

モデルの予測と評価だけしたい人向けの説明。

zip ファイルに保存したモデルのフォルダが圧縮されている。まず zip を展開してフォルダを用意する。次にモデルを読み込む。load_model() を使う。展開された my_model ディレクトリのパスを与える。

```
from tensorflow import keras

model = keras.models.load_model("my_model")
```

テスト用のデータセットを読み込む。dataset_processor.py を import し、DatasetProcessor クラスをインスタンス化し、process_dataset() を実行する。

以下引数の説明。

- dirpath
XML ファイルのあるディレクトリのパスを与える。必須。
- save_path
作成した numpy 配列をファイルに保存するときの保存先ディレクトリのパスを与える。任意。デフォルトでは dirpath を指定したことになる。
- max_file
処理するファイルの数を指定する。任意。デフォルトでは 100。

```
import dataset\_processor as dp

pr = dp.DatasetProcessor(
    dirpath=' /content/drive/MyDrive/実験 2/prj-14/mj_files/2012' ,
    save_path=' /content/drive/MyDrive/実験 2/prj-14/mj_files' ,
    max_file=200
)

pr.process_dataset()
```

実行が正常に終わったことの確認。データセットのアクセス方法は、入力データは「インスタンス変数.data」、ラベルは「インスタンス変数.label」。以下の例ではデータ数 191548 となっていることが分かる。

```
print(pr.data.shape)
print(pr.label.shape)
```

```
(191548, 3209)
(191548,)
```

Keras の `evaluate()` を実行するとモデルの精度評価が得られる。accuracy, F1, PR-AUC, Precision, Recall が得られる。特に重要なのは Precision と Recall。

```
acc, f1, auc, pre, rec = model.evaluate(pr.data, pr.label)
```

Keras の `predict()` を実行すると、データに対する予測値が得られる。

```
predictions = model.predict(pr.data)
```

学習時の検証データでは、Precision は 0.5 程度、Recall は 0.35 程度になった。データが 1 に分類された元でそれが正解である確率は 0.5 で、データのラベルが 1 である時にそれが 1 に分類される確率は 0.35 といった状態。