, \( \alpha = -100 \) and \( \alpha = 50 \), we obtain the following responses over time (again only using the three middle pyramidal neurons):
in which the sensory neuron responses are drawn on the upper graph, interurons in the middle graph and three pyramidal cells on the three lower graphs.

Exercise: Describe how the magnitude of sensory neuron responses was determined.

We now add the same amount of lateral inhibition than in the previous simulations:

One can see that now for each stimulus, a single pyramidal cell responds!

This result can be understood by comparing to the previous graphs: