CS2222 Project Presentation
EEG and SNNs

12/06/21

Agenda

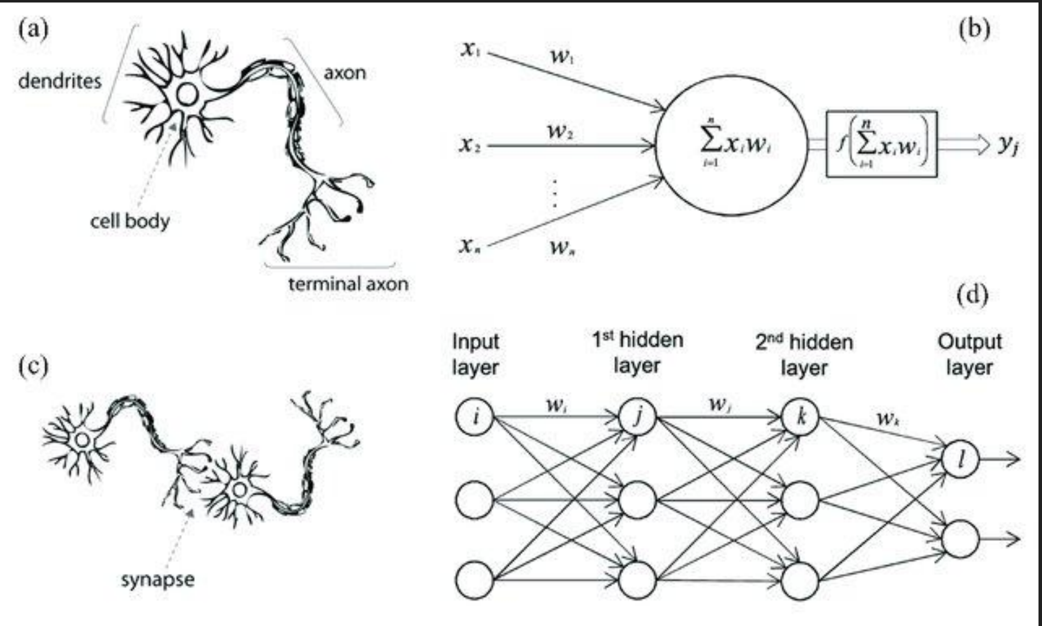
- Motivating (biologically plausible) SNN
- Dynamics, learning and data
- Code-gen neural simulators

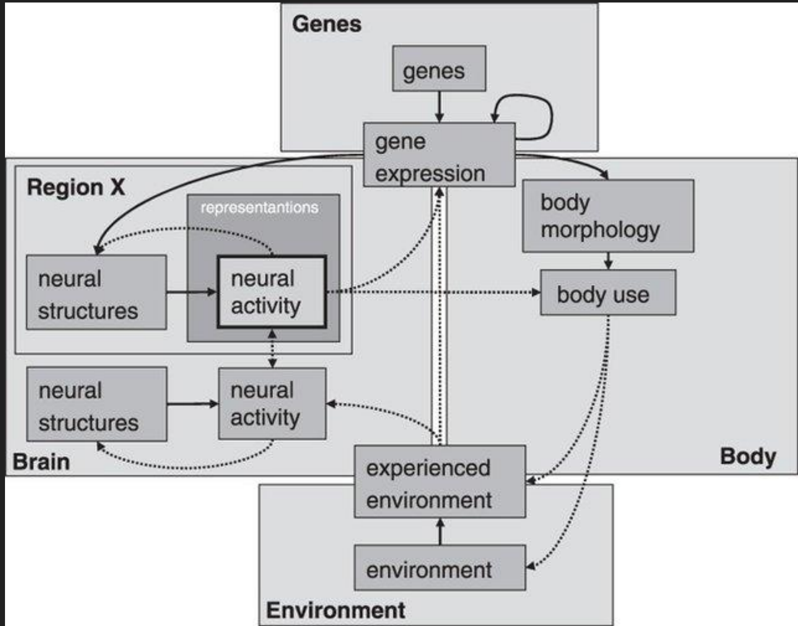
Goal: Attempt to answer practitioner/developer question
'***how is code-generation used in neuroscience?***'

Is cognition computation?
If so, how does brain represent information?

Neural networks and ‘brain-inspired’ models

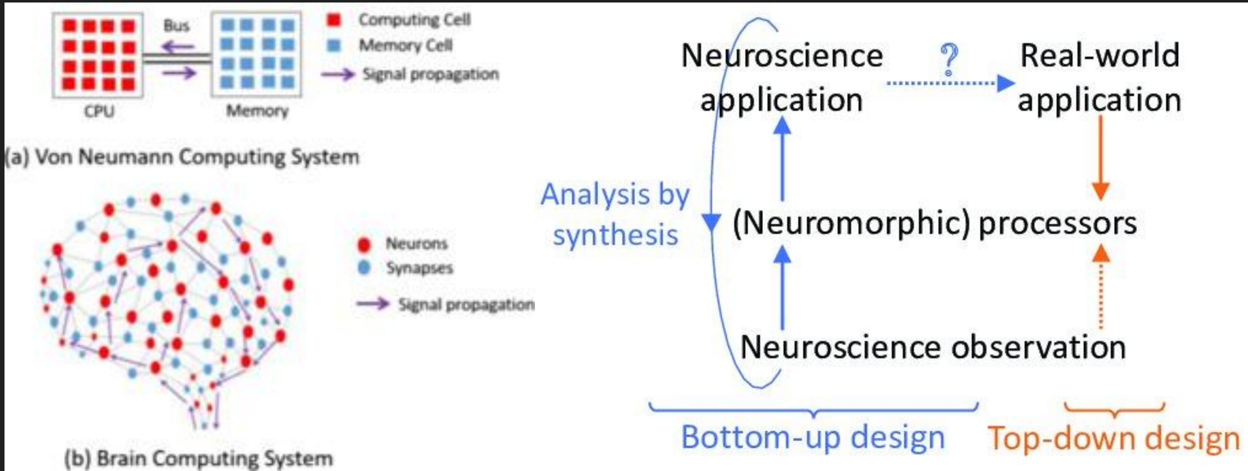
- Aristotle → Descartes → Kant → McCulloch-Pitts → Hodgin-Huxley → (Heidegger)
  - We’ve inherited a program of rationalizing/formalizing operation of mind
- Spiking neural networks bridge connectionist and dynamical theories





Why study spikes? Two disparate reasons

1. **Neuromorphic promise**: low-energy, parallel, async event-driven computation
    - Useful in robotics and embedded systems
    - Still gains to be made optimizing neural simulators on Von Neumann hardware
2. **Scientific inquiry**: biologically plausible models of neurons
    - Iterating on neurocognitive biomimicry (build better neural models for better performance)
    - Seeking adaptive complex systems description of brain



Some challenges with SNNs
(dynamics, learning, data)

Some challenges: dynamics

- Consider a single neuron
  - What quantities exist in this system (e.g. voltage)?

- Use <u>canonical neural model</u> approach
  - Different families of models for different neurons (w/ biological or aux variables)

- E.g. classes of excitability behaviour
  - *Class 1 integrator*: low frequency activation
    - Input current increase ~ spikes

  - *Class 2 resonator*: specific-frequency activation
    - Insensitive to input current increase

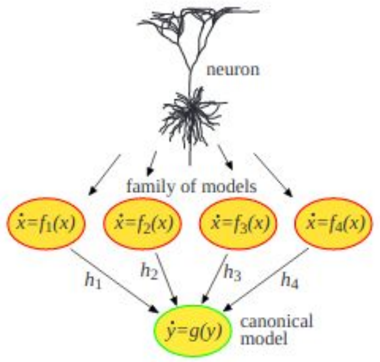
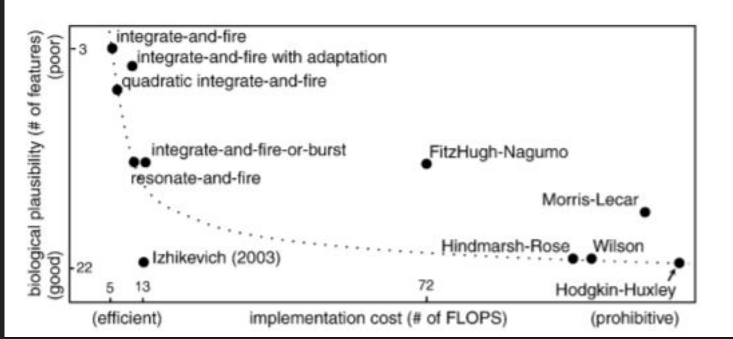


Figure 8.10: Dynamical system $\dot{y} = g(y)$ is a canonical model for the family $\{f_1, f_2, f_3, f_4\}$ of neural models $\dot{x} = f(x)$ because each such model can be transformed into the form $\dot{y} = g(y)$ by the piecewise continuous change of variables h_i.

E.g. Izhikevich model



- Izhikevich (2003) DEs for a single neuron
  - ***v***: membrane potential, ***u***: membrane recovery
    - **These are abstractions of the neurophysiology of observed cell behaviours**

$$\frac{dv}{dt} = 0.04v^2 + 5v + 140 - u + I$$

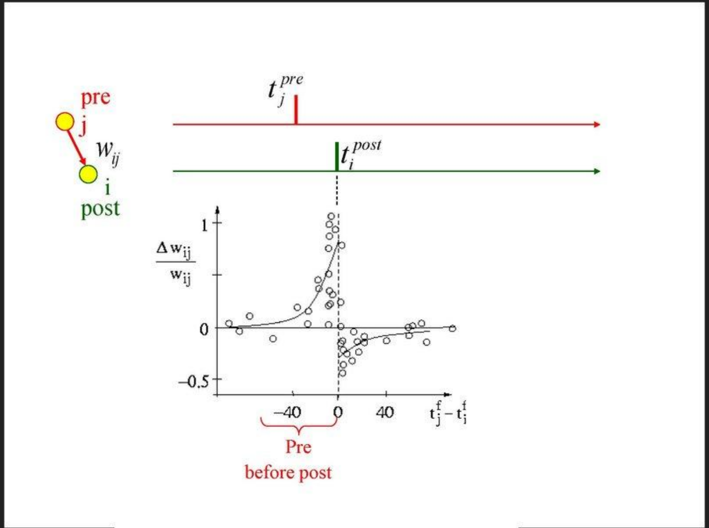
$$\frac{du}{dt} = a\left\{bv - u\right\}$$

$$\text{if } v \geq 30mV, \text{Then } v \leftarrow c, u \leftarrow u + d$$

$$I_i = \left(\sum_{j\in N_i} w_{ij}\frac{dv_j}{dt} + \frac{dW_{ij}}{dt}v_j\right)\left(1 - tanh\left(\sum_{j\in N_i} w_{ij}v_j\right)^2\right) + \frac{d\theta_i}{dt}, \qquad (5)$$

Some challenges: learning

- How will our network learn parameters? Need something unsupervised
  - Backprop popularized ANNs, but biological neurons can't backpropagate error gradients…
- Synaptic plasticity: strength between neurons change in response to stimuli
  - **Hebbian learning**: 'cells that fire together wire together'
  - [Spike-time-dependent plasticity](): if pre- and post-synaptic firing is short then strengthen

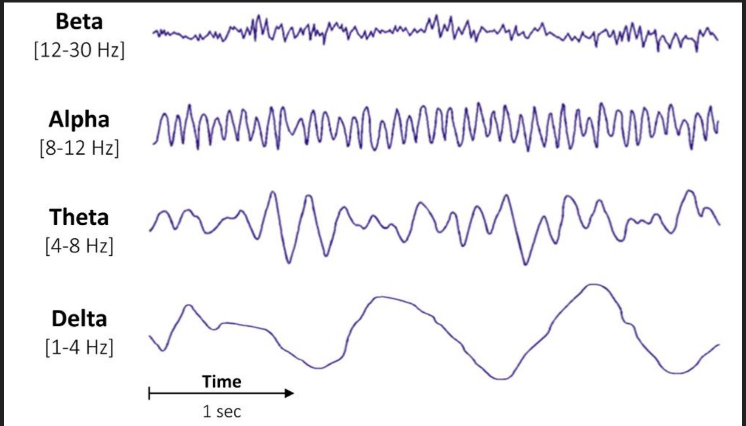


Some challenges: data

- Three classes over recall task: {resting state, word recall, image recall}
- Many ways to record a brain (e.g. PET, CT, MEG, MRI, …)

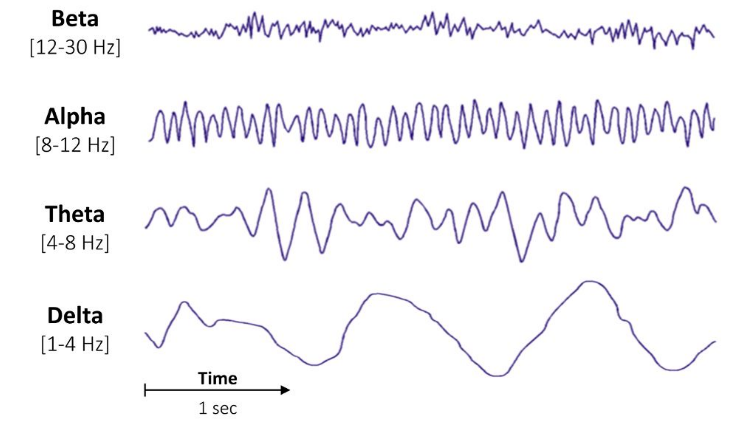
Some challenges: data

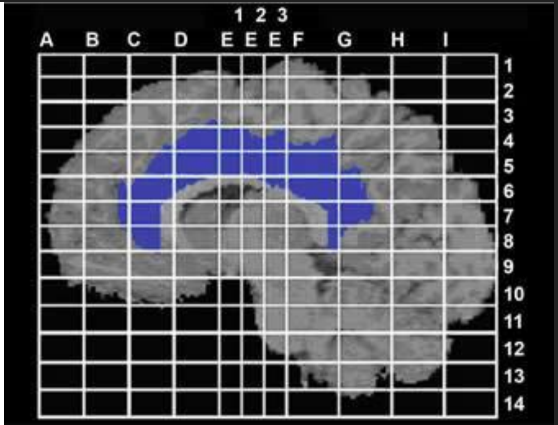
- Three classes over recall task: {resting state, word recall, image recall}
- Many ways to record a brain (e.g. PET, CT, MEG, MRI, …)
    - EEG
        - Measures electrical activity on surface of head
        - Many channels of continuous, non-smooth signal



Some challenges: data

- Three classes over recall task: {resting state, word recall, image recall}
- Many ways to record a brain (e.g. PET, CT, MEG, MRI, …)
  - EEG
    - Measures electrical activity on surface of head
    - Many channels of continuous, non-smooth signal
  - fMRI
    - Measures blood flow (metabolic function where neuronal activity present)





We have the ingredients for an **inverse problem**,

I.e. seeking (model of) neural process generating EEG data

Our experiment and data

- Want to model EEG data using populations of neurons

- Need software to model neurons, consider...
  - At what spatial scale?
    - *'Highly specific/descriptive neuronal models' vs 'large-scale brain network'*
  - Bottom-up or top-down approach? How abstract?
    - *'Functionalist view of the brain' vs '(neuro)constructivist'?*

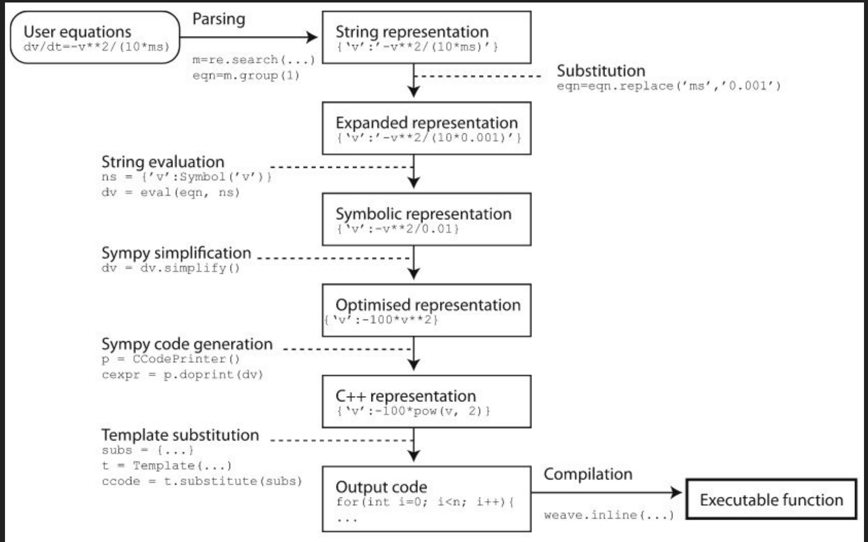
- ...so we used Brian
  - Uses code-gen to transform descriptions of neural models into system of ODEs

Code generation and neural simulators

	Models	Platforms	Techniques
Brian (2.1)	Point and multicompartmental neurons; plastic and static synapse models	CPUs; GPUs via GeNN	AST transformations; Symbolic model analysis; Code optimization
GeNN (2.2)	Models that can be defined by timestep update code snippet; mostly point neurons and synapses with local update rules	GPUs and CPUs	Direct code generation by a C++ program
Myriad (2.3)	Compartmental neurons; arbitrary synapse models	CPUs; GPUs	Custom object models; AST transformations
NESTML (2.4)	Point neurons	CPUs via NEST	Custom grammar definitions; AST transformations; model equation analysis
NeuroML/LEMS (2.5)	Point and multicompartmental neurons; plastic and static synapse models	CPUs via NEURON and Brian; SBML	Procedural generation; template-based generation; semantic model construction
NineML (2.6)	Models defined by a hybrid dynamical system; mostly point neurons and synapses with local update rules	CPUs via NEURON, NEST and PyNN	symbolic analysis; template-based generation
NEURON/NMODL (2.7)	Point and multicompartmental neurons; plastic and static synapse models; linear circuits; reaction-diffusion; extracellular fields; spike and gap junction coupled networks	CPUs; GPUs via CoreNEURON	Custom grammar; parse tree transformations; GUI Forms
SpineML (2.8)	Models defined by a timestep update code snippet; mostly point neurons and synapses with local update rules; generic inputs support compartments and non-spiking components	CPU via BRAHMS and PyNN; GPU via GeNN and Neuorkernel	XSLT code templates and libSpineML
SpiNNaker (2.9)	Common point neuron models with either static of plastic synapses	SpiNNaker	Hand crafted modular source code, loaded through a complex mapping process from a graph representation
TVB-HPC (2.10)	Neural mass models	CPUs; GPUs	AST transformations

Brian: code-gen for neural simulator

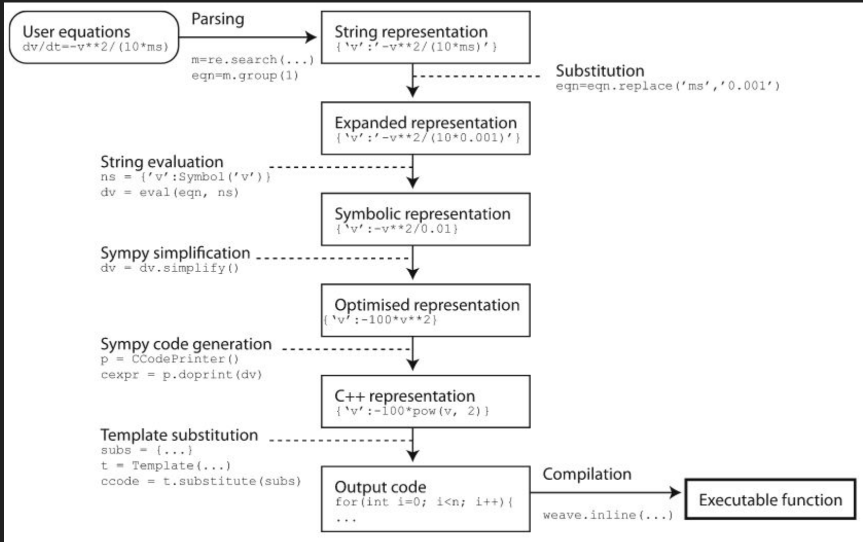
1. User specifies models (neuronal models, synapses connections, learning rule) in Python
2. Code generated for expensive operations from Brian templates
   a. *Equation strings translated into ODE solving code*
3. Simulation loop runs, executing fast code when needed
   a. *Memory is shared between Python and code, no copying required*



Brian: code-gen for neural simulator

1. User specifies models (neuronal models, synapses connections, learning rule) in Python
2. Code generated for expensive operations from Brian templates
   a. *Equation strings translated into ODE solving code*
3. Simulation loop runs, executing fast code when needed
   a. *Memory is shared between Python and code, no copying required*

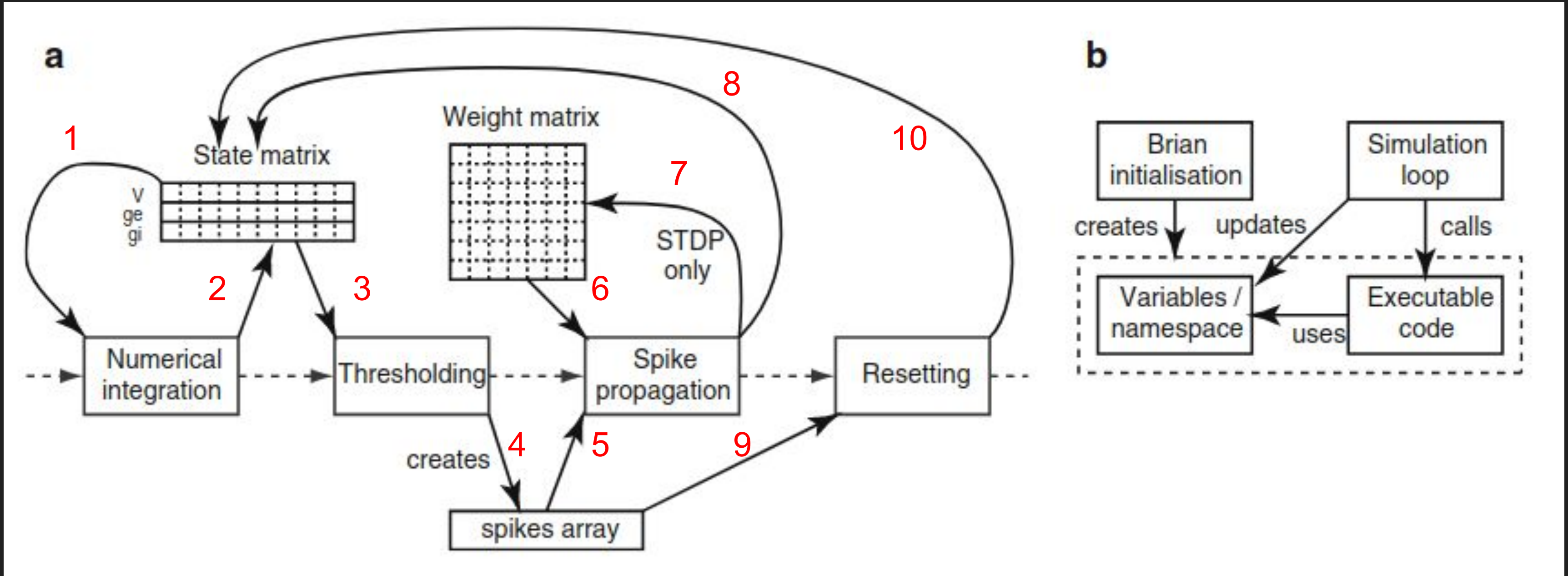
- **Pro**: practitioner's model becomes optimized simulation code that can be modified as needed
  - E.g. Change learning rules easily between runs
- **Con**: code-objects limited to available templates for substitution, limits the forms of models
  - E.g. Not all neural models come in differential form



So we take neural descriptions and generate code, how does the code run?

Brian: what is actually happening on device?

- Every solid arrow in (a) is an operation to/from memory (state/weight matrix)
  - E.g. update state values, update weight matrix, reset values on spike
- Use performance profiler to ask:
  - “Which steps (1-10) are the biggest bottleneck for the simulation loop?”

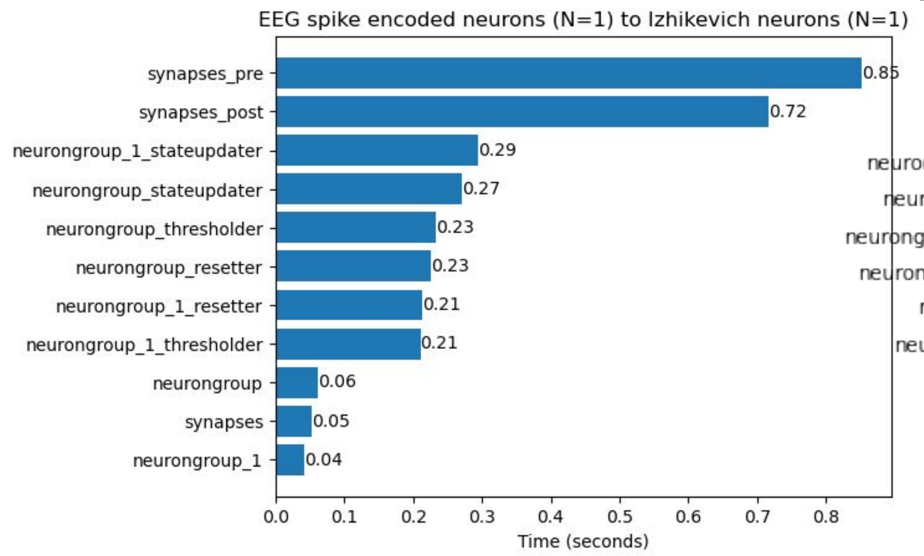


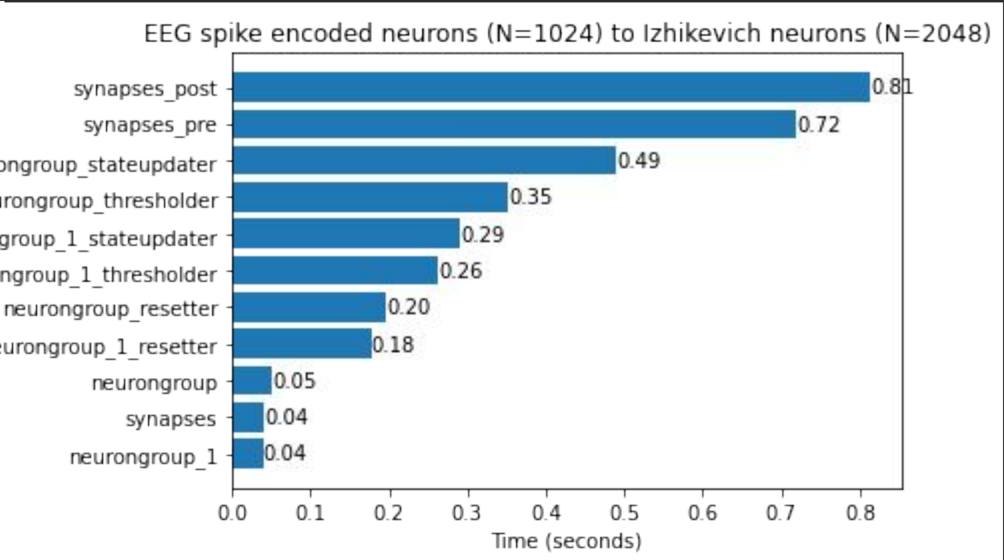
Experiment

- Encode EEG data into spike trains using BSA
- Use spike train as pre-synapse for neural population of Izhikevich neurons
- Train synapse weights using STDP
- Ask, what’s taking the longest?

Profiling

- Synaptic steps (i.e. learning) takes the most (simulation) time
- Integrating ODEs takes much less time, adding neurons costs more



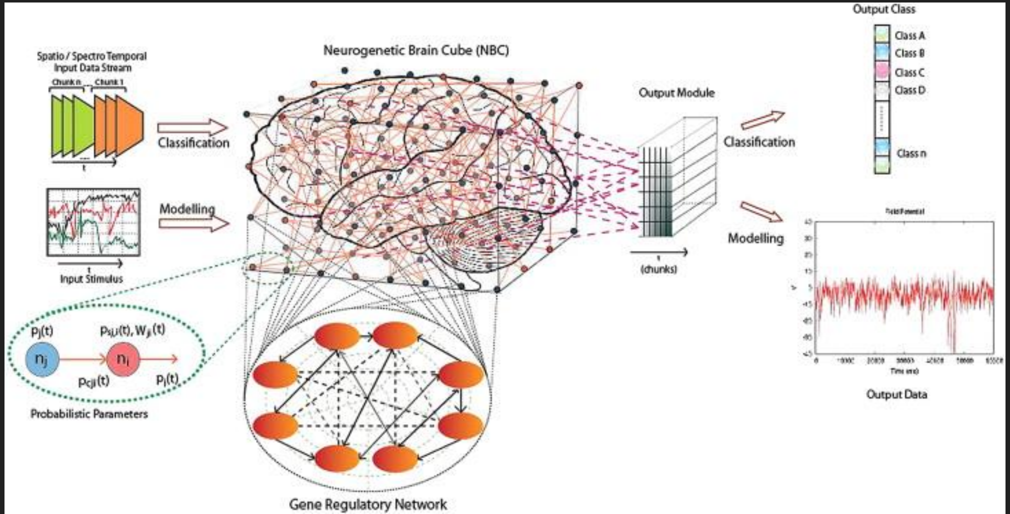


Thanks and good luck

Appendix slides

Doborjeh mindfulness paper

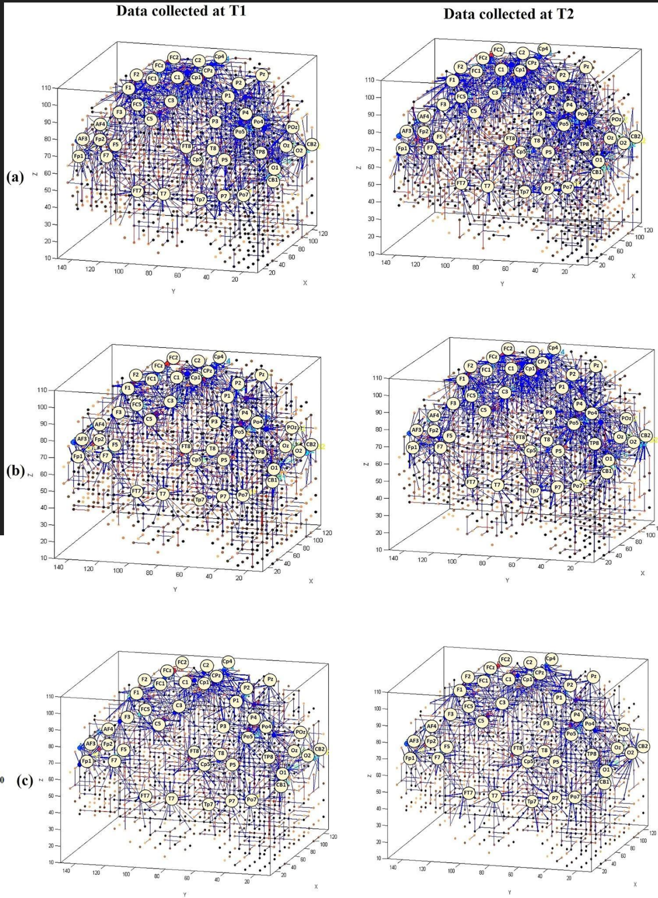
- **Idea**: train SNNs from EEG data before and after mindfulness training
  - Non-depressed (ND), depressed before but not after (D-), depressed before and after (D+)
  - Two networks (T1, T2) per subject, compare network weights (i.e. connectivity)
- **Result**: differences in regional activations for different subject conditions
- **So what?**: can use models to predict efficacy of mindfulness training

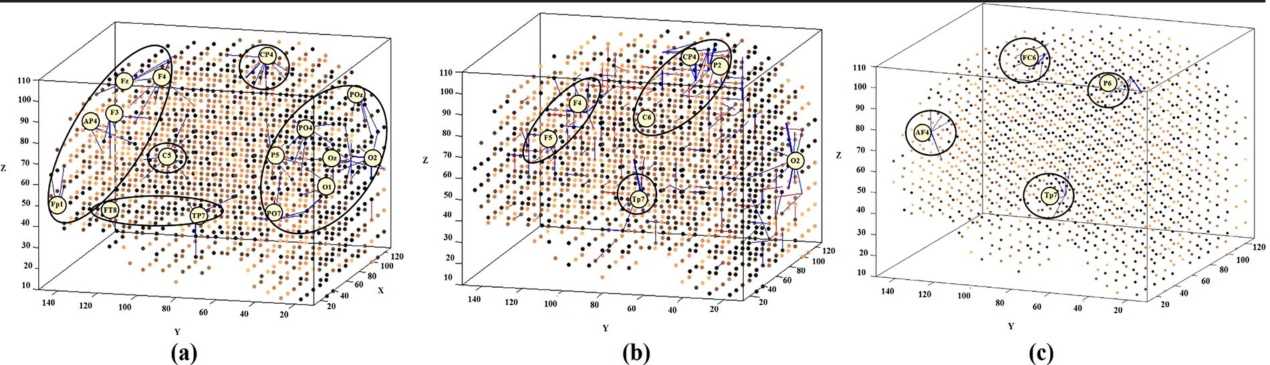


What does the data training look like?

- Resting state EEG mapped to a brain atlas
  - (a): non-depressed, ND
  - (b): depressed mindfulness responsive, D-
  - (c): depressed mindfulness non-responsive, D+

- Found regional differences between subjects when comparing before/after mindfulness training

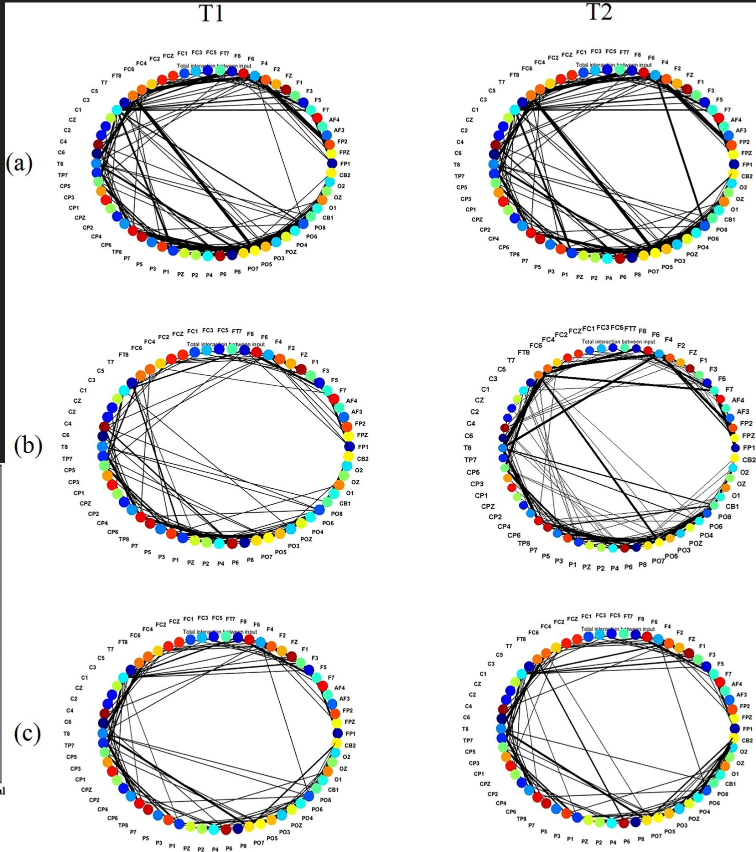


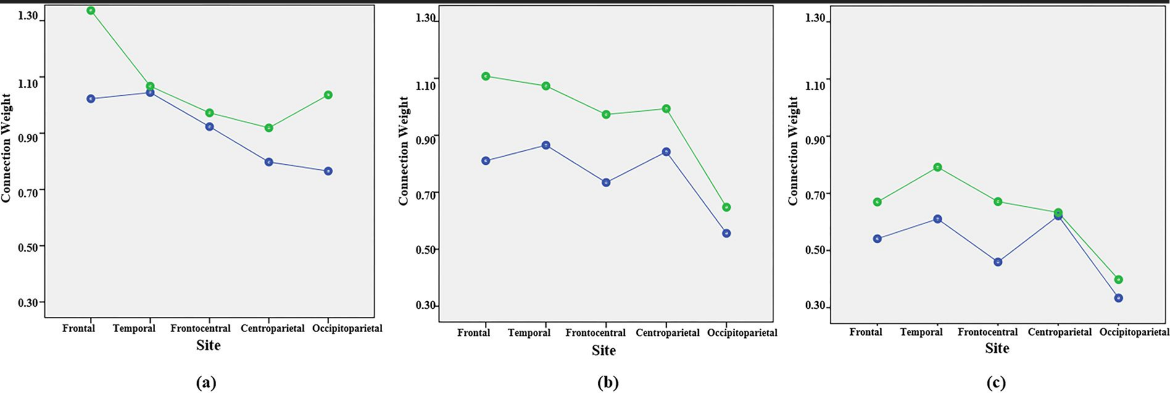


What kind of differences?

- Different subject groups had different levels of spike transmission between regions

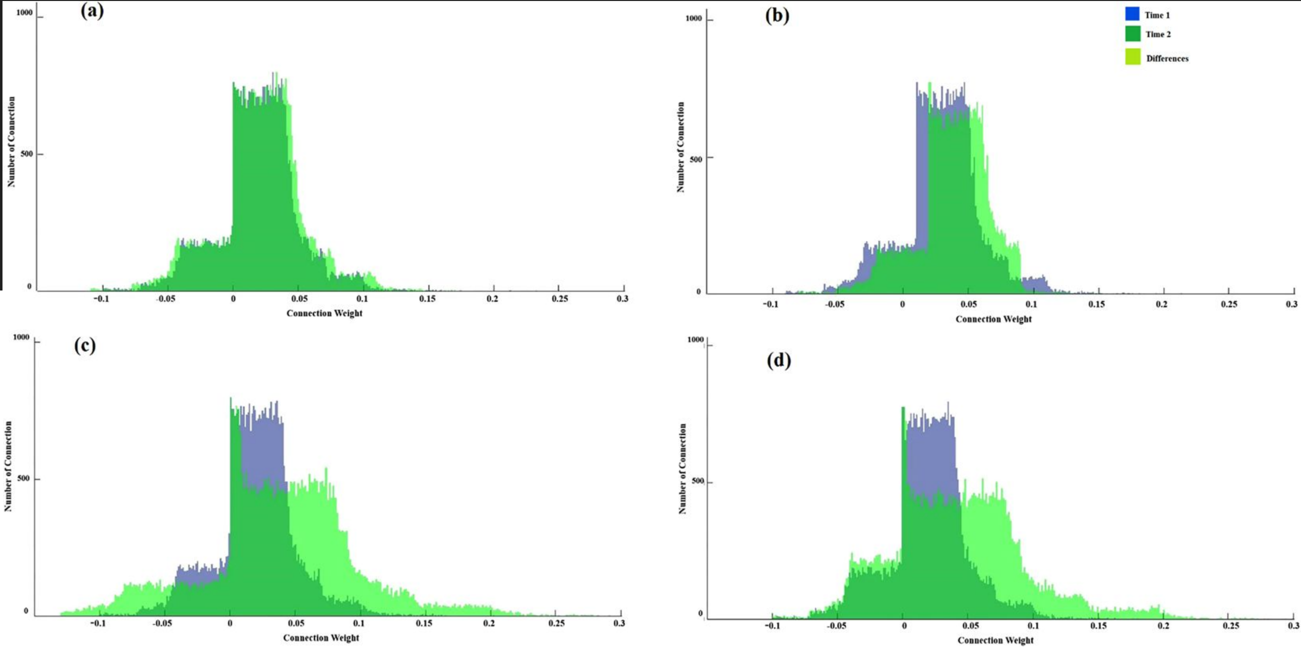
- Using SNNs, model differences between subjects at the neural level

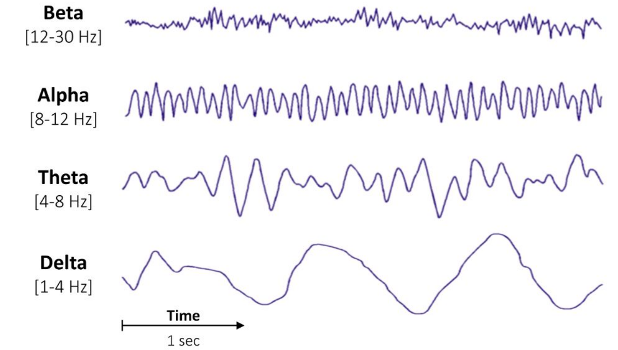




What about the D+ group?

- Different bandwidths responded differently to mindfulness training
  - (a) Delta, (b) Theta, (c) Alpha, (d) Beta
  - Histogram of connection weights by value shows (c,d) bandwidths spread out, (b) shifted





Plausibility and computational constraints

Models	biophysically meaningful	tonic spiking	phasic spiking	tonic bursting	phasic bursting	mixed mode	spike frequency adaptation	class 1 excitable	class 2 excitable	spike latency	subthreshold oscillations	resonator	integrator	rebound spike	rebound burst	threshold variability	bistability	DAP	accommodation	inhibition-indiced spiking	inhibition-induced bursting	chaos	# of FLOPS
integrate-and-fire	-	+	-	-	-	-	-	+	-	-	-	-	+	-	-	-	-	-	-	-	-	-	5
integrate-and-fire with adapt.	-	+	-	-	-	-	+	+	-	-	-	-	+	-	-	-	-	+	-	-	-	-	10
integrate-and-fire-or-burst	-	+	+		+	-	+	+	-	-	-	-	+	+	+	-	+	+	-	-	-		13
resonate-and-fire	-	+	+	-	-	-	-	+	+	-	+	+	+	+	-	-	+	+	+	-	-	+	10
quadratic integrate-and-fire	-	+	-	-	-	-	-	+	-	+	-	-	+	-	-	+	+	-	-	-	-	-	7
Izhikevich (2003)	-	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	13
FitzHugh-Nagumo	-	+	+	-		-	-	+	-	+	+	+	-	+	-	+	+	-	+	+	-	-	72
Hindmarsh-Rose	-	+	+	+			+	+	+	+	+	+	+	+	+	+	+	+	+	+		+	120
Morris-Lecar	+	+	+	-		-	-	+	+	+	+	+	+	+		+	+	-	+	+	-	-	600
Wilson	-	+	+	+			+	+	+	+	+	+	+	+	+	+		+	+				180
Hodgkin-Huxley	+	+	+	+			+	+	+	+	+	+	+	+	+	+	+	+	+	+		+	1200