Neuroscience, Learning, Memory, Cognition Course

Mohammad Nourbakhsh marvast

Student id: 401200482

November 2023

1 Generalized Firing-Rate-Based Network Models

We can summarize the activity in a generalised firing-rate-based neural network using two coupled differential equations:

$$\tau_s \frac{dI_s}{dt} = -I_s + w.u$$

$$\tau_r \frac{dv}{dt} = -v + F(I_s(t))$$

Where w and u are matrices representing the synaptic weights and presynaptic neurons' firing rates respectively and F is a non-linear function of the total synaptic input current.

One such famous function used in computational neuroscience is the Sigmoid function defined as:

$$S(x) = \frac{1}{1+e^{-x}}$$

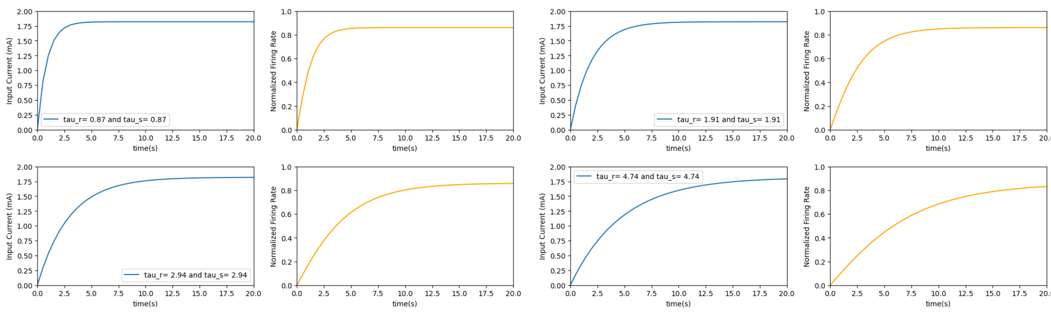
Questions:

1. By manipulating the time constants, Can you make the input reach a steady state much faster than the output? (5 Points)

   ans. Let us consider the ODE $\begin{cases} \dot{X} = \frac{1}{\tau} f(X) \\ X(0) = X_0 \end{cases}$.

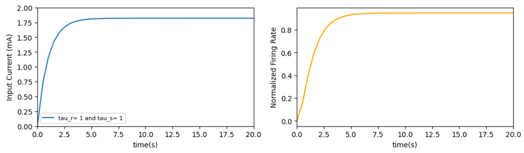
   If $\gamma(t)$ is the integral curve of this ODE, then for any constants a, $\gamma(at)$ is the answer of the ODE $\begin{cases} \dot{X} = \frac{a}{\tau} f(X) \\ X(0) = X_0 \end{cases}$. Thus, the answer is **YES**.

   By decreasing the constants, we can reach an equilibrium state faster. However, the place of this steady state stays the same. Here are the plots for our equations:

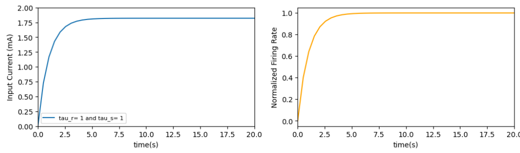


2. Try out the model for other non-linear functions. Do you spot any difference?

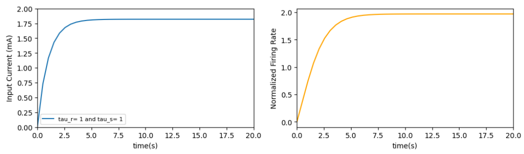
   ans. There are some differences spotted. The range of each activation function used is different. Also, the gradient of firing rates (speed of learning) is different. we can understand that from the figures:



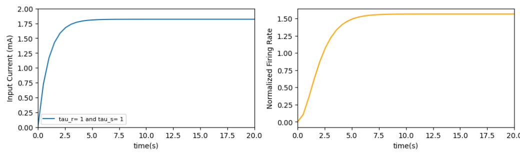
(a) tanh



(b) softmax



(c) Softplus



(d) SiLU

3. Can you compare this model with a perceptron network? Can we classify input using this network as well?

   ans. The perceptron network is a type of neural network that uses a linear activation function and a linear classifier (binary). It is limited to linearly separable problems. Non-linear functions allow networks to learn more complex relationships between their inputs and outputs. This model can also be used to classify inputs, but it requires a different learning algorithm than a perceptron network. One possible algorithm is the gradient descent method, which minimizes the error between the desired and actual outputs by adjusting the weights in the direction of the negative gradient of the error function. This method can handle nonlinear and non-separable patterns, but it may also encounter some problems such as local minima, slow convergence, or overfitting.