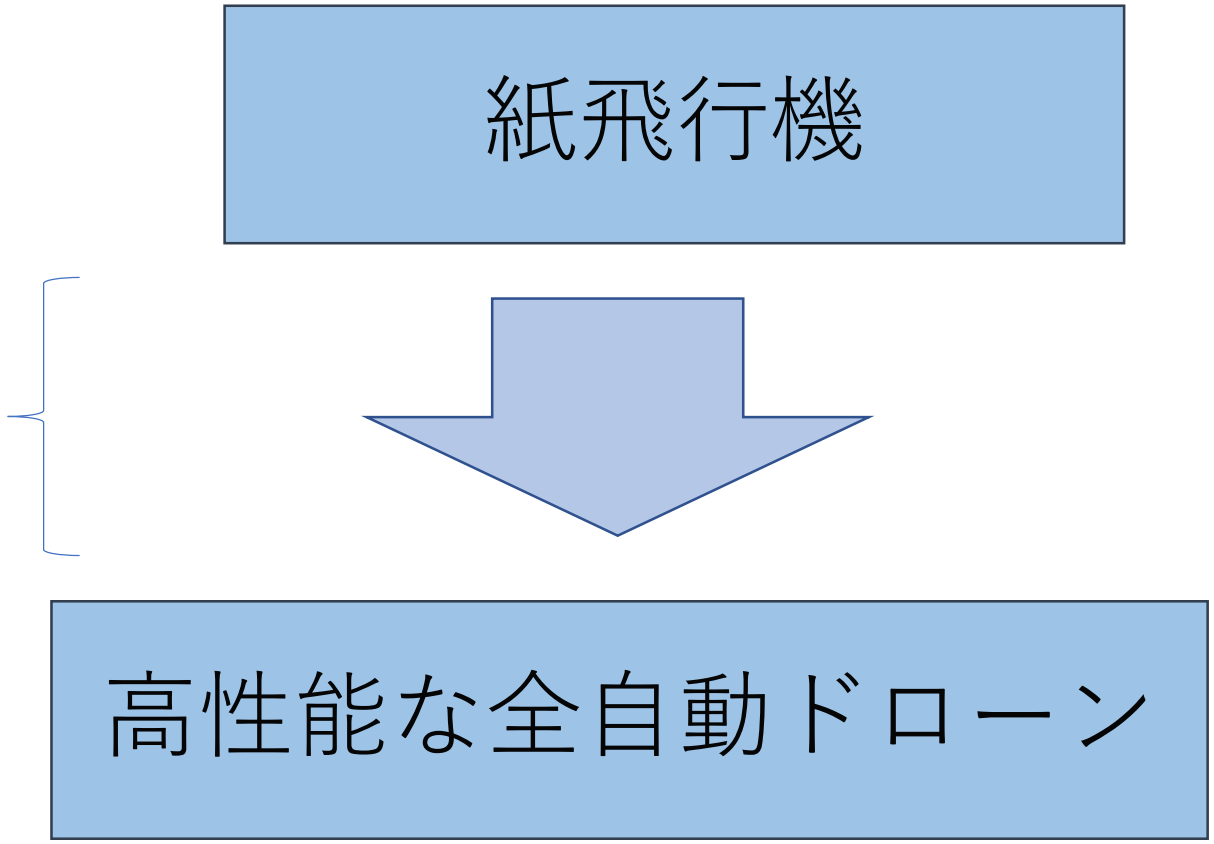
PythonとKerasによるディープラーニング

19nd302h マブン

7.2 Kerasのコールバックと TensorBoardを使ったディープラーニングモデルの調査と監視

訓練中にモデルの内部で起きていることを詳しく確認し、制御するための方法を紹介します

model.fit()や
model.fit_generator
を使って大規模な
データセットの訓練
を数十エポックにわ
たって実行すること



7.2.1訓練中にコールバックを使ってモデルを制御する

コールバックとは

fit呼び出しを通じてモデルに渡され、訓練中に様々なタイミングでモデルを呼び出されるオブジェクトのことです。

このオブジェクトは、モデルの状態やその性能など利用可能なデータのすべてにアクセスし、訓練の中止、モデルの保存、異なる重みの読み込み、モデルの状態を簡単に変更できます。

コールバックの使用例

- **モデルのチェックポイント化**

訓練中の様々な時点でモデルの現在の重みを保存します。

- **訓練の中止**

検証データでの損失値がそれ以上改善しなくなったところで、訓練を中止します。（訓練全体で改善のモデルを保存します。）

- **特定のパラメータの動的な調整**

訓練中にオプティマイザの学習率といってパラメータの値を動的調整します。

- **訓練と検証の指標を記録**

訓練と検証の指標をログに記録するか、モデルによって学習された表現をそれらの更新に応じて可視化します。おなじみのKerasのプログレスバーはコールバックです。

ModelCheckpointコールバック

訓練中にモデルを繰り返し保存できます。（必要であれば、
その時点で最善のモデル、つまり、1つのエポックを終了した
時点で最も良い性能を達成したモデルだけを保存します。）

keras.callbacks.ModelCheckpoint

(filepath, monitor='val_loss', verbose=0,save_best_only=**False**, save_weights_only=**False**, mode='auto', period=1)

各エポック終了後にモデルを保存します.

filepathは，（on_epoch_endで渡された）epochの値とlogsのキーで埋められた書式設定オプションを含むことができます.

例えば，filepathがweights.{epoch:02d}-{val_loss:.2f}.hdf5の場合，複数のファイルがエポック数とバリデーションロスの値を付与して保存されます.

EarlyStoppingコールバック

監視している成果指標が一定のエポック数にわたって改善されなかった場合は、EarlyStoppingコールバックを使って、訓練を中止できます。

keras.callbacks.EarlyStopping
(monitor='val_loss', min_delta=0, patience=0, verbose=0, mode='auto’)

監視する値の変化が停止した時に訓練を終了します

通常、ModelCheckpointコールバックとEarlyStoppingコールバックは組み合わせで使用されます。

ReduceLRnPlateauコールバック

- 評価値の改善が止まった時に学習率を減らします。

```
callbacks_list = [
    keras.callbacks.ReduceLROnPlateau(
        monitor='val_loss'  # モデルの検証データセットでの損失値を監視
        factor=0.1,         # コールバックが起動したら学習率を10で割る
        patience=10,        # 検証データでの損失値が10エポックにわたって
                            # 改善しなかった場合はコールバックを起動
    )
]

# このコールバックはval_lossを監視するため、
# fit呼び出しにvalidation_dataを指定する必要がある
model.fit(x, y,
          epochs=10,
          batch_size=32,
          callbacks=callbacks_list,
          validation_data=(x_val, y_val))
```

コールバックの作成

基底クラスのkeras.callbacks.Callbackを拡張することで，カスタムコールバックを作成できます。 コールバックは，self.modelプロパティによって，関連したモデルにアクセスできます.

訓練中の各バッチの損失のリストを保存する簡単な例は，以下のようになります.

```
class LossHistory(keras.callbacks.Callback):
    def on_train_begin(self, logs={}):
        self.losses = []

    def on_batch_end(self, batch, logs={}):
        self.losses.append(logs.get('loss'))
```

7.2.2 TensorBoard : TensorFlowの可視化フレームワーク

TensorBoardとは

TensorFlowに含まれているブラウザベースの可視化ツールです。

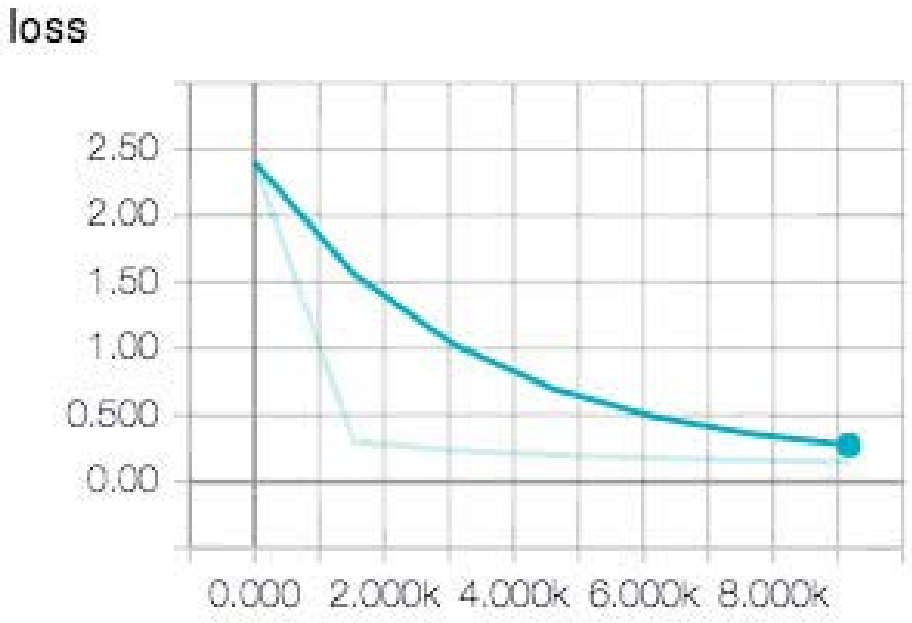
TensorBoardは訓練中にモデルの内部で起きていることをすべて視覚的に監視できるようにすることです。これにより、モデルの最適化やパラメーターチューニングに対する示される。

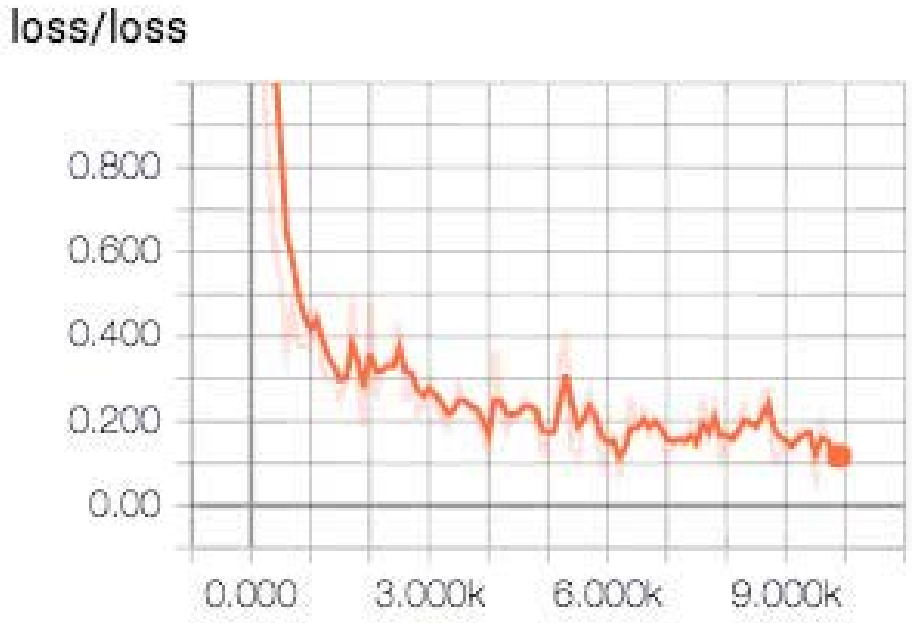
TensorBoardの機能

- 訓練中に指標を視覚的に監視
- モデルのアーキテクチャの可視化
- 活性化と勾配のヒストグラムの可視化
- 埋め込みを三次元で調査
- 画像の可視化
- 音声の再生

折れ線グラフ

折れ線グラフは、モデルのaccuracyやloss、学習速度などのスカラー値を可視化するためによく使われます。以下の画像は、MNIST文字認識のモデルの学習時にlossをログに残した例です。

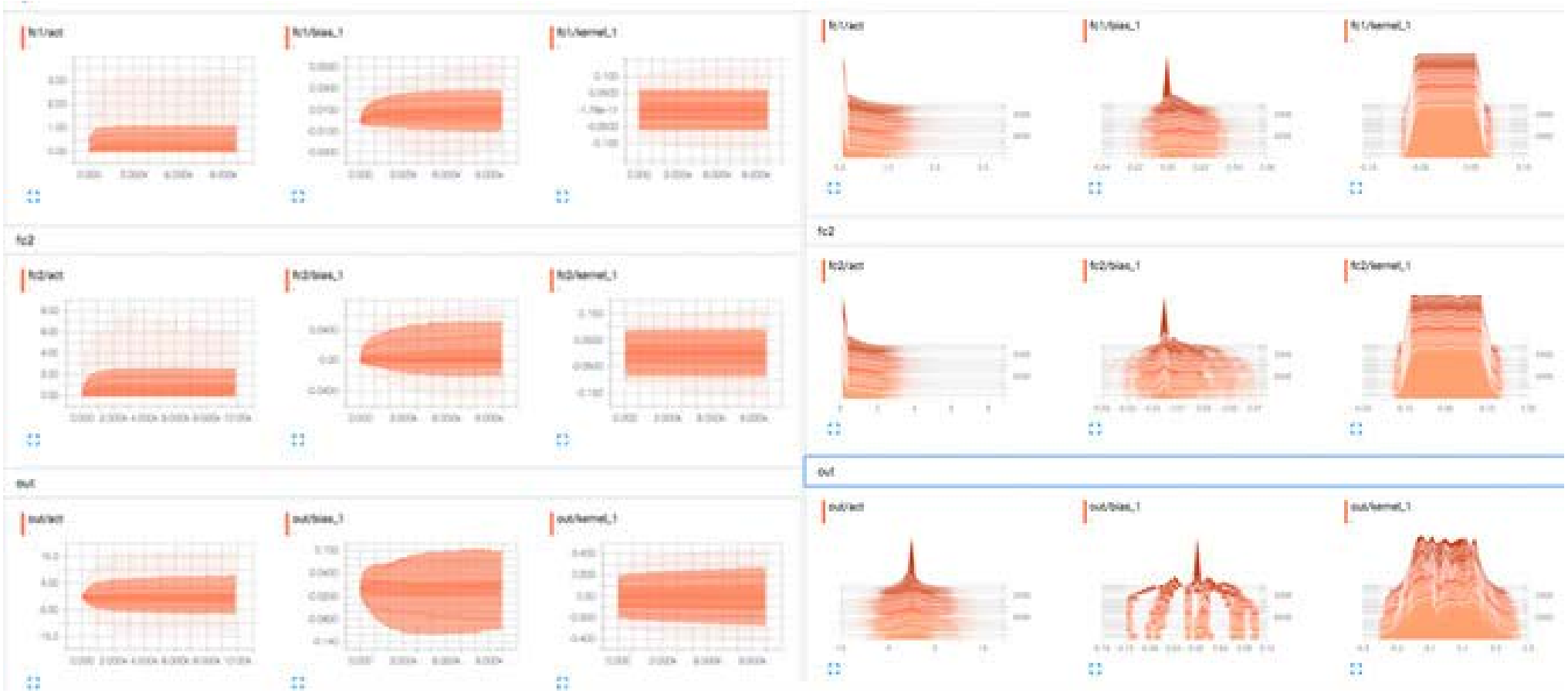




左側がテストデータセットのlossで、右側が訓練データセットのlossです。最初大きかったlossが徐々に小さくなっているので、正しく学習されていることが分かります。

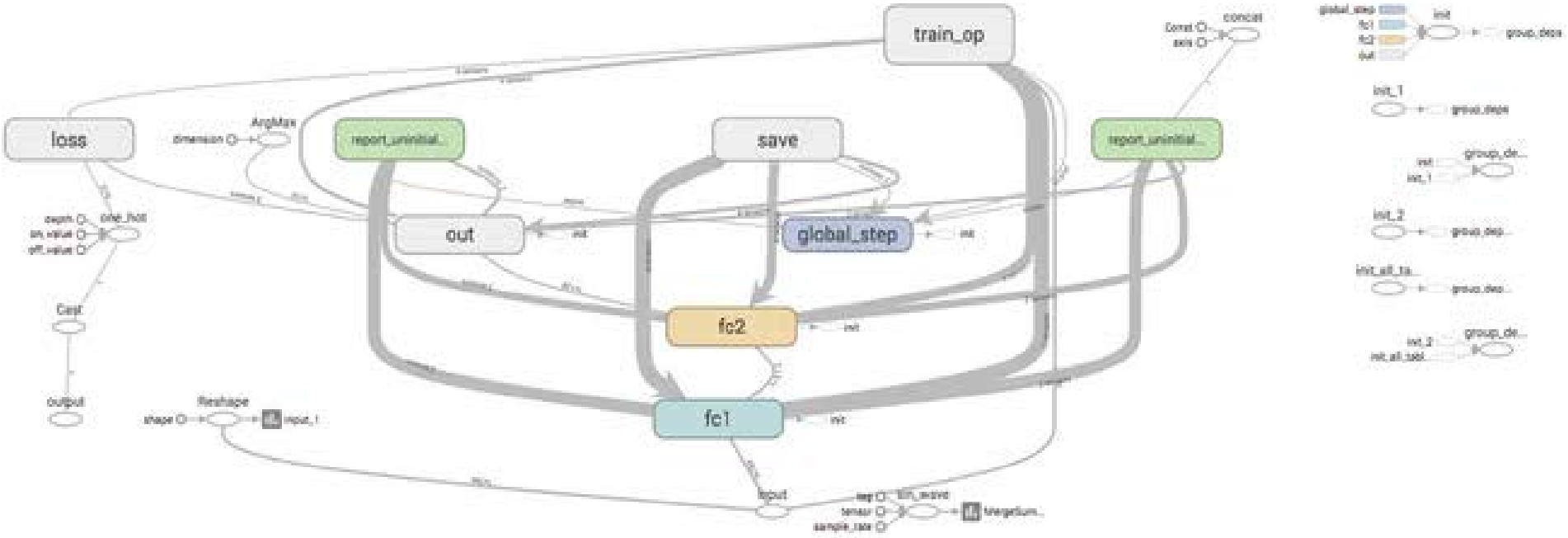
ヒストグラム

ヒストグラムの可視化にも対応しています。ニューラルネットワークの各層での重みやバイアスが初期値からどのように変化したのかを可視化すると便利です。可視化することで、各層で勾配消失していないかどうかを確認することができます。



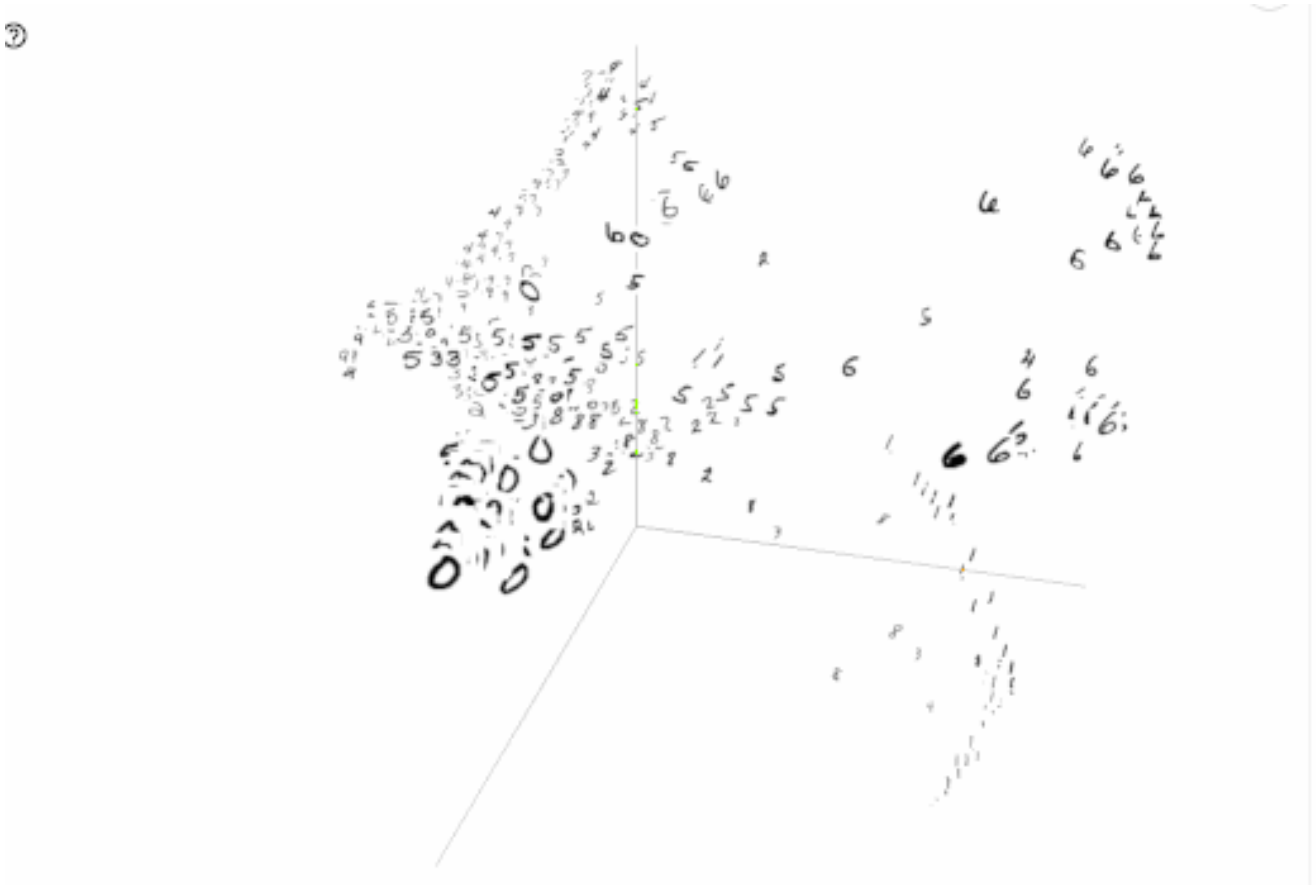
計算グラフ

計算グラフの可視化もできます。モデルが設計通り正しく実装されているか、ひと目で確認することができます。



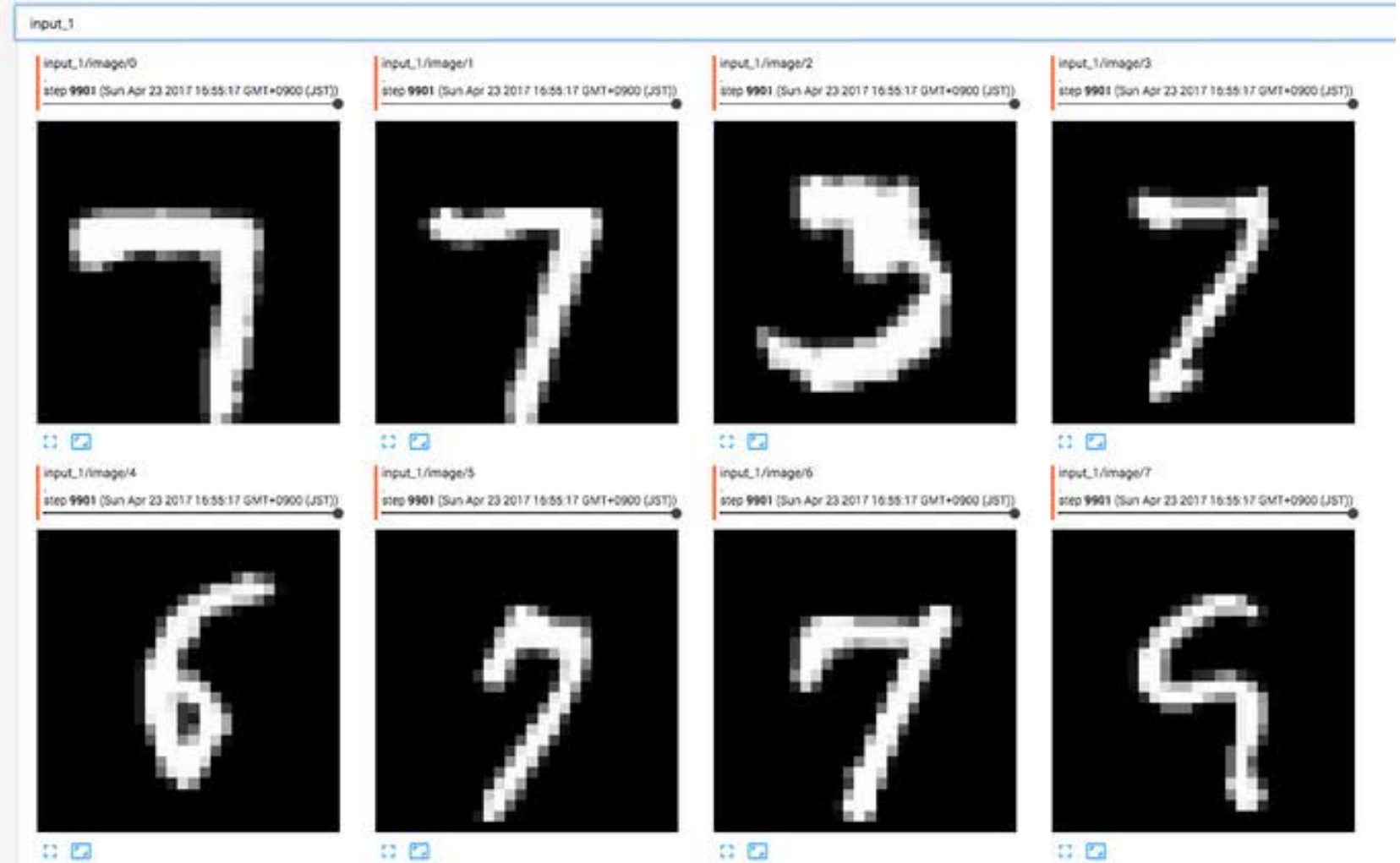
次元削減のプロット

次元削減のプロットは、TensorFlowの0.12から導入された機能です。単語の分散表現などを可視化してみたいときに便利です。
以下の図は、TensorBoardでMNISTの画像をt-SNEで可視化した例です。



画像

スカラー値と同様に、画像も可視化することができます。以下の画像は、MNISTの訓練データを可視化した例です。



音声

TensorBoardは、音声の再生もサポートしています。画像の時と同様に、前処理で音声を処理した後に、正しく処理されたかどうかを確かめることができます。

音量の調節やダウンロードもサポートされています。

7.2.3まとめ

- Kerasのコールバックは、訓練中のモデルを監視し、モデルの状態に基づいて自動的にアクションを実行するための単純な手段となる。
- Tensorflowを使用している場合、Tensorboardはモデルの状態をブラウザで可視化するための優れた手段となる。KerasのモデルでTensorboardを使用するには、Tensorboardコールバックを使用する