



딥러닝 홀로서기#11

1. How to Parametrize Entire Code

- 어떻게 하이퍼 파라미터를 정의할까?
 - 코드 앞부분에 명시를 해놓고 함수 사용시 Input으로 받아 알아서 모델 생성부터 실험 결과 리포팅까지 가능하도록 만든다
 - 하이퍼 파라미터 값이 Code 안에서 살아있도록 놔두면 안된다
- Why Parametrization?
 - 다양한 변수들을 한곳에서 관리
 - 하이퍼파라미터가 모델 안에 고정되어 있는 상황을 회피 (Safe Coding)
 - 재사용이 가능
 - 가독성이 높아짐 (ex) MLP (200, 100)으로하면 어떤 파라미터인지 모름
 - Auto Hyperparameter Optimizer에게 Parameter 튜닝을 맡기기 위해서
- Which Parameter?
 - Hyperparameter : Non-Trainable Parameter를 말함
 - 일반 Parameter는 Trainable한 성질을 갖고 있어 차이점이 있다
 - Determine Model : 모델을 결정하는 하이퍼파라미터
 - Optimizer Related : 최적화를 결정하는 하이퍼파라미터 (Learning Rate 등)
 - Training, Evaluation Process Related : 학습과 평가를 위한 하이퍼파라미터
 - Device : CPU, GPU Resource를 위한 하이퍼파라미터
 - Saving Exp Result : 실험 결과를 뒀나 하이퍼파라미터

- Introduce Argparse

- Dictionary객체는 사용해야 해야하는데, 이는 typing하기(대괄호, 따옴표) 복잡하므로 ':'으로 입력가능한 객체

```
import argparse

# Argparse Define
parser = argparse.ArgumentParser()
args = parser.parse_args("")

#Input data to argparse
args.num_layer = 5
print(args.num_layer)
print(args)

args.in_dim = 100
print(args)

#Input data from argparse.num_layer to Linear List
linears = []
for i in range(args.num_layer):
    linears.append(i)

print(linears)

#Changing type from Argparse to Dictionary
d = vars(args)
print(d, type(d))
```

- Experiment Manager?

- Function. Experiment의 기능
 - 모델 생성
 - Cost, Loss Function 생성
 - Optimizer 생성
 - Training and Report 진행
 - 평가 결과 레포트