**지금 두 계정 차이점 :**

앞으로 구현할것.

~~drop out p = 0.5 -> 0.1로 바꾸기~~

~~train, test 모두 zip파일로 불러오기~~

~~data normalize~~

~~batch normalization 적용~~

여기까지 했을때 성능은 ?  47% -> 51~54% 정확도.

~~model.eval() 추가~~  : 54% -> 60% 정확도

~~train data -> train/valid 분류. (약 8:2)를 통해 overfitting 개선~~

* ~~total\_batch와 train for문 수정~~-> early stopping. 구현

여기서 cuda 메모리초과문제 발생. 따라서 batch size 100 -> 64으로 바꿈

성능 : validation set을 추가하고, 정확도가 60% -> 50%로 낮아졌다.

(원인 : train data양의 감소)

~~validation을 만들기위해서 pokemon\_train dataset을 shuffle하고 8:2로 분할했다. shuffle을 하지 말고 각 label별로 8:2로 분할하면 어떨까?~~

* 정확도 50% -> 56%

~~CNN모델 더 깊게 만들어보기.~~

~~어떻게?~~

~~validation 비율 0.2 -> 0.1~~

~~image resize (64, 128) -> (256, 256)~~

~~layer 5층까지 추가, channel 256->512~~

~~batchsize 64 -> 16 (메모리초과문제)~~

성능변화 : 정확도 감소 56% -> 50%

~~dropout 제거했을 때~~ 성능은? epoch를 반복해도 cost가 전혀 감소하지 않았다. 정확도 0.5%

~~layer 4, 5제거하고, 전결합층 3개 -> 1개로 줄임. (메모리 초과)~~

~~( channel 512 -> 256, fc 256 \* 16 \* 16 -> 151)~~

여기까지 했을때 성능은 ?  정확도 32%됨.

~~이미지 증강 : transform 추가. centercrop등.~~

~~Resize (256 -> 128),~~

~~centercrop(112)정도만 적용해보자.~~

lr 1e-6 으로 수정.overfitting 해결 : dropout 0.1 -> 0.2,

활성화 함수로 ReLU 대신 LeakyReLU를 쓰면?

성능 : 정확도 53%

ResNet 사용해보기

(보류) 이미지 증강으로 GAN사용하면 어떨까?

이미지 증강 심화 : 여러가지 이미지 증강을 비교해보자.

추론 과정에서도 이미지 증강 가능

cross validation

kfold

알아보기

clova ai코드보면서 리팩토링하기.