

ETRI Fashion-How Season 5

모델 Weight 저장 방법 가이드

모델 Weight 저장 경로 및 평가 방법

- 경로: 제출할 모델 폴더 내의 **“model/”** 아래에 저장합니다.
 - 평가 시에는 해당 폴더의 크기를 측정하므로 경로 내에는 저장할 Weight 외의 불필요한 요소가 포함되지 않도록 합니다.
 - 모델은 해당 경로에 있는 Weight만을 이용하여 추론(및 필요한 경우 전/후처리)을 수행해야 합니다.
- 압축 또는 기타의 방법으로 저장된 크기를 변조하지 않습니다. (양자화는 허용)
 - 검증 시에는 파일 크기 및 형식, 파라미터 크기와 weight의 dtype을 이용하여 정상적으로 저장된 Weight인지 확인합니다.
 - 또한 모델이 해당 Weight를 사용하는지, 지정된 폴더 내에 저장되지 않은 Weight나 모델이 있는지 확인합니다.
 - 부정행위 시에는 스코어가 인정되지 않습니다.
- 파일이 여러 개인 경우에도 해당 폴더 아래에 저장합니다.
 - 전처리 단계에서 별도의 모델을 필요로 하거나, 앙상블 목적으로 여러 모델을 사용하는 경우 역시 제시된 경로 아래에 저장합니다.
- 저장 방식: 각 프레임워크별 지정된 방식으로 저장합니다.
 - 명시되지 않은 프레임워크, 저장 방식은 본 대회에서는 지원되지 않습니다.
 - 지원 프레임워크: (PyTorch, TensorFlow(+Keras))

Weight 저장방식

- PyTorch / PyTorch 양자화:

- `model.state_dict()` 방식으로 저장하되 확장자를 `.pt`로 저장

예) `torch.save(model.state_dict(), 'model/weight.pt')`

- PyTorch 양자화 (위 방식이 적용되지 않는 경우):

- `torch.jit.script()` 방식으로 저장하되 확장자를 `.pt`로 저장

예) `m = torch.jit.script(model)`
`torch.jit.save(m, 'model/weight.pt')`

Weight 저장방식

- Tensorflow, tf.keras 및 Tensorflow 양자화 :
 - Model.save_weights() 방식으로 저장하되, 확장자를 `.h5`로 저장

예) `model.save_weights('model/weights.h5')`

- Tensorflow - TFLite양자화 (위 방식이 적용되지 않는 경우):
 - 확장자 `.tflite` 형식으로 저장

예)

```
converter = tf.lite.TFLiteConverter.from_keras_model(model)
converter.optimizations = [tf.lite.Optimize.DEFAULT]
tflite_model = converter.convert()
tflite_model_file = "model/weight.tflite"
tflite_model_file.write_bytes(tflite_model)
```

Weight 저장방식

- Scikit-Learn 등 :
 - Joblib으로 dump하되, 압축 없이 확장자를 `.pkl`로 저장

예) `joblib.dump(model, 'model/preprocessing.pkl', compress=0)`