Computational Neuroscience
Dr. Sharba Bandyopadhyay
Department of Electronics and Electrical Communication Engineering
Indian Institute of Technology Kharagpur
Week – 10
Lecture – 48

Lecture 48 : Hebbian Plasticity

Hello, welcome. So, we have discussed spike timing dependent plasticity. And so now we will be discussing the general approaches that have been used in terms of computational modeling involving plasticity. So first of all we will allude or rather go back once more to our general integrated and fire model just to remind ourselves. And so if you recollect we had been we had this idea that you have a capacitor and then for the leakage we have G_{leak} and then E_{leak} and we have an $I_{external}$ that can drive this. So, in this model we had removed the sodium and potassium channels that were present in the Hodgkin-Huxley.

And also if in other models where we have other kinds of channels those are removed per say for now to make the most simple possible kind of neural model. So, this is our E_{leak} and we have the capacitance. And so in this case what we know is that if we drive this system with $I_{external}$ of a certain shape or certain profile. So, this is time and this is current $I_{external}$ let us say $I_{external}$ is 0 here initially and then there is some positive current injected.

And so accordingly over here we will model the voltage across the capacitor and that will integrate and reach threshold that is pre decided. And we assign an action potential once it reaches threshold and we reset the voltage below and we get an action potential whenever there is a threshold crossing. And that this leaky integrate and fire model is very useful when we are trying to model activity mainly rate based activity that to explain behavior of an entire network or some kind of learning and so on. And when precise spike timing is not important in terms of the question then these kind of models are very useful. And in fact, this can be further reduced to an even simpler model where we say that we have one output neuron whose activity or firing rate is given by V and we have a set of input neurons.

Let us say this is u_1, u_2 up to u_{nb} and there are synaptic inputs on to the output neuron and each of these have weights w_1, w_2 and w_{nb}. Basically what we have is the input activity is represented by u_1 up to u_{nb} and hence we can represent it with a vector u and that is the rate responses or the rate of inputs of the presynaptic neuron. And the post synaptic neuron V takes these inputs basically we have a weighted. So, w is now a weight that is V a vector which is

made up of the elements w_1, w_2 up to w_{nb} where nb is the number of inputs. And what we are representing by this w is the synaptic strength and each of the input rates are modified by this synaptic strength and a proportionate some activity is being added on to the V output.

That is we will say that our V is changing. Let us say we have a dV/dt that is the output V is changing with the time constant τ and that is this and we have input $w \cdot u$. So, sorry this is our V and this is a plus $w \cdot u$. So, the V is relaxing to go towards steady state value that dV/dt is 0 with the time constant τ and is being modified by this $w \cdot u$. So, I mean simplification or conversion of the integrate and fire, integrate and fire neuron can with certain assumptions can be shown to be of this form when we consider rates as inputs and not spike timings or spike trains as inputs.

So, V what we are saying is that the output potential is always trying to go towards the steady state value if where dV/dt is 0 and basically a 0 steady state value when there are no inputs. And when there are inputs it is its V is increased by the rate of the inputs. And now this time constant of relaxation is extremely small it is very or rather we will say that it is reaching there quite fast I am sorry. So, that our dV/dt is quickly reaching 0 and in that case our V can be said to be simply the dot product of w times u. So, if we have this as the model then the now we consider what we have learnt from our Hebbian plasticity that is now V equals w times dot product with u.

What our Hebbian plasticity suggest is that if the weights increasing that is our dw/dt is proportional to the correlation of z and u. So, $z \times u$ and there is a time constant based on which I mean that is the dynamics of the w how it reaches the changed value. And this is happening this τ_w is a much larger than how the patterns of input are being presented. So, the weight is changing slowly. So, w the weight is slow changing and there are a number of inputs that are coming in by the time w is changing.

So, the overall $\tau_w dw/dt$ then would be represented by an average of all the patterns of input that is being provided. So, time average or the based on the all the inputs. So, with this as our learning rule that is we have the average of v times the u driving our weight change or the larger the correlation or larger the product value of v and u larger is the increase in w. So, what are the sort of consequences of such a learning rule? So, this is what the basic Hebb rule is, but as we will see soon that there are a number of issues, number of problems with this overall idea and we will need to modify this gradually to other forms which may or may not be biologically based. And actually the course of development of implementation of plasticity in fact, went out of hand just because it became more

like a mathematical treatment rather than treatment based on the biology.

So, anyway let us see how this progress went on. So, if we now consider that the w weight how is it changing. So, let us say we consider $\tau_w dw/dt$ what is the average of vu? Average of vu is we replace our v by $w \cdot u$ and then u and average of that. So, essentially what we have here is here we have a vector w. So, let us consider just one element of w that is dw/dt and that will be the average of.

So, when we are talking of this $w_b dt$ this is a dot product. So, it is a scalar. So, here we only consider u_b that is one particular input. And so, here what we have this is u_b times summation $w_i u_i$, i equals 1 to n_u and we are averaging this. So, now as you can see when we average this we will end up averaging the correlations in u within the input that is $u'_b u_b$ or $u'_i u_i u_b$.

So, let me not write it in this manner here with two averages set the summation let us say we make it b' equals 1 to n_u and we have $w_{b'} u_{b'}$ this sum and we have only u_b here and the average of that. So, obviously, here what we are seeing is w_b. So, each of these terms the summation can come out and we will have summation b' going from 1 to n_u and inside we have $w_{b'}$ and average of $u_{b'} u_b$. So, this $u_{b'}$ and u_b sorry. So, basically with this w outside $w_{b'}$ outside and its sum you can easily show that this dw/dt or rather the whole now let us write the whole thing dw/dt will be the input correlation matrix Q times w.

So, what we have in Q are the elements. So, let us write it here. So, what do we have in Q. Q has basically at its n_b by n_b square matrix and we are multiplying this by w_b. So, we have $w_1 w_2$ up to w_{nb} and here we have average of $u_1 u_1$ up to.

So, we have. So, basically each of these terms are average of the product of the input rates or input rates in each of the input neurons and so this is the correlation matrix. So, and so this is also called the correlation based learning rule and now if we want to analyze this further you can you can easily see here that the dw/dt that we have if we constrain the rates to be positive our increase in weights is always positive and it keeps on increasing throughout and there is no way for even implementing long term depression in here. So, this is a unidirectional change in weights. So, given v and u are positive it is simply not possible to have any kind of decreasing weights and all of them will keep on increasing. So, in in general this would fail and the and so essentially it is sort of a runaway kind of module where the weights keep on increasing and so ultimately what happens is that it is an unstable situation that is created.

So, in order to modify that in order to so we do not have L T D in the correlation based learning rule not model cost learning rule. So, what is introduced in there is dw/dt that is how the rates are changing with time again with the same idea we have either v times u minus θu and average of that or we can have τ_w as

dw/dt equals average of so this θu is a vector v minus θv times u and average of that. So, what we mean here is that we are introducing a threshold threshold in the sense that when the activity v is above greater than θv in that case we have potentiation. So, I am talking of the second form which is this one and when the output is lowered below the threshold we get depression. So, we I mean with this we are getting the I mean the possibility of potentiation and depression and similarly in the other case which is the first case again we are setting a threshold θu for each and every input that is those are the elements of the vector θu.

And so we are getting we would get potentiation or depression based on the input activity at each of the synapses. We can have both of them it it gets the mathematical treatment gets a little complicated when we have θ u and θ and θv both of them in the system ah, but our ah treatment here that we will do is ah following ah the one one of the two here which is τ_w let us say dw/dt equals the average of v and u minus θu and this is a vector. So now, what is our average of v ah times u minus θu. So, again if we write ah so essentially what we are having here is our v is going to be ah vu ah is our output ah I am sorry $w \cdot u$ is our output. So, we write $w \cdot u$ times u minus θu.

So, here if we represent our θu ah if we choose our θu to be the average of u ah then what we get is u minus average of u multiplied by $w \cdot u$. So, our v has to be not v times u, but v has to be ah w times u, but w times u minus θ u ah because we have ah done our thresholding. So, I am sorry about this part let us say our w times u minus θu ah. So, essentially if we choose ah so here also we change this u to u minus average of u. So, this turns out to be if if we take the product of these two and average them the w goes out and what we are left with is our covariance matrix.

Let us say we name it C because it is the average of u minus the average of u and u minus the average of u which is the definition of the covariance matrix. And so our learning rule then ah becomes a covariance learning rule that is $\tau_w dw/dt$ is equal to our C ah dot w. So, now that we have derived this again although we get L T B and L T D both of them in these two learning rules the problem again is that the dw/dt is always positive and that is may that makes this stable. So, the increase in I mean the if we take the why why would we say that that is because if we take this τ is equal to dw/dt equals C times u C times w. Let us consider how the norm of the weight changes with time what we will get is it is simply a w $2w$ times dw/dt which is $2w$ times our Cw and so that again brings that the norm is always positive.

So, the overall even though we may get decreases in one of the one or other of the w the overall norm keeps on being positive because we get a w^2 multi-

plying the covariance matrix. So, in this in these both these models which are Hebbian and either correlation based or covariance based we have a problem the main problem being that it is unstable and also there is no competition among the synapses which is something that is observed in actual neurons that is when some synapses strengthen other synapses weaken to maintain sort of level of activity. And so without the competition again this is there is a blow up of this kind of you know output activity. So, in order to compensate for these kind of problems there are various methodologies that have been adopted. And in the next class we will be discussing continuing our discussion on implementation of long term potentiation and long term depression starting with these kind of simple just multiple input single output cases which can later be generalized to more number of neurons. Thank you.