ERC Report - May 10, 2010

Amir Hesam Salavati
E-mail hesam.salavati@epfl.ch

Supervisor: Prof. Amin Shokrollahi
E-mail amin.shokrollahi@epfl.ch
Algorithmics Laboratory (ALGO)
Ecole Polytechnique Federale de Lausanne (EPFL)

May 10, 2010

1 Introduction

In this report, I will briefly explain the outcome of my readings about application of coding theory in neuronal systems. A great deal of what I read concerns networks of neuron. These networks are mainly responsible for learning, pattern recognition and memory in the brain. One class of these networks is particularly interesting as it governs associative memory, the part of brain responsible for identifying objects that are resemble those observed before. For instance, when you read a misspelled word, associative memory identifies the correct word based on this "erroneous" input.

Furthermore, I read some classical works on the applications of information theory in analyzing neural coding. In addition, I have studied some recent works on neural error correction methods. While it is not completely the same error correction issue that we have in mind, it seems promising that research on this topic has already started.

2 Neuronal Networks

As mentioned in the previous report, neurons are connected to each other via synapses. A group of neurons that are inter-connected constitute a network. There are two important factors that affect the properties of such networks: the number of synapses between the neurons and the weights of these synapses. By appropriately adjusting these two parameters, one can get different networks performing various tasks.

In cortex and other areas of the brain, one finds three major classes of interconnections:

- Feed-forward: that is a connection from an earlier processing stage to a later one.
- Top-down: which is the reverse of the feed-forward connection.
- Recurrent synapses: that is an interconnection within the same processing stage.

2.1 Network Model

To model a network of neurons, we would like to determine the output firing rate of the neuron according to the firing rate of its pre-synaptic neurons. To do so, we first model the effects of the input firing rates (u) of the pre-synaptic neurons on the electrical current to somma (I_s) and then derive the output firing rate (v) as a function of I_s [1].

In order to capture the effect of the input firing rates on I_s, suppose we have a neuron with N_u input (pre-synaptic) neurons. The firing rate of these neurons is denoted by the row vector u. Moreover, since all synapses have weights, we denote the weight of synapses by the vector w. Figure 1 illustrates these notations [1].

Now if a spike from a pre-synaptic neuron arrives at time t_i, it contributes to I_s with $w_b * K_s(t)$, where K_s is the synaptic kernel (for simplicity, we consider the same kernel for all synapses). Assuming that the spikes at a single synapse are independent, the total contribution of the pre-synaptic neuron to I_s is [1]:

$$w_b \sum_{t_i<t} K_s(t-t_i) = w_b \int_{-\infty}^{t} K_s t - \tau \rho_b(\tau) d\tau \tag{1}$$

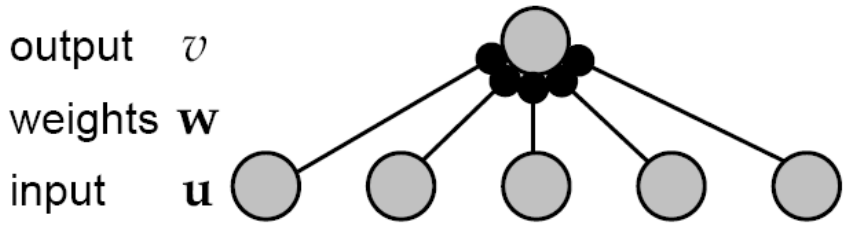

Figure 1: Network model [1]

where

$$\rho(\tau) = \sum_i \delta(\tau - t_i) \tag{2}$$

is neural response function of neuron b.

Now since we are interested in doing large-scale simulations and since each neuron has many input neurons, we can estimate the neural response function of neuron b (ρ_b) with its firing rate, $u_b(t)$, which gives us:

$$I_s = \sum_{b=1}^{N_u} w_b \int_{-\infty}^{t} K_s t - \tau \rho_b(\tau) d\tau \tag{3}$$

We can rewrite the above equation in differential equations format as below:

$$\tau_s \frac{dI_s}{dt} = -I_s + \sum_{b=1}^{N_u} w_b u_b = -I_s + w.u \tag{4}$$

So far, we have derived I_s. Now we need to estimate the output firing rate, v, as a function of I_s, i.e. $v = F(I_s)$. Different choices for $F(.)$ are sigmoid function or a linear-threshold function $F(I_s) = [I_s - \gamma]_+$ where γ is the firing threshold.

Hence, the output of the network is governed by the dot product of weight vector w and the input firing rates, u. In fact, this is a crucial property of neuronal networks since by adjusting w, they can perform different dynamical behaviors which help them to accomplish different tasks including memorizing, learning and pattern recognition. As we will discuss in the next

section, a special class of these networks helps brain to perform some sort of classification and error correction.

2.2 Plasticity and Learning

In all types of learnings, we have a network of neuron whose task is to adjust its weights such that an appropriate relationship is constructed between the input and output of the network. Here, weights refer to the weight of synapses in the network (see figure 1). There are two types of learning:

1. Supervised: in supervised learning, we give both input and output to the network of neurons. Hence, the network knows the answer in advance. All the network has to do is to choose its synapse weights such that the probability of correct classification is maximized.

2. Unsupervised: in this case, we do not give the answer to the network and it has to figure out the input/output relationship all by itself.

In supervised learning, the goal is not only to "remember" the previously "seen" patterns in the training period, but also be able to classify the new patterns that are close to the previous ones. This task is accomplished by appropriately adjusting network weights. This could be done in an iterative manner. In other words, we estimate weights and produce an output (firing rate) based on this estimation. Since we have the actual output, we compare the estimated output with the actual one and refine our weights according to the comparison results. We repeat this process until satisfactory approximation of the output is found out.

In [1] some weight evolution methods are proposed such that the network adjust its weights to correct estimation errors. Since neuronal networks resemble the graph of iterative codes, coding theory may be able to propose better updating rules that are in agreement with experimental results.

2.3 Associative Memory

In conventional memories, like RAM, recall is based on address. However, in an associative memory, recall is based on content [1]. Associative memory networks have been suggested as models of various parts of the mammalian brain in which there is substantial recurrent feedback.

During a recall process, an associative memory performs a pattern matching procedure. During this process, the part of the memory that is the closest match to distorted activity pattern is identified. This is very similar to decoding an erroneous data block in which the closest codeword is found and returned as the output.

Furthermore, an associative memory performs this task in iterative manner, i.e. it starts from one of the memory patterns and approaches a fixed point which is the closest match to the input pattern.

The basic conclusions from studies of associative memory models with threshold linear or saturating units is that large networks can store even larger numbers of patterns, particularly if the patterns are sparse and if a few errors in recall can be tolerated [1].

2.4 Representational Learning

Brain is a very good processor. Consider tons of data delivered by the visual system for example: there are 10^8 photoreceptors that bombard brain with data. Yet, brain is not only able to select useful stuff out of this bulk of data, but also do it in real time. The key trick here is that the way data is represented vary in different stages of visual data processing system, from retina to cortex. It becomes more high level. The power of brain comes in extracting "features" of natural stimuli and focusing on them.

Representational Learning is a kind of unsupervised learning in which we would like to analyze data in terms of its underlying features. The output is our estimate of the features. Since the process is unsupervised, the goal is to adjust network weights such that the "statistics" of the produced output matches closes those of the input.

This approach could be useful in decoding algorithms. For instance, we could provide the decoder with a training period during which it receives noisy codewords. After a while, received data start clustering around valid codewords. Using this clustering effect, we can determine the network structure that extract this feature and help in decoding. Similar examples in human brain could be found in [1].

3 Neural Error Correction

In [2] and [3] authors have addressed the issue of neural error correction. The effects of spike timing precision on error correction process are also investigated. In their approach, two different spike trains, one normal and one with jitter in spike timings, are fed into a neuron as input and the output spike train of the neuron is observed. Then, the outputs are compared for cases when a single spike is missed in the input. The results show that the effects of an error depends greatly on the ongoing dynamical behavior of the neuron: in case of phase locking a high degree of presynaptic spike timing precision can provide significantly faster error recovery. In contrast, for other cases, isolated missed spikes do not have a big effect on the neuron's output.

In brief, the method used in [2] and [3] is as follows: first, we generate a high-precision spike train H in which the inter-arrival times between spikes are precisely determined and are equal to each other. Then, an erroneous version of this spike train, denoted by H_e, is produced by randomly deleting one spike (or more, but the missed spikes are separated at least 15s from each other so that neuron reaches its stationary behavior before next error). This kind of error is equivalent to complete failure of the synapse to transmit a spike. Then H and H_e are fed to the neuron and the resulting output spike trains are compared. The same process is repeated for a low-precision spike train, L, in which the inter-arrival time of spikes are roughly the same, i.e. we get H and add a random jitter (according to a uniform distribution in the interval [-epsilon,epsilon]) to the arrival time of each spike (see the figure below). Again, an erroneous version L_e is also produced and the results are compared.

To compare the results, the time interval between each output spike in the erroneous case and the most recent one in the normal case was measure (showed by $\varphi_{j,g}^{H/L}$, for high precision and low precision respectively, where j is the index for the missing spike and g is the index of post-synaptic spikes with respect to this spike in a time frame of c spikes (c is around 40 spikes after the occurrence of error). Figure 2 illustrates the notations. The faster $\varphi_{j,g}^{H/L}$ decreases (as g is increased meaning that time has elapsed), the faster the error recovery would be.

Figure 3 depicts an example result from [2] in which the perturbation graph for both high precision and low precision sequences of a phase locking neuron is shown.

Similar results in [2] and [3] show that even small amounts of jitter can

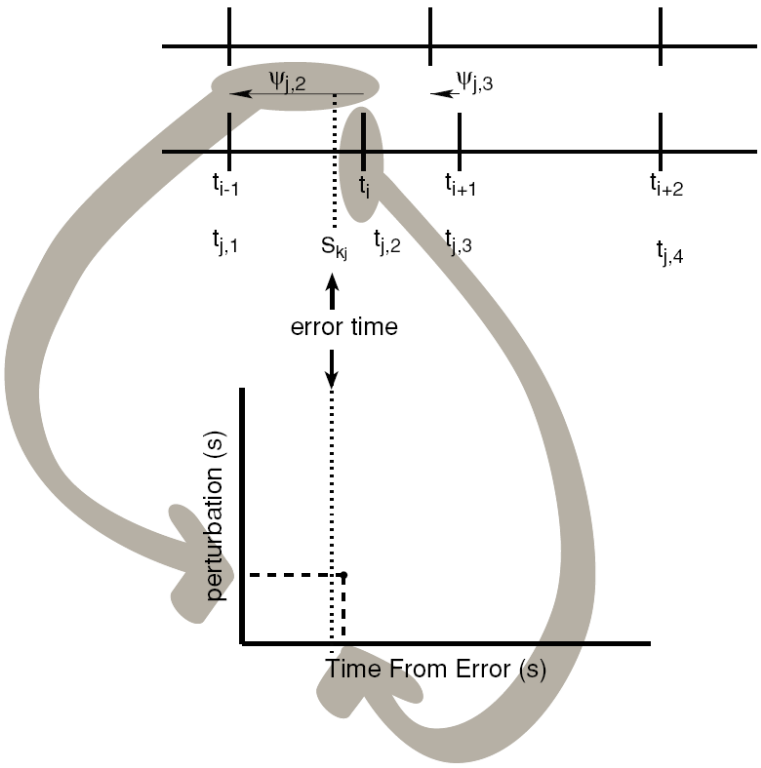

Figure 2: Perturbation Graph [2]

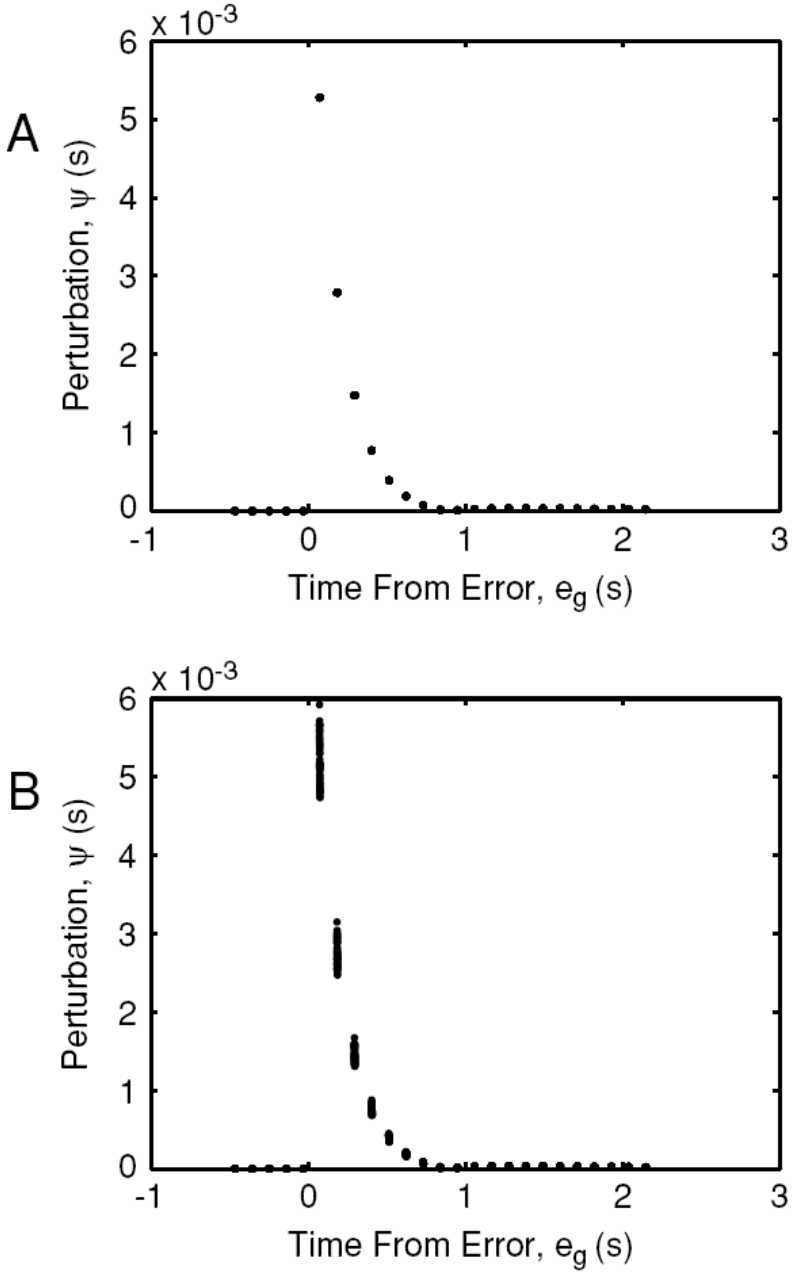

Figure 3: Error recovery of a phase locking neuron [2]

have a significant effect on error recovery.

While their work seems promising, the term "error correction" is quite different form what is used in coding theory. In coding theory, error correction refers to the procedure that return the correct code such that the output of the system is unaffected (to an extent which the code allows) even if the input is erroneous. However, in [2] and [3], error correction refers to error recovery. In other words, the authors have investigated the time it takes after occurring of an error for the neuron to resume its normal behavior.

Nevertheless, the authors argue that the fact that a neuron may produce differing discharges in response to the same stimulus at different times is often given as evidence that neural coding is unreliable. However, the neurons nonlinear dynamical nature means that such responses do not necessarily indicate that a neurons computation is not sensitive to the precise timing of input spikes [3].

Moreover, since neuron is non-linear system, authors believe that Shannons information theory is conceptually incompatible with neural information processing. Therefore, it gives lower bounds on matters such as spike timing precision. However, I think if we can write the state of the neuron as a function of input as well, then we can model neurons as convolutional encoders.

4 Information Theory and Neural Coding

In [4], authors review the applications of information theory in neural coding. Researcher generally use information theory to investigate neural coding in three ways:

1. They estimate the capacity of the neural channel as a function of firing rate and compare it to the actual amount of information transmitted to obtain a measure of coding efficiency.

2. They directly measure the actual amount of information being transmitted in the neural channel without any assumption on a particular parameter of the stimulus being encoded. Then, the results are compared with those of a specific stimulus-response model. The comparison could help us in quantitatively evaluating the quality of the model.

3. They have used information theory to determine the limiting precision

of spike timing and that is the minimum time scale beyond which spike timings convey a meaning and information.

Their overview shows that linear models, in which neurons encode stimuli linearly, are essentially good and capture much of the transmitted information. Moreover, as discussed before, each spike carries information. Information-theoretic calculations also show that certain neurons use precise temporal (millisecond) spiking patterns in encoding. Precise spike timing had previously been identified in the auditory system, where it is important for sound localization and echolocation, and also more recently elsewhere in the CNS [1], [4].

A very interesting phenomenon mentioned in [4] is the encoding method used by mock neurons to represent the intensity of a stimulus. The neuron responds by generating a firing rate according to a Gaussian distribution whose mean varies from 20Hz for weak stimulus to 80Hz for strong ones. This is just like that the neuron uses firing "levels" (just like voltage levels) and then these levels are susceptible to additive Gaussian noise. Moreover, since different levels are used, spikes are used to encode them in the binary format and this binary sequence is transmitted over the nerve. In other words, it is like that neuron uses some "levels" as the output and uses spikes just to encode them in the binary format and the other end should decode this sequence by using optimal thresholds such that the error probability is minimized

4.1 Estimating Neuron's Channel Capacity

In [4], a simple method is proposed for estimating the capacity of a neuron. To estimate the capacity of a neuron, a constant stimulus is applied to the neuron a couple of times. The resulting response is recorded each time and then average to obtain the average response (everything is done in frequency domain). This average response is subtracted from each response to derive the noise in each trial. The power spectrum of the noise instances are computed and then average to obtain the average noise power spectrum. Having the average response and the average noise power spectrum, we can calculate the estimated SNR as a function of frequency. This SNR is used to determine the channel capacity of a neuron (see figure 4).

Furthermore, this method could be used to determine the limiting precision of spike timings, that is the minimum precision beyond which spike

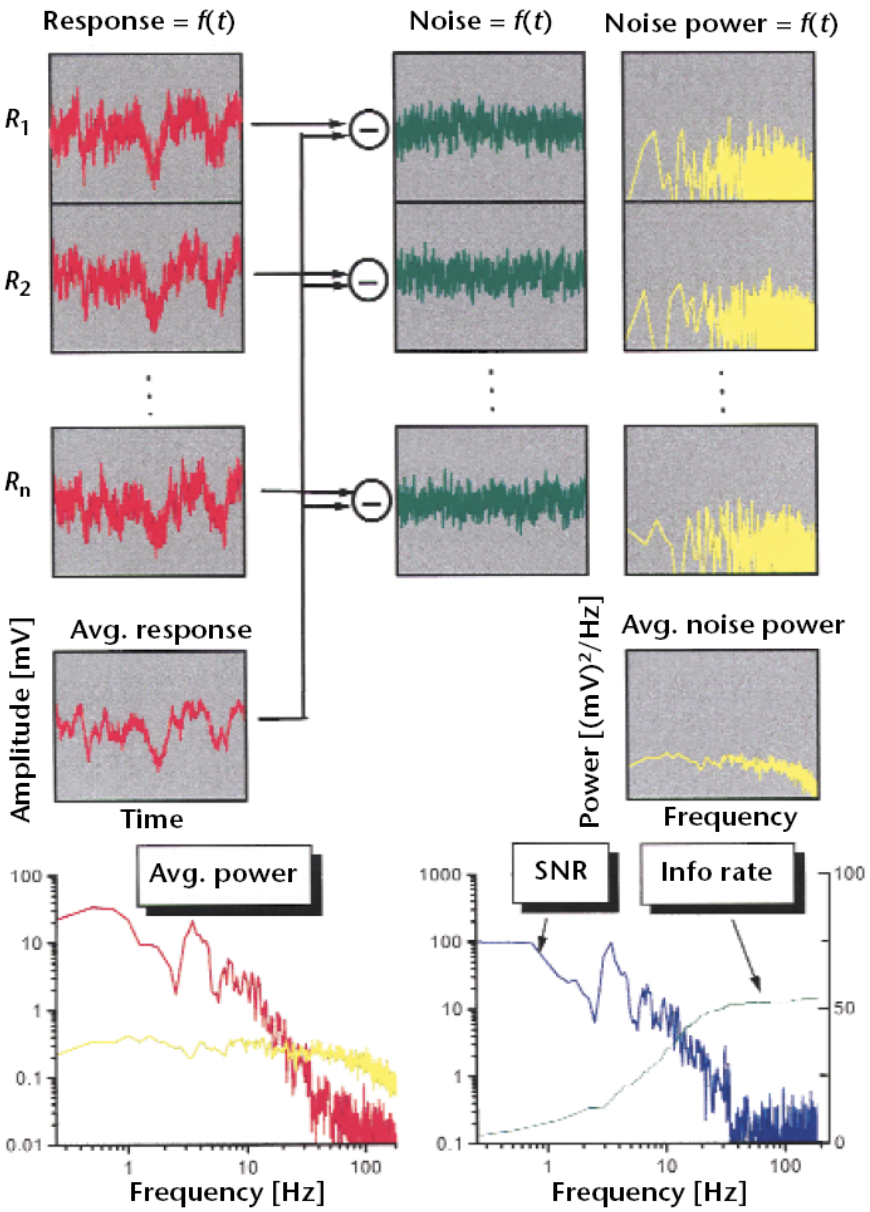

Figure 4: Estimating the capacity of a neuron [4]

timings do not convey information. The limiting timing precision could be determined according the mutual information vs. frequency graph. Usually, we have a low or bandpass behavior in which information rate goes to zero beyond a cutoff frequency (see figure 4). This means that the temporal changes beyond the precision dictated by this cutoff frequency is not encoded by the neuron. The cutoff frequency gives us the minimum timing precision of the neuron. In other words, The information should increase as the window size is made smaller until it plateaus. This approach revealed spike timing resolutions of roughly one millisecond [4].

Moreover, high information rates (per spike) could indicate the importance of single spikes and hence the existence of temporal codes. These results highlight the importance of each spike to the neural code. However, an example of temporal encoding for dynamic stimuli has not yet been found. On the other hand, temporal encoding for stimuli with very slow dynamics (usually presented as static stimuli) has been shown both in single neurons [21] and in neuronal ensembles (see [4] and references therein).

References

[1] P. Dayan, L.F. Abbott, "Theoretical Neuroscience: Computational and Mathematical Modeling of Neural Systems", MIT Press, 2004.

[2] M. Stiber, "Spike Timing Precision and Neural Error Correction: Local Behavior", Neural Computation, Vol. 17, No. 7, 2005, Pages 1577-1601.

[3] M. Stiber, T. Holderman, "Global Behavior of Neural Error Correction", Proc. IEEE International Joint Conference on Neural Networks, 2004.

[4] A. Borst, F.E. Theunissen, "Information Theory and Neural Coding", Nature neuroscience 1999, Vol 2, No 11, pp. 947-957.