

【要点まとめ】

音声データも画像と同じように、畳み込みニューラルネットワークを適用することができる。

※1層の畳み込み層とMaxPooling

kerasライブラリで双方向RNNを実装するには、Bidirectionalレイヤーを準備する。

また、勾配のクリッピングはコンパイル時のOptimizerの引数一つで可能（便利）。

※実装例は実装演習に記述する

【実装演習】

双方向RNN

```
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras import layers

tf.keras.backend.clear_session()

model = tf.keras.models.Sequential()

model.add(layers.Input((NUM_DATA_POINTS, 1)))
model.add(layers.Bidirectional(layers.LSTM(64))) # ここで双方向の準備をしている
model.add(layers.Dense(10, activation='softmax'))

model.summary()

model.predict(sample[0]).shape
```

勾配クリッピング

```
model.compile(
    loss=tf.keras.losses.SparseCategoricalCrossentropy(),
    optimizer=tf.keras.optimizers.Adam(clipvalue=0.5), # この引数で指定
    metrics=['accuracy']
)
```

【実装演習考察】

kerasなどのライブラリではいとも簡単に双方向RNNや勾配のクリッピングを実装できてしまいました。

ですが他人に説明できるように、動画やテキストで教えていただいた理論をしっかりとれるようにしていきたいと思います。

【自己学習】

※参考資料：ゼロから作るDeepLearning②

```
rate = threshold / norm
if rate < 1:
```

```
    return grad * rate
return grad
```

勾配のクリッピングは上記のような実装となると学んだが、一旦rate変数に格納する必要性について参考資料に記載があった。

「複数の勾配を更新するときに、rate変数に格納しておくを使い回せて便利」
こういった観点で実装を進められるように精進すべきだと感じた。