

- 모델:  
/yolox/models/yolox\_p6.py (직접 접근할 일 없음)
- Hyperparameter 설정 값  
batch size : 64  
basic\_lr : 0.01  
더 많은 Setting 값은  
Training Phase 1: exps/p6/yolox\_m\_p6.py  
Training Phase 2: exps/p6/yolox\_m\_p6\_tune.py 에서 확인할 수 있습니다.
- 실험 환경:  
A100 (40GB) x4대, Ubuntu 20.04 (18.04 가능)  
CUDA 11.0, PyTorch 1.9
- 실행 과정 (학습 과정이 복잡하여 최대한 Step 별로 자세히 기술하였습니다.)

### Step1. 설치 및 Dataset 세팅

모든 명령어는 제출 폴더 최상위 Directory 에서 실행됩니다. (setup.py 가 있는 폴더)

#### Package 설치

```
$python setup.py develop
$pip install -e requirements.txt
```

#### Dataset 세팅

```
$ln -s data/coco ./datasets/COCO
```

### Step2. 학습 (총 2 단계로 나눌 수 있습니다.)

#### 2-1. Training Phase1

```
$python tools/train.py -f exps/p6/yolox_m_p6.py -d 4 -b 64 --fp16 -o --cache
```

#### 2-2. move checkpoint file

```
$cp YOLOX_outputs/yolox_m_p6/epoch_280_ckpt.pth .
$mv epoch_280_ckpt.pth pre_m_p6.pth
```

**2-3. Dataset Cache File 지우기** (2 번째 학습 단계에서는 Dataset Image Size 를 다르게 하여 학습하기 때문에, Cache File 을 지워야 합니다.)

```
$rm datasets/COCO/img_resized_cache_train2017.array
```

#### 2-4. Training Phase 2

```
$python tools/train.py -f exps/p6/yolox_m_p6_tune.py -d 4 -b 64 --fp16 -o --cache
```

### 2-5. move checkpoint file

```
$cp YOLOX_outputs/yolox_m_p6_tune/best_ckpt.pth .
```

### 2-5. Pruning 적용 및 Sparse Matrix 변환

```
$python 01_mask_generator.py
```

```
$python 02_direc_pruning.py
```

```
$python 03_jh_merge.py
```

```
$mv merged_49.pth choijhanyangackr/weights/
```

### Step3. 추론

추론 데이터셋 위치는 대회 제출 때의 세팅 입니다. /home/data/agc2021

데이터셋 Directory 변경은 choijhanyangackr/config/yolox\_m\_p6\_sparse.json 에서 변경할 수 있습니다.

```
$cd choijhanyangackr
```

```
$python main.py
```