# Logistic Regression (Part II)

Till now we have seen logistic regression for binary classification, that uses Sigmoid function

Let's consider the following dataset,

| Weight ($x_1$) | Color ($x_2$) | fruit ($y$) |
|---|---|---|
| 150 | Red | Apple |
| 10 | green | Grape |
| 120 | yellow ~~green~~ | Banana |
| 8 | green | Grape |
| 200 | Red | Apple |

features

↓ target with 3 classes Apple, Grape and banana.

we can convent this dataset into (by doing one hot encoding) →

2nd sample has class grape, thats why $y_2 = 1$ and $y_1 = y_3 = 0$.

| $x_1$ | $x_2$ | $y_1$ = Apple | $y_2$ = Grape | $y_3$ = Banana |
|---|---|---|---|---|
| 150 | Red | 1 | 0 | 0 |
| 10 | green | 0 | 1 | 0 |
| 120 | yellow | 0 | 0 | 1 |
| 8 | green | 0 | 1 | 0 |
| 200 | Red | 1 | 0 | 0 |

features

Class is Apple, so $y_1 = 1$ and $y_2$ and $y_3$ is 0.

As we have three classes here, now we will consider ~~each classes~~ z-score for each classes.

so, we will 3 ~~the~~ z-score equation,

$$z_1 = w_{11}x_1 + w_{12}x_2 + \cdots + b_1 \rightarrow \text{for } y_1 \text{ column}$$

$$z_2 = w_{21}x_1 + w_{22}x_2 + b_2 \rightarrow \text{for } y_2 \text{ column}$$

$$z_3 = w_{31}x_1 + w_{32}x_2 + b_3 \rightarrow \text{for } y_3 \text{ column.}$$

$w_{32}$ → 3rd z score
→ weight for second feature $x_2$.

so, if we have K classes will have $z_1, z_2 \cdots z_K$ equations. If we have n samples in the dataset, and features

| | $x_1 \cdots x_f$ | $y_1$ | $y_2$ | .... | $y_K$ |
|---|---|---|---|---|---|
| $x^{(1)}$ | | | | | |
| $x^{(2)}$ | | | | | |
| ⋮ | | | | | |
| $x^{(n)}$ | | | | | |

we will have K number of z equations →

$$z_1 = w_{11}x_1 + w_{12}x_2 + \cdots + w_{1f}x_f + b_1$$ (f → number of features.)

$$z_2 = w_{21}x_1 + w_{22}x_2 + \cdots + w_{2f}x_f + b_2$$

$$\vdots$$

$$z_K = w_{K1}x_1 + w_{K2}x_2 + \cdots + w_{Kf}x_f + b_K$$

Again our Goal is to determine these 3 equations for the model. For that we need to find the weights and biases.

Let us we have one sample only, with four features and three-classes like below →

| | $x_1$ | $x_2$ | $x_3$ | $x_4$ | $\hat{y}_1$ | $\hat{y}_2$ | $\hat{y}_3$ |
|---|---|---|---|---|---|---|---|
| sample ← | 1 | 0.5 | 2.0 | 3.0 | ? | ? | ? |
| | | features | | | Actual target prediction | | |

As we have three classes, 3 scores →

$$z_1 = w_{11}x_1 + w_{12}x_2 + w_{13}x_3 + w_{14}x_4 + b_1$$

$$z_2 = w_{21}x_1 + w_{22}x_2 + w_{23}x_3 + w_{24}x_4 + b_2$$

$$z_3 = w_{31}x_1 + w_{32}x_2 + w_{33}x_3 + w_{34}x_4 + b_3$$

Let us assume we know all of these weights and bias values. We have to find the prediction for this sample.

Probability of sample being in class $y_1$ →

$$\hat{y}_1 = P(y_1 = 1 \mid x) = \frac{e^{z_1}}{e^{z_1} + e^{z_2} + e^{z_3}}$$

(class 1)

This will give us a prediction probability of the sample being in class 1.

The formula we have used here is **Softmax** function. This function is used inplace of sigmoid function for multinomial/multiclass logistic classification

Similarly, Probability of the sample being class 2 / $y_2$ =

$$\hat{y}_2 = P(y_2 = 1|x) = \frac{e^{z_2}}{e^{z_1} + e^{z_2} + e^{z_3}}$$

and, Probability of the sample being in class 3 / $y_3$ →

$$\hat{y}_3 = P(y_3 = 1|x) = \frac{e^{z_3}}{e^{z_1} + e^{z_2} + e^{z_3}}$$

So, For a vector $z$ of dimentionality $K$ ($K$ classes), softmax is defined as →

$$\text{Softmax}(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}, \text{ where } i \in [1, K]$$

we will have →

$$\text{Softmax}(z) = \left[ \frac{e^{z_1}}{\sum_{j=1}^{K} e^{z_j}}, \frac{e^{z_2}}{\sum_{j=1}^{K} e^{z_j}} \cdots \frac{e^{z_K}}{\sum_{j=1}^{K} e^{z_j}} \right]$$

If we get,

$\hat{y}_1 = P(y_1 = 1|x) = 0.3$ → predicted class = 0

$\hat{y}_2 = P(y_2 = 1|x) = 0.6$ → predicted class = 1

$\hat{y}_3 = P(y_3 = 1|x) = 0.2$ → predicted class = 0

So, According to this our class will be $y_2$ as it's probability is largest.

| $x_1$ | $x_2$ | $x_3$ | $x_4$ | $\hat{y}_1$ | $\hat{y}_2$ | $\hat{y}_3$ |
|---|---|---|---|---|---|---|
| 1 | 0.5 | 2.0 | 3 | 0 | 1 | 0 |

$P(y_2 = 1|x) = 0.6$
↳ largest

But while training our model we need to update the weights and biases like we did in logistic regression.

We know to update the parameters we need a loss function. In logistic regression we have used Cross Entropy loss function. We will use it in multiclass classification as well. We know for n samples,

$$L_{CE} = -\sum_{i=1}^{n} y^{(i)} \log \hat{y}^{(i)} + (1 - y^{(i)}) \log (1 - \hat{y}^{(i)})$$

In our previous example → let us Assume the Actual classes as below

| $x_1$ | $x_2$ | $x_3$ | $x_4$ | $y_1$ | $y_2$ | $y_3$ | $\hat{y}_1$ | $\hat{y}_2$ | $\hat{y}_3$ |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 0.5 | 2.0 | 3.0 | 0 | 1 | 0 | 0.3 | 0.6 | 0.2 |

Actual target — Prediction Probabilities

The General Loss function in Softmax →

$$L_{CE} = \sum_{i=1}^{k} y_k \log \hat{y}_k$$

[$(1-y_k)$ term is implicit here because of softmax function]

In our example $k=3$,

$$L_{CE} = \sum_{i=1}^{3} y_k \log \hat{y}_k$$

$$= y_1 \log \hat{y}_1 + y_2 \log \hat{y}_2 + y_3 \log \hat{y}_3$$

($y_1 = 0$, $y_2 = 1$, $y_3 = 0$)

$$= 0 \times + 1 \times \log 0.6 + 0$$

↓

because $y_2$ is the correct class here.

The full calculation for LCE here →

$$z_1 = w_{11}x_1 + w_{12}x_2 + w_{13}x_3 + w_{14}x_4 + b_1 = w_1 x + b_1$$

$$z_2 = w_{21}x_1 + w_{22}x_2 + w_{23}x_3 + w_{24}x_4 + b_2 = w_2 x + b_2$$

$$z_3 = w_{31}x_1 + w_{32}x_2 + w_{33}x_3 + w_{34}x_4 + b_3 = w_3 x + b_2$$

↓ Ve eton fonm

$$\hat{y}_2 = P(y_2 = 1|x) = \frac{e^{z_2}}{e^{z_1} + e^{z_2} + e^{z_3}}$$

$$\therefore L_{CE} = \sum_{k=1}^{K} y_k \log \hat{y}_k$$

$$= y_2 \log \hat{y}_2 \rightarrow [y_1 = y_3 = 0]$$

$$= 1 \times \log \frac{e^{z_2}}{e^{z_1} + e^{z_2} + e^{z_3}}$$

$$= \log \frac{e^{w_2 x + b_2}}{e^{w_1 x + b_1} + e^{w_2 x + b_2} + e^{w_3 x + b_3}}$$

↓ Will have to calculate this term using all weights and biases.

Now to update the weights using gradient descent we will need to use the derivative of the loss function, Let's 2 equations are →

$$z_1 = w_{11}x_1 + w_{12}x_2 + b_1$$
$$z_2 = w_{21}x_1 + w_{22}x_2 + b_2$$
$$z_3 = w_{31}x_1 + w_{32}x_2 + b_3$$

We need to find all of these 9 parameeter here.

Let us consider, $w_{11}$.

We know while updating

$$w_{11}(\text{new}) = w_{11}(\text{old}) - \eta \frac{\partial L_{EE}}{\partial w_{11}}$$

∴ As we have seen in binary Logistic regression we will use →

$$\frac{\partial L_{EE}}{\partial w_{11}} = (\hat{y}_1 - y_1)x_1$$

$\hat{y}_1$ → Prediction for $y_1$ class

$y_1$ class (Actual)

$\frac{\partial}{\partial z_1}$ → for any $y_1$ class

$w_1$ with $x_1$ feature

Similarly we will need all other derivatives,

$$\frac{\partial L_{EE}}{\partial w_{12}}, \frac{\partial L_{EE}}{\partial w_{21}}, \frac{\partial L_{EE}}{\partial b_2}, \frac{\partial L_{EE}}{\partial w_{31}}, \frac{\partial L_{EE}}{\partial w_{32}}$$

and $\frac{\partial L_{EE}}{\partial b_1}, \frac{\partial L_{EE}}{\partial b_2}, \frac{\partial L_{EE}}{\partial b_3}$.

here $\frac{\partial L_{CE}}{\partial W_{31}} = (\hat{y}_3 - y_3) x_1$

$\hat{y}_3$ → Prediction for $y_3$ class

$W_{31}$ → $z_3$ for $y_3$ class; Weight with $x_1$

and $\frac{\partial L_{CE}}{\partial b_3} = (\hat{y}_3 - y_3)$

$\hat{y}_3$ → Prediction for $y_3$ class.

$b_3$ → bias for $z_3$ equation. $z_3$ is for $y_3$ class

Using these derivatives we can update the parameter in Gradient Descent approach.

The General Formual ~~fo~~ for derivative of $L_{CE}$ with respect to $W_{ki}$

$k$ → class no.

$i$ → feature no.

$$\frac{\partial L_{CE}}{\partial W_{ki}} = (\hat{y}_k - y_k) x_i$$

and with respect to $b_k$

$$\frac{\partial L_{CE}}{\partial b_k} = (\hat{y}_k - y_k)$$

Example:

Do, one cycle of gradient Descent update for the following dataset.

| $x_1$ | $x_2$ | $y_1$ | $y_2$ | $y_3$ |
|---|---|---|---|---|
| 1 | 2 | 1 | 0 | 0 |

where, $w_{11} = 0.2$, $w_{12} = (-0.10)$, $w_{21} = 0.0$, $w_{22} = 0.10$

$w_{31} = -0.2$, $w_{32} = 0.05$

$\eta = 0.1$, $b_1 = 0.10$, $b_2 = 0.00$, $b_3 = -0.05$

Solution: here,

$$z_1 = w_{11}x_1 + w_{12}x_2 + b_1$$

$$z_2 = w_{21}x_1 + w_{22}x_2 + b_2$$

$$z_3 = w_{31}x_1 + w_{32}x_2 + b_3$$

$\therefore z_1 = (0.2) \times 1 + (-0.10) \times 2 + (0.10) = 0.10$

$z_2 = (0.0) \times 1 + (0.10) \times 2 + (0.00) = 0.20$

$z_3 = (-0.2) \times 1 + (0.05) \times 2 + (-0.05) = -0.15$

$\therefore e^{z_1} = 1.105$

$e^{z_2} = 1.221$

$e^{z_3} = 0.861$

And $e^{z_1} + e^{z_2} + e^{z_3} = 3.187$

$$\hat{y}_1 = P(y_1 = 1) = \text{softmax}(z_1)$$

$$= \frac{e^{z_1}}{e^{z_1} + e^{z_2} + e^{z_3}} = \frac{1.105}{3.187}$$

$$= 0.3467$$

$$\hat{y}_2 = P(y_2 = 1) = \text{softmax}(z_2)$$

$$= \frac{e^{z_2}}{e^{z_1} + e^{z_2} + e^{z_3}} = \frac{1.221}{3.187}$$

$$= 0.383$$

$$\hat{y}_3 = P(y_3 = 1) = \text{softmax}(z_3)$$

$$= \frac{e^{z_3}}{e^{z_1} + e^{z_2} + e^{z_3}} = \frac{1.221}{3.187} \quad \frac{0.8607}{3.187}$$

$$= 0.27$$

As $\hat{y}_2 = 0.38$, is highest, predicted class is $y_2$.

$$L_{CE} = \sum_{i=1}^{3} y_k \log \hat{y}_k$$

$$= y_1 \log \hat{y}_1 + y_2 \log \hat{y}_2 + y_3 \log \hat{y}_3$$

$$= 1 \times \log 0.3467 + 0 \times \log \hat{y}_2 + 0 \log \hat{y}_3$$

$$= 1.059$$

Updates →

$$w_{11} = w_{11} - \eta(\hat{y}_1 - y_1)x_1 = 0.2 - 0.1 \times (0.3467 - 1)1$$

$$= 0.2653$$

$$w_{12} = w_{12} - \eta(\hat{y}_1 - y_1)x_2 = (-0.10) - 0.1(0.3467 - 1)2$$

$$= 0.0307$$

$$w_{21} = w_{21} - \eta(\hat{y}_2 - y_2)x_1 = 0.00 - 0.1(0.3832 - 0)1$$

$$= -0.0383$$

$$w_{22} = w_{22} - \eta(\hat{y}_2 - y_2)x_2 = 0.10 - 0.1 \times (0.3832 - 0)2$$

$$= 0.02334$$

$$w_{31} = w_{31} - \eta(\hat{y}_3 - y_3)x_1 = -0.2 - 0.1 \times (0.27 - 0)1$$

$$= -0.227$$

$$w_{32} = w_{32} - \eta(\hat{y}_3 - y_3)x_2 = 0.05 - 0.1(0.27 - 0)2$$

$$= -0.004$$

and, $$b_1 = b_1 - \eta(\hat{y}_1 - y_1) = 0.10 - 0.1 \times (0.3467 - 1)$$

$$= 0.1653$$

$$b_2 = b_2 - \eta(\hat{y}_2 - y_2) = 0.0 - 0.1 \times (0.3832 - 0)$$

$$= 0 - 0.03832$$

$$b_3 = b_3 - \eta(\hat{y}_2 - y_3) = 0.05 - 0.1 \times (0.27 - 0)$$

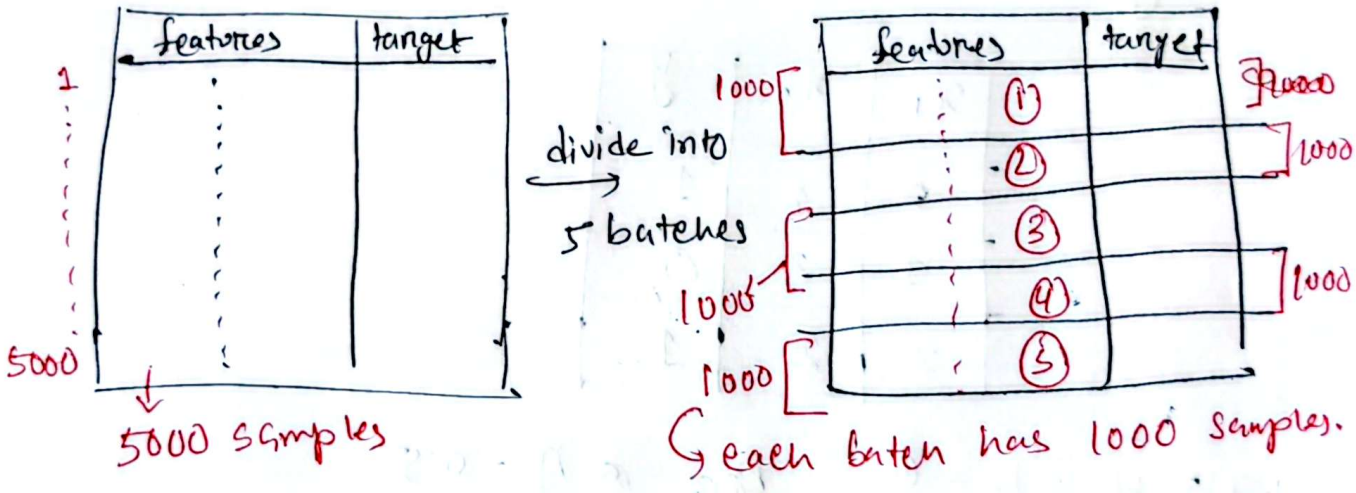
$$= -0.077$$

# Minibatch Gradient Descent (For Binary Classification):

We have already seen an example for stochastic Gradient Descent.

In minibatch process, we devide the ~~dataset~~ training dataset into small ~~training~~ batches to update the parameters. So, here updates takes place batch by batch, not sample by sample.

For example →




The Gradient Deseent will run for each batch and update the parameters after processing each batch. So, ~~$~~ updak will take place 5 times here instead of 5000 times like stochastic gradient descent. These 5 batches are called **1 Epoch**

The entire process is ~~the~~ repeated for

multiple epochs until the model converges.

**Batch Gradient Descent:** The entire dataset gets processed at once. So, the parameters gets updated after processing all samples of the dataset.

**Example:**

Perform one Gradient Descent cycle for the following ~~dataset~~ batch using minibatch Gradient Descent

| $x_1$ | $x_2$ | $y$ |
|---|---|---|
| 2 | 1 | 1 |
| 1 | 3 | 0 |
| 2 | 2 | 1 |

$w_1 = w_2 = b = 0$, $\eta = 0.5$

**Solution:**

$$z = w_1 x_1 + w_2 x_2 + b$$

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

We know to update the parameters,

$$w_1 = w_1(\text{old}) - \eta \left[\frac{\delta L_{CE}}{\delta w_1}\right]$$

$$w_2 = w_2(\text{old}) - \eta \left[\frac{\delta L_{CE}}{\delta w_2}\right]$$

$$b = b(\text{old}) - \eta \left[\frac{\delta L_{CE}}{\delta b}\right]$$

→ These gradients will be the average of individual gradients.

We know for each sample,

$$\frac{\delta L_{CE}}{\delta w_1} = (\hat{y} - y) x_1 \qquad \frac{\delta L_{CE}}{\delta w_2} = (\hat{y} - y) x_2$$

And $$\frac{\delta L_{CE}}{\delta b} = (\hat{y} - y)$$

But in batch gradient descent it becomes →

$$\frac{\delta L_{CE}}{\delta w_1} = \frac{1}{m} \sum_{i=1}^{m} \left[\hat{y}^{(i)} - y^{(i)}\right] x_1^{(i)}$$

$$\frac{\delta L_{CE}}{\delta w_2} = \frac{1}{m} \sum_{i=1}^{m} \left[\hat{y}^{(i)} - y^{(i)}\right] x_2^{(i)}$$

$$\frac{\delta L_{CE}}{\delta b} = \frac{1}{m} \sum_{i=1}^{m} \left[\hat{y}^{(i)} - y^{(i)}\right]$$

↓

Average of all individual samples' gradients.

## Cycle – 1 $(w_1 = w_2 = b = 0)$

$z = w_1 x_1 + w_2 x_2 + b$

| | $x_1$ | $x_2$ | $y$ | $z$ | $\hat{y}$ | $\hat{y} - y$ | $(\hat{y} - y) x_1$ | $(\hat{y} - y) x_2$ |
|---|---|---|---|---|---|---|---|---|
| ① | 2 | 1 | 1 | 0 | 0.5 | -0.5 | -1 | -0.5 |
| ② | 1 | 3 | 0 | 0 | 0.5 | 0.5 | 0.5 | 1.5 |
| ③ | 2 | 2 | 1 | 0 | 0.5 | -0.5 | -1 | -1 |

$m = 3$

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

$$\therefore \frac{\delta L}{\delta w_1} = \frac{1}{m} \sum_{i=1}^{m} \left[ \left( \hat{y}^{(i)} - y^{(i)} \right) x^{(i)} \right]$$

$$= \frac{1}{m} \left[ \left( \hat{y}^{(1)} - y^{(1)} \right) x^{(1)} + \left( \hat{y}^{(2)} - y^{(2)} \right) x^{(2)} + \left( \hat{y}^{(3)} - y^{(3)} \right) x^{(3)} \right]$$

$$= \frac{1}{3} (-1 + 0.5 - 1) = \frac{-1.5}{3} = -0.5$$

$$\therefore \frac{\delta L}{\delta w_2} = \frac{1}{3} (0.5 + 1.5 - 1) = \frac{0}{3} = 0$$

$$\therefore \frac{\delta L}{\delta b} = \frac{1}{3} (-0.5 + 0.5 - 0.5) = \frac{-0.5}{3} = -0.167$$

Updates → $w_1 = 0 - 0.5 \times (-0.5) = 0.25$

$w_2 = 0 - 0.5 \times 0 = 0$

$b = 0 - 0.5 \times (-0.167) = 0.835$