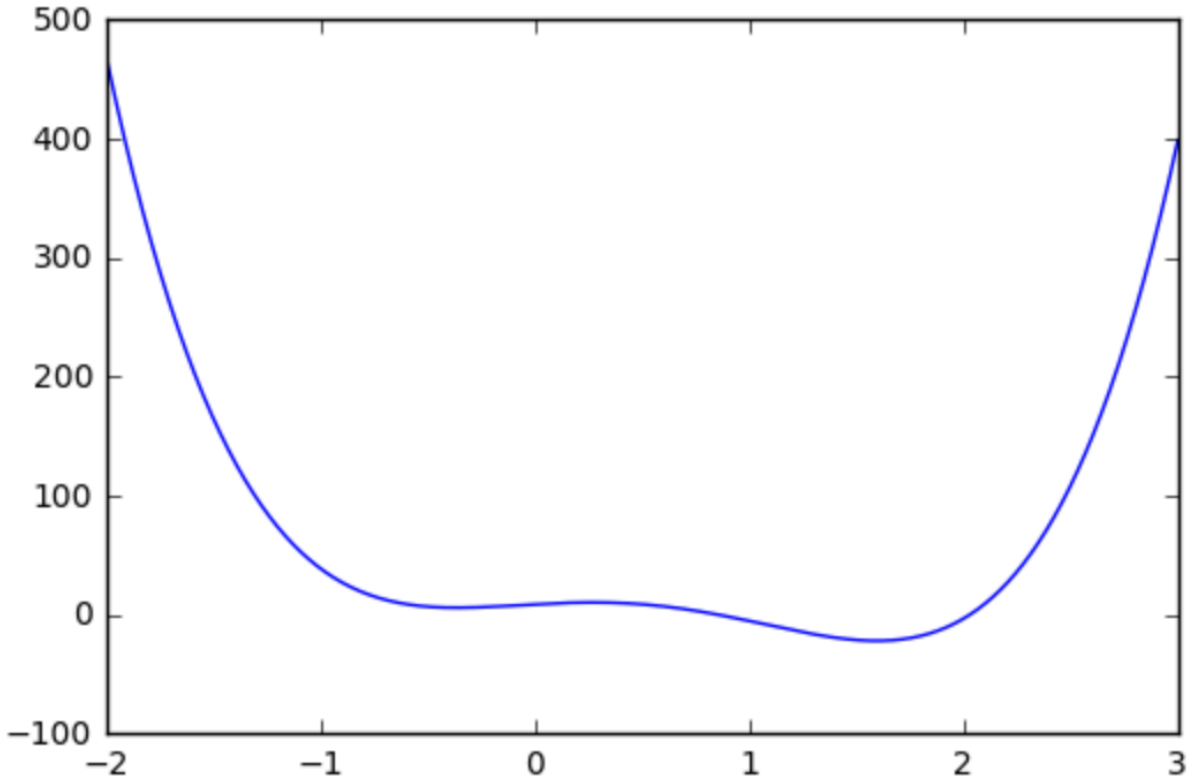
Part-II Programming Exercises

Mohammed Ilyas - mih278
Faiyaz Sundrani - 1459

1.

a) **Local Minimum:** -0.364336433643
Global Minimum: 1.59535953595



b) **When x = -1 and eta = 0.001 and only 5 iterations**

```
Before entering the iteration Value of x is: -1 Value of f is: 39
At the end of iteration 1 Value of x is: -0.866 Value of f is: 24.122076702976
At the end of iteration 2 Value of x is: -0.776294662656 Value of f is: 17.196933470898333
At the end of iteration 3 Value of x is: -0.7109220204970491 Value of f is: 13.432358919601501
At the end of iteration 4 Value of x is: -0.6607817416581312 Value of f is: 11.182090980505741
At the end of iteration 5 Value of x is: -0.6209723259625907 Value of f is: 9.746936730571225
```

When x = -1, eta = 0.001, and running for 1000 iterations

```
At the end of iteration 996 Value of x is: -0.36422374257061546 Value of f is: 6.124226461427285
At the end of iteration 997 Value of x is: -0.36422374257061546 Value of f is: 6.124226461427285
At the end of iteration 998 Value of x is: -0.36422374257061546 Value of f is: 6.124226461427285
At the end of iteration 999 Value of x is: -0.36422374257061546 Value of f is: 6.124226461427285
At the end of iteration 1000 Value of x is: -0.36422374257061546 Value of f is: 6.124226461427285
```

The value of x has converged to the local minimum.

c)

When x = 2, eta = 0.001, and running for 1000 iterations

```
Before entering the iteration Value of x is: 2 Value of f is: -3
At the end of iteration 1 Value of x is: 1.894 Value of f is: -12.280893572864002
At the end of iteration 2 Value of x is: 1.823848257024 Value of f is: -16.471988590606028
At the end of iteration 3 Value of x is: 1.7740854376290425 Value of f is: -18.621275496013634
At the end of iteration 4 Value of x is: 1.7372611541368062 Value of f is: -19.813420352480247
At the end of iteration 5 Value of x is: 1.7092287072971393 Value of f is: -20.51063658566802
At the end of iteration 996 Value of x is: 1.5953147000510859 Value of f is: -21.696233024904366
At the end of iteration 997 Value of x is: 1.5953147000510859 Value of f is: -21.696233024904366
At the end of iteration 998 Value of x is: 1.5953147000510859 Value of f is: -21.696233024904366
At the end of iteration 999 Value of x is: 1.5953147000510859 Value of f is: -21.696233024904366
At the end of iteration 1000 Value of x is: 1.5953147000510859 Value of f is: -21.696233024904366
```

d)

When x = -1, eta = 0.01, and running for 1000 iterations:

```
Before entering the iteration Value of x is: -1 Value of f is: 39
At the end of iteration 1 Value of x is: 0.3400000000000001 Value of f is: 10.43128576
At the end of iteration 2 Value of x is: 0.38022144000000013 Value of f is: 10.221092268269524
At the end of iteration 3 Value of x is: 0.44466297051775 Value of f is: 9.676878465941176
At the end of iteration 4 Value of x is: 0.5493557211358288 Value of f is: 8.231163257954496
At the end of iteration 5 Value of x is: 0.7208664355233224 Value of f is: 4.384921376586129
At the end of iteration 996 Value of x is: 1.5953147000510854 Value of f is: -21.69623302490436
At the end of iteration 997 Value of x is: 1.5953147000510852 Value of f is: -21.69623302490436
At the end of iteration 998 Value of x is: 1.5953147000510854 Value of f is: -21.69623302490436
At the end of iteration 999 Value of x is: 1.5953147000510852 Value of f is: -21.69623302490436
At the end of iteration 1000 Value of x is: 1.5953147000510854 Value of f is: -21.69623302490436
```

We observe that x has converged to the global minimum.

e)

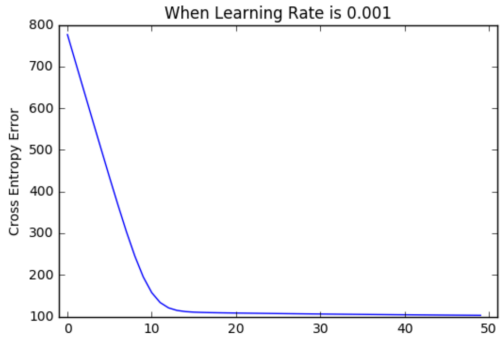
```
Before entering the iteration Value of x is: -1 Value of f is: 39
At the end of iteration 1 Value of x is: 5.7 Value of f is: 10769.505600000002
At the end of iteration 2 Value of x is: -426.9056000000001 Value of f is: 533919847405.0095
At the end of iteration 3 Value of x is: 249843171.64200968 Value of f is: 6.234331865204753e+34
At the end of iteration 4 Value of x is: -4.9905961701560285e+25 Value of f is: 9.924981332099234e+103

---------------------------------------------------------------------------
OverflowError                             Traceback (most recent call last)
<ipython-input-216-13f9ff800ec5> in <module>()
----> 1 apply_grad_descent(Symbol('x'), -1, (16*(x**4))-(32*(x**3))- (8*(x**2))+ (10*x) + 9, 0.05, 100)

<ipython-input-212-038b5f537ff8> in apply_grad_descent(x, xval, f, step_size, iterations)
     30
     31         if i > 995 or i <= 5:
---> 32             print("At the end of iteration", i, "Value of x is:", xval, "Value of f is:", int_f2(xval))
     33
     34

/Applications/anaconda/lib/python3.5/site-packages/numpy/__init__.py in <lambda>(_Dummy_12062)

OverflowError: (34, 'Result too large')
```

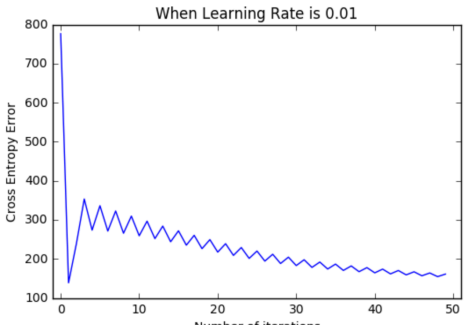
We see that the value of x does not converge. This is because the learning rate is too high. And because of this, the value of x keeps bouncing to different points, skipping the converging point all together.

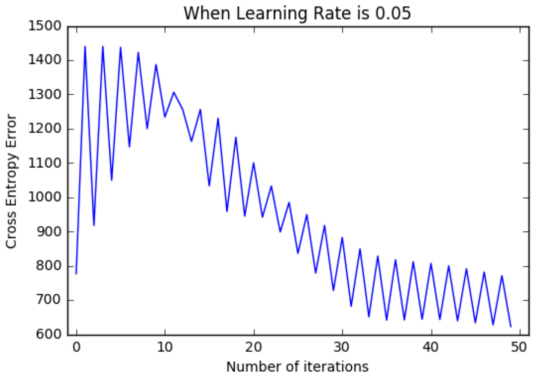
2.

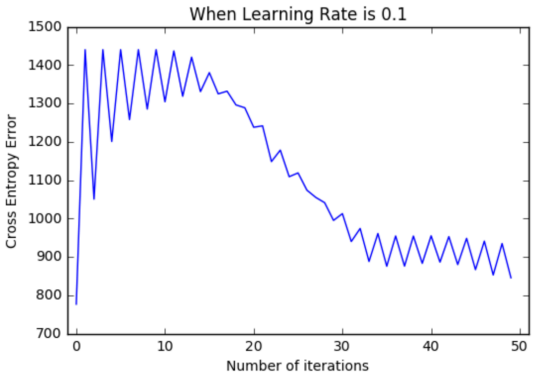
a)

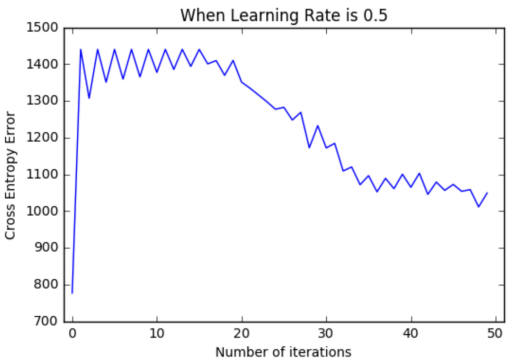
i.


When Learning Rate is 0.001
Cross Entropy Error


When Learning Rate is 0.01
Cross Entropy Error


When Learning Rate is 0.05
Cross Entropy Error
Number of iterations


When Learning Rate is 0.1
Cross Entropy Error
Number of iterations


When Learning Rate is 0.5
Cross Entropy Error
Number of iterations


When Learning Rate is 1.0
Cross Entropy Error
Number of iterations


When Learning Rate is 1.5
Cross Entropy Error
Number of iterations

ii. and iii.

η	0.001	0.01	0.05	0.1	0.5	1.0	1.5
Cross-Entropy Error (training)	102.695	152.485	759.863	930.2209	1024.835	1047.44	1056.3445
Classification error (training)	28.33%	36.11%	36.11%	36.66%	36.66%	36.66%	36.66%
$\|\|w\|\|_2$	2.615	6.596	32.759	65.007	324.261	649.271	973.892

b)

i. and ii.

λ	0	0.05	0.1	0.2	0.3	0.4	0.5
Cross-Entropy Error (training)	102.695	102.884	103.272	103.443	103.809	104.168	104.522
Classification error (training)	28.33%	28.33%	28.33%	28.33%	27.77%	27.77%	27.77%
Classification error (cross-validation)	33.33%	33.33%	33.33%	32.77%	32.77%	32.77%	32.77%
$\|\|w\|\|_2$	2.6156	2.609	2.6038	2.5921	2.5804	2.5688	2.557

c)

For L2 Norm: As the lambda increases, it penalizes (suppresses, decreases) the weights. Thus, we see that L2 norm (which is the square root of sum of squares of w's) decreases as lambda increases

For Cross-Entropy Error: As the lambda increases, the cross-entropy error increases. This is because when we increase the lambda, we are making our model simpler. As our model becomes simpler, bias is introduced. As the model becomes more biased, its error function (the cross entropy error) increases (though marginally in this case).

For Cross-Validation Error: We observe that as lambda increases, the cross validation classification error decreases. As we increase the lambda, we reduce "overfitting", thus resulting in less error.

3.

Running cross validation to choose a lambda which yields the smallest classification error is a good approach. However, there are some concerns that should be addressed:

i) Running cross validation can be computationally expensive, especially for k-fold cross validation. As the value of k increases, the slower the process will be

ii) For cross validation to work well, we need to have a large enough validation dataset

iii) If we have prior information available for a model, then it is better to use methods such as Akaike's Information Criterion (AIC), Bayesian Information Criterion (BIC), Structural Risk Minimization (SRM), and Minimum Description Length (MDL), as these methods incorporate the use of priors. Cross validation on the other hand, makes no prior assumption about the model

iv) If we perform attribute subset selection (select a subset of important attributes from a set of total attributes) and then use cross validation, cross validation may show biased error rate (eg. 3% instead of the true error rate 50%). This is because during attribute subset selection, the attributes have already seen the supposedly "never before seen samples".
Thus, when we evaluate the model using cross validation on a "never before seen dataset" (i.e., the test set), cross validation outputs biased error rate.

To prevent this, attribute subset selection should be performed inside cross validation and not before it.