Scribes: Mahika Bhalla
Vasudha Rengarajan
COMS 6998-06: Computation and the Brain Jayant Madugula
Christos Papadimitriou 10/10/2018

Contents

1 Introduction and Recap

1.1 Talks by Kiran and Jacob

Kiran talked about mapping fMRI data to stimulus semantics. His team created a map from fMRI data from certain areas of the brain of subjects watching a movie, to a corpus of annotations on the movie. He also discussed word embeddings, which is a useful technique that encodes words in a corpus as vectors in R^d.

Jacob discussed computation in the fly's brain. There are lots of potential project topics in his talk!

1.2 STDP and biological plausibility of deep nets

We concluded discussing synaptic plasticity, which is how brains learn. The STDP graph shows that even fractions of a second can make a significant difference in synaptic weight gain. If spike arrives in time, some gain. Just in time, big gain. If it misses it, some loss. Just misses it, big loss.

This topic attracted the interest of people in deep learning, including Bengio et al., who published "Towards Biologically Plausible Deep Learning Through STDP" in 2016. They found that the equations for gradient descent (a deep learning concept) and STDP were very similar.

Gradient descent: $\Delta x^t = \alpha\delta(t)\nabla f(x^t)$
Update happens at time t.

STDP: $\Delta w^t = \beta\delta(t)\nabla V(w^t)$
t is the time the spike appears at the synapse.

Bengio et al. explored the idea of using an STDP feedforward net to optimize some objective function whose "local derivative" is V. They found that some learning can be done, but there are catches and different kinds of biological implausibility.

Papadimitriou's take on biological plausibility: People in deep nets drew inspiration from the brain, but it wasn't a completely faithful reproduction. There are many aspects of deep nets that have no neurological analogue. Deep nets can be made biologically plausible (in some well-defined sense) because forward computation is plausible (as in the visual cortex) and backprop can be thought of as modeling evolution.

1.3 Formula for entropy of a distribution D

$H(D) = -\Sigma_j \text{ Prob}[r_j] \log_2 (\text{Prob}[r_j])$

1.4 Shannon's Second Theorem

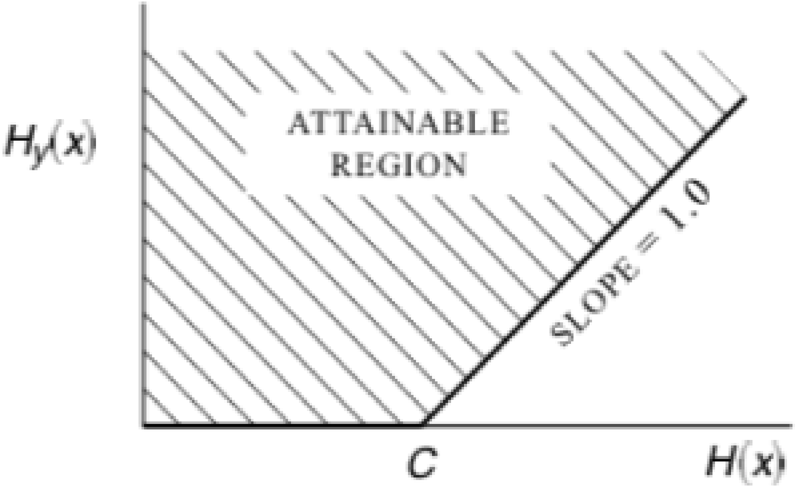
We can add random noise to a gradient (only 2% correlated with real gradient) and learn from it.

Shannon's second theorem:
If a channel has capacity C and noise $p < 1/2$, then:

(a) Any rate $R < C(1 - h(p))$ can be achieved by coding.

(b) No rate greater than $C(1 - h(p))$ can be achieved.

(c) If equivocation (uncertainty in decoding, $H(x|y) > 0$) is allowed, a certain attainable region is as shown (shaded part of the graph).



We ran through proofs of each of the subparts of this theorem in class.

Coding theory since Shannon: People have strived to achieve the Shannon bound, and 70 years later, we barely have done so. Polar codes, turbo-codes, and sparse graph codes are codes that just hug the Shannon line.

1.5 Joint, Relative, and Mutual Entropy

We discussed entropy of a joint distribution, conditional entropy, the chain rule (rather, Bayes' rule), mutual information, and "how far from independent" the random variables in these formulas are. We also discussed the Kullback-Leibler divergence of two distributions P and Q. Find all of these formulas in the slides.

McAllester (2018) connected this to deep learning: given the distribution P(image, label) in the world, you want to create another distribution $Q_{N(\Theta)}$(image, label) where Θ are the parameters (weight, etc.) of the CNN. To do so, you want to $\max_\Theta E_{(image,label)\sim P}$ log $Q_{N(\Theta)}$ (label|image), or, equivalently, $\min_\Theta KL(P, Q_{N(\Theta)})$.

Note that discrete information theory can be extended to continuous random variables and distributions: $\Sigma \to \int$.

One application lies in how flies encode contrast. To maximize entropy of the encoding, the response distribution should mimic the stimulus distribution.

Scribes: Mahika Bhalla
Vasudha Rengarajan
Jayant Madugula

COMS 6998-06: Computation and the Brain
Christos Papadimitriou
10/10/2018

1.6 Today's topics

- Continue on information theory and the brain
- Introduction to dynamical systems
- Examples of dynamical systems models of the brain

Professor Papadimitriou's thoughts on the talk that we watched for our weekly response: information theorists are critical of the results presented. The process of how neural nets are successful is a mystery.

2 Information Theory and the Brain

Another example of applying information theory to the brain is to ask: What is the information contained in the spike train of a neuron responding to a stimulus?

- Answer 1: Entropy rate: $H \lessapprox -R\int_0^\infty p(t)\ \log_2(p(t)\Delta t)dt$
  - $p(t)$ = probability that there is a spike in $[t, t+\Delta t]$
  - Δt small enough that two spikes are unlikely
  - Measure this over many experiments and estimate how much of this spiking rate has information.
  - For Poisson spikes: $H = R(1-\ \ln(R\Delta t))/\ln 2$
- Answer 2 (a better approach): $H = -[\Sigma_B p(B)\ \log_2 p(B)]/T$
  - Small Δt, and $T = m\Delta t$, where m = some integer, i.e. 6
  - $p(B)$ is the probability that B in $0, 1^m$ occurs in time interval T
  - Note that there is noise in spikes. See slides for equation to calculate noise entropy.

Rieke et al. (1995) found that frog auditory neurons respond with much higher H_{true} to sounds that resemble frog calls than to white noise – despite the fact that the latter has higher entropy.

The extent and scope of results linking information theory and the brain exist somewhat below Papadimitriou's expectations, despite the two being a "match made in heaven."

Note: Don't confuse information theory with Fisher information, often called just "information" in statistics. Fisher information is occasionally used instead of entropy in the study of brain systems.

Scribes: Mahika Bhalla
Vasudha Rengarajan
Jayant Madugula
10/10/2018

COMS 6998-06: Computation and the Brain
Christos Papadimitriou

3 Dynamical Systems

Dynamical Systems are also known as Ordinary Differential Equations, or ODEs.

Goal: Find $\dot{x} = f(x)$ given the value of x at $t = 0$.
$x(t)$ is an unknown function of time t (usually a vector function) and $\dot{x}$ can also be written as $\frac{dx}{dt}$.

A linear dynamical system is defined as $\dot{x} = Ax$, where A is a matrix.

(a) Solution: $x(t) = x(0) * e^{At}$ for one dimension

(b) This also works for any number of dimensions

(c) Linear systems are only useful as local approximations to nonlinear systems

Professor Papadimitriou recommended the book “Nonlinear Dynamics and Chaos” by Steven H. Strogatz (2nd Edition) to learn more.

3.1 Two-Body Problem

A motivation for dynamical systems came from the two-body problem solved by Newton in 1687.

The problem looks at the Earth and Moon, ignoring everything else. Specifically, how do the two opposing forces from the Earth and Moon’s gravity effect each other?

We can model these forces as follows:

$$F(x, y) = M\bar{x}$$
$$-F(x, y) = m\bar{y}$$

Newton’s solution showed the two-body problem can be easily solved, with the two bodies moving at constant velocity.

3.2 Three-Body Problem

The three-body problem (Earth-Sun-Moon) is “essentially unsolvable.”

We have found periodic solutions but we do not know the full set of solutions.
The original breakthrough came from Poincaré in the 1890s, who brought focus to qualitative questions: “Will the moon ever fly away?” This encouraged looking at limit behavior of systems, or how systems behave at their limits.

Scribes: Mahika Bhalla
Vasudha Rengarajan
Jayant Madugula

COMS 6998-06: Computation and the Brain
Christos Papadimitriou

10/10/2018

3.3 One-Dimensional Systems

1D systems only have (stable/unstable) equilibria where $f(x)$ intersects with the x-axis. Periodic solutions are not possible.

We can prove convergence using Lyapunov/potential functions.

1D System Examples:

(a) Exponential Growth: $\bar{x} = ax$

(b) Logistic Equation (growth with limits): $\bar{x} = ax(1 - x)$

(c) Discrete-Time Logistic Equation: $x^{t+1} = ax^t(1 - x^t)$

(d) Bifurcation: $\bar{x} = rx - x^3$

(e) Imperfect Bifurcation: $\bar{x} = h + rx - x^3$

3.4 Two-Dimensional Systems

2D systems allow for periodic solutions, such as the harmonic oscillator and pendulum examples.

These problems are defined as:

$$m\bar{x} = -kx$$

3.5 1D and 2D System Summary

Limit behavior of 1D Dynamical Systems is equilibrium. In 2D systems, we have both stationary (equilibrium) or periodic (cycles) limit behavior. See the Poincaré-Bendixson Theorem.

2D systems have no chaos.

3.6 Three-Dimensional Systems

The Lorenz oscillator in 1963 is an example of this.

Equations:

$$\bar{x} = a(y - x)$$
$$\bar{y} = x(b - z) - y$$
$$\bar{z} = xy - cz$$

Chaos is introduced in 3D systems.

Scribes: Mahika Bhalla
Vasudha Rengarajan
Jayant Madugula

COMS 6998-06: Computation and the Brain
Christos Papadimitriou

10/10/2018

3.7 Chaos

Chaos is defined in lecture as “exponentially small perturbations in parameters and initial conditions leading to qualitatively different behaviors.”
This can be shown as a system that never truly cycles despite appearing to exhibit periodic behavior (Lorenz).
Attractors are strange (fractal-like).
These systems cannot be solved nor understood.

3.8 Properties Against Chaos: Useful Properties in Dynamical Systems

(a) Conservative Systems: conserve energy (or other quantities of interest)

(b) Reversible Systems: can be “run backwards”

(c) Systems that progress towards convergence. These systems have a Lyapunov function.

3.9 Fundamental Theorem of Dynamical Systems

“Poincaré-Benedixson Envy” – the theorem attempts to restore the Poincaré-Benedixson Theorem to systems where chaos is present (when $D > 2$).

The theorem is as follows:

“Suppose for all $\epsilon > 0$ there is a N such that from x we can come back to x with a sequence of $< N$ steps alternating with jumps of length $< \epsilon$. Such a point x is chain-recurrent.”
“The domain of any dynamical system can be decomposed in the chain recurrent components (CRC) and the transient parts. There is a Lyapunov function that drives any transient point towards CRCs.” – Conley, 1984

The theorem was summarized in lecture (and the slides) as “if you squint a little, chaos goes away.”

4 Networks

4.1 Feedforward Networks

Feedforward Networks have two populations of neurons, with feedforward synaptic connections between the two populations. The connections go in one direction, from one population to the other.
We can define vectors u, v to be vectors of spiking rates and W to be a matrix of synaptic weights. v is the “destination” population, with the synaptic connections coming from u.

Scribes: Mahika Bhalla
Vasudha Rengarajan
Jayant Madugula
10/10/2018

COMS 6998-06: Computation and the Brain
Christos Papadimitriou

$$\tau * \frac{dv}{dt} = -vF(W * u)$$

4.2 Feedforward and Recurrent Networks

We still have two populations of neurons and the feedforward connections, but now the neurons in the destination population have synaptic connections to themselves.
Here, the vectors u, v are firing rates. W remains the matrix of synaptic weights. R is the matrix of recurrent synaptic weights.

$$\tau * \frac{dv}{dt} = -v + F(W * u + R * v)$$

An "interesting case" discussed in lecture has v containing both excitatory and inhibitory neurons. These can be described by negative columns in R. This is explained by Dale's Law. Dale's Law says if we have neurons, negative weights mean the neuron a connection is going to is inhibitory. Thus, W now has negative columns, since if one weight in a layer is inhibitory, then all are inhibitory.

4.3 Hopfield Net

The Hopfield Net is a discrete-time system that contains a set of nodes connected by edges. Each node can have one of two values: 1 or -1. An undirected edge e_{ij} connects node i to j and has an assigned weight w_{ij}.
We define a node i to be "happy" when:

$$\Sigma_j v_i v_j w_{ij} \geq 0$$

Essentially, a node is happy if it and "friends" (nodes with the same value) have a positive connecting weight and "enemies" (nodes with opposing value) have negative connecting weight.
The algorithm to satisfy a Hopfield Net is simple: while there exists an unhappy node, flip it. Hopfield proved in 1982 this system converges. This theorem was proven by showing the system's Lyapunov function always increases. This model can be applied to brain computation by making nodes neurons and edges bidirectional synapses. Thus, a weakness of this model is a lack of directed edges.
Finally, Hopfield Nets can be used for pattern completion. Equilibria are contained in "regions of attraction." The brain analogy here is associative memory. We end up in regions of attraction and can then go to the closest equilibria.