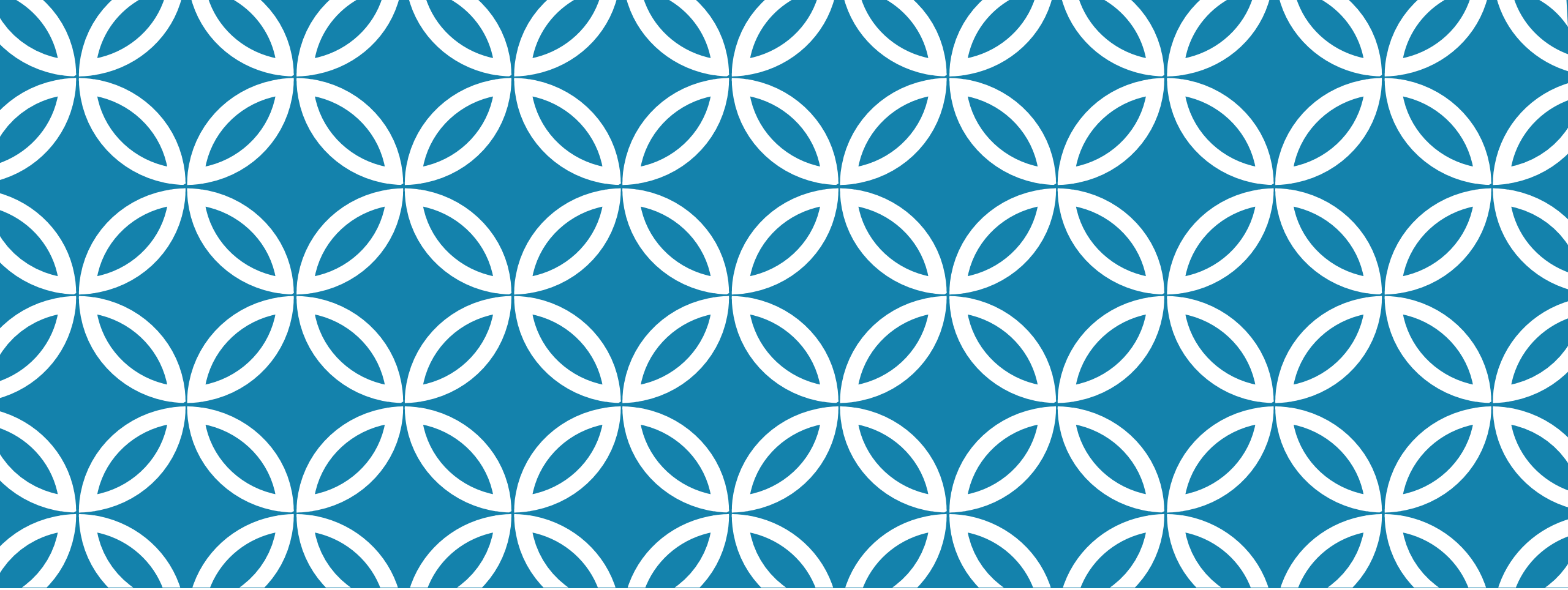


SPIKING NEURAL NETWORK ACTOR-CRITIC REINFORCEMENT LEARNING WITH TEMPORAL CODING AND REWARD MODULATED PLASTICITY

Vlasov D.S., Rybka R.B.,
Serenko A.V., Sboev A.G.

MOTIVATION AND GOAL

Spiking neural networks implemented in memristor-based hardware can provide fast and efficient in-memory computation as their weights can be changed locally in a self-organized manner without the demand for high-precision changes calculated with the use of information almost from the entire network.

This problem is rather relevant for solving control tasks with neural-network reinforcement learning methods, as those are highly sensitive to any source of stochasticity in a model initialization, training, or decision-making procedure.

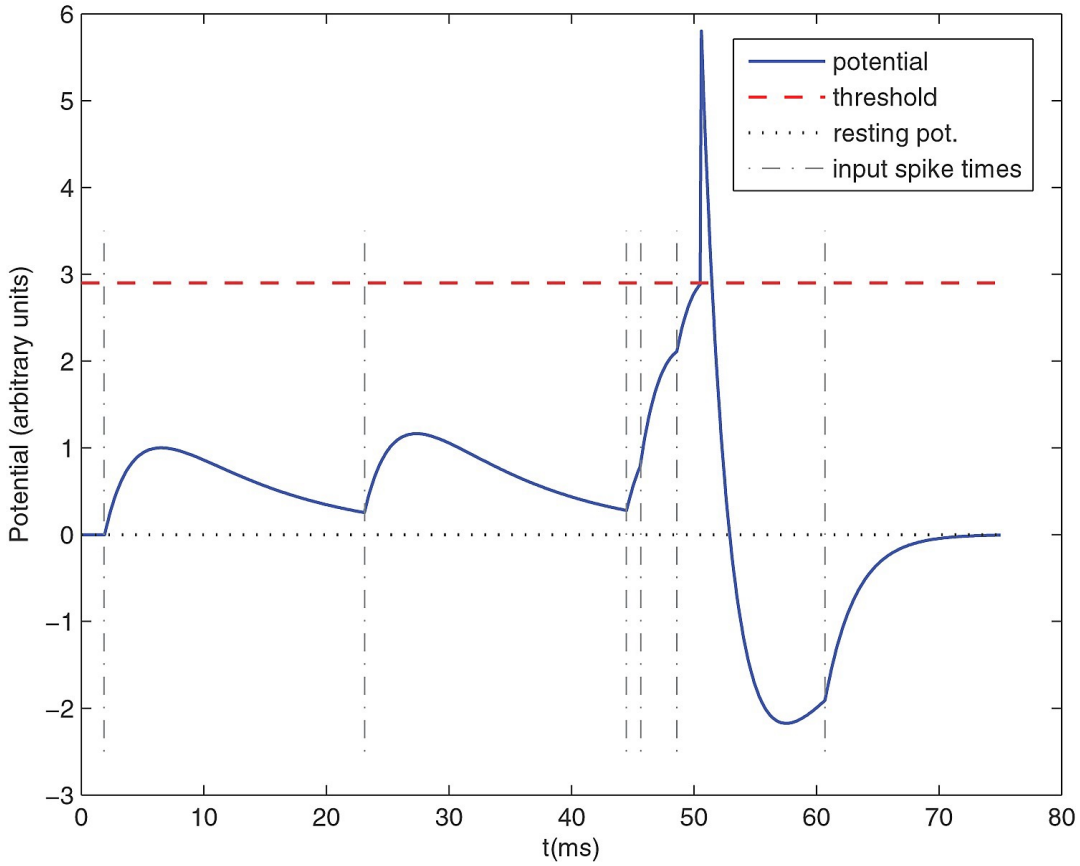
The goal is to develop an online reinforcement learning algorithm in which the change of connection weights is carried out after processing each environment state during interaction-with-environment data generation.

NEURON MODEL

Leaky integrate-and-fire:

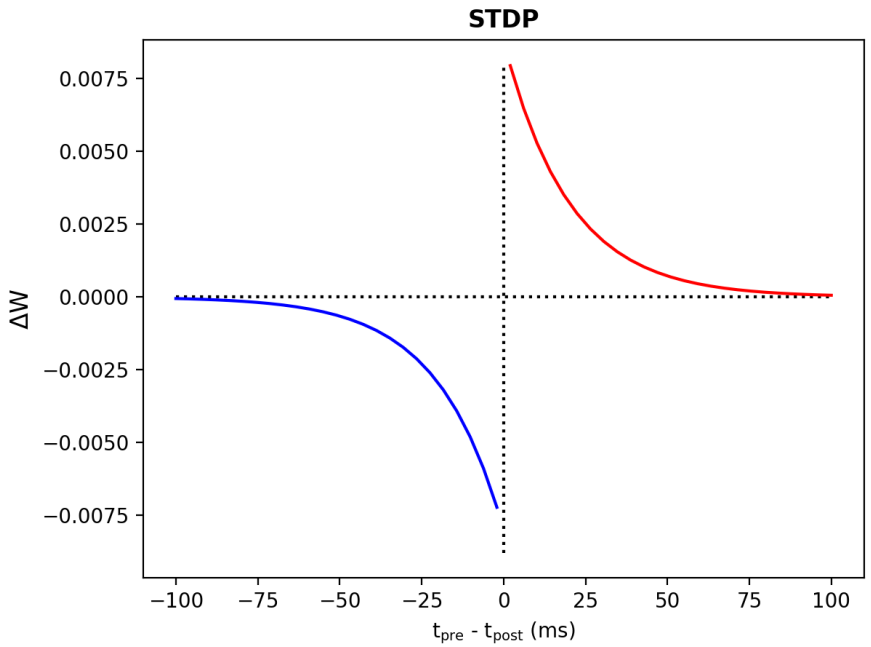
$$\frac{dV}{dt} = \frac{-V(t)-V_{rest}}{\tau_m} + \frac{I_{syn}(t)}{C_m} + I_{ext}(t)$$

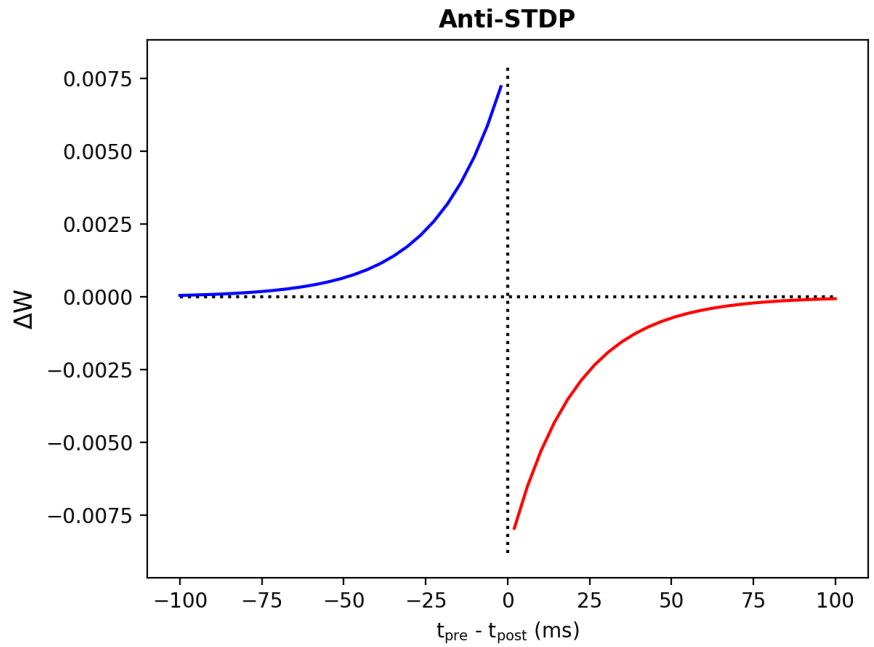
$$I_{syn}(t) = \sum_i w_i \sum_{s \epsilon S_i} \frac{q_{\text{syn}}}{\tau_{\text{syn}}} e^{\frac{t-s}{\tau_{\text{syn}}}} \Theta(t-s)$$



Masquelier, Timothée; Guyonneau, Rudy; J. Thorpe, Simon (2015): Leaky Integrate-and-Fire (LIF) neuron.. PLOS ONE. Figure. https://doi.org/10.1371/journal.pone.0001377.g003

SPIKE TIME DEPENDENT PLASTICITY MODEL

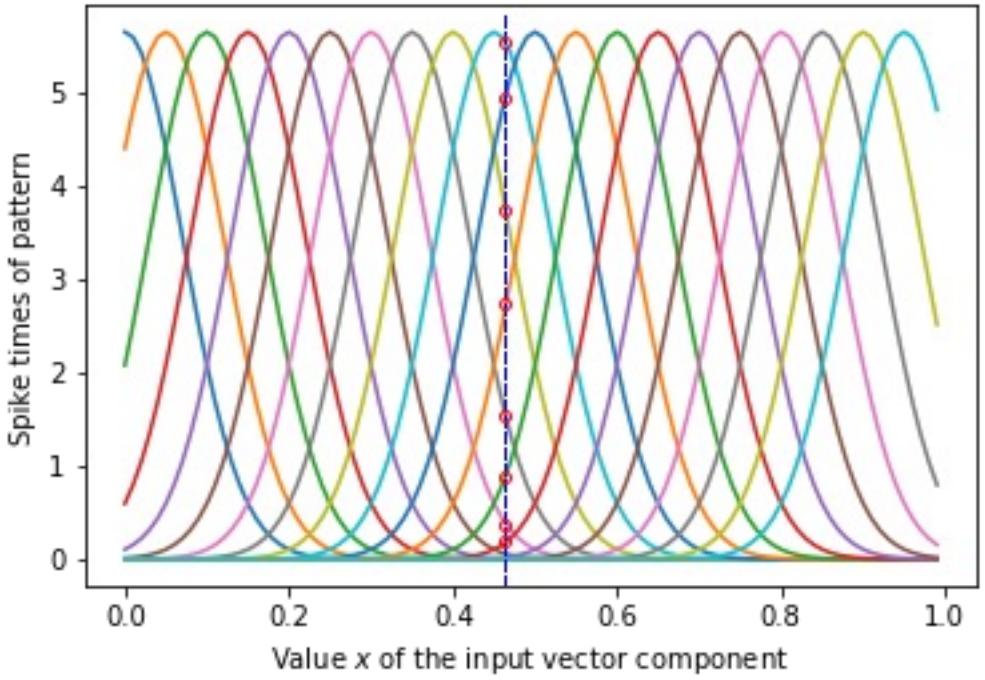


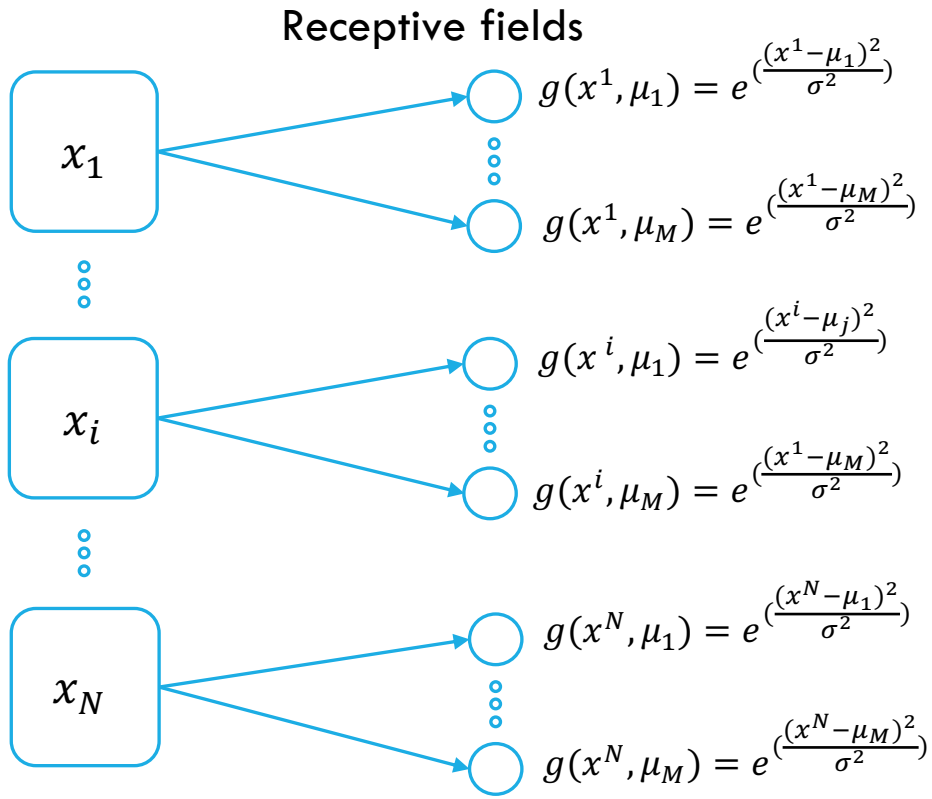


$$\Delta w = \begin{cases} -\alpha\lambda \cdot e^{\left(-\frac{t_{pre}-t_{post}}{\tau_-}\right)}, & if\ t_{pre} - t_{post} > 0 \\ \lambda \cdot e^{\left(\frac{t_{post}-t_{pre}}{\tau_+}\right)}, & if\ t_{pre} - t_{post} < 0 \end{cases}$$

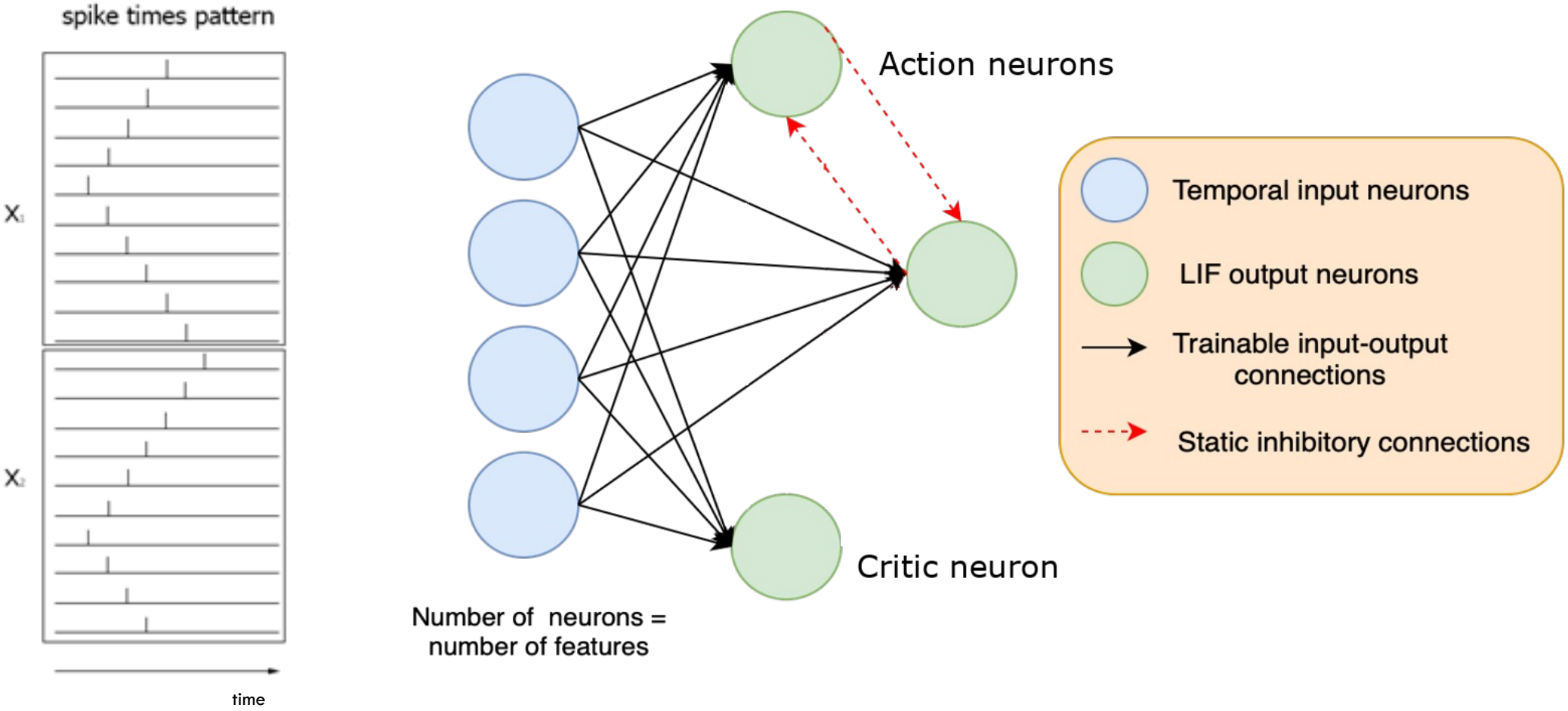
TEMPORAL CODING

Each component of vector x_i transforms into M components $g(x^i, \mu_j) = e^{(\frac{(x^i-\mu_j)^2}{\sigma^2})}$





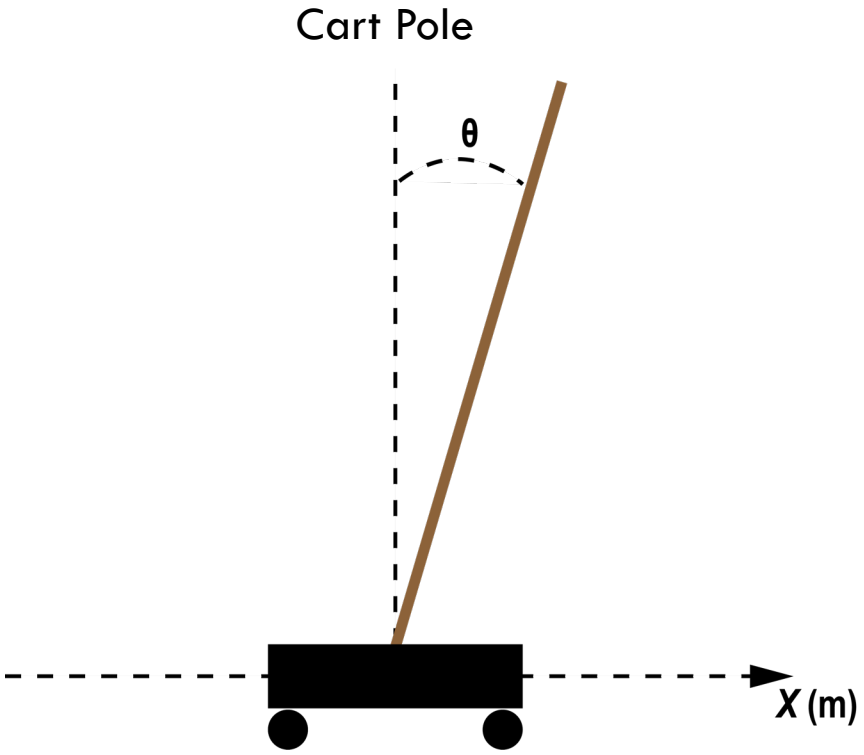
NETWORK TOPOLOGY



ACTOR-CRITIC LEARNING ALGORITHM

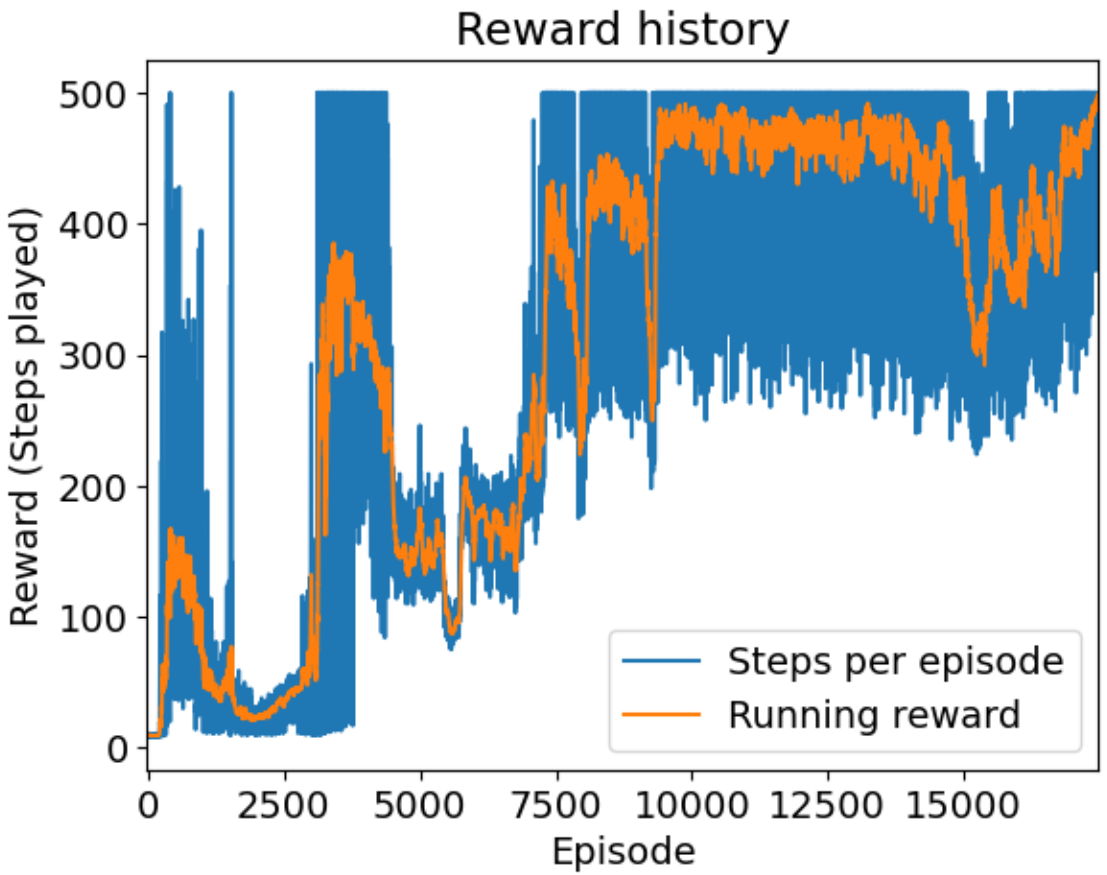
1. Acquire environment state S_t
2. Convert state to spike times
3. Acquire an action a_t and reward r_t using zero synaptic change amplitude $\lambda=0$
4. Calculate K and $\lambda=\lambda_{initial} \cdot \Delta r \cdot \mathrm{K}$ using $\Delta r = r_t - r_{t-1}$
5. Apply the state S_t along with modulating plasticity of critic neuron and neuron predicted action
6. Calculate running reward $R_i = 0.05 \cdot S + 0.95 \cdot R_{i-1}$
7. Go to 1

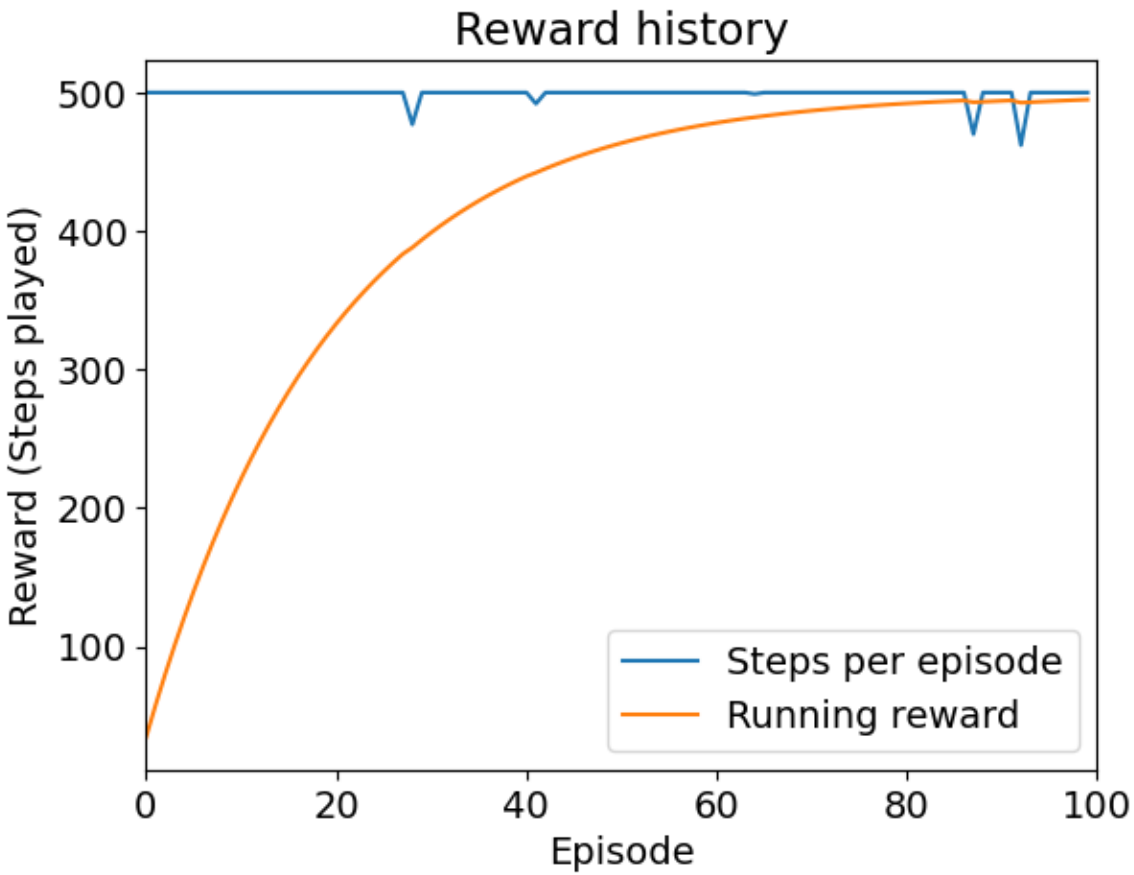
$$K = \begin{cases} 0 \; if \; \max(S_t, R_{i-1}) = S_{max} \\ 0.025 \; if \; \max(S_t, R_{i-1}) > 450 \\ 0.05 \; if \; \max(S_t, R_{i-1}) > 400 \\ 0.075 \; if \; \max(S_t, R_{i-1}) > 350 \\ 0.1 \; if \; \max(S_t, R_{i-1}) > 300 \\ 1 \; if \; \max(S_t, R_{i-1}) \leq 300 \end{cases}$$



The task is considered solved successfully if, during 100 episodes, the cart control algorithm keeps the pole from falling for at least 475 time steps.

RESULTS


Reward history
Reward (Steps played)
500
400
300
200
100
0
0
2500
5000
7500
10000
12500
15000
Episode
Steps per episode
Running reward

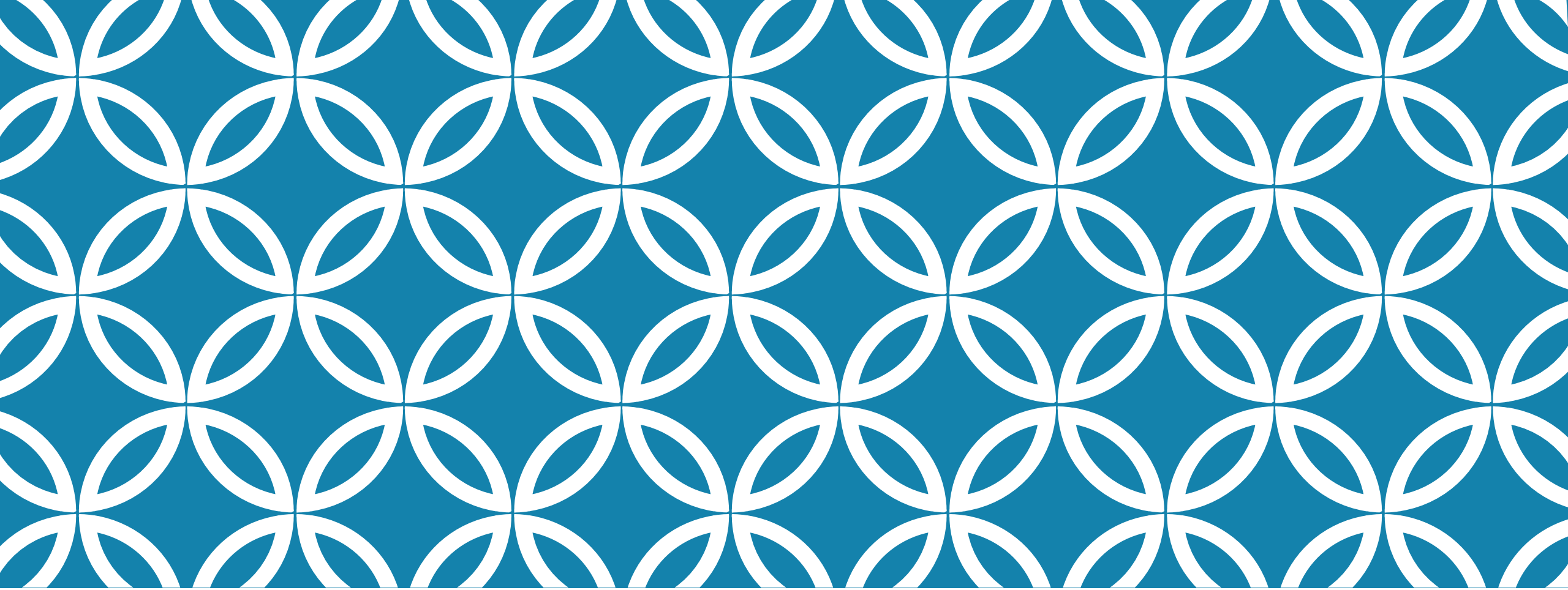

Reward history
Reward (Steps played)
500
400
300
200
100
0
20
40
60
80
100
Episode
Steps per episode
Running reward

CONCLUSION

The proposed learning algorithm modulates synaptic plasticity of action-control and critic neurons based on difference value of the expected reward.

The results demonstrate the success of this type of learning for CartPole task.

The proposed method is based on temporal coding, leading to lower power consumption compared to frequency coding.



THANK YOU FOR ATTENTION