

深層学習レポート day3 (11 点以上で合格)

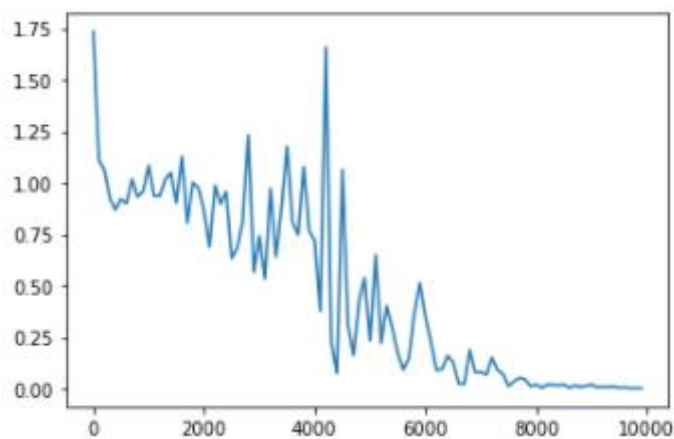
➤ 再帰型ニューラルネットワークの概念

✓ 要点のまとめ

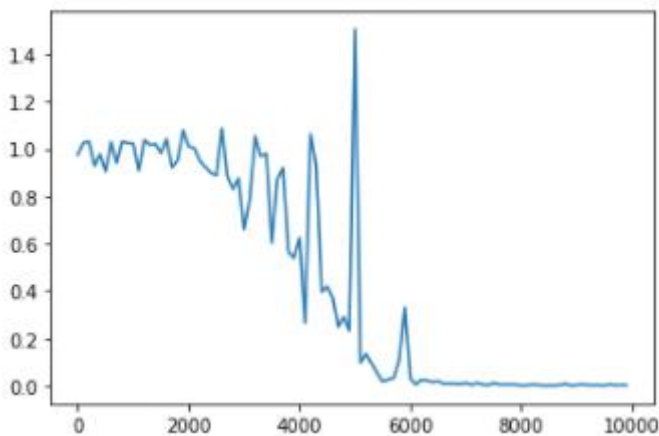
再帰型ニューラルネットワークとは **RNN** と呼ばれ、時系列データに対応ができるニューラルネットワークのことである。時系列データとは、データの中に時間的な関係のあるデータである。例えば、音声データや株価のデータ、動画のデータなどがある。RNN では、重みのパラメータ更新に過去の重みの情報を活用している。

✓ 実装演習サマリーと考察

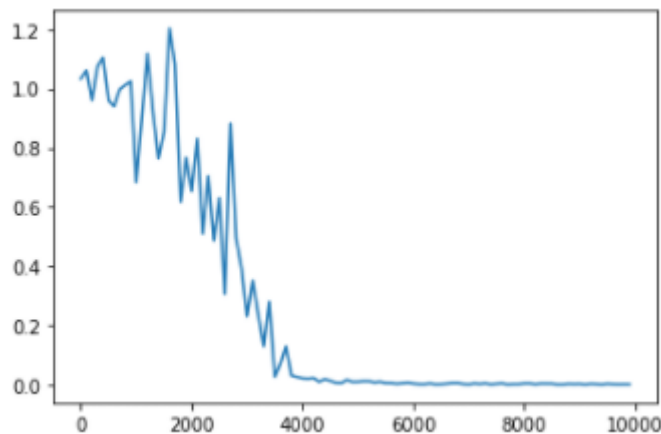
3_1_simple_RNN_after の実行結果。



重みの初期値を Xavier に変更。



重みの初期値を He に変更。



初期値 He において、最も低い Loss を記録した。

➤ LSTM

✓ 要点のまとめ

RNN の課題として、長い時系列の場合勾配が消失するため、学習が困難という点があった。RNN における勾配消失問題を解決したのが LSTM。LSTM では、忘却ゲートという考え方を導入し、過去のデータすべてを保持するのではなく、必要な過去データのみを保持しつつ学習を進めることができる。

✓ 確認テストの考察

確認テスト

以下の文章をLSTMに入力し空欄に当てはまる単語を予測したいとする。
文中の「とても」という言葉は空欄の予測において
なくなっても影響を及ぼさないと考えられる。
このような場合、どのゲートが作用すると考えられるか。

「映画おもしろかったね。ところで、とてもお腹が空いたから何か ____。」
(3分)

回答は「忘却ゲート」忘却ゲートは必要なくなった過去のデータを破棄する役割を果たす。計算量の減少と、勾配消失や勾配爆発が起きにくくなる効果があ

る。LSTM にはほかに入力ゲートと出力ゲートという 2 つのゲートがある。

➤ GRU

✓ 要点のまとめ

LSTM には、パラメータ数が多く、計算負荷が高いという課題があった。それを解決しようとしたのが GRU。GRU では LSTM に比べ、パラメータを大きく削減している。精度は LSTM と同等あるいはそれ以上。GRU ではリセットゲートと更新ゲートという二つのゲートを使用している。

✓ 実装演習の結果と考察

GoogleColab ではデフォルトの tensorflow バージョンが 2.0 以降であり、演習用に配布されていたサンプルコードがそのままでは動かなかった。

よって、以下のコードでバージョンの調整を行った。

```
!pip3 uninstall tensorflow
!pip3 install tensorflow==1.14.0
```

```
Uninstalling tensorflow-2.5.0:
  Would remove:
    /usr/local/bin/estimator_ckpt_converter
    /usr/local/bin/import_pb_to_tensorboard
    /usr/local/bin/saved_model_cli
    /usr/local/bin/tensorboard
    /usr/local/bin/tf_upgrade_v2
    /usr/local/bin/tflite_convert
    /usr/local/bin/toco
    /usr/local/bin/toco_from_protos
    /usr/local/lib/python3.7/dist-packages/tensorflow-2.5.0.dist-info/*
    /usr/local/lib/python3.7/dist-packages/tensorflow/*
```

「some of them looks like」をインプットにした場合は<??>が予測されてしまった。

```
<??> : 0.47543268
Prediction: some of them looks like <??>
9.139572e-07
```

ほかの例では文脈によるがありえる文章が作成された。

```
Prediction: My favorite food in Japan is that
array([3.42511271e-06, 8.32151130e-08, 3.79441154e-08, ...,
       1.13813674e-10, 1.25209371e-10, 8.14242640e-07], dtype=float32)
```

```
Prediction: Father's father is you
array([3.5700094e-08, 1.5500591e-06, 4.1283865e-09, ..., 4.0393132e-12,
       4.4272728e-12, 1.4025529e-03], dtype=float32)
```

✓ 確認テスト

LSTMとCECが抱える課題について、それぞれ簡潔に述べよ。
(3分)

LSTM：パラメータ数が多く、計算の負荷が高い。

GRU：LSTM に比べると表現力は劣る。LSTM ほどの長期記憶は保持できない。

➤ 双方向 RNN

✓ 要点のまとめ

過去の情報だけでなく、未来の情報も用いて精度を向上させるためのモデル。主な用途には機械翻訳や文章の推敲などがある。学習に未来のデータを必要とするため、過去のデータをもとに未来を予測するようなタスク(株価の予測など)には用いることができない。

➤ Seq2Seq

✓ 要点のまとめ

Seq2Seq とは、インプットをベクトルに変換するエンコーダと、ベクトルをアウトプットに変換するデコーダを備えた **Encoder-Decoder** モデルを使って、系列データを別の系列データに変換するモデルのこと。インプットを英語の文章、アウトプットを日本語の文章とするなどして、機械翻訳などに活用されている。

確認テスト

下記の選択肢から、seq2seqについて説明しているものを選び。

- (1) 時刻に関して順方向と逆方向のRNNを構成し、それら2つの中間層表現を特徴量として利用するものである。
- (2) RNNを用いたEncoder-Decoderモデルの一種であり、機械翻訳などのモデルに使われる。
- (3) 構文木などの木構造に対して、隣接単語から表現ベクトル（フレーズ）を作るという演算を再帰的に行い（重みは共通）、文全体の表現ベクトルを得るニューラルネットワークである。
- (4) RNNの一種であり、単純なRNNにおいて問題となる勾配消失問題をCECとゲートの概念を導入することで解決したものである。

(3分)

回答は「2」。(1)は双方向 RNN の説明になっている。(3)は Word2Vec の説明になっている。(4)は LSTM あるいは GRU の説明になっている。

➤ Word2vec

✓ 要点のまとめ

RNN では、固定長のデータしかインプットに指定できないという制約がある。そのため、文章に含まれる単語を用いてボキャブラリを作成する。各単語はこのボキャブラリを用いて **one-hot-vector** であらわすことができるようになる。このベクトルをインプットに **Skip-Gram** 法を用いて周辺単語の出現確率を学習させる。その際にできた重みのパラメータがボキャブラリに含まれる各単語のベクトル表現になる。

➤ Attention Mechanism

✓ 要点のまとめ

seq2seq では長い文章への対応が難しいという問題に対する解決策。

Attention Mechanism では入力された文章の中の単語と、出力の中の単語との対応関係を学習する。そのため、長い文章を学習するには内部表現の次元も大きくなるような仕組みになっている。

深層学習レポート day4 (8点以上で合格)

➤ 強化学習

✓ 要点のまとめ

強化学習とは、長期的に報酬を最大化できるように「環境」の中で「行動」を選択するエージェントを作ることを目指す機械学習の分野。行動の結果（報酬やペナルティ）をもとに、エージェントの行動原理を形作っていく。

➤ AlphaGo

✓ 要点のまとめ

AlphaGo は Google DeepMind が開発し、2015 年に初めてプロの囲碁棋士に互先で勝利したプログラム。AlphaGo Lee ではネット囲碁の記録を大量に使用した教師あり学習と強化学習を共に使い、プログラムを作成していた。2017 年に発表された AlphaGo Zero では人間の対戦記録は一切使わず、自己対戦のみで Lee の性能を凌駕した。Lee では特徴量に様々な人間の囲碁ノウハウを考慮したものが使われていた一方で、Zero では盤面の状況のみを特徴量としている。

➤ 軽量化・高速化技術

✓ 要点のまとめ

三つの手法が紹介されていた。一つ目が「量子化」扱うデータの精度を倍精度から単精度に下げること、計算速度を上げ、メモリを節約することができる。デメリットは計算精度の低下。二つ目は「蒸留」複雑で大規模な精度の高いモデルを教師モデルとし、学習済みの重みを活用する。生徒モデルとする小規模なモデルを、より少ない試行回数で精度良く学習させることができる。三つめは「プルーニング」では、寄与の少ないニューロンの接続を切ることで、軽量化・高速化を実現する。

➤ 応用モデル

✓ 要点のまとめ

「MobileNet」通常の畳み込みでは空間方向とチャネル方向をどうじに畳み込むが、それらを別々に畳み込むことで計算量を削減した。「DenseNet」前方にある全ての層からの出力を入力とする DenseBlock を導入したネットワーク。「バッチ正規化」レイヤーの間を流れるデータを、ミニバッチ単位で正規化する。学習時間の低減、初期値依存の低減、過学習の抑制効果がある。「WeveNet」時系列データに対する畳み込みの手法。層が深くなるにつれて畳み込むリンクを離している。

➤ Transformer

✓ 要点のまとめ

2018 年に発表された BERT や 2019 年の T5 など、自然言語分野の SoTA モデルは Transformer をベースに開発されている。それまでの自然言語のモデルは CNN や RNN を使用するのが主流であったが、Transformer は Attention のみを使用した Encoder-Decoder モデルで計算量、精度ともに改善している。

✓ 実装演習サマリーと考察

サンプルコードを実行し、BLEU の評価は「約 21.97」となった。

```
[46] src, tgt = next(test_dataloader)

src_ids = src[0][0].cpu().numpy()
tgt_ids = tgt[0][0].cpu().numpy()

print('src: {}'.format(' '.join(ids_to_sentence(vocab_X, src_ids[1:-1]))))
print('tgt: {}'.format(' '.join(ids_to_sentence(vocab_Y, tgt_ids[1:-1]))))

preds, enc_slf_attns, dec_slf_attns, dec_enc_attns = test(model, src)
pred_ids = preds[0].data.cpu().numpy().tolist()
print('out: {}'.format(' '.join(ids_to_sentence(vocab_Y, trim_eos(pred_ids)))))

src: the water was cut off yesterday . </S> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD>
tgt: 昨日 水道 を 止め られ た 。 </S> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD>
out: 昨日 水 は 水 を 切っ た 。
```

BLEUの評価

```
# BLEUの評価
test_dataloader = DataLoader(
    test_X, test_Y, 128,
    shuffle=False
)
refs_list = []
hyp_list = []

for batch in test_dataloader:
    batch_X, batch_Y = batch
    preds, *_ = test(model, batch_X)
    preds = preds.data.cpu().numpy().tolist()
    refs = batch_Y[0].data.cpu().numpy()[1:].tolist()
    refs_list += refs
    hyp_list += preds
bleu = calc_bleu(refs_list, hyp_list)
print(bleu)
```

21.969125443443037

✓ 関連レポート調査

講義内で何度か出てきた「BLEU スコア」について調べてみた。

このスコアの指針は、「プロの翻訳者と答えが近ければ近いほど、その翻訳の精度は高い」である。一般的に、翻訳の答えは一つではないため、BLEU スコアを算出する際には政界を複数用意する。スコア 40 以上が高品質な翻訳とされている。欠点としては、字面しか見ていないため、類語表現が評価できない。語順が考慮されていない。などがある。

参考：<https://to-in.com/blog/102282>

➤ 物体検知・セグメンテーション

✓ 要点のまとめ

物体認識のタスクは大きく 4 つに分類される。「分類」「物体検知」「意味領域分割」「個体領域分割」意味領域分割と個体領域分割の間には、インスタンスの違いを認識するかどうかには差がある。物体認識の代表的なデータセットには「VOC12」「MS COCO18」「ILSVRC17」「OICOD18」がある。物体検出タスクにおいては、物体の分類のほか、検出した位置についても評価の必要がある。一の評価には「Area of Overlap/Area of Union」別名 Jaccord 係数が用いられる。Jaccord 係数は自然言語などの領域でも使用されている。