LING 572
HW4
Author: Dhwani Serai

Q1:

Table 1: Zero-order methods

Expt id	Func name	α	N	Method id	w^0	n	w^n	$func(w^n)$
Z1	f1	0.05	200	1	(1,2)	200	(0.01522, 0.01618)	2.00049
Z2	f1	0.05	200	2	(1,2)	61	(-3.19189e-16 -1.20736e-15)	2.0
Z3	f1	0.05	200	3	(1,2)	82	(-3.19189e-16 -1.20736e-15)	2.0
Z4	f1	0.05	200	1	(1,2.03)	200	(-0.00061 -0.00116)	2.00000
Z5	f1	0.05	200	2	(1,2.03)	63	(-3.19189e-16,-0.02000)	2.0004
Z6	f1	0.05	200	3	(1,2.03)	84	(-3.19189e-16,-0.02000)	2.0004
Z7	f1	0.05	200	grad	(1,2)	200	(7.05509e-10,1.41102)	2.0
Z8	f1	0.05	200	grad	(1,2.03)	200	(7.05509e-10,1.4322)	2.0

Z9	f1	0.05	200	1	(100, 20)	200	(91.80330, 15.00652)	8655.04175
Z10	f1	0.05	200	2	(100, 20)	200	(90.00,20.0)	8502.0
Z11	f1	0.05	200	grad	(100, 20)	200	(7.05508e-08,1.41102e-08)	2.0

- (a) Z1-Z3: At the end of iterations, which of the three methods approach got close, but does not reach, a minimum point? Why? Which reaches the minimum point with the least number of iterations? Why?

  Ans: The random search method got very close but did not reach the absolute minimum because the approach for finding the minimum is very random and it is also not replicable. It also suffers from the curse of dimensionality problem due to which the probability of getting to a global minima is just 50% for a quadratic equation.

  The method which gets to the optimal value in the least number of iterations is coordinate descent. That is the case because it chooses the best descent direction at every step by analyzing the cost in each coordinate axis of the input. Coordinate search computes more values at every step than coordinate descent but from the results obtained above it converges in less number of iterations.

- (b) Z4-Z6: At the end of iterations, which methods did not reach the minimum point? Why? Which method has the lowest function value? Why?

  The coordinate search and coordinate descent methods did not reach a minimum point in this possibly because they only search for a minima with respect to the input coordinate axes at each step whereas random search checks for random directions from a given point.

  Random search has the lowest function value because it searches for a minimum point in random directions and not just the input coordinate axis. So for the given starting point random search gave a more optimal solution.

- (c) Z7-Z8: Does gradient descent reach the minimum point? Why or why not?

  Yes, for both of these experiments gradient descent search does reach a minimum point. It is easy for Gradient descent to reach an optimal value for some functions. It may

sometimes take more iterations to converge because it slows down near a minima or maxima but if we choose a good value of learning_rate then it becomes faster to converge.

- (d) Z9-Z11: Which method reaches the minimum point?

  The gradient descent method reaches the minimum point because it works by calculating the derivative of the function at each point in the graph. Whereas random search and coordinate search go step by step so it may take longer time to converge based on the input parameters.

- (e) Z1-Z11: What conclusion can you draw from those experiments?

  Learning rate, input coordinates are also very important parameters in finding the optimal point in addition to the type of method being used. Some method may work for one case but not for another case. So, every parameter needs to be taken into consideration while trying to find a minima for a function.

Q4:

Table 2: Gradient Descent Results with α = 0.01 and N = 200, but different w^0

Expt id	Func name	w^0	converge?	w^{200}	$func(w^{200})$
E1	g1	-1.8	yes	-2.61799	-0.9999
E2	g1	1.8	yes	1.57079	-1.0
E3	g2	-1.8	yes	-2.56080	-0.32954
E4	g2	1.8	yes	1.53658	-0.75862
E5	g3	-1.8	yes	-0.031658	0.20100
E6	g3	1.8	yes	0.03166	0.20100
E7	g4	-1.8	yes-but-diverge	-1.83067	-6.13520
E8	g4	1.8	yes	0.15081	0.00343
E9	g5	-1.8	yes	-1.43459	-0.16104

E10	g5	1.8	no	0.91235	0.2129
E11	g6	-1.8	yes	0.77265	1.00001
E12	g6	0.77	yes	0.76882	1.00000
E13	g6	0.05	yes	0.22746	1.00000
E14	g6	0.3	no	0.39961	0.87587
E15	g6	0.76	no	0.61827	0.91177
E16	f1	(1,2)	yes	(0.01759, 0.03517)	2.00155
E17	f2	(1,2)	yes	(0.01758,2.24502e-18)	0.00031
E18	f3	(1,2)	yes-but-diverge	(0.44617, 5861739789737785.0)	4.12319e+33
E19	f4	(1,2)	yes-but-diverge	(0.90236,4.00489e+36	1.92470e+36

Q5:

What's your observation when coming the experiments in Q4? One-sentence answers should be sufficient. For instance, for (a), you can simply say that different w_0 values lead to different global minima.

(a) E1 and E2: different w^0 values have led to a minima at different points with same values

(b) E3 and E4: different w^0 has led to different minima

(c) E5 and E6: different w^0 has lead to the same minima value at different points

(d) E7: w^0 value has led the algorithm away from the minima

(e) E8: w^0 has converged to a possible minima

(f) E9 and E10: different w^0 values led the algorithm to different points

(g) E11: the given w^0 value has converged to a possible global minima

(h) E12 and E13: both w^0 values have converged to same minima value at different points

(i) E14: this w^0 value did not converge

(j) E12 and E15: different w^0 values led to different func(w200) values one converged and one didn't

(k) E11-E15: different w^0 values led to different function values. Some converged but some didn't

(l) E16-E19: the same w^0 value did not work for all 4 functions. It worked for some but didn't work for others

(m) Summarize your findings from E1-E19: the function type and starting point (w^0) matters in case of gradient descent. Different starting points work for different functions

Q6:

- What conclusion can you draw from Expt L1-L6?

  Ans: The learning rate plays a big role in the convergence of gradient descent algorithm. Small learning rates are more probable to converge but if the rate is too small then it will be very slow. At the same time larger learning rates may not converge anyway.

Table 3: Gradient Descent Results with w_0 = 100 and N = 200, but different learning rate α

Expt id	Func name	α	converge?	w^{200}	$func(w^{200})$
L1	g3	0.01	no	1.75879	3.29335
L2	g3	0.1	yes	4.15540e-15	0.2
L3	g3	0.3	yes	1.91500e-15	0.2
L4	g3	0.6	yes	1.92032 e-15	0.2
L5	g3	1.0	no	99.99999	10000.1999
L6	g3	1.01	no	5248.4897	27546644.72828