1 Introduction

We'll learn classical ideas from neural networks that have almost been forgotten. "The mark of an old fart is that (s)he teaches the same thing from when (s)he became a professor." Even then neural nets had gone out of style. The backpropagation upswing was revived from the 1980's; people didn't know them anymore. What classic ideas will be the next big thing? "I'll make my guesses and teach them to you."

This is the third age of neural networks; there's a long history of forgotten lessons.

This class is about these equations, **classical neurodynamics**.

$$\tau \frac{dx_i}{dt} + x_i = f\left(\sum_j W_{ij} x + b_i\right). \tag{1}$$

eq:classical-neuro

Here

1. x_i are N variables representing activity.
2. W_{ij} are synaptic strength.
3. b_i is bias.

Parameters are W_{ij} and b_i.

These are used in

1. brain modeling,
2. by computer scientists for artificial intelligence (computer vision, etc.), and in
3. dynamical systems theory (it can inform the use of these equations).

f makes the system nonlinear.

Depending on W_{ij}, we can make these equations do all kinds of interesting things. In particular, they can compute all boolean functions.

Today is the only day we'll worry about what these equations mean; later we'll take them for granted.

1.1 Biological interpretation

"I'll give a hand-wavy explanation."

A neuron has a cell body, several dendrites, and one axon. Dendrites are thicker than axons. Note the generic term "neuron" is an oversimplification because there are many types.

The law of dynamic polarization (1996 Nobel prize): the dendrites are the input and the axon is the output. The neurons communicate with each other via synapses (intersections between axon and dendrite). At a synapse the axon sends the message and the dendrite receive the message.

The dendrites are more linear in their behavior than the axon. As a first approximation it linearly sums its inputs.

The axon is nonlinear; it's all-or-none. It give signals in the form of 1 millisecond binary pulses. Why is it so nonlinear? The obvious reason is that it's very long—they can go from one part of the brain to another.

If you have a long conduit, there are many problems: attenuation, noise. (Cf. the first transatlantic cable.) One solution is to be digital. Active cholesteral properties retain the pulse. For dendrites the signals attenuate.

The all-or-none character of an axon makes it possible for it to make a decision. "All computation depends on nonlinearity; a linear system is limited in what it can compute."

Returning to (1),

1. The linear sum $Wx + b$ approximates what the synapses do.

2. f approximates what the axon does.

Neuroscience experiment: take a microelectrode, stick it into the neuron, fill it with conductive saline solution, measure the voltage with an amplifer. It's faking the dendrite, trying to measure f directly.

If we inject a little current, voltage only goes up a little bit. If we inject a lot of current, we get a lot of spikes. There is a threshold above which we get spikes; increasing it beyond that gives more frequent spikes.

We can make a graph of current (x) vs. frequency (y): the f-I curve.

The sum $\sum_j W_{ij}x_j + b_i$ is the total current. How do we relate the current x_i to the frequency $f\left(\sum_j W_{ij}x_j + b_i\right)$? What exactly is x: is it a current or rate? It's some kind of activity. One way neuroscientists quantify neuronal activity is by frequency: the neuron is active at 5 hertz, 10 hertz, etc. τ sets the elementary time scale. Mathematically it's simple; physically it's less clear.

A neuron takes a linear combination with large fan-in, passing the result through a nonlinear scalar function; this is basically the simplest dynamical system that does that.

Something else might bother you: the x_j vary continuously in time, but neurons makes spikes—it seems almost discontinuous. In the equation there's no explicit representation of the time of a spike. You won't see sudden brief pulses. The justification of these equations depends on our being able to neglect the spiking of neurons in favor of variables more like rates. This is "classical neurodynamics." Neural activities is quantized; spikes are packets; we'll neglect spikes in favor of rates. Cf. how we neglect the particle nature of light when the rate of photon arrival is high. We'll derive conditions under which we can neglect the spiking.

These are our assumptions.

1. Each synapse is a current source controlled by presynaptic spiking.

2. The dendrite adds the currents of multiple synapses.

3. The total current drives spiking in the axon.

Voltage clamp measurement of synaptic transmission: measure the current in post-synaptic neuron.

Model the blip with a decaying exponential.

$$I(t) = Wg(t)$$
$$g(t) = \begin{cases} \tau^{-1} e^{-t/\tau}, & t \geq 0 \\ 0, & t < 0. \end{cases}$$

Assume:

1. Temporal summation: Assume that currents from successive spikes add linearly.
2. Currents of divergent synapses share the same time course. They share the time constant τ and only differ in amplitude. (I.e. different multiples of the current get sent to the different neurons.)

The normalized current is

$$I_{ij}(t) = W_{ij} \sum_a g(t - t_j^a) = W_{ij} x_j(t)$$
$$x_j(t) = \sum_a g(t - t_j^a).$$

When τ gets large, each exponential decays before the next one comes in. If $\tau = 100$ms, then we can approximate it with a smooth function; if $\tau = 5$ms, the smooth function is a bad approximation.

We can express this as the leaky integrator model: if there is a spike $x_j := x_j + \frac{1}{\tau}$. Otherwise, exponential decay with time $\tau \frac{dx_j}{dt} + x_j = \sum_a \delta(t - t_j^a)$. If τ is long, we can replace it by a rate v_j.

Summation of currents from convergent synapses: liner superposition $I_i = \sum I_{ij}$.

Putting it all together we get (1).

The biophysical interpretation is

1. f is firing rate as function of current.
2. x is normalized synaptic current (in Hz).
3. τ is synaptic time constant. How long does it take for the current to decay?
4. W is the charge/presynaptic spike. (Amps/hertz, which is coulombs (charge).) The W_{ij} is the total charge in neuron i injectedby spike in neuron j; the area underneath the synaptic current. $f(\cdots)$ is the low-pass version.

Typical rates of firing: τ can be 5–100ms, even 1s.

There are as many synaptic currents as synapses, in principle n^2. But they are in groups of n (those coming out of the same neuron) that behave the same temporally. What is b? In the absence of synaptic current, we get $f(b_i)$ current. For $b_i < 0$, this represents activation necessary.

What do people use for f? Three popular ones:

1. $[u]^+ = \max\{u, 0\}$: rectification ($(x \geq 0)|x|$). (This is biological—many neurons behave as rectification.)

2. $H(u)$ step function

3. $\sigma(u)$ sigmoidal function.

f-I curve from experiments with squid giant axon (not giant squid axon, from a normal squid, a mm thick); it's popular for neuroscience experiments because it's so big.

1. EEngineers like rectification because it's halfway a linear system.

2. There are neurons that look more binary, like $H(u)$. It is beloved by theoretical computer scientists because they can reduce to boolean functions.

3. The advantage of σ is having smooth derivative; it's a smooth version of $H(u)$. It is a compromise: binary for large inputs and linear for small inputs. Used are $\sigma = \frac{1}{1-e^{-u}}$ and $\tanh u = \frac{e^u - e^{-u}}{e^u + e^{-u}}$.

Some things train faster with tanh.

Deficiencies: There are many ways real neurons vary even from the spiking model.

1. Spike frequency adaptation: it responds to transient faster spiking.

2. Short-term depression: stimulating it many times in quick succession, later pulses are weaker, as if it gets tired.

3. They are not perfect current sources. There are conducting changes, giving a nonlinear interaction between multiple synapses.

This simple model is very rich.

The importance of the deviations are still hotly debated. The model is wrong, but how badly is it wrong?

Question: these equations look so simple, are they a good model for the brain? The complexity is in the synaptic interactions (connectionism, we can do anything with good connections).

This is the Matlab model of the brain:

$$\tau \frac{d\mathbf{x}}{dt} + \mathbf{x} = \mathbf{f}(\mathbf{b} + W\mathbf{x}), \qquad \mathbf{f}(\mathbf{u}) = \begin{pmatrix} f(u_1) \\ \vdots \\ f(u_N) \end{pmatrix}.$$

"The brain is a vast parallel computer that computes matrix functions."

Most entries of the matrix are 0. (Otherwise your brain would be too large.) Estimate: the number of neurons is in $[10^{11}, 10^{12}]$; the number of synapses is 10^{15}. 1 synapse every micron. 1 cubic millimeter has 1 billion synapses. It's 3-D large-scale integration.

Sparse connectivity is efficient: synapses occupy volume, consume power. The number of synapses is a crude measure of consumption of biological resources.

The time complexity of simulating is the time complexity of matrix multiplication. Even if we simulate it on a digital computer, the sparsity determines how quickly we can run it.

Connectivity is mostly local: between neighboring neurons the probability of connections is higher (most are within 1-2mm). Presumably the layout of the brain has been optimized by evolution to minimize "wiring length." (Integrated circuit designers face a similar problem.)

Why is your brain inside your head? The brain is in your head to minimize the length of wiring to sensors. The number of wires to your spine is the about the same as to sensors.

Dale's Law: a single neuron makes either excitatory or inhibitory synapses but not both; excitatory neurons generally outnumber inhibitory neurons. Every column of the matrix $(W_{i\bullet})$ is either nonnegative or nonpositive. Computer scientists tend to ignore this constraint.

Unanswered: Is it a weird biological constraint that has no function, or actually useful?

2 Neural nets

For the next few weeks, we'll consider neural nets with no loops (feedforward networks)—directed acyclic graphs. (I.e. there is a permutation of the nodes so that the matrix is strictly lower triangular.) Here W_{ij} represents the weight of the connection from j to i; the only connections are to larger nodes. (There are cases in biology where neurons make connections onto themselves, called autapses.)

The steady-state equation, when $\frac{d\mathbf{x}}{dt} = 0$.

$$\mathbf{x} = f(W\mathbf{x} + \mathbf{b}).$$

In general this is hard to solve but for a feedforward network, we can simply evaluate the x_i in increasing order:

$$\begin{aligned} x_1 &= f(b_1) \\ x_2 &= f(W_{21}x_1 + b_2) \\ &\dots \end{aligned}$$

(cf. how it's easy to solve a triangular system of equations.)

There are feedforward, lateral, and feedback connections.

We start by understanding a single layer of neurons, and start by talking about what a single model neuron does.

A neuron can have a fan-in of 100,000; why so much?

2.1 Single neuron

Consider the special case of binary output,

$$H(\sum_j w_j x\_\theta) = H(\mathbf{w} \cdot \mathbf{x} - \theta)$$

What boolean functions can be realized by a LT (linear threshold) neuron? This is the beginning of theoretical neuroscience., "A logical calculus of the ideas imminent in nervous

activity," by McCollough and Pitts, 1940's. Neurons are doing logical operations. Boole called logic the laws of thought.

Suppose $w_i = 1$ for all i; the expression is $H(\sum_j x_j - \theta)$.

- $\wedge$: for n variables, let $\theta = n - \frac{1}{2}$.
- $\vee$: let $\theta = \frac{1}{2}$.

Adding in inhibition, we have $H(\sum_{j=1}^{n} x_j - \sum_{j=n+1}^{N} x_j - \theta)$. Think of inhibition as negation, so

$$x_1 \wedge \overline{x_2} \wedge x_3 = H(x_1 - x_2 + x_3 - 1.5).$$

Any conjunction or disjunction of N variables or negations can be realized by an LT neuron.

Weighted voting model. For $\mathbf{x} \in \mathbb{R}^N$, we need $\mathbf{x}$ to be on one side of a the **separating hyperplane**.

A function can be decided by a LT neuron iff the 0's and 1's are linearly separable, for example, XOR cannot be.

The perferred stimulus is the direction along which minimal amplitude is needed for activation.

2.2 Delta rule

How do you set W_{ij}? Use a learning rule to get the network to organize itself.

How can we learn to recognize a "2" with a LT neuron?

The **delta rule** (**perceptron learning rule**) is (ignoring bias)

$$\Delta \mathbf{w} = \underbrace{\eta}_{\text{learning parameter}} [\underbrace{y}_{\text{desired}} - \underbrace{H(\mathbf{w}^T \mathbf{x})}_{\textit{actual}}]\mathbf{x}.$$

You're driven by the different between desire and reality. This is mistake-driven learning. ($\mathfrak{b}$ would be like another component of the weight vector; it's just a special case.)

There are 2 types of mistakes.

1. False positive: $y = 0, H(\mathbf{w}^T \mathbf{x}) = 1$. Then $\Delta w = -\eta \mathbf{x}$.
2. False negative: $y = 1, H(\mathbf{w}^T \mathbf{x}) = 0$. Then $\Delta w = \eta \mathbf{x}$.

We can update $\mathbf{w}$ after each example, or update after the whole batch of examples, or mini-batch. The intuition is that averaging reduces the noise in the update.

It's common to initialize with small random values.

A more sophisticated interpretation: this is an optimization procedure. It is a gradient-based optimization algorithm.

The cost function is

$$e(\mathbf{w}, \mathbf{x}, y) = |y - H(\mathbf{w}^T\mathbf{x})| \cdot |\mathbf{w}^T\mathbf{x}| = \begin{cases} \mathbf{w}^T\mathbf{x}, & \text{false positive} \\ -\mathbf{w}^T\mathbf{x}, & \text{false negative} \\ 0, & \text{correct.} \end{cases}$$

$$\frac{\partial e}{\partial \mathbf{w}} = \begin{cases} -\mathbf{x}, & \text{false positive} \\ \mathbf{x}, & \text{false negative} \\ 0, & \text{correct.} \end{cases}$$

$$\Delta w = -\eta \frac{\partial e}{\partial \mathbf{w}}.$$

Batch update is gradient descent on the sum of errors.