PRACTICAL 3 - LINEAR & LOGISTIC REGRESSION

QUESTION 1

$x^{(1)} = [-2.0]$ $y^{(1)} = 2.0$
$x^{(2)} = [-1.0]$ $y^{(2)} = 3.0$
$x^{(3)} = [0.0]$ $y^{(3)} = 1.0$
$x^{(4)} = [2.0]$ $y^{(4)} = -1.0$

more computationally expensive ↑ alternative to a closed form solution in linear regression is an iterative optimization method such as gradient descent

$\hat{y} = w^T x + b = w' x'$ ↑ w/ bias

1. Find the closed form solution for a linear regression that minimizes the sum of squared errors on the training data.

design matrix $n \times (d+1)$ ← accounting for bias
n = num of inputs, $d+1$ = num of features

4×2
$4 \times 1+1$

$$X = \begin{bmatrix} 1 & -2 \\ 1 & -1 \\ 1 & 0 \\ 1 & 2 \end{bmatrix} \quad Y = \begin{bmatrix} 2 \\ 3 \\ 1 \\ -1 \end{bmatrix}$$

$$L(w) = \frac{1}{2} \|y - Xw\|^2$$ (↑ labels, ↑ inputs)

$\nabla_b \, a^T b = a$

$\nabla_b \, b^T A b = Ab + A^T b$

$$= \frac{1}{2} (y - Xw)^T (y - Xw)$$

$$= \frac{1}{2} (y^T y - 2 y^T X w + w X^T X w)$$

Now find the $w = [w_0 \; w_1]^T$ that minimizes sum of squared errors

$y = w^T x \iff y^T = x^T w \iff x^T w = y^T \iff w = (x^T)^{-1} y^T$

if x not invertible we use the pseudoinverse

$w = x^+ y^T$
$= (x x^T)^{-1} x y$? chatgpt
$= (x^T x)^{-1} x^T y$
pseudo inverse x^+

$$\nabla_w L(w) = \nabla_w \left[\frac{1}{2} (y^T y - 2 y^T X w + w^T X^T X w) \right]$$

$$= \frac{1}{2} \underbrace{\nabla_w y^T y}_{0} - \nabla_w \underbrace{y^T X w}_{X^T y} + \frac{1}{2} (*1)$$

$$(*1) + \frac{1}{2} \nabla_w \underbrace{(w^T X^T X w)}_{2 X^T X w}$$

$$= X^T X w - X^T y$$

$\nabla_w L(w) = 0 \iff$

$\iff X^T X w - X^T y = 0$
$\iff w = (X^T X)^{-1} X^T y$

$$w = (x^T x)^{-1} x^T y$$

$$= \left(\begin{bmatrix} 1 & 1 & 1 & 1 \\ -2 & -1 & 0 & 2 \end{bmatrix} \begin{bmatrix} 1 & -2 \\ 1 & -1 \\ 1 & 0 \\ 1 & 2 \end{bmatrix} \right)^{-1} \times \begin{bmatrix} 1 & 1 & 1 & 1 \\ -2 & -1 & 0 & 2 \end{bmatrix} \begin{bmatrix} 2 \\ 3 \\ 1 \\ -1 \end{bmatrix}$$

$$= \begin{bmatrix} 4 & -1 \\ -1 & 9 \end{bmatrix}^{-1} \begin{bmatrix} 1 & 1 & 1 & 1 \\ -2 & -1 & 0 & 2 \end{bmatrix} \begin{bmatrix} 2 \\ 3 \\ 1 \\ -1 \end{bmatrix}$$

$$= \frac{1}{\det(A)} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix} (\dots)$$
$(ad - bc)$

$$= \frac{1}{36 - 1} (\dots)$$

$$= \frac{1}{35} \begin{bmatrix} 9 & 1 \\ 1 & 4 \end{bmatrix} (\dots)$$

$$= \begin{bmatrix} 1.0286 \\ -0.8857 \end{bmatrix}$$

2. Predict the target value for $x_{query} = [1]$

$$\hat{y} = w^T x$$

$$= \begin{bmatrix} 1.0286 \\ -0.8857 \end{bmatrix}^T \cdot \begin{bmatrix} 1 \\ 1 \end{bmatrix} = 0.1429$$
dot product

3. Sketch the hyperplane along which the linear regression predicts points will fall.

hyperplane equation is given for a general input:

$x = \begin{bmatrix} 1 \\ x_1 \end{bmatrix}$ by setting the $\hat{y} = wx$ to zero $\hat{y} = wx = 0 \Rightarrow \begin{bmatrix} 1.0286 \\ -0.8857 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ x_1 \end{bmatrix} = 0$

$\Rightarrow 1.0286 - 0.8857 x_1 = 0$

4. Compute the mean squared error produced by the linear regression

$(y^{(1)} - \hat{y}^{(1)})^2 = (y^{(1)} - w \cdot x^{(1)})^2 = (2.0 - 2.8)^2 = 0.64$

$(y^{(2)} - \hat{y}^{(2)})^2 = (y^{(2)} - w \cdot x^{(2)})^2 = 1.1789$

$(y^{(3)} - \hat{y}^{(3)})^2 = 0.0008$

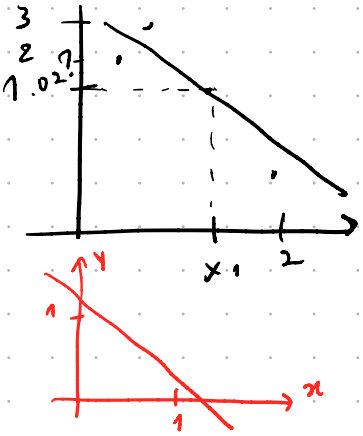
$= 0.0661$

MSE = 0.4714

number of training examples

$MSE = \frac{1}{N} \sum_{k=1}^{N} (\hat{y}^{(k)} - y^{(k)})^2$

$L(w) = \frac{1}{2} \|y - Xw\|^2$



QUESTION 2

binary logistic regression $P_w(y=1|x) = \sigma(w \cdot x) = \frac{1}{1 + \exp(-w \cdot x)}$

$w \leftarrow w + y' x'$

cross entropy loss function $L(w) = -\sum_{i=1}^{N} \log\left(P_w(y^{(i)} | x^{(i)})\right) =$

$= -\sum_{i=1}^{N} \left(y^{(i)} \log \sigma(w \cdot x^{(i)}) + (1 - y^{(i)}) \log(1 - \sigma(w \cdot x^{(i)}))\right)$

gradient descent w/ update rule of size η towards the opposite direction of the gradient of the error function w/ respect to the weights

$w \leftarrow w - \eta \frac{\partial L(w)}{\partial w}$

$L_i(w) = -\left[y^{(i)} \log\left(\sigma(wx^{(i)})\right) + (1 - y^{(i)}) \log\left(1 - \sigma(w \cdot x^{(i)})\right)\right]$

$\frac{\partial \log(\sigma(w \cdot x^{(i)}))}{\partial w} \overset{\text{chain rule}}{=} \frac{1}{\sigma(wx^{(i)})} \frac{\partial \sigma(w \cdot x^{(i)})}{\partial w}$

probability wrong signal

$\sigma(z) = \frac{1}{1 + e^{-z}}$ $\quad \frac{\partial \sigma(z)}{\partial z} = \sigma(z)(1 - \sigma(z))$ $\overset{\text{chain rule}}{=} \frac{\partial \sigma(wx^{(i)})}{\partial w} =$

$= \sigma(wx^{(i)})(1 - \sigma(wx^{(i)})) \frac{\partial wx^{(i)}}{\partial w} x^{(i)}$

$\frac{\partial \log(\sigma(wx^{(i)}))}{\partial w} = (1 - \sigma(w \cdot x^{(i)})) x^{(i)}$ (1)

probability version of logistic

$$\frac{\partial}{\partial w} \log\left(1-\sigma(w\cdot x^{(i)})\right) = \frac{1}{1-\sigma(w\,x^{(i)})} \cdot \frac{\partial\left(-\sigma(w\,x^{(i)})\right)}{\partial w} = -\sigma(w\,x^{(i)})\,x^{(i)} \quad (2)$$

so by (1) & (2):

$$\frac{\partial L_i(w)}{\partial w} = x^{(i)}\left[-y^{(i)}\left(1-\sigma(w\cdot x^{(i)})\right) + (1-y^{(i)})\,\sigma(w\,x^{(i)})\right]$$

$$= x^{(i)}\left[\sigma(w\,x^{(i)} - y^{(i)})\right]$$

$$\frac{\partial L(w)}{\partial w} = \sum_{i=1}^{N} \frac{\partial L_i(w)}{\partial w} = +\sum_{i=1}^{N} x^{(i)}\left[y^{(i)} - \sigma(w\cdot x^{(i)})\right]$$

$$w = w - \eta\,\frac{\partial L(w)}{\partial w} = w + \sum_{i=1}^{N} x^{(i)}\left[y^{(i)} - \sigma(w\cdot x^{(i)})\right]$$

(better solution to this below)

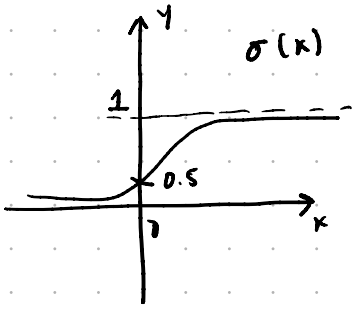
2) compute the stochastic gradient descent update assuming initialization of all zeros. Assume learning rate = 1.

In SGD we make one update for each training sample instead of summing across all data points.

$$w = w + \eta\, x^{(i)}\left(y^{(i)} - \sigma(w\cdot x^{(i)})\right) =$$

$$= \begin{bmatrix}0\\0\\0\end{bmatrix} + 1\begin{bmatrix}1\\-1\\0\end{bmatrix}\left(0 - \sigma(0)\right) =$$

$$= \begin{bmatrix}0\\0\\0\end{bmatrix} + \begin{bmatrix}-0.5\\0.5\\0\end{bmatrix} = \begin{bmatrix}-0.5\\0.5\\0\end{bmatrix}$$



(shortest version vs.)

2 d) we compute the MSE and the one that gives the least error is the better

3 (r-bernoulli) (is basically the sigmoid function) and then we use the update term (so basically GD??)

3 (b) stochastic (what's the difference?!)

3 (?) softmax - converting values into probabilities

NOTES from code Python

QUESTION 2 REMARK

Logistic Regression (Binary)

$w^T x = w_1 x_1 + \dots + w_N x_N$

$w^T \phi(x) = w_1 x_1 + \dots + w_{N+1} x_1$

$$\phi(x) = \begin{bmatrix} 1 \\ x_1 \\ \vdots \\ x_N \\ x_i \\ \vdots \end{bmatrix}$$

get higher order, its where we transition from regular ML to deeper learning

↙ the gradient of the sum is the sum of the gradients

$$\frac{\partial L(w)}{\partial w} = \frac{\partial}{\partial w}\left[-\sum_{i=1}^{N} y^{(i)} \log\left(\sigma(w^T x^{(i)})\right) + (1-y^{(i)}) \log\left(1-\sigma(w^T x^{(1)})\right)\right]$$

simpler deriving each entry and then try to unite everything in matrix notation

$$\frac{\partial}{\partial w_i}\left[y \log\left(\sigma(w^T x)\right) + (1-y)\log\left(1-\sigma(w^T x)\right)\right]$$

$$\frac{\partial}{\partial x} \log(f(x)) = \frac{1}{f(x)} \cdot \frac{\partial}{\partial x} f(x)$$

$$= y \frac{\partial}{\partial w_i} \log\left(\sigma(w^T x)\right) + (1-y) \frac{\partial}{\partial w_i} \log\left(1-\sigma(w^T x)\right)\Big]$$

$$= y \frac{1}{\sigma(w^T x)} \frac{\partial}{\partial w_i} \sigma(w^T x) + (1-y) \frac{1}{1-\sigma(w^T x)} \frac{\partial}{\partial w_i} \left(1-\sigma(w^T x)\right)$$

$$\sigma'(z) = \left(\frac{1}{1+e^{-z}}\right)' = \frac{0-(-e^{-z})}{(1+e^{-z})^2} = \underbrace{\frac{1}{1+e^{-z}}}_{\sigma(z)} \cdot \frac{-(-e^{-z})}{1+e^{-z}} = \sigma(z) \cdot \frac{1-e^{-z}-1}{1+e^{-z}}$$

$$= \sigma(z)\left(1-\sigma(z)\right)$$

$\sigma'(z) = \sigma(z)\left(1-\sigma(z)\right)$

$$y \frac{1}{\sigma(w^T x)} \frac{\partial}{\partial w_i} \sigma(w^T x) + (1-y) \frac{1}{1-\sigma(w^T x)} \frac{\partial}{\partial w_i} \left(1-\sigma(w^T x)\right)$$

$$= y \frac{1}{\sigma(w^T x)} \sigma(w^T x)\left(1-\sigma(w^T x)\right) \frac{\partial}{\partial w_i} \underbrace{w^T x}_{x_i} + (1-y) \frac{1}{1-\sigma(w^T x)} (-1) \cdot \sigma(w^T x)\left(1-\sigma(w^T x)\right) \quad (*)$$

$(*)$ $\frac{\partial}{\partial w_i} \underbrace{w^T x}_{x_i}$ $\quad g(f(x))' = g'(f(x))\, f'(x)$

$$= y\left(1-\sigma(w^T x)\right) x_i + (1-y)(-1)\,\sigma(w^T x)\, x_i$$

$$= y x_i - y\sigma(w^T x) x_i - \sigma(w^T x) x_i + y\sigma(w^T x) x_i$$

$$= (y - \sigma(w^T x)) x_i$$

$$\frac{\partial \mathcal{L}}{\partial w_i} = -\sum_{k=1}^{N} \left(y^{(k)} - \sigma(w^T x^{(k)})\right) x_i^{(k)}$$

$\hat{y}^{(k)}$

$$\nabla_w \mathcal{L}(w) = \sum_{k=1}^{N} \left(\sigma(w^T x^{(k)}) - y^{(k)}\right) x^{(k)}$$

$$\nabla_w \mathcal{L}(w) = \begin{bmatrix} \frac{\partial}{\partial w_1} \mathcal{L}(w) \\ \vdots \\ \frac{\partial}{\partial w_D} \mathcal{L}(w) \end{bmatrix}$$

$$w \leftarrow w - \eta \nabla_w \mathcal{L}(w) = w - \eta \boxed{\sum_{k=1}^{N}} \left[\sigma(w^T x^{(k)}) - y^{(k)}\right] x^{(k)}$$

gradient descent

learning rate

Stochastic GD

adapted to stochastic GD

$$w \leftarrow w - \eta \left[\sigma(w^T x^{(i)}) - y^{(i)}\right] x^{(i)}$$

$$w = [-0.5 \quad 0.5 \quad 0]^T$$