AMRITA
VISHWA VIDYAPEETHAM
DEEMED TO BE UNIVERSITY

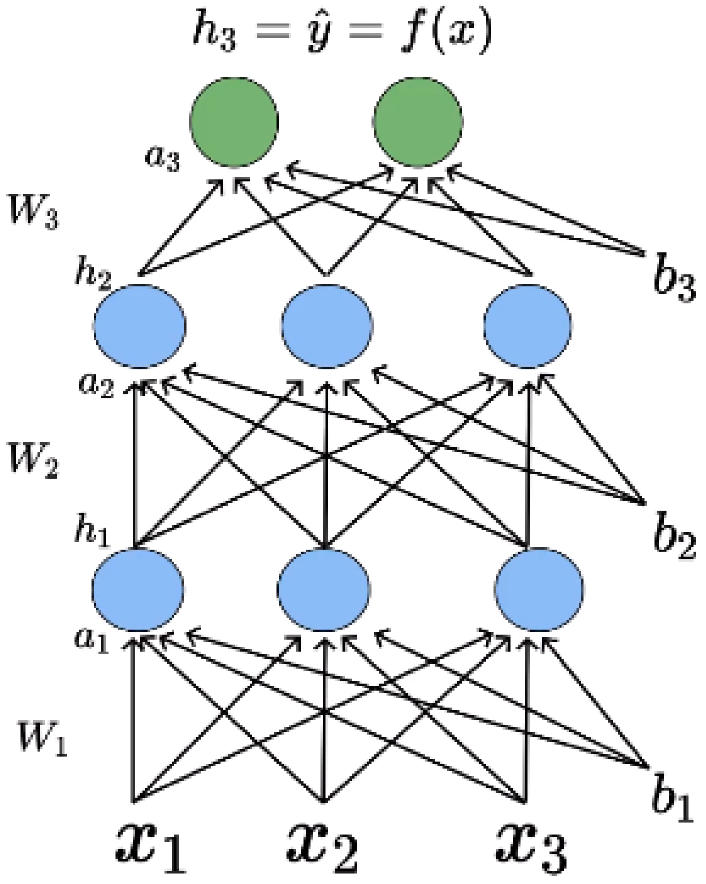
19CSE437
DEEP LEARNING FOR COMPUTER VISION
L-T-P-C: 2-0-3-3

Feed Forward Neural Networks

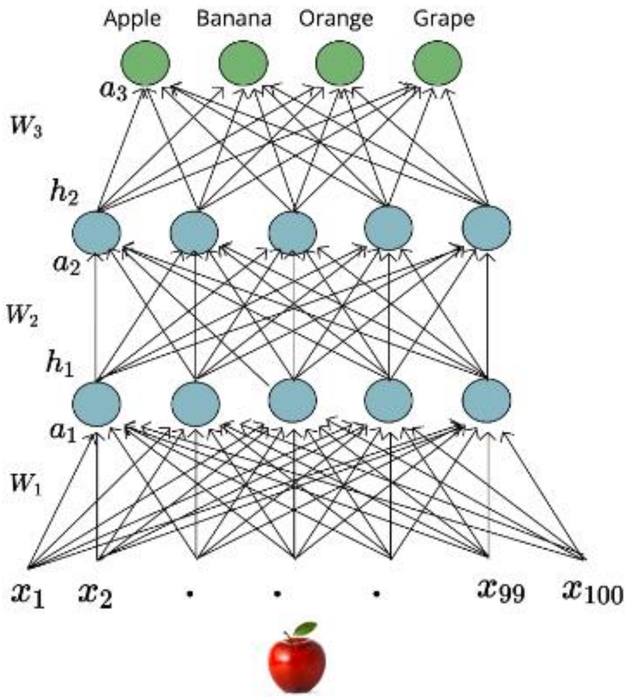
- Back propagation Algorithm

*Citation Note: content, of this presentation were inspired by the awesome lectures and the material offered by Prof. **Mitesh M. Khapra** on NPTEL's Deep Learning course*

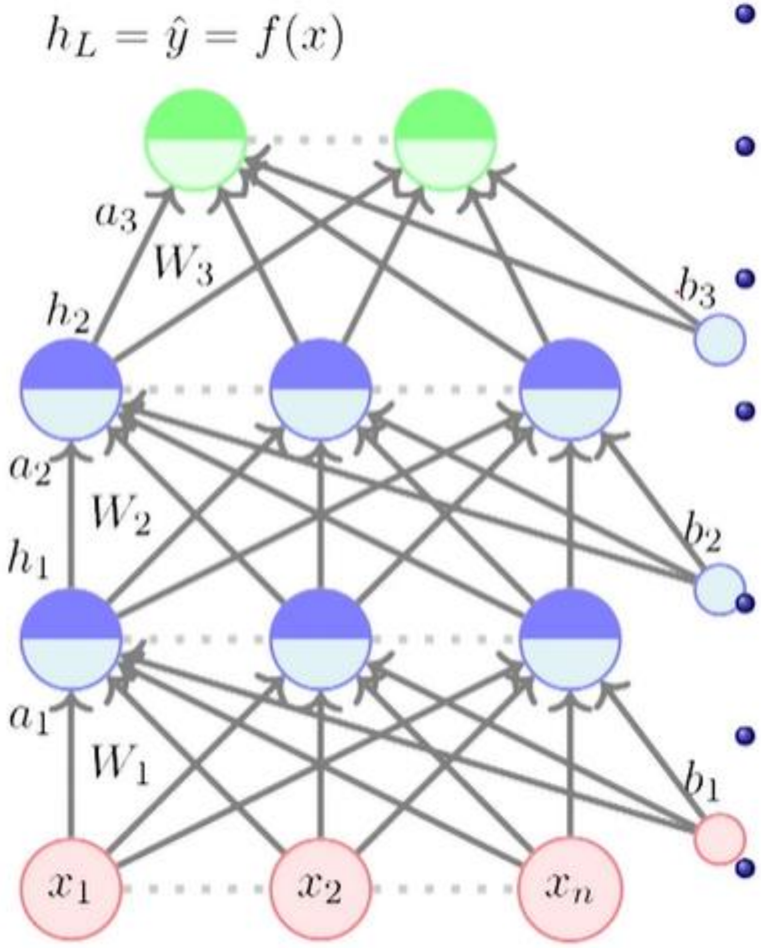
Feed Forward Neural Networks



- The pre-activation at layer 'i' is given by
  $$a_i(x) = W_i h_{i-1}(x) + b_i$$
- The activation at layer 'i' is given by
  $$h_i(x) = g(a_i(x))$$
  where 'g' is called as the activation function
- The activation at output layer 'L' is given by
  $$f(x) = h_L = O(a_L)$$
  where 'O' is called as the output activation function

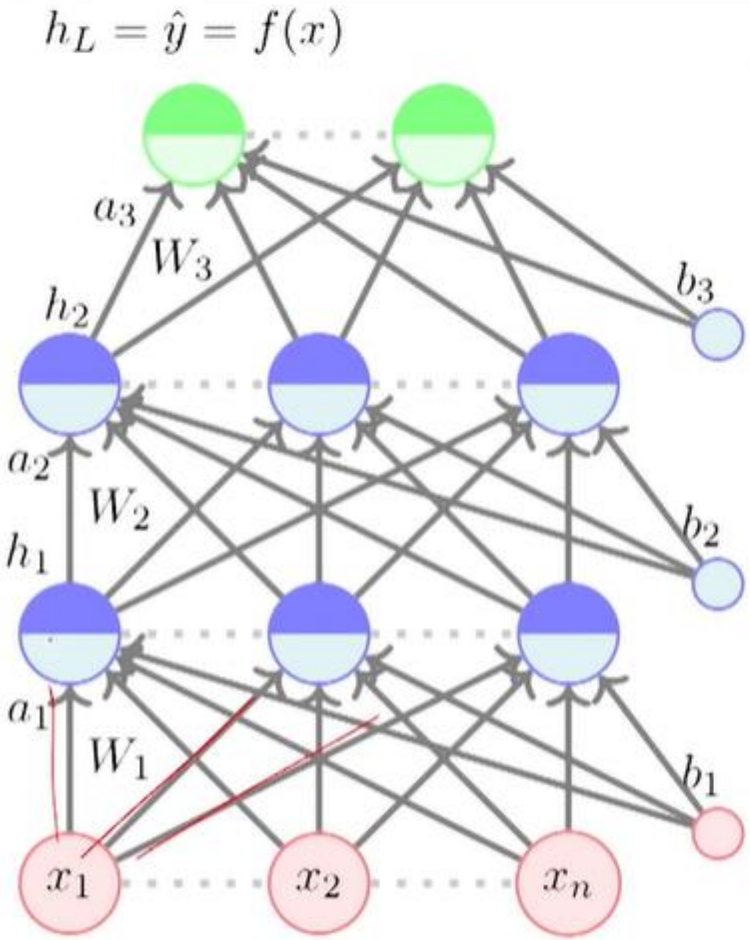


Feed Forward Neural Networks



- The input to the network is an **n**-dimensional vector
- The network contains $\mathbf{L-1}$ hidden layers (2, in this case) having **n** neurons each
- Finally, there is one output layer containing **k** neurons (say, corresponding to **k** classes)
- Each neuron in the hidden layer and output layer can be split into two parts : pre-activation and activation (a_i and h_i are vectors)
- The input layer can be called the 0-th layer and the output layer can be called the (L)-th layer
- $W_i \in \mathbb{R}^{n \times n}$ and $b_i \in \mathbb{R}^n$ are the weight and bias between layers $i-1$ and i $(0 < i < L)$
- $W_L \in \mathbb{R}^{n \times k}$ and $b_L \in \mathbb{R}^k$ are the weight and bias between the last hidden layer and the output layer ($L = 3$ in this case)

Feed Forward Neural Networks



- The pre-activation at layer i is given by

$$a_i(x) = b_i + W_i h_{i-1}(x)$$

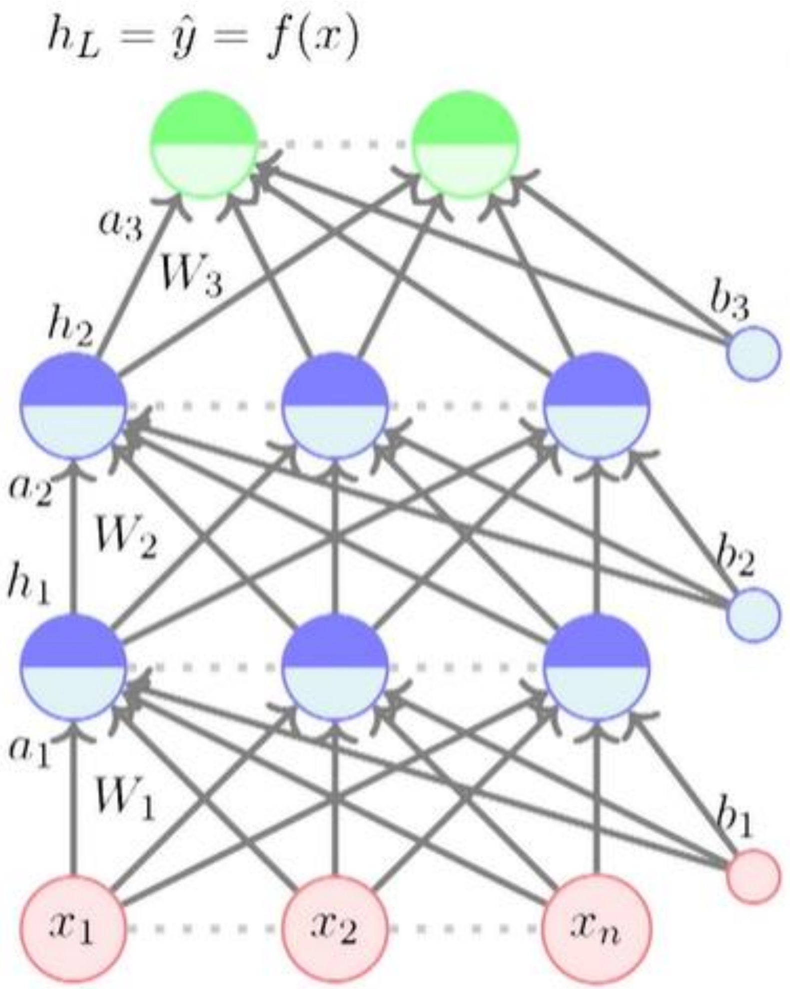
For example, $a_1 = b_1 + W_1 h_0$

$$\begin{bmatrix} a_{11} \\ a_{12} \\ a_{13} \end{bmatrix} = \begin{bmatrix} b_{11} \\ b_{12} \\ b_{13} \end{bmatrix} + \begin{bmatrix} W_{111} & W_{112} & W_{113} \\ W_{121} & W_{122} & W_{123} \\ W_{131} & W_{132} & W_{133} \end{bmatrix} \begin{bmatrix} h_{01} = x_1 \\ h_{02} = x_2 \\ h_{03} = x_3 \end{bmatrix}$$

$$= \begin{bmatrix} b_{11} \\ b_{12} \\ b_{13} \end{bmatrix} + \begin{bmatrix} W_{111}x_1 + W_{112}x_2 + W_{113}x_3 \\ W_{121}x_1 + W_{122}x_2 + W_{123}x_3 \\ W_{131}x_1 + W_{132}x_2 + W_{133}x_3 \end{bmatrix}$$

$$= \begin{bmatrix} \sum W_{11i}x_i + b_{11} \\ \sum W_{12i}x_i + b_{12} \\ \sum W_{13i}x_i + b_{13} \end{bmatrix}$$

Feed Forward Neural Networks



- The pre-activation at layer i is given by

$$a_i = b_i + W_i h_{i-1}$$

- The activation at layer i is given by

$$h_i = g(a_i)$$

where g is called the activation function (for example, logistic, tanh, linear, *etc.*)

- The activation at layer i is given by

$$f(x) = h_L = O(a_L)$$

where O is the output activation function (for example, softmax, linear, *etc.*)

Learning Algorithm- backpropagation

Initialise w, b

Iterate over data:

$compute\ \hat{y}$

$compute\ \mathscr{L}(w, b)$

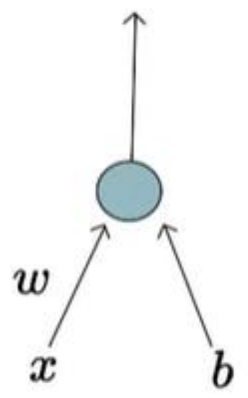
$w_{111} = w_{111} - \eta \Delta w_{111}$

$w_{112} = w_{112} - \eta \Delta w_{112}$

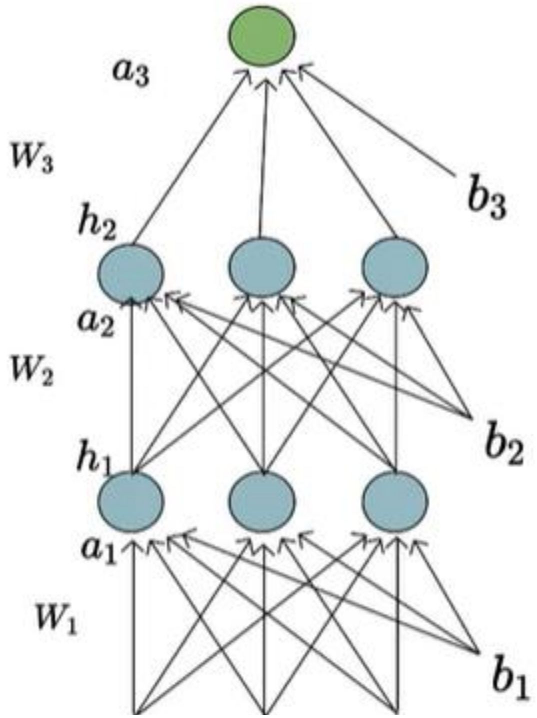
....

$w_{313} = w_{313} - \eta \Delta w_{313}$

till satisfied



$$\Delta w_t = \frac{\partial \mathscr{L}(w,b)}{\partial w}$$



$$\mathscr{L} = \frac{1}{5} \sum_{i=1}^{i=5} (f(x_i) - y_i)^2$$

$$\frac{\partial \mathscr{L}}{\partial w} = \frac{\partial}{\partial w} [\frac{1}{5} \sum_{i=1}^{i=5} (f(x_i) - y_i)]^2$$

$$\Delta w = \frac{\partial \mathscr{L}}{\partial w} = \frac{1}{5} \sum_{i=1}^{i=5} \frac{\partial}{\partial w} (f(x_i) - y_i)^2$$

Calculation of Δw

$$\nabla w = \frac{\partial}{\partial w}[\tfrac{1}{2} * (f(x) - y)^2]$$

$$= \frac{1}{2} * [2 * (f(x) - y) * \frac{\partial}{\partial w}(f(x) - y)]$$

$$= (f(x) - y) * \frac{\partial}{\partial w}(f(x))$$

$$= (f(x) - y) * \frac{\partial}{\partial w}\left(\frac{1}{1+e^{-(wx+b)}}\right)$$

$$= (f(x) - y) * f(x) * (1 - f(x)) * x$$

$$\frac{\partial}{\partial w}\left(\frac{1}{1+e^{-(wx+b)}}\right)$$

$$= \frac{-1}{(1+e^{-(wx+b)})^2}\frac{\partial}{\partial w}(e^{-(wx+b)}))$$

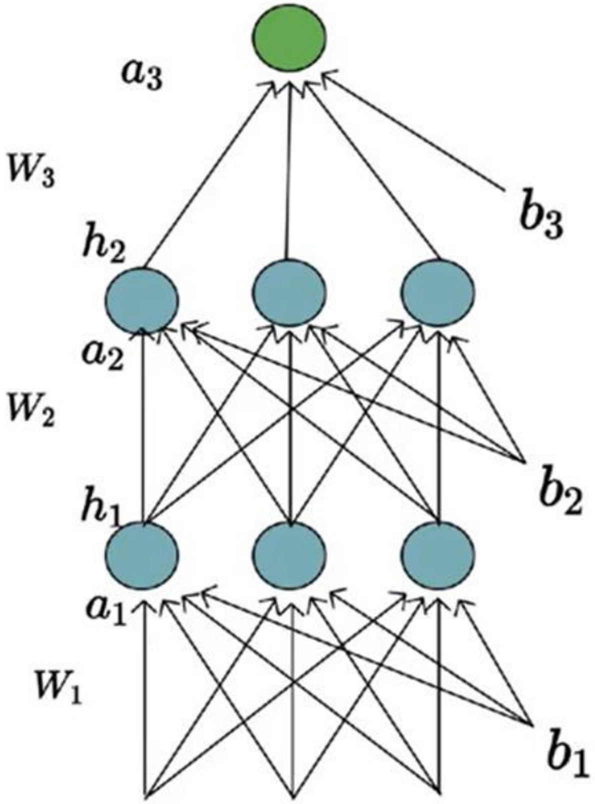
$$= \frac{-1}{(1+e^{-(wx+b)})^2} * (e^{-(wx+b)})\frac{\partial}{\partial w}(-(wx+b)))$$

$$= \frac{-1}{(1+e^{-(wx+b)})} * \frac{e^{-(wx+b)}}{(1+e^{-(wx+b)})} * (-x)$$

$$= \frac{1}{(1+e^{-(wx+b)})} * \frac{e^{-(wx+b)}}{(1+e^{-(wx+b)})} * (x)$$

$$= f(x) * (1 - f(x)) * x$$

Partial derivative , Gradient



$$\Delta w_t = \frac{\partial \mathscr{L}(w,b)}{\partial w}$$

$$\begin{bmatrix} \frac{\partial \mathscr{L}(\theta)}{\partial W_{111}} & \cdots & \frac{\partial \mathscr{L}(\theta)}{\partial W_{11n}} & \frac{\partial \mathscr{L}(\theta)}{\partial W_{211}} & \cdots & \frac{\partial \mathscr{L}(\theta)}{\partial W_{21n}} & \cdots & \frac{\partial \mathscr{L}(\theta)}{\partial W_{L,11}} & \cdots & \frac{\partial \mathscr{L}(\theta)}{\partial W_{L,1k}} & \frac{\partial \mathscr{L}(\theta)}{\partial W_{L,1k}} & \frac{\partial \mathscr{L}(\theta)}{\partial b_{11}} & \cdots & \frac{\partial \mathscr{L}(\theta)}{\partial b_{L1}} \\ \frac{\partial \mathscr{L}(\theta)}{\partial W_{121}} & \cdots & \frac{\partial \mathscr{L}(\theta)}{\partial W_{12n}} & \frac{\partial \mathscr{L}(\theta)}{\partial W_{221}} & \cdots & \frac{\partial \mathscr{L}(\theta)}{\partial W_{22n}} & \cdots & \frac{\partial \mathscr{L}(\theta)}{\partial W_{L,21}} & \cdots & \frac{\partial \mathscr{L}(\theta)}{\partial W_{L,2k}} & \frac{\partial \mathscr{L}(\theta)}{\partial W_{L,2k}} & \frac{\partial \mathscr{L}(\theta)}{\partial b_{12}} & \cdots & \frac{\partial \mathscr{L}(\theta)}{\partial b_{L2}} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{\partial \mathscr{L}(\theta)}{\partial W_{1n1}} & \cdots & \frac{\partial \mathscr{L}(\theta)}{\partial W_{1nn}} & \frac{\partial \mathscr{L}(\theta)}{\partial W_{2n1}} & \cdots & \frac{\partial \mathscr{L}(\theta)}{\partial W_{2nn}} & \cdots & \frac{\partial \mathscr{L}(\theta)}{\partial W_{L,n1}} & \cdots & \frac{\partial \mathscr{L}(\theta)}{\partial W_{L,nk}} & \frac{\partial \mathscr{L}(\theta)}{\partial W_{L,nk}} & \frac{\partial \mathscr{L}(\theta)}{\partial b_{1n}} & \cdots & \frac{\partial \mathscr{L}(\theta)}{\partial b_{Lk}} \end{bmatrix}$$

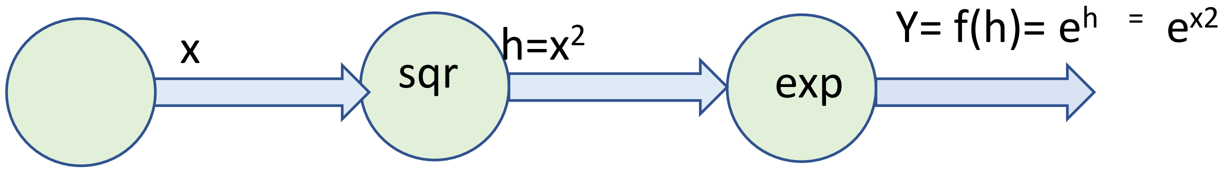
The partial derivative notation is used **to specify the derivative of a function of more than one variable with respect to one of its variables**.

The gradient of a function f, denoted as ∇f , is **the collection of all its partial derivatives into a vector**.

Calculus basics - Chain rule

$$\frac{de^x}{dx} = e^x \qquad \frac{dx^2}{dx} = 2x \qquad \frac{d(1/x)}{dx} = -\frac{1}{x^2}$$

$$\frac{de^{x^2}}{dx} = \frac{de^{x^2}}{dx^2} \cdot \frac{dx^2}{dx} = \frac{de^z}{dz} \cdot \frac{dx^2}{dx} = (e^z).(2x) = (e^{x^2}).(2x) = 2xe^{x^2}$$



h=f(x)

Y=f(h)= e^h

$$\frac{dy}{dx} = \frac{dy}{dh}\frac{dh}{dx} = \frac{de^h}{dh}\frac{dx^2}{dx} = e^h\, 2x = 2x\, e^{x2}$$

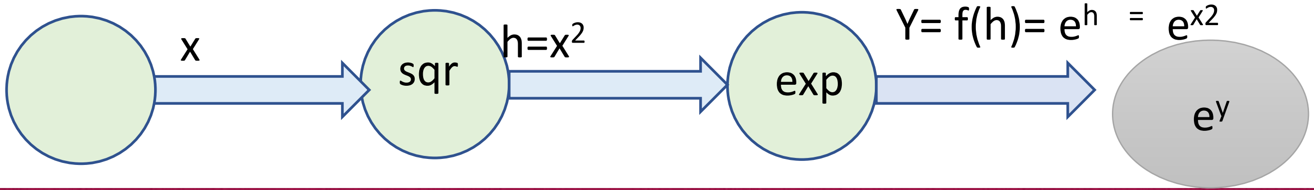
Chain rule

$$\boxed{\frac{de^{e^{x^2}}}{dx} = \frac{de^{e^{x^2}}}{de^{x^2}} \cdot \frac{de^{x^2}}{dx^2} \cdot \frac{dx^2}{dx}}$$

$$\frac{de^x}{dx} = e^x \qquad \frac{dx^2}{dx} = 2x \qquad \frac{d(1/x)}{dx} = -\frac{1}{x^2}$$

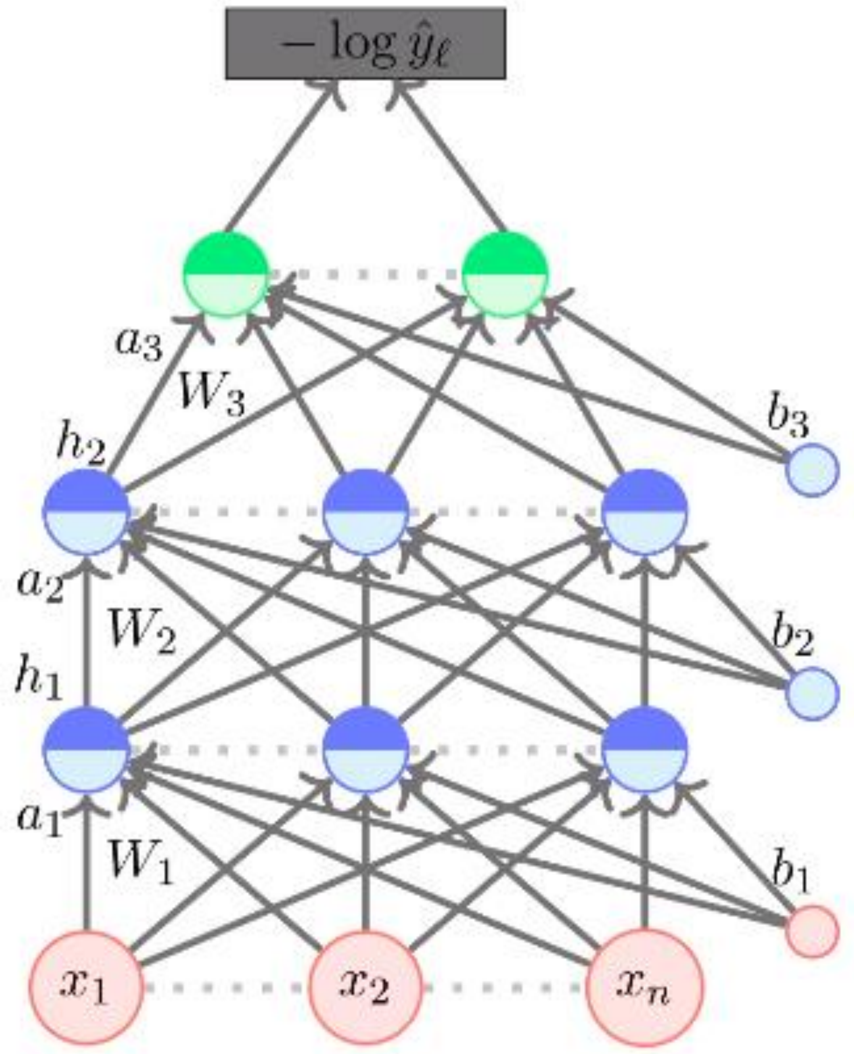
$$\frac{de^{x^2}}{dx} = \frac{de^{x^2}}{dx^2} \cdot \frac{dx^2}{dx} = \frac{de^z}{dz} \cdot \frac{dx^2}{dx} = (e^z).(2x) = (e^{x^2}).(2x) = 2xe^{x^2}$$

$$\frac{de^{e^{x^2}}}{dx} = \frac{de^{e^{x^2}}}{de^{x^2}} \cdot \frac{de^{x^2}}{dx} = \frac{de^z}{dz} \cdot \frac{de^{x^2}}{dx} = (e^z).(2xe^{x^2}) = (e^{e^{x^2}}).(2xe^{x^2}) = 2xe^{x^2}e^{e^{x^2}}$$



Backpropagation

$$\underbrace{\frac{\partial \mathscr{L}(\theta)}{\partial W_{111}}}_{\text{Talk to the weight directly}} = \underbrace{\frac{\partial \mathscr{L}(\theta)}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial a_3}}_{\text{Talk to the output layer}} \underbrace{\frac{\partial a_3}{\partial h_2} \frac{\partial h_2}{\partial a_2}}_{\text{Talk to the previous hidden layer}} \underbrace{\frac{\partial a_2}{\partial h_1} \frac{\partial h_1}{\partial a_1}}_{\text{Talk to the previous hidden layer}} \underbrace{\frac{\partial a_1}{\partial W_{111}}}_{\text{and now talk to the weights}}$$



$$\Delta w_t = \frac{\partial \mathscr{L}(w,b)}{\partial w}$$

The partial derivative notation is used **to specify the derivative of a function of more than one variable with respect to one of its variables**.

The gradient of a function f, denoted as ∇f , is **the collection of all its partial derivatives into a vector**.

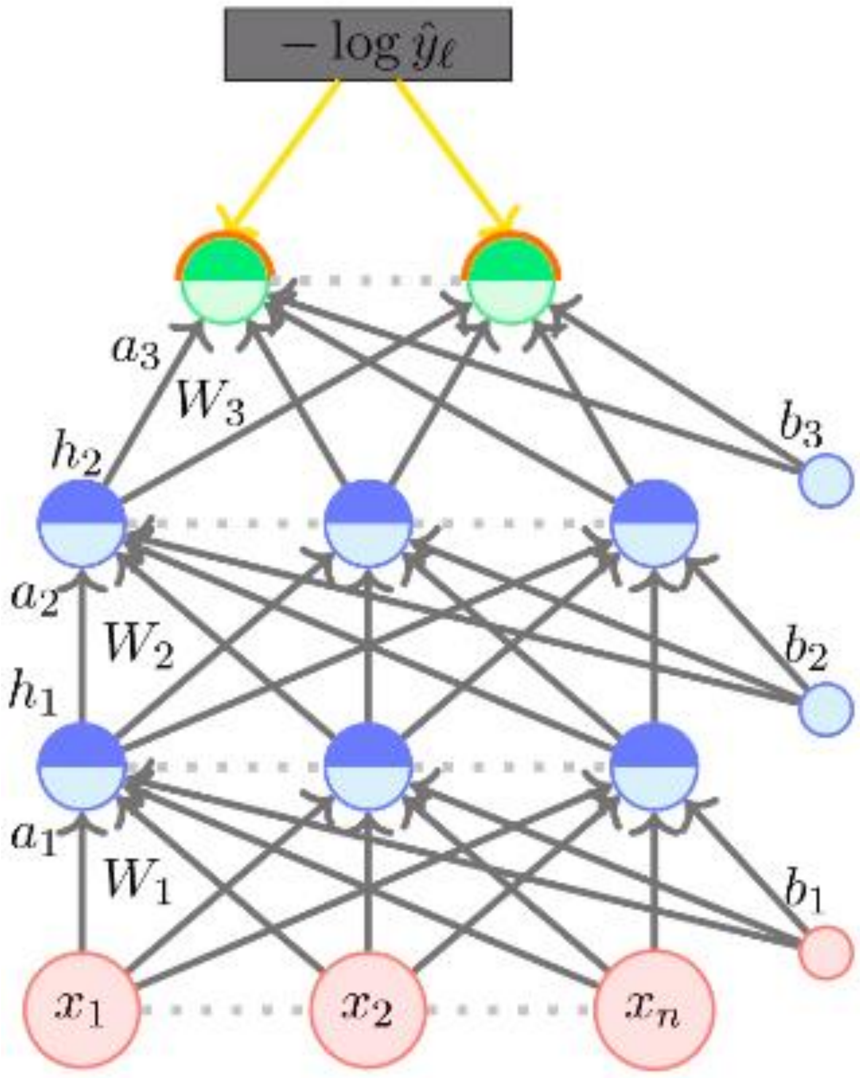
Backpropagation

$$\underbrace{\frac{\partial \mathscr{L}(\theta)}{\partial W_{111}}}_{\text{Talk to the weight directly}} = \underbrace{\frac{\partial \mathscr{L}(\theta)}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial a_3}}_{\text{Talk to the output layer}} \underbrace{\frac{\partial a_3}{\partial h_2} \frac{\partial h_2}{\partial a_2}}_{\text{Talk to the previous hidden layer}} \underbrace{\frac{\partial a_2}{\partial h_1} \frac{\partial h_1}{\partial a_1}}_{\text{Talk to the previous hidden layer}} \underbrace{\frac{\partial a_1}{\partial W_{111}}}_{\text{and now talk to the weights}}$$

$$\Delta w_t = \frac{\partial \mathscr{L}(w,b)}{\partial w}$$

The partial derivative notation is used **to specify the derivative of a function of more than one variable with respect to one of its variables**.

The gradient of a function f, denoted as ∇f , is **the collection of all its partial derivatives into a vector**.



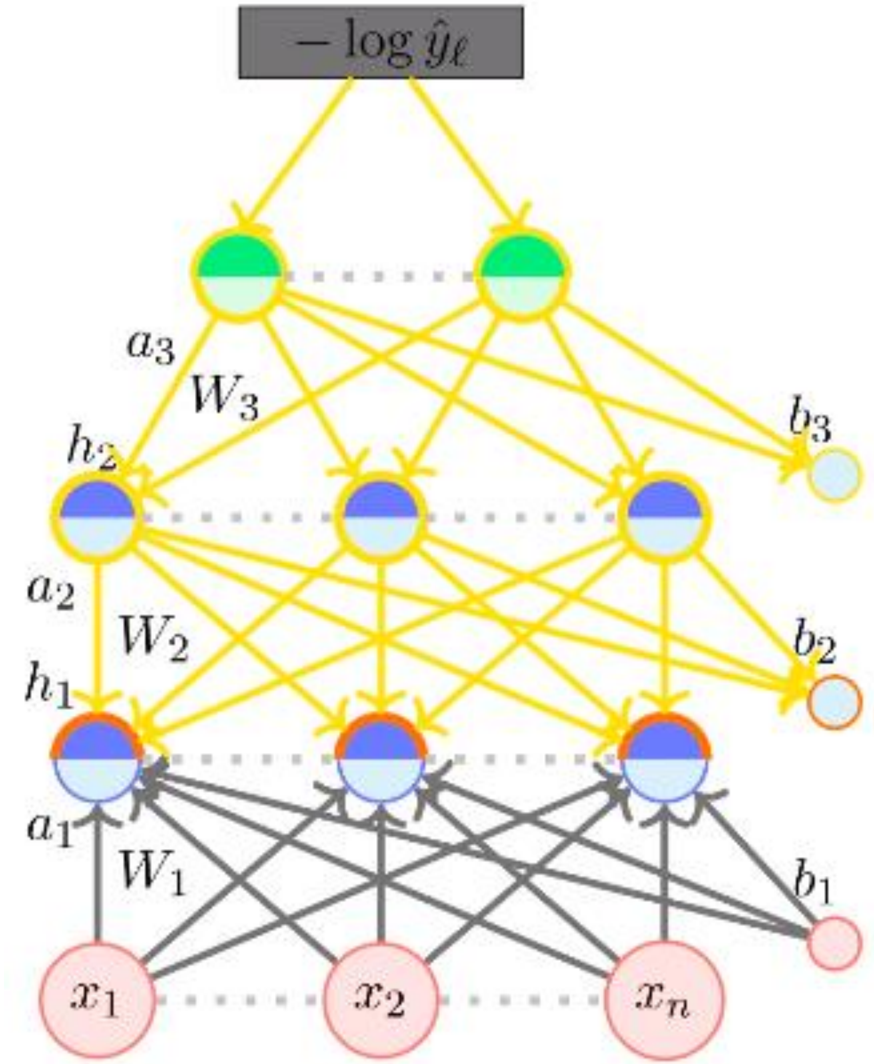
Backpropagation

$$\underbrace{\frac{\partial \mathscr{L}(\theta)}{\partial W_{111}}}_{\text{Talk to the weight directly}} = \underbrace{\frac{\partial \mathscr{L}(\theta)}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial a_3}}_{\text{Talk to the output layer}} \underbrace{\frac{\partial a_3}{\partial h_2} \frac{\partial h_2}{\partial a_2}}_{\text{Talk to the previous hidden layer}} \underbrace{\frac{\partial a_2}{\partial h_1} \frac{\partial h_1}{\partial a_1}}_{\text{Talk to the previous hidden layer}} \underbrace{\frac{\partial a_1}{\partial W_{111}}}_{\text{and now talk to the weights}}$$

$$\Delta w_t = \frac{\partial \mathscr{L}(w,b)}{\partial w}$$

The partial derivative notation is used **to specify the derivative of a function of more than one variable with respect to one of its variables**.

The gradient of a function f, denoted as ∇f , is **the collection of all its partial derivatives into a vector**.



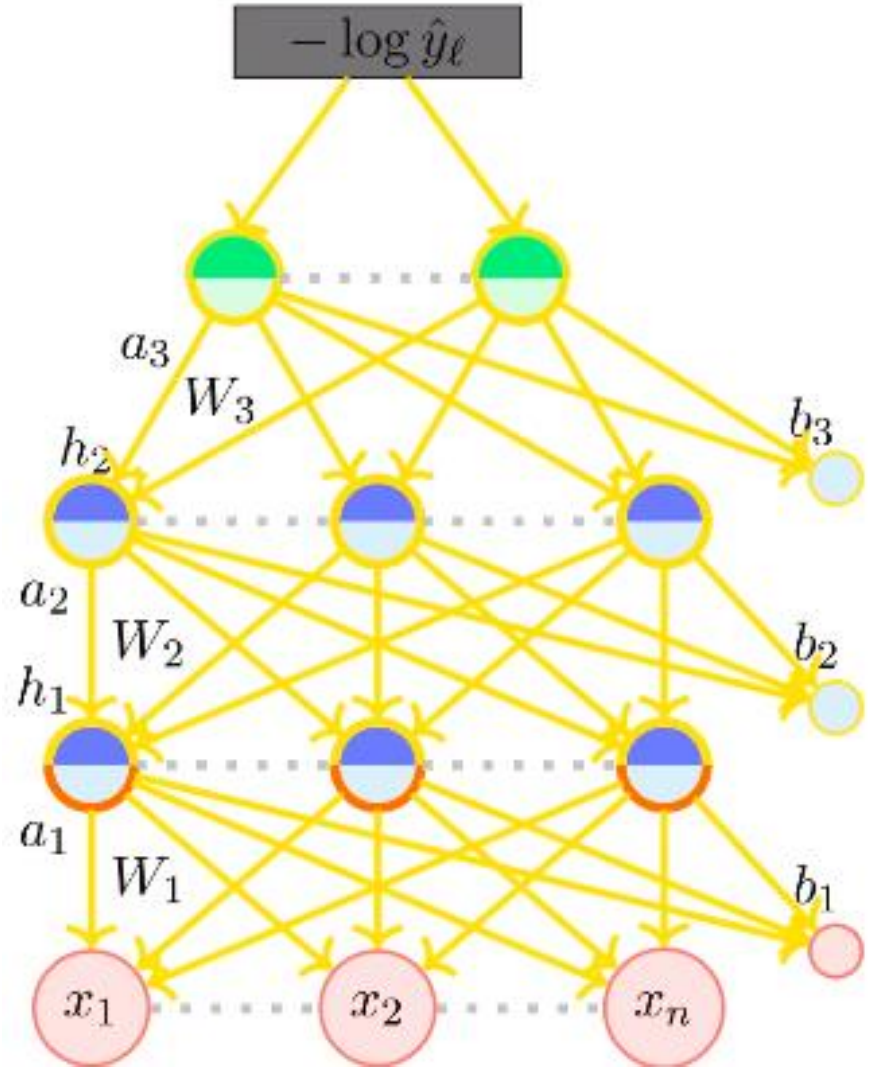
Backpropagation

$$\underbrace{\frac{\partial \mathscr{L}(\theta)}{\partial W_{111}}}_{\text{Talk to the weight directly}} = \underbrace{\frac{\partial \mathscr{L}(\theta)}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial a_3}}_{\text{Talk to the output layer}} \underbrace{\frac{\partial a_3}{\partial h_2} \frac{\partial h_2}{\partial a_2}}_{\text{Talk to the previous hidden layer}} \underbrace{\frac{\partial a_2}{\partial h_1} \frac{\partial h_1}{\partial a_1}}_{\text{Talk to the previous hidden layer}} \underbrace{\frac{\partial a_1}{\partial W_{111}}}_{\text{and now talk to the weights}}$$

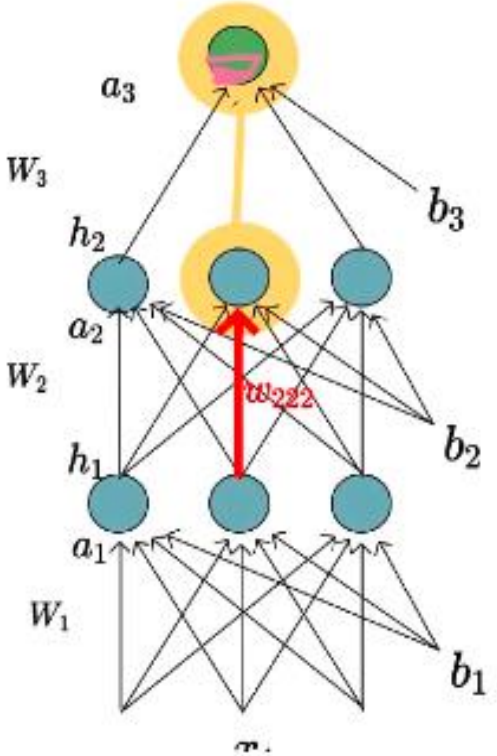
$$\Delta w_t = \frac{\partial \mathscr{L}(w,b)}{\partial w}$$

The partial derivative notation is used **to specify the derivative of a function of more than one variable with respect to one of its variables**.

The gradient of a function f, denoted as ∇f , is **the collection of all its partial derivatives into a vector**.



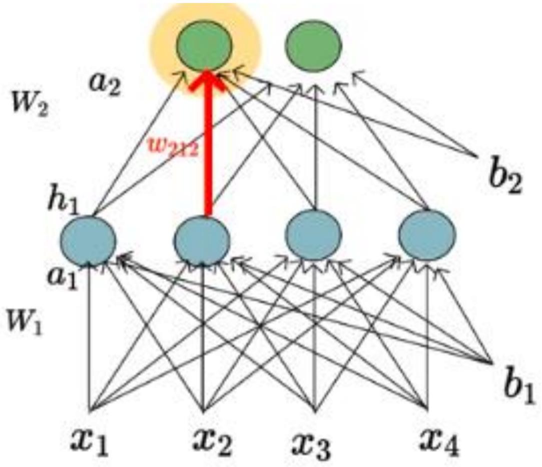
Learning algorithm- Back propagation



- Let us focus on the highlighted weight (w_{222})
- To learn this weight, we have to compute partial derivative w.r.t loss function

$$(w_{222})_{t+1} = (w_{222})_t - \eta * \left(\frac{\partial L}{\partial w_{222}}\right)$$

$$\begin{aligned}
\frac{\partial L}{\partial w_{222}} &= \left(\frac{\partial L}{\partial a_{22}}\right).\left(\frac{\partial a_{22}}{\partial w_{222}}\right) \\
&= \left(\frac{\partial L}{\partial h_{22}}\right).\left(\frac{\partial h_{22}}{\partial a_{22}}\right).\left(\frac{\partial a_{22}}{\partial w_{222}}\right) \\
&= \left(\frac{\partial L}{\partial a_{31}}\right).\left(\frac{\partial a_{31}}{\partial h_{22}}\right).\left(\frac{\partial h_{22}}{\partial a_{22}}\right).\left(\frac{\partial a_{22}}{\partial w_{222}}\right) \\
&= \left(\frac{\partial L}{\partial \hat{y}}\right).\left(\frac{\partial \hat{y}}{\partial a_{31}}\right).\left(\frac{\partial a_{31}}{\partial h_{22}}\right).\left(\frac{\partial h_{22}}{\partial a_{22}}\right).\left(\frac{\partial a_{22}}{\partial w_{222}}\right)
\end{aligned}$$



$b = [\ 0 \quad 0\ ]$

$$W_1 = \begin{bmatrix} 0.1 & 0.3 & 0.8 & -0.4 \\ -0.3 & -0.2 & 0.5 & 0.5 \\ -0.3 & 0 & 0.5 & 0.4 \\ 0.2 & 0.5 & -0.9 & 0.7 \end{bmatrix}$$

$$W_2 = \begin{bmatrix} 0.5 & 0.8 & 0.2 & 0.4 \\ 0.5 & 0.2 & 0.3 & -0.5 \end{bmatrix}$$

$x = [\ 2 \quad 5 \quad 3 \quad 3\ ]$ $\qquad$ $y = [\ 1 \quad 0\ ]$

$$\frac{\partial L}{\partial w_{212}} = \left(\frac{\partial L}{\partial a_{21}}\right).\left(\frac{\partial a_{21}}{\partial w_{212}}\right) = \left(\frac{\partial L}{\partial \hat{y}_1}\right).\left(\frac{\partial \hat{y}_1}{\partial a_{21}}\right).\left(\frac{\partial a_{21}}{\partial w_{212}}\right)$$

$$\frac{\partial L}{\partial \hat{y}_1} = -2(y_1 - \hat{y}_1) \qquad = -0.46$$

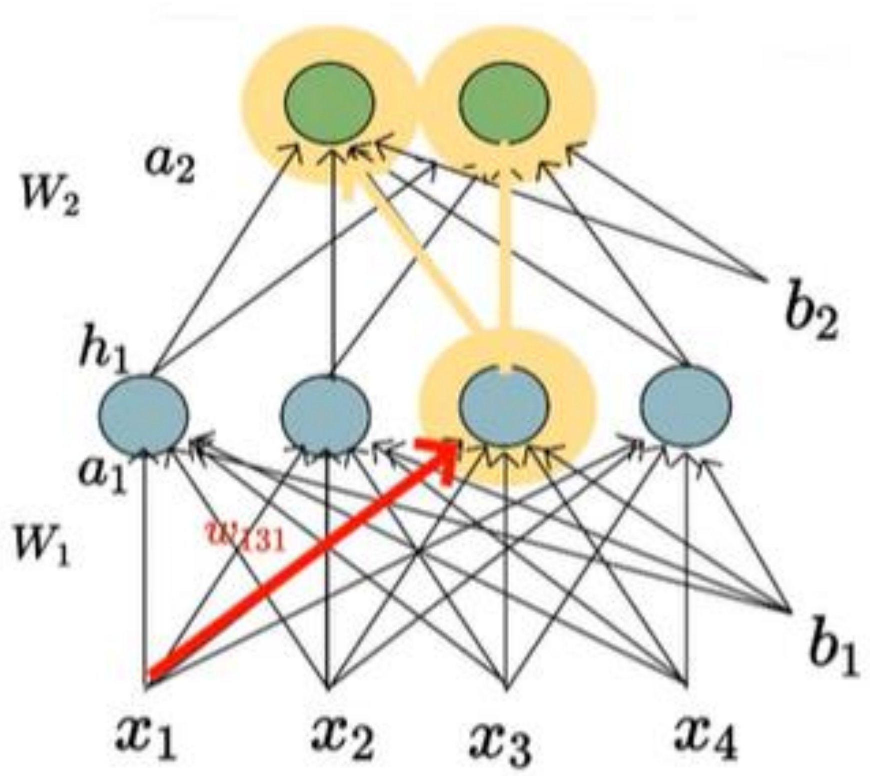
$$\frac{\partial \hat{y}_1}{\partial a_{21}} = \hat{y}_1 * (1 - \hat{y}_1)(-a_{22}) \quad = -0.079$$

$$\frac{\partial a_{21}}{\partial w_{212}} = h_{12} \qquad = 0.80$$

$$\frac{\partial L}{\partial w_{212}} = (-2(y_1 - \hat{y}_1)) * (\hat{y}_1(1 - \hat{y}_1)(-a_{22})) * (h_{12})$$

$$= (-0.46) * (-0.079) * (0.80) = -0.029$$

Multiple paths



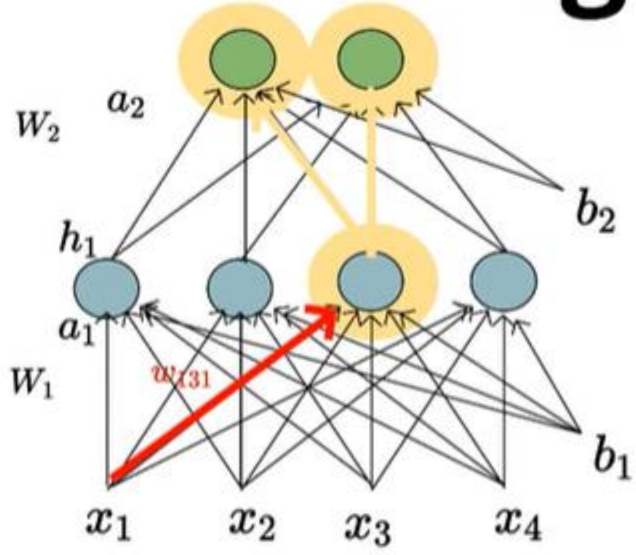
- There are 2 different paths connecting w131 with loss function
- Consider all those paths through which gradient is flowing back
- Sum up the gradients along all these paths(2 here)
- ie Apply independent chain rules to all those multiple paths and sum up the derivatives across all these paths and get the total derivative of the loss function w.r.t w131

$$\frac{\partial a_{21}}{h_{13}} = w_{211}h_{11} + w_{212}h_{12} + w_{213}h_{13} + w_{214}h_{14}$$

$$= w_{213}.$$

Can we see one more example ?

Learning Algorithm



$$x = [\ 2 \quad 5 \quad 3 \quad 3\ ] \qquad y = [\ 1 \quad 0\ ]$$

$$\frac{\partial L}{\partial w_{131}} = \left(\frac{\partial L}{\partial \hat{y}_1}.\frac{\partial \hat{y}_1}{\partial a_{21}}.\frac{\partial a_{21}}{\partial h_{13}} + \frac{\partial L}{\partial \hat{y}_2}.\frac{\partial \hat{y}_2}{\partial a_{22}}.\frac{\partial a_{22}}{\partial h_{13}}\right).\left(\frac{\partial h_{13}}{\partial a_{13}}\right).\left(\frac{\partial a_{13}}{\partial w_{131}}\right)$$

$$\frac{\partial L}{\partial \hat{y}_1} = -2(y_1 - \hat{y}_1) \quad = -0.46 \qquad \frac{\partial L}{\partial \hat{y}_2} = -2(y_2 - \hat{y}_2) \quad = 0.46$$

$$\frac{\partial \hat{y}_1}{\partial a_{21}} = \hat{y}_1 * (1 - \hat{y}_1) * (-a_{22}) = -0.079 \qquad \frac{\partial \hat{y}_2}{\partial a_{22}} = \hat{y}_2 * (1 - \hat{y}_2)(-a_{21}) = -0.293$$

$$\frac{\partial a_{21}}{\partial h_{13}} = w_{213} \quad = 0.20 \qquad \frac{\partial a_{22}}{\partial h_{13}} = w_{223} \quad = 0.30$$

$$\frac{\partial h_{13}}{\partial a_{13}} = h_{13} * (1 - h_{13}) = 0.0979 \qquad \frac{\partial a_{13}}{\partial w_{131}} = x_1 \quad = 2$$

$$b = [\ 0 \quad 0\ ]$$

$$W_1 = \begin{bmatrix} 0.1 & 0.3 & 0.8 & -0.4 \\ -0.3 & -0.2 & 0.5 & 0.5 \\ -0.3 & 0 & 0.5 & 0.4 \\ 0.2 & 0.5 & -0.9 & 0.7 \end{bmatrix}$$

$$W_2 = \begin{bmatrix} 0.5 & 0.8 & 0.2 & 0.4 \\ 0.5 & 0.2 & 0.3 & -0.5 \end{bmatrix}$$

$$\frac{\partial L}{\partial w_{131}} = (-2(y_1 - \hat{y}_1) * \hat{y}_1(1 - \hat{y}_1) * w_{213} + -2(y_2 - \hat{y}_2) * \hat{y}_2(1 - \hat{y}_2) * w_{223}) * h_{13}(1 - h_{13}) * x_1$$

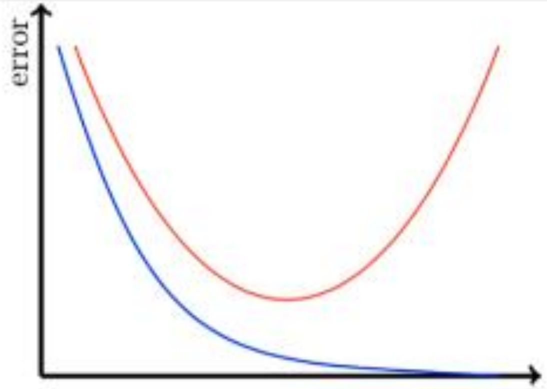
Hyper parameter tuning

Network Architectures

- Number of layers
- Number of neurons

Algorithms

- Vanilla/Momentum /Nesterov GD
- AdaGrad
- RMSProp
- Adam



Activation Functions

- tanh (RNNs)
- relu (CNNs, DNNs)
- leaky relu (CNNs)

Strategies

- Batch
- Mini-Batch (32, 64, 128)
- Stochastic
- Learning rate schedule

Regularization

- L2
- Early stopping
- Dataset augmentation
- Drop-out
- Batch Normalizat

Initialization Methods

- *Xavier*
- *He*

Namah Shivaya