Signal propagation in neural networks

Eric Shea-Brown [1*], Alex Cayco-Gajic[1], and colleagues[1]
[1]University of Washington
[*]etsb@uw.edu

April 5, 2017

An overview for a workshop exploring the stability of signal processing in neural networks and introducing MATLAB coding, binary networks, and allied concepts from scratch.

Introduction

Signals about the sensory environment, motor commands, and more are transmitted from one place to the next in our brains. This has to occur in such a way that signal strength is maintained: it neither dies away or blows up. It turns out that achieving this demands interesting balancing act that is not fully understood. Intriguingly, it may be naturally learned over time as a consequence of network plasticity mechanisms. Let's dig into this! We'll start by learning the coding and math we need to explore this open question from scratch!

Suggested flow and pacing

Day 1

- Introduction to basic physiology and neuroscience. Slides: `introduction_to_neurons`
- Reading: First 7 pages of *Theoretical Neuroscience* by Dayan and Abbott
- Where does math come in? An overview from PI and/or members of research group on pillars of computational neuroscience, and on work being done in their group.
- Introduction to MATLAB: begin working through `MATLAB_tutorial`

Day 2

- Reading on signal propagation in neural networks: Kumar et al.
  Discussion points:
  - pg. 1: define/illustrate convergent/divergent connections (3rd paragraph)
  - pg. 2:
    * describe difference between strength synaptic couplings vs. abundance of shared connections (1st paragraph)
    * explain why transmission line is unreliable/synaptic failure (2nd paragraph)
    * describe idea of plasticity a bit, esp. hebbian "fire together wire together" plasticity (4th paragraph)
  - pg. 3: explain what feedforward networks embedded in recurrent networks means (3rd paragraph)
  - pg. 4:
    * define in and out degree (1st paragraph)
    * describe correlations and how they relate to dichotomous rate/synchrony idea (3rd paragraph)
  - pg. 5: explain spike volley width linear relationship (3rd paragraph)
  - pg. 6
    * figure 3: emphasize (1) what rasters/histograms/rate v.s group plots mean, (2) why we prefer the left panel to the right, and (3) that the lines in the rasters and histograms are evidence of correlations building up
- Work through MATLAB tutorial up to "plotting"

Day 3

- Work through MATLAB tutorial to the end of "neural explosion problems"

Day 4

Goal: code up a multilayer, nonlinear neural network which receives a simple signal at the first layer and propagates this through subsequent layers. The document below will guide you through this. Here, we suggest breaking up into two teams, each of which will code an element of this.

- Begin modeling of feedforward networks: work through `feedforward_network_coding_workshop`

Day 5

Goal: finish your network code, and design a "scoring" function that measures how faithfully the network transmits its input signal across layers.

- Continue modeling of feedforward networks: finish `feedforward_network_coding_workshop`

Day 6

Goal: learn about mechanisms and models of plasticity, the process by which networks rewire and in an "experience" (or activity) dependent way.

- Readings on plasticity: Bi and Poo, J Neuroscience 1998; Turrigiano and Nelson, "Hebb and Homeostatic Plasticity"
- Code up homeostatic and Hebbian plasticity rules MATLAB, and plan out how to incorporate them into your network code.

Days 7-9

- Research time! Your question: What combination of Hebbian and homeostatic plasticity rules, if any, allows the neural network to faithfully transmit signals?
- To evaluate signal transmission, use the network "score" introduced in the handout, or come up with your own.