

深層学習 day3

Section1: 再帰型ニューラルネットワークの概念

- RNN(再帰型ニューラルネットワーク)
- 時系列データを扱うことの出来るニューラルネットワーク
- 時系列データとは音声データ、テキストデータ、株価データなど時間的な繋がりがあるもの。
- 1つ前の時間(t-1)の状態を用いて、今の時間(t)の出力を求める再帰的な構造を持つ。

RNNの式

$$z^t = f(W_{(in)}x^t + Wz^{t-1} + b)$$

$$y^t = g(W_{(out)}z^t + c)$$

z: 中間層、x: 入力、 z^{t-1} : 1つ前の中間層の値、y: 出力

確認テスト: RNNの3つの重み

RNNの3つの重みは、

- 入力から現在の中間層を定義する際にかけられる重み
- 中間層から出力を定義する際にかけられる重み

残り1つは何か？

答え：1つ前の中間層から今の中間層を定義する際にかけられる重み

BPTT

- BPTT(Back Propagation Through Time)
 - RNNにおける逆伝播
 - 誤差から微分を逆算することで不要な計算を省略

確認テスト: dz/dx

連鎖律の原理を使い、dz/dxを求めよ。

$$z=t^2, t=x+y$$

答え：dz/dx = dz/dt * dt/dx = 2t * 1 = 2t = 2(x+y)

確認テスト: y_1 を数式で表す

下図の y_1 を $x \cdot z_0 \cdot z_1 \cdot w_{in} \cdot w \cdot w_{out}$ を用いて数式で表せ。

※バイアスは任意の文字で定義せよ。

※また中間層の出力にシグモイド関数 $g(x)$ を作用させよ。

答え：

$$\begin{aligned} y_1 &= g(W_{(out)} * z_1 + c) \\ &= g(W_{(out)} * f(W_{(in)} * x_1 + W z_0 + b) + c \end{aligned}$$

実装演習

ファイル：my_3_1_simple_RNN.ipynb

以下を実施。

- [try]weight_init_stdやlearning_rate, hidden_layer_sizeを変更してみよう
- [try]重みの初期化方法を変更してみよう（Xavier）
- [try]重みの初期化方法を変更してみよう（He）
- [try]中間層の活性化関数を変更1: ReLU
- [try]中間層の活性化関数を変更2: TanH

Section2: LSTM

- RNNの課題
 - 時系列を遡るほど勾配が消失してしまう
 - 長い時系列の学習が難しい

この課題を解決するのがLSTMである。

勾配消失問題が起きやすいのはsigmoid関数である。

確認テスト: sigmoid()の微分の最大値

答え：sigmoid()を微分をした時、入力値=0で最大値0.25を取る

CEC

- CEC(Constant Error Carousel)
 - 定誤差カルーセル
 - 記憶することだけを任せるもの
 - 勾配が常に1とし、勾配消失や勾配爆発を防ぐ

- 課題：ニューラルネットワークの学習特性がない

入力・出力ゲート

- 入力ゲート
 - CECへどう覚えさせるかを重み W_i , U_i を使って指定
- 出力ゲート
 - どんな風にCECを使うかを重み W_o , U_o を使って指定

忘却ゲート

- 忘却ゲート
 - 過去の情報が不要となった場合に忘却する機能
 - 忘却ゲートの入力に対し、重み W_f , U_f を使ってどれくらい忘れるかを指定する

確認テスト: LSTMでの単語予測

「とても」という単語は空欄の予測において、なくなっても影響は及ぼさない。
この場合、どのようなゲートが作用しているか？

答え：忘却ゲート

演習チャレンジ:

LSTMでセル状態を更新し、中間層の出力を更新するコードはどれか？

(け)に当てはまるものを選び。

答え：(3) `input_gate + forget_gate`

ただ、ここの`forget_gate*c`は正しくは `forget_gate*prev_c` なのではないかと思う。

覗き穴結合

各ゲートの出力に現在のCECの値も使用して算出する手法。

Section3: GRU

LSTMではパラメータ数が多く、計算負荷が大きいという欠点があった。
GRUはパラメータを大幅に削減し、LSTMと同等以上の精度が見込める構造。

LSTMとGRUの比較

種類	構成	特徴
LSTM	入力ゲート、出力ゲート、忘却ゲート、セル	パラメータが多く計算負荷が大きい
GRU	リセットゲート、更新ゲート	パラメータが少なく、計算負荷が小さい

確認テスト:LSTMとCECの課題

答え：

- LSTMの課題：パラメータが多く計算量が多いこと
- CECの課題: 勾配が1固定で学習の仕組みがないこと

Section4: 双方向RNN

- 双方向RNN
 - 過去の情報だけでなく、未来の情報を加味して精度を向上
 - 実用例：機械翻訳、文章の推敲など自然言語処理

演習チャレンジ

双方向RNNの順伝播を行うプログラムにおいて、
空欄(か)に当てはまる処理はどれか？

答え：(4)np.concatenate([h_f, h_b[::-1]], axis=1)

Section5: Seq2Seq2

- Seq2Seq
 - Encoder-Decoderモデルとも呼ばれる。
 - 時系列データから時系列データを得る手法

確認テスト

選択肢からseq2seqについて説明しているものを選び。

答え：(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。」

演習チャレンジ

Encoder処理の空欄(き)に当てはまるのは？

答え：(1)E.dot(w)

HRED

- HRED(the hierarchical recurrent encoder-decoder)
 - Seq2seq + Context RNN
 - seq2seqでは一問一答しか出来なかったが、HREDでは文脈を引き継げるようにした。
 - 課題：当たり障りのない回答しか返さず、バリエーションに乏しい。「うん」「そうだね」など短い回答をしがち。

VHRED

- VHRED
 - HREDの改良版
 - HREDに、VAEの潜在変数の概念を追加したもの
 - HREDより長い文章を返す傾向がある

確認テスト

seq2seqとHREDとVHREDの違いを簡潔に述べよ。

答え：seq2seqは一問一答を行うモデルでエンコーダー、デコーダーを持つ。

HREDはseq2seqに文脈を引き継ぐContext RNNを追加したモデル。

VHREDはHREDの課題であった、当たり障りのない回答しか返さないという問題に対し、VAEの潜在変数の概念を追加して改良したモデルとなる。

VAE

- VAE: 変分オートエンコーダー(Variational auto-encoder)
- オートエンコーダー：教師無し学習の一つ。
 - 入力→Encoder→ z (潜在変数)→Decoder→出力
- VAEは潜在変数 z に確率分布 $z \sim N(0,1)$ を仮定したもの

確認テスト

VAEに関する下記の説明文中の空欄に当てはまる言葉を答えよ
自己符号器の潜在変数に「 z 」を導入したもの。

答え：確率分布

Section6: Word2Vec

- word2vecとは
- 単語表現を数値のベクトルに落とし込む手法
- 単語の分散表現ベクトルを得る
- CBOWモデルとSkipgramモデルがある
 - CBOWモデル：文章の単語の並びから間にある単語を予測（コンテキストからターゲットを推測）
 - Continuous Bag-of-Words
 - Skipgramモデル：ある単語から周りにある単語を予測（ターゲットからコンテキストを推測）

Section7: Attention Mechanism

seq2seqの課題：文章の長さによらず、エンコーダーで固定長のベクトルに変換してしまうため、長い文章では必要な情報が落ちてしまう。

- Attentionメカニズム
 - 「入力と出力のどの単語が関連しているか」の関連度を学習する仕組み
 - 何等かの直腸が合った時にその特徴のどこを重視すればよいかを学習する機構
 - 機械翻訳のために提案されたモデル

確認テスト

RNNとword2vec, seq2seqとAttentionの違いを簡潔に述べよ

答え：

- RNNとword2vecの違い
 - RNNは時系列データを扱うネットワークで、1つ前の中間層の出力を使用して、現在の中間層を更新する。word2vecはRNNのうち、単語をベクトル化し分散表現取得に特化したモデル。
- seq2seqとAttentionの違い
 - seq2seqでは長い文章への対応が難しいという課題があった。Attentionではこの問題に対し、「入力と出力のどの単語が関連しているのか」を学習させることで解決。