Neurons

How do they do it?

Detector Model

Each neuron detects some set of conditions (e.g., smoke detector).

Neurons feed on each other’s outputs — layers of ever more complicated detectors.

(Things can get very complex in terms of content, but each neuron is still carrying out basic detector function).

Understanding Neural Components in Detector Model

It’s Pandemonium!
Pandemonium Example

Each neuron has a simple job, but together...

Layers of more and more complicated detectors.

Simple example, but raises question of what kind of detectors needed for language, face recognition, creativity, etc.?

How do we simulate this?

- Neural activity (and learning) can be characterized by mathematical equations.
- We use these equations to specify the behavior of artificial neurons.
- The artificial neurons can then be put together to explore behaviors of networks of neurons.
- Simulation.

A Real Neuron

- It's a cell: body, membrane, nucleus, DNA, RNA, proteins, etc.
- Membrane has channels, passing ions (salt water).
- Cell has electrical potential (voltage), integrated in cell body, activates action potential output in axon, releases neurotransmitter.
- Neurotransmitter activates potential via dendritic synaptic input channels.
- Excitation and inhibition are transmitted by different neurons!
**The Synapse**

- **Dendrite**
- **Axon**
- **Microtubule**
- **Cleft**
- **Vesicles**
- **Neurotransmitter (glutamate)**
- **Receptors**
  - mGlu
  - AMPA
  - NMDA
- **Terminal Button**
- **Postsynaptic**
- **Presynaptic**

**Weight = Synaptic Efficacy**

**Synaptic efficacy** = activity of **presynaptic** (sending) neuron communicated to **postsynaptic** (receiving) neuron:

- Presynaptic: # of vesicles released, NT per vesicle, efficacy of reuptake mechanism.
- Postsynaptic: # of receptors, alignment & proximity of release site & receptors, efficacy of channels, geometry of dendrite/spine.

**Drugs:** Prozac (serotonin reuptake), L-Dopa (NT in vesicles)

---

**Abstract Neural Nets**

1. Compute weighted, summed net input:
   \[ \eta_j = \sum_i x_i w_{ij} \]

3. Pass this through a **sigmoidal** function to get output:
   \[ y_j = \frac{1}{1 + e^{-\eta_j}} \]

---

**Bio Neural Nets**

1. Compute weighted, summed net input (actually averages):
   \[ \eta_j \approx \sum_i a_i w_{ij} \approx g_e \] (1)

2. Compute \( V_m \):
   \[ V_m = \frac{g_e \gamma E_e + g_l \gamma_l E_l + g_i \gamma_i E_i}{g_e g_i + g_l \gamma_i + g_l \gamma_l} \] (2)

3. Compute output as: Spikes Or, rate code via **sigmoidal** function:
   \[ a_j = \frac{1}{1 + (\gamma [V_m(t) - \Theta]^+)^{-1}} \] (3)
Neurophysiology

The neuron is a miniature electro-chemical system:

1. Balance of electric and diffusion forces.
2. Principal ions.
3. Putting it all together.

Balance of Electric and Diffusion Forces

Ions flow into and out of the neuron under the forces of electricity and concentration gradients (diffusion).

The net result is a electric potential difference between the inside and outside of the cell — the membrane potential $V_m$.

This value represents an integration of the different forces, and an integration of the inputs impinging on the neuron.

Electricity

Positive and negative charge (opposites attract, like repels).

Ions have net charge: Sodium ($Na^+$), Chloride ($Cl^-$), Potassium ($K^+$), and Calcium ($Ca^{++}$) (brain = mini ocean).

Current flows to even out distribution of positive and negative ions.

Disparity in charges produces potential (the potential to generate current..)

Resistance

Ions encounter resistance when they move.

Neurons have channels that limit flow of ions in/out of cell.

The smaller the channel, the higher the resistance, the greater the potential needed to generate given amount of current (Ohm’s law):

$$I = \frac{V}{R} \quad (4)$$

Conductance $G = 1/R$, so $I = VG$
Diffusion

Constant motion causes mixing – evens out distribution.

Unlike electricity, diffusion acts on each ion separately — can’t compensate one + ion for another.

(same deal with potentials, conductance, etc)

\[ I = -DC \]  

(Fick’s First law)

Equilibrium

Balance between electricity and diffusion:

\[ E = \text{Equilibrium potential} = \text{amount of electrical potential needed to counteract diffusion} \]

\[ I = G(V - E) \]  

Also:

- **Reversal** potential (because current reverses on either side of \( E \))
- **Driving** potential (flow of ions drives potential toward this value)

The Neuron and its Ions

Everything follows from the sodium pump, which creates the "dynamic tension" (compressing the spring, winding the clock) for subsequent neural action.

Glutamate \( \rightarrow \) opens Na+ channels \( \rightarrow \) Na+ enters (excitatory)

GABA \( \rightarrow \) opens Cl- channels \( \rightarrow \) Cl- enters if \( V_m \) ↑ (inhibitory)
Drugs and Ions

- Alcohol: closes Na
- General anesthesia: opens K
- Scorpion: opens Na and closes K

Putting it Together

\[ I_c = g_c(V_m - E_c) \] (7)
\[ e = \text{excitation (Na}^+) \]
\[ i = \text{inhibition (Cl}^-) \]
\[ l = \text{leak (K}^+) \]

\[ I_{net} = g_e(V_m - E_e) + g_i(V_m - E_i) + g_l(V_m - E_l) \] (8)

\[ V_m(t + 1) = V_m(t) \pm dt v_m I_{net} \] (9)

or

\[ V_m(t + 1) = V_m(t) \pm dt v_m I_{net} \] (10)

Putting it Together: With Time

\[ I_c = g_c(t) g_e(V_m(t) - E_e) \] (11)
\[ e = \text{excitation (Na}^+) \]
\[ i = \text{inhibition (Cl}^-) \]
\[ l = \text{leak (K}^+) \]

\[ I_{net} = g_e(t) g_e(V_m(t) - E_e) + g_i(t) g_i(V_m(t) - E_i) + g_l(t) g_l(V_m(t) - E_l) \] (12)

\[ V_m(t + 1) = V_m(t) \pm dt v_m I_{net} \] (13)

or

\[ V_m(t + 1) = V_m(t) \pm dt v_m I_{net} \] (14)

It’s Just a Leaky Bucket

\[ g_e = \text{rate of flow into bucket} \]
\[ g_{i/l} = \text{rate of “leak” out of bucket} \]
\[ V_m = \text{balance between these forces} \]
**Or a Tug-of-War**

**Excitation**

- $g_e$ causes $E_e$ to increase $V_m$

**Inhibition**

- $g_i$ causes $E_i$ to decrease $V_m$

**In Action**

(Two excitatory inputs at time 10, of conductances .4 and .2)

---

**Overall Equilibrium Potential**

If you run $V_m$ update equations with steady inputs, neuron settles to new equilibrium potential.

To find, set $I_{net} = 0$, solve for $V_m$:

$$V_m = \frac{g_e \tilde{g}_e E_e + g_i \tilde{g}_i E_i + g_i \tilde{g}_i E_i}{g_e \tilde{g}_e + g_i \tilde{g}_i + g_i \tilde{g}_i}$$  \hfill (15)

Can now solve for the equilibrium potential as a function of inputs.

Simplify: ignore leak for moment, set $E_e = 1$ and $E_i = 0$:

$$V_m = \frac{g_e \tilde{g}_e}{g_e \tilde{g}_e + g_i \tilde{g}_i}$$  \hfill (16)

Membrane potential computes a balance (weighted average) of excitatory and inhibitory inputs.
Computational Neurons (Units) Summary

1. Weights = synaptic efficacy; weighted input = $x_i w_{ij}$.
Net conductances (average across all inputs) excitatory ($\text{net} = g_e(t)$), inhibitory $g_i(t)$.

2. Integrate conductances using $V_m$ update equation.

3. Compute output $y_j$ as spikes or rate code.

Thresholded Spike Outputs

Voltage gated $Na^+$ channels open if $V_m > \Theta$, sharp rise in $V_m$.

Voltage Gated $K^+$ channels open to reset spike.

In model: $y_j = 1$ if $V_m > \Theta$, then reset (also keep track of rate).

Rate Coded Output

Output is average firing rate value.
One unit = % spikes in population of neurons?

Rate approximated by X-over-X-plus-1 ($\frac{x}{x+1}$):

$$y_j = \frac{\gamma [V_m(t) - \Theta]_+}{\gamma [V_m(t) - \Theta]_+ + 1}$$
(17)

which is like a sigmoidal function:

$$y_j = \frac{1}{1 + (\gamma [V_m(t) - \Theta]_+)^{-1}}$$
(18)

compare to sigmoid: $y_j = \frac{1}{1 + e^{-\gamma y}}$

$\gamma$ is the gain: makes things sharper or duller.

Convolution with Noise

X-over-X-plus-1 has a very sharp threshold
Smooth by convolve with noise (just like “blurring” or “smoothing” in an image manip program):
Computing Excitatory Input Conductances

One projection per group (layer) of sending units.

Average weighted inputs \( \langle x_i w_{ij} \rangle = \frac{1}{n} \sum_i x_i w_{ij} \).

Bias weight \( \beta \): constant input.

Factor out expected activation level \( \alpha \).

Other scaling factors \( a, s \) (assume set to 1).

Computing \( V_m \)

Use \( V_m(t+1) = V_m(t) + dV_m I_{net} \) with biological or normalized (0-1) parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>( mV )</th>
<th>(0-1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>( V_{rest} )</td>
<td>-70</td>
<td>0.15</td>
</tr>
<tr>
<td>( E_l (K^+) )</td>
<td>-70</td>
<td>0.15</td>
</tr>
<tr>
<td>( E_i (Cl^-) )</td>
<td>-70</td>
<td>0.15</td>
</tr>
<tr>
<td>( \Theta )</td>
<td>-55</td>
<td>0.25</td>
</tr>
<tr>
<td>( E_e (Na^+) )</td>
<td>+55</td>
<td>1.00</td>
</tr>
</tbody>
</table>

Normalized used by default.
## Detector vs. Computer

<table>
<thead>
<tr>
<th></th>
<th>Computer</th>
<th>Detector</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Memory &amp; Processing</strong></td>
<td>Separate, general-purpose</td>
<td>Integrated, specialized</td>
</tr>
<tr>
<td><strong>Operations</strong></td>
<td>Logic, arithmetic</td>
<td>Detection (weighing &amp; accumulating evidence, evaluating, communicating)</td>
</tr>
<tr>
<td><strong>Complex Processing</strong></td>
<td>Arbitrary sequences of operations chained together in a program</td>
<td>Highly tuned sequences of detectors stacked upon each other in layers</td>
</tr>
</tbody>
</table>