분산처리 지원을 위한
모델 저장과 복원 방법 검토

2022. 04. 01.

배경 : 분산 처리 지원을 위한 최적화 확장

- 학습 루틴 확장
  - (From) 학습이 진행되는 교차로 이외는 고정 신호 이용 신호 제어

```
[%] python run.py --mode train --target-TLs “SA 101”  ….
```

  - (To) 일부는 학습된 모델 이용 추론, 일부는 고정 신호 이용 신호 제어

```
[%] python run.py –mode train  …..
    --target_TLs “ SA 101” –infer_TLs “SA 104, SA107” --infer_model_number 3 --infer_model_path /tmp/tso
```

  - 인자 의미
    - ✓ targetTLs : 학습을 진행하는 교차로 정보
    - ✓ infer_TLs : 학습된 모델의 추론을 통해 신호 제어를 하는 교차로 정보
    - ✓ infer_model_number : 추론에 이용될 학습된 모델의 번호
    - ✓ infer_model_path : 추론에 이용될 학습된 모델이 저장된 경로

```
*_SA101_trial_0.data-0000-of-0001
*_SA101_trial_0.index
*_SA101_trial_0.meta
```

```
*_SA104_trial_0.data-0000-of-0001
*_SA104_trial_0.index
*_SA104_trial_0.meta
```

…

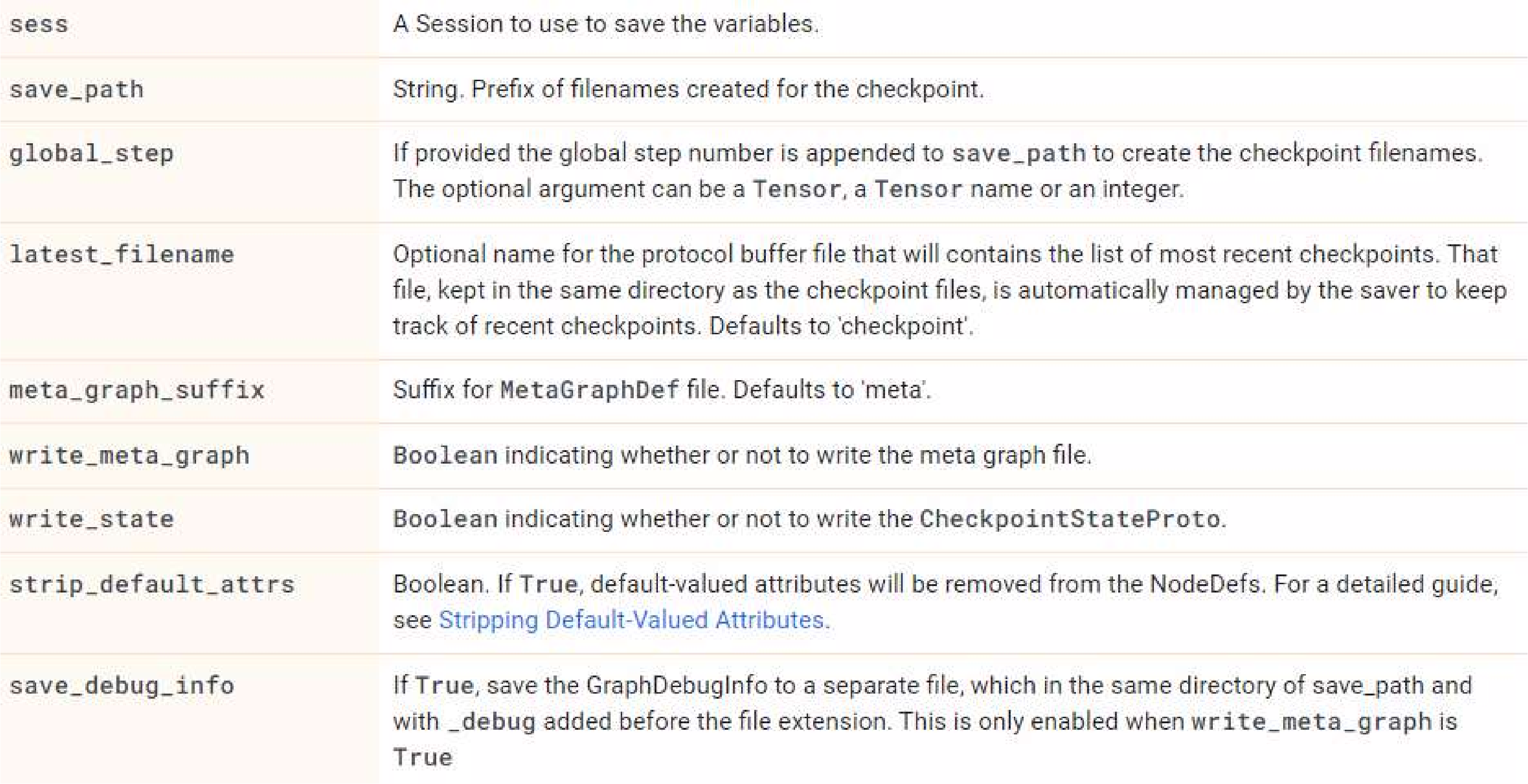
- 시험 루틴 확장
  - (From) 미리 내부에 정의된 경로의 모델 이용
  - (To) 인자로 전달하는 특정 경로의 모델 이용하도록 확장

```
[%] python run.py –mode test  …..
    --target_TLs “ SA 101, SA 104, SA 107” --infer_model_number 3 --infer_model_path /tmp/tso
```

    - ✓ SA 101_trial_0.*
  - 인자 의미
    - ✓ targetTLs : 시험을 진행하는 교차로 정보
    - ✓ infer_model_number : 추론에 이용될 학습된 모델의 번호 (분산 학습 횟수)…… 기존 model-num 이용해도 될까?
    - ✓ infer_model_path : 추론에 이용될 학습된 모델이 저장된 경로

TF 1.x

- Saver 객체 이용하여 Session 별로 저장 & 복원
  - tensorflow.compat.v1.train.Saver
  - Saver 는 Session과 함께 TF 2.x 에서 더 이상 지원하지 않음
  - save(sess, save_path, global_step=None, latest_filename=None, meta_graph_suffix='meta', write_meta_graph=True, write_state=True, strip_default_attrs=False, save_debug_info=False)

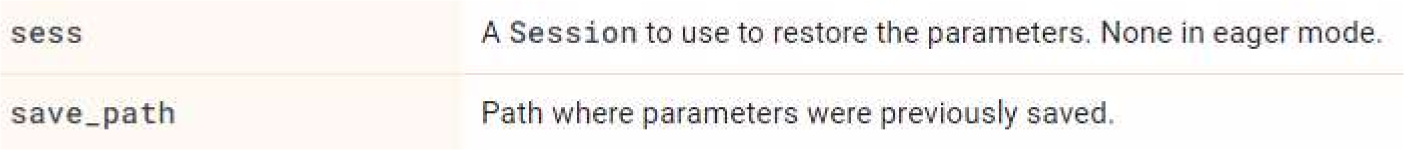


`sess`	A Session to use to save the variables.
`save_path`	String. Prefix of filenames created for the checkpoint.
`global_step`	If provided the global step number is appended to `save_path` to create the checkpoint filenames. The optional argument can be a `Tensor`, a `Tensor` name or an integer.
`latest_filename`	Optional name for the protocol buffer file that will contains the list of most recent checkpoints. That file, kept in the same directory as the checkpoint files, is automatically managed by the saver to keep track of recent checkpoints. Defaults to 'checkpoint'.
`meta_graph_suffix`	Suffix for `MetaGraphDef` file. Defaults to 'meta'.
`write_meta_graph`	`Boolean` indicating whether or not to write the meta graph file.
`write_state`	`Boolean` indicating whether or not to write the `CheckpointStateProto`.
`strip_default_attrs`	Boolean. If `True`, default-valued attributes will be removed from the NodeDefs. For a detailed guide, see Stripping Default-Valued Attributes.
`save_debug_info`	If `True`, save the GraphDebugInfo to a separate file, which in the same directory of save_path and with `_debug` added before the file extension. This is only enabled when `write_meta_graph` is `True`

Ref. https://www.tensorflow.org/api_docs/python/tf/compat/v1/train/Saver

TF 1.x

- Saver 객체 이용하여 Session 별로 저장 & 복원
  - tensorflow.compat.v1.train.Saver
  - Saver는 Session과 함께 TF 2.x 에서 더 이상 지원하지 않음
  - save(sess, save_path, global_step=None, latest_filename=None, meta_graph_suffix='meta', write_meta_graph=True, write_state=True, strip_default_attrs=False, save_debug_info=False)
  - restore(sess, save_path)

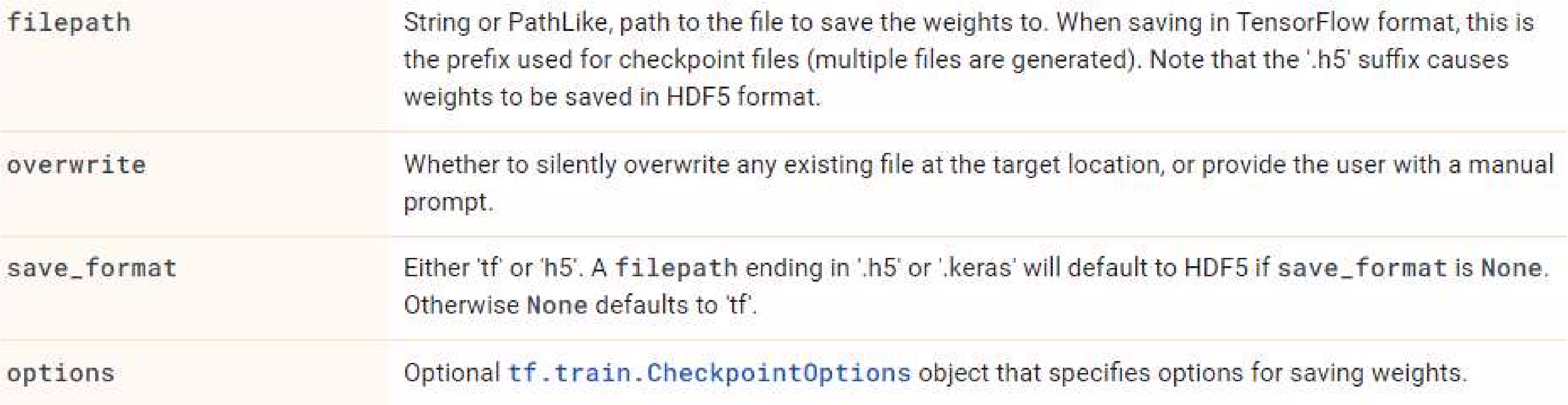


sess	A Session to use to restore the parameters. None in eager mode.
save_path	Path where parameters were previously saved.

Ref. https://www.tensorflow.org/api_docs/python/tf/compat/v1/train/Saver

TF 2.x

- Model 객체 이용하여 모델별 저장 & 복원
  - tensorflow.keras.Model : save_weights & load_weights
    - save_weights (filepath, overwrite=True, save_format=None, options=None)
      - ✓ Either saves in HDF5 or in TensorFlow format based on the save_format argument.
      - ✓ tensorflow.keras.Model

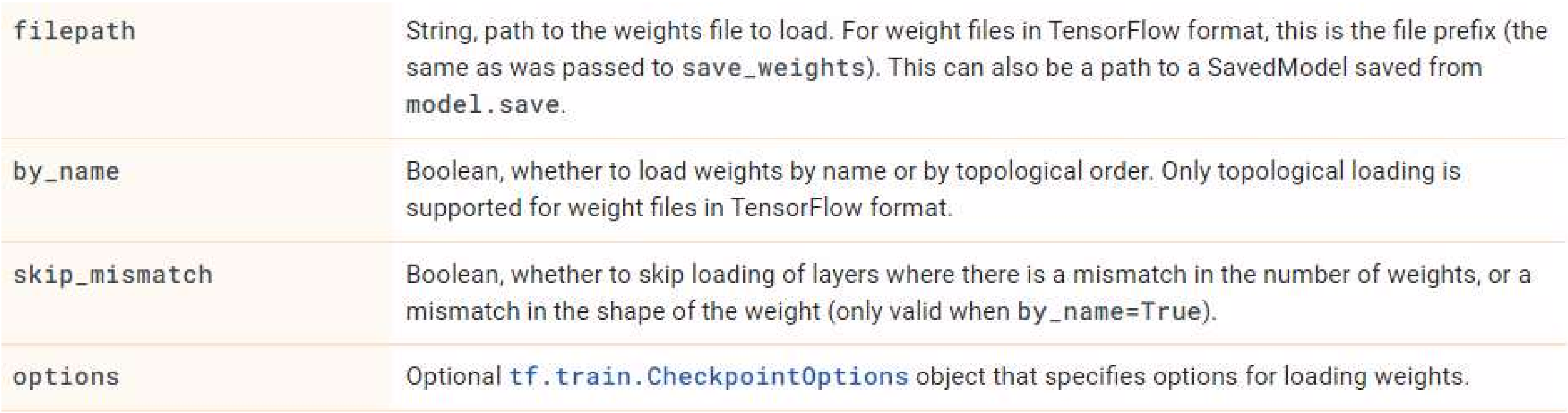


`filepath`	String or PathLike, path to the file to save the weights to. When saving in TensorFlow format, this is the prefix used for checkpoint files (multiple files are generated). Note that the '.h5' suffix causes weights to be saved in HDF5 format.
`overwrite`	Whether to silently overwrite any existing file at the target location, or provide the user with a manual prompt.
`save_format`	Either 'tf' or 'h5'. A `filepath` ending in '.h5' or '.keras' will default to HDF5 if `save_format` is `None`. Otherwise `None` defaults to 'tf'.
`options`	Optional `tf.train.CheckpointOptions` object that specifies options for saving weights.

Ref. https://www.tensorflow.org/api_docs/python/tf/keras/Model

TF 2.x

- Model 객체 이용하여 모델별 저장 & 복원
  - tensorflow.keras.Model : save_weights & load_weights
    - save_weights (filepath, overwrite=True, save_format=None, options=None)
    - load_weights(filepath, by_name=False, skip_mismatch=False, options=None)
      - ✓ Loads all layer weights, either from a TensorFlow or an HDF5 weight file
      - ✓ tensorflow.keras.Model

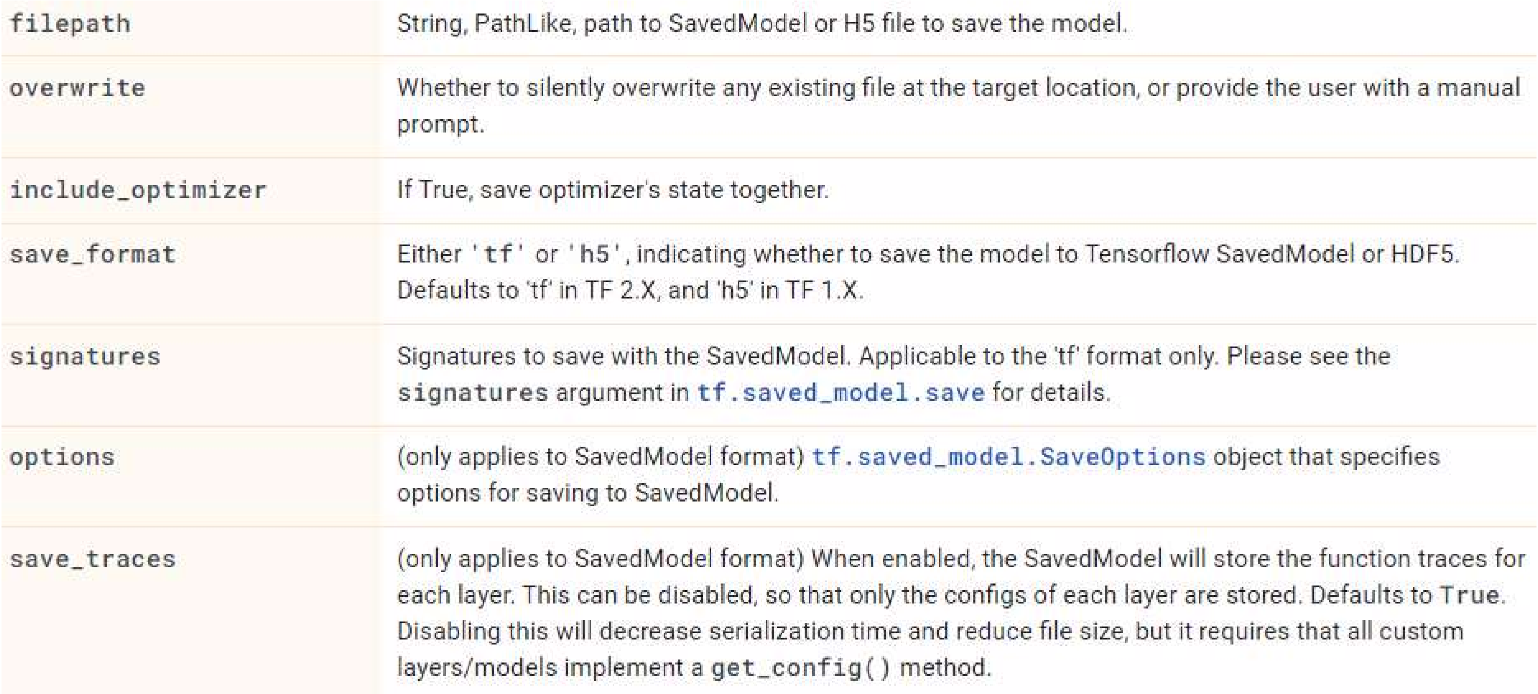


`filepath`	String, path to the weights file to load. For weight files in TensorFlow format, this is the file prefix (the same as was passed to `save_weights`). This can also be a path to a SavedModel saved from `model.save`.
`by_name`	Boolean, whether to load weights by name or by topological order. Only topological loading is supported for weight files in TensorFlow format.
`skip_mismatch`	Boolean, whether to skip loading of layers where there is a mismatch in the number of weights, or a mismatch in the shape of the weight (only valid when `by_name=True`).
`options`	Optional `tf.train.CheckpointOptions` object that specifies options for loading weights.

Ref. https://www.tensorflow.org/api_docs/python/tf/keras/Model

TF 2.x

- Model 객체 이용하여 모델별 저장 & 복원
  - tensorflow.keras.Model : save_weights & load_weights
  - save, save_model & load_model
    - save(filepath, overwrite=True, include_optimizer=True, save_format=None, signatures=None, options=None, save_traces=True)
      - ✓ Saves the model to Tensorflow SavedModel or a single HDF5 file
      - ✓ tensorflow.keras.Model

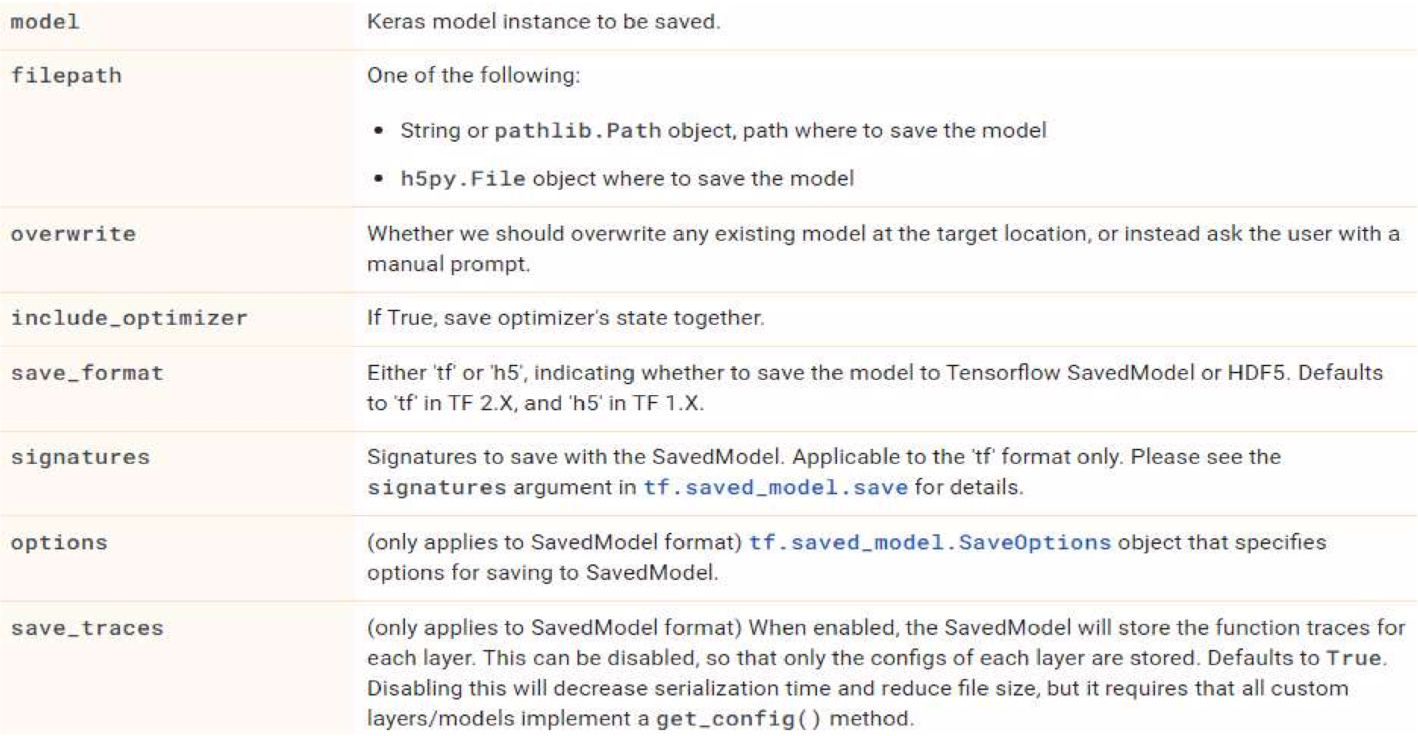


`filepath`	String, PathLike, path to SavedModel or H5 file to save the model.
`overwrite`	Whether to silently overwrite any existing file at the target location, or provide the user with a manual prompt.
`include_optimizer`	If True, save optimizer's state together.
`save_format`	Either `'tf'` or `'h5'`, indicating whether to save the model to Tensorflow SavedModel or HDF5. Defaults to 'tf' in TF 2.X, and 'h5' in TF 1.X.
`signatures`	Signatures to save with the SavedModel. Applicable to the 'tf' format only. Please see the `signatures` argument in `tf.saved_model.save` for details.
`options`	(only applies to SavedModel format) `tf.saved_model.SaveOptions` object that specifies options for saving to SavedModel.
`save_traces`	(only applies to SavedModel format) When enabled, the SavedModel will store the function traces for each layer. This can be disabled, so that only the configs of each layer are stored. Defaults to `True`. Disabling this will decrease serialization time and reduce file size, but it requires that all custom layers/models implement a `get_config()` method.

Ref. https://www.tensorflow.org/api_docs/python/tf/keras/Model

TF 2.x

- Model 객체 이용하여 모델별 저장 & 복원
  - tensorflow.keras.Model : save_weights & load_weights
  - save, save_model & load_model
    - save(filepath, overwrite=True, include_optimizer=True, save_format=None, signatures=None, options=None, save_traces=True)
    - save_model(model, filepath, overwrite=True, include_optimizer=True, save_format=None, signatures=None, options=None, save_traces=True)
      - ✓ Saves a model as a TensorFlow SavedModel or HDF5 file.
      - ✓ tensorflow.keras.models

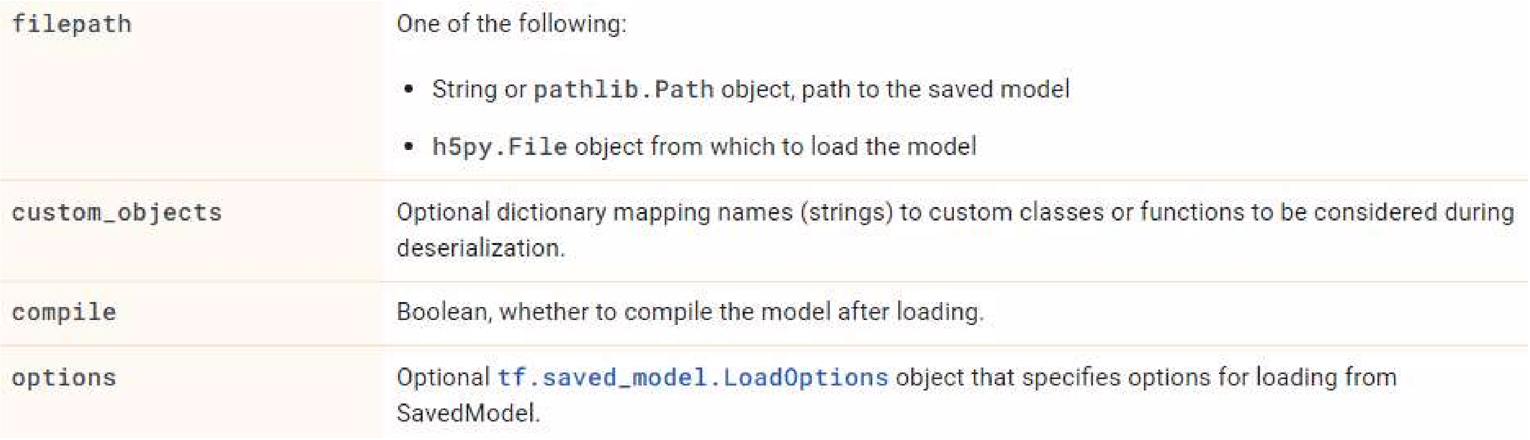


`model`	Keras model instance to be saved.
`filepath`	One of the following: • String or `pathlib.Path` object, path where to save the model • `h5py.File` object where to save the model
`overwrite`	Whether we should overwrite any existing model at the target location, or instead ask the user with a manual prompt.
`include_optimizer`	If True, save optimizer's state together.
`save_format`	Either 'tf' or 'h5', indicating whether to save the model to Tensorflow SavedModel or HDF5. Defaults to 'tf' in TF 2.X, and 'h5' in TF 1.X.
`signatures`	Signatures to save with the SavedModel. Applicable to the 'tf' format only. Please see the `signatures` argument in `tf.saved_model.save` for details.
`options`	(only applies to SavedModel format) `tf.saved_model.SaveOptions` object that specifies options for saving to SavedModel.
`save_traces`	(only applies to SavedModel format) When enabled, the SavedModel will store the function traces for each layer. This can be disabled, so that only the configs of each layer are stored. Defaults to `True`. Disabling this will decrease serialization time and reduce file size, but it requires that all custom layers/models implement a `get_config()` method.

Ref. https://www.tensorflow.org/api_docs/python/tf/keras/models/save_model

TF 2.x

- Model 객체 이용하여 모델별 저장 & 복원
  - tensorflow.keras.Model : save_weights & load_weights
  - save, save_model & load_model
    - save(filepath, overwrite=True, include_optimizer=True, save_format=None, signatures=None, options=None, save_traces=True)
    - save_model(model, filepath, overwrite=True, include_optimizer=True, save_format=None, signatures=None, options=None, save_traces=True)
    - load_model(filepath, custom_objects=None, compile=True, options=None)
      - ✓ Loads a model saved via model.save()
      - ✓ tensorflow.keras.models



`filepath`	One of the following: • String or `pathlib.Path` object, path to the saved model • `h5py.File` object from which to load the model
`custom_objects`	Optional dictionary mapping names (strings) to custom classes or functions to be considered during deserialization.
`compile`	Boolean, whether to compile the model after loading.
`options`	Optional `tf.saved_model.LoadOptions` object that specifies options for loading from SavedModel.

Ref. https://www.tensorflow.org/api_docs/python/tf/keras/models/load_model

저장 & 복원 방법을 비교해보니…

- 분산 처리를 위한 요구사항
  - 학습시 학습 대상이 아닌 교차로군에 대해 다른 곳에서 학습된 모델을 이용한 추론 혹은 고정 신호로 제어
  - 시험시 여러 곳에서 분산 학습한 모델들을 로딩하여 추론을 통해 신호 제어

- **Session 별로 모델을 다루는 TF 1.x 에서는 어렵다**..(많이 고쳐야… 복잡…)
  - 하나의 세션에 속하는 여러 모델을 한꺼번에 저장 & 복원
    - *.data, *.index, *.model 의 3개 파일 활용
  - 여러 곳에서 학습한 것을 한 곳에서 로딩하는 방법이 없어 보임
  - **여러 세션을 이용 → 이론 상 가능해 보이나 Sample이 없다**
    - 각 저장된 모델 별로 세션 생성 : sess , infer_sess[]

- **모델별로 저장 & 복원이 가능한 TF 2.x 형태로 개발해야 함**
  - PPO TF 2.x 버전 필요
  - 통계 정보 관련 코드 수정 필요
  - env.step(), run_sappo() 추론 관련 코드 추가 필요

- TF 2.x로 가면서 사라진 것들
  - Session, Saver, FileWriter
  - placeholder

Ref. https://intrepidgeeks.com/tutorial/tensorflow-loads-multiple-models-at-the-same-time
https://stackoverflow.com/questions/41607144/loading-two-models-from-saver-in-the-same-tensorflow-session