# Introduction to Neural Networks
U. Minn. Psy 5038

Daniel Kersten

Lecture 2

## More on getting started with *Mathematica*

### Front-end and Notebooks: Organize, outline, document, program
### Kernel: Separate program does the calculation

### Numerical Calculations

Last time you saw how you can do arithmetic. Let's try some other simple operations.

Evaluate 4*3

Compare with 4 3 (i.e. 4 followed by a space, and then 3). Note that 4 3, where a space separates the digits is also interpreted as multiplication.

**Compare (2^.000000000001)^1000000000000 with (2^(1/1000000000000))^1000000000000**

**Front-end stuff**

You can go back and select an expression by clicking on the brackets on the far right. These brackets are features of the user interface and serve to organize text and calculations into a Notebook with outlining features. You can group or ungroup cells for text, graphs, and expressions in various ways to present your calculations. Explore these options under Cell in the menu. You can see the possible cell types under the **Style** menu.

- **By ending an expression with ; you can suppress the output--this is VERY useful later when the output might be a list of a 10,000 neural activity levels!**

```
(3/4)/6;
(3 4)/6;
```

**The most recent result of a calculation is given by %, the one before by %%, and so forth. Try it on the previous two outputs**

### Built-in functions

*Mathematica* has a very large library of built-in functions. They all begin with an uppercase letter and the arguments are enclosed by square brackets. Knowing that, you can often guess the form of a function

**Try taking the logarithm of 8.0**

---

**Did it return log to the base 10 or e? Check the definition by typing ?Log**

---

You can get information more about a function, by clicking on the resulting link more.

**Try ?Exp**

---

**Try ?Plot**

---

If you type two question marks before a function, **??Plot**, you'll get more information. Try it.

**What does the Random[ ] function do?**

---

## Defining functions

Soon, you will use *Mathematica* to model the generic connectionist neuron. Part of the model will require defining a function that suppresses large inputs. Here is an example:

```
squash[x_] := N[1/(1 + Exp[-x+4])];
```

The underscore, **x_ is important** because it tells Mathematica that x represents a slot, not an expression.

Also note that our squashing function was defined with **N[]**. *Mathematica* trys to keep everything exact as long as possible and thus will try to do symbol manipulation if we don't explicitly tell it that we want numerical representations and calculations.



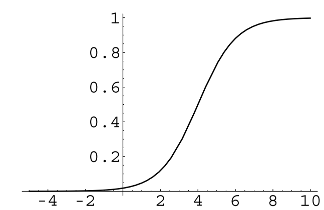
**Define a new function squashedExp[ ] that applies squash to an exponentiated value (i.e. takes Exp[x] as the argument of squash[ ])**

---

## Graphics & more function defintions

Let's plot a graph of the squash function using the syntax we discovered above for $-5<x<10$

```
Plot[squash[x], {x,-5, 10}];
```

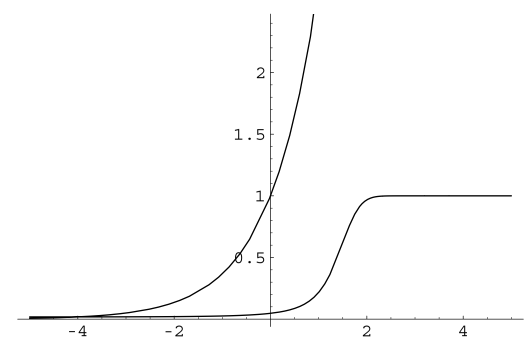



This squashing function is often used to model the small-signal compression and large signal saturation characteristics of neural output.

**Plot squashedExp[ ] for x going from -5 to 5**

---

```
Plot[{Exp[x], squashedExp[x]}, {x,-5, 5}];
```




Even though Exp grows exponentially fast with x (by definition!), squash keeps a lid on it.

### Ask *Mathematica* for the definition of squashedExp[ ]

It can be important to check your definitions like this. One reason is that, as we will see later, *Mathematica* definitions can be built up with multiple constraints. And sometimes you might add to a function unwittingly and it appears to misbehave. You can check your defintion by asking *Mathematica* for it.

### := vs =

Note that when defining a function, we used a colon followed by equals ( := ) instead of just an equals sign (=). When you use an equals sign, the value is calculated immediately. When there is a colon in front of the equals, the value is calculated only when called on later. So we use := for function definition because we need to define the function for later use and evaluation.

### Let's define r1 using :=, and r2 using =

```
r1 := Random[];
r2 = Random[];
```

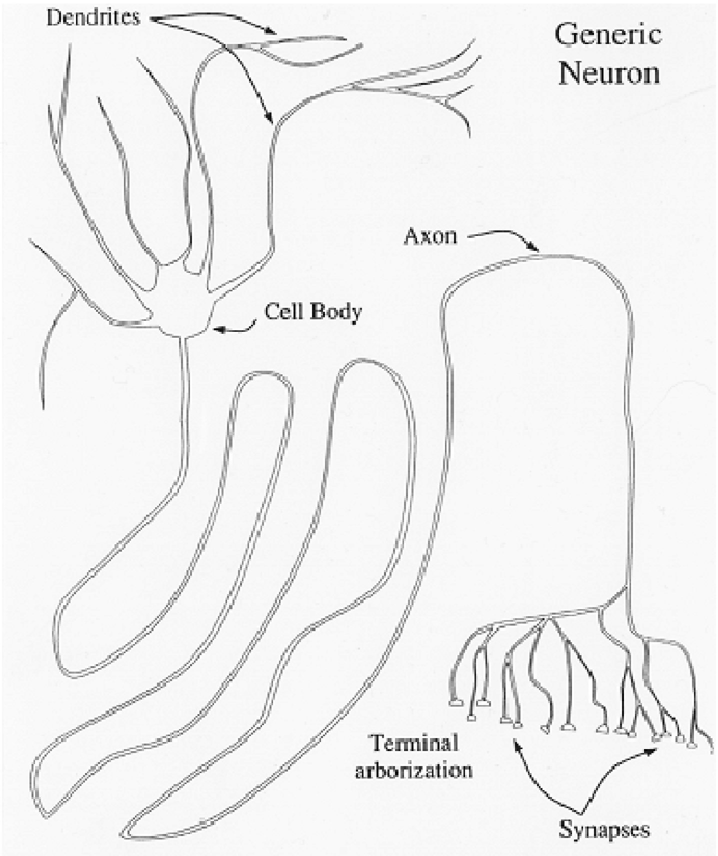


### Now evaluate r1 and r2 three times each. What is the difference between the two definitions?

### Plot squash[x]+r2 for -5<x<5

### Now plot squash[x]+r1 for -5<x<5

## The Neuron - overview of structure




From Anderson (1995)

## Basic Structure

Information flow: dendrites -> (soma -> axon hillock) -> axon -> terminal buds



### ■ Dendrites

The information receiving end of a neuron is a tree-like structure consisting of "dendrites" with special processes or connection sites called synapses. Much computational power is thought to reside in the strength of connections, and in the dendritic tree itself. In this course, we will primarily examine the computational properties of groups of simple neurons, rather than aggregates of dendrites in a single neuron.

Dendrites play the role of wires that convey information through changes in voltage. But they behave rather differently than copper wires. These neural processes are tubes of ionized cytoplasm sitting in a bath of ionized fluid whose ionic composition is not that much different from "seawater". The inside of these tubes during resting state sits at about minus 70 millivolts relative to the outside of the cell. The tubes are on the order of microns in diameter, (but other processes, such as the axons discussed below, can reach half a millimeter or so). And for further comparison, the membranes making up the tubes are on the order of 50 Angstroms (50 x 10-10 meters) thick.

Signal transmission is limited by high resistance of the axoplasm, and high capacitance of the neural membrane. Information transmission consequences of these properties are:

• the voltage potential changes have a short range of influence, with the amplitude decreasing rapidly as one moves away from the synaptic source.

• the signals travel relatively slowly.

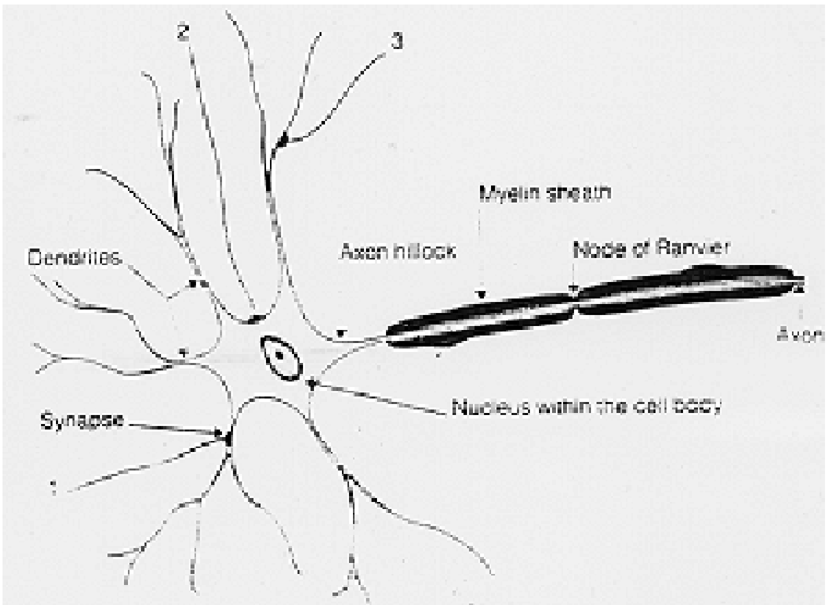
We'll take a quantitative look at these facts shortly.

### ■ Soma (or cell body)

• Integrates dendritic signals

• The storage of electrical charge across the membrane, and the chemical nature of synaptic transmission leads to temporal integration of signals. This observation together with spatial integration of signals from the dendritic tree arriving at the axon hillock will lead to our basic model of the neuron.

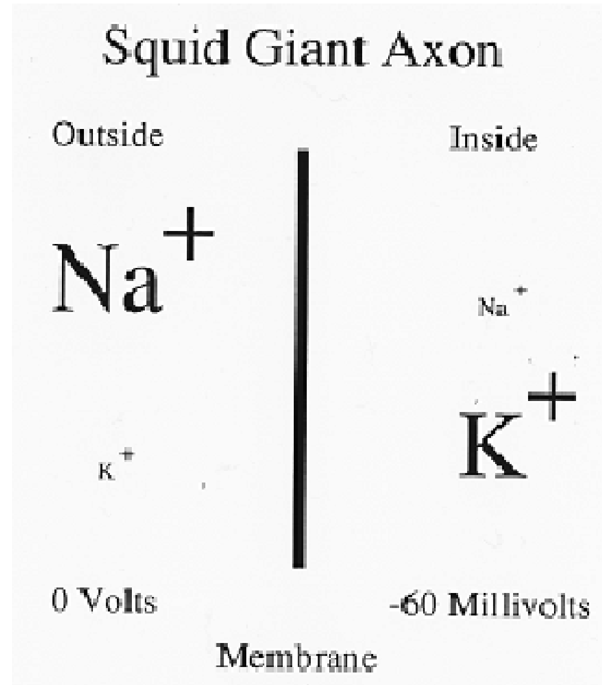
## ■ Axon hillock and axon




**How can the range and speed be increased?**

Certain neurons are equipped with a specialized process called an axon that serves to "digitize" the data into all-or-none responses (voltage changes) called action potentials or spikes. This digitization occurs at the axon hillock near the cell body. There is *passive* or electrotonic conduction along the dendrites up to the axon hillock at which point, if there is a sufficient potential change to reach threshold, an *active* process of depolarization kicks in leading to a spike in membrane voltage. (*Depolarization* means the voltage potential difference across the membrane *decreases*; *hyperpolarization* can also occur, where the voltage difference *increases*).






The action potential signals are carried by rapid (1 msec) voltage depolarizations going from -70 to +40 mV via Na+ influx, and K+ outflow through the membrane. From the axon hillock on, a myelin sheath serves to lower the capacitance and speed up conduction. However it interferes with the regenerative processes that preserve the all-or-none response. At periodic points (Nodes of Ranvier) the myelin sheath is interrupted where high extracellular concentrations of Na+ ions exist with sodium gates. When a small depolarization arrives, this decreases membrane conductance allowing an increased depolarizing influx of Na+, regenerating the spike.

## ■ Terminal arborization and terminal buds

Neurons with axons end in a terminal arborization . The terminal buds make synaptic contacts with the dendrites of subsequent neurons, and we have the beginnings of a neural network. Synaptic contacts can either be electrical or chemical, but more about these later.

# Basic electrophysiology

## Passive properties

Above we noted that the potential maintained by ionic imbalance (excess Na+ outside, and K+ inside). Balance between ionic concentration and electric field forces - determined by the Nernst equation (see Anderson text for a derivation).

We noted two problems: passing a signal over a long distance and with sufficient speed--BIG problem for an organism that has to transmit signals fast over a few feet. Let's take a more quantitative look at these problems that arise from the passive electrical properties of neuronal "electronics".

Model the passive electrical properties as a function of **time** :

• Across a small portion of the membrane modeled by an "RC circuit" where R stands for resistance, and C for capacitance.

We'll see shortly that there is a temporal delay in voltage response characterized by time constant $\tau$ or rise time.

RC-circuits are "low pass temporal filters", i.e. favor signals with low temporal frequencies.

and **space**:

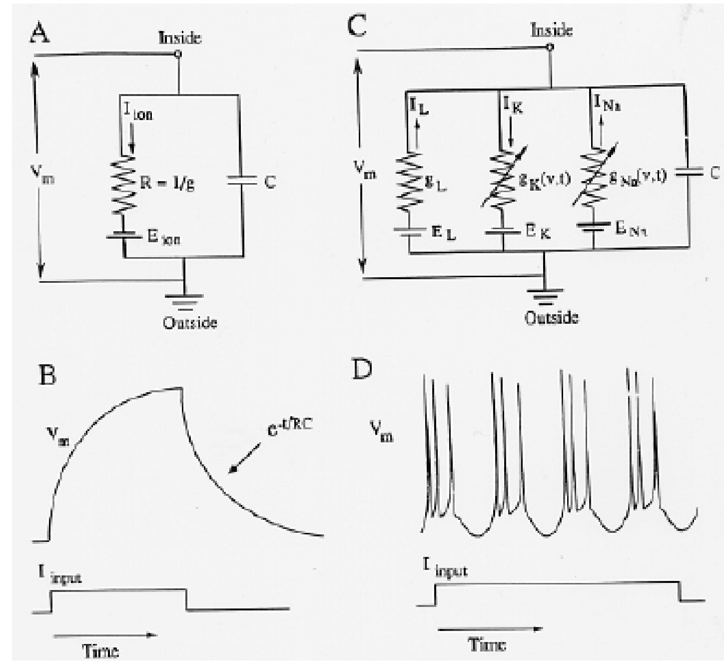
• If we imagine cascading a series of RC-circuits, each connected by additional resistances, we have a discrete model of a section of neural membrane. This kind of model is good for computer simulation. But a continuous model can be solved exactly.

A continous model over time and short lengths is by the "Cable equation" (see pages 25-32 of Anderson)

We'lll see shortly how that the cable equation predicts an exponential drop-off of voltage with distance for constant current. Length constant $\lambda$ ( distance to the 1/e drop-off point or 63% drop) is on the order of millimeters.

By solving the cable equation governing the voltage change over distance and time, we can get a quantitative idea of how voltage drops with distance, and how voltage changes with time--change is not instantaneous.






A. illustrates an RC-circuit at a single point of *passive* membrane. B is the temporal response to a step current input. C illustrates additional variable conductance components that model the electrical processes of spike generation (panel D)--the *active* properties. To model the active properties, one needs a more complicated set of differential equations: the Hodgkin-Huxley equations.

From Segev (1992)

### ■ Solutions of the cable equation

The cable equation is a (partial) differential equation and is given by:

$$V = \lambda^2 \frac{\partial^2 V}{\partial x^2} - \tau \frac{\partial V}{\partial t}$$

where V = V(x,t) is a function of distance x along the membrane and time t.

**Space.** In a later Notebook, we'll see how to use *Mathematica* to find solutions to equations, including differential equations. For now, let's take the steady-state solution for a fixed voltage, V0, at a specific place, say x = 0, and see how the voltage drops away from zero. Steady-state means that the voltage is no longer changing with time, or in another words that $\frac{\partial V}{\partial t}=0$.

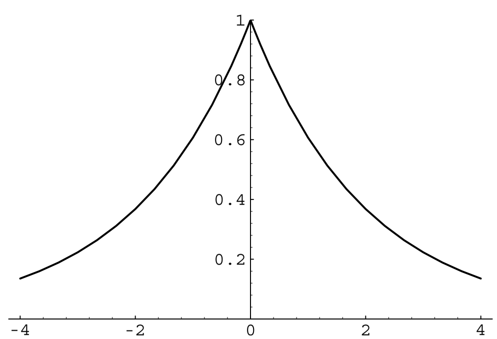
$$V = \lambda^2 \frac{\partial^2 V}{\partial x^2}$$

The solution of this equation (which you can verify be differentiating twice, see exercise below) is a standard result. The voltage drops exponentially. Lambda is the "space constant", which for an axon would be about 3 to 5 mm.

$$V(x) = V_0 e^{-x/\lambda}$$

```
V0 = 1; lambda = 2; (*Space constant*)
V[x_] := V0 Exp[-Abs[x]/lambda];
```

```
Plot[V[x], {x,-4,4}, PlotRange->{0,1}];
```




**Using *Mathematica*'s derivative function D[ ] to verify the solution. Differentiate V(x) twice with respect to x, where V(x) is re-defined over positive x values (it is simpler to treat positive and negative x separately):**

---

```
Clear[V];
V[x_] := V0 Exp[-x / λ];
```



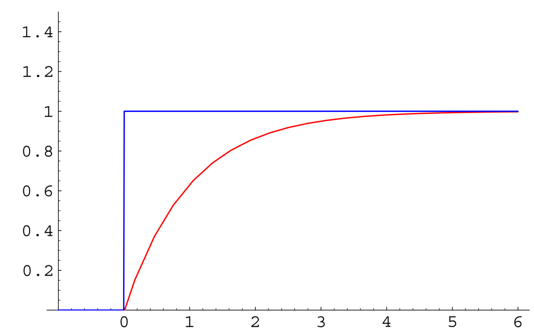
Now let's see how membrane voltage changes with time at a single location by plotting the dynamical solution to the cable equation. We'll assume no space variation. Standard integration techniques can be used to solve this simplified cable equation. Let's assume some initial conditions.

Suppose a 1 volt step is applied (through some resistance to generate a step current change) across the neuron membrane at time t=0. How does voltage change with time at a fixed point x? It doesn't immediately change to 1 volt, but grows gradually. A typical time constant $\tau$ is on the order of 1 or 2 msec.

$$V(t) = 1 - e^{-t/\tau}$$

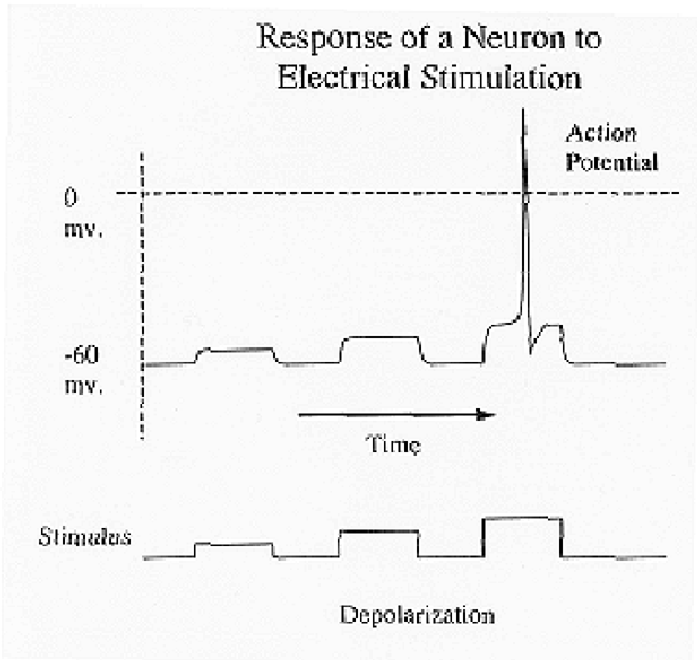
```
τ = 1; (*Time constant*)
V[t_] := 1 - Exp[-t/τ];

Plot[{V[t], UnitStep[t]}, {t,-1,6},
    PlotRange->{0,1.5}, AxesOrigin->{-1,0},
    PlotStyle->{RGBColor[1,0,0], RGBColor[0,0,1]}];
```




(UnitStep is a built-in *Mathematica* function. You could define your own as: **myUnitStepstep[x_] := If[x<0,0,1];** )

## Active (non-linear) properties




Action potentials, spike trains are nature's solution to the problem of fast long distance signalling.

"toilet flush model" - small push of the handle and a little leak, but reach a threshold, and the whole thing goes.

To quantitatively model these voltage changes, one needs to add extra terms to the RC circuit (panels C and D above). The general equations due to Hodgkin & Huxley equations are more complicated that the cable equation.

**Time properties**

Refractory period: absolute, and relative

Absolute refractory period is a brief time (~1 msec) right after the depolarization where no strength of input current would be sufficient for another spike. The ion pumps need time to restore some of the ionic imbalance.

Relative refractory period- threshold gradually lowers with time. One can elicit a spike, but it requires a stronger input signal. This is one of the factors that leads to the idea of frequency coding. A constant step input leads to a series of spikes of a particular frequency (e.g. if the absolute refractory period is 1 msec, we'd expect a maximum spike frequency of 1000 spikes/second), but usually much lower (e.g. 20 spikes/second). To get a faster rate, the input voltage would have to be increased. This leads to the idea of the neuron as a "voltage-to-frequency" converter. But it gets a little more complicated because neurons often show "adaptation" and the firing rate declines for a fixed step input. More on that later.



**Space properties**

..and the solution to the rapid decline in voltage signal over distance due to passive properties?

Action potential at one location provides the depolarization stimulus at a nearby spatial location, travels like a lit fuse down the membrane.

And how about the speed problem?

myelin sheath (insulation reduces leakage)

Nodes of Ranvier, action potentials jump from node to node, up to 30x.

## Synaptic Integration

**Some definitions:**

**pre-synaptic potential** --across the membrane of the terminal of the "transmitting" neuron

**post-synaptic potential** -- across the membrane of the dendrite of the "receiving" neuron

Two types: **excitatory** (make the cell more likely to fire) and **inhibitory** (less likely to fire) post-synaptic potentials

long duration - fast EPSP is 1 to 2 msec rise time and 3 to 5 msecs decay (action potential 1-2 msecs)

synaptic delay - about 500 micro-seconds.

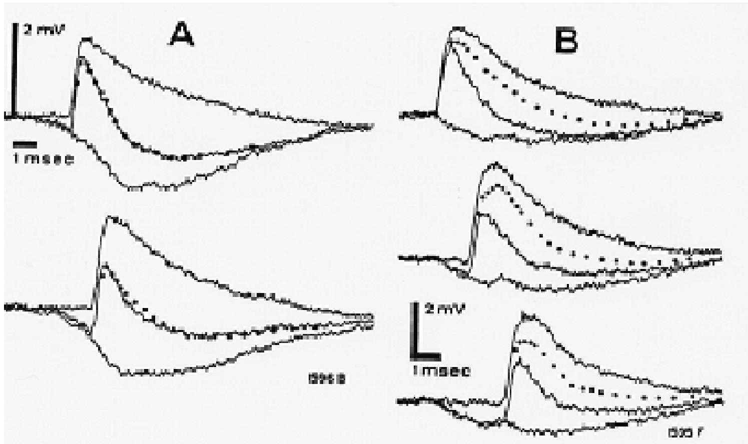
**Interaction of synaptic potentials**

Imagine 1000's of synaptic inputs to a single neuron. How do they interact?

axon hillock

algebraic? sometimes but not always

Our generic connectionist model introduced later will assume linear algebraic summation.

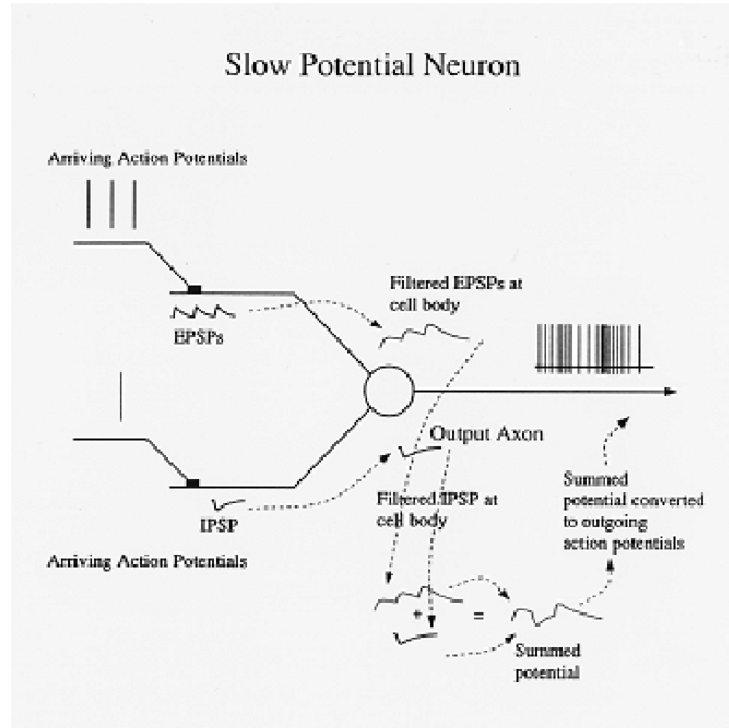



Anderson (1995). Interaction of inhibitory postsynaptic potentials (IPSP) and excitatory postsynaptic potentials (EPSP) can be linear (left column, A) or non-linear (right column, B). From Rall (1967). Dotted line shows linear prediction.

## Qualitative summary of slow potential neuron model

Let's summarize the essential qualitative features of signal integration and transmission of a neuron with what is called the "slow potential model".






Slow potential at axon hillock waxes and wanes (because of low-pass temporal characteristics and the spatial distribution of the inputs) depending on the number of active inputs, whether they are excitatory or inhibitory, and their arrival times.

The slow integrated voltage potential now and then exceeds threshold producing an axon potential.

Further, if the slow potential goes above threshold, frequency of firing is related to size of slow potential.

**Caveat:** Not all signal transmission in neural computation is done through action potentials. For example, of the 6 types of cells in the retina of your eye, essentially 1 type, the ganglion cells,uses action potentials, the others communicate via slow potentials.

But spike generation isn't a strictly deterministic process. There is "n**oise" or random fluctuation that can**

ion channels open and close probabilistically, quantized

neurotransmitter release in discrete packages - Poisson

sensory receptors can produce spontaneous signals

Over long distances spike train frequency is roughly like a Poisson process (better--an interval Gamma distribution) whose mean is modulated by the already noisy slow potential.

## References

Koch, C. (1998). Biophysics of Computation : Information Processing in Single Neurons . Oxford: Oxford Univ. Press.

Koch, C., & Segev, I. (1998). Methods in Neuronal Modeling : From Ions to Networks. Cambridge, MA: MIT Press, 671 pages.

Segev, I. (1992). Single neurone models: oversimple, complex and reduced. Trends in Neuroscience, 15(11), 414-421.