Diabetes prediction

OptML Project 1

Mukesh Kumar Ram
20654

Abstract - Diabetes is a chronic disease that affects millions of people worldwide and is associated with significant health complications. Early detection and intervention can improve health outcomes, reduce healthcare costs, and prevent complications. In this project I have built a classifier that can predict whether a person is diabetic or not by using diagnostic measurements and observed the behavior of "Gradient Descent" and "Gradient Descent with Nesterov Momentum".

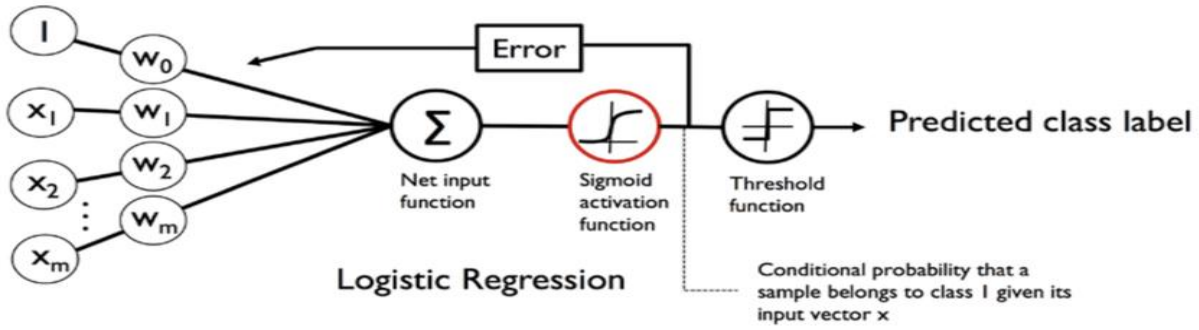
Introduction-

- Dataset- "Diabetes Dataset" available on Kaggle. It includes diagnostic measurements. It has 768 samples of 8 features and one target variable shown in figure 1.

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0

figure 1

- Model- Logistic Regression is used as classifier.

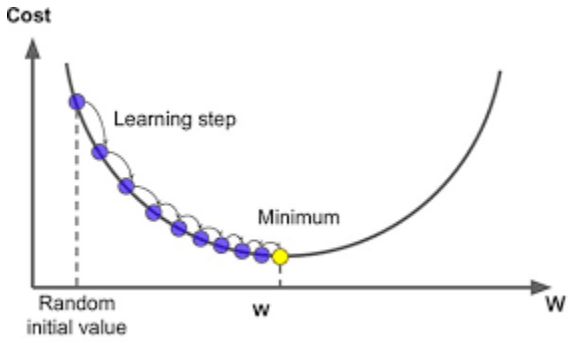


- **Loss Function**

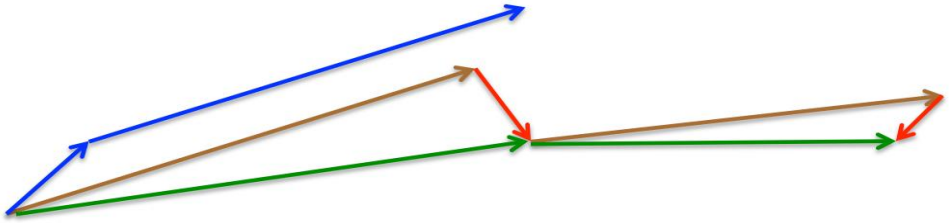
1. Cross Entropy Loss- Binary cross Entropy - $L(y, \hat{y}) = - [y * \log(\hat{y}) + (1-y) * \log(1-\hat{y})]$. The function is Convex, but not L-smooth function.
2. Hinge Loss - $L(y, f(x)) = \max(0, 1 - y * f(x))$

- **Gradient Descent**- Gradient descent is an optimization algorithm that iteratively minimizes a cost function by updating model parameters in the opposite direction of the gradient of the cost function.

w = w - learning_rate * gradient_w



- **Gradient Descent with Nesterov Momentum** - Nesterov Momentum is an enhanced version of gradient descent that uses a "momentum" term to speed up the optimization process.



v = gamma * v + learning_rate * gradient_w

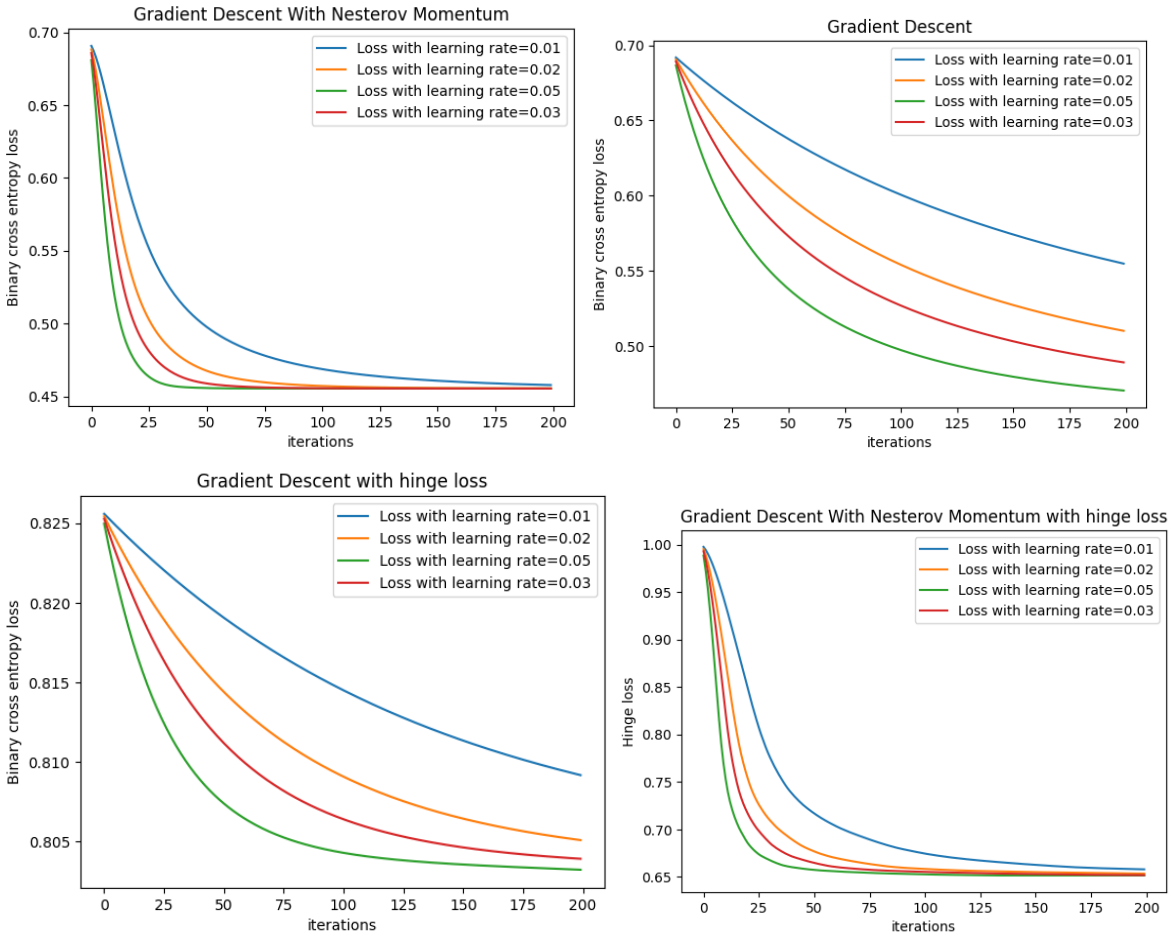
w = w - gamma * v + learning_rate * gradient_w

- v' is the previous update vector
- 'gamma' is the momentum term, typically set between 0.9 and 0.99
- 'Learning rate' is the step size or learning rate.

Data Pre-Processing – The dataset is unbalanced. It has 500 negative samples and 268 positive samples. Used oversampling for balancing the data then standardized the dataset.

Loss implementation – Implemented both “Cross Entropy” and “Hinge Loss” in logistic regression and train the model using dataset. Observe the behavior of “Gradient Descent” and “Gradient Descent with Nesterov Momentum” for different learning rate.

Results



Observation- I observed that with gradient descent with Nesterov's momentum I obtain a faster convergence for both cases in hinge as well as binary cross entropy loss.

Novelty - Implement hinge loss for logistic regression.