(Brest State Technical University 2009 Spring Semester: Course Practice)

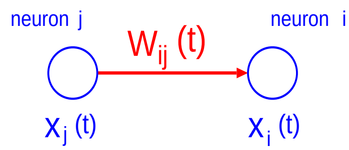
Recurrent Neural Network - II

Akira Imada
(e-mail akira@bstu.by)

This document is still under construction and was lastly modified on

May 12, 2009

1 Model and Method



1.1 Continuous Neuron

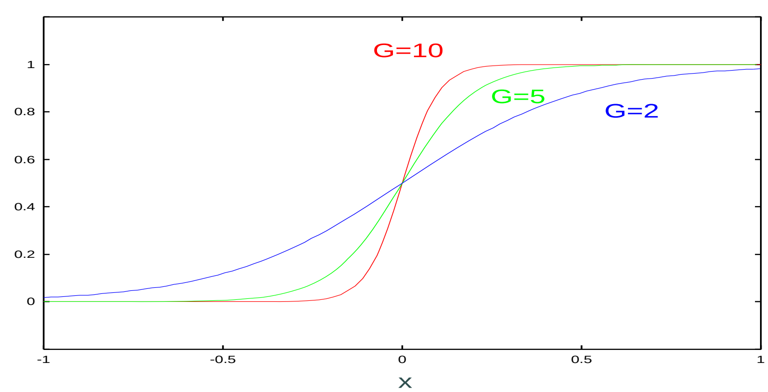
Renew of states

$$x_i(t+1) = f(\sum_{j=1}^{N} w_{ij}(t)x_j(t) + I_i) \qquad (i = 1, 2, 3 \cdots, N)$$

Transfer function – Sigmoid

$$f(x) = \frac{1 + \tanh(Gx)}{2}$$

[1]



Excersize 1 *On your monitor screen, display 100 neuron each of which takes a state of continuous value from 0 to 1 at random. This is the pattern at* $t = 0$*. Then renew states step by step* $(t = 1, 2, 3, \cdots)$ *and display states of 100 neurons at each time step using a grey scale (white - grey - black). This may be like a rectangle Christmas Tree.*

[1] If your program language does not support $\tanh(x)$, like PASCAL, use the following formula: $\tanh(x) = \sinh(x)/\cosh(x)$ where $\sinh(x) = \{\exp(x) - \exp(-x)\}/2$, and $\cosh(x) = \{\exp(x) + \exp(-x)\}/2$.

1.2 A contrller of movile robot

Choose 10 neurons from those 100 neurons and connect them to two output neurons. To be more specific, connect from the 91st neuron to the 95th neuron to output neuron 1 and from 96th neuron to 100th neuron to output neuron 2. Connection weights from those 10 neurons to 2 output neurons should be all positive value from 0 to 1. Then the states of these 2 output neurons are calculated as before:

$$o_1(t) = f(\sum_{j=91}^{95} w_j x_j(t))$$

and

$$o_2(t) = f(\sum_{j=96}^{100} w_j x_j(t)).$$

Then, we can make a robot navigate in a x-y coordinat by $x(t+1) = x(t) + o_1(t)$ and $y(t+1) = y(t) + o_2(t)$ starting from the origin $(0, 0)$.

Excersize 2 *Show the robot movement on your screen.*

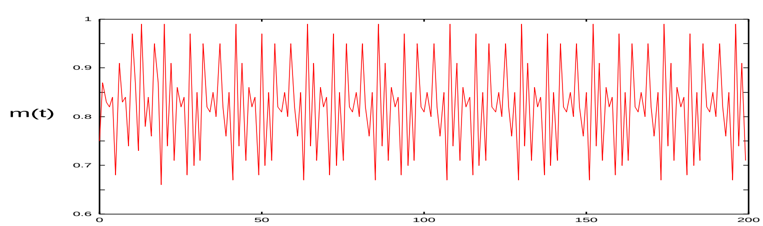
1.3 Dynamics of states of neurons – from atractor to chaos

When we observe the change of state of each of these 100 neurons, you may found some are quiet (always 0), some are stable (taking some value and not changing), and some are always change its value. However, its not so easy to observe those changes, so we take an average of these 100 states.

$$m(t) = \frac{1}{N} \sum_{i=1}^{N} x(t)$$

Excersize 3 *(1) Plot $m(t)$ as a function of t. (2) Also plot in the x-y cordinate as $x = m(t)$ and $y = m(t+1)$ for $t = 1, 2, \cdots$. Observe the result with changeng G in the sigmoid function, for example $G = 5.0, 5.2, 5.4, \cdots 6.9, 7.0$.*

Result of (1) might be something like the followings.



1.4 Hebian Learning

Update of weights

$$w_{ij}(t+1) = w_{ij}(t) + \eta x_i(t+1) x_j(t)$$

(where η is called *learning rate* and set to 0.01 here, for example.)