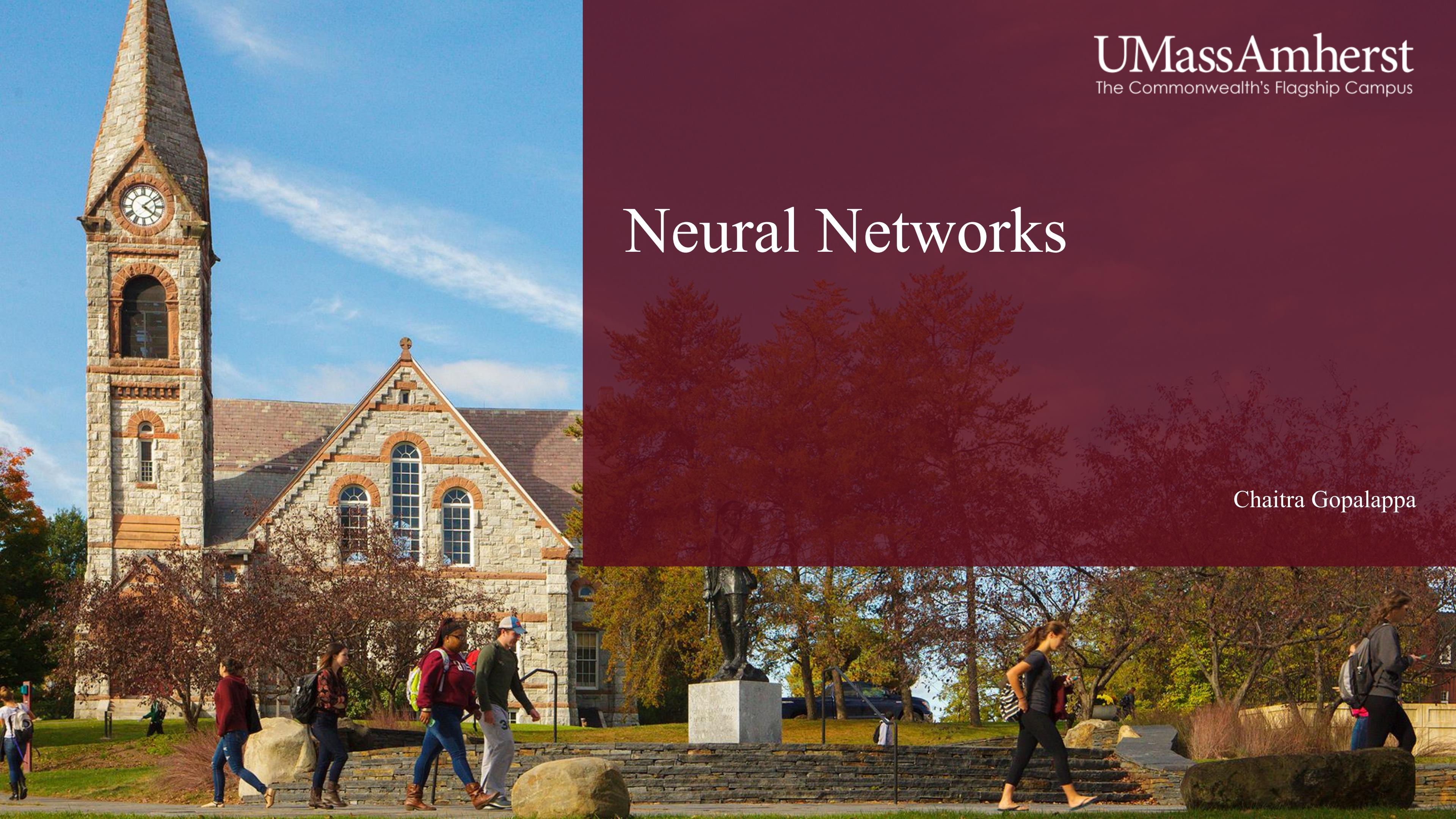
UMassAmherst
The Commonwealth's Flagship Campus

Neural Networks

Chaitra Gopalappa

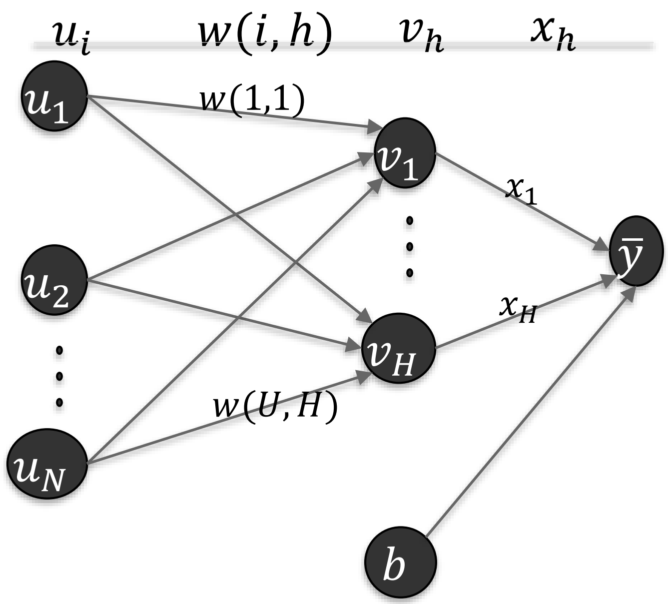


Backprop (back propagation)
Example with one hidden layer and sigmoid activation

- **Backprop concepts**
- Objective

$$\min_{b,\boldsymbol{x},\boldsymbol{w}} L = \min_{b,\boldsymbol{x},\boldsymbol{w}} \frac{1}{2} \sum_{p=1}^{P} \left(y_p - \bar{y}_p\right)^2$$

  - $b, \omega[.,.]$, and $x[.]$ are the decision variables from perspective of Backprop.
  - $\bar{y}_p$ = output from trained net for $\vec{u}_p$; p are the data samples
  - y_p = actual data
  - $L = f(b, \omega[.,.], x[.])$ is the lost function
  - P = number of samples; N = number of input nodes



Backprop algorithm

1. Initialize
   1. Initialize $b, \omega(.,.)$, and $x(.)$ to random values
   2. Set SSE_{old} to very large value.
   3. Set $m = 0$ (iteration number)

2. Forward pass
   1. Compute $v_p^*[h] = \sum_i w[i,h]u_p[\text{i}]$ for each p and each h.
   2. Compute $v_p[h] = \frac{1}{1+e^{-v_p^*[h]}}$ for each p and each h
   3. Compute $\bar{y}_p = b + \sum_h x[h]v_p[h]$for each p (data samples)

3. Apply SD transformations

$$b_m \leftarrow b_{m-1} - \mu \frac{\partial(L)}{\partial b}$$

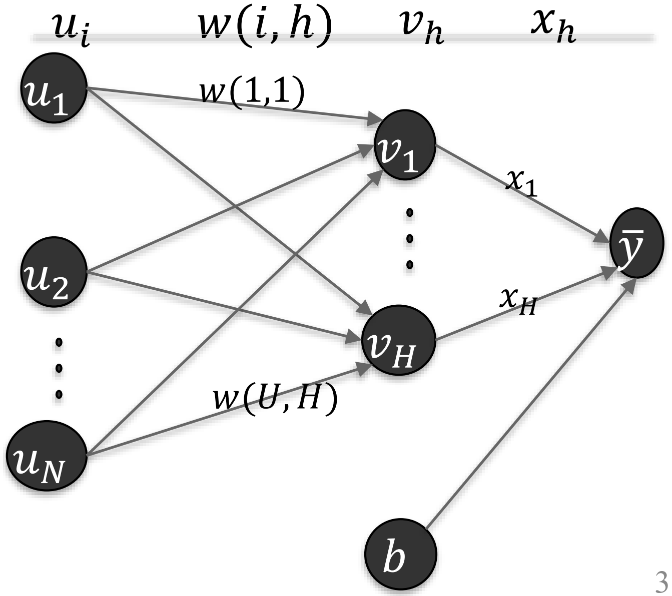
$$\omega_m[i,h] \leftarrow \omega_{m-1}[i,h] - \mu \frac{\partial(\text{L})}{\partial\omega[i,h]}$$

$$x_m[h] \leftarrow x_{m-1}[h] - \mu \frac{\partial(L)}{\partial x[h]}$$

4. Set $m = m + 1$
   1. Calculate $L_{new} = \sum_p \left(y_p - \bar{y}_p\right)^2$
   2. Update μ
   3. If $[L_{\text{new}} - L] <$ tolerance $\rightarrow STOP$ Otherwise set $L_{\text{old}} = L_{\text{new}}$ got step 2.

Derivation of Backprop derivatives

- Main transformation is the steepest descent $\vec{x} \leftarrow \vec{x} - \mu \nabla \mathrm{f}(\vec{x})$; but here we are solving for $\omega[.,.]$, and $x[.]$
- $b_m \leftarrow b_{m-1} - \mu \frac{\partial(L)}{\partial b}$
- $\omega_m[i,h] \leftarrow \omega_{m-1}[i,h] - \frac{\mu \partial(\mathrm{L})}{\partial \omega[i,h]}$
- $x_m[h] \leftarrow x_{m-1}[h] - \mu \frac{\partial(L)}{\partial x[h]}$



$\frac{\partial L}{\partial x[h]}$=? $\frac{\partial L}{\partial b}$=? Apply chain rule of calculus

- $\frac{\partial L}{\partial x[h]} = \frac{1}{2}\sum_{p=1}^{P} \frac{\partial}{\partial x[h]}\left(y_p - \bar{y}_p\right)^2$

$$= \frac{2}{2}\sum_{p=1}^{P} \left(y_p - \bar{y}_p\right)\frac{\partial}{\partial x[h]}\left(y_p - \bar{y}_p\right)$$

$$= \sum_p \left(y_p - \bar{y}_p\right)\left(-\frac{\partial \bar{y}_p}{\partial x[h]}\right)$$

$$= \sum_p \left(y_p - \bar{y}_p\right)\left(-\frac{\partial\left(b + \sum_h x[h]v_p[h]\right)}{\partial x[h]}\right)$$

$$\frac{\partial L}{\partial x[h]} = -\sum_p \left(y_p - \bar{y}_p\right)v_p[h]$$

- $\frac{\partial L}{\partial b} = -\sum_p \left(y_p - \bar{y}_p\right)$

$$\frac{\partial L}{\omega(i,h)} = ?$$

$$\frac{\partial L}{\omega[i,h]} = \frac{1}{2}\sum_p \frac{\partial}{\partial\omega[i,h]}\left(y_p - \bar{y}_p\right)^2$$

$$= \frac{1}{2}\sum_p 2\left(y_p - \bar{y}_p\right)\frac{\partial}{\partial\omega[i,h]}\left(y_p - \bar{y}_p\right)$$

$$= \sum_p \left(y_p - \bar{y}_p\right)\left(-\frac{\partial}{\partial w[i,h]}\bar{y}_p\right)$$

$$= -\sum_p \left(y_p - \bar{y}_p\right)\cdot\left(\frac{\partial \bar{y}_p}{\partial v_p[h]}\frac{\partial v_p[h]}{\partial v_p^*[h]}\frac{\partial v_p^*[h]}{\partial\omega[i,h]}\right)$$

$$\frac{\partial L}{\partial \mathrm{w}[i,h]} = -\sum_p \left(\left(y_p - \bar{y}_p\right)x[h]\,v_p[h]\left(1 - v_p[h]\right)u_p[i]\right)$$

This changes for different activation functions

- Feed forward equations
  - $v_p^*[h] = \sum_i \omega[i,h]u_p[\mathrm{i}];$
  - $v_p[h] = \frac{1}{1+e^{-v^*[h]}};$
  - $y_p = b + \sum_h v_p[h]x[h]$
- $\frac{\partial v_p^*[h]}{\partial w(i,h)} = u_p[\mathrm{i}]$
- $\frac{\partial v_p[h]}{\partial v_p^*[h]} = \frac{\partial\left(1+e^{-v_p^*[h]}\right)^{-1}}{\partial v_p^*[h]}$

$$= \frac{-1}{\left[1+e^{-v_p^*[h]}\right]^2}e^{-v_p^*[h]}(-1)$$

$$= \frac{1+e^{-v_p^*[h]}-1}{\left[1+e^{-v_p^*[h]}\right]^2} \textit{ (add + 1 , - 1 in numerator)}$$

$$= \frac{1+e^{-v_p^*[h]}}{\left[1+e^{-v_p^*[h]}\right]^2} - \frac{1}{\left[1+e^{-v_p^*[h]}\right]^2}$$

$$= \frac{1}{1+e^{-v_p^*[h]}} - \frac{1}{\left[1+e^{-v_p^*[h]}\right]^2}$$

$$= v_p[h]\left(1 - v_p[h]\right)$$

Backprop algorithm

1. Initialize
   1. Initialize $b, \omega(.,.)$, and $x(.)$ to random values
   2. Set SSE_{old} to very large value.
   3. Set $m = 0$ (iteration number)
2. Forward pass
   1. Compute $v_p^*[h] = \sum_i w[i,h]u_p[\mathrm{i}]$ for each p and each h.
   2. Compute $v_p[h] = \frac{1}{1+e^{-v_p^*[h]}}$ for each p and each h
   3. Compute $\bar{y}_p = b + \sum_h x[h]v_p[h]$ for each p (data samples)
3. Apply SD transformations

$$b_m \leftarrow b_{m-1} - \mu \frac{\partial(L)}{\partial b}; \qquad \frac{\partial L}{\partial b} = -\sum_p \left(y_p - \bar{y}_p\right)$$

$$\omega_m[i,h] \leftarrow \omega_{m-1}[i,h] - \mu \frac{\partial(L)}{\partial \omega[i,h]}; \qquad \frac{\partial L}{\partial \mathrm{w}[i,h]} = -\sum_p \left((y_p - \bar{y}_p)x[h]v_p[h](1 - v_p[h])u_p[i]\right)$$

$$x_m[h] \leftarrow x_{m-1}[h] - \mu \frac{\partial(L)}{\partial x[h]}; \qquad \frac{\partial L}{\partial x[h]} = -\sum_p \left(y_p - \bar{y}_p\right)v_p[h]$$

4. Set $m = m + 1$
   1. Calculate $L_{new} = \sum_p \left(y_p - \bar{y}_p\right)^2$
   2. Update μ
   3. If $[L_{\text{new}} - L_{\text{old}}] <$ tolerance $\rightarrow$ *STOP* Otherwise set $L_{\text{old}} = L_{\text{new}}$ got step 2.

- Do multiple iterations of the algorithm to start at different initial points
- Initialize $b, \omega[.,.]$, and $x[.]$ to select from a random range; Try a few different ranges
- Try a few different options for learning rate ($\mu = small\ constant; \mu = \frac{A}{B+m}; \mu = \frac{1}{m};$)

This applies to sigmoid activation; change if using a different activation function

UMassAmherst
The Commonwealth's Flagship Campus