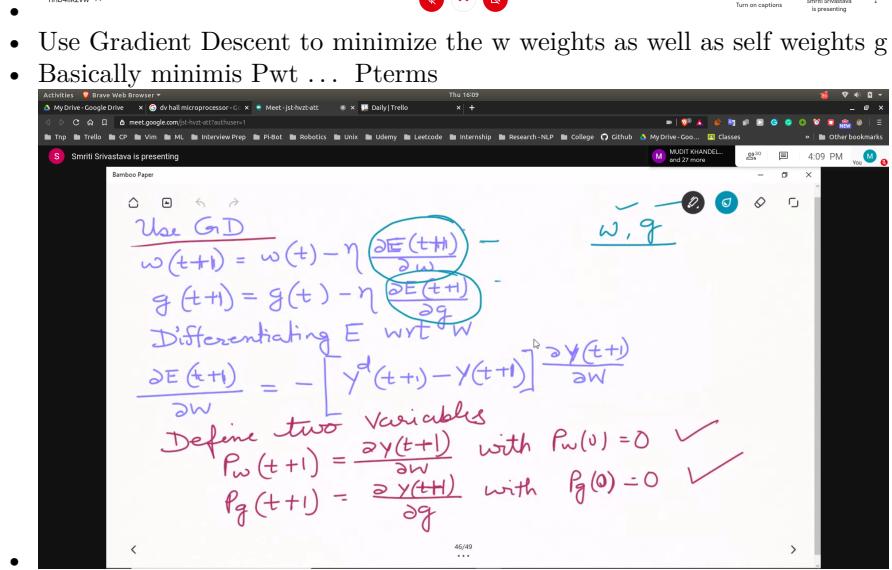
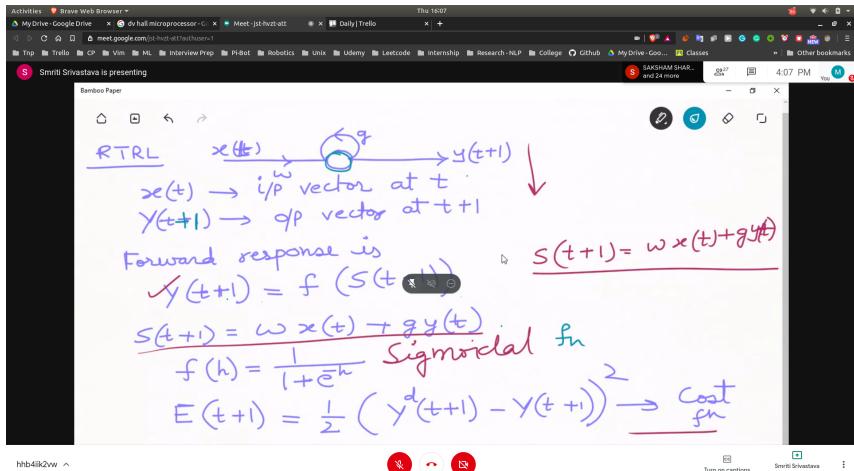


RT RL



- Now we derive the recursive relation

Activities Brave Web Browser Thu 16:10

My Drive - Google Drive x du hall microprocessor - G Meet jst-hw2-att Daily | Trello x

meet.google.com/jst-hw2-att Other bookmarks

ASHISH YADAV and 29 more

4:10 PM

Bamboo Paper

objective : Derive a recursive relation for $P_w(t+1)$ & $P_g(t+1)$ in terms of $P_w(t)$ & $P_g(t)$

$y(t+1) = \frac{1}{1+\epsilon^{P(t+1)}}$

Find $\frac{\partial y(t+1)}{\partial w} = \frac{\partial y(t+1)}{\partial S(t+1)} \times \frac{\partial S(t+1)}{\partial w}$

$$\frac{\partial y(t+1)}{\partial S(t+1)} = y(t+1)(1-y(t+1))$$

$$\frac{\partial S(t+1)}{\partial w} = \frac{\partial}{\partial w} [\omega x(t) + g y(t)]$$

$$= x(t) + g \frac{\partial y(t)}{\partial w} = x(t) + g P_w(t)$$

$$\therefore \frac{\partial y(t+1)}{\partial w} = y(t+1)(1-y(t+1)) [x(t) + g P_w(t)]$$

47/49

Activities Brave Web Browser Thu 16:13

My Drive - Google Drive x du hall microprocessor - G Meet jst-hw2-att Daily | Trello x

meet.google.com/jst-hw2-att Other bookmarks

HARSH and 29 more

4:13 PM

Bamboo Paper

$\frac{\partial y(t+1)}{\partial g} = y(t+1)(1-y(t+1)) [y(t) + g P_g(t)]$

which implies $\frac{\partial E(t+1)}{\partial w} = -[Y^d(t+1) - Y(t+1)]$

$$y(t+1)(1-y(t+1)) [x(t) + g P_w(t)]$$

$$\frac{\partial E(t+1)}{\partial g} = -[Y^d(t+1) - Y(t+1)] Y(t+1)$$

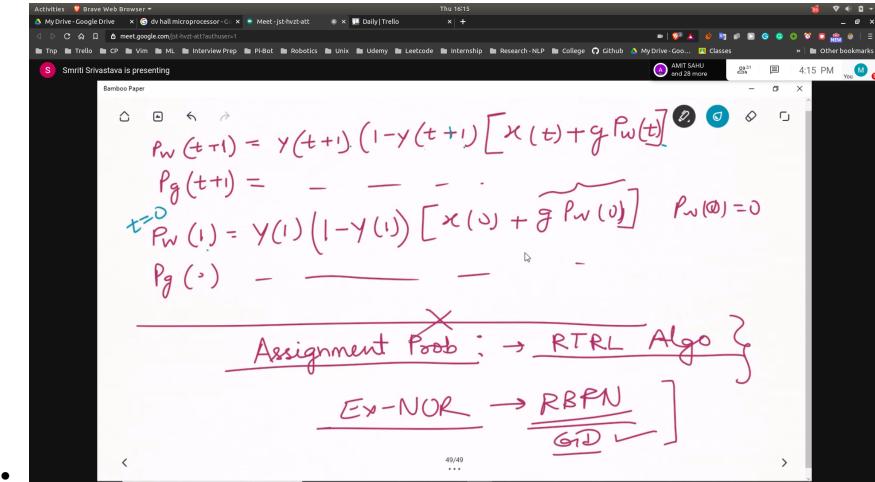
$$(1-y(t+1)) [y(t) + g P_g(t)]$$

$$\omega(t+1) = \omega(t) + \eta [Y^d(t+1) - Y(t+1)] P_w(t+1)$$

$$g(t+1) = g(t) + \eta [Y^d(t+1) - Y(t+1)] P_g(t+1)$$

48/49

- Finally we get $w(t+1)$ and $g(t+1)$

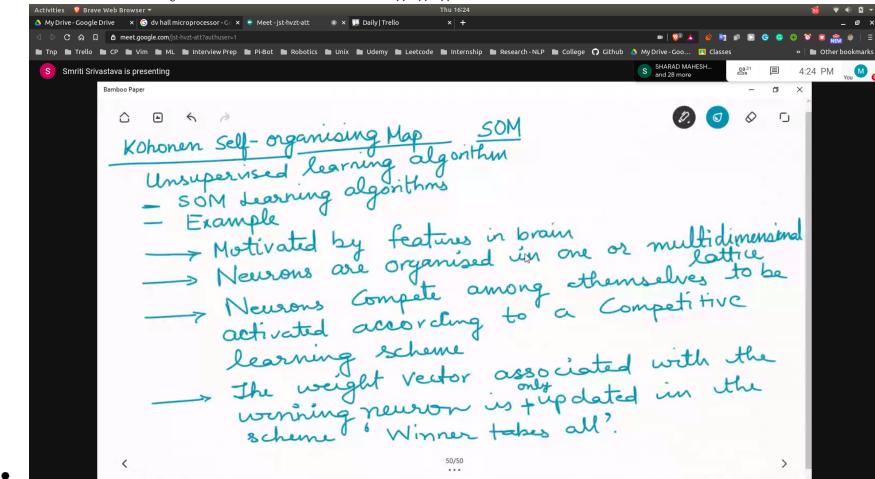


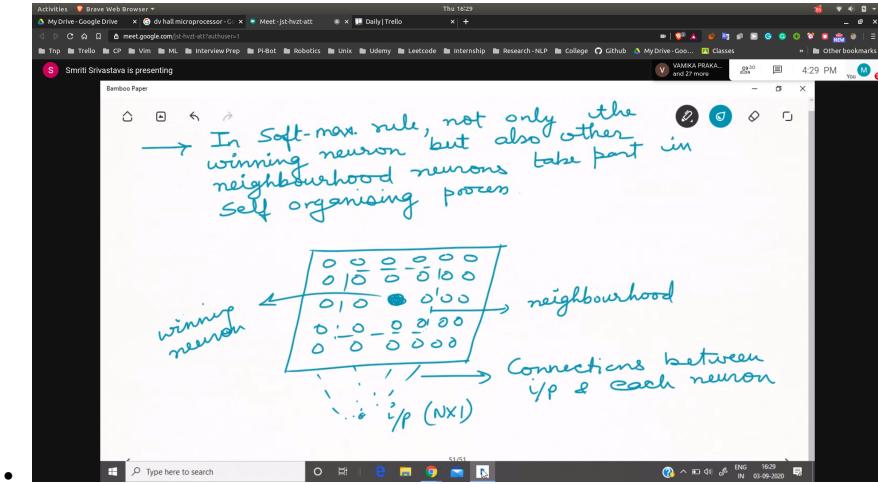
Assignment PROB → RTRL Algo

- We have to apply RTRL to a network

Kohonen Self-organising MAP (SOM)

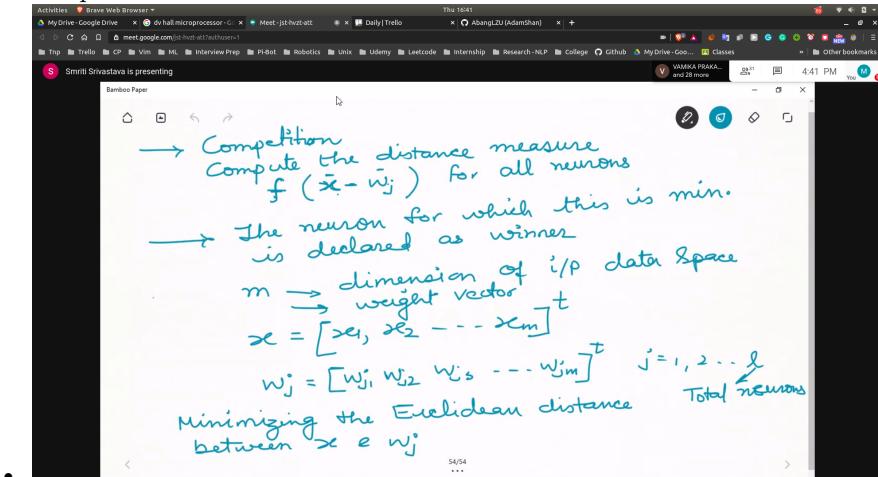
- unsupervised learning algorithm
- Motivated by features in brain #### Characteristics



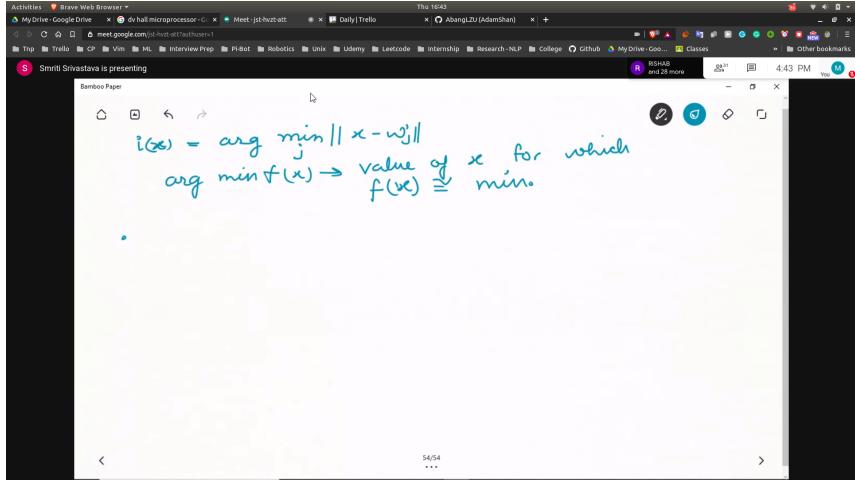


Steps

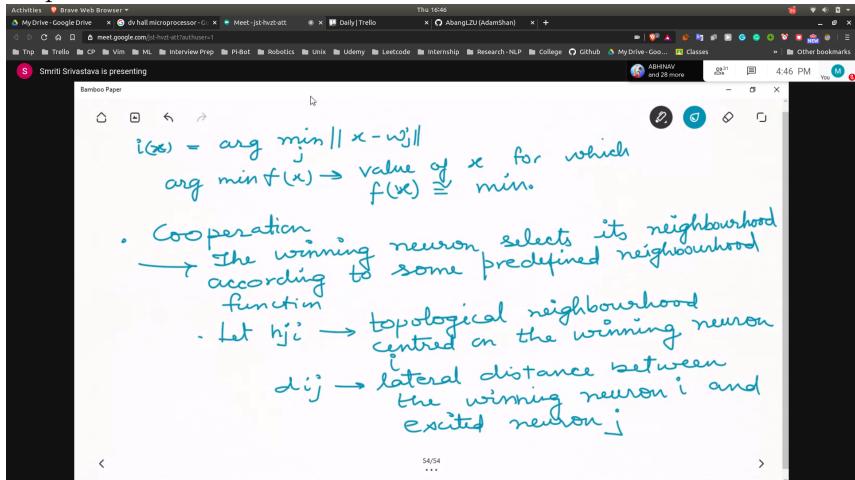
- initialize random weights
- competition



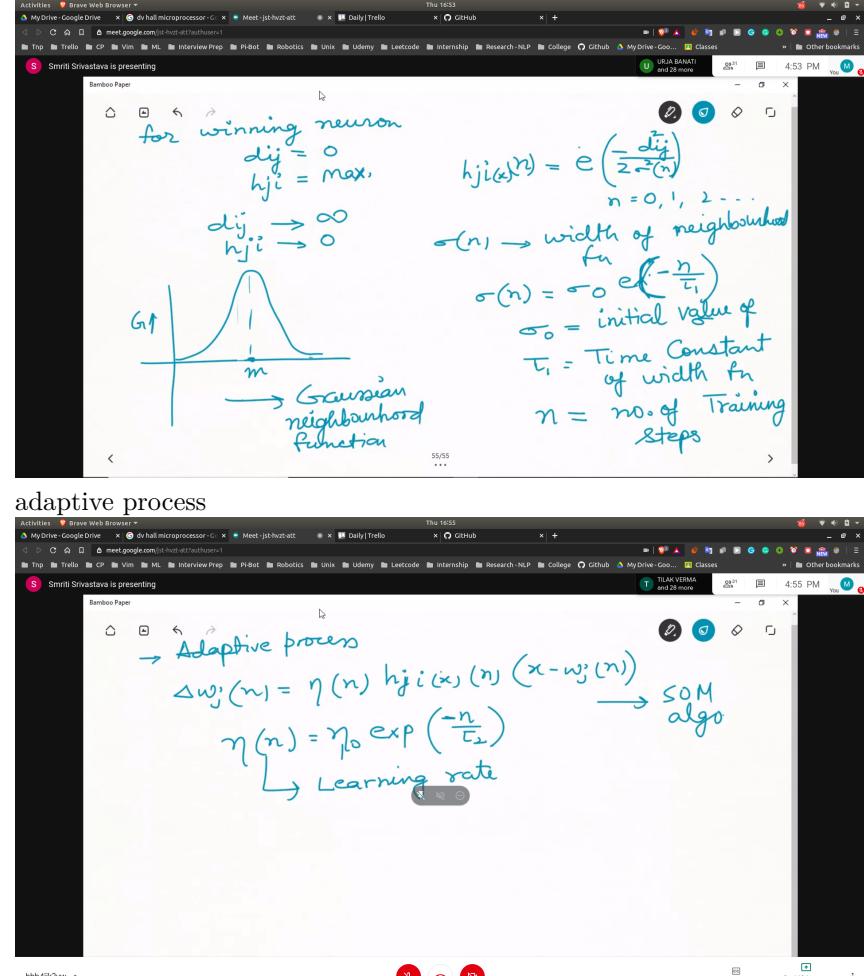
- $i(x)$ is the function to identify the best neuron

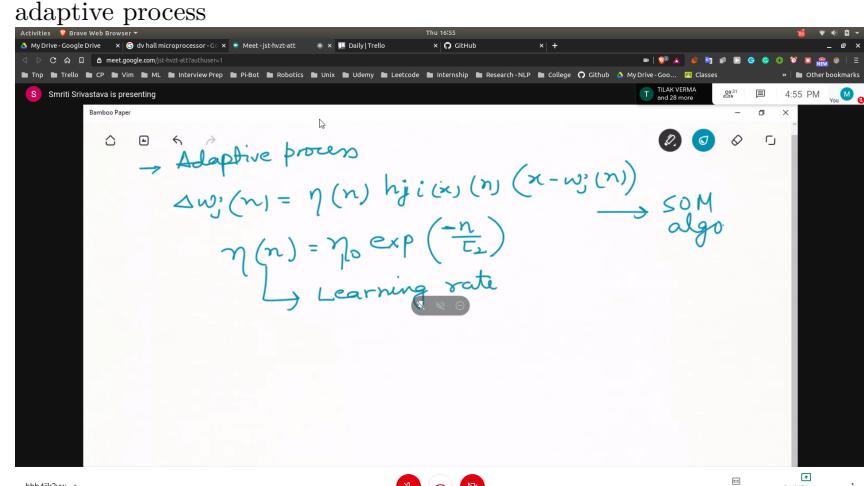


- after this the competition is over and best neuron is selected
- cooperation



- assign values using gaussian neighbourhood function

- 

A screenshot of a Bamboo Paper presentation slide. The slide contains handwritten mathematical notes and a graph. On the left, there is a graph of a Gaussian function labeled G_1 , with a peak at m . The graph is labeled "Gaussian neighbourhood function". Above the graph, handwritten text says "for winning neuron" and defines $d_{ij} = 0$ and $h_{ji} = \max$. To the right, the formula for the Gaussian function is given as $h_{ji}(n) = e^{\left(\frac{-d_{ij}}{2\sigma^2(n)}\right)}$ with the condition $n = 0, 1, 2, \dots$. Below this, the width of the neighborhood is defined as $\sigma(n) \rightarrow \text{width of neighbourhood}$. The formula for $\sigma(n)$ is shown as $\sigma(n) = \sigma_0 e^{\left(-\frac{n}{\tau_i}\right)}$. The initial value is σ_0 , the time constant is τ_i , and the number of training steps is n .
- 

A screenshot of a Bamboo Paper presentation slide. It shows the adaptive process equation $\Delta w_j(n) = \eta(n) h_{ji}(n) (x - w_j(n))$ and the learning rate formula $\eta(n) = \eta_0 \exp\left(-\frac{n}{\tau_s}\right)$. A bracket under the learning rate formula is labeled "Learning rate". To the right, it says "SOM algo".