④双方向RNN

=====

目次:

1.双方向LSTM(TimeBiLSTM)の実装

【要約】

・双方向RNN　→過去の情報だけでなく、未来の情報を加味することで、精度を向上させるためのモデル。(実用例:文章の推敲、機械翻訳など)

・文章の場合は、過去～未来の情報すべてを含んでいるので、未来の情報を学習で活用することができる。

・ネットワーク構造　→未来の情報が遡って伝わってくるようになっている。

=====

1.双方向LSTM(TimeBiLSTM)の実装

→コードの確認

In [1]:

```
class TimeBiLSTM:
    def __init__(self, Wx1, Wh1, b1,
                 Wx2, Wh2, b2, stateful=False):
        self.forward_lstm = TimeLSTM(Wx1, Wh1, b1, stateful)
        self.backward_lstm = TimeLSTM(Wx2, Wh2, b2, stateful)
        self.params = self.forward_lstm.params + self.backward_lstm.params
        self.grads = self.forward_lstm.grads + self.backward_lstm.grads

    def forward(self, xs):
        o1 = self.forward_lstm.forward(xs)
        o2 = self.backward_lstm.forward(xs[:, ::-1])
        o2 = o2[:, ::-1]

        out = np.concatenate((o1, o2), axis=2)
        return out

    def backward(self, dhs):
        H = dhs.shape[2] // 2
        do1 = dhs[:, :, :H]
        do2 = dhs[:, :, H:]

        dxs1 = self.forward_lstm.backward(do1)
        do2 = do2[:, ::-1]
        dxs2 = self.backward_lstm.backward(do2)
        dxs2 = dxs2[:, ::-1]
        dxs = dxs1 + dxs2
        return dxs
```

(実装例):双方向LSTM

In [ ]:

```
# Bidirectional LSTM
model = Sequential()
model.add(Embedding(s_vocabsize, EMBED_SIZE, input_length=MAX_SEQLEN))
model.add(Dropout(0.2))
model.add(Bidirectional(LSTM(HIDDEN_SIZE, dropout=0.2, recurrent_dropout=0.2)))
model.add(RepeatVector(MAX_SEQLEN))
model.add(Bidirectional(LSTM(HIDDEN_SIZE, return_sequences=True)))
model.add(TimeDistributed(Dense(t_vocabsize)))
model.add(Activation("softmax"))
model.compile(loss="categorical_crossentropy",
              optimizer="adam", metrics=["accuracy"])
model.summary()
model.fit(Xtrain, Ytrain, batch_size=BATCH_SIZE,
          epochs=NUM_EPOCHS, validation_data=[Xtest, Ytest])
score, acc = model.evaluate(Xtest, Ytest, batch_size=BATCH_SIZE)
print("Bidirectional LSTM : Test score: {:.3f}, accuracy: {:.3f}¥n¥n".format(score, acc))

backend.clear_session()
```

↓比較:普通のLSTM　コードとしては、あまり変わらない→Bidirectionalの有無くらい

In [ ]:

```
# LSTM
model = Sequential()
model.add(Embedding(s_vocabsize, EMBED_SIZE, input_length=MAX_SEQLEN))
model.add(Dropout(0.2))
model.add(LSTM(HIDDEN_SIZE, dropout=0.2, recurrent_dropout=0.2))
model.add(RepeatVector(MAX_SEQLEN))
model.add(LSTM(HIDDEN_SIZE, return_sequences=True))
model.add(TimeDistributed(Dense(t_vocabsize)))
model.add(Activation("softmax"))
model.compile(loss="categorical_crossentropy",
              optimizer="adam", metrics=["accuracy"])
model.summary()
model.fit(Xtrain, Ytrain, batch_size=BATCH_SIZE,
          epochs=NUM_EPOCHS, validation_data=[Xtest, Ytest])
score, acc = model.evaluate(Xtest, Ytest, batch_size=BATCH_SIZE)
print("LSTM : Test score: {:.3f}, accuracy: {:.3f}¥n¥n".format(score, acc))
```

In [ ]:

```
```