

VIP チートシート: リカレントニューラルネットワーク

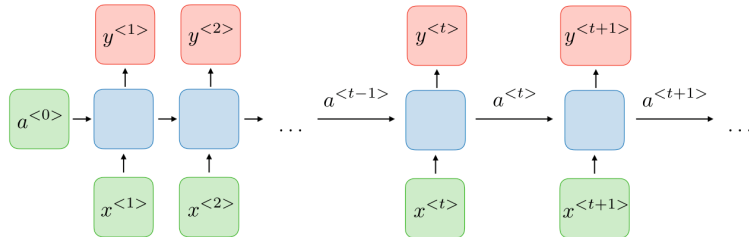
アフシンアミディ・シエルビンアミディ 著

October 7, 2019

浜野秀明・中井喜之記

概要

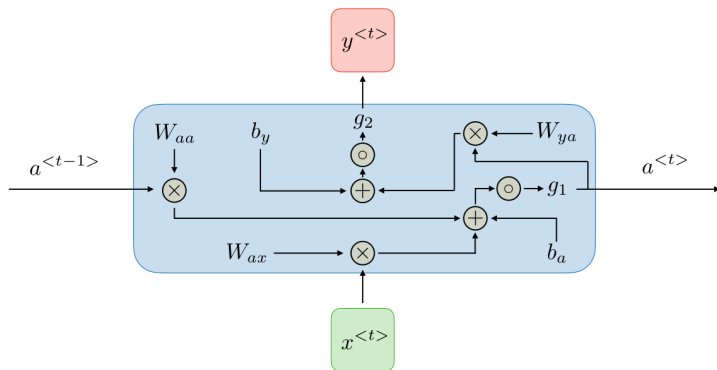
□ **一般的なRNNのアーキテクチャ** – RNNとして知られるリカレントニューラルネットワークは、隠れ層の状態を利用して、前の出力を次の入力として取り扱うことを可能にするニューラルネットワークの一種です。一般的なモデルは下記ようになります。



それぞれの時点 t において活性化関数の状態 $a^{<t>}$ と出力 $y^{<t>}$ は下記のように表現されます。

$$a^{<t>} = g_1(W_{aa}a^{<t-1>} + W_{ax}x^{<t>} + b_a) \quad \text{そして} \quad y^{<t>} = g_2(W_{ya}a^{<t>} + b_y)$$

$W_{ax}, W_{aa}, W_{ya}, b_a, b_y$ は全ての時点で共有される係数であり、 g_1, g_2 は活性化関数です。



一般的なRNNのアーキテクチャ利用の長所・短所については下記の表にまとめられています。

長所	短所
<ul style="list-style-type: none"> - 任意の長さの入力を処理できる - 入力サイズに応じてモデルサイズが大きくなる - 計算は時系列情報を考慮している - 重みは全ての時点で共有される 	<ul style="list-style-type: none"> - 遅い計算 - 長い時間軸での情報の利用が困難 - 現在の状態から将来の入力を予測不可能

□ **RNNの応用** – RNNモデルは主に自然言語処理と音声認識の分野で使用されます。以下の表に、さまざまな応用例がまとめられています。

RNNの種類	図	例
一対一 $T_x = T_y = 1$		伝統的なニューラルネットワーク
一対多 $T_x = 1, T_y > 1$		音楽生成
多対一 $T_x > 1, T_y = 1$		感情分類
多対多 $T_x = T_y$		固有表現認識
多対多 $T_x \neq T_y$		機械翻訳

□ **損失関数** – リカレントニューラルネットワークの場合、時間軸全体での損失関数 L は、各時点での損失に基づき、次のように定義されます。

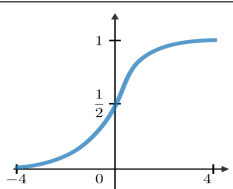
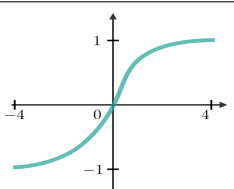
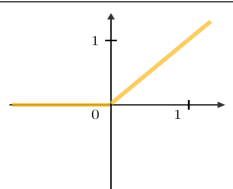
$$\mathcal{L}(\hat{y}, y) = \sum_{t=1}^{T_y} \mathcal{L}(\hat{y}^{<t>}, y^{<t>})$$

□ **時間軸での誤差逆伝播法** – 誤差逆伝播(バックプロパゲーション)が各時点で行われます。時刻 T における、重み行列 W に関する損失 L の導関数は以下のように表されます。

$$\frac{\partial \mathcal{L}^{(T)}}{\partial W} = \sum_{t=1}^T \frac{\partial \mathcal{L}^{(T)}}{\partial W} \Big|_{(t)}$$

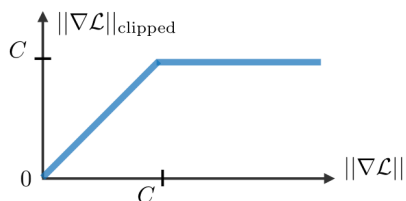
長期依存関係の処理

□ **一般的に使用される活性化関数** – RNNモジュールで使われる最も一般的な活性化関数を以下に説明します。

シグモイド	Tanh	RELU
$g(z) = \frac{1}{1 + e^{-z}}$	$g(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$	$g(z) = \max(0, z)$
		

□ **勾配消失と勾配爆発について** – 勾配消失と勾配爆発の現象は、RNNでよく見られます。これらの現象が起こる理由は、掛け算の勾配が層の数に対して指数関数的に減少/増加する可能性があるため、長期の依存関係を捉えるのが難しいからです。

□ **勾配クリッピング** – 誤差逆伝播法を実行するときに時折発生する勾配爆発問題に対処するために使用される手法です。勾配の上限値を定義することで、実際にこの現象が抑制されます。



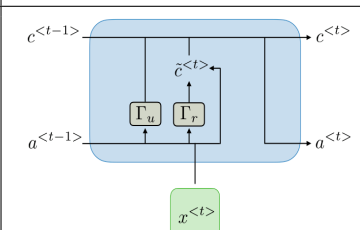
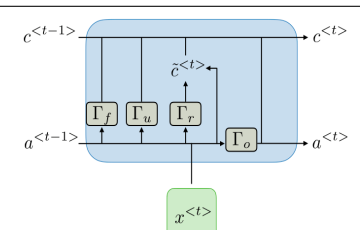
□ **ゲートの種類** – 勾配消失問題を解決するために、特定のゲートがいくつかのRNNで使用され、通常明確に定義された目的を持っています。それらは通常 Γ と記され、以下のように定義されます。

$$\Gamma = \sigma(Wx^{<t>} + Ua^{<t-1>} + b)$$

ここで、 W, U, b はゲート固有の係数、 σ はシグモイド関数です。主なものは以下の表にまとめられています。

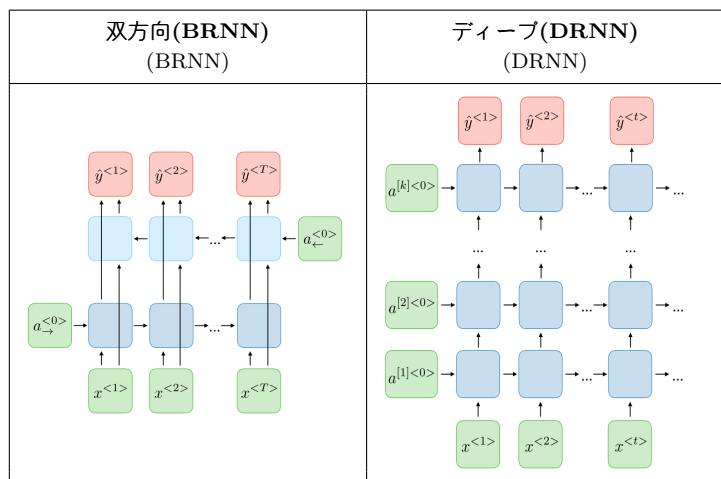
ゲートの種類	役割	下記で使われる
更新ゲート Γ_u	過去情報はどのくらい重要ですか？	GRU, LSTM
関連ゲート Γ_r	前の情報を削除しますか？	GRU, LSTM
忘却ゲート Γ_f	セルを消去しますか？しませんか？	LSTM
出力ゲート Γ_o	セルをどのくらい見せますか？	LSTM

□ **GRU/LSTM** – ゲート付きリカレントユニット (GRU) およびロングショートタームメモリユニット (LSTM) は、従来のRNNが直面した勾配消失問題を解決しようとしています。LSTMはGRUを一般化したものです。以下は、各アーキテクチャを特徴づける式をまとめた表です。

	ゲート付きリカレントユニット (GRU)	ロングショートタームメモリ (LSTM)
$\tilde{c}^{<t>}$	$\tanh(W_c[\Gamma_r * a^{<t-1>}, x^{<t>}] + b_c)$	$\tanh(W_c[\Gamma_r * a^{<t-1>}, x^{<t>}] + b_c)$
$c^{<t>}$	$\Gamma_u * \tilde{c}^{<t>} + (1 - \Gamma_u) * c^{<t-1>}$	$\Gamma_u * \tilde{c}^{<t>} + \Gamma_f * c^{<t-1>}$
$a^{<t>}$	$c^{<t>}$	$\Gamma_o * c^{<t>}$
依存関係		

備考：記号 $*$ は2つのベクトル間の要素ごとの乗算を表します。

□ **RNNの変種** – 以下の表は、一般的に使用されている他のRNNアーキテクチャをまとめたものです。



単語表現の学習

この節では、 V は語彙、そして $|V|$ は語彙のサイズを表します。

□ **表現のテクニック** – 単語を表現する2つの主な方法は、以下の表にまとめられています。

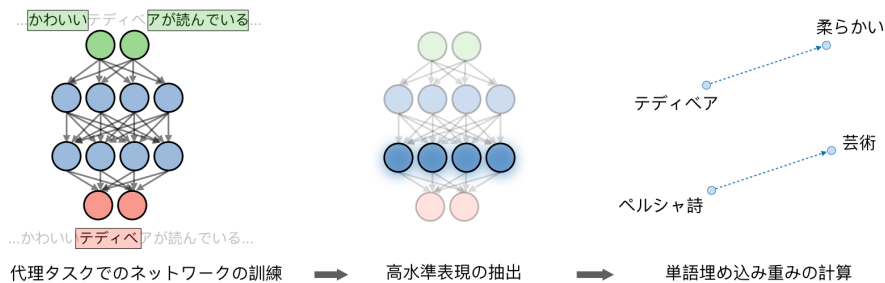
1-hot表現	単語埋め込み
<ul style="list-style-type: none"> - owと表記される - 素朴なアプローチ、類似性情報なし 	<ul style="list-style-type: none"> - ewと表記される - 単語の類似性を考慮に入れる

□ **埋め込み行列** – 与えられた単語 w に対して、埋め込み行列 E は、以下のように1-hot表現 o_w を埋め込み行列 e_w に写像します。

$$e_w = E o_w$$

注：埋め込み行列は、ターゲット/コンテキスト尤度モデルを使用して学習できます。

□ **Word2vec** – Word2vecは、ある単語が他の単語の周辺にある可能性を推定することで、単語の埋め込みの重みを学習することを目的としたフレームワークです。人気のあるモデルは、スキップグラム、ネガティブサンプリング、およびCBOWです。



□ **スキップグラム** – スキップグラムword2vecモデルは、あるターゲット単語 t がコンテキスト単語 c と一緒に出現する確率を評価することで単語の埋め込みを学習する教師付き学習タスクです。 t に関するパラメータを θ_t と表記すると、その確率 $P(t|c)$ は下記の式で与えられます。

$$P(t|c) = \frac{\exp(\theta_t^T e_c)}{\sum_{j=1}^{|V|} \exp(\theta_j^T e_c)}$$

注： softmax 部分の分母の語彙全体を合計するため、このモデルの計算コストは高くなります。 $CBOW$ は、ある単語を予測するため周辺単語を使用する別のタイプのword2vecモデルです。

□ **ネガティブサンプリング** – ロジスティック回帰を使用したバイナリ分類器のセットで、特定の文脈とあるターゲット単語が同時に出現する確率を評価することを目的としています。モデルは k 個のネガティブな例と1つのポジティブな例のセットで訓練されます。コンテキスト単語 c とターゲット単語 t が与えられると、予測は次のように表現されます。

$$P(y = 1|c, t) = \sigma(\theta_t^T e_c)$$

注：この方法の計算コストは、スキップグラムモデルよりも少ないです。

□ **GloVe** – GloVeモデルは、単語表現のためのグローバルベクトルの略で、共起行列 X を使用する単語の埋め込み手法です。ここで、各 $X_{i,j}$ は、ターゲット i がコンテキスト j で発生した回数を表します。そのコスト関数 J は以下の通りです。

$$J(\theta) = \frac{1}{2} \sum_{i,j=1}^{|V|} f(X_{i,j})(\theta_i^T e_j + b_i + b'_j - \log(X_{i,j}))^2$$

ここで、 f は $X_{i,j} = 0 \implies f(X_{i,j}) = 0$ となるような重み関数です。このモデルで e と θ が果たす対称性を考えると、最後の単語の埋め込み $e_w^{(\text{final})}$ は下記のようにになります。

$$e_w^{(\text{final})} = \frac{e_w + \theta_w}{2}$$

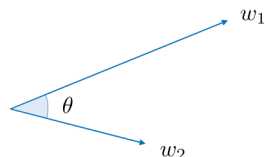
注：学習された単語の埋め込みの個々の要素は、必ずしも解釈可能ではありません。

単語の比較

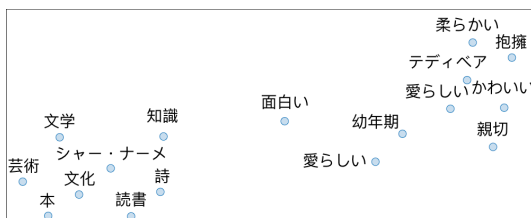
□ **コサイン類似度** – 単語 w_1 と w_2 のコサイン類似度は次のように表されます。

$$\text{similarity} = \frac{w_1 \cdot w_2}{\|w_1\| \|w_2\|} = \cos(\theta)$$

注： θ は単語 w_1 と w_2 間の角度です。



□ **t-SNE** – t-SNE (t-分布型確率的近傍埋め込み) は、高次元埋め込みから低次元埋め込み空間への次元削減を目的とした手法です。実際には、2次元空間で単語ベクトルを視覚化するために使用されます。



言語モデル

□ **概要** – 言語モデルは文の確率 $P(y)$ を推定することを目的としています。

□ **n-gramモデル** – このモデルは、トレーニングデータでの出現数を数えることによって、ある表現がコーパスに出現する確率を定量化することを目的とした単純なアプローチです。

□ **パープレキシティ** – 言語モデルは一般的に、PPとも呼ばれるパープレキシティメトリックを使用して評価されます。これは、単語数 T により正規化されたデータセットの逆確率と解釈できます。パープレキシティは低いほど良く、次のように定義されます。

$$PP = \prod_{t=1}^T \left(\frac{1}{\sum_{j=1}^{|V|} y_j^{(t)} \cdot \hat{y}_j^{(t)}} \right)^{\frac{1}{T}}$$

注：PPはt-SNEで一般的に使用されています。

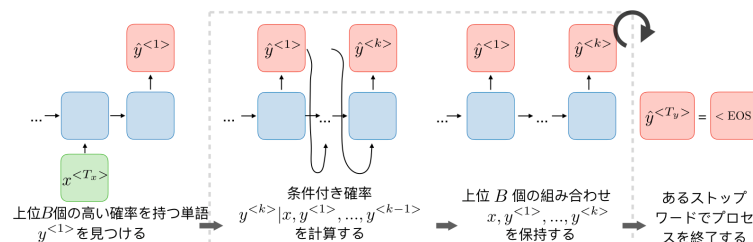
機械翻訳

□ **概要** – 機械翻訳モデルは、エンコーダーネットワークのロジックが最初に付加されている以外は、言語モデルと似ています。このため、条件付き言語モデルと呼ばれることもあります。目的は次のような文 y を見つけることです。

$$y = \arg \max_{y^{<1>}, \dots, y^{<T_y>}} P(y^{<1>}, \dots, y^{<T_y>} | x)$$

□ **ビーム検索** – 入力 x が与えられたとき最も可能性の高い文 y を見つけるために、機械翻訳と音声認識で使用されるヒューリスティック探索アルゴリズムです。

- ステップ 1：上位 B 個の高い確率を持つ単語 $y^{<1>}$ を見つける。
- ステップ 2：条件付き確率 $y^{<k>} | x, y^{<1>}, \dots, y^{<k-1>}$ を計算する。
- ステップ 3：上位 B 個の組み合わせ $x, y^{<1>}, \dots, y^{<k>}$ を保持する。



注意：ビーム幅が1に設定されている場合、これは単純な貪欲法と同等です。

□ **ビーム幅** – ビーム幅 B はビーム検索のパラメータです。 B の値を大きくするとより良い結果が得られますが、探索パフォーマンスは低下し、メモリ使用量が増加します。 B の値が小さいと結果が悪くなりますが、計算量は少なくなります。 B の標準値は10前後です。

□ **文章の長さの正規化** – 数値の安定性を向上させるために、ビーム検索は通常次のように正規化 (対数尤度正規化) された目的関数に対して適用されます。

$$\text{Objective} = \frac{1}{T_y^\alpha} \sum_{t=1}^{T_y} \log \left[p(y^{<t>} | x, y^{<1>}, \dots, y^{<t-1>}) \right]$$

注：パラメータ α は緩衝パラメータと見なされ、その値は通常0.5から1の間です。

□ **エラー分析** – 予測された \hat{y} の翻訳が良くない場合、以下のようなエラー分析を実行することで、なぜ y^* のような良い翻訳を得られなかったのか考えることが可能です。

症例	$P(y^* x) > P(\hat{y} x)$	$P(y^* x) \leq P(\hat{y} x)$
根本原因	ビーム検索の誤り	RNNの誤り
改善策	ビーム幅の拡大	- さまざまなアーキテクチャを試す - 正規化 - データをさらに取得

□ **Bleuスコア** – Bleu (Bilingual evaluation understudy) スコアは、n-gramの精度に基づき類似性スコアを計算することで、機械翻訳がどれほど優れているかを定量化します。以下のように定義されています。

$$\text{bleu score} = \exp \left(\frac{1}{n} \sum_{k=1}^n p_k \right)$$

ここで、 p_n はn-gramでのbleuスコアで下記のようにだけ定義されています。

$$p_n = \frac{\sum_{\text{n-gram} \in \hat{y}} \text{count}_{\text{clip}}(\text{n-gram})}{\sum_{\text{n-gram} \in \hat{y}} \text{count}(\text{n-gram})}$$

注：人為的に水増しされたブルースコアを防ぐために、短い翻訳評価には簡潔さへのペナルティが適用される場合があります。

アテンション

□ アテンションモデル – このモデルを使用するとRNNは重要であると考えられる入力の特定期分に注目することができ、得られるモデルの性能が実際に向上します。時刻 t において、出力 $y^{<t>}$ が活性化関数 $a^{<t'>}$ とコンテキスト $c^{<t>}$ とに払うべき注意量を $\alpha^{<t,t'>}$ と表記すると次のようになります。

$$c^{<t>} = \sum_{t'} \alpha^{<t,t'>} a^{<t'>} \quad \text{および} \quad \sum_{t'} \alpha^{<t,t'>} = 1$$

注：アテンションスコアは、一般的に画像のキャプション作成および機械翻訳で使用されています。



かわいいディディベアがベルシャ文学を読んでいます



かわいいディディベアがベルシャ文学を読んでいます

□ アテンションの重み – 出力 $y^{<t>}$ が活性化関数 $a^{<t'>}$ に払うべき注意量 $\alpha^{<t,t'>}$ は次のように計算されます。

$$\alpha^{<t,t'>} = \frac{\exp(e^{<t,t'>})}{\sum_{t''=1}^{T_x} \exp(e^{<t,t''>})}$$

注：この計算の複雑さは T_x に関して O 次です。

★ ★ ★