Discussion 7

EE599: Deep Learning

Olaoluwa Adigun

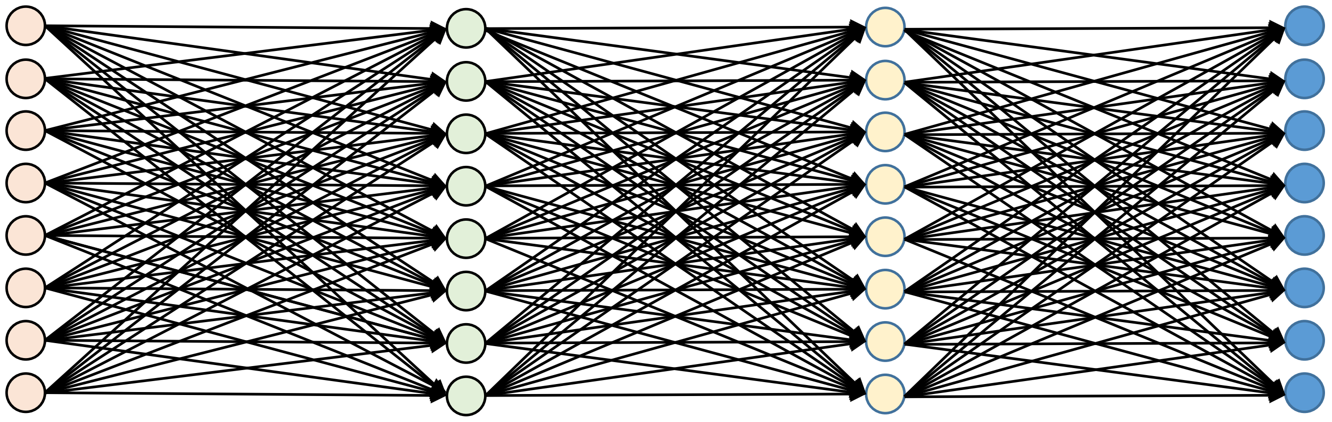
Spring 2020

USC University of Southern California

Contributors: Olaoluwa Adigun

Training MLP with Backpropagation

Find the best parameter Θ^* that minimizes the cost function



Input layer x

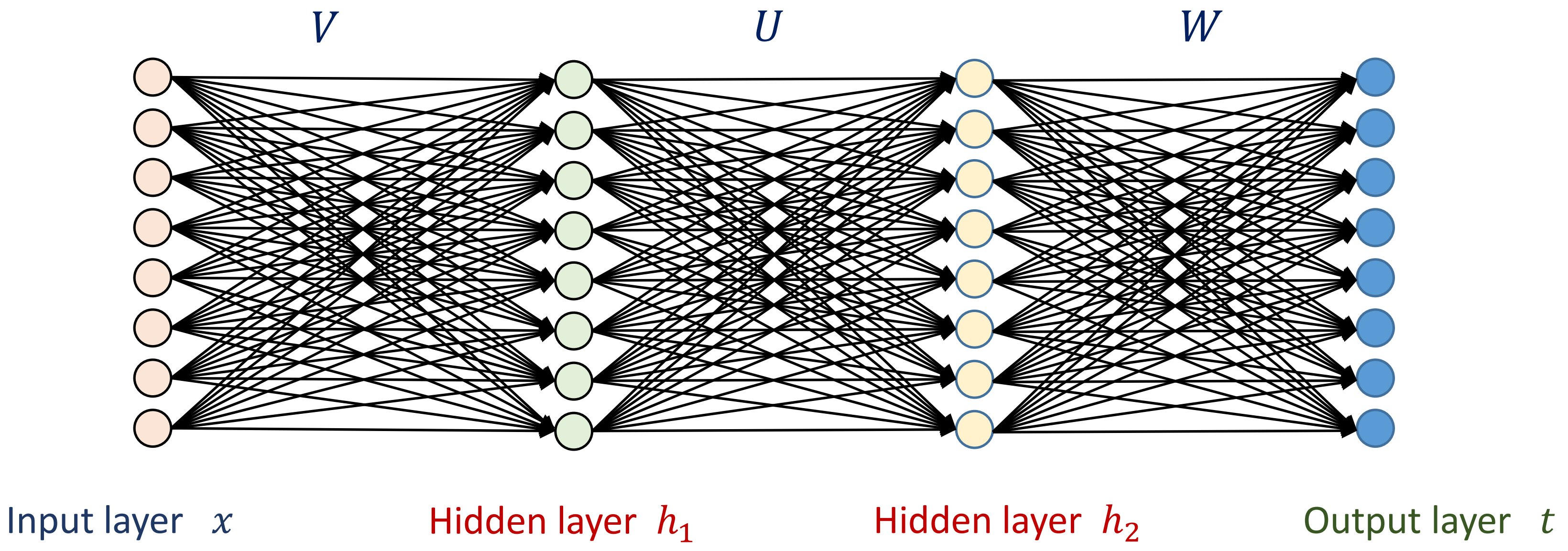
Hidden layer h_1

Hidden layer h_2

Output layer t

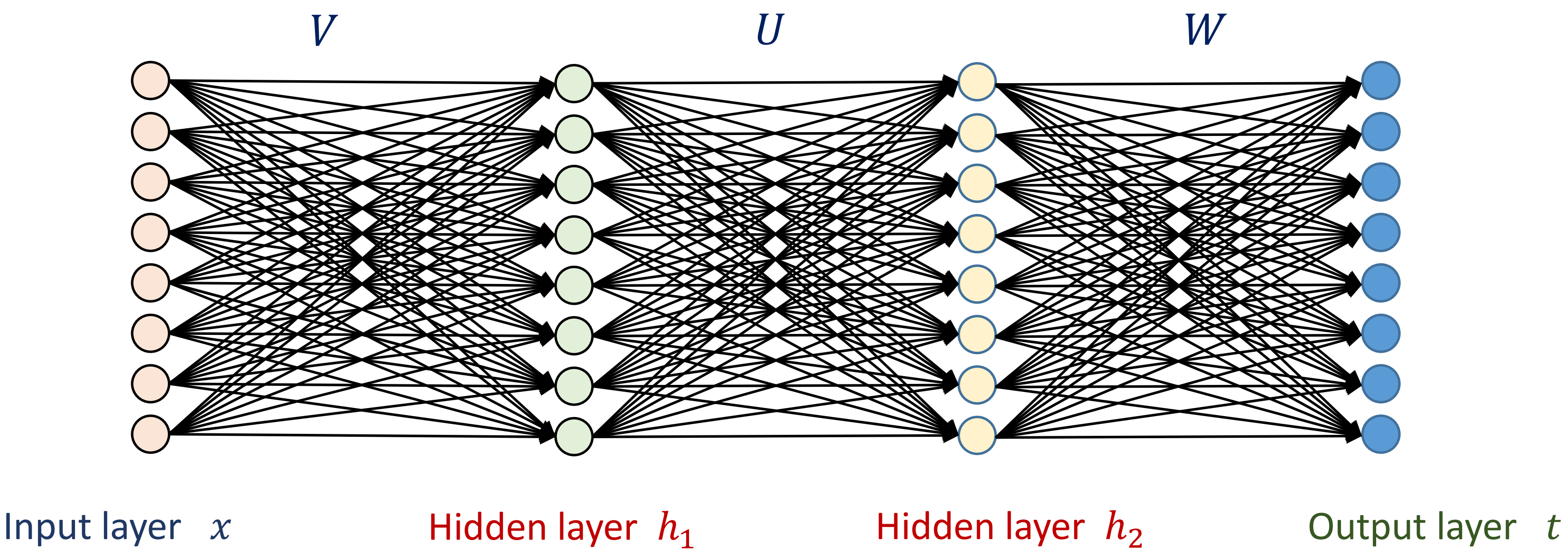
Backpropagation uses a variant of stochastic gradient descent to updates the weights

Model Architecture



- Input Layer x: L neurons with *identity* activation
- Hidden Layer h_1: I neurons with *sigmoid* activation
- Hidden Layer h_2: J neurons with *sigmoid* activation
- Output Layer t: J neurons with *softmax* activation

Forward Pass over MLP Model (Inference)



$$o_l^{h_1} = \sum_{i=1}^{I} v_{li}\ a_i^x + b_l^{h_1}$$

$$a_l^{h_1} = \frac{1}{1 + \exp^{-o_l^{h_1}}}$$

$$o_j^{h_2} = \sum_{l=1}^{L} w_{jl}\ a_l^{h_1} + b_j^{h_2}$$

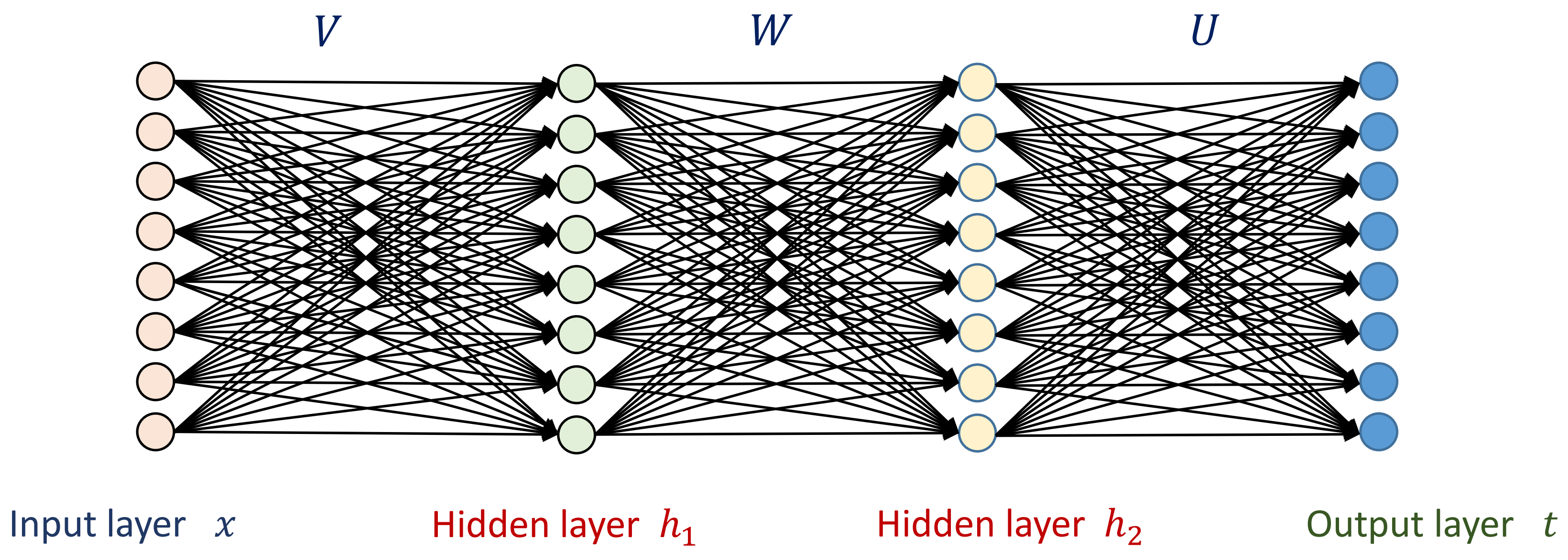
$$a_j^{h_2} = \frac{1}{1 + \exp^{-o_j^{h_2}}}$$

$$o_k^t = \sum_{j=1}^{J} u_{kj} a_j^{h_2} + b_k^t$$

$$a_k^t = \frac{\exp(o_k^t)}{\sum_{r=1}^{K} \exp(o_r^t)}$$

- $b_l^{h_1}$: Bias to the l^{th} neuron at the hidden layer h_1
- v_{li} : Weight connecting the l^{th} neuron of the hidden layer h_1 to the i^{th} input neuron
- $b_j^{h_2}$: Bias to the j^{th} neuron at the hidden layer h_2
- w_{jl} : Weight connecting the l^{th} neuron of layer h_1 to the l^{th} neuron of layer h_2
- b_k^t : Bias to the k^{th} output neuron
- u_{kj} : Weight connecting the j^{th} neuron of the hidden layer h_1 to the k^{th} output neuron

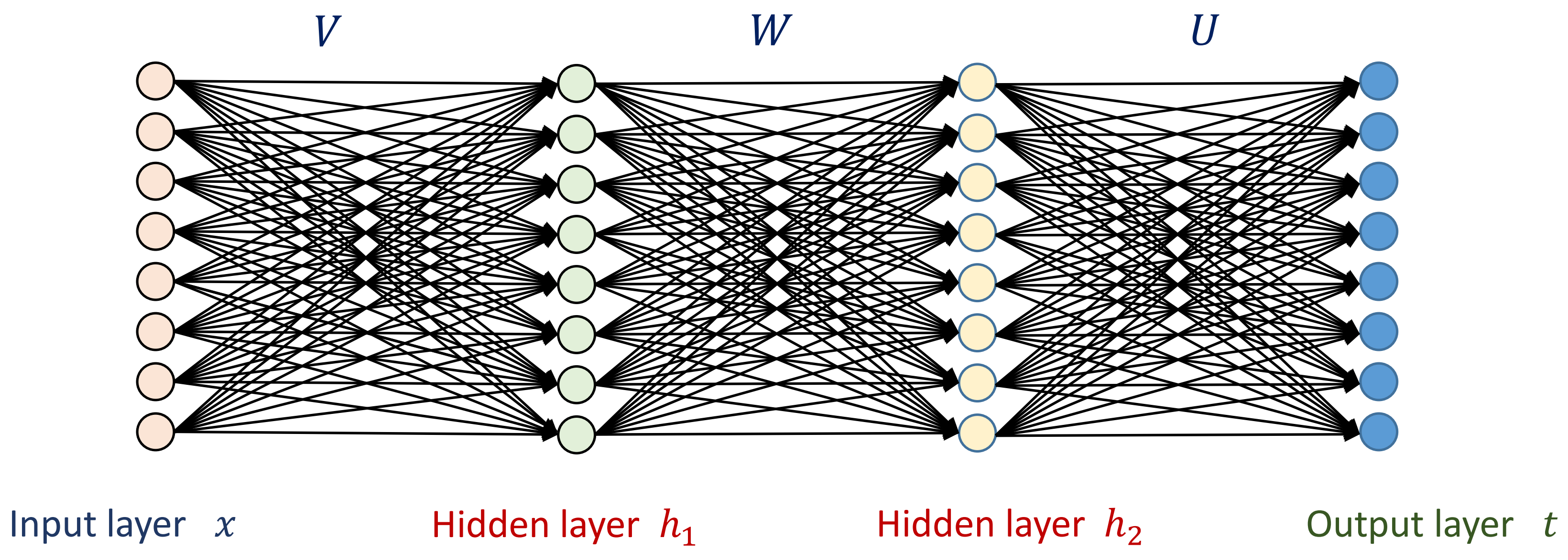
Forward Pass over MLP Model (Inference)



$$\mathbf{o}^{h_1} = \left(\mathbf{V} \times \mathbf{a}^{x}\right) + \mathbf{b}^{h_1}$$
$$\mathbf{a}^{h_1} = \sigma\left(\mathbf{o}^{h_1}\right)$$
$$\mathbf{o}^{h_2} = \left(\mathbf{W} \times \mathbf{a}^{h_1}\right) + \mathbf{b}^{h_2}$$
$$\mathbf{a}^{h_2} = \sigma\left(\mathbf{o}^{h_2}\right)$$
$$\mathbf{o}^{t} = \left(\mathbf{U} \times \mathbf{a}^{h_2}\right) + \mathbf{b}^{t}$$
$$\mathbf{a}^{t} = \mathrm{Softmax}(\mathbf{o}^{t})$$

- $\boldsymbol{b}^{h_1}$: Bias vector to the hidden layer $\boldsymbol{h}_1$
- $\boldsymbol{V}$: Weight connecting the input layer to hidden layer $\boldsymbol{h}_1$
- $\boldsymbol{b}^{h_2}$: Bias vector to the hidden layer $\boldsymbol{h}_2$
- $\boldsymbol{W}$: Weight connecting the hidden layer $\boldsymbol{h}_1$ to hidden layer $\boldsymbol{h}_2$
- $\boldsymbol{b}^{t}$: Bias vector to the output layer
- $\boldsymbol{U}$: Weight connecting the hidden layer $\boldsymbol{h}_2$ to the output layer

Error Function and Backpropagation



$$\mathbf{o}^{h_1} = (\mathbf{V} \times \mathbf{a}^{x}) + \mathbf{b}^{h_1}$$
$$\mathbf{a}^{h_1} = \sigma(\mathbf{o}^{h_1})$$
$$\mathbf{o}^{h_2} = (\mathbf{W} \times \mathbf{a}^{h_1}) + \mathbf{b}^{h_2}$$
$$\mathbf{a}^{h_2} = \sigma(\mathbf{o}^{h_2})$$
$$\mathbf{o}^{t} = (\mathbf{U} \times \mathbf{a}^{h_2}) + \mathbf{b}^{t}$$
$$\mathbf{a}^{t} = \mathrm{Softmax}(\mathbf{o}^{t})$$

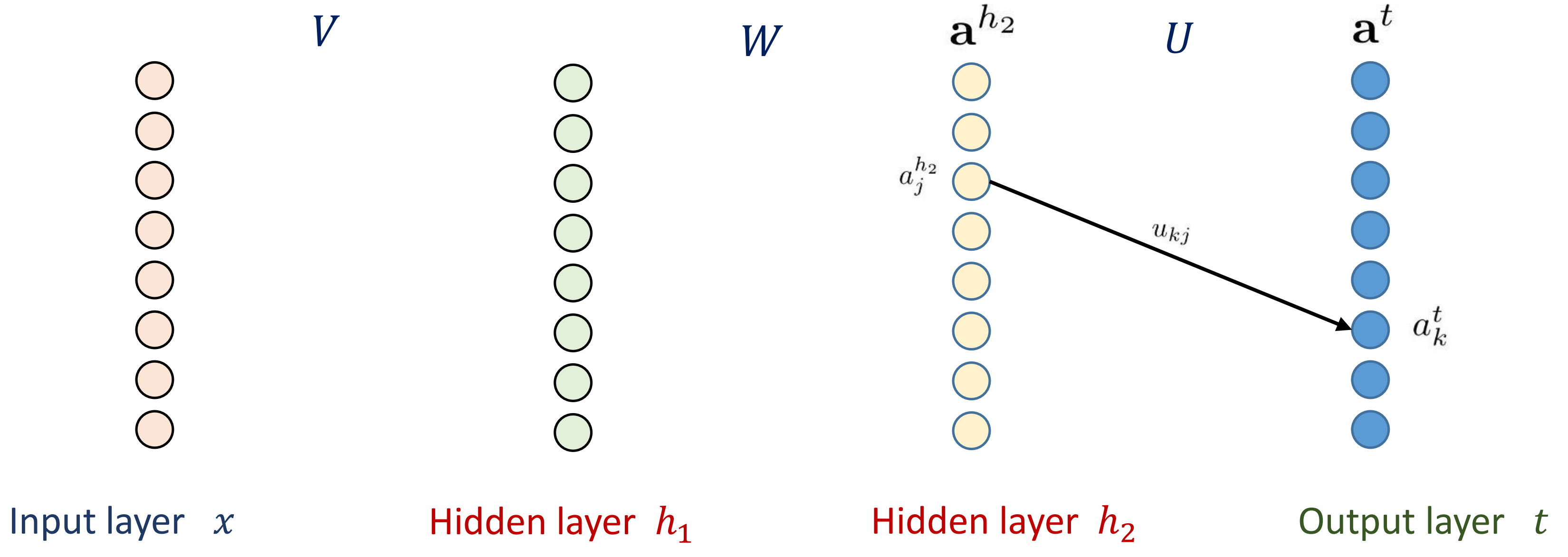
Cross entropy

$$E(\Theta) = -\sum_{k=1}^{K} y_k \log a_k^t = -\mathbf{y}^T \log \mathbf{a}^t$$

Update rule

$$\Theta^{(n+1)} = \Theta^{(n)} - \eta \nabla E(\Theta)\Big|_{\Theta=\Theta^{(n)}}$$

Backpropagation Update Rule

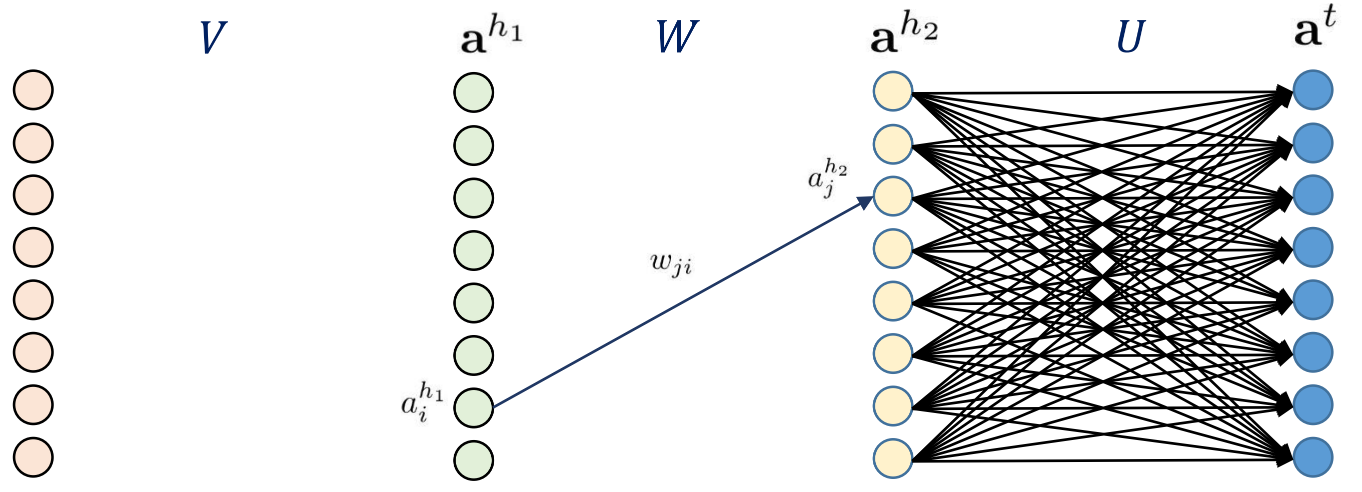


$$
\begin{aligned}
\mathbf{o}^{h_1} &= \left(\mathbf{V} \times \mathbf{a}^{x}\right) + \mathbf{b}^{h_1} \\
\mathbf{a}^{h_1} &= \sigma\left(\mathbf{o}^{h_1}\right) \\
\mathbf{o}^{h_2} &= \left(\mathbf{W} \times \mathbf{a}^{h_1}\right) + \mathbf{b}^{h_2} \\
\mathbf{a}^{h_2} &= \sigma\left(\mathbf{o}^{h_2}\right) \\
\mathbf{o}^{t} &= \left(\mathbf{U} \times \mathbf{a}^{h_2}\right) + \mathbf{b}^{t} \\
\mathbf{a}^{t} &= \mathrm{Softmax}(\mathbf{o}^{t})
\end{aligned}
$$

$$\frac{\partial E}{\partial u_{kj}} = \left(y_k - a_k^t\right) a_j^{h_2}$$

$$\nabla_U E(\Theta) = \left(\mathbf{y} - \mathbf{a}^t\right)^T \mathbf{a}^{h_2}$$

Backpropagation Update Rule



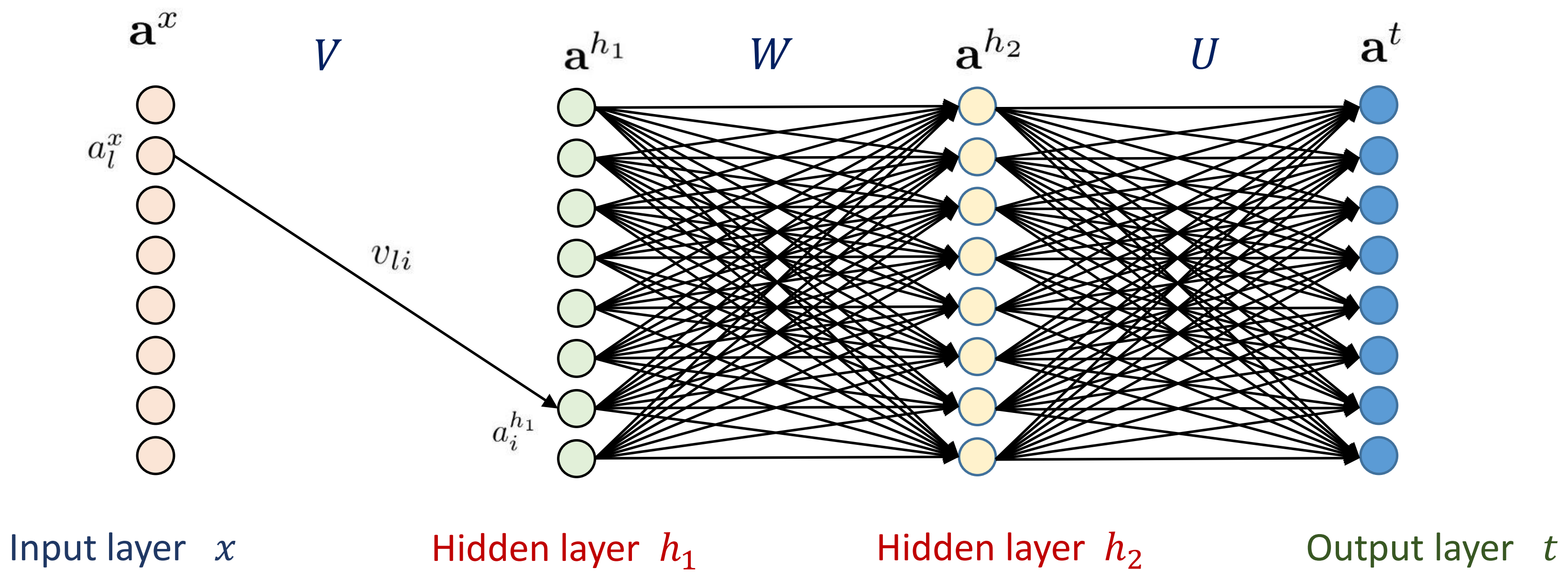
Input layer x Hidden layer h_1 Hidden layer h_2 Output layer t

$$
\begin{aligned}
\mathbf{o}^{h_1} &= \left(\mathbf{V} \times \mathbf{a}^{x}\right) + \mathbf{b}^{h_1} \\
\mathbf{a}^{h_1} &= \sigma\left(\mathbf{o}^{h_1}\right) \\
\mathbf{o}^{h_2} &= \left(\mathbf{W} \times \mathbf{a}^{h_1}\right) + \mathbf{b}^{h_2} \\
\mathbf{a}^{h_2} &= \sigma\left(\mathbf{o}^{h_2}\right) \\
\mathbf{o}^{t} &= \left(\mathbf{U} \times \mathbf{a}^{h_2}\right) + \mathbf{b}^{t} \\
\mathbf{a}^{t} &= \mathrm{Softmax}(\mathbf{o}^{t})
\end{aligned}
$$

$$
\frac{\partial E}{\partial w_{jl}} = \left(\sum_{k=1}^{K} (y_k - a_k^t) u_{kj}\right) a_j^{h_2} (1 - a_j^{h_2}) a_l^{h_1}
$$

$$
\nabla_W E(\Theta) = \left(\left(\left(\mathbf{y} - \mathbf{a}^t\right)^T \mathbf{U}\right) \odot \mathbf{a}^{h_2} \odot (1 - \mathbf{a}^{h_2})\right)^T \mathbf{a}^{h_1}
$$

Backpropagation Update Rule



$$
\begin{aligned}
\mathbf{o}^{h_1} &= \left(\mathbf{V} \times \mathbf{a}^x\right) + \mathbf{b}^{h_1} \\
\mathbf{a}^{h_1} &= \sigma\left(\mathbf{o}^{h_1}\right) \\
\mathbf{o}^{h_2} &= \left(\mathbf{W} \times \mathbf{a}^{h_1}\right) + \mathbf{b}^{h_2} \\
\mathbf{a}^{h_2} &= \sigma\left(\mathbf{o}^{h_2}\right) \\
\mathbf{o}^t &= \left(\mathbf{U} \times \mathbf{a}^{h_2}\right) + \mathbf{b}^t \\
\mathbf{a}^t &= \mathrm{Softmax}(\mathbf{o}^t)
\end{aligned}
$$

Update rule

$$\frac{\partial E}{\partial v_{li}} = \Big(\sum_{j=1}^{J}\Big(\sum_{k=1}^{K}(y_k - a_k^t)u_{kj}\Big)a_j^{h_2}(1 - a_j^{h_2})w_{jl}\Big)a_l^{h_1}(1 - a_l^{h_1})a_i^x$$

$$\nabla_V E(\Theta) = \left(\left(\left(\left(\mathbf{y} - \mathbf{a}^t\right)^T \mathbf{U}\right) \odot \mathbf{a}^{h_2} \odot \left(1 - \mathbf{a}^{h_2}\right)\right) \times \mathbf{W}^T\right) \odot \mathbf{a}^{h_1} \odot \left(1 - \mathbf{a}^{h_1}\right)\right)^T \mathbf{a}^x$$