CSDS 440: Machine Learning

Soumya Ray (he/him, sray@case.edu)

Olin 516

Office hours T, Th 11:15-11:45 or by appointment

Zoom Link

Recap

- To minimize a function, the J____ has to be ____ and the H_____ has to be p____ d____.
- In iterative optimization, we first g___ the solution. Then until c____, we choose a d___ and a s___. We create the next iterate by moving our g___ in (what way). We stop when _____.
- Gradient ascent takes a small step in the ___ direction.
- Why does the step need to be small?
- Convergence is usually slow with gradient ascent/descent, because _____.
- The Newton Raphson method approximates the function locally as a q_____.
- This provides faster r___ of c____ than gradient descent, but may d____ in some cases if started f___ from the solution.
- What is a quasi-Newton method?

Review of Calculus and Optimization

- Optimization
- Artificial Neural Networks (Ch 4, Mitchell)

Iterative Optimization (One variable)

- *Initialize* the solution candidate with a *random guess*
- Until we find the maximum or minimum ("convergence") loop:
  1. Choose a *direction* d
  2. Choose a *stepsize* λ
  3. Move the current guess by λ in the d direction
  4. Check: are we at a minimum/maximum?
     - By evaluating $\frac{df}{dx} = 0$ at the current guess, and ensuring $\frac{d^2f}{dx^2} \geq 0$ (if minimum) or $\frac{d^2f}{dx^2} \leq 0$ (if maximum)
     - In a computer, always check $\left|\frac{df}{dx}\right| \leq tolerance$ (a small quantity such as $1\text{e-}6$)

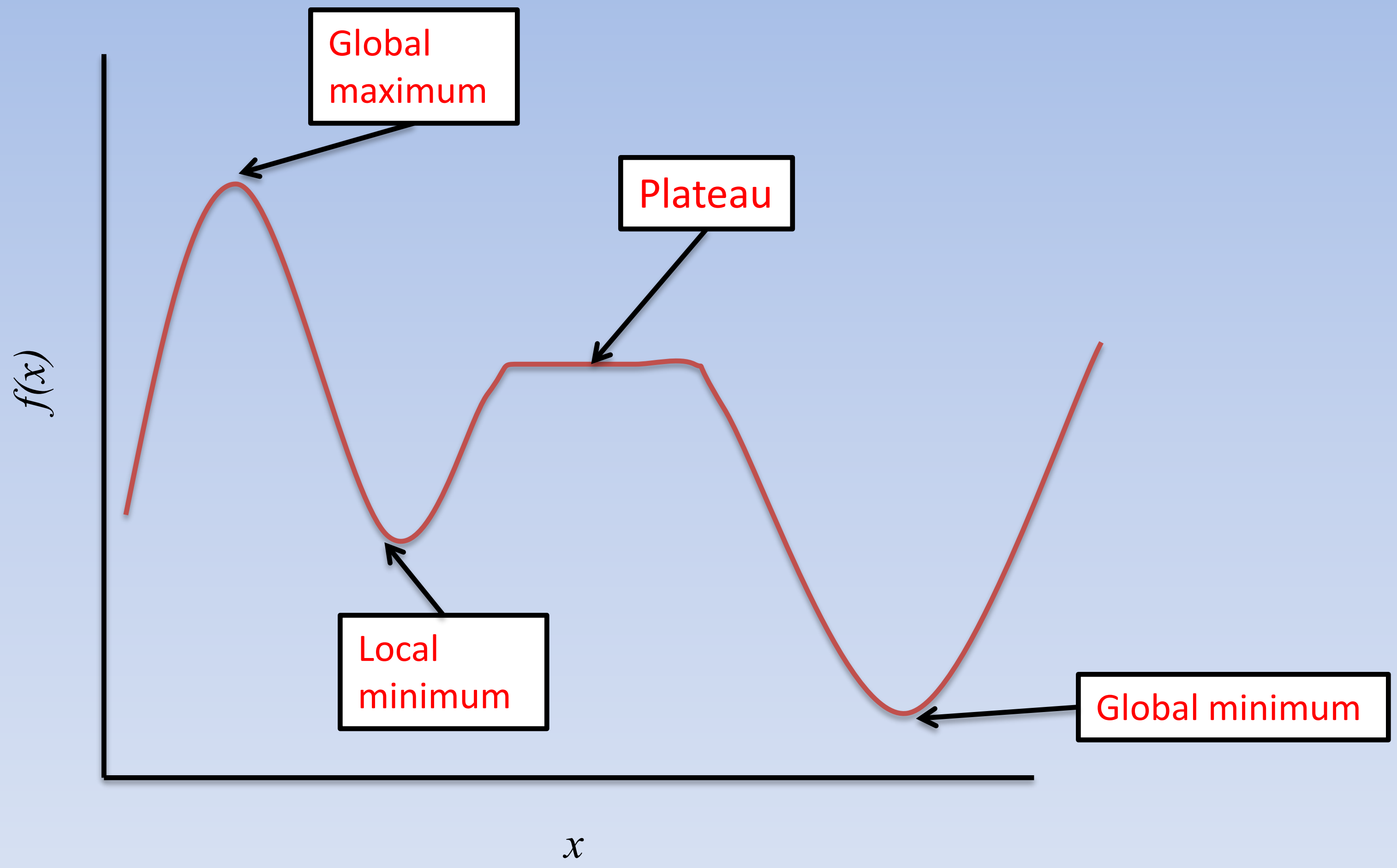
Different optimization algorithms will do these steps differently

Random Restarts

- The solution we get from Gradient Ascent/Descent depends on the initialization

- One way to make it less dependent is to use *random restarts*
  - Run multiple gradient descents with *different, random initializations* and keep the best overall solution
  - Can be done in parallel

Characterizing Solutions



Global maximum

Plateau

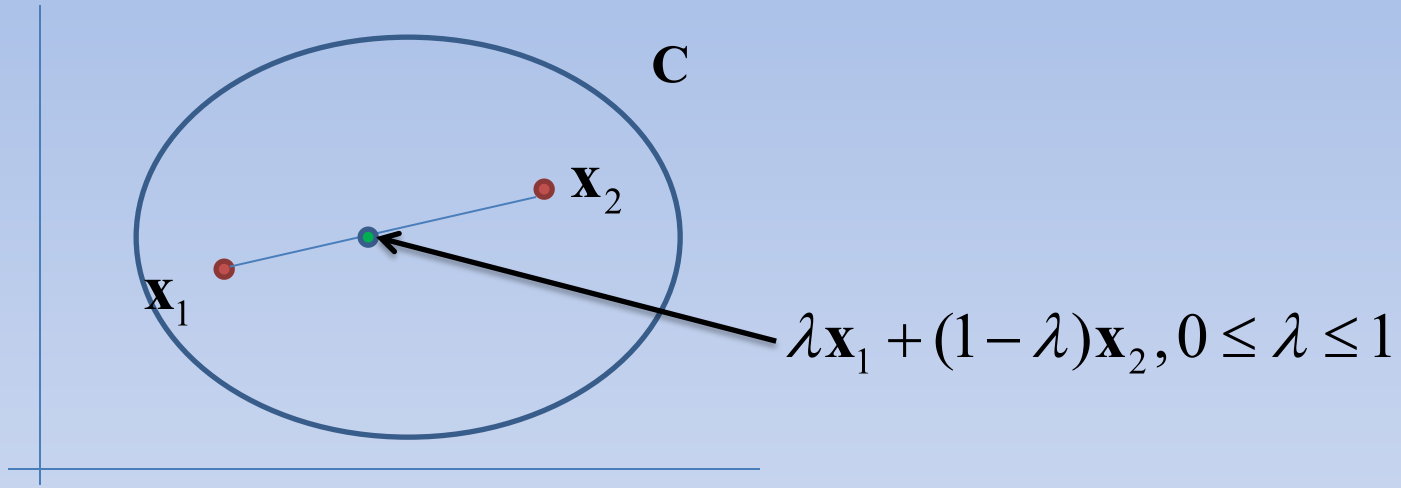
Local minimum

Global minimum

Local and Global Optima

- A global minimum for a function is a point x where $f(x) \leq f(x+u)$ for all u
- A local minimum is an x where $f(x) \leq f(x+u)$ for all $|u|<\varepsilon$, for some positive ε
- In general there is no algorithm that is guaranteed to find the global optimum of an arbitrary function in a finite number of steps

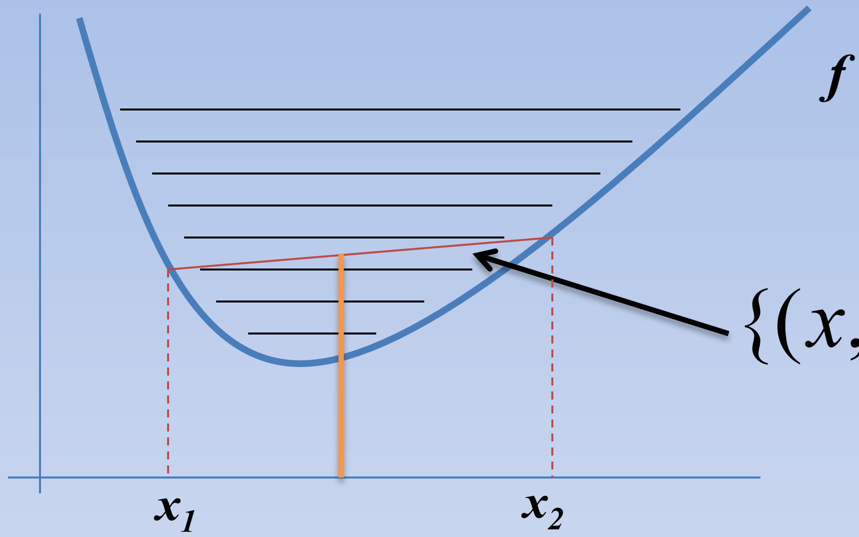
Convex Sets



C
$\mathbf{x}_2$
$\mathbf{x}_1$
$\lambda\mathbf{x}_1+(1-\lambda)\mathbf{x}_2, 0\le\lambda\le 1$

A set **C** is convex if for any x_1, x_2 in **C**, $\lambda x_1 +(1-\lambda)\, x_2$ is also in **C**.

Convex Functions



$$\{(x,y) \mid y \geq f(x)\}$$

A function ***f*** is convex if its epigraph is a convex set.

$$f(\lambda x_1 + (1-\lambda)x_2) \leq \lambda f(x_1) + (1-\lambda) f(x_2)$$

Jensen's inequality

For a convex function, every local optimum is also a global optimum.

Constrained Optimization

$$\begin{aligned} &\min_{\mathbf{x}} f(\mathbf{x}) \\ &s.t. \;\; g_i(\mathbf{x}) \geq 0, i = 1, \ldots, m \\ &\qquad\;\; h_j(\mathbf{x}) = 0, j = 1, \ldots, k \end{aligned}$$

The constraints define a "**feasible region**" where the solution must lie.

Linear Programming

- A special case of constrained optimization where the objective and the constraints are all linear functions

$$\min_{\mathbf{x}} \sum_i c_i x_i$$

$$s.t. \quad \sum_i a_{ri} x_i \geq 0, r = 1, \ldots, m$$

$$\sum_i b_{si} x_i = 0, s = 1, \ldots, k$$

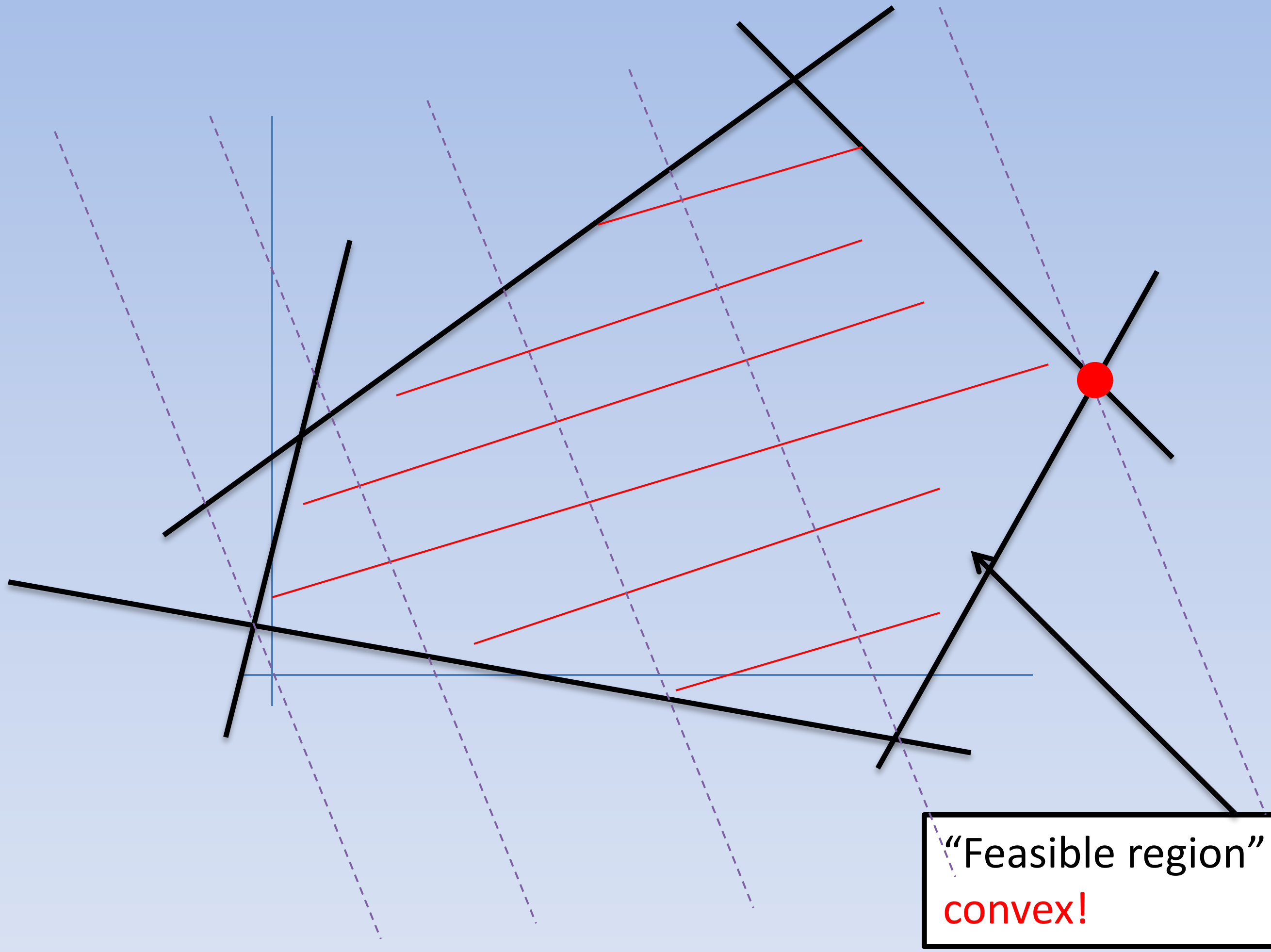
$$\min_{\mathbf{x}} \mathbf{c}^T \mathbf{x}$$

$$s.t. \quad A\mathbf{x} \geq 0,$$

$$B\mathbf{x} = 0$$

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ & & \end{pmatrix}, B = \begin{pmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{pmatrix}$$

Geometry



“Feasible region” (polyhedron) convex!

Simplex Algorithm

- Simple idea: around the polyhedron we go
- From any feasible vertex, walk along the edges of the polyhedron, following the vertices
- Once you are at a vertex where the neighboring vertices have higher f values, stop
- This is a local optimum
  - But this is a convex problem, so this is also a global optimum ☺

Properties of the Simplex Algorithm

- Very simple, easy to implement and works well in practice
- However, since it works by traversing vertices, and there might be exponentially many vertices for n constraints, the worst case runtime complexity is exponential
  - Average case under various distributions has been shown to be polynomial
- Other algorithms exist, such as “interior point methods”, which have polynomial bounds*

Duality in Linear Programming

- From any “primal” LP, we can derive a “dual” LP in the following way:

$$\begin{aligned} \min_{\mathbf{x}} \; & c^T \mathbf{x} \\ s.t. \; & A\mathbf{x} \geq b \\ & \mathbf{x} \geq 0 \end{aligned}$$

“Primal” problem

$$\begin{aligned} \max_{\mathbf{u}} \; & b^T \mathbf{u} \\ s.t. \; & A^T \mathbf{u} \leq c \\ & \mathbf{u} \geq 0 \end{aligned}$$

“Dual” problem

Primal and Dual LPs

- The primal has a solution iff the dual has a solution

- Further, the dual LP is a *lower bound* on the primal LP
  - That is, if we pick any feasible $\boldsymbol{x}$ and any feasible $\boldsymbol{u}$, we always have $\boldsymbol{c}^T\boldsymbol{x} \geq \boldsymbol{b}^T\boldsymbol{u}$ (prove)

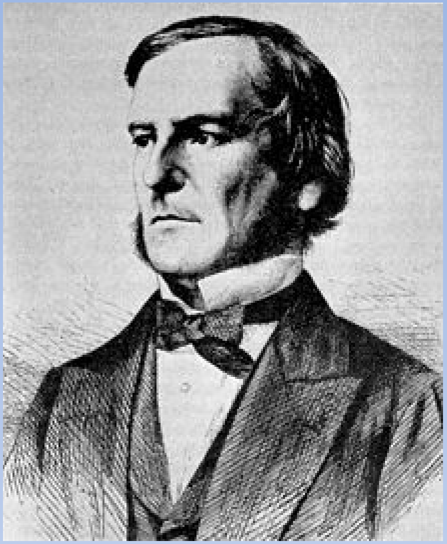
- From the relationship between primal and dual LPs, we can derive a set of conditions that characterize the optimal solution of a primal/dual pair of LPs

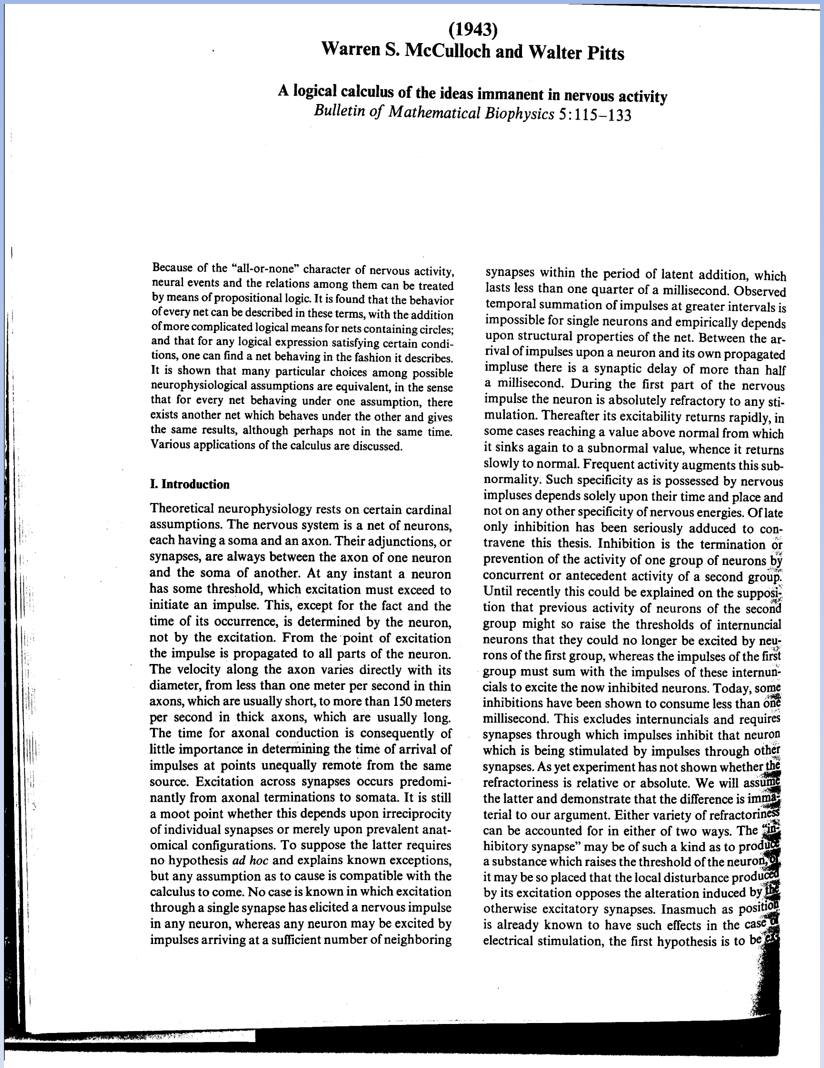
Karush-Kuhn-Tucker Conditions

- A set of conditions that are necessary and sufficient for optimal solutions of a primal/dual pair of linear (or more generally convex) programs

- Essentially, at the optimal solution, $\boldsymbol{x}$ and $\boldsymbol{u}$ are feasible and the objective functions $\boldsymbol{c}^T\boldsymbol{x}$ and $\boldsymbol{b}^T\boldsymbol{u}$ are equal
  - And some other stuff (later)

Artificial Neural Networks





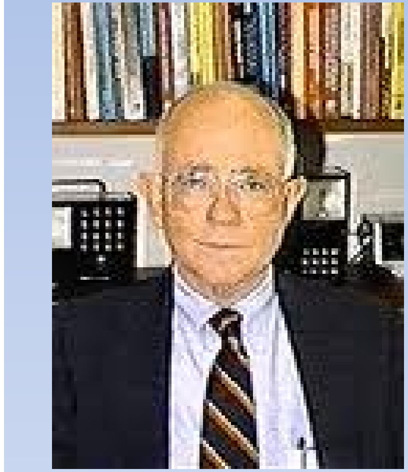
(1943)
Warren S. McCulloch and Walter Pitts

A logical calculus of the ideas immanent in nervous activity
Bulletin of Mathematical Biophysics 5:115–133

Because of the "all-or-none" character of nervous activity, neural events and the relations among them can be treated by means of propositional logic. It is found that the behavior of every net can be described in these terms, with the addition of more complicated logical means for nets containing circles; and that for any logical expression satisfying certain conditions, one can find a net behaving in the fashion it describes. It is shown that many particular choices among possible neurophysiological assumptions are equivalent, in the sense that for every net behaving under one assumption, there exists another net which behaves under the other and gives the same results, although perhaps not in the same time. Various applications of the calculus are discussed.

I. Introduction

Theoretical neurophysiology rests on certain cardinal assumptions. The nervous system is a net of neurons, each having a soma and an axon. Their adjunctions, or synapses, are always between the axon of one neuron and the soma of another. At any instant a neuron has some threshold, which excitation must exceed to initiate an impulse. This, except for the fact and the time of its occurrence, is determined by the neuron, not by the excitation. From the point of excitation the impulse is propagated to all parts of the neuron. The velocity along the axon varies directly with its diameter, from less than one meter per second in thin axons, which are usually short, to more than 150 meters per second in thick axons, which are usually long. The time for axonal conduction is consequently of little importance in determining the time of arrival of impulses at points unequally remote from the same source. Excitation across synapses occurs predominantly from axonal terminations to somata. It is still a moot point whether this depends upon irreciprocity of individual synapses or merely upon prevalent anatomical configurations. To suppose the latter requires no hypothesis *ad hoc* and explains known exceptions, but any assumption as to cause is compatible with the calculus to come. No case is known in which excitation through a single synapse has elicited a nervous impulse in any neuron, whereas any neuron may be excited by impulses arriving at a sufficient number of neighboring synapses within the period of latent addition, which lasts less than one quarter of a millisecond. Observed temporal summation of impulses at greater intervals is impossible for single neurons and empirically depends upon structural properties of the net. Between the arrival of impulses upon a neuron and its own propagated impluse there is a synaptic delay of more than half a millisecond. During the first part of the nervous impulse the neuron is absolutely refractory to any stimulation. Thereafter its excitability returns rapidly, in some cases reaching a value above normal from which it sinks again to a subnormal value, whence it returns slowly to normal. Frequent activity augments this subnormality. Such specificity as is possessed by nervous impulses depends solely upon their time and place and not on any other specificity of nervous energies. Of late only inhibition has been seriously adduced to contravene this thesis. Inhibition is the termination or prevention of the activity of one group of neurons by concurrent or antecedent activity of a second group. Until recently this could be explained on the supposition that previous activity of neurons of the second group might so raise the thresholds of internuncial neurons that they could no longer be excited by neurons of the first group, whereas the impulses of the first group must sum with the impulses of these internuncials to excite the now inhibited neurons. Today, some inhibitions have been shown to consume less than one millisecond. This excludes internuncials and requires synapses through which impulses inhibit that neuron which is being stimulated by impulses through other synapses. As yet experiment has not shown whether the refractoriness is relative or absolute. We will assume the latter and demonstrate that the difference is immaterial to our argument. Either variety of refractoriness can be accounted for in either of two ways. The "inhibitory synapse" may be of such a kind as to produce a substance which raises the threshold of the neuron, or it may be so placed that the local disturbance produced by its excitation opposes the alteration induced by the otherwise excitatory synapses. Inasmuch as position is already known to have such effects in the case of electrical stimulation, the first hypothesis is to be ex



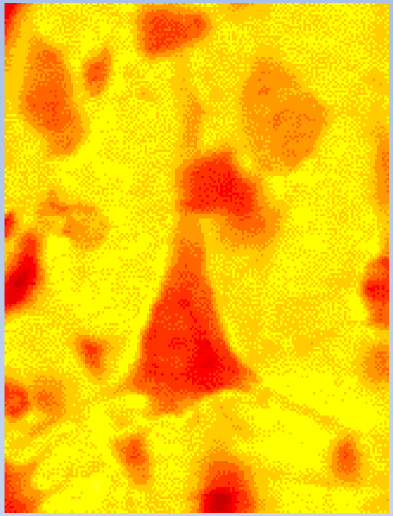
History

- We want “artificial intelligence”
  - Well, the brain possesses intelligence (sometimes)

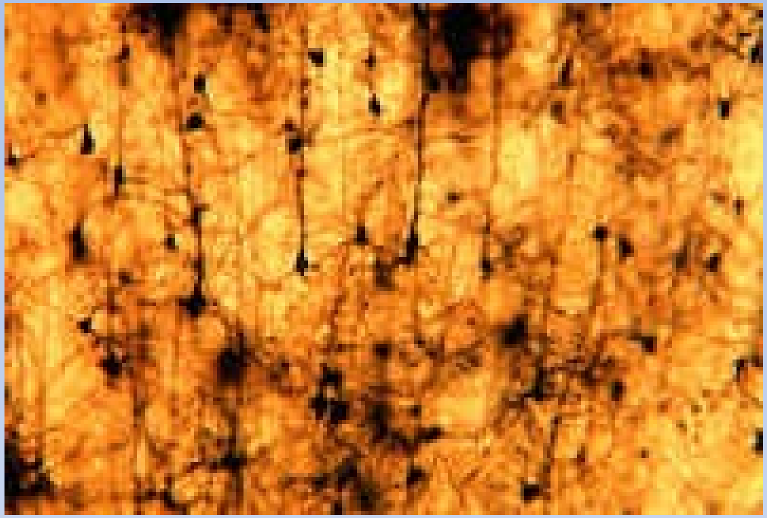
- Let’s try to simulate the *structure* of the brain and hope the *function* will follow

- Create basic simulation of neuron, connect them up in large numbers, and stand back
  - Maybe it will sing “Daisy, Daisy”
  - Thus the school of “Connectionism” was born
  - http://en.wikipedia.org/wiki/Connectionism

Neurons



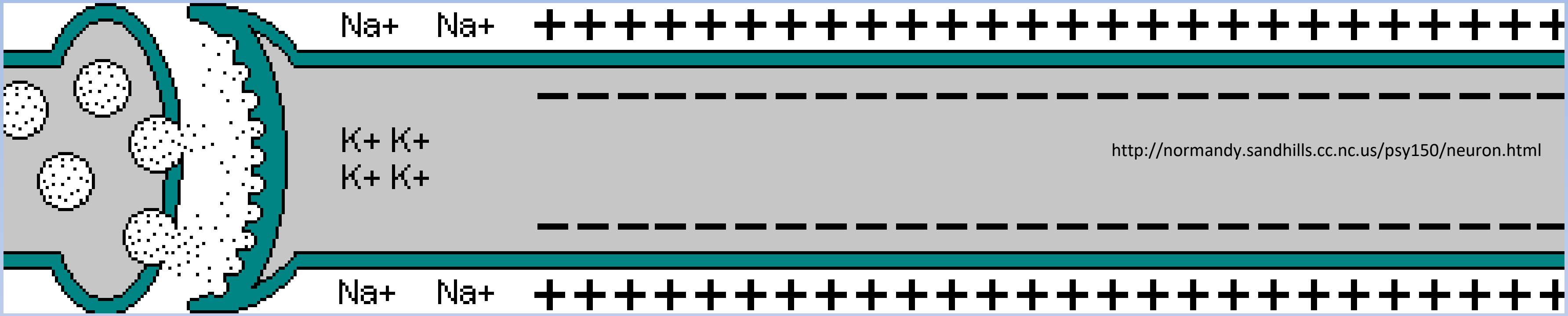
Cell body located in the deeper layers of the cerebral cortex. This is called a pyramidal neuron based on its shape.
From
http://faculty.washington.edu/chudler/cellpyr.
html



Neurons located in the cerebral cortex of the hamster.
From
http://faculty.washington.edu/chudler/cellpyr.h
tml

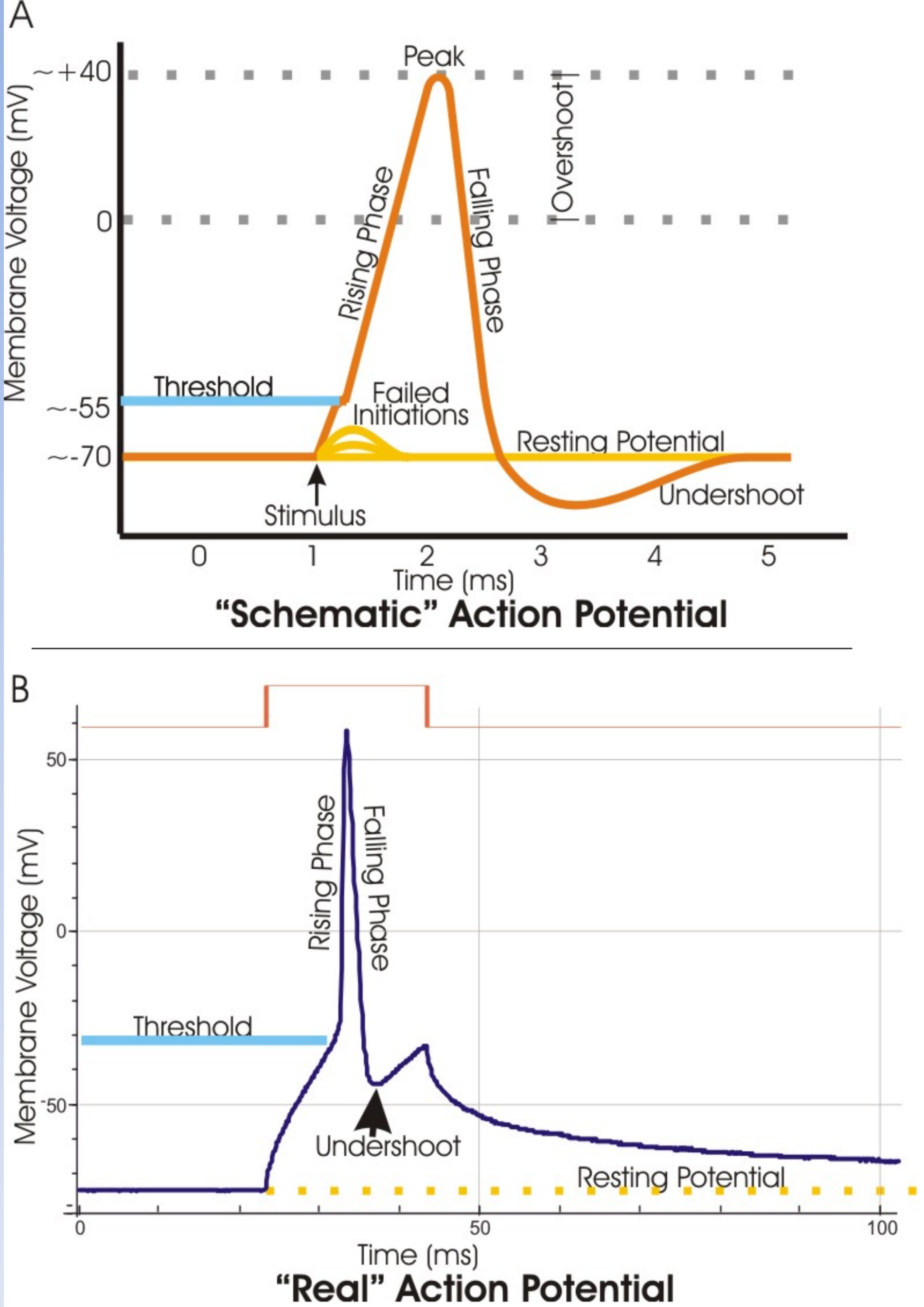
See http://en.wikipedia.org/wiki/Neuron for more details.

Basic Neuronal Cell Biology



http://normandy.sandhills.cc.nc.us/psy150/neuron.html

- Resting state of neuron maintains -70mV because of ion imbalance
- Signals from neighboring neurons reach end of axons
- Vesicles containing neurotransmitter released into synapse and attach to receptors on neighboring neuron
- When enough vesicles attach, molecular “gates” open on the membrane and allow positive ions in, rapidly depolarizing the neuron
- Eventually, these ions are transported back outside, returning the cell to its rest potential

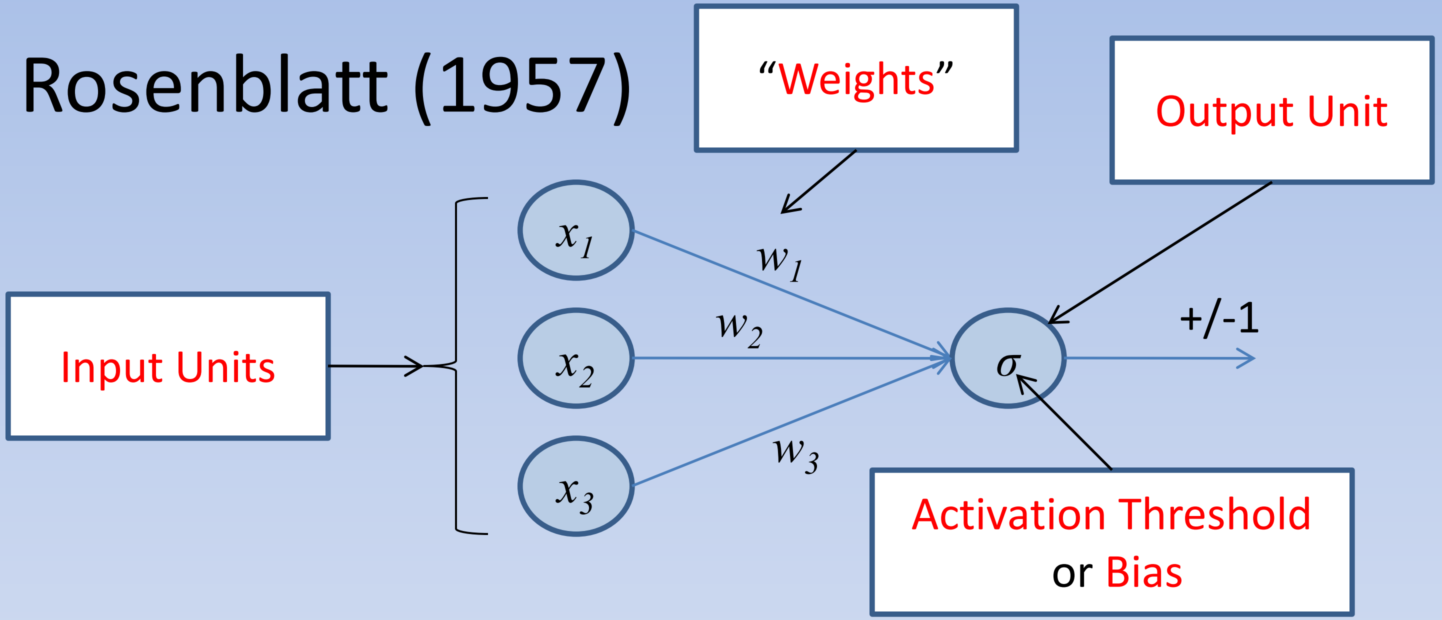


“Integrate-then-fire”

http://en.wikipedia.org/wiki/File:Action_potential_vert.png

Perceptron/Linear Threshold Unit

- Rosenblatt (1957)



“Weights”

Output Unit

Input Units

Activation Threshold or Bias

$$h(\mathbf{x};\mathbf{w},\sigma)=\begin{cases}+1 \text{ if } \mathbf{w}\bullet\mathbf{x}\ge\sigma \\ -1 \text{ else}\end{cases} = sign(\mathbf{w}\bullet\mathbf{x}-\sigma)$$

Activation Function

sign(x)=+1 if x > 0, −1 else

Parameters of the Perceptron

$$h(\mathbf{x};\mathbf{w},\sigma)=\begin{cases}+1 \text{ if } \mathbf{w}\bullet\mathbf{x}\ge\sigma\\ -1 \text{ else}\end{cases} = sign(\mathbf{w}\bullet\mathbf{x}-\sigma)$$

Parameters of the perceptron to be learned from data

Evaluation Phase

- Given (w, σ), classify an example $\mathbf{x}$
- $w=(1,2)\ \sigma=0.5$

x_1	x_2	h
0	0	-1
0	1	1