과제2에 대한 중요 업데이트

0. Weight와 bias 초기화 관련

1차 과제에서 쓴 np.random.random 함수가 아닌 weight는 np.random.randn bias는 np.zeros를 사용해 He 방법으로 초기화 해주십시오.

이유)

1) Np.random.randn은 가우시안 정규 분포에 따른 랜덤값 생성입니다. He 초기화는 가우시안 정규분포를 기반으로 하기에 np.random.randn으로 초기화를 해야합니다.
2) Bias의 값은 보편적으로 초기화 값이 0으로 시작하는게 좋다고 알려져있습니다.

1. minibatch_Optimizer 관련

먼저 minibatch_Optimizer의 Accuracy가 1이 되지 않는다. 라는 질문을 받아서 확인해본 결과 제가 드린 코드와 제 테스트 환경의 차이가 있기에 생기는 문제였습니다. 이 부분에 대해 오해를 야기시킨 점, 그로 인해 과제로 고통받게 한 점에 대해 사과 드리겠습니다.(실제 제 환경은 batch_size= 100, learning_rate = 0.1 이었습니다. 이 부분에 대해 체크가 늦었습니다. Learning_rate가 0.01일땐 epoch을 400번정도 돌리면 training_accuracy 가 1이 되긴 합니다.)

실제로 코드 부분을

trained_minibatch, tmb_train_acc_list, tmb_test_acc_list, tb_loss_list = minibatch_Optimization(dataset, TNN_minibatchOptimizer, 0.01, epoch=100, batch_size=1000)

이 부분에서

trained_minibatch, tmb_train_acc_list, tmb_test_acc_list, tb_loss_list = minibatch_Optimization(dataset, TNN_minibatchOptimizer, **0.1**, epoch=100, **batch_size=100**) 으로 바꾸어주세요.

2.Dropout 관련

Dropout과 관련되서 얘기를 드리자면 Dropout 관련 함수를 만드셔도 되고, Dropout 클래스를 만드셔도 됩니다. 실제로 드롭아웃의 구현방법은 여러 가지 있기에 어떤 방법을 다 써도 됩니다. 드롭아웃과 관련된 채점기준에 대해 말씀드리자면 위의 batchOptimization의 결과값 training_accuracy와 test_accuracy의 차이와 DropoutOptimization의 training_accuracy와 test_accuracy의 차이가 더 적으면 됩니다. (다만 조교는 현재 흔히 알려진 두 방법으로만 테스트 해보았고 두 방식은 제 조건에 맞는 바운드 안에 들어갔습니다.) 단 구현방식에 따라 Accuracy의 차이가 줄어들지 않았다고 하여도 코드 구현방식이 틀리지 않는다면 만점입니다. 만약 Accuracy의 차이가 안 줄어들었다면 주석을 참고해서 채점을 할 것이기에 주석에 제대로 된 설명이 되어 있어야합니다.

(조교가 사용한 방법 1 은 LinearLayer 와 ReLU 사이에 Dropout 클래스를 만들어 LinearLayer의 결과값을 랜덤하게 죽이는 방법을 사용하였고, 사용한 방법 2는 Weight를 킬하는방법이었습니다.)

3. 보고서 관련

Document에 적혀진 것과 실제 2차 과제의 코드와 차이가 있는 부분은 수정을 해도 좋고 안해도 좋습니다.

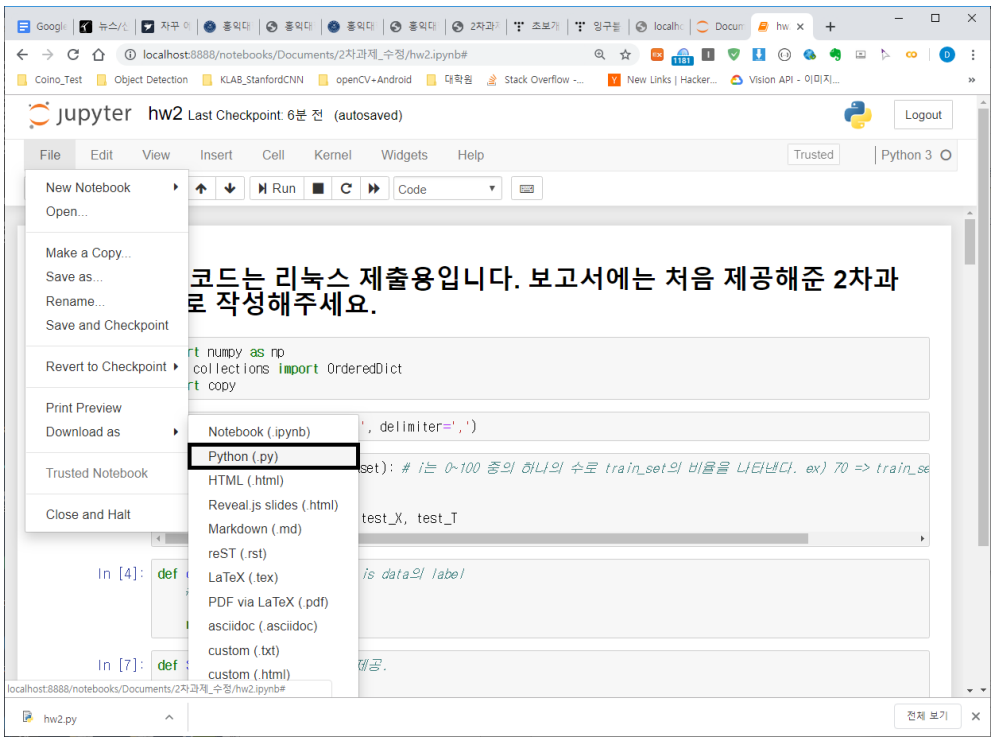
보고서에는 지금 제공된(etc : if I % 10 ==0) 코드와 주석 그리고 트레이닝 할 때 나오는 프린트문을 첨부하셔야 합니다.

물론 프린트 된 결과 전부를 보여줄 필요는 없습니다. **Batch**의 경우 **epoch == 0, 10, 20, 980, 990, 1000**에 해당하는 프린트된 값(Loss, Train Acc, Test_acc)을(**즉 제공해준 코드를 수정하셔야 합니다. Epoch이 1000번 돌아가게) dropout**의 경우 똑같이 **epoch == 0, 10, 20, 980, 990, 1000**에 해당하는 프린트된 값을 스크린샷을 찍어 보고서에 첨부하시면 됩니다.(더 길게 첨부하셔도 문제는 없습니다), **minibatch**의 경우 Train_Accuracy가 1이 되는 순간까지의 프린트된 값을 스크린샷에 찍어 보고서에 첨부하시면 됩니다.

보고서에 들어갈 것은 코드 전체와 거기에 해당하는 주석 그리고 각 Optimization에 대한 프린트된 값의 스크린샷이 있어야합니다.

4. Linux Submit 관련

리눅스에 제출하는 코드는 서버과부하 방지와 채점을 용이하게 하기 위해 Optimization 함수들의 출력부분(print문, list 삭제)과 Learning_rate, batch_size, epoch를 수정하였습니다. 이 코드에 작업하시거나 원래 코드를 copy한 후 이 부분의 양식에 맞게 수정을 하고 hw2.py로 변환시킵니다.



[수정부분 : epoch, learning_rate, print문을 바꾸었고, accuracy를 저장하는 list 삭제(append 삭제)]

Submit 방식은 자신의 리눅스 계정 접속 후 hw2 디렉토리를 만들고(mkdir hw2) 그 안에 hw2.py와 mnist.csv 파일 넣은 후

submit K2019AIS hw2 로 하면 됩니다.

예시)

```
(venv) [K2019AIS@linux2 hw2]$ submit K2019AIS hw2
================================
== SUBMIT ver 2.0.3 ===========
================================
> Connected at 05/19/19 - 04:05AM.
Normal/Submission is possible.
> Here are descriptions.
  * DUE DATE    :  05/22/19 - 04:00PM  <+ 83 H/ 55 M>
  * RECEIVER    :  K2019AIS  </lab/da/K2019AIS/.//hw2/K2019AIS>
  * MULTIPLE    :  yes
  * SUB FOLDER  :  yes
  * FILES IN (/lab/da/K2019AIS/hw2)
        hw2.py mnist.csv
  * WARNING     :  No warning.
> Are you sure to submit? (yes/no) :
```

주의) 현재 학교 리눅스계정에서는 pip를 바로 사용이 불가능합니다. 즉 numpy 라이브러리를 받을 수 없기에 코드가 돌아가는지에 대해 바로 컴파일 해보는 것이 불가능 합니다. 그렇기에 pip를 사용할 수 있는 가상환경을 설정해야 합니다.

자신의 리눅스 계정에 로그인 후 python3 -m virtualenv venv 로 가상환경을 만듭니다.

그 후 source venv/bin/activate 를 실행하면 가상환경으로 설정이 바뀌고 그 후 pip install numpy 로 numpy 라이브러리를 다운받으신 후 submit할 코드를 테스트 하시면 됩니다.

예시)[예시이기에 실제코드를 다 돌린 것이 아닙니다. 여러분들의 hw2.py는 아래와는 다르게 나와야 합니다.]

```
[K2019AIS@linux2 ~]$ source venv/bin/activate
(venv) [K2019AIS@linux2 ~]$ pip install numpy
Collecting numpy
  Using cached https://files.pythonhosted.org/packages/c1/e2/4db8df8f6cddc98
d17bf0c3177ab3cc6beac7f/numpy-1.16.3-cp36-cp36m-manylinux1_x86_64.whl
Installing collected packages: numpy
Successfully installed numpy-1.16.3
(venv) [K2019AIS@linux2 ~]$ cd hw
-bash: cd: hw: 그런 파일이나 디렉터리가 없습니다
(venv) [K2019AIS@linux2 ~]$ cd hw2
(venv) [K2019AIS@linux2 hw2]$ ls
hw2.py  mnist.csv
(venv) [K2019AIS@linux2 hw2]$ python3 hw2.py
8000  Trainset and 2000  Testset
50      번째  Loss =  0.00553
50      번째  Train_Accuracy :  1.0
50      번째  Test_Accuracy :  0.956
------------------------------------------------------------
```

5. 과제제출기한

업데이트가 늦어진 관계로 과제 제출 기간을 금요일에 말했던 것 보다 조금 더 연장하여 **수요일 오후 6시(05/22 18:00) 까지로 하겠습니다.** 클래스넷에 수요일 오후 6시까지 .ipynb(처음 제공하였던 if i%10==0) 파일과 보고서를 자신의 학번_hw2.zip(예시 B211073_hw2.zip)으로 압축해 클래스넷에 올려주시면 되고, hw2.py와 mnist.csv를 위 방식에 맞게 submit 하시면 됩니다. 질문은 화요일 오후 9시 전까지 클래스넷과 메일로 오는 질문에 대해 답해드리겠습니다. 1차과제와 마찬가지로 Late는 없습니다.

마치며 : 과제를 만들어보는 것이 처음이었기에 과제부분과 관련해서 꼼꼼하지 못했던 점 그렇기에 학우분들에게 본의 아닌 고생시킨 점에 대해 다시 한 번 사과드리겠습니다. 3차 과제는 더더욱 꼼꼼히 체크해 이런 문제가 없게 하겠습니다.