

【要点まとめ】

従来のLSTMではパラメータが多かったため、計算量もそれに伴って多かった。
GRUはパラメータを大幅に削減したが、LSTMと同等以上の精度が望めるようになった。

- CECは廃止
- リセットゲート：過去の情報をどの程度消去するかを演算
- 更新ゲート：過去の情報をどの程度取り込むかを演算

【実装演習】

順伝播の実装

```
def forward(self, x, h_prev):
    Wx, Wh, b = self.params
    H = Wh.shape[0]
    Wxz, Wxr, Wxh = Wx[:, :H], Wx[:, H:2*H], Wx[:, 2*H:]
    Whz, Whr, Whh = Wh[:, :H], Wh[:, H:2*H], Wh[:, 2*H:]
    bz, br, bh = b[:H], b[H:2*H], b[2*H:]

    z = sigmoid(np.dot(x, Wxz) + np.dot(h_prev, Whz) + bz)
    r = sigmoid(np.dot(x, Wxr) + np.dot(h_prev, Whr) + br)
    h_hat = np.tanh(np.dot(x, Wxh) + np.dot(r*h_prev, Whh) + bh)
    h_next = (1-z) * h_prev + z * h_hat

    self.cache = (x, h_prev, z, r, h_hat)

    return h_next
```

【実装演習考察】

LSTMと比較してゲートが少なくなったため、変数の数とコード量、関数の返却値が減ったことを理解できました。

【自己学習】

※参考資料：ゼロから作るDeepLearning②

LSTMとGRUのどちらを使用すべきか...最近はLSTMが多く使われているようだが

GRUは計算量が小さいため、データセットのサイズが小さい場合や繰り返しの施行が必要な場合に軍配があがる。