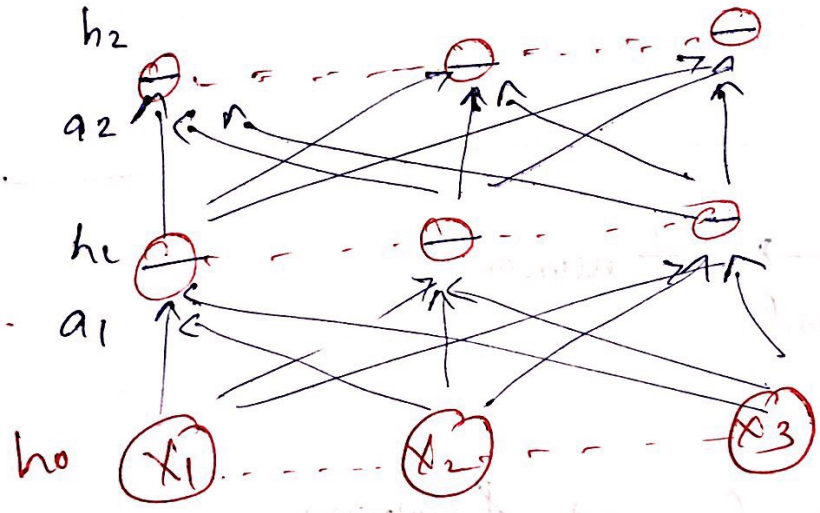
Lecture-4

* Feedforward Neural Networks

& Backpropogation

$h_L = \hat{y} = f(x) \in \mathbb{R}^K$

K neurons with K classes



a_i - preactivation (aggregation)

h_i - activation (non-linearity)

↳ vectors

Pre-activation

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

ex: $a_1 = b_1 + W_1 h_0$

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

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

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

$$h_i(x) = g(a_i(x))$$

$$\begin{bmatrix} h_{11} \\ h_{12} \\ h_{13} \end{bmatrix} = g\left(\begin{bmatrix} a_{11} \\ a_{12} \\ a_{13} \end{bmatrix}\right) = \begin{bmatrix} g(a_{11}) \\ g(a_{12}) \\ g(a_{13}) \end{bmatrix}$$

$$g(a_{13}) = \sigma(a_{13}) = \frac{1}{1+e^{-a_{13}}}$$

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

$$\hat{y}_i = f(x_i) = O\left(W^3 g\left(W^2 g\left(W^1 x + b_1\right) + b_2\right) + b_3\right)$$

(W^1: $\mathbb{R}^n \to (n \times n)(n) + (n)$)

$$\theta = W_1 \ldots W_L, b_1, b_2 \ldots b_L$$

Lecture 4.2

Learn Parameters of Feedforward Neural Networks

Algorithm: Gradient Descent

```
t ← 0;
max_iterations ← 1000;
Initialize θ_0 = [W_0, b_0];
while t++ < max_iterations do
    θ_{t+1} ← θ_t − η ∇θ_t;
end
```

where $\nabla\theta_t = \left[\frac{\partial \mathcal{L}(\theta)}{\partial w_t}, \frac{\partial \mathcal{L}(\theta)}{\partial b_t}\right]^T$

$\theta = [w, b]$

is converted to

$\theta = W_1, W_2, \ldots W_L, b_1, b_2 \ldots b_L$

Lecture 4.3

Output Functions & Loss Functions

$y_i \in \mathbb{R}^*$

Loss Function

$$\mathcal{L}(\theta) = \frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{K} (\hat{y}_{ij} - y_{ij})^2$$

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

$$= W_0 a_L + b_0$$

Entropy - measure of uncertainty

$$H(X) = H(p_1 \cdots p_n)$$

Entropy $$= -\sum_{i=1}^{n} p_i \log_2 p_i$$

Cross Entropy

Softmax Function

$$a_L = [a_{L_1}, a_{L_2}, a_{L_3}]$$

$$a_L = [10, -20, 30]$$

$$\hat{y} = [\hat{y}_1 \ \hat{y}_2 \ \hat{y}_3]$$

$$\hat{y}_1 = \frac{e^{10}}{e^{10} + e^{-20} + e^{30}}$$

Softmax Function $\Rightarrow$

$$a_L = w_L h_{L-1} + b_L$$

$$\hat{y}_j = O(a_L)_j$$

$$= \frac{e^{a_{L,j}}}{\sum_{i=1}^{K} e^{a_{L,i}}}$$

Principle way of computing the distribution ~~Events~~ is Cross Entropy

Cross Entropy

$$\mathcal{L}(\theta)_2 = -\sum_{c=1}^{K} y_c \log \hat{y}_c$$

minimise $\mathcal{L}(\theta) = -\log \hat{y}_L$ → (predicted correct label)

or

maximise $-\mathcal{L}(\theta) = \log \hat{y}_L$

or

$\log \hat{y}_L \rightarrow$ log-likelihood of the data.

	Outputs	
	Real Values	Probability
Output Activation	Linear	Softmax
Loss Function	Squared error	Cross-Entropy

Module 4.4

Backpropogation

$$L(\theta) = -\log \hat{y}_k$$

$W(w_{112})$ — o/p layer — previous hidden layer

$$\frac{\partial L(\theta)}{\partial w_{111}} = \frac{\partial L(\theta)}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial a_3} \frac{\partial a_3}{\partial h_2} \frac{\partial h_2}{\partial a_2}$$

$$\frac{\partial a_2}{\partial h_1} \frac{\partial h_1}{\partial a_1} \frac{\partial a_1}{\partial w_{11}}$$

$\frac{\partial a_2}{\partial h_1}$ — previous hidden layer

$\frac{\partial a_1}{\partial w_{11}}$ — the weights from the ith layer

→ Compute the gradients with respect to the o/p units

→ Compute the gradients with respect to the hidden ~~layer~~ units

→ Compute the gradient w.r.t the weight & bias units

Module 4.5

Compute the gradients with respect to the output units.

$$\frac{\partial \mathcal{L}(\theta)}{\partial a_L}$$

$$\mathcal{L}(\theta) = -\log \hat{y}_l \quad (l = \text{true class label})$$

$$\frac{\partial}{\partial \hat{y}_i}(\mathcal{L}(\theta)) = \frac{\partial}{\partial \hat{y}_i}(-\log \hat{y}_l)$$

↳ no. of classes

$$= \frac{-1}{\hat{y}_l} \quad \text{if } i = l$$

$$= 0, \quad \text{otherwise}$$

More compactly

$$\frac{\partial}{\partial \hat{y}_i}(\mathcal{L}(\theta)) = -\frac{\mathbb{I}_{(i=l)}}{\hat{y}_l}$$

$$\frac{\partial}{\partial \hat{y}_i}(\mathcal{L}(o)) = -\frac{\mathbb{I}(\ell = i)}{\hat{y}_\ell}$$

Gradient with respect to $\hat{y}$

$$\nabla_{\hat{y}} \mathcal{L}(o) = \begin{bmatrix} \frac{\partial \mathcal{L}(o)}{\partial \hat{y}_1} \\ \vdots \\ \frac{\partial \mathcal{L}(o)}{\partial \hat{y}_k} \end{bmatrix} = \frac{-1}{\hat{y}_\ell} \begin{bmatrix} \mathbb{I}_{\ell=1} \\ \vdots \\ \mathbb{I}_{\ell=k} \end{bmatrix}$$

$\rightarrow$ one hot vector

$$= \frac{-1}{\hat{y}_\ell} e(\ell)$$

$$\frac{\partial \mathcal{L}(o)}{\partial a_{Li}} = \frac{\partial(-\log \hat{y}_\ell)}{\partial a_{Li}}$$

$$= \frac{\partial(-\log \hat{y}_\ell)}{\partial \hat{y}_\ell} \frac{\partial \hat{y}_\ell}{\partial a_{Li}}$$

$$\hat{y}_\ell = \frac{\exp(a_{L\ell})}{\sum_i \exp(a_{Li})}$$

$$\frac{\partial}{\partial a_{Li}}(-\log \hat{y}_\ell) = \frac{-1}{\hat{y}_\ell}\frac{\partial}{\partial a_{Li}}(\hat{y}_\ell)$$

$$= \frac{-1}{\hat{y}_\ell}\frac{\partial}{\partial a_{Li}}\text{softmax}(a_L)_\ell$$

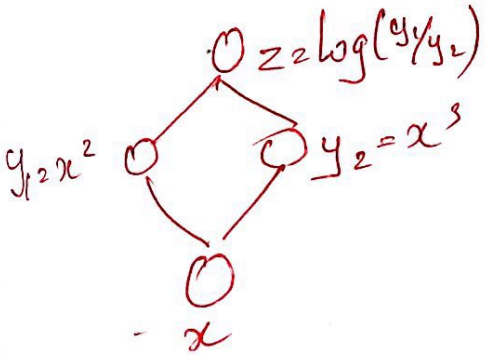
$$\frac{\partial \frac{u(x)}{v(x)}}{\partial x} = \frac{\partial u(x)}{\partial x}\frac{1}{v(x)} - \frac{u(x)}{v(x)^2}\frac{\partial v(x)}{\partial x}$$

$$= \frac{-1}{\hat{y}_\ell}\cdot\frac{\partial}{\partial a_{Li}}\frac{\exp(a_L)_\ell}{\sum_{i+}\exp(a_L)_\ell}$$

$$= \frac{-1}{\hat{y}_\ell}\left(\frac{\partial}{\partial a_{Li}}\exp(a_L)_\ell - \right)$$

expand

Module 4.6 Backpropogation : Computing Gradients with respect to Hidden Layers



$$\frac{dz}{dx} = \sum_{i=1}^{2}\frac{dz}{dy_i}\frac{dy_i}{dx}$$

$$a_{i+1} = W_{i+1}h_{ij} + b_{i+1}$$

$$(W_{i+1,:j})^T \nabla_{a_{i+1}} \mathcal{L}(\theta) =$$

$$\sum_{m=1}^{k} \frac{\partial \mathcal{L}(\theta)}{\partial a_{i+1,m}} W_{i+1,m,j}$$

$$\frac{\partial \mathcal{L}(\theta)}{\partial h_{ij}} = (W_{i+1,:j})^T \nabla_{a_{i+1}} \mathcal{L}(\theta)$$

$$\nabla_{h_i} \mathcal{L}(\theta) = \begin{bmatrix} \frac{\partial \mathcal{L}(\theta)}{\partial h_{i1}} \\ \\ \frac{\partial \mathcal{L}(\theta)}{\partial h_{in}} \end{bmatrix} =$$

Module 4.7 Computing Gradients

Backpropogation with respect Parameters

$$a_k = b_k + W_k h_{k-1}$$

$$\nabla_{W_k} \mathcal{L}(\theta) \quad \frac{\partial \mathcal{L}(\theta)}{W_{kij}}$$

$$\frac{\partial a_{ki}}{\partial w_{kij}} = h_{k-1,j}$$

$$\frac{\partial L(\theta)}{\partial w_{kij}} = \frac{\partial L(\theta)}{\partial a_{ki}} \frac{\partial a_{ki}}{\partial w_{k,i,j}}$$

$$= \frac{\partial L(\theta)}{\partial a_{ki}} h_{k-1,j}$$

$$\nabla w_k L(\theta) = \begin{bmatrix} \frac{\partial L(\theta)}{\partial w_{k00}} & & \frac{\partial L(\theta)}{\partial w_{k,n-1,n-1}} \end{bmatrix}$$

4.8

Algorithm

[forward propagation
backward propagation

Forward propagation

for $k = 1$ to $L-1$ do

$a_k = b_k + W_k h_{k-1}$;

$h_k = g(a_k)$

end

$a_L = b_L + W_L h_{L-1}$;

$\hat{y} = O(a_L)$;

Backward

$\nabla_{a_L} L(o) = -(e(y) - f(x))$;

for $k = L$ to 1 do

// w.r.t parameters

$\nabla_{W_k} L(o) = \nabla_{a_k} L(o)\, h_{k-1}^T$;

$\nabla_{b_k} L(o) = \nabla_{a_k} L(o)$

// w.r.t to layer below

$\nabla_{h_{k-1}} L(o) = W_k^T (\nabla_{a_k} L(o))$

// w.r.t layer below pre activation

$\nabla_{a_{k-1}} L(o) = \nabla_{h_{k-1}} L(o) \odot [\ldots, g'(a_{k-1,j}), \ldots]$

end

A.9 Derivative of the activation function

$$g(z) = \sigma(z)$$

$$= \frac{1}{1+e^{-z}}$$

$$g'(z) = -1 \left(\frac{1}{(1+e^{-z})^2}\right) \frac{d}{dz}(1+e^{-z})$$

$$= (-1) \frac{1}{(1+e^{-z})^2} (-e^{-z})$$

$$= \frac{1}{1+e^{-z}} \left(\frac{1+e^{-z}-1}{1+e^{-z}}\right)$$

$$= g(z)(1-g(z))$$

tanh

$$g(z) = \tanh(z) = \frac{e^{z}-e^{-z}}{e^{z}+e^{-z}}$$

$$g'(z) = 1-(g(z))^2$$

4.10

Information content,

Entropy,

Cross Entropy

$X \rightarrow$ R B W

$[0.25 \quad 0.35 \quad 0.4]$ P

$p_1 \quad p_2 \quad p_3$

estimate $[0.34 \quad 0.45 \quad 0.2]$ Q

$q_1 \quad q_2 \quad q_3$

One way Square error $\sum_{i=1}^{3} (p_i - q_i)^2$

$X \rightarrow$ A B C D

P $[0 \quad 1 \quad 0 \quad 0]$

All the probability mass is focused on one item.

Prediction

Q $[0.3 \quad 0.4 \quad 0.2 \quad 0.1]$

$\sum_{i=1}^{4} (p_i - q_i)^2$

Classification

A B C D

X $[0 \quad 1 \quad 0 \quad 0]$

prediction

$[0.25 \quad 0.45 \quad 0.2 \quad 0.1]$

$L(0) = p \| q$

Square Error $\Rightarrow \sum_{i=1}^{4} (p_i - q_i)^2$

Expectation

X	A	B	C	D
P(X=?)	0.3	0.4	0.2	0.1
V(X=?)	10k	5k	10k	-30k (gain)

Expected Reward →

$$\text{Expected Gain} = \sum_{i \in (A,B,C,D)} P(X=i)\ V(X=i)$$

Information content

X = A B C D

Information gain is high, when the event which happens is a surprising event

$$IC(A) \propto \frac{1}{P(A)}$$

(A → Event; P(A) → Event)

$$f(P(A))$$

A B

$$IC(X \wedge Y) = IC(X) + IC(Y)$$

independent events

X	Y
A	ON
B	OFF
C	ON
D	OFF

$$f(P(X \cap Y)) = f(P(X)) + f(P(Y))$$

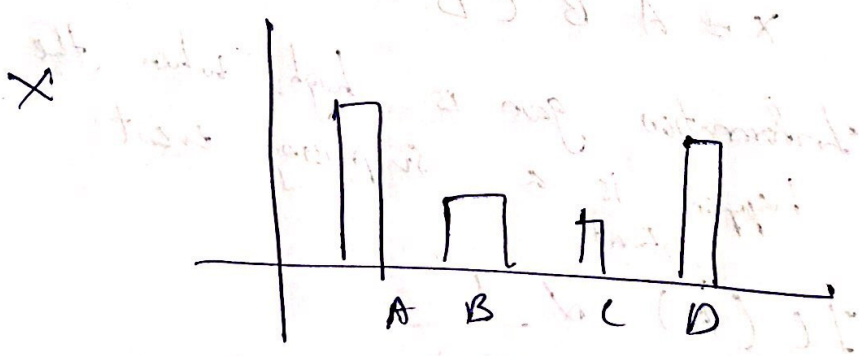
~~$f(P(X))$~~

$$f(P(X) \cdot P(Y)) = f(P(X)) + f(P(Y))$$

$$f(a \cdot b) = f(a) + f(b)$$

$$I(X=A) = \log\left(\frac{1}{P(X=A)}\right)$$

$$I(X=A) = -\log P(X=A)$$



Expectation

$$-\sum_{i \in (A,B,C,D} P(X=i) \log(P(X=i))$$

A B C D

$\times$ (P(X=A) P(X=B) P(X=C) P(X=D))

↓ ↓ ↓ ↓

I(X=A) I(X=B) I(X=C) I(X=D)

X A B C D

A	0	0	1/4
B	0	1	1/4
C	1	0	1/4
D	1	1	1/4

$I(C(this)) = -\log_2(1/4)$

$= -\log_2(2^{-2})$

$= 2$ no. of bits.

X : A B C D E F G H

000	$-\log_2(1/8)$
001	
010	$-\log_2(2^{-3})$
100	
101	3
110	
111	

A	1/2	1
B	1/4	2
C	1/8	3
D	1/8	3

	A	B	C	D
P	1/2	1/4	1/8	1/8]
	1	2	3	3

Equation

$$-\sum_{i \in A,B,C,D} P(X=i) \log_2(P(X=i))$$

Entropy = 1.75

A B C D

P $[p_1\ p_2\ p_3\ p_4]$ - true

Q $[q_1\ q_2\ q_3\ q_4]$ - estimation

$-\log q_i$ bit for the i^{th} message

P $[p_1\ p_2\ p_3\ p_4]$

Q $-\log q_1\ \ -\log q_2\ \ -\log q_3\ \ -\log q_4$

How to compute expectation.

$$\mathcal{L}(p,q) = -\sum p_i \log q_i$$ (minimum value when $p = q$)

P We want Q such that $-\sum p_i \log q_i$

$\Downarrow$

minimise $-\sum p_i \log q_i$ w.r.t q_i

$\min_x x^2$ $\quad$ $\min_{x,y} x^2 + y^2$

$x = 0$

$$\frac{\partial}{q_2}\left(p_1 \log q_1 + p_2 \log q_2 + \cdots p_k \log q_k\right)$$

$$\underset{q_i}{\text{minimize}} \quad -\sum p_i \log q_i$$

$$\text{s.t } \sum q_i = 1$$

$$\text{minimise} \quad -\sum p_i \log q_i + \lambda\left(\sum q_i - 1\right)$$

Lqrasian value.

↓

↳ if violent this will be added

↳ if not correct, then the value will be zero.

$$-\frac{p_2}{q_2} + \lambda_1 = 0$$

$$p_2 = \lambda q_2$$

$$q_i = q_i$$

will be minimised of