

Assignment 3 - report - Soft Computing

Mtech - 2nd Sem

K.Ranjith - MIT2020017

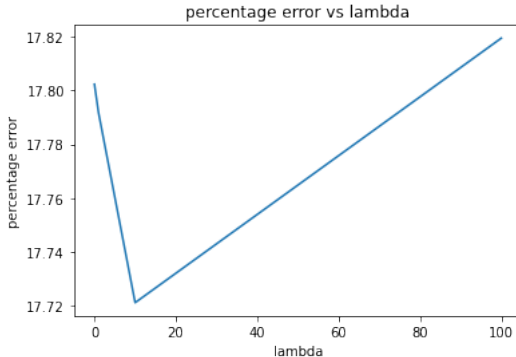
01-02-2021

IIIT Allahabad

CLICK HERE

problem 1a

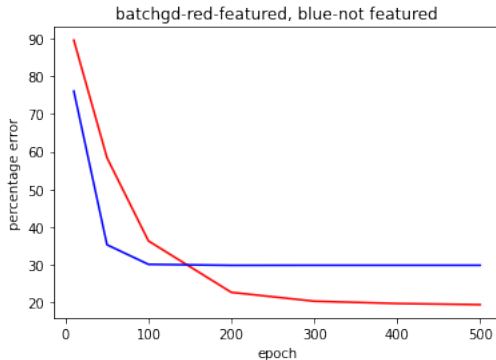
plot of percentage error wrt to λ



1. $\lambda = 10$, we will get best model with percentage error in prediction 17.7211
2. however for no regularization i.e $\lambda = 0$, error percentage is 17.802
3. so after regularization we get better model

problem 1b - batch gradient decent

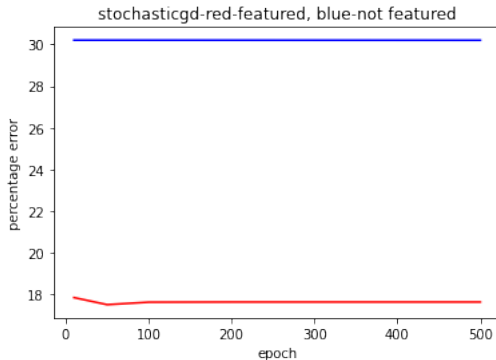
hyperparameters [no-train = 382, no-test = 163, alpha-normal = 0.000000001,
alpha-featured = 0.01,]



1. we will get a better model after feature scaling as the model converges to 18.2 percentage error
2. so we can conclude that after feature scaling performance of gradient decent increases

problem 1b - stochastic gradient decent

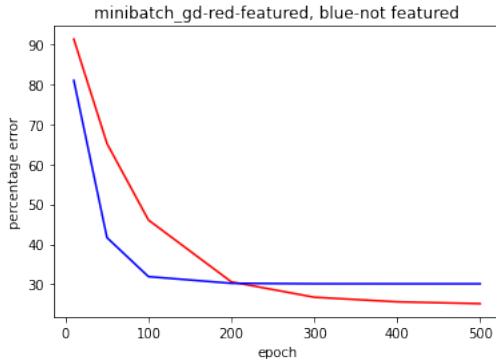
hyperparameters [no-train = 382, no-test = 163, alpha-normal = 0.000000001,
alpha-featured = 0.01]



1. we will get a better model after feature scaling as the model converges to 17.8 percentage error
2. so we can conclude that after feature scaling performance of stochastic gradient decent increases and convergence is faster

problem 1b - minbatch gradient decent

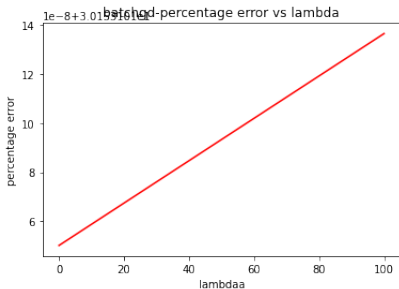
hyperparameters [no-train = 382, no-test = 163, alpha-normal = 0.000000001,
alpha-featured = 0.01, batch-size = 100]



1. we will get a better model after feature scaling as the model converges to 24.44 percentage error
2. so we can conclude that after feature scaling performance of stochastic gradient decent increases and convergence is faster

problem 1c - gradient decent

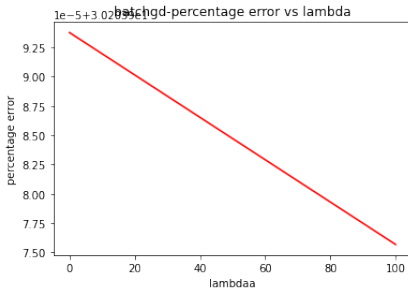
hyperparameters [no-train = 382, no-test = 163, alpha= 0.000000001, lambda = [0, 0.1, 1, 10, 100]]



1. adding regularization only leads to more time complexity and doesn't yield good results
2. even after regularization percentage errors don't change much. for $\lambda = [0, 0.1, 1, 10, 100]$ percentage error = [30.153101050116145, 30.153101050202512, 30.153101050979718, 30.153101058752014, 30.153101136474838]

problem 1c -stochastic gradient decent

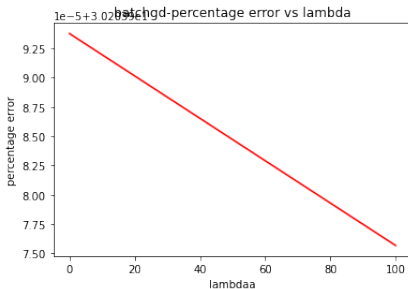
hyperparameters [no-train = 382, no-test = 163, alpha= 0.000000001, lambda = [0, 0.1, 1, 10, 100]]



1. adding regularization only leads to more time complexity and doesnt yeild good results
2. even after regularization percentage errors doesnt change much. for lambda =[0, 0.1, 1, 10, 100] percentage error = [30.203993775198366, 30.203993757100566, 30.203993594220297, 30.203991965400824, 30.203975675525893]

problem 1c -mini batch gradient decent

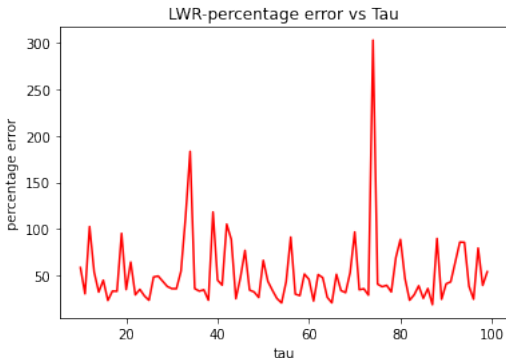
hyperparameters [no-train = 382, no-test = 163, alpha= 0.000000001, lambda = [0, 0.1, 1, 10, 100]]



1. adding regularization only leads to more time complexity and doesnt yeild good results
2. even after regularization percentage errors doesnt change much. for $\lambda = [0, 0.1, 1, 10, 100]$ percentage error = [31.892907662751604, 31.89290766345898, 31.892907669825433, 31.8929077334899, 31.892908370134418]

problem 1d -LWR algorithm

hyperparameters [no-train = 382, no-test = 163, alpha= 0.000000001]



1. lowest percentage error is = 18.842937027158193 for tau value = 87
2. model with normal equations (problem 1a) works better than this model remaining all models (problem 1b,problem 1c) works poorly comparative to this model