Assignment#1

2017029970 우원진

1. Vectorized Version vs Element-Wise Version 시간비교

```
uwonjin-ui-MacBookPro:딥러닝 과제 woowonjin$ python3 practice#1.py
Vectorized-Version consuming time : 0.06972575187683105
Element-Wise-Version consuming time : 11.772079944610596
```

Training Data 갯수 : 1000개, Learning-Iterate 횟수 : 2000회 에 대해 Vectorized-Version과 Element-Wise-Version의 학습하는데 사용한 시간은 위와 같다.(생각보다 시간차가 너무 많이 나서 놀랐다.)

2. W & b

```
[w1, w2, b, J] = [1.0447388369807455, 1.0282505891403702, -0.533197198897953, 0.03449354270459478]
[w1, w2, b, J] = [1.046425267743285, 1.0299251199379802, -0.5319174495571165, 0.0344302255556443]
[w1, w2, b, J] = [1.0481057399904516, 1.0315938823129256, -0.5306419453000102, 0.03436734477280504]
[w1, w2, b, J] = [1.0497803052658774, 1.0332569254181148, -0.5293706586870945, 0.034304895316594364]
[w1, w2, b, J] = [1.0514490144220037, 1.0349142977528676, -0.5281035625786733, 0.03424287222966557]
[w1, w2, b, J] = [1.0531119176326242, 1.0365660471747118, -0.5268406301301444, 0.03418127063507489]
[w1, w2, b, J] = [1.0547690644051404, 1.0382122209109113, -0.5255818347873501, 0.0341200857345933]
[w1, w2, b, J] = [1.0564205035925385, 1.0398528655697303, -0.5243271502820221, 0.034059312807062016]
[w1, w2, b, J] = [1.0580662834050927, 1.0414880271514464, -0.5230765506273207, 0.03399894720679041]
[w1, w2, b, J] = [1.0597064514218075, 1.043117751059113, -0.5218300101134644, 0.03393898436199379]
[w1, w2, b, J] = [1.0613410546016027, 1.044742082109082, -0.5205875033034497, 0.033879419773272204]
[w1, w2, b, J] = [1.0629701392942457, 1.046361064541294, -0.5193490050288563, 0.03382024901212608]
[w1, w2, b, J] = [1.0645937512510433, 1.047974742029338, -0.5181144903857366, 0.03376146771951107]
[w1, w2, b, J] = [1.066211935635297, 1.0495831576902914, -0.5168839347305891, 0.0337030716044268l]
[w1, w2, b, J] = [1.0678247370325256, 1.051186354094342, -0.5156573136764098, 0.03364505644254333]
[w1, w2, b, J] = [1.0694321994604676, 1.0527843732742015, -0.514434603088824, 0.03358741807485923]
[w1, w2, b, J] = [1.0710343663788597, 1.0543772567343135, -0.5132157790822941, 0.033530152406394564]
[w1, w2, b, J] = [1.0726312806990086, 1.0559650454598626, -0.5120008180164004, 0.033473255404914644]
[w1, w2, b, J] = [1.0742229847931524, 1.0575477799255895, -0.5107896964921965, 0.03341672309968626]
[w1, w2, b, J] = [1.075809520503620l, 1.0591255001044153, -0.5095823913486356, 0.03336055158026304]
[w1, w2, b, J] = [1.0773909291518002, 1.060698245475885, -0.5083788796590647, 0.03330473699530084]
[w1, w2, b, J] = [1.0789672515469069, 1.0622660550344294, -0.5071791387277872, 0.03324927555140186]
[w1, w2, b, J] = [1.0805385279945718, 1.0638289672974512, -0.5059831460866927, 0.033194163511985356]
[w1, w2, b, J] = [1.082104798305249, 1.0653870203132447, -0.5047908794919485, 0.03313939719618727]
[w1, w2, b, J] = [1.0836661018024414, 1.0669402516687447, -0.5036023169207575, 0.033084972977784l]
[w1, w2, b, J] = [1.0852224773307575, 1.0684886984971171, -0.5024174365681745, 0.033030887284143697]
```

사진과 같이 w1, w2는 거의 비슷한 값을 가지고, b는 0에 가까워 지려고 하는 것 같다. Training이 지속될수록 Cost값은 작아진다.

3. 경험적으로 알수있는 가장 좋은 Learning Rate

Learning-Rate = 0.01 일때 : Accuracy for the m training data is 99.5%
Accuracy for the n test data is 98.0%

Learning-Rate = 0.1 일때 : Accuracy for the m training data is 99.0% nan 발생
Accuracy for the n test data is 100.0%

Learning-Rate = 1 일때 : Accuracy for the m training data is 99.7% nan 발생
Accuracy for the n test data is 100.0%

Learning-Rate = 0.05 일때 : Accuracy for the m training data is 99.4% nan 발생
Accuracy for the n test data is 100.0%

Learning-Rate = 0.03 일때 : Accuracy for the m training data is 99.0% nan 발생
Accuracy for the n test data is 98.0%

Learning-Rate = 0.02 일때 : Accuracy for the m training data is 98.6% nan 발생
Accuracy for the n test data is 100.0%

Learning-Rate = 0.005 일때 : Accuracy for the m training data is 97.0%

Accuracy for the n test data is 100.0%

Nan발생은 log계산을 할때 로그 안의 값이 0에 가까워서 발생하는것 같다.

Nan발생을 제외하면 Learning-Rate는 0.01일때 가장 무난한것 같다.

4. Accuracy about Learning-Rate(0.01)

	m = 10, n = 100, K = 2000	m = 100, n = 100, K = 2000	m = 1000, n = 100, K = 2000
Accuracy("m" Training Set)	100%	98.0%	98.1%
Accuracy("n" Test Samples)	96.0%	99.0%	99.0%

	m = 1000, n = 100, K = 20	m = 1000, n = 100, K = 200	m = 1000, n = 100, K = 2000
Accuracy("m" Training Set)	89.4%	96.7%	98.1%
Accuracy("n" Test Samples)	88.0%	97.0%	99.0%

5. 이 실험을 통해 알게된 것

데이터 셋을 구축할때 x1, x2값을 랜덤으로 생성하고, x1과 x2를 더한 값이 0을 넘으면 1, 그렇지 않으면 0으로 라벨링을 했다. 이 조건을 생각하고, 결과를 보면 w1과 w2는 서로 같아지려고 하고, b는 0으로 가려는 성질이 있다. 이는 $z = wx + b$ 이고, $a = sigmoid(z)$ 인데, $z = sigmoid(z)$에서 z가 양수면 a의 값이 0.5보다 크고, 음수면 0.5보다 작게 된다. 여기서 테스트를 할때 a의 값이 0.5보다 크면 1이라고 생각하고, 0.5보다 작으면 0으로 생각한다는 점에서, 즉 z가 양수면 1이 되고, 음수면 0이 된다고 볼 수 있다. 여기서 z는 $z = wx + b$ 로 표현 되는데 이 식에서 만약 b가 0이 되고, w의 w1과 w2가 같다면 결국 z의 값은 x1, x2의 영향만 받게 되는데, x1+x2가 0보다 크다면 z도 0보다 크고, x1+x2가 0보다 작다면 결국 z도 0보다 작게된다. 이러한 성질을 따라가기 위해 w1, w2는 같아져 가고, b는 0으로 가는것 같았다.

그리고 또 알게된 점은 딥러닝이라는 분야가 코드를 구축하는데 사용하는 시간보다는 Learning-Rates, Iterate 횟수 등 여러가지 최선의 조건을 찾는데 시간이 더 필요하다는것을 느꼈다.