Diabetes prediction

OptML Project 1

Aim – Diabetes prediction using “Gradient Descent ” and “Gradient Descent With Nesterov Momentum ”.

Problem Statement – Implement the “Gradient Descent ” and “Gradient Descent With “Nesterov Momentum” from scratch for classifying the diabetes data.

Motivation – Predict the possibility of having diabetes, based on certain diagnostic measurements like Glucose, blood pressure, skin thickness, etc..

Novelity – Observed behaviors of GD and Nesterov momentum with “Hinge loss”.

Mukesh Kumar Ram
20654

Dataset- “Diabetes Dataset” having diagnostic measurements.

Source - https://www.kaggle.com/datasets/akshaydattatraykhare/diabetes-dataset

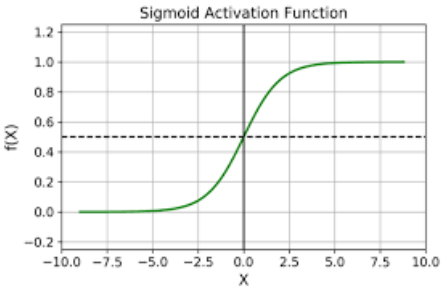
Dimension – 768 x 9

	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

Model- Logistic Regression

Loss – Binary cross Enrtopy - $L(y, \hat{y}) = - [y * \log(\hat{y}) + (1-y) * \log(1-\hat{y})]$

Activation function – Sigmoid



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.

- Waight update

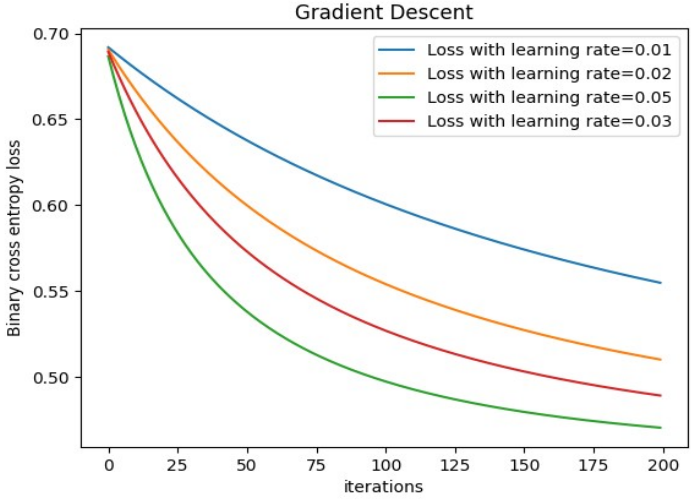
$$w = w - learning\_rate * gradient\_w$$

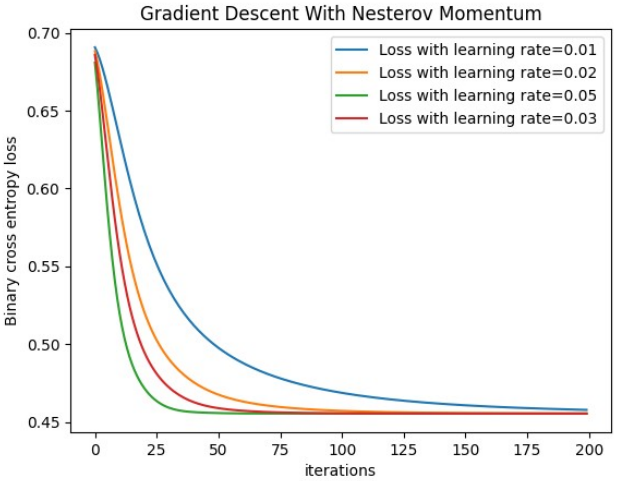
Gradient Descent With Nesterov Momentum - Nesterov Momentum is an enhanced version of gradient descent that uses a "momentum" term to speed up the optimization process.

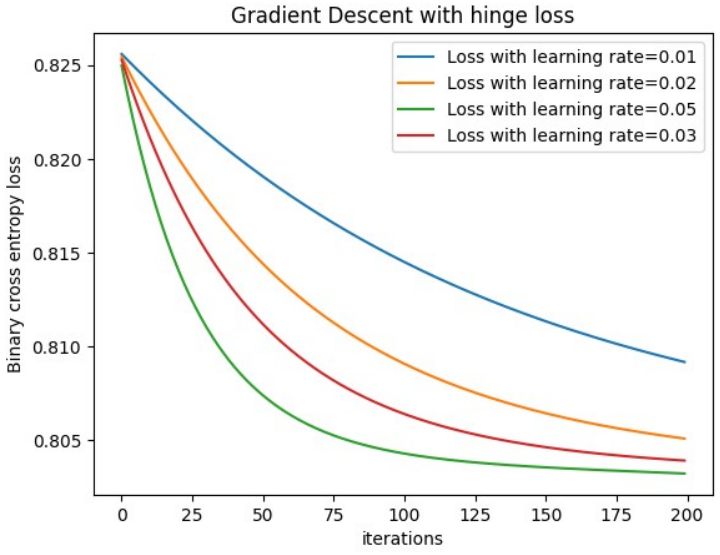
- Update

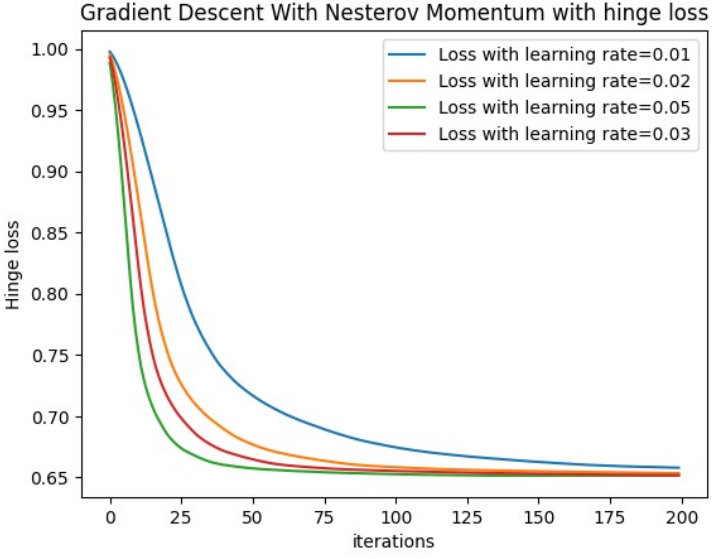
$$v = gamma * v + learning\_rate * gradient\_w$$
$$w = w - gamma * v + learning\_rate * gradient\_w$$

- 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
- 'gradient_w' is the gradient of the cost function with respect to 'w'


Gradient Descent
Loss with learning rate=0.01
Loss with learning rate=0.02
Loss with learning rate=0.05
Loss with learning rate=0.03
Binary cross entropy loss
iterations


Gradient Descent With Nesterov Momentum
Loss with learning rate=0.01
Loss with learning rate=0.02
Loss with learning rate=0.05
Loss with learning rate=0.03
Binary cross entropy loss
iterations


Gradient Descent with hinge loss
Loss with learning rate=0.01
Loss with learning rate=0.02
Loss with learning rate=0.05
Loss with learning rate=0.03
Binary cross entropy loss
iterations


Gradient Descent With Nesterov Momentum with hinge loss
Loss with learning rate=0.01
Loss with learning rate=0.02
Loss with learning rate=0.05
Loss with learning rate=0.03
Hinge loss
iterations

Obserevation

- The data set is linearly separable, because Gradient descent is working similarly for cross-entropy and hinge loss in logistic regression.
- The loss function is convex.

Thank You