Introduction to Deep Learning

Chris Arnold, Cardiff University, July 2021

Logistic Regression

$$\hat{y} = P(y=1 \mid x)$$

Parameters: $w \in \mathbb{R}^{n_x}$, $b \in \mathbb{R}$

Function $\hat{y} = \sigma(w^T x + b)$

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

Loss function

goal: how far are we off?

$$\mathcal{L}(\hat{y}, y) = -\left[y \log \hat{y} + (1-y) \log(1-\hat{y}) \right]$$

if $y = 1$:

$\mathcal{L}(\hat{y}, y) = -\log \hat{y}$ $\Big\{$
- as large as possible
- but not > 1

if $y = 0$:

$$\mathcal{L}(\hat{y}, y) = -\log(1 - \hat{y})$$

- as small as possible
- but not < 0

goal: minimise

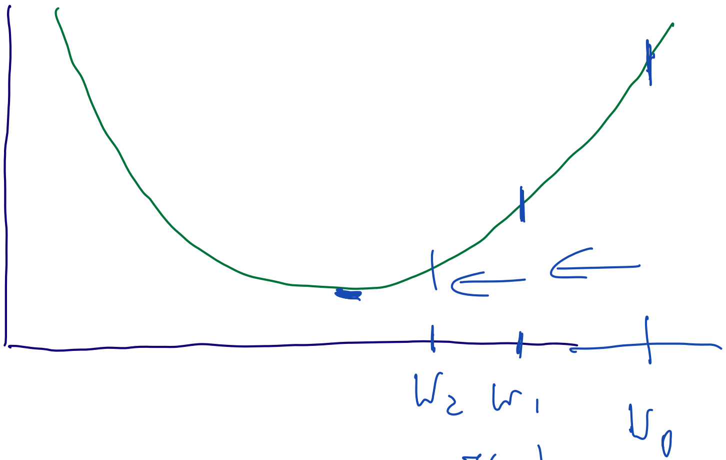
Cost function

$$J(w, b) = \frac{1}{m} \sum_{i=1}^{m} \mathcal{L}(\hat{y}^{(i)}, y^{(i)})$$

$$= -\frac{1}{m} \sum_{i=1}^{m} \left[y^{(i)} \log \hat{y}^{(i)} + (1 - y^{(i)}) \log(1 - \hat{y}^{(i)}) \right]$$

Gradient Descent Intuition

Goal: $\min J(w, b)$



Repeat:

$$w := w - \underbrace{\alpha}_{\text{learning rate}} \underbrace{\frac{d\, J(w)}{dw}}_{\text{der. of } J \text{ twds } w}$$

$$w_1 = w_0 - \alpha \frac{d\, J(w_0)}{dw}$$

$$w_2 = w_1 - \alpha \frac{d\, J(w_1)}{dw}$$

$\vdots$

General Case:

$J(w, b)$:

$$w := w - \alpha \frac{d\, J(w, b)}{dw}$$

$$b := b - \alpha \frac{d\, J(w, b)}{db}$$

Logistic Regression Backpropagation

Goal: update values

Refresh

$$z = w^T x + b$$

$$\hat{y} = a = \sigma(z) = \frac{1}{1+e^{-z}}$$

$$\mathcal{L}(a, y) = -\left[y \log(a) + (1-y)\log(1-a) \right]$$

x_1, w_1, x_2, w_2, b $\rightarrow$ $\boxed{z = w_1 x_1 + w_2 x_2 + b}$ $\rightarrow$ $\boxed{\hat{y} = a = \sigma(z)}$ $\rightarrow$ $\boxed{\mathcal{L}(a, y)}$

"da": $$\frac{d\mathcal{L}(a,y)}{da} = -\frac{y}{a} - \frac{1-y}{1-a} \qquad (1)$$

"dz": $$\frac{da}{dz} = a(1-a) \qquad (2)$$

$$\frac{d\mathcal{L}}{da} \cdot \frac{da}{dz} = \left(-\frac{y}{a} + \frac{1-y}{1-a}\right) \cdot a(1-a) = a - y \qquad (3)$$

$$\text{"}dw_1\text{"}: \frac{d\mathcal{L}}{dw_1} = x_1 \cdot dz$$

$$\text{"}dw_2\text{"}: \frac{d\mathcal{L}}{dw_2} = x_2 \cdot dz$$

$$\text{"}db\text{"}: \frac{d\mathcal{L}}{db} = dz$$

Updating

$$w_1 = w_1 - \alpha\, dw_1$$

$$w_2 = w_2 - \alpha\, dw_2$$

$$b = b - \alpha\, db$$

Neural Networks

Notation

layer

activation a $\qquad a^{[0]} = x$

$$a^{[1]} = \begin{bmatrix} a_1^{[1]} \\ a_2^{[1]} \\ a_3^{[1]} \\ a_4^{[1]} \end{bmatrix}$$

$$a^{[2]} = \hat{y}$$

weights w $\qquad w^{[1]} = \begin{bmatrix} \text{———} \\ \text{———} \\ \text{———} \\ \text{———} \end{bmatrix}$

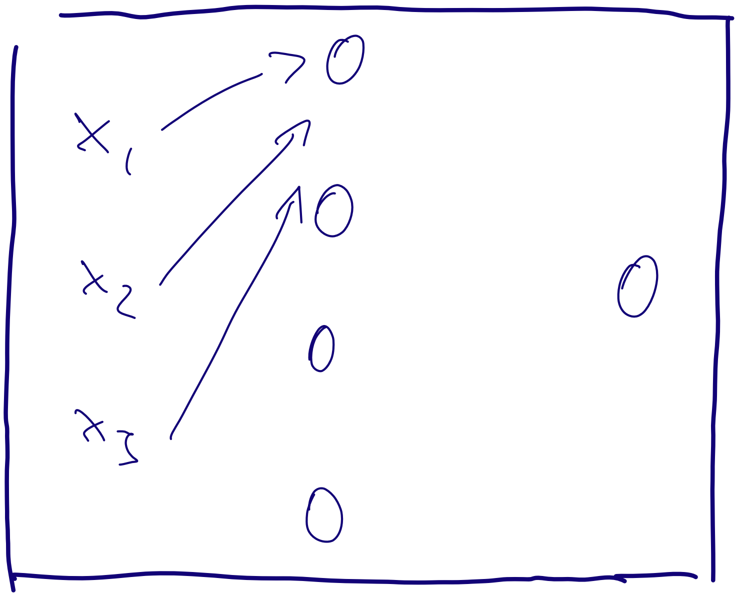
$$w^{2} = \begin{bmatrix} \text{————} \end{bmatrix}$$

bias b $\qquad b^{[1]} = \begin{bmatrix} - \\ - \\ - \\ - \end{bmatrix} \qquad b^{[2]} = \begin{bmatrix} - \end{bmatrix}$

Shallow Neural Network :

I Forward Propagation

— Zoom —



Layer 1

$$z_1^{[1]} = w_1^{[1]T} x + b_1^{[1]}$$

$$a_1^{[1]} = \sigma(z_1^{[1]})$$

$$z_2^{[1]} \ \ldots$$

$$a_2^{[1]} \ \ldots$$

$$\vdots$$

Vectorising

goal: compute z as a vector

$$\underbrace{\begin{bmatrix} — & w_1^{[1]T} & — \\ — & w_2^{[1]T} & — \\ — & w_3^{[1]T} & — \\ — & w_4^{[1]T} & — \end{bmatrix}}_{W^{[1]}} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} b_1^{[1]} \\ b_2^{[1]} \\ b_3^{[1]} \\ b_4^{[1]} \end{bmatrix} = \underbrace{\begin{bmatrix} z_1^{[1]} \\ z_2^{[1]} \\ z_3^{[1]} \\ z_4^{[1]} \end{bmatrix}}_{z^{[1]}}$$

$$\Rightarrow a^{[1]} = \begin{bmatrix} a_1^{[1]} \\ \vdots \\ a_4^{[1]} \end{bmatrix} = \sigma(z^{[1]})$$

calculating

$$z^{[1]}_{(4,1)} = W^{[1]}_{(4,3)} a^{[0]}_{(3,1)} + b^{[1]}_{(4,1)}$$

$$a^{[1]}_{(4,1)} = \sigma(z^{[1]}_{(4,1)})$$

$$z^{[2]}_{(1,1)} = W^{[2]}_{(1,4)} a^{[1]}_{(4,1)} + b^{[2]}_{(1,1)}$$

$$a^{[2]}_{[1,1]} = \sigma(z^{[2]}_{[1,1]})$$

II How wrong is it?

Cost function:

$$J(w^{[1]}, b^{[1]}, w^{[2]}, b^{[2]}) = \frac{1}{m} \sum_{i=1}^{n} \mathcal{L}(\hat{y}, y)$$

III Backprop the error (skip in class)

Forward Prop:

$$z^{[1]} = W^{[1]} a^{[0]} + b^{[1]}$$

$$a^{[1]} = g^{[1]}(z^{[1]})$$

$$z^{[2]} = W^{[2]} a^{[1]} + b^{[2]}$$

$$a^{[2]} = g^{[2]}(z^{[2]}) = \sigma(z^{[2]})$$

g can be any activation function

Backprop

$$dz^{[2]} = a^{[2]} - y$$

$$dW^{[2]} = dz^{[2]} a^{[1]T} \quad *_2$$

$$db^{[2]} = dz^{[2]} \quad *_1$$

$$dz^{[1]} = W^{[2]T} dz^{[2]} \odot g^{[1]'}(z^{[1]})$$

element wise product

$$dW^{[1]} = dz^{[1]} x^{T} \quad *_2$$

$$db^{[1]} = dz^{[1]} \quad *_1$$

compare

compare

IV Update

$$W^{[1]} = W^{[1]} - \alpha\, dW^{[1]}$$

$$b^{[1]} = b^{[1]} - \alpha\, db^{[1]}$$

$$W^{[2]} = W^{[2]} - \alpha\, dW^{[2]}$$

$$b^{[2]} = b^{[2]} - \alpha\, db^{[2]}$$

Activation functions

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

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

relu: $$a = \max(0, z)$$

leaky relu: $$a = \max(az, z)$$

Why activation functions?

$$a^{[1]} = z^{[1]} = W^{[1]} x + b^{[1]}$$
$$a^{[2]} = z^{[2]} = W^{[2]} a^{[1]} + b^{[2]}$$
$$a^{[2]} = \underbrace{W^{[2]} W^{[1]}}_{W'} x + \underbrace{W^{[2]} b^{[1]} + b}_{b'}$$

L2 regularisation

log reg

$$J(w,b) = \frac{1}{m}\sum_{i=1}^{m} \mathcal{L}(\hat{y}^{(i)}, y^{(i)}) + \frac{\lambda}{2m}\|w\|_2^2$$

$$\|w\|_2^2 = \sum_{j=1}^{n_x} w_j^2 = w^T w$$

n_x is number of cases

NN

$$J(w^{[l]}, b^{[l]}) = \frac{1}{m}\sum_{i=1}^{m} \mathcal{L}(\hat{y}^{(i)}, y^{(i)}) + \frac{\lambda}{2m}\sum_{l=1}^{l}\|w^{[l]}\|_2^2$$

$$\|w^{[l]}\|_2^2 = \sum_{i=1}^{n^{[l-1]}}\sum_{j=1}^{n^{[l]}} (w_{ij}^{[l]})^2$$

Backprop

$$dw^{[l]} = \text{from backprop} + \frac{\lambda}{m} w^{[l]}$$

Updating

$$w^{[l]} := w^{[l]} - \alpha\, dw^{[l]}$$

$$w^{[l]} := w^{[l]} - \alpha\left[\text{from bp} + \frac{\lambda}{m} w^{[l]}\right]$$

$$w^{[l]} := w^{[l]} - \alpha(\text{from bp}) - \underbrace{\frac{\alpha\lambda}{m} w^{[l]}}$$

extra penalty
depends on

- α
- λ
- m
- but also on $w^{[l]}$