REPORT

Analysis of binary Logistic Regression :

1. **Learning rate** -- Effect of the learning rate on the loss and accuracy.

   For Binary Classification after completion of model we had conducted analysis to find the best hyperparameters firstly to find suitable learning rate, we fixed the number of epochs to be 100 and analyse the loss and accuracy graphs with different learning rates. From the result in we got Table 1.

Learning rate	Train Loss	Validation Loss	Train Accuracy	Validation Accuracy
0.1	0.16	0.36	0.95	0.87
10^{-2}	0.35	0.50	0.76	0.83
10^{-4}	0.49	0.61	0.71	0.67
10^{-6}	0.49	0.61	0.71	0.67

   Table 1 : Analysis of Binary classifier at different learning rates with epoch fixed to 100.

   We started with learning rate 0.1 and found that model converges to a point with preferred loss and accuracy. As we decrease the learning rate to 10^{-2}, the model took smaller steps to converge and in 100 epoch it is giving more loss and less accuracy than previous learning rate. Now by decreasing the learning rate further by 10^{-4} and 10^{-6} we found that model was taking more time to converge with very little steps and in 100 epoch it just moved less distance from initial value towards the minima resulting in more loss and less accuracy comparing with previous values of learning rate. After analysing with four different we found that learning rate 0.1 is suitable for this model.

2. **Number of epochs** -- Effect of the number of epochs on the loss and accuracy.

   In this part we kept the learning rate to 0.1 and changed epoch values and analysed the loss and accuracy graphs with different epochs .By observation we got Table 2.d

Epoch	Train Loss	Validation Loss	Train Accuracy	Validation Accuracy
100	0.16	0.36	0.95	0.87
250	0.13	0.37	0.97	0.87
500	0.12	0.4	0.96	0.87
1000	0.11	0.45	0.96	0.87

   Table 2: Analysis of Binary classifier at different epochs with learning rate fixed to 0.1.

   At 100 epoch we got the observation which we observed in learning rate analysis with 0.1. Now by increasing the number of epochs to 250 we observed that sight decrease in loss and slight increase in accuracy than 100. As we increase further to 500 and 1000, we observed that the model was getting overfitting as training loss was reducing and validation loss was increasing with respect to number of epochs by maintain the almost similar accuracy. After observation we took number of epochs as 250.

3. **Momentum** -- Effect of the momentum on the loss and accuracy.

   In this by keeping the learning rate and epoch to 0.1 and 250, we did analysis with momentum and analyse the loss and accuracy graphs with different momentum. The results we observed was listed in Table 3.

Momentum	Train Loss	Validation Loss	Train Accuracy	Validation Accuracy
0.0	0.13	0.37	0.97	0.87
0.9	0.1	0.58	0.96	0.87

Table 3 : Analysis of Binary classifier at momentum 0 and 0.9 with learning rate and epoch fixed to 0.1 and 250.

For the momentum of 0.9 which is usually used to reduce the oscillation during updating the weights i.e., taking the steps. So, at $\mu = 0.9$ the speed of the convergence is fast. But we can see increase in validation loss and also slight variation in training accuracy. By analysis we took $\mu = 0$ as momentum.

After tuning the hyper-parameters, we get learning rate – 0.1, number of epochs – 250 and momentum – 0. We can see the variation of loss and accuracy with respect to epoch at final hyper- parameters in Fig1 and Fig2.

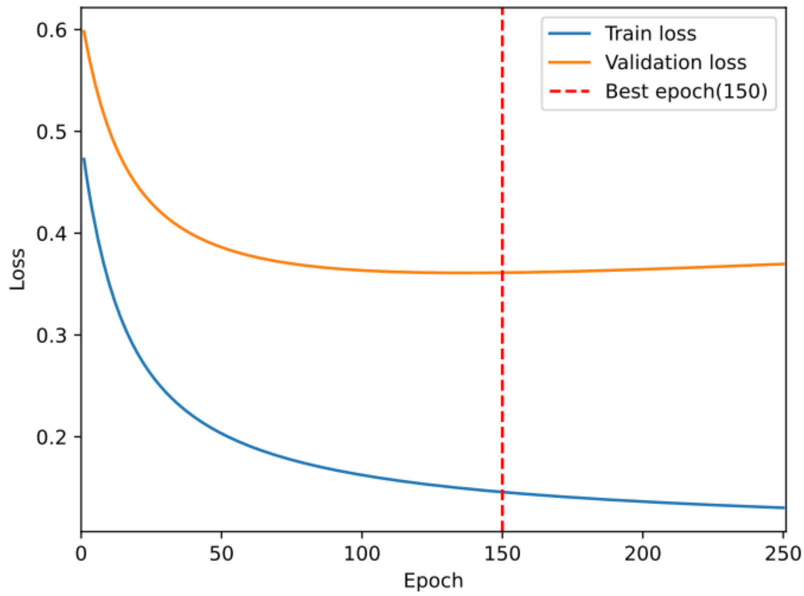

Fig 1. Variation of loss with respect to epoch at fixed learning rate – 0.1, epoch – 250 and momentum - 0

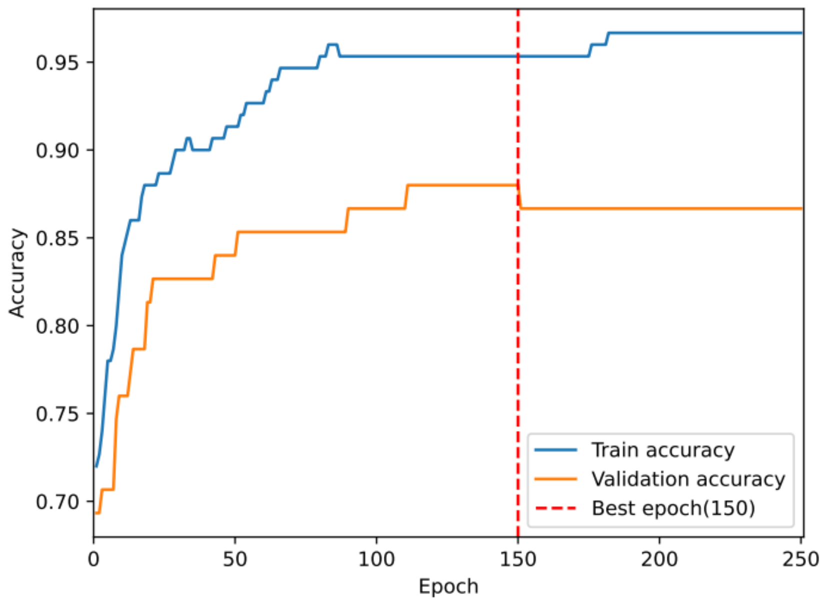

Fig 2. Variation of accuracy with respect to epoch at fixed learning rate – 0.1, epoch – 250 and momentum - 0

Analysis of Iris Linear Classifier:

1. **Learning rate** -- Effect of the learning rate on the loss and accuracy.

   For Iris Linear Classification after completion of model we had conducted analysis to find the best hyperparameters. Firstly, to find suitable learning rate, we fixed the number of epochs to be 100 and analyse the loss and accuracy graphs with different learning rates. From the result in Table4.

Learning rate	Train Loss	Validation Loss	Train Accuracy	Validation Accuracy
0.1	0.49	0.43	0.96	0.96
10^{-2}	0.79	0.77	0.76	0.8
10^{-4}	0.73	0.73	0.31	0.4
10^{-6}	0.69	0.69	0.31	0.4

Table 4 : Analysis of Linear classifier at different learning rates with epoch fixed to 100.

Firstly, we started with learning rate 0.1 and found the small loss with preferred accuracy as the model converges to minima. As we decreasing the learning rate to 10^{-2} we found that model was taking more time to converge with small steps and in 100 epoch it results in increase in the loss and reduction in accuracy compared to previous observation. Then upon reducing the model further by 10^{-4} and 10^{-6} we found that loss was increasing further and even accuracy was reduced drastically due to the less variation in the updated values from the previous during updating weights. After observation we choose 0.1 as learning rate.

2. **Number of epochs** -- Effect of the number of epochs on the loss and accuracy.

   In this part we kept the learning rate to 0.1 and changed epoch values and analysed the loss and accuracy graphs with different epochs .By observation we got Table 5.

Epoch	Train Loss	Validation Loss	Train Accuracy	Validation Accuracy
100	0.49	0.43	0.96	0.96
250	0.42	0.35	0.97	1
500	0.37	0.29	0.96	1
1000	0.34	0.25	0.96	1

Table 5: Analysis of Linear classifier at different epochs with learning rate fixed to 0.1.

By starting with epochs as 100 and learning rate as 0.1 we observed the loss was a bit more and with preferred accuracy. Upon changing the number of epochs to 250 we found that model reduction in loss and slight increase in accuracy values. By increasing the epochs, the loss was even reduced by maintaining the accuracy level.
By analysis we took number of iterations to be 1000.

3. **Momentum** -- Effect of the momentum on the loss and accuracy.

In this by keeping the learning rate and epoch to 0.1 and 1000, we did analysis with momentum and analyse the loss and accuracy graphs with different momentum. The results we observed was listed in Table 6.

Momentum	Train Loss	Validation Loss	Train Accuracy	Validation Accuracy
0.0	0.34	0.25	0.96	1
0.9	0.27	0.16	0.98	1

Table 6 : Analysis of Linear classifier at momentum 0 and 0.9 with learning rate and epoch fixed to 0.1 and 1000.

For the momentum of 0.9 which is usually used to reduce the oscillation during updating the weights in gradient descent i.e., taking during steps upgradation. So, at $\mu = 0.9$ the speed of the convergence is fast and we found that decrease in loss and also slight increase in accuracy of the model. By analysis we took $\mu = 0.9$ as momentum.

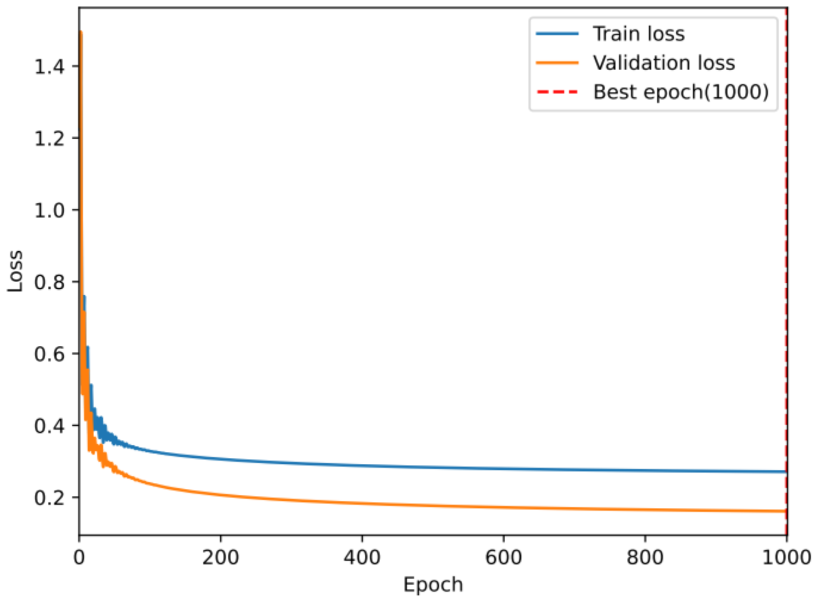

Fig 3. Variation of loss with respect to epoch at fixed learning rate – 0.1, epoch – 1000 and momentum – 0.9.

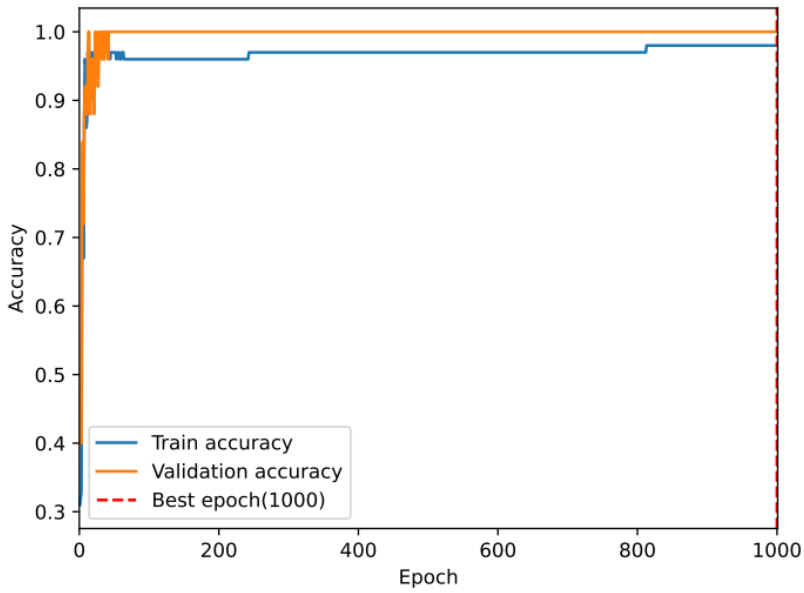

Fig 4. Variation of accuracy with respect to epoch at fixed learning rate – 0.1, epoch – 1000 and momentum – 0.9.

Mathematical derivation of the gradient term for logistic regression which you must [2 marks] have used while implementation of logistic regression.

Sol.4

$$\text{sigmoid} = \frac{1}{1+e^{-A}}$$

$$y = \frac{1}{1+e^{-A}} \quad \text{and} \quad \boxed{A = (w_0 + w_1x_1 + x_2w_2 + \cdots + w_d x_d)}$$

now loss fxn is given by :-

$$\text{loss} = -\left[\left(y\log(\hat{y}) + (1-y)\log(1-\hat{y}) \right) \right]$$

$$\text{loss} = -\left[\log(1-\hat{y}) + y\left(\log\left(\frac{\hat{y}}{1-\hat{y}} \right) \right) \right]$$

substituting value of $\hat{y}$

$$\text{loss} = -\left[\log\left(\frac{e^{-A}}{1+e^{-A}} \right) + y\left(\log\left(e^{A} \right) \right) \right]$$

$$\text{loss} = -\left[-\log(1+e^{A}) + yA \right]$$

$$\text{now gradient} = \frac{\partial L}{\partial w}$$

$$\text{grad} = \frac{\partial}{\partial w}\left[\log(1+e^{A}) - yA \right]$$

$$\text{grad} = \left[\frac{e^{A}}{1+e^{A}} - y \right] \frac{\partial A}{\partial w}$$

$$\text{grad} = \left[\hat{y} - y \right] \frac{\partial A}{\partial w}$$

now Avg gradient: $\boxed{\frac{1}{N} \sum_{i=1}^{N} [\hat{y} - y] \frac{\partial A}{\partial w}}$

$$\boxed{\frac{\partial A}{\partial w} = \begin{cases} x_i & \text{for } w = w_i \quad i > 0 \\ 1 & \text{for } w = w_0 \end{cases}}$$

Study on iris. How to adopt logistic regression to multi-class setting.

We've seen that we could use logistic regression in multinomial settings by using the SoftMax function. The SoftMax function instead of taking just the argmax value among the calculated score where the score of 2 classes can be close, but when we use the SoftMax function it blows up the differences and which is better for our classification.

The scores that are talked above are calculated for each set of weights for each class.

Each class has its own set of weights and for each input(xi) we calculate the scores and what we think is the maximum score should be calculated for the true label class if not we try with the next set of weights this is our training procedure.

Main aim is that we want to reduce the difference between true and pred labels/class.

In the Iris Example the

Classes(k) = 3

Features(d) = 4

To predict the output belongs to class ‘i’ given for the input sample is ‘x’.

$$P(y = i|x) = \frac{e^{zi}}{\sum_{j=1}^{k} e^{zj}}$$

Here K is the number of classes present in the model i.e., 3 and z_i is the linear combination of weights for the input in class i.