

# 최신컴퓨터 특강 2

- Linear Regression -

과 목 명	최신컴퓨터 특강2
제출일자	2018. 10. 23.
조 원	전성배
학 번	201302476

## 과제 설명

1-1. code 2를 이용하여 다음의 데이터를 가지고 학습시킨다.

X=[1,2,3,4,5,6,7,8,9,10], Y=[2.2, 5.2, 6.1, 7.9, 10.5, 11.8, 15, 16, 18.2, 20]

```
C:\Users\JeonSeongBae\Anaconda3\python.exe "C:/Users/JeonSeongBae/Pycha
C:\Users\JeonSeongBae\Anaconda3\lib\site-packages\h5py\__init__.py:36:
    from ._conv import register_converters as _register_converters
2018-10-23 16:05:19.710292: I T:\src\github\tensorflow\tensorflow\core\
0 426.5167 [-0.94719553] [-0.5965721]
20 16.424139 [1.4714271] [-0.24336739]
40 0.8920525 [1.9414513] [-0.17010888]
60 0.3023106 [2.0322757] [-0.15136689]
80 0.27847236 [2.0493119] [-0.14327161]
100 0.27609086 [2.0519934] [-0.13728541]
120 0.27454698 [2.0518868] [-0.13174656]
140 0.27305856 [2.051243] [-0.12633134]
...
3840 0.18921804 [1.9788156] [0.37812436]
3860 0.18915279 [1.9786533] [0.3792554]
3880 0.18908852 [1.978492] [0.38037702]
3900 0.18902545 [1.9783323] [0.38148922]
3920 0.18896334 [1.9781741] [0.38259214]
3940 0.1889021 [1.9780167] [0.3836858]
3960 0.18884218 [1.9778612] [0.38477036]
3980 0.18878302 [1.9777063] [0.3858458]
4000 0.18872502 [1.9775535] [0.3869123]

Process finished with exit code 0
```

- W와 b를 Variable로 설정하여 변경되는 값을 저장할 수 있도록 한다.
- X = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10], Y = [2.2, 5.2, 6.1, 7.9, 10.5, 11.8, 15, 16, 18.2, 20]를 지나는 최소 cost의 직선을 찾는다.
- GradientDescentOptimizer를 통해 learning\_rate는 0.001로 설정하여 모델을 구한다.

- 1-2.  $w : [1,3]$  ,  $b : [0,2]$  범위 값에서 0.1 간격으로 cost값을 계산해보고 그 중 최소 cost를 가지는  $w$ ,  $b$ 를 찾아 1-1과 비교해보시오.

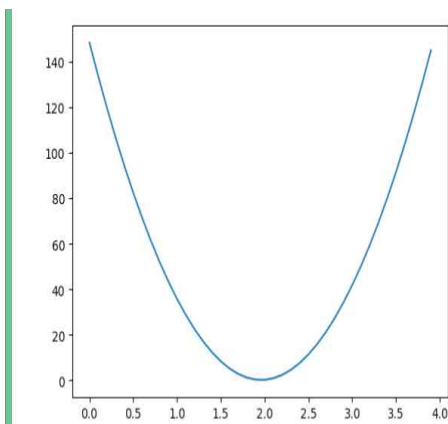
```
# print
print(cost_val[tf.argmin(cost_val).eval(session=sess)])
print(w_val[tf.argmin(cost_val).eval(session=sess)])
print(b_val[tf.argmin(cost_val).eval(session=sess)])

2주차 실습3 x
C:\Users\LeonSeongBae\Anaconda3\lib\site-packages\h5py\__init__
from ._conv import register_converters as _register_conv
2018-10-23 16:09:57.256972: I T:\src\github\tensorflow\tens
0.19899979
2.0
0.3

Process finished with exit code 0
```

- $X = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]$ ,  $Y = [2.2, 5.2, 6.1, 7.9, 10.5, 11.8, 15, 16, 18.2, 20]$ 를 지나는 최소 cost의 직선을 찾는다.
- 반복문을 10 - 30, 0 - 20으로 0.1로 나누어  $w$ 는 1 - 3,  $b$ 는 0 - 2의 범위를 0.1의 간격으로 반복할 수 있도록 한다.
- GradientDescentOptimizer를 통한 방법이 아닌  $w$ 와  $b$ 를 바꾸며 최소 cost를 구한다.

2. 아래의 데이터를 가지고 cost 함수의 그래프를 그려보시오 ( $w : [0,4]$  범위에서  $b = 0.5$ 로 고정)  $X=[1,2,3,4,5,6,7,8,9,10]$ ,  $Y=[2.2, 5.2, 6.1, 7.9, 10.5, 11.8, 15, 16, 18.2, 20]$



- feed\_dict를 통해  $b$ 는 0.5로 고정한다.
- 반복문을 통해  $w$ 의 범위를 0 - 4의 범위로 한정한다.
- 2.0 보다 조금 작은 부분에 기울기가 0인 부분(최소 cost지점)이 형성된다.
- 2차방정식 곡선이 형성되는 것을 확인 할 수 있다.

3 . data.csv 파일의 데이터를 읽어 4/5를 trainingset으로 하고 나머지를 testset으로 해서, trainingset으로 linearregression을 수행하여 testset에 대한 처음 다섯행은 예측 값과 실제 값을 프린트하고, 전체 cost의 평균값을 프린트하는 코드를 작성하고 테스트하시오.

```

2주차 실습6 x
C:\Users\JeonSeongBae\Anaconda3\python.exe
C:\Users\JeonSeongBae\Anaconda3\lib\site-packages\
from ..conv import register_converters
(10000, 5) [[96. 86. 30. 12. 52.]
[97. 86. 34. 13. 42.]
[99. 83. 38. 11. 38.]
...
[98. 90. 35. 10. 55.]
[92. 84. 32. 15. 50.]
[95. 88. 35. 15. 35.]] 10000
(10000, 1) [[90.508]
[87.308]
[84.808]
...
[94.808]
[91.408]
[85.408]]
2018-10-23 16:12:34.265838: I T:\src\github
Training Complete
0 Cost: 2.8271093
Prediction:
[[91.15638 ]
[92.020195]
...
Prediction:
[[91.53203 ]
[92.11051 ]
[89.57823 ]
...
[96.879234]
[90.24466 ]
[84.07188 ]]

2000 Cost: 1.7024407
Prediction:
[[91.53361 ]
[92.11103 ]
[89.579834]
...
[96.87755 ]
[90.24463 ]
[84.07258 ]]

...
Process finished with exit code 0

```

- .loadtxt 메소드를 통해 data.csv의 파일을 읽는다.
- dataset으로 설정할 x와 y 데이터를 8,000개(전체의 4/5), testdataset으로 설정할 x와 y 데이터를 2,000개씩(전체의 1/5) 각각 저장한다.
- GradientDescentOptimizer를 통해 최소 cost를 구한다.
- Traing set을 통해 먼저 traing 시킨 뒤 Training Complete를 출력하여 구분한다.
- Test set을 통해 cost와 Prediction 값을 출력한다.
- 각각의 반복문에 들어가는 X와 Y데이터는 다른 값이며 test는 traing된 모델에 의해 testing된다.