Singularity

RNN camp #1

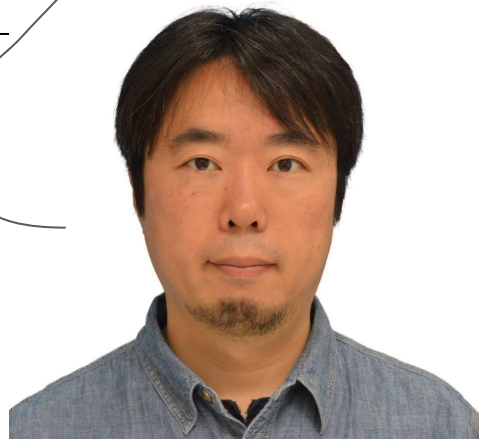
浅川伸一 Shin Asakawa <asakawa@ieee.org>

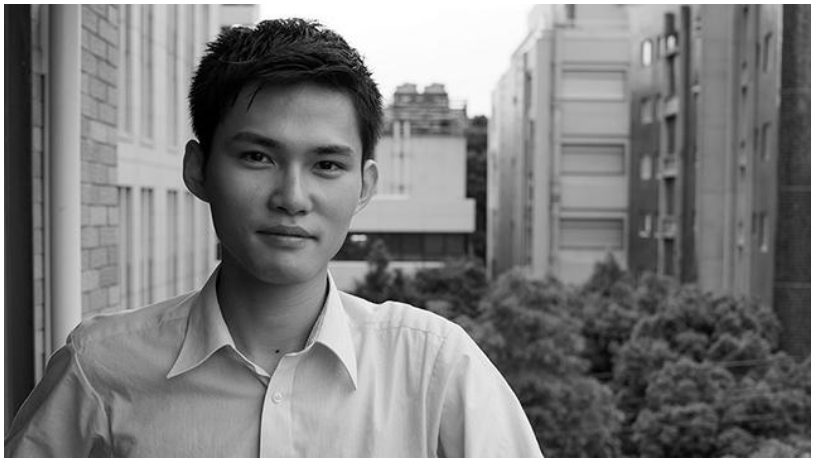
注意事項

- 本日のトークでは途中でペアワーク，グループワークを行ないます。隣の席に座っている方と簡単な自己紹介をしてお互いに面通ししてください。
- GitHub からダウンロードをお願いしますhttps://github.com/ShinAsakawa/rnncamp.git
- Python, C++ コンパイラはインストールされていますか？
  - `pip install --upgrade autograd`
  - `pip install --upgrade termcolor`

謝辞

- KUNO 佐藤傑様
- C8 lab 新村拓也様
- Google 佐藤一憲様





本日の予定

19:00 - 19:10 自己紹介，進め方についての注意事項
19:10 - 19:50 リカレントニューラルネットワークの概要
19:50 - 20:00 休憩
20:00 - 20:40 バックプロパゲーションスルータイム
20:40 - 21:00 実習と質疑応答

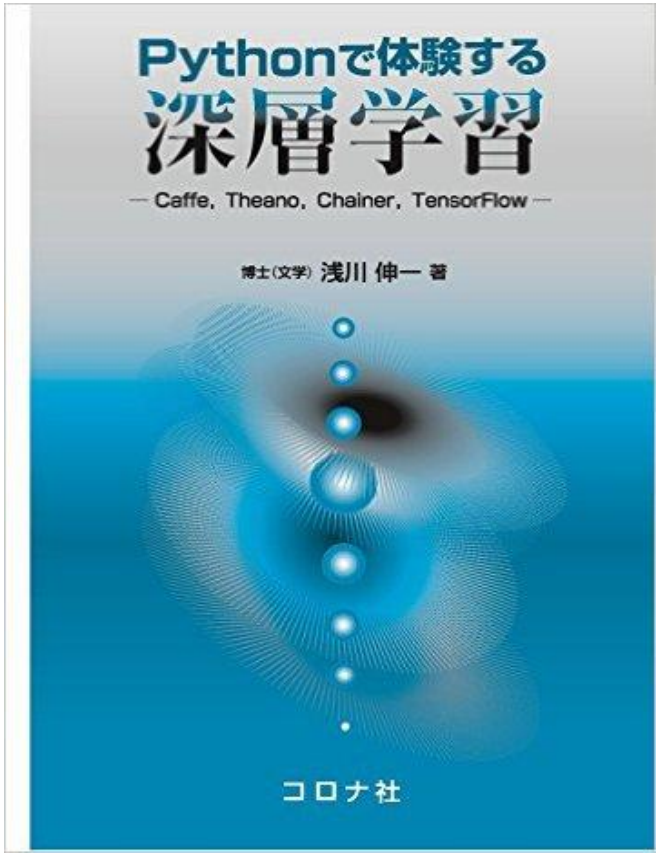
メニュー

1. 自己紹介
2. RNN camp 計画(案)
3. RNN camp #1
    3.1. リカレントニューラルネットワークとは何か
    3.2. リカレントニューラルネットワークの最近の成果
    3.3. 古典的リカレントニューラルネットワーク
    3.4. ミコロフ革命
    3.5. バックプロパゲーションスルータイム

1. 自己紹介

自己紹介 浅川伸一

博士(文学) 東京女子大学情報処理センター勤務。早稲田大学在学時はピアジェの発生論敵認識論に心酔する。卒業後エルマンネットの考案者ジェフ・エルマンに師事，薫陶を受ける。以来人間の高次認知機能をシミュレートすることを通して知的であるとはどういうことかを考えていると思っていた。著書に「ディープラーニング，ビッグデータ，機械学習あるいはその心理学」(2015) 新曜社。「ニューラルネットワークの数理的基礎」「脳損傷とニューラルネットワークモデル，神経心理学への適用例」いずれも守一雄他編「コネクショニストモデルと心理学」(2001) 北大路書房など



Python で体験する深層学習，コロナ社, (7月26日発売).https://www.amazon.co.jp/dp/4339028517/

RNN camp の目的

深層学習の一つリカレントニューラルネットワークの紹介，情報共有

可能性と限界を知りつつ応用問題を考える機会を持ちたい

RNN camp の諸元

- プロジェクトページhttp://www.cis.twcu.ac.jp/~asakawa/rnncamp/
- ソースコードhttps://www.github.com/shinasakawa/rnncamp
- ハッシュタグ #rnncamp

2. RNN camp 計画(案)

RNN camp 今後の計画

- 第1回 SRN, BPTT, 確率的勾配降下法(今回)
- 第2回 LSTM, GRU, BiRNN, 最適化，正規化，勾配消失／爆発問題(8月または9月)
- 第3回 NIC, text2image, 注意の導入，1ショット／0ショット学習，画像チューリングチャレンジ(9月または10月)
- 第4回 QA システム, 画像QA システム, ニューラルチューリングマシン, ニューラルGPU, メモリーネットワーク(10月または11月)

告知（別プロジェクト）

- **TensorFlowと機械学習に必要な数学を基礎から学ぶ会**
- 開催時期
  - 2016年8月下旬開始予定。隔週または3週毎のウィークディ19時から21時くらい
- 開催場所　未定（おそらく都内）
- 対象者　機械学習に強い興味を抱く初心者
- 参加費　無料
- Google+ のコミュニティ Math primer for TensorFlow ja で案内、告知、募集（「Tensorflow と機械学習を理解するための涙なしの数学入門」は却下された）または tensorflow.ja@gmail.com へ申し込み希望メールを送る

3.1 リカレントニューラルネットワークとは何か

3.1.1. 知性とは

知性 ≒ 学習能力，知性 ≒ 予測能力，知性 ≒ 状況判断力

- 画像分類：教師あり学習，損失関数の最小化 max p(ラベル|画像) ← 深層フィードフォワード型ニューラルネット
- 系列情報処理（言語情報処理）：系列予測 max $p(x_t \mid x_{t-1}, x_{t-2}, \ldots)$ ←リカレントニューラルネットワーク

今まで観察してきた事実（履歴）から次に起こる事象を予測

- 強化学習 ：報酬予測を学習信号とする

3.1.2. リカレントニューラルネットワークの仲間

- アトラクターネットワーク
- ホップフィールドネットワーク
- エコーステートネットワーク
- ボルツマンマシン（制限付きではない方）
- …

3.1.3 ヒントン先生曰く https://www.youtube.com/watch?v=VhmE_UXDOGs

- 任意の文章を思考ベクトルへ変換，文書とは思考ベクトルの系列
- 深層リカレントニューラルネットワークによる思考ベクトル系列の学習

  推論，理解へ到達する可能性

- 人間のレベルの理解に到達するためには数億，数兆のニューロンが必要

  古典的統計学：雑音除去 ----> AI：分布の学習

3.1.4. リカレントニューラルネットワークの特徴

1) 過去の状態を保持する中間層
2) 非線形性
3) 深層化（多層化）

しかし... 1980年代からの論文を紐解くと，黒魔法の数々

勾配チェック，勾配クリップ，勾配正規化，忘却バイアス，様々な初期化／正規化／正則化

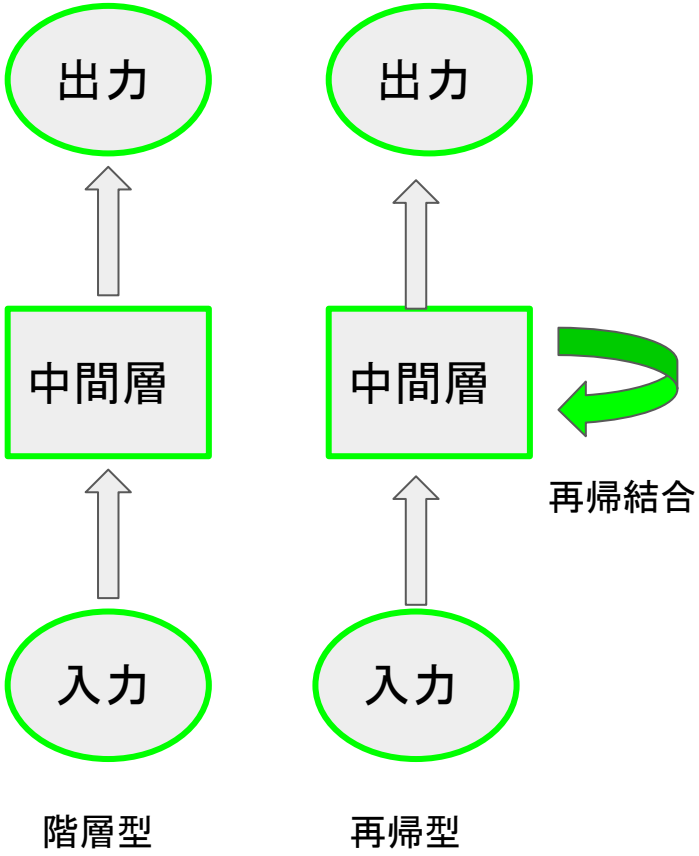
3.1.5. 近年の進歩

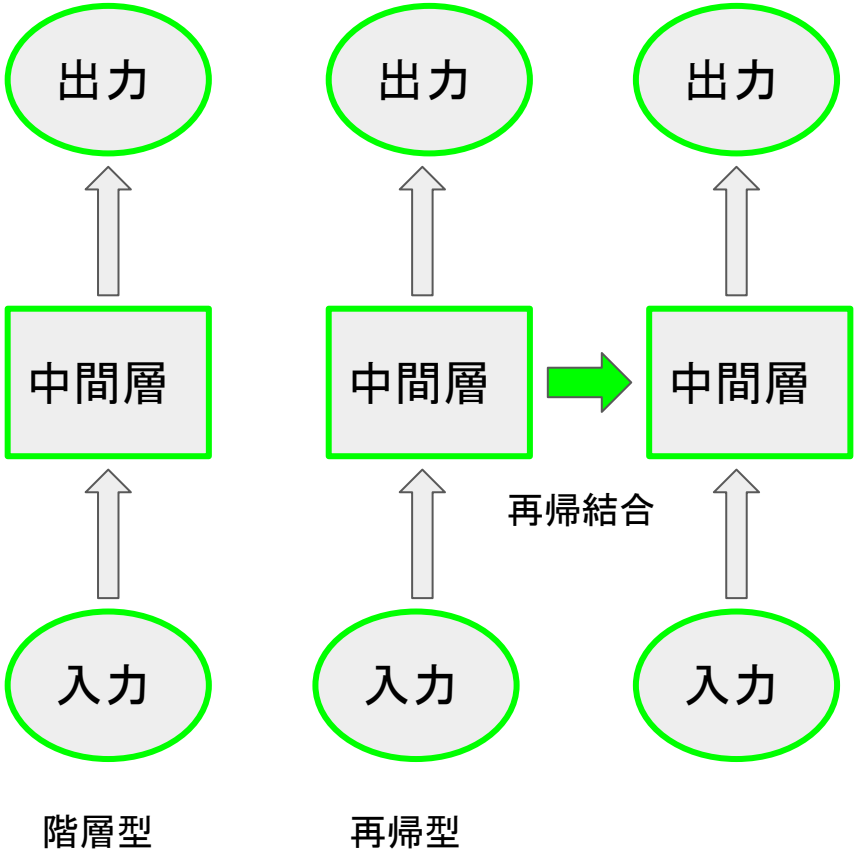
1. 黒魔法が整備
2. 演算速度が向上した
3. 記憶容量が増大した
4. 内部状態（短期記憶）を（長期的に）保持する素子（長期の短期記憶 Long Short-Term Memory: LSTM), GRU
5. 従来手法を凌駕 NLP, MT, V-QA, NIC,...
6. LSTMを基本素子としてネットワーク構造の作り込み
   : NTM, Neural GPU, Memory Network などの発展

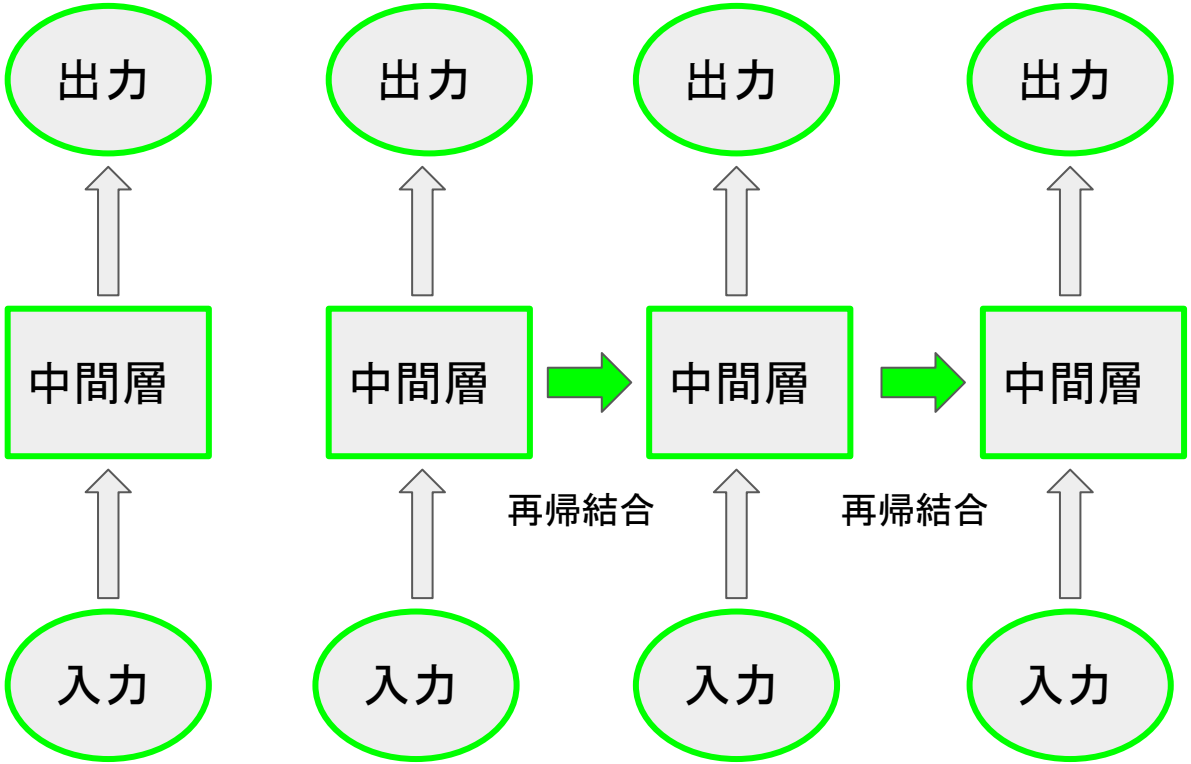
3.1.6. 系列情報を扱う手法の比較

- 内部状態無しモデル
  - 自己回帰モデル AR ≒ NetTalk, ベンジオ(2003)
- 内部状態有りモデル:
  - 隠れマルコフモデル HMM
  - 線形力学系モデル Linear dynamical systems
    - データ同化，カルマンフィルター

$$\frac{d}{dt}\boldsymbol{x}(t)=\boldsymbol{A}\boldsymbol{x}(t)$$


出力
出力
中間層
中間層
再帰結合
入力
入力
階層型
再帰型


出力
出力
出力
中間層
中間層
中間層
再帰結合
入力
入力
入力
階層型
再帰型



階層型　　再帰型

時間発展。時間方向→に見れば多層ニューラルねっt

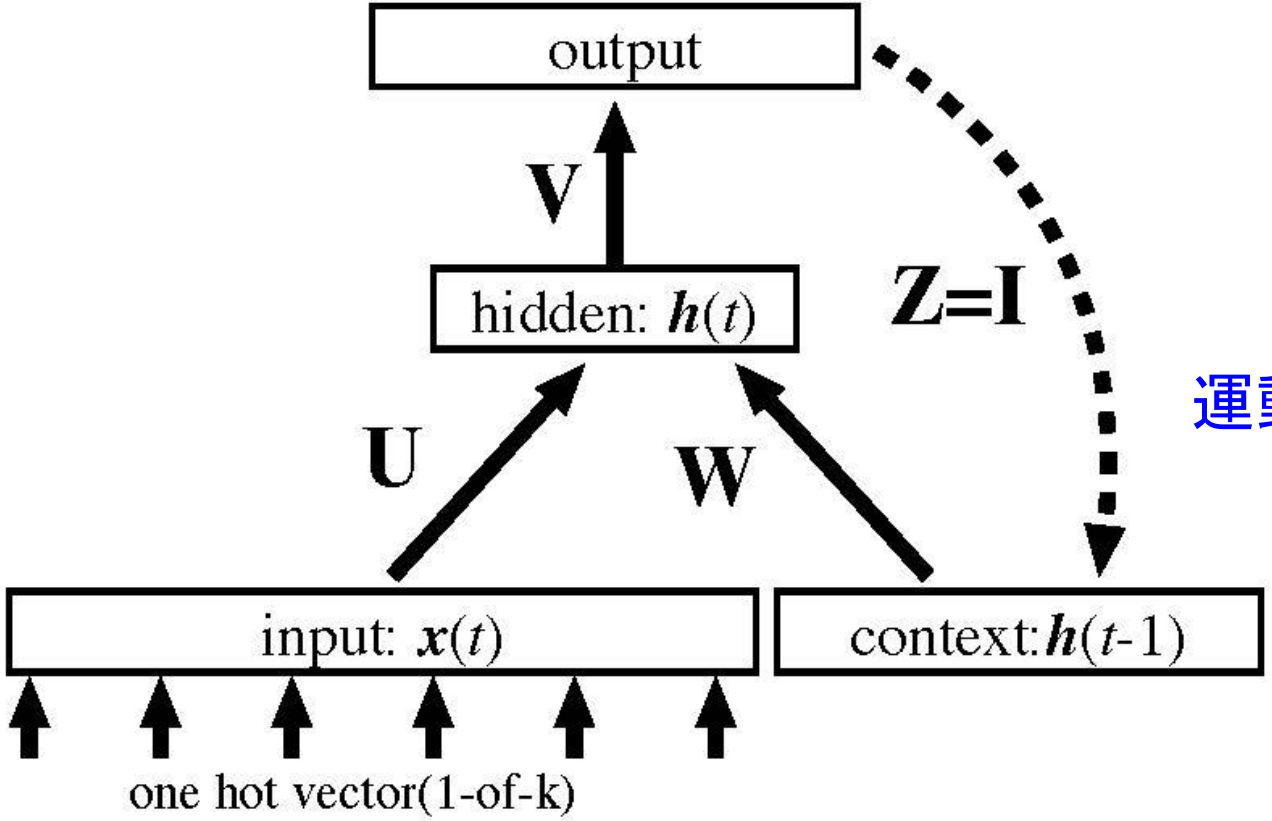
3.2 最近の成果

リカレントニューラルネットワークの成果(SOTAを含む)

1. 手書き文字認識(Graves et al., 2009)
2. 音声認識(Graves & Jaitly, 2014; Graves, Mohamed, & Hinton, 2013)
3. 手書き文字生成(Graves, 2013)
4. 系列学習(Sutskever, Vinyals, & Le, 2014)
5. 機械翻訳(Bahdanau, Cho, & Bengio, 2015; Luong, Sutskever, Le, Vinyals, & Zaremba, 2015)
6. 画像脚注付け(Kiros, Salakhutdinov, & Zemel, 2014; Vinyals, Toshev, Bengio, & Erhan, 2015)
7. 構文解析(Vinyals et al., 2015)
8. プログラムコード生成(Zaremba & Sutskever, 2015)

Actor is Schmithuber who proposed LSTM https://www.youtube.com/watch?v=-OodHtJ1saY

3.3 古典的リカレントニューラルネットワーク



運動制御

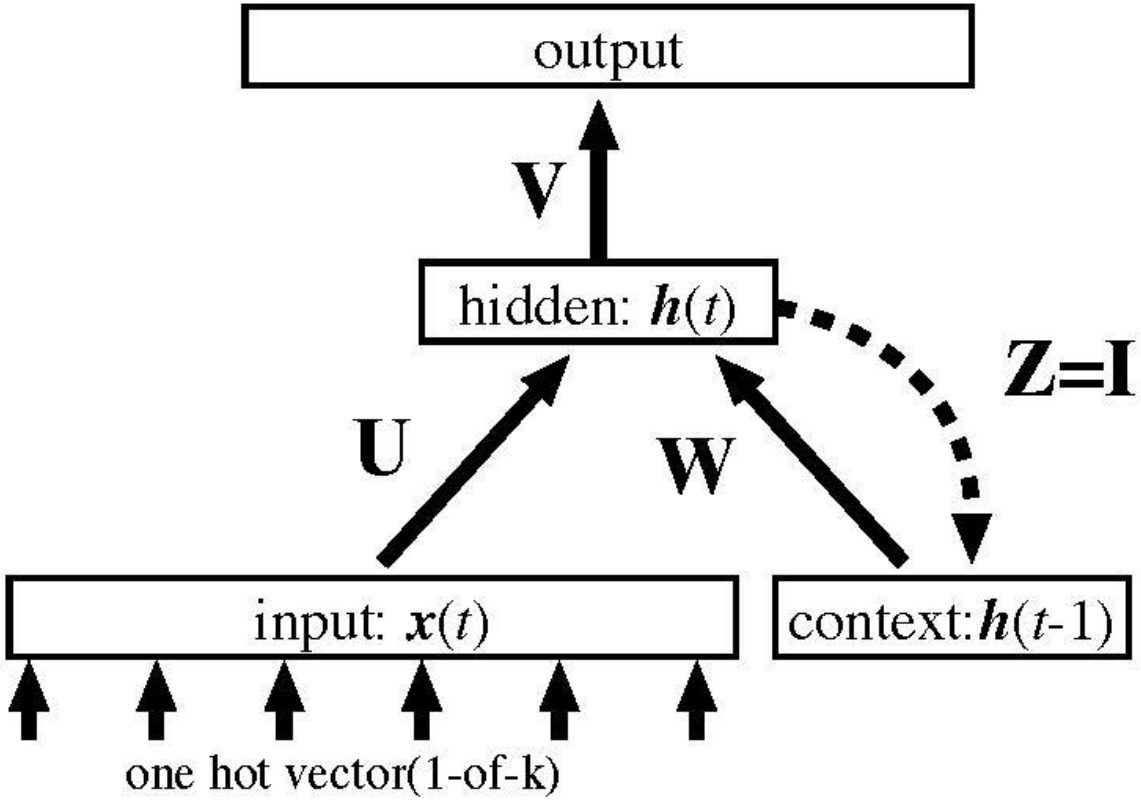
マイケル・ジョーダン発案のジョーダンネット(1986)

だが彼ではない！

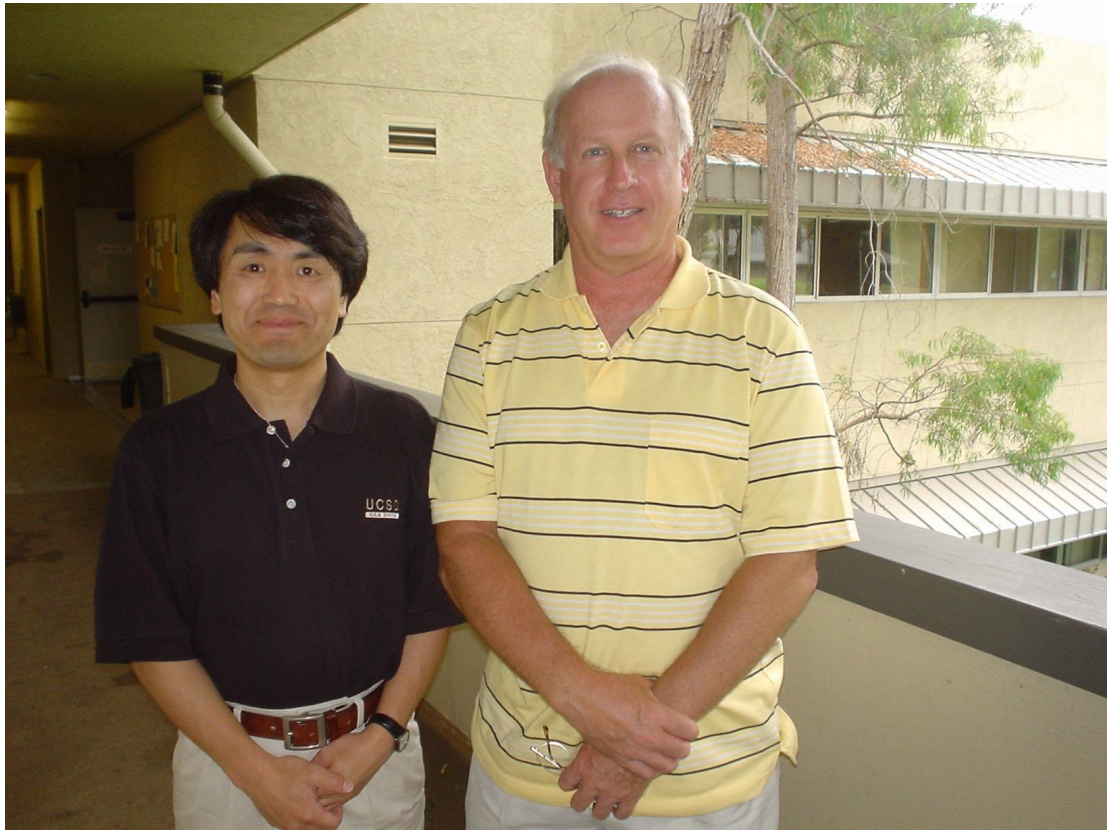
マイケル・エアー・ジョーダン

マイケル・アーヴィン・ジョーダン

現ジャーナルオブマシンラーニング現編集長
現人神。ミスター機械学習。混合エキスパートモデル，トピックモデル（中華料理屋過程，中華料理フランチャイズ過程,…）



エルマンネット（1990, 1993）



師匠ジェフ・エルマンと

軽く実習

1. カルパセィさんの min-char-rnn.py
2. 拙作 elman.py 暴力的に画面にグラフを描画します
3. 1 は文字レベルのエルマンネット，２は単語レベルのエルマンネットです。
4. 一般に日本語の言語モデルでは分かち書きの前処理が必要
5. だが文字レベルのリカレントニューラルネットワークで従来手法を上回る性能のモデルが報告されている(Chung et al.2016)

elman.py によるペアワーク

コマンドライン引数
--activate_f 活性化関数 [tanh|logistic|relu|elu]
--grad_clip 勾配クリップ
--hidden 中間層のニューロン数
--lr 学習係数
--max_iter 最大繰返し数
--sample_n 予測する単語数
--seed 乱数の種
--seq_length 系列長--snapshot_t スナップショットの間隔
--train 訓練データファイル名

elman.py によるペアワーク

ペアを組んだ相手と同じ条件で 活性化関数 logistic と tanh とを比較する

$$\sigma(x)=\frac{1}{1+exp(-x)} \qquad \phi(x)=\frac{exp(x)-exp(-x)}{exp(x)+exp(-x)}$$

$$\sigma'(x)=\sigma(x)(1-\sigma(x)) \qquad \phi'(x)=1-\sigma(x)^2$$

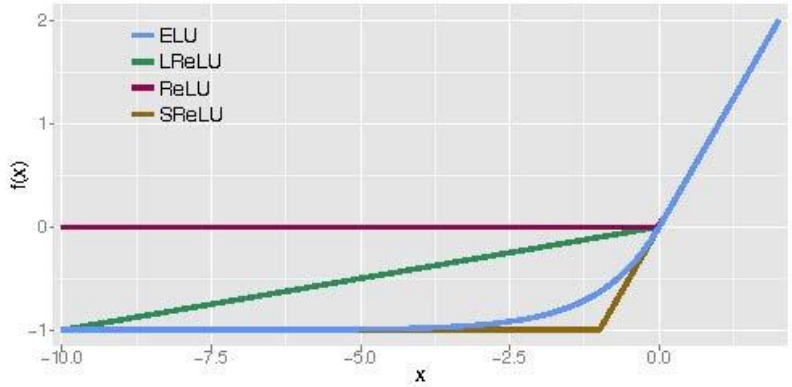
他の条件を変更して学習結果を確認する
損失関数が小さくなった方が勝ち

LeCun のレシピ論文以来 logistic 関数の替わりに tanh を用いるのがスタンダードであった(2012年までは)

今や 整流線形ユニットReLU，指数線形ユニットelu

Clevert, Unterthiner & Sepp Hochreiter(2016)

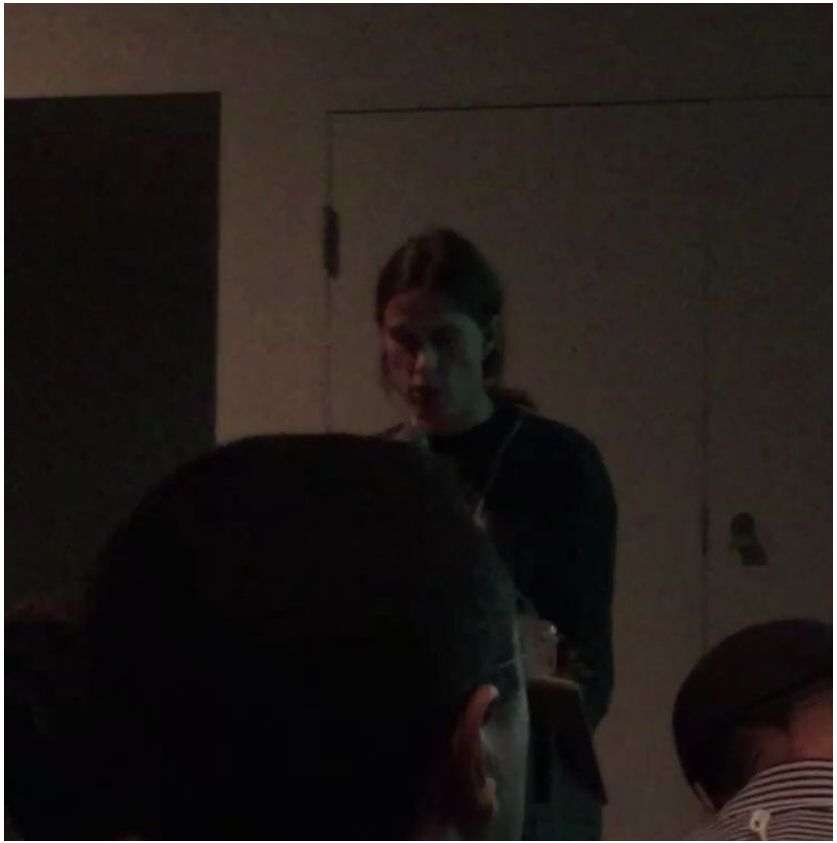
ReLU は Krizensky(2012)
で有名



$$f(x) = \begin{cases} x & \text{if } x > 0 \\ \alpha\left(\exp(x) - 1\right) & \text{if } x \leq 0 \end{cases}, \quad f'(x) = \begin{cases} 1 & \text{if } x > 0 \\ f(x) + \alpha & \text{if } x \leq 0 \end{cases}.$$

3.4 ミコロフ革命

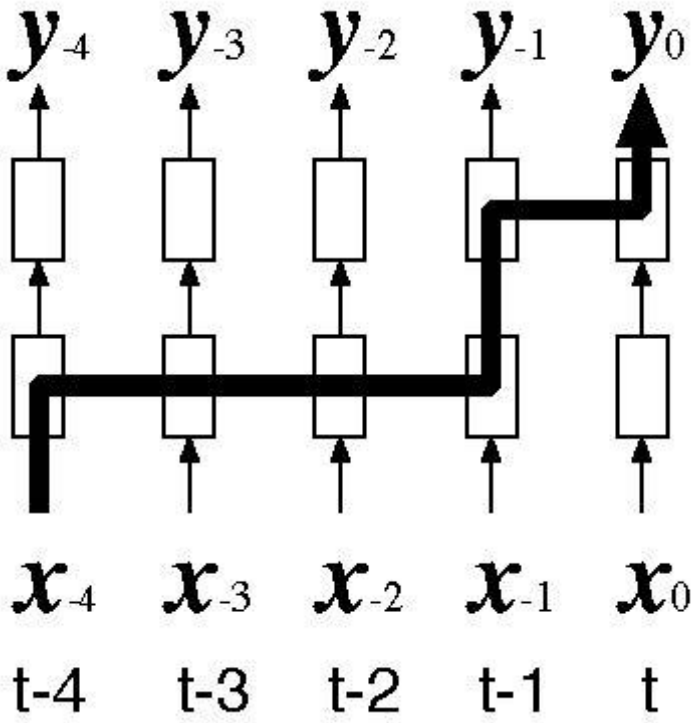
Tomáš Mikolov



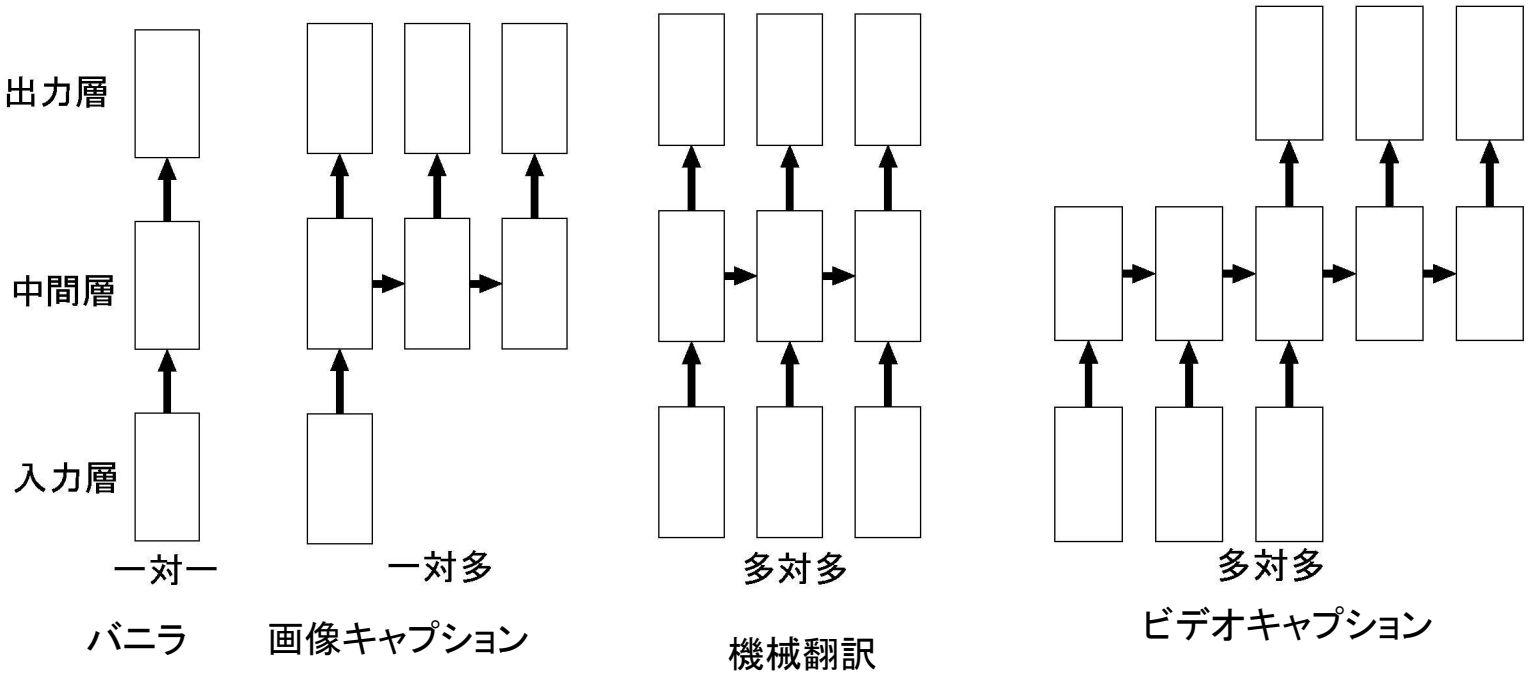
Tomas Mikolov @NIPS2015 RAM ワークショップにて

RAM：reasoning, attention, and memory

3.4.1 長距離依存



リカレントニューラルネットワークの様々な入出力形態



リカレントニューラネット再掲載

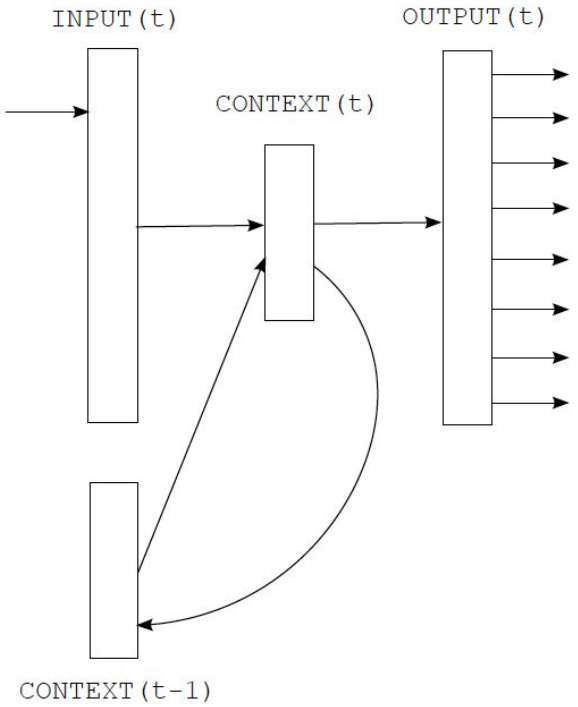


Figure 1: *Simple recurrent neural network.*

ボーデンの図

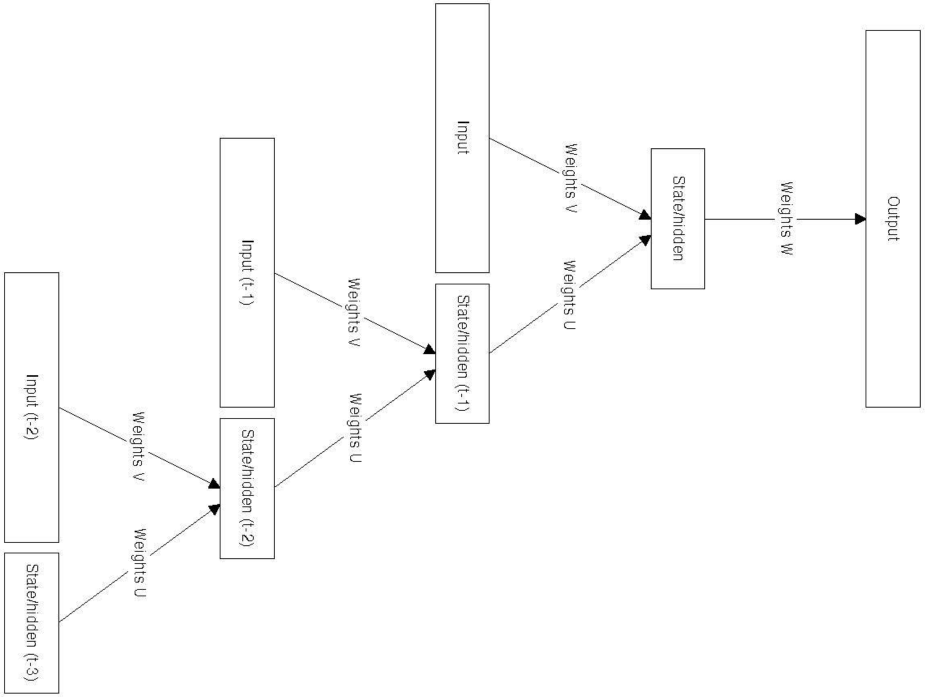


Figure 5: The effect of unfolding a network for BPTT ($\tau = 3$).

3.4.1 ミコロフ革命

- ニューラルネットワーク言語モデル
- 訓練アルゴリズム

リカレントニューラルネットワーク

エントロピー最大化言語モデル

3.4.2 ミコロフ革命

統計的言語モデル 単語系列に確率を与える
良い言語モデルは有意味文に高い確率を与え，曖昧な文には低い確率を与える

言語モデルは人工知能の問題

3.4.3 ミコロフ革命チューリングテスト

ューリングテストは原理的に言語モデルの問題とみなすことが可能
会話の履歴が与えられた時，良い言語モデルは正しい応答に高い確率を与える
　例:
P(月曜|今日は何曜日ですか？）＝？

3.4.4 ミコロフ革命チューリングテスト

ューリングテストは原理的に言語モデルの問題とみなすことが可能
会話の履歴が与えられた時，良い言語モデルは正しい応答に高い確率を与える
　例:
P(月曜|今日は何曜日ですか？）＝？

3.4.5 ミコロフ革命 N-グラム言語モデル

文脈h の中で単語w が何回出現したかをカウント。
観測した全ての文脈h で正規化

$$p(w|h) = \frac{c(h,w)}{c(h)}$$

3.4.6 ミコロフ革命 N-グラム言語モデル

類似した言語履歴h について, N-gram 言語モデルは言語履歴h が完全一致することを要請
実用的には，N-gram 言語モデルはN 語の単語系列パターンを表象するモデル
N-gram 言語モデルではN の次数増大に従って，パラメータは指数関数的に増大する

3.4.7 ミコロフ革命 N-グラム言語モデル

類似した言語履歴h について, N-gram 言語モデルは言語履歴h が完全一致することを要請。

実用的には，N-gram 言語モデルはN 語の単語系列パターンを表象するモデル
N-gram 言語モデルでは N の次数増大に従って，パラメータは指数関数的に増大する。

パラメータ推定に必要な言語情報のコーパスサイズは，次数増大に伴って，急激に増大する

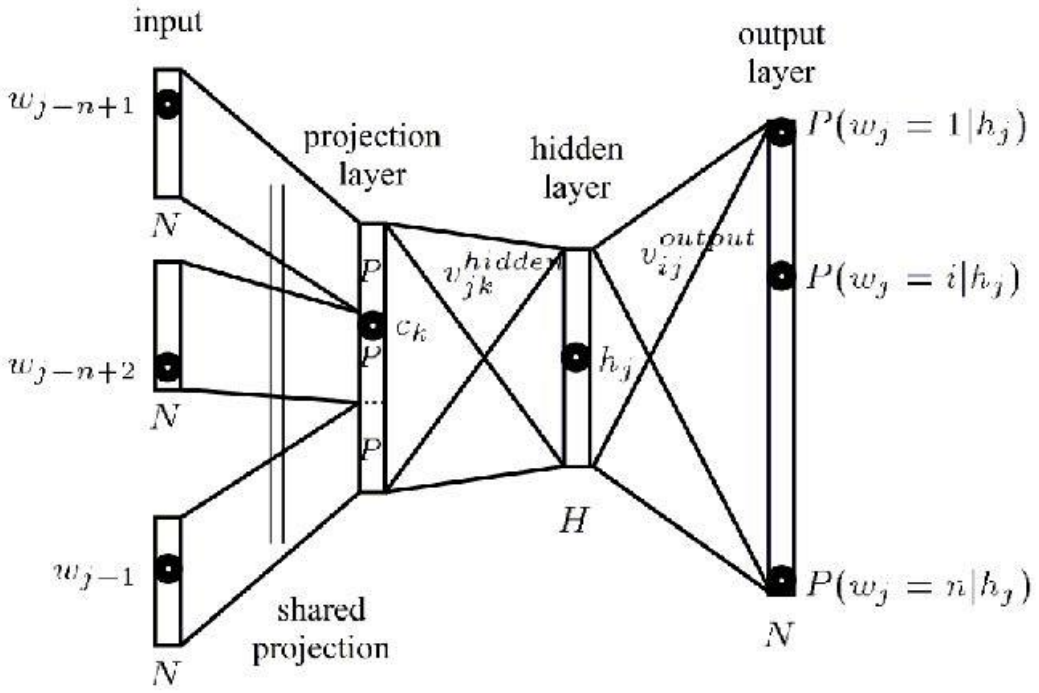
3.4.8 ミコロフ革命 RNN 言語モデル

スパースな言語履歴h は低次元空間へと射影される。類似した言語履歴は群化する
類似の言語履歴を共有することで，ニューラルネットワーク言語モデルは頑健(訓練データから推定すべきパラメータが少ない)

3.4.9 ミコロフ革命 RNN 言語モデル

スパースな言語履歴h は低次元空間へと射影される。類似した言語履歴は群化する
類似の言語履歴を共有することで，ニューラルネットワーク言語モデルは頑健(訓練データから推定すべきパラメータが少ない)

3.4.10 ミコロフ革命 参照言語モデル



3.4.12 ミコロフ革命 RNNLM

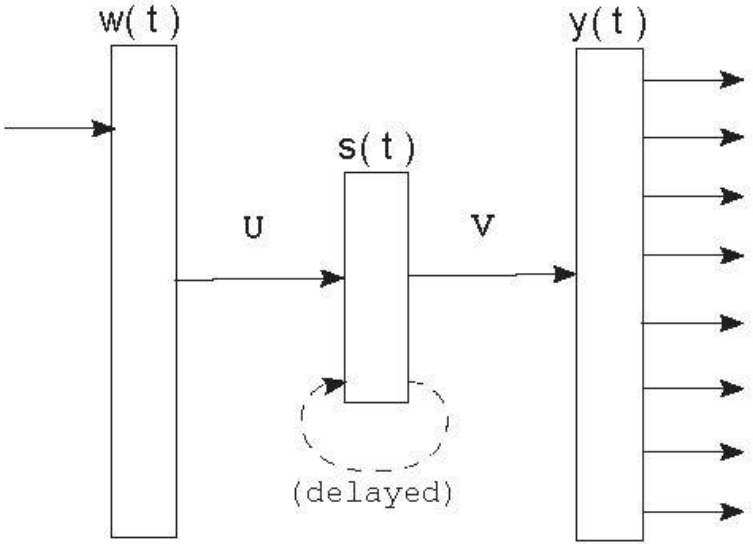


Fig. 1. *Simple recurrent neural network.*

3.4.11 ミコロフ革命 RNNLM

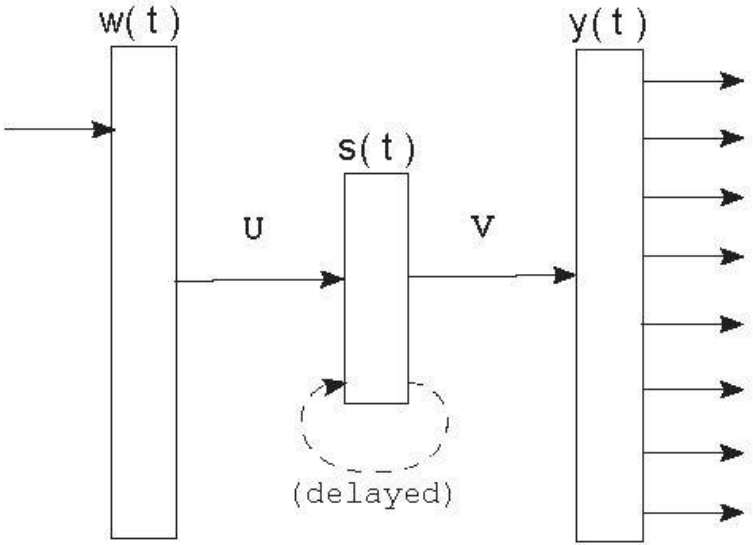


Fig. 1. *Simple recurrent neural network.*

3.4.12 ミコロフ革命 RNNLM

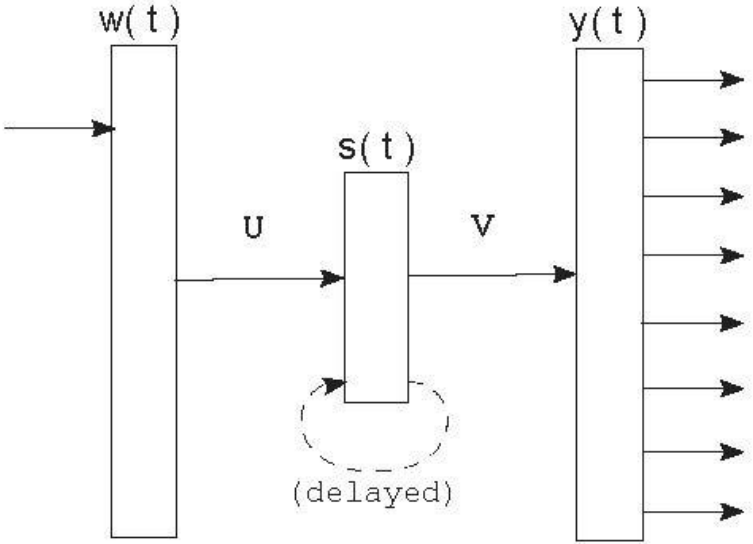


Fig. 1. *Simple recurrent neural network.*

3.4.13 ミコロフ革命 RNNLM

$$\boldsymbol{s}(t)=f\left(\boldsymbol{U}\boldsymbol{w}(t)+\boldsymbol{W}\boldsymbol{s}(t-1)\right)$$

$$\boldsymbol{y}(t)=g\left(\boldsymbol{V}\boldsymbol{s}(t)\right)$$

3.4.14 ミコロフ革命 RNNLM

f (x) はロジスティック関数，g (x) はソフトマックス関数。最近のほとんどのニューラルネットワークと同じく出力層にはソフトマックス関数を用いる。出力を確率分布とみなすように，全ニューロンの出力確率を合わせると1となるように

$$g(x)=\frac{e^{\frac{x_i}{T}}}{\sum_j e^{\frac{x_j}{T}}}$$

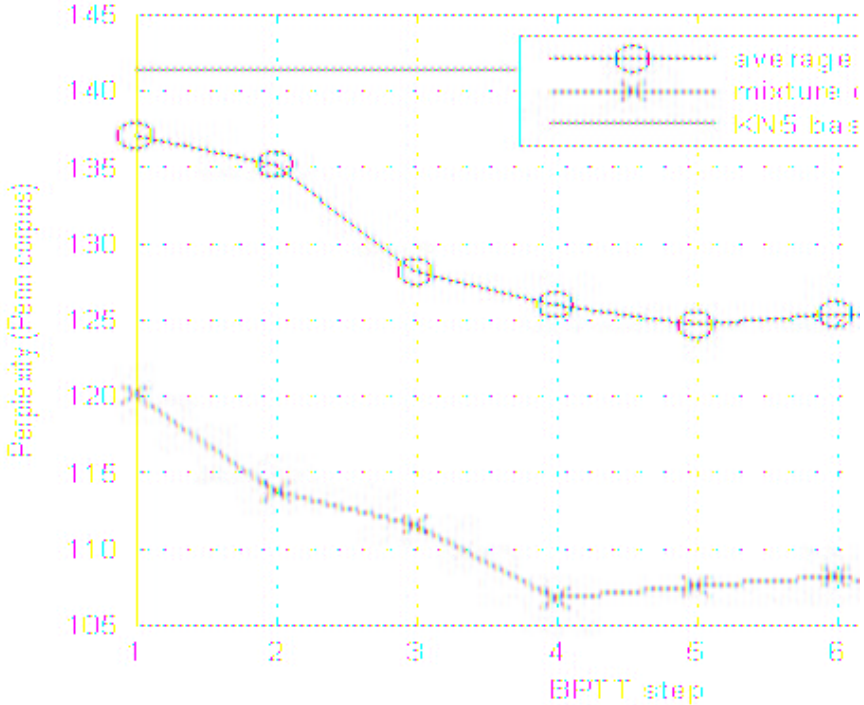
3.4.15 ミコロフ革命 RNNLMの学習

時刻t における入力層から中間層への結合係数行列U は，ベクトル
s (t) の更新を以下のようにする。

$$U(t+1) = U(t) + \alpha \boldsymbol{w}(t) \boldsymbol{e}_h(t)^T$$

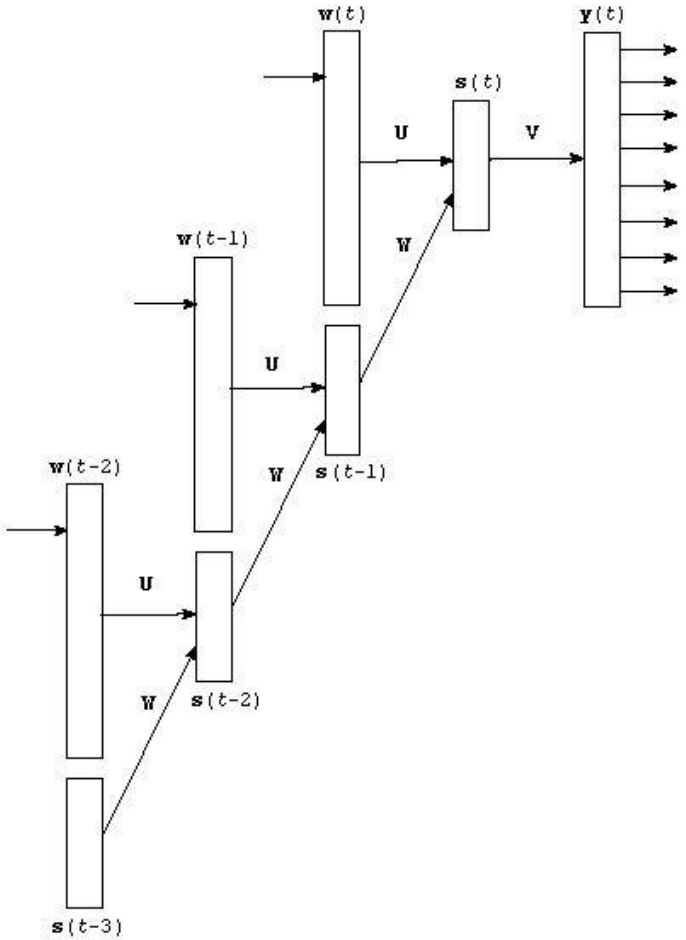
時刻t における入力層ベクトルw(t) は，一つのニューロンを除き全て 0 である。上式のように結合係数を更新するニューロンは入力単語に対応する一つのニューロンのそれを除いて全て0 なので，計算は高速化できる。

3.4.16 ミコロフ革命 BPTT



3.4.17 ミコロフ革命 BPTT(2)

リカレントニューラルネットワークを時間展開して，多層フィードフォワードニューラルネットワークとみなす。3 ステップ分を表示してある



3.4.17 ミコロフ革命 BPTT(3)

バックプロパゲーションスルータイムでは，前の時刻の中間層の状態を保持しておく必要がある。

$$e_h(t-\tau-1)=d_h(e_h(t-\tau)^T W,\ \ t-\tau-1)$$

各タイムステップで，繰り返しで微分して勾配ベクトルの計算が行われる。各タイムステップの時々刻々の刻みを経るごとに急速に勾配が小さくなる勾配消失問題

3.4.17 ミコロフ革命 BPTT(4)

活性化関数がロジスティック関数 f (x) =(1 + exp (-x))^-1 であれば、その微分は f′ (x) = x (1 - x) であった。ハイパータンジェント ϕ (x) =(exp(x) - exp(-x)/(exp(x) + exp(-x))であれば ϕ′ (x) = (1-x^2)であるから、いずれの活性化関数を用いる場合でもニューロンxの値域（取
りうる値）が 0<= x <= 1 である限り、ロジスティック関数であれハイパータンジェント関数であれ、元の値より 0 に近い値となる。これと反対の現象勾配爆発問題が起きる可能性がある。

3.4.18 ミコロフ革命 BPTT(5)

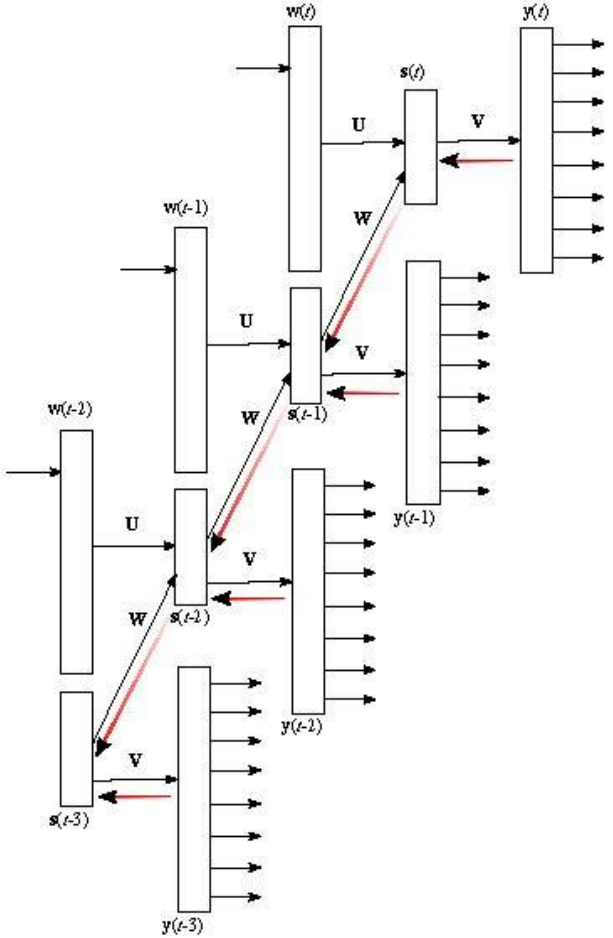
再帰結合係数行列 W の更新には次の式を用いる

$$W(t+1) = W(t) + \alpha \sum_{z=0}^{T} s(t-z-1)\ e_h(t-z)^T$$

行列W の更新は誤差が逆伝播するたびに更新されるのではなく、一度だけ更新する。

3.4.19 ミコロフ革命 BPTT(6)

赤い矢印は誤差勾配がリカレント
ニューラルネットワークの時間展開を
遡っていく様子を示している。



実習ミコロフのコードを読んでみよう

Code:

- Recurrent Neural Network Language Model
  http://www.fit.vutbr.cz/~imikolov/rnnlm/
- Word2vec: https://github.com/dav/word2vec

補足

お伝えし忘れました。ミコロフの rnnlm をちゃんと評価するためには

Srilm-toolkit が必要になります。GitHub のREADME.MD には書いておきましたが口頭でお伝えするのを忘れました。以下にURLを示します。http://www.speech.sri.com/projects/srilm/download.html

利用するには，ID を登録する必要があります。

補足2モデルアンサンブル

1. 質問のあったモデルのアンサンブルについて
2. 同じモデルを，異なる初期化，交差検証データセット，ハイパーパラメータで実行する方が性能が出ます。
3. 検証データセットを変えるとモデルの評価が変わるので他のパラメータが同じでも異なるモデルができあがります。
4. 異なるハイパーパラメータで学習したモデルをアンサンブルするか，ハイパーパラメータの平均値を用いて新たなモデルを訓練するかなど方法が提案されています。

http://cs231n.github.io/neural-networks-3/#ensemble

おわりに

参加してくださった皆様，ありがとうございました。

このプロジェクト RNN camp のプロジェクトページを立ち上げました。

ご意見をお寄せください

メールアドレス：rnncamp.jp@gmail.com

プロジェクトホームページ：http://www.cis.twcu.ac.jp/~asakawa/rnncamp