Niraj Mahajan - 180050069

CS 335 - CS 337 Assignment - 3.

1.1]. Features used :-

a). Perimeter.

- Idea is to use the perimeter of the shape to distinguish between shapes. Stars will have a ~~[illegible]~~ very high perimeter, as compared to others, while triangles will have a lower perimeter.
- Method.
  - Our input is a reshaped 2500 dimensional vector of the image. This is basically all rows of the image concatenated one after the other.
  - A pixel will be on the perimeter if it has all zeros on 'one' side and all ones on another side.
  - If we convolve a vector [1,1,1,1,1] through the 2500 element input, and chose the elements with value 2, or 3, then we basically get all the vertical edge pixels.
  - Similarly, if we convolve the same mask on X.reshape(50,50).T.reshape(-1) where X is the input, we will get the number of horizontal edge pixels.
  - If we sum up these two numbers, then we have a rough estimate of the ~~parameter~~. perimeter.

b) Total Area

→ Idea

→ Idea for this feature is quite simple: all shapes will occupy different areas and this can be a good estimate to ~~get the total~~ differentiate between classes.

→ Method

→ Simply add the input image pixels.

c) Ring Area. (Using just Ring Area, without other features gives 100% Test Accuracy).

→ Idea

→ This, according to me, is the best feature that differentiates between classes.

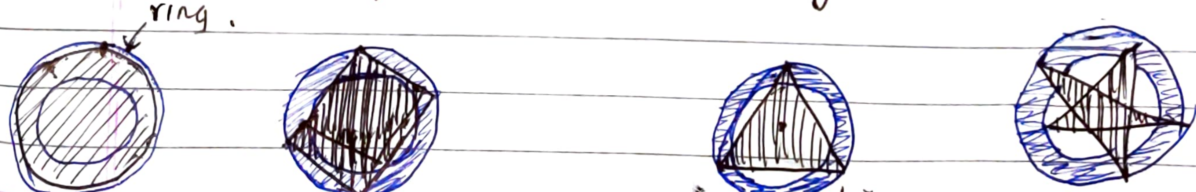
→ Consider a ring (donut) with ~~out~~ centre (25,25) and outer radius = max distance of any pixel from centre, ~~inner radius set to outer radius - 6~~, and, inner radius = max(4, outer radius − 6)

→ If we consider the white area inside this ring (donut), then it can, by itself give 100% Test Accuracy.

Illustration:

Circle. Square. Triangle. Star.

ring.



No white area in ring. | Some white area in ring. | Huge white area in ring. | Lower area in the ring.

Q - 2.1

(a) For a binary logistic regression, our labels were 0, 1. So, lets look at the one hot encoding of these labes.

$$y_{(0)} = [1, 0]$$
$$y_{(1)} = [0, 1]$$

Hence $y_{one\ hot} = [(1-y) \quad y]$, for any image sample in data set.

So, if we ~~app~~ have the following formula for multi class LR.

$$E(w) = \frac{-1}{N} \sum_{i=1}^{N} \sum_{k=1}^{K} y_k^{(i)} \log\left(P\left(Y=k \mid w_k, \phi(x^{(i)})\right)\right).$$

For binary we have $K = 2$, which gives us.

$$E(W) = \frac{-1}{N} \sum_{i=1}^{N} \left\{ \left[(1-y^{(i)}) \log\left(P\left(Y=0 \mid w_0, \phi(x^{(i)})\right)\right) \right] + \left[y^{(i)} \log\left(P\left(Y=1 \mid w_1, \phi(x^{(i)})\right)\right) \right] \right\}$$

In the slides, eqn 3, $P(Y=1 \mid w_1, \phi(x^{(i)})) = \sigma_w(x^{(i)})$.

Hence, $P(Y=0 \mid w_0, \phi(x^{(i)})) = 1 - \sigma_w(x^{(i)})$.

~~Hence we have~~

$$\cancel{E(w) = \frac{-1}{N} \sum_{i=1}^{N} \left\{ y^{(i)} \log \sigma_w(x^{(i)}) + (1-y^{(i)}) \right.}$$

Hence we have.

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

Hence proved, binary L·R is a special case of Multiclass L·R.

b) Define $Z_{N\times K} = \phi(x)_{N\times D} \cdot W_{D\times K}$.

Each element of Z - $z_{ik} = \phi(x^{(i)})_{1\times D} \cdot W^{K}_{D\times 1}$ (scalar).

We need $\frac{dE}{dW}_{D\times K}$ ~~dE~~

Any ~~element~~ ~~column~~ element of this matrix,

$$\left(\frac{dE}{dW}\right)_{D\times K} = \frac{dE_{k}}{dw_{k,d}} = \frac{dE_{k}}{dz} \cdot \frac{dz}{dW_{kd}}.$$

→ To Find $\frac{dE}{dz}$

$$P\left(Y=k \,/\, \phi(x)^{(i)}, w^{k}\right) = \frac{e^{z_{ki}}}{\sum_{k=1}^{K} e^{z_{ki}}}$$

$$\log\left(P\left(Y=k \,/\, \phi(x)^{(i)}\, w_k\right)\right) = z_{ki} - \log\left(\sum_{k=1}^{K} e^{z_{ki}}\right).$$

b). Define $Z_{N\times K} = \phi(x)_{N\times D} \cdot W_{D\times K}$

$z_{i,k}$: Each element of Z_{NK} $= \phi(x^{(i)})_{1\times D} \cdot W^{k}_{D\times 1}$

We have

$$E = -\frac{1}{N} \sum_{i=1}^{N} \sum_{k=1}^{K} y_k^{(i)} \log\left(P\left(Y=k \mid w_k, \phi(x^{(i)})\right)\right)$$

$$= -\frac{1}{N} \sum_{i=1}^{N} \sum_{k=1}^{K} y_k^{(i)} \left\{ z_{ik} - \log\left(\sum_k e^{z_{ik}}\right) \right\}.$$

$$= -\frac{1}{N} \left\{ \sum_{i=1}^{N} \sum_{k=1}^{K} y_k^{(i)} z_{ik} - \sum_i \left(\sum_k y_k^i\right) \log\left(\sum_k e^{z_{ik}}\right) \right\}.$$

Since $\sum_k y_{ik} = 1$ as it is one-hot encoding for a class

$$= -\frac{1}{N} \left\{ \sum_i \sum_k y_{ik} z_{ik} - \sum_i \log\left(\sum_k e^{z_{ik}}\right) \right\}. \quad \text{— (1)}$$

So, we want $\frac{dE}{dW_{D\times K}}$ ~~the~~

The $(d,k)^{th}$ element of this Matrix will be $\frac{dE}{dw_{dk}}$.

$$\frac{dE}{dw_{d,k}} = \left(\sum_i \frac{dE}{dz_{ik}}\right) \cdot \frac{dz_{ik}}{dw_{dk}}.$$

$\rightarrow$ ~~$\frac{dz_{ik}}{dw_{dk}} = \sum_d \phi(x^{(i)})_d \, w_{dk}$~~

→ For ~~dz~~ $\frac{dz_{ik}}{dw_{dk}}$,

$$z_{ik} = \sum_{d} w_{dk} \cdot \phi(x^{(i)})_d$$

$$\frac{dz_{ik}}{dw_{dk}} = \phi(x^{(i)})_d = \phi_{i,d} \quad \text{—— ②}$$

→ For $\frac{dE}{dz_{ik}}$,

Using eq^n ①

$$\frac{dE}{dz_{ik}} = \frac{-1}{N}\left(y_{ik} - \frac{e^{z_{ik}}}{\sum_{k} e^{z_{ik}}}\right).$$

$$\Rightarrow \cancel{\frac{dE}{dW_{dk}} = \frac{-1}{N}\phi_{i,d}\left(y_{ik} - e\right.}$$

$$\Rightarrow \boxed{\frac{dE}{dW_{dk}} = \frac{-1}{N}\sum_{i}\phi_{i,d}\left(y_{ik} - \frac{e^{z_{ik}}}{\sum_{k} e^{z_{ik}}}\right)}$$

Vectorized:-

$$\boxed{\vec{\nabla}_W E_{D\times K} = \frac{-1}{N}\,\phi(x)^{T}_{D\times N}\left(Y_{N\times K} - \text{Softmax}(Z, dim=1)\right)}$$

• Here Softmax (Z, dim = 1)
= np.exp(z) / np.sum(Z.exp(), 1).

⇒ Vectorized:

$$\boxed{\vec{\nabla}_w E_{D\times K} = \frac{-1}{N}\, \phi(X)^T_{D\times N} \left(Y_{N\times K} - \text{Softmax}\left(\phi(x)_{N\times D} \cdot W_{D\times K}, \text{dim} = 1 \right) \right)}$$

Q - 2.2.

b). Test Accuracy = 86.33 %
For Model M (all zeros)
↳ test accuracy = 84.18 %
Here Accuracy is not a good metric to gauge the model performance. This is because the data is heavily imbalanced. The number of negative samples are much more than the positive samples, and hence accuracy will give a high value even for such a bad model.

c). F_1 score = 0.301
F_1 score for Model M = 0.
I believe F_1 score is a good metric as compared to Accuracy. In cases where data is non uniform, with low number of positives, like Cancer detector / Fraud Detector, accuracy will give high ~~accuraci~~ value even for bad modes.

But F_1 score, on the other hand, considers False Negatives, False positives as well, and hence will give a better estimate in such cases, ~~whether~~ of whether our model is good or not.
Also, most real life datasets are not uniform. and hence, F_1 score is better than Accuracy.

2.2) c) Test Accuracy for multiclass LR = 84.43 %.
Test Accuracy for multiclass perceptron = 78.28 %.

Logistic Regression achieves a higher test accuracy. The reason for this is, although both the models try to exploit the linear separability in the data, ~~there actually is~~ linear regression (since it use sigmoid/softmax) has a continuous range of outputs, whereas, the perceptron gives a binary Yes/NO output.
Also, while traing, a perceptron, the gradient from an image is not considered even if it is 'marginally' correctly classified, whereas in case of Logistic regression, the gradient tries to adjust the ~~plane~~ model even for correctly classified images, in order to maximise their classification score (probability). This leads to a better separating plane (~~more~~ closer to the ideal optima) in case of Logistic Regression as compared to perceptron.
Hence, the accuracy of Logistic Regression is better than that of a perceptron.