Idea Factory Intensive Program #2

# 답러닝 홀로서기

#18

이론강의/PyTorch실습/코드리뷰

딥러닝(Deep Learning)에 관심이 있는 학생 발굴을 통한 딥러닝의 이론적 배경 강의 및 오픈소스 딥러닝 라이브러리 PyTorch를 활용한 실습

#### Acknowledgement

#### Sung Kim's 모두를 위한 머신러닝/딥러닝 강의

- <a href="https://hunkim.github.io/ml/">https://hunkim.github.io/ml/</a>
- https://www.youtube.com/playlist?list=PLIMkM4tgfjnLSOjrEJN31gZATbcj\_MpUm

#### Andrew Ng's and other ML tutorials

- https://class.coursera.org/ml–003/lecture
- <u>http://www.holehouse.org/mlclass/</u> (note)
- Deep Learning Tutorial
- Andrej Karpathy's Youtube channel

#### WooYeon Kim & SeongOk Ryu's KAIST CH485 Artificial Intelligence and Chemistry

https://github.com/SeongokRyu/CH485——Artificial—Intelligence—and—Chemistry

SungJu Hwang's KAIST CS492 Deep Learning Course Material

Many insightful articles, blog posts and Youtube channels

#### Facebook community

- Tensorflow KR (<a href="https://www.facebook.com/groups/TensorFlowKR/">https://www.facebook.com/groups/TensorFlowKR/</a>)
- Pytorch KR (<a href="https://www.facebook.com/groups/PyTorchKR/">https://www.facebook.com/groups/PyTorchKR/</a>)

#### Medium Channel and Writers

Toward Data Science (<a href="https://towardsdatascience.com/">https://towardsdatascience.com/</a>)

# Today's Time Schedule

Assignment #2 Review ——— 40 mins

Improvement in DL Optimizer — 40 mins

How to Save Experiment Results

How to Load and Manipulate Experiment Results

How to Visualize Results

1.5 hour

How to Save Experiment Results

#### Experiment

```
# --- Random Seed Initialization --- #
seed = 123
np.random.seed(seed)
torch.manual_seed(seed)
parser = argparse.ArgumentParser()
args = parser.parse_args("")
# ==== #ode | Capacity ==== #
args.in_dim = 3072
args.out_dim = 10
args.hid_dim = 100
args.act = 'relu'
# ==== Regularization ====== #
args.dropout = 0.2
args.use_bn = True
args.12 = 0.00001
args.use_xavier = True
# === Optimizer & Training === #
args.optim = 'FMSprop' #'AMSprop' #SGD, AMSprop, ADAM...
args.lr = 0.0015
args.epoch = 10
args.train_batch_size = 256
args.test_batch_size = 1024
```

Args에 있는 세팅 값들을 저장해야 함

Epoch에 따른 train loss, val loss train acc, val acc

최종 train acc, val acc, test acc도 저장해야 함

각 실험 결과들을 Dictionary에 넣고 리스트에 하나씩 append하자!

각 실험 결과들을 Dictionary에 넣고 리스트에 하나씩 append하자!



만약 중간에 인터넷이 튕기거나 프로세스를 종료해야 하는 상황이 온다면..?

→ 램에 올라가 있단 지금까지의 실험 결과가 다 날라감..

각 실험 결과들을 Dictionary에 넣고 리스트에 하나씩 append하자!



만약 중간에 인터넷이 튕기거나 프로세스를 종료해야 하는 상황이 온다면..?

→ 램에 올라가 있단 지금까지의 실험 결과가 다 날라감..



그럼 하드디스크에 저장하면 중간에 프로세스가 멈춰도 실험 결과들은 남아 있지 않을까?

각 실험 결과들을 Dictionary에 넣고 리스트에 하나씩 append하자!



만약 중간에 인터넷이 튕기거나 프로세스를 종료해야 하는 상황이 온다면..?

→ 램에 올라가 있단 지금까지의 실험 결과가 다 날라감..



그럼 하드디스크에 저장하면 중간에 프로세스가 멈춰도 실험 결과들은 남아 있지 않을까?



어떤 식으로 Dictionary를 하드디스크에 파일 형태로 저장하면 좋을까?

각 실험 결과들을 Dictionary에 넣고 리스트에 하나씩 append하자!



만약 중간에 인터넷이 튕기거나 프로세스를 종료해야 하는 상황이 온다면..?

→ 램에 올라가 있단 지금까지의 실험 결과가 다 날라감..



그럼 하드디스크에 저장하면 중간에 프로세스가 멈춰도 실험 결과들은 남아 있지 않을까?



어떤 식으로 Dictionary를 하드디스크에 파일 형태로 저장하면 좋을까?



JSON 포맷을 활용 하자!

파일 이름은..?

Okay. Dictionary는 JSON형태로 쉽게 저장할 수 있군! 근데 파일 이름은 뭘로 해야하지..?

Okay. Dictionary는 JSON형태로 쉽게 저장할 수 있군! 근데 파일 이름은 뭘로 해야하지..?



시간\_랜덤 숫자.json ? (e.g. 190128\_3242.json)

Okay. Dictionary는 JSON형태로 쉽게 저장할 수 있군! 근데 파일 이름은 뭘로 해야하지..?



시간\_랜덤 숫자.json ? (e.g. 190128\_3242.json)

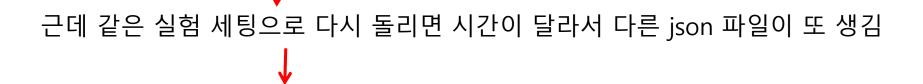


근데 같은 실험 세팅으로 다시 돌리면 시간이 달라서 다른 json 파일이 또 생김

Okay. Dictionary는 JSON형태로 쉽게 저장할 수 있군! 근데 파일 이름은 뭘로 해야하지..?



시간\_랜덤 숫자.json ? (e.g. 190128\_3242.json)

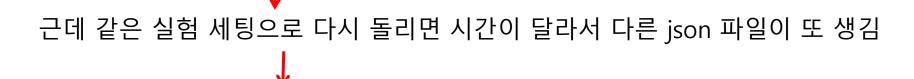


그럼 변수 값들을 파일 제목에 같이 넣어볼까? (e.g. N\_layer\_4\_hid\_dim\_100.json)

Okay. Dictionary는 JSON형태로 쉽게 저장할 수 있군! 근데 파일 이름은 뭘로 해야하지..?



시간\_랜덤 숫자.json ? (e.g. 190128\_3242.json)



그럼 변수 값들을 파일 제목에 같이 넣어볼까? (e.g. N\_layer\_4\_hid\_dim\_100.json)



하이퍼파라미터가 너무 많아지면 최대 파일 이름 길이를 초과함 ㅠ

Okay. Dictionary는 JSON형태로 쉽게 저장할 수 있군! 근데 파일 이름은 뭘로 해야하지..?



시간\_랜덤 숫자.json ? (e.g. 190128\_3242.json)

근데 같은 실험 세팅으로 다시 돌리면 시간이 달라서 다른 json 파일이 또 생김

그럼 변수 값들을 파일 제목에 같이 넣어볼까? (e.g. N\_layer\_4\_hid\_dim\_100.json)



하이퍼파라미터가 너무 많아지면 최대 파일 이름 길이를 초과함 ㅠ



실험 이름 + 실험 세팅 값의 해쉬.json으로 파일 이름을 정하자!

```
Epoch이 달라져도 해쉬 값이 달라짐 ㅠ
Test_batch_size는 실험 결과에 영향이 전혀 없는데 해쉬 값이 달라짐 ㅠ
Del 명령어로 해당 값은 Dictionary에서 지워주자!
```

```
del setting['epoch']
del setting['test_batch_size']
hash_key = hashlib.sha1(str(setting).encode()).hexdigest()[:6]
```