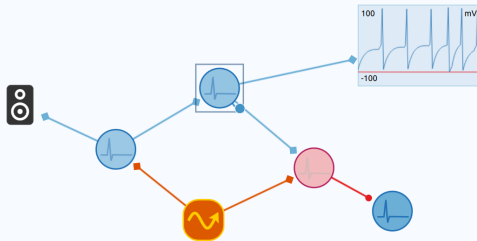


# Neuronify: a new tool for creating simple neural networks

Simen Tennøe,  
Svenn-Arne Dragly,  
Andreas V. Solbrå,  
Milad H. Mobarhan

CINPLA - University of Oslo  
Wednesday 20<sup>th</sup> May, 2015



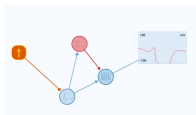
# In this lecture we will introduce Neuronify, a new tool for creating neural networks



Integrate and fire neurons

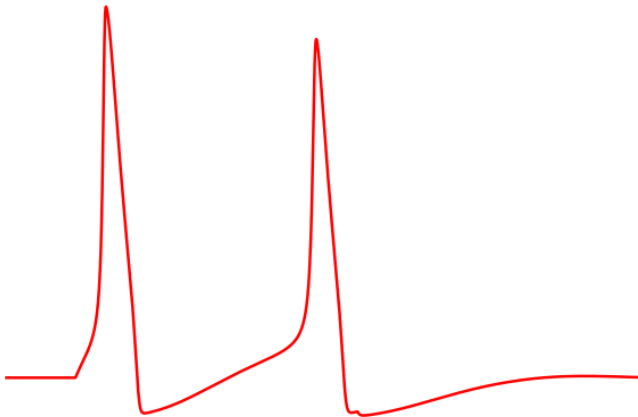


Neuronify

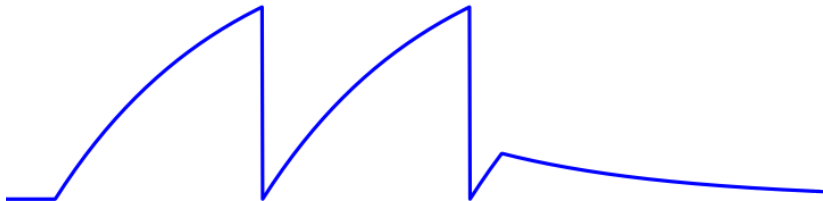


Exercises

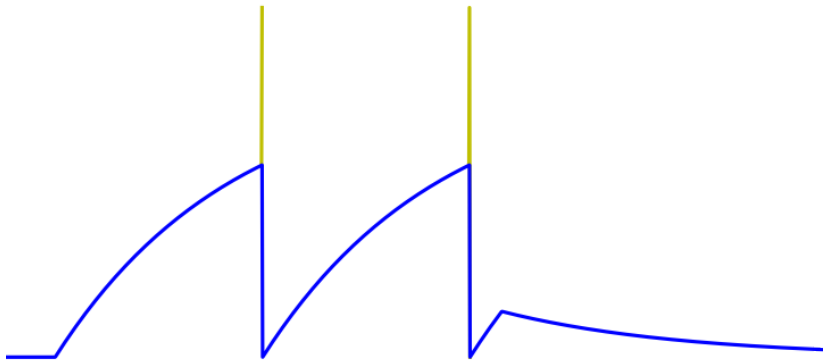
We do not need all information in the action potential for a neuron in a network and can create an approximation



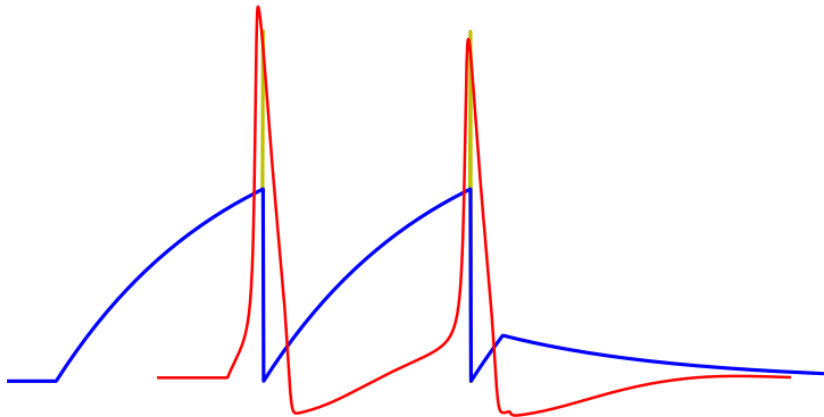
We do not need all information in the action potential for a neuron in a network and can create an approximation



We do not need all information in the action potential for a neuron in a network and can create an approximation



We do not need all information in the action potential for a neuron in a network and can create an approximation



The integrate and fire neuron is much faster to evaluate than the Hodgkin-Huxley neuron

Hodgkin-Huxley

The integrate and fire neuron is much faster to evaluate than the Hodgkin-Huxley neuron

## Hodgkin-Huxley

$$\begin{aligned}C \frac{dV}{dt} &= I_{inj} - \bar{g}_{Na} m^3 h (V - V_{Na}) - \bar{g}_K n^4 (V - V_K) - g_L (V - V_L) \\ \frac{dn}{dt} &= \alpha_n(V)(1 - n) - \beta_n(V)n \quad \frac{dm}{dt} = \alpha_m(V)(1 - m) - \beta_m(V)m \\ \frac{dh}{dt} &= \alpha_h(V)(1 - h) - \beta_h(V)h \quad \alpha_n(V) = \frac{0.01(V + 55)}{1 - \exp[-(V + 55)/10]} \\ \beta_n(V) &= 1.125 \exp[-(V + 65)/80] \quad \alpha_m(V) = \frac{0.1(V + 40)}{1 - \exp[-(V + 40)/10]} \\ \beta_m(V) &= 4 \exp[-(V + 65)/18] \quad \alpha_h(V) = 0.07 \exp[-(V + 65)/20] \\ \beta_h(V) &= \frac{1}{1 + \exp[-(V + 35)/10]}\end{aligned}$$



The integrate and fire neuron is much faster to evaluate than the Hodgkin-Huxley neuron

## Hodgkin-Huxley

$$\begin{aligned}C \frac{dV}{dt} &= I_{inj} - \bar{g}_{Na} m^3 h (V - V_{Na}) - \bar{g}_K n^4 (V - V_K) - g_L (V - V_L) \\ \frac{dn}{dt} &= \alpha_n(V)(1 - n) - \beta_n(V)n \quad \frac{dm}{dt} = \alpha_m(V)(1 - m) - \beta_m(V)m \\ \frac{dh}{dt} &= \alpha_h(V)(1 - h) - \beta_h(V)h \quad \alpha_n(V) = \frac{0.01(V + 55)}{1 - \exp[-(V + 55)/10]} \\ \beta_n(V) &= 1.125 \exp[-(V + 65)/80] \quad \alpha_m(V) = \frac{0.1(V + 40)}{1 - \exp[-(V + 40)/10]} \\ \beta_m(V) &= 4 \exp[-(V + 65)/18] \quad \alpha_h(V) = 0.07 \exp[-(V + 65)/20] \\ \beta_h(V) &= \frac{1}{1 + \exp[-(V + 35)/10]}\end{aligned}$$

## Integrate and fire

The integrate and fire neuron is much faster to evaluate than the Hodgkin-Huxley neuron

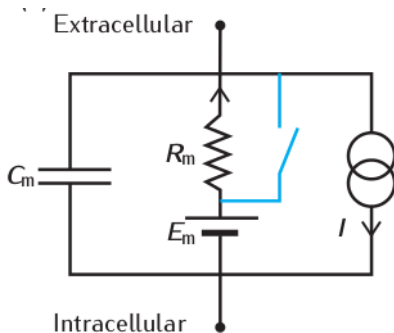
## Hodgkin-Huxley

$$\begin{aligned}C \frac{dV}{dt} &= I_{inj} - \bar{g}_{Na} m^3 h (V - V_{Na}) - \bar{g}_K n^4 (V - V_K) - g_L (V - V_L) \\ \frac{dn}{dt} &= \alpha_n(V)(1 - n) - \beta_n(V)n \quad \frac{dm}{dt} = \alpha_m(V)(1 - m) - \beta_m(V)m \\ \frac{dh}{dt} &= \alpha_h(V)(1 - h) - \beta_h(V)h \quad \alpha_n(V) = \frac{0.01(V + 55)}{1 - \exp[-(V + 55)/10]} \\ \beta_n(V) &= 1.125 \exp[-(V + 65)/80] \quad \alpha_m(V) = \frac{0.1(V + 40)}{1 - \exp[-(V + 40)/10]} \\ \beta_m(V) &= 4 \exp[-(V + 65)/18] \quad \alpha_h(V) = 0.07 \exp[-(V + 65)/20] \\ \beta_h(V) &= \frac{1}{1 + \exp[-(V + 35)/10]}\end{aligned}$$

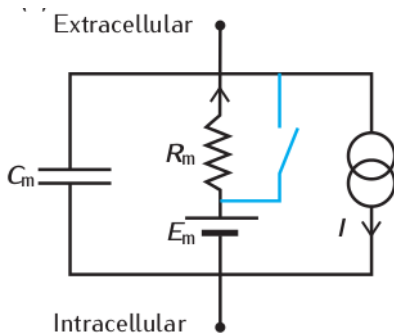
## Integrate and fire

$$C_m \frac{dV(t)}{dt} = - \frac{V(t) - E_m}{R_m} + I$$

The integrate and fire model is modeled as a simple RC circuit that is shortcircuited once a threshold is reached

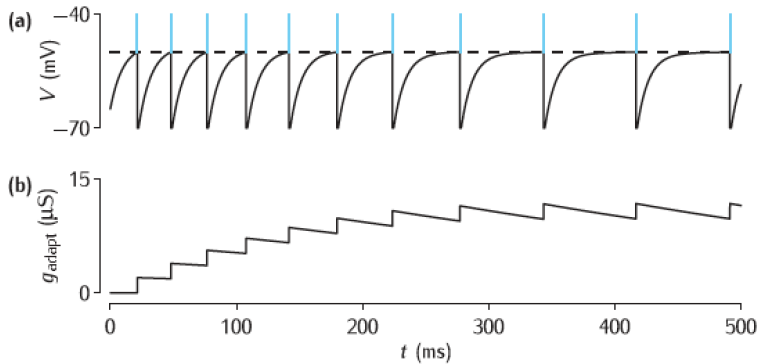


The integrate and fire model is modeled as a simple RC circuit that is shortcircuited once a threshold is reached



$$C_m \frac{dV(t)}{dt} = -\frac{V(t) - E_m}{R_m} + I$$

# Adaptive Firing Rate



# Adaptive Firing Rate

This can be modeled by a reverse current

$$I_{\text{adapt}} = g_{\text{adapt}}(V - E_m)$$

where the conductance is incremented by a step  $\Delta g$  on each action potential, and then decays.

$$\frac{d}{dt}g_{\text{adapt}} = -\frac{g_{\text{adapt}}}{\tau_{\text{adapt}}}.$$

## Adaptive Firing Rate



## Spike generators

Neurons exhibit irregular firing patterns. In the Stein model this is achieved by letting the neuron receive random inhibitory and excitatory spikes, which follow a Poisson process,

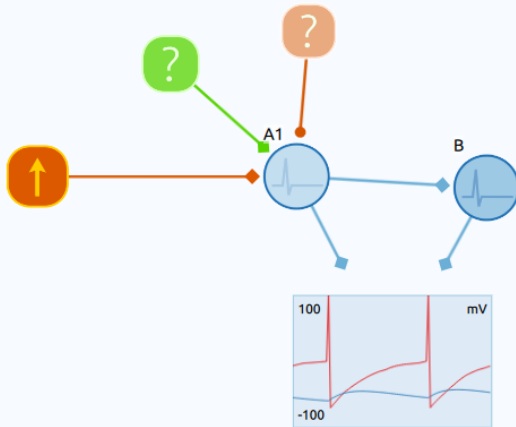
$$Pr(X = n) = f(n) = \frac{\lambda^n e^{-\lambda}}{n!}$$



## Spike generators - Implementation

At each time step we draw a random number  $x$ . If  $x < r\Delta t$ , the generator fires. Otherwise it does not.

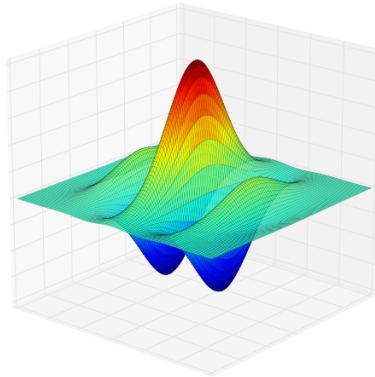
# Spike generators - Neuronify



# Receptive Field

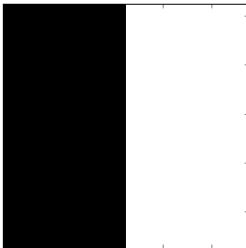
$$r(t) = r_0 + \int \int D(\mathbf{r}, \tau) s(\mathbf{r}, t - \tau) d\tau d\mathbf{r} \quad (1)$$

Gabor function

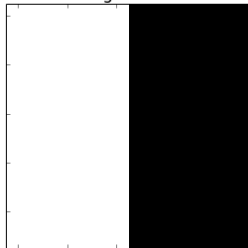


# Receptive Field

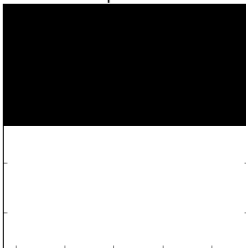
Left-OFF



Right-OFF



Up-OFF



Bottom-OFF

