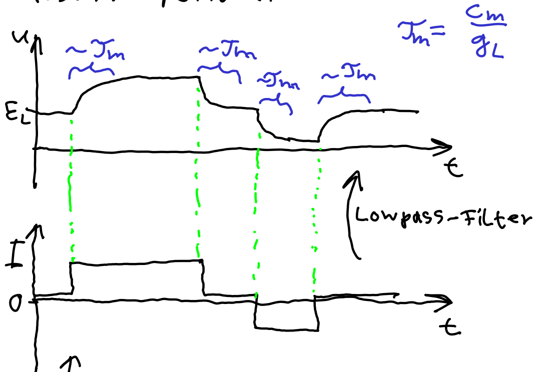
Cheatsheet BIC, Eekhof, 4110592

LIF:

Leaky Integrator ODE:

$$c_m \frac{du}{dt} = g_L (E_L - u) + I_{ext}$$

Assume point-like neuron $\tau_m = \frac{c_m}{g_L}$



Real neuron has much steeper spikes and more non-linearity.

⇒ Hodkin-Huxley

LIF compared to HH has:
- Absolute, not relative τ_{ref}
- Fixed threshold potential
- No inhibitory rebound

Output spike train:

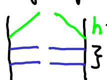
$$\rho_{(t)} = \sum_S \delta(t - t_s)$$

Activation Fct.:

$$\nu_{out}(I) = \lim_{T\to\infty}\left(\frac{\#spikes[0,T]}{T}\right)$$

Hodkin-Huxley:

Voltage-gated ion channels with prob. of being open m, h, k, concatenated:



$P_{open} = h \cdot m^2$

u-dependent: $\frac{dx}{dt} = \frac{1}{\tau_{x(u)}}(x_{0(u)} - x)$ $(x \in \{m, h, k\})$

$$c_m \frac{du}{dt} = g_L(E_L - u) + g_{Na} m^3 h (E_{Na} - u) + g_K n^4 (E_K - u) + I_{ext}$$

⇒ No firing threshold, (post) inhibitory rebound, resonance

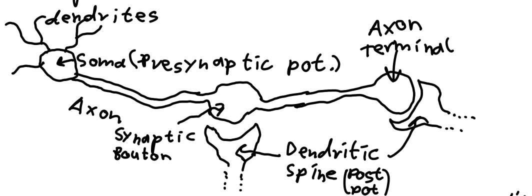
Calculate firing rate/activation fct.:

$\nu_{(I)} = (T_{inter\ spike\ interval})^{-1}$

$T_{isi} = T + \tau_{ref}$

↳ setze $V_{Thres} \overset{!}{=} u(T)$ und stelle nach T um

Biological neuron:



Signal speed: $1-10 \frac{m}{s}$; $10-100 \frac{m}{s}$ with myelin sheets

Resting potential: ≈ -70 mV

Density: $10^8 - 10^9 \frac{Syn}{mm^3}$; $10^3 - 10^4 \frac{Syn.}{Neuron}$

Syn. Cleft: 20 nm

Dales principle: A neuron releases the same trasmitters at everyone of its synapses.

COBA:

$$c_m \frac{du}{dt} = g_L(E_L - u) + g_{ext}(t)(E_{ext} - u) + g_{inh}(t)(E_{inh} - u) + I^{ext}$$

$$g_{exc/inh} = \sum_{\substack{\text{input Neuron}\\ k}} \sum_{\substack{\text{spike}\\ s}} w_k \underbrace{\kappa_{exc/inh}}_{\text{Kernel}}(t - t_s^k)$$

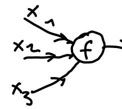
CUBA:

$$c_m \frac{du}{dt} = g_L(E_L - u) + I^{syn}_{(t)} + I^{ext}$$

$$I_{syn} = \sum_k \sum_s w_k \kappa_{exc/inh}(t - t_s^k)$$

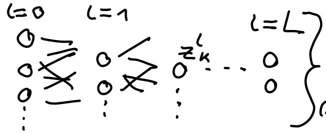
COBA	vs.	CUBA
- DYNAMIC RANGE limited by $E_{EX/INH}$		- DYN. RANGE NOT limited
- PSPs influence each other		- PSP of same syn. stays the same
- $\tau_{eff(t)} = \frac{c_m}{g_{TOT}(t)} = \frac{c_m}{g_L + g_{Exc(t)} + g_{inh}}$		- τ_m static $\tau_m = \frac{c_m}{g_L}$

Neural Networks:



$y_k = f\left(\sum_i x_i w_{ik} + b_k\right)$ ← non linear (ReLU; Sigmoid) $\max(0,x)$; $\frac{1}{1+e^{-x}}$

Multilayer perceptron:



Universal approximation Theorem (can approximate any reasonable function) (at least 2 layers)

Training: Gradient ascend on:

$$\mathcal{L} = -\frac{1}{2}\sum_{\{x\}} \|z^L_{(x)} - f_{(x)}\|^2 \qquad z^*_j(x) = f(x)$$

$$\mathcal{L}_{(x)} = -\frac{1}{2}\sum_j |z^L_j(x) - z^*_j(x)|^2$$

Update rules:

$$w_{ki}^l \to w_{ki}^l + \eta \frac{\partial \mathcal{L}}{\partial w_{ki}^l}; \quad b_k^l \to b_k^l + \eta \frac{\partial \mathcal{L}}{\partial w_{ki}^l}$$

Top layer update:

$$\frac{\partial \mathcal{L}}{\partial w_{ki}} = \sum_{\{x\}} \frac{\partial}{\partial w_{ki}}\left(-\frac{1}{2}\left|\sum_j f(b_k^L + \sum_i w_{ki} z_i^{L-1}) - f_{(x)}\right|^2\right)$$

$$= \sum_{\{x\}} \underbrace{\left(-(z_k^L - z_k^*) f'(b_k^L + \sum_i w_{ki} z_i^{L-1})\right)}_{\delta_k^L} z_i^{L-1}$$

Deep Layer update:

$$\frac{\partial \mathcal{L}_{(x)}}{\partial w_{ki}} = \frac{\partial \mathcal{L}_{(x)}}{\partial z^L_{(x)}} \cdot \frac{\partial z^L_{(x)}}{\partial z^{L-1}_{(x)}} \cdot \ldots \cdot \frac{\partial z^{l+1}_{(x)}}{\partial z^l_{(x)}} \cdot \frac{\partial z^l_{(x)}}{\partial w_{ki}} \quad \text{(Chain Rule)}$$

↳ Not just chain rule → can be calculated interatively by backpropagating error terms.

Training algorithm:

1) initialize parameters randomly; 2) Present input sample; 3) Forward propagate activation; 4) Backward propagate error; 5) Calculate gradients; 6) Update weights; Repeat!

↳ Beware of overfitting

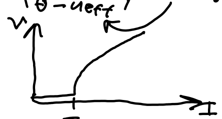
Neuron states:

Firing rate: $\nu_k = \langle \rho_k^{(t)} \rangle_T = \frac{\int_0^T \rho_k^{(t)} dt}{T} = \frac{n_{spikes[0,T]}}{T}$

↳ At constant input I:

$$\nu_{(I)} = \frac{1}{\tau_{ref} + \tau_m \ln\left(\frac{E_R - u_{eff}}{\theta - u_{eff}}\right)} \qquad u_{eff} = \frac{I}{g_L} + E_L$$

$\tau_{ref} = 0$, $I \to \infty$: $\approx \frac{1}{\tau_m(\theta - E_R)}$

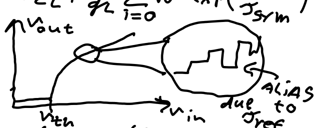


↳ At const. input rate ν_{in}:

$$u^{(2)}_{eff(t)} = E_L + \frac{I}{g_L} = E_L + \frac{1}{g_L}\sum_{i=0}^{\infty} w \cdot \exp\left(\frac{t - i\Delta t}{\tau_{syn}}\right)$$

where $\Delta t = \frac{1}{\nu_{in}}$

↳ Better model of multiple neurons inputs:



Poisson noise: Also reduces aliasing, noisy curve, $Var(\nu) \to 0$ for $n \to \infty$, no more absolute ν_{th}, reduces initial kink

Spike time encoding:
- Easy to implement in biology and neuromorphic hardware
- Problem: Causal sets need to be known (only inputs before output time)

Loss function:

$$\mathcal{L} = \sum_{\text{Samples}} \sum_i |t_i - t_i^*|^2$$

Sampling:

If multiple outputs are equally likely, use probabilistic sampling.

Probabilities:

$P(X = x \wedge Y = y) =: P_{(x,y)}$

Sum Rule: $P_{(x)} = \sum_y P_{(x;y)}$

Product rule: $P_{(x,y)} = P_{(y|x)} P_{(x)}$ ← Prior

Bayes Theorem: $P_{(x|y)} = \frac{P_{(y|x)} \cdot P_{(x)}}{P_{(y)}}$ (Posterior cond. prob.; Likelihood; evidence)

Independence of Random Variables $\Leftrightarrow P_{(a,b)} = P_{(a)} \cdot P_{(b)}$

Cond. Indep. $\Leftrightarrow P_{(a,b|c)} = P_{(a|c)} \cdot P_{(b|c)}$

Markov Random field:

$$P(\vec{z}) = \frac{1}{Z} e^{-E(\vec{z})}$$

In Boltzmann case:

$$E(\vec{z}) = \frac{1}{2}\sum_{ij} w_{ij} z_i z_j + \sum_i b_i z_i$$

⇒ Problem: Too expensive

⇒ Sample most likely probs.

Gibbs-Sampling:

1) Prepare $z_k^{(t+1)} = 1$;

2) Calculate $P(z_k^{(t+1)} = 1 | \text{state of all others})$;

3) Accept $z_k^{(t+1)} = 1$ with that probability, otherwise $= 0$;

4) Go to next unit $k \to k+1$ and add new state to set of sampled states

Poisson-Dist: $P_T(N) = \lambda^N \frac{e^{-N}}{N!}$

Plasticity:

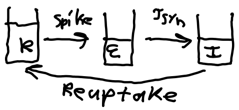
- Biological neurons mostly store info. and memory in network structure
- Bio. neu. adapt and form their structure based on local history

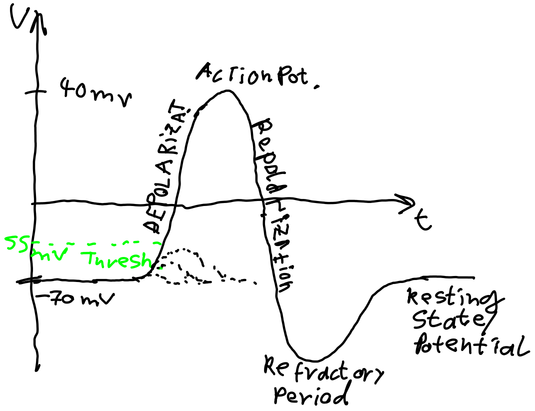
⇒ Short term plast. (STP)

↪ Timescale 10ms – 1s

↪ TSO-Mechanism: Categorize NTMs: Recovered; Effective; Inactive $(R+E+I=1)$

Ground state: $\begin{pmatrix} R \\ E \\ I \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$





⇒ Long term plasticity:

Bi & Poo: Causal (Pre before Post) ⇒ Long term potentiation; Acausal (Pre after Post) ⇒ Long term depression

Top-Down-Modelling:

Obtain average stimulus before spike (spike triggered average STA) through white noise stimulus:

$C(\tau) = \langle x(t_s - \tau) \rangle_{t_s} = \frac{1}{\nu} C_{\rho x}(-\tau)$

crosscor. of ρ and x with lag $-\tau$

Neural encoding in biology:

Rate code:
- Long interpretation time

Place code:
- Location/identity of neuron encodes information (e.g. Retina)

Temporal encoding:
- Spike timing encodes info (e.g. owl)

Time to first spike:
- E.g. Salamander retina

Phase code:
- Relative timing to oscillating signal (e.g. in hyppocampus)

HH-ion concentrations:

	in	out
Na^+	60mM	440mM
K^+	400mM	20mM

⇒ Ion concentrations create electrical potential. When certain voltage E_{Rev} is applied, the currents of the ions change sign

Spike in HH: By interplay of variable Na^+ and K^+ channel conductances, which can be self-facilitating under right conditions

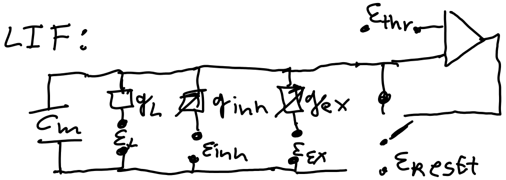
Postinhibitory Rebound:

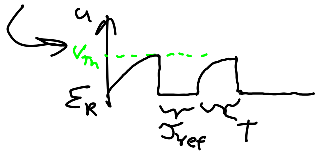
After hyperpolarization, Neuron gets depolarized, may spike

Lack of Rheobase:

Very low resting potential

LIF:





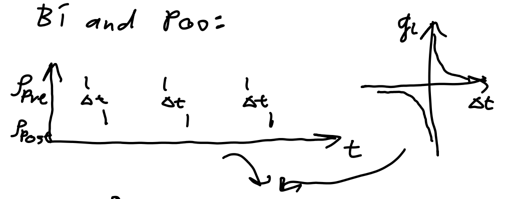
COBA shunting inhibition:

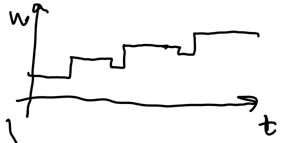
- Shunting synapse lowers its resistance, short-circuiting "shunting" the excitatory signal, thus inhibiting a spike

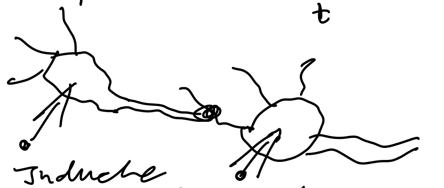
Markov-chain:

Collection of nodes connected via probabilities/weights, that represent transition probability. Allows investigation of how often certain states are occupied, iteratively

Bi and Poo:







Induce regular spike trains, and observe EPSP before and after