

SAPT 第9回知能化分科会

一色 浩
(SAPT知能化分科会長)
2018.08.04

SAPT電子会議室C

目 次

1. はじめに...ニューロの特徴
2. 知識を獲得する2つの方法
3. 時系列処理: **RCN, LSTM**とは
4. 瀧 雅人著「これならわかる深層学習」
 9章 再帰型ニューラルネット
5. 深層学習ニューロの応用
6. 学習の手引き
7. Sony Neural Network Console

1. はじめに...ニューロの特徴

人間の大脳

脳は学習(教師あり／教師なし)により賢くなる.

脳はいかなる問題も解ける.

脳は**非線形ネットワーク**である.



ニューラル・ネットワーク(NN)

NNは脳をモデルにしている.

NNも学習により賢くなる.

NNもいかなる論理問題も解ける.

NNも**非線形ネットワーク**である.

非線形ネットワークの性質を理論的に解明するのは難しい. **数値的な結果から推測せざるを得ない.**

2. 知識を獲得する2つの方法

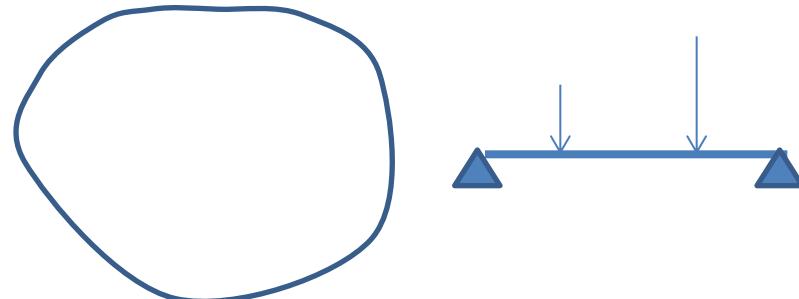
知識…事物の関係を把握…事物間の関数関係



関数

- ・数学的論理的に捉える…**数学関数**
(RBA: Rule Based Approach)
ニュートン力学の大成功：現代科学の原動力
- ・経験的直感的に捉える…**ニューロ**
(DBA: Data Based Approach)
現代科学の行き詰まりに対するブレークスルー

板に荷重が掛かる。最大たわみと応力、位置



RBA: 微分方程式を解く

DBA: NNで推定する
(実験式, 実験図)

数値水槽(風洞)...RBA, 水槽(風洞)実験...DBA

言語理解(習得の仕方が全く違う)

文法重視...受験英語

経験重視...赤ちゃんは自然に言葉を覚える

ラジオの作り方(理解の仕方が全く違う)

ラジオ少年

電子工学の技術者

音声認識、翻訳で起きたこと
最近(2016)、著しく性能が向上
(Google翻訳、スマホ、AIスピーカ、skype)
統計学的方法→深層学習ニューラル・ネットワーク

2018.07.23 NHK-World-Japan

The commander of US Forces Korea has stressed the need to carefully study whether North Korea is sincere about abandoning its nuclear capabilities.

米軍司令官は、北朝鮮が核能力を放棄することについて誠実であるかどうかを慎重に検討する必要性を強調した。

General Vincent Brooks **addressed** a gathering on national security in the **western** US state of Colorado on Saturday.

Vincent Brooks将軍は、土曜日にコロラド州**西部の州**西部で国家安全保障に関する集会に**取り掛かった**。

画像認識能力

RBA...バーコード, 郵便番号, 活字, 指紋

- ・ケースが限られているのものはすべてのケースを覚える.
- ・特徴がよくわかっているもの.
- ・手書き数字の標準データ: 28×28 ピクセル...白黒の場合でも場合の数は $2^{28 \times 28}$ 乗で全数学習は不可能.

DBA...2015年にNNが画像認識で人間を越える.

RBAでは解けない問題:

夫の浮気

人間の好き嫌い

美しい, 美しくない, 心地よい

犯罪の捜査

スポーツの勝敗

経験的 методы тренировки требуют времени

一人前の職人になるのに10年かかる

計算機の進歩が大きい

3. 時系列処理： RNN, LSTMとは

RNN (Recurrent Neural Network) :

時系列を扱えるように、出力の一部を入力に戻す。
短期記憶を保持する。

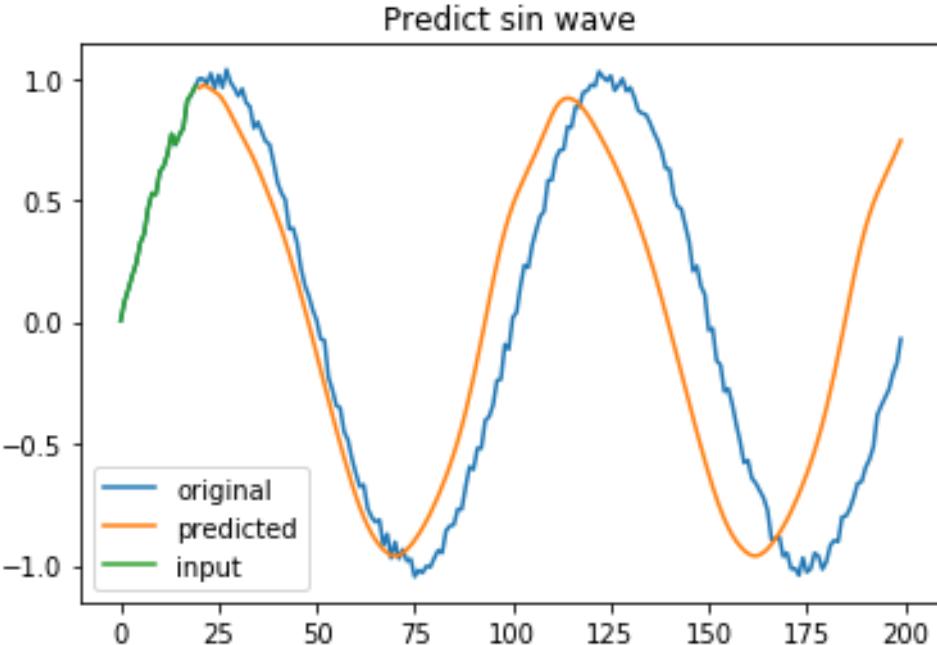
LSTM (Long Short-Time Memory) :

短期記憶だけでなく、長期記憶も保持する。
飛躍的に性能が向上。

入門 Keras (1) Windows に Tensorflow と
Keras をセットアップ

<https://www.infiniteloop.co.jp/blog/2017/11/learning-keras-01/>

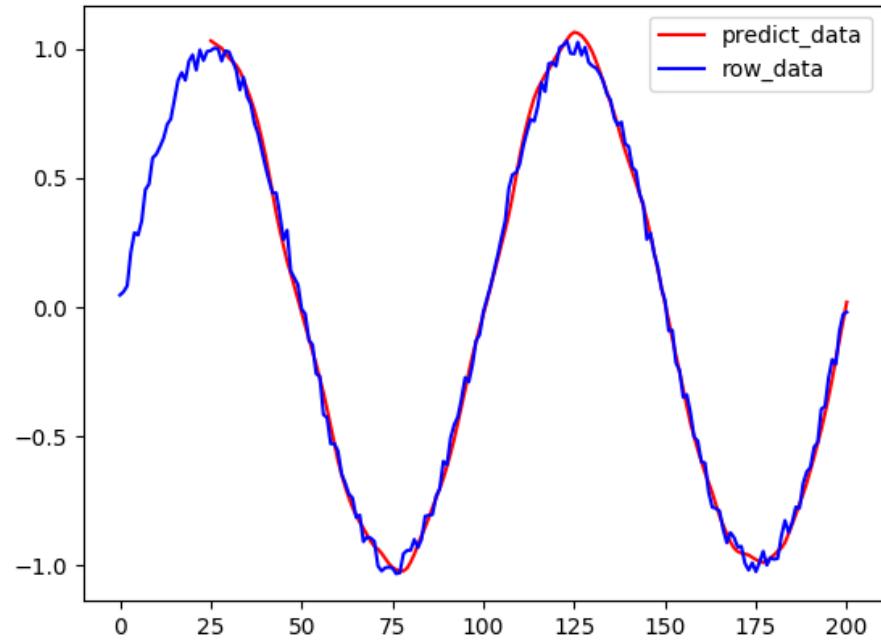
RNN



緑が入力データで、オレンジの線が自分で予測したデータ。予測した値を使ってまた次の予測を行っていますので、次第に誤差が大きくなっていく。

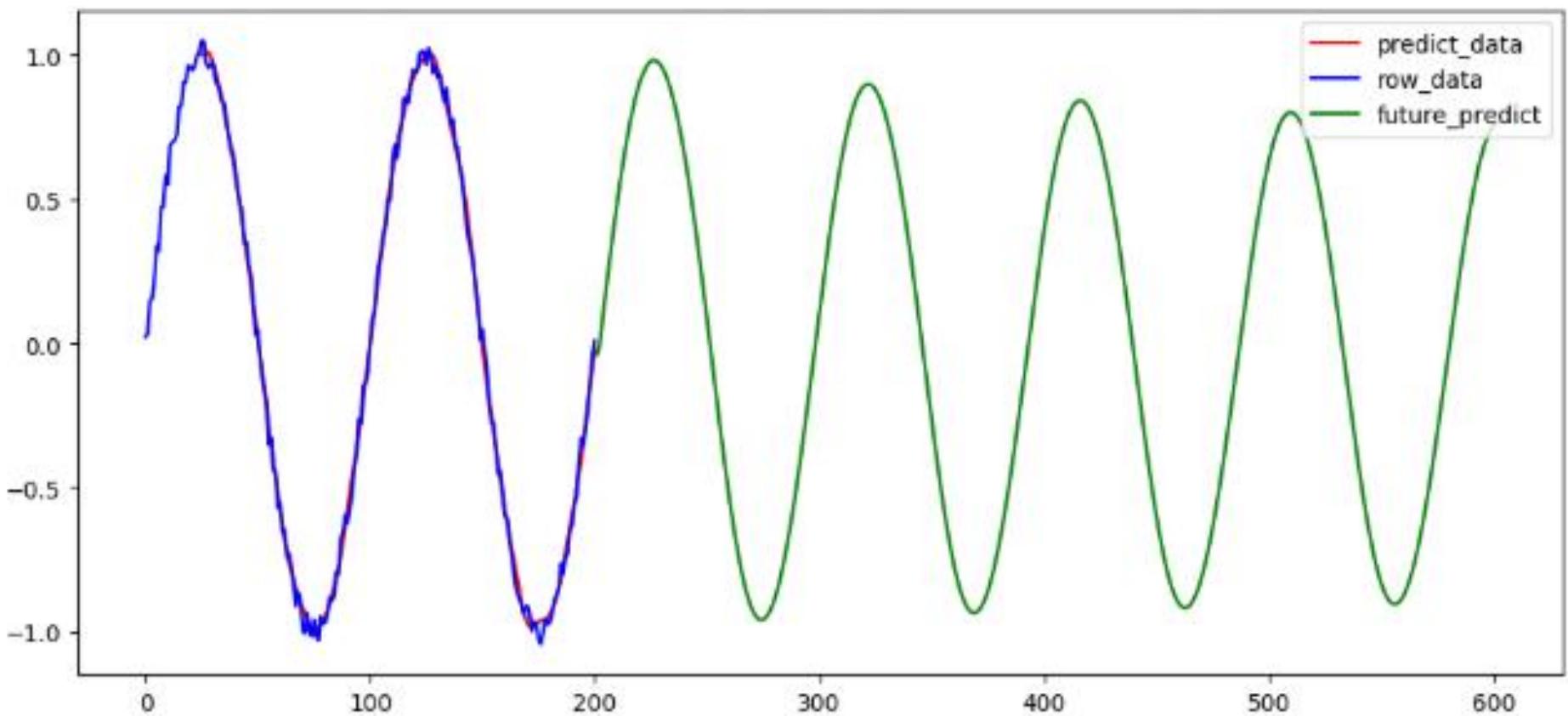
入門 Keras (7) 最終回：リカレントニューラルネットワークを体験する
<https://www.infineloop.co.jp/blog/2018/02/learning-keras-07/>

LSTM



ほとんどノイズの影響を受けずに予測することができます。

KerasでRNN(LSTM)を試してみる
<https://qiita.com/sasayabaku/items/b7872a3b8acc7d6261bf>



未来の予測：青は生データ，赤は予測データ，緑は未来データ

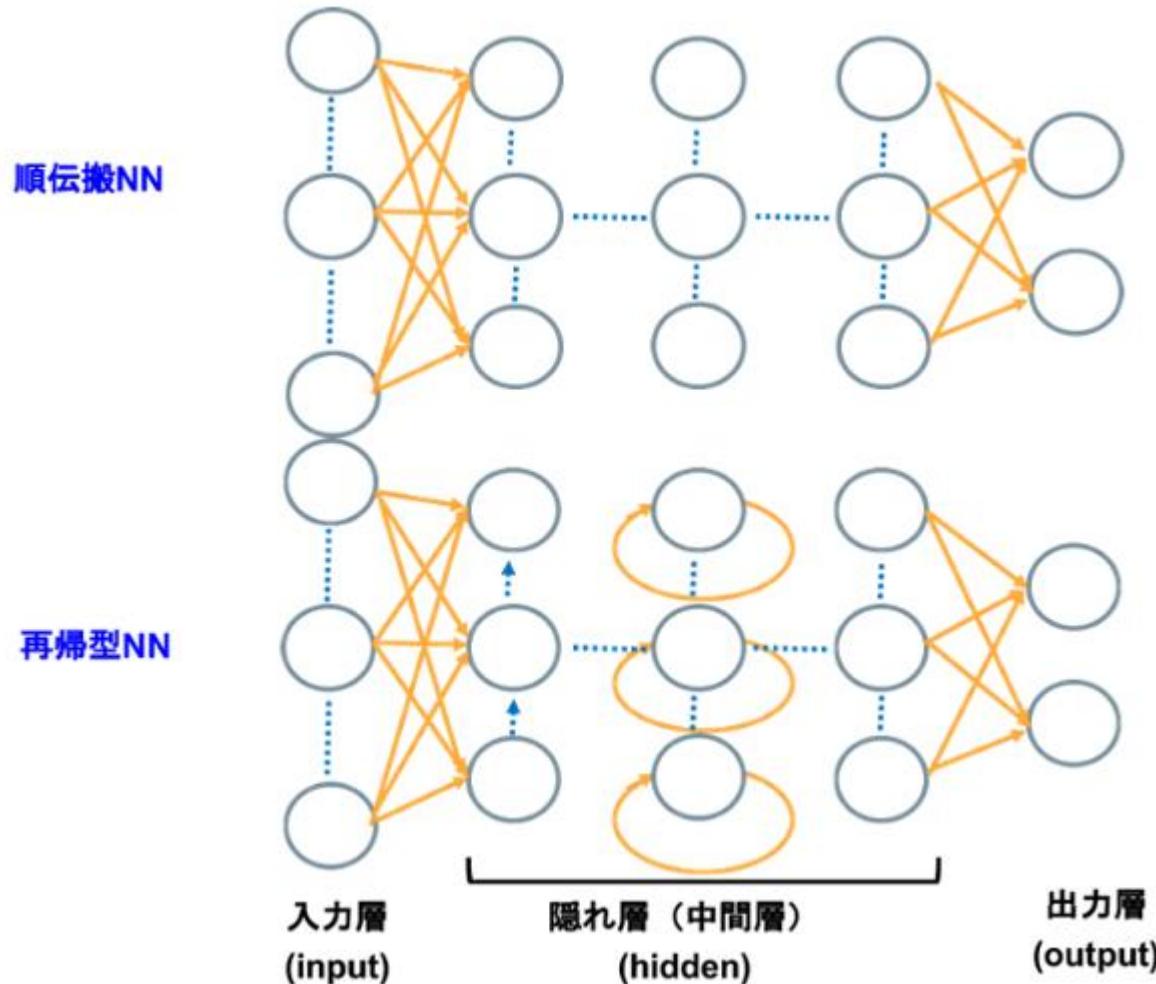
KerasでRNN(LSTM)を試してみる

<https://qiita.com/sasayabaku/items/b7872a3b8acc7d6261bf>

(株)システムインテグレーター梅田社長のブログ

リカレントニューラルネットワーク_RNN(Vol.17)

<https://products.sint.co.jp/aisia/blog/vol1-17>



**順伝播NNと
再帰型NN**

中間層出力は過去の記憶の関数であるので、過去の記憶が再帰により保持される。

問題

麻里ちゃんの好きなケーキはなんですか？

$t - 1$ t $t + 1$

$t - 1$

麻里ちゃんの



t

麻里ちゃんの 好きな

$t + 1$

麻里ちゃんの 好きな ケーキは

ピンポ～ン

解答

麻里ちゃんの 好きな ケーキは なんですか？

チーズケーキ

言語は時系列

$t - 1$, t , $t + 1$ の次の $t + 2$ に何が出てくるかを予測するためには、過去の記憶が必要。

自然言語処理
National Language Processing

自然言語理解
National Language Understanding

機械翻訳
Machine Translation

テキスト分析
Text Analysis

音声認識
Speech to Text

画像分析
Image Analysis

感情分析
Emotion Recognition

文章生成
Deep Writing

音声合成
Text to Speech

動画分析
Video Analysis

パーソナルアシスタント
Personal Assistant

チャットボット
Chat Bot

リカレントニューラルネットワークが使われる技術分野

LSTM ~Long Short-Term Memory~ (Vol.18) <https://products.sint.co.jp/aisia/blog/vol1-18>

前回は、時系列で可変長なデータを処理できるリカレントニューラルネットワーク(Recurrent Neural Network)について説明しました。そこで取り上げたRNNは長いデータを処理すると計算が爆発するため、記憶打ち切り型通時的逆伝搬(Truncated Back propagation Through time)により、短期記憶情報だけで処理していました。でも、やっぱりそう割り切るとAIの精度も限定的になります。そこで、この単純RNN(Simple Recurrent NN)の長期依存性問題を解決する構造を持った長・短期記憶ユニット(Long Short-Term Memory)というモデルが現れ、これが現在のRNNの主流となっています。

過去の記憶を保持するしくみ

相空間

カオス技術で混沌から明るい道へ

<http://www.taos.tokyo/rdinfo/51.html#takens>

力学系における運動は、時間と位置のグラフでは、その運動に関するすべての情報を含まない。初期値に依存しない一般的な運動の表現方法は運動方程式のn個の変数を軸にとった空間(相空間あるいは状態空間)の一点として状態を表現すればその系の振る舞いが軌跡として一目で把握できる。

アトラクター

アトラクタ(attractor)は、非線形力学システムにおいて過渡状態を経た後に定常的に観察される状態である。アトラクタとは、引き付けるという意味である。決定論に従う力学系の構造はd個の状態変数の関数として記述でき、その運動は相空間における軌道(トラジェクトリ)が落ち着く先、すなわちアトラクタとして表現できる。アトラクタの近傍を初期状態とする解は、文字通りアトラクタに引き付けられていく。

ターケンスの埋め込み定理

1個の状態変数からd個の状態変数を復元するターケンスの埋込み定理がある。ターケンスの方法によって再構成されたアトラクタは本来のアトラクタと本質的に微分同相が保証されている。ターケンスの手法により遅れ時間用いて未知の状態変数を復元することを埋め込み(embedding)。遅れ時間のことを埋め込み遅延時間、状態変数の個数のことを埋め込み次元という。

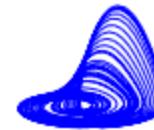
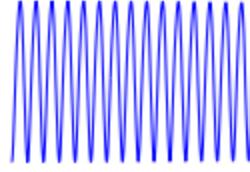
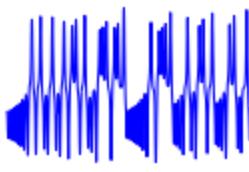
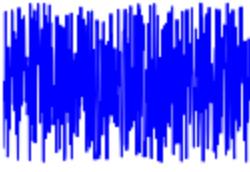
状態変数 $x(t)$ から作られた n 次元データ：

$$(x(t), x(t - \Delta t), \dots, x(t - (n-1)\Delta t))$$

を、 n 次元空間の点としてプロットすることを埋め込みという。

力学系のアトラクタ

<http://www.hisenkei.net/~tohru/Lectures/2003/NLS/PDFs/tsa.pdf>

| | 平衡点 | リミットサイクル | k トーラス | ストレンジアトラクタ | ランダム |
|--------------|---|---|--|---|---|
| 状態空間 |  |  |  |  |  |
| 振舞 | 平衡状態 | 周期 | 準周期 | カオス | ノイジー |
| 構造 | 点 | 閉曲線 \mathbb{R}/\mathbb{Z} | $\mathbb{R}^k/\mathbb{Z}^k$ ($k \geq 2$) | フラクタル | 無構造 |
| 次元 | 0 | 1 | k | 非整数 | 状態空間 d |
| リアプロノフスペクトラム | $\lambda_i < 0$ ($i=1, \dots, d$) | $\lambda_1 = 0$ $\lambda_i < 0$ ($i=2, \dots, d$) | $\lambda_i = 0$ ($i=1, \dots, k$) $\lambda_i < 0$ ($i=k+1, \dots, d$) | $\lambda_i > 0$ ($i=1, \dots, m-1$) $\lambda_m = 0$ $\lambda_i < 0$ ($i=m+1, \dots, d$) | |
| 波形 |  |  |  |  |  |

ターケンスの埋め込み定理

例：単振動

$$\frac{dx}{dt} = v$$

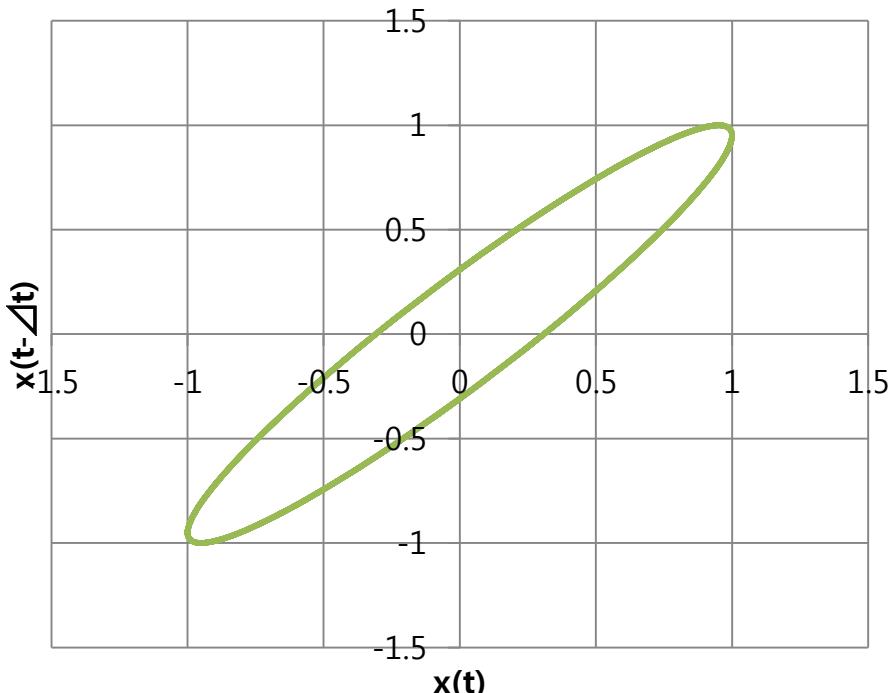
$$\frac{dv}{dt} = -\omega^2 x$$

$$x_i = A \sin(\omega t_i + \phi)$$

$$v_i = A\omega \cos(\omega t_i + \phi)$$

$$t_i = i\Delta t$$

横軸を t 縦軸を x, v として時系列を描くと A と ϕ により図が変わる。



横軸を $x(t)$ 縦軸を $x(t-\Delta t)$ として時系列を描くと、2次元空間の1本の線になる。

A と ϕ を未知数とすると、この曲線は2点が決まると完全に決まる。過去、未来すべての情報がこの曲線にある。

カオス的時系列の短期予測

木戸和彦, 菊池 洋一, ニューロコンピューティングを用いたカオス的データの短期予測, 数理解析研究所講究録 1043巻 1998年 50-58

木戸和彦, カオス的時系列の短期予測に関する研究 - 台風の動きは予測可能か ?, 数理解析研究所講究録 1252巻 2002年 147-153

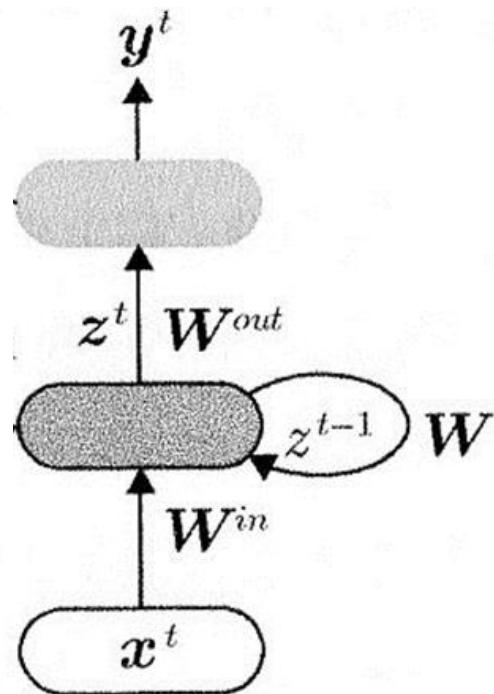
ターケンスの埋め込み定理を用いているが, 残念ながら再帰型NNではない. 台風の動きを予測しているのは興味深い.

足立悠, 「ソニー開発のNeural Network Console入門」の RNN(リカレントNN)およびLSTM(長短期 記憶)の説明

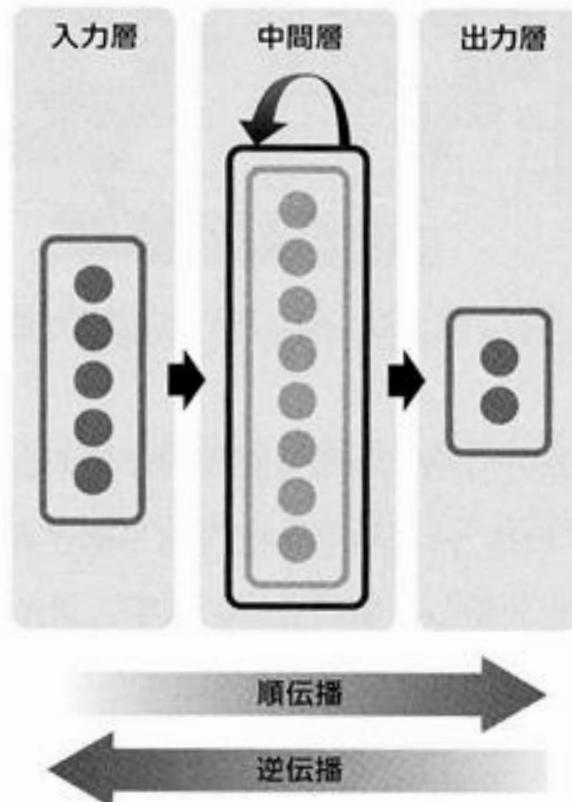
この本の紹介は後出

再帰型ニューラルネットワーク

通常の表記

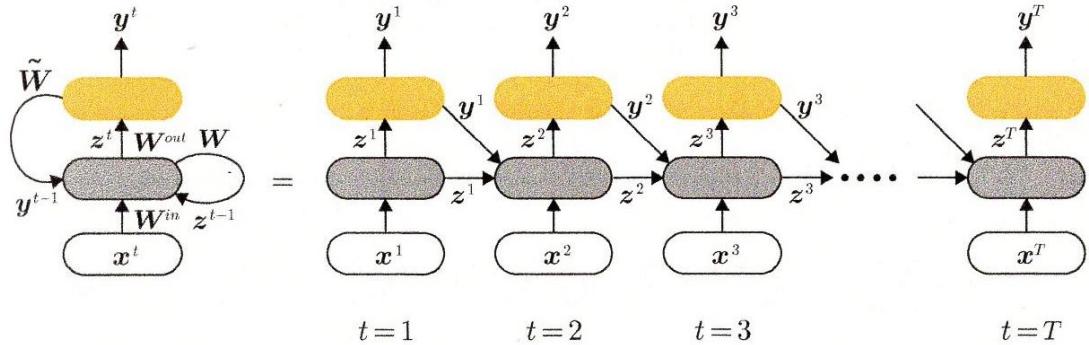


この本の表記



時系列などの系列データを扱う.
中間層は過去の状態を記憶して再利用.

図 2-25 : RNN の仕組み



並べて示すと

時刻 $t-1$ の順伝播を考えると、時刻 $t-1$ の入力層のデータと時刻 $t-2$ の中間層のデータを使って結果を出力。

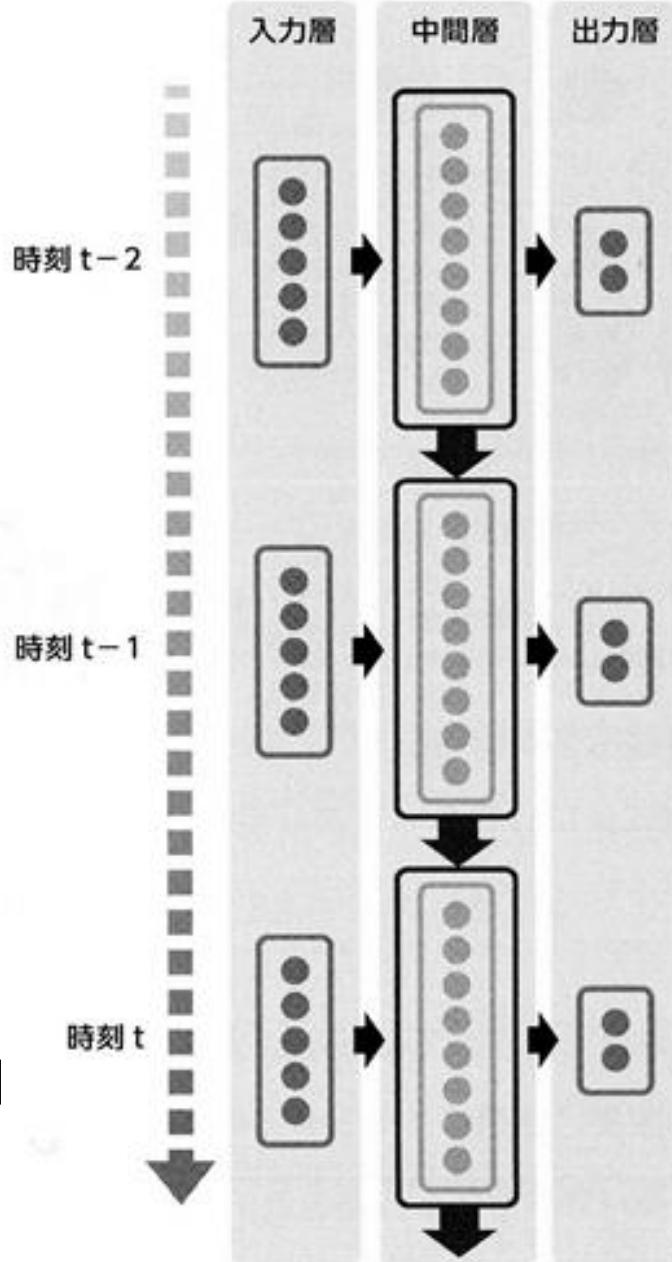


図 2-26 : RNN の順伝播 (時間軸で展開)

時刻 $t-1$ の逆伝播を考えると、中間層には時刻 t の誤差が伝播される。

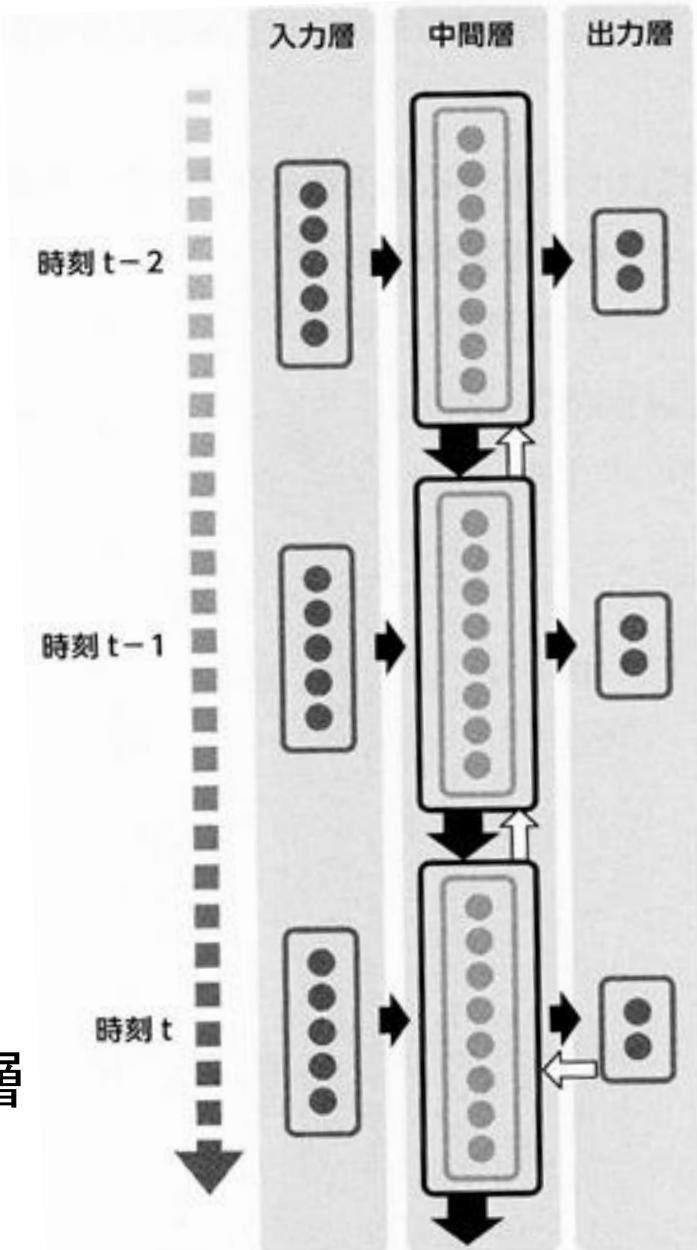


図 2-27 : RNN の逆伝播 (時間軸で展開)

LSTMの仕組み

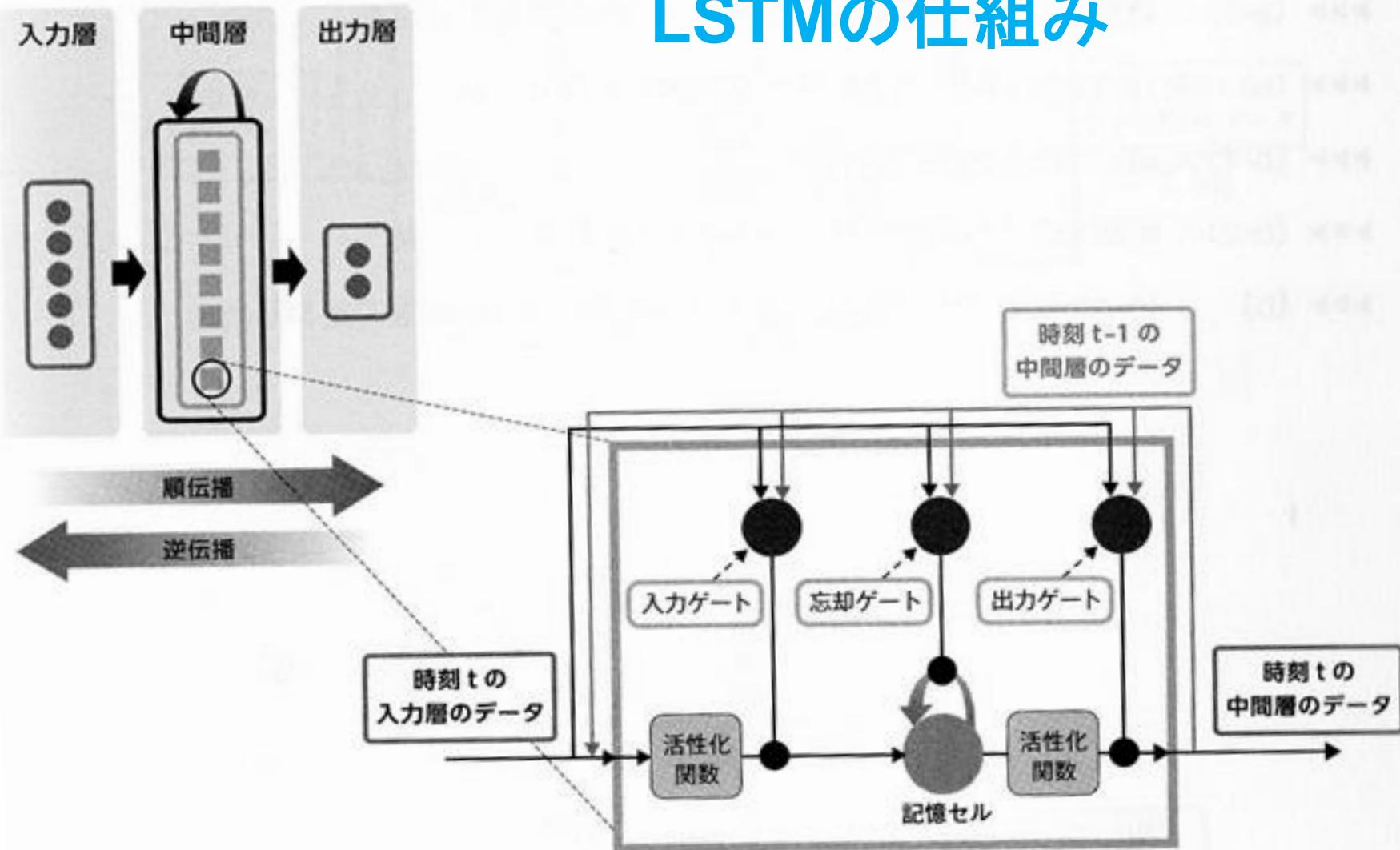


図 2-28 : LSTM の仕組み(全体像)^[4]

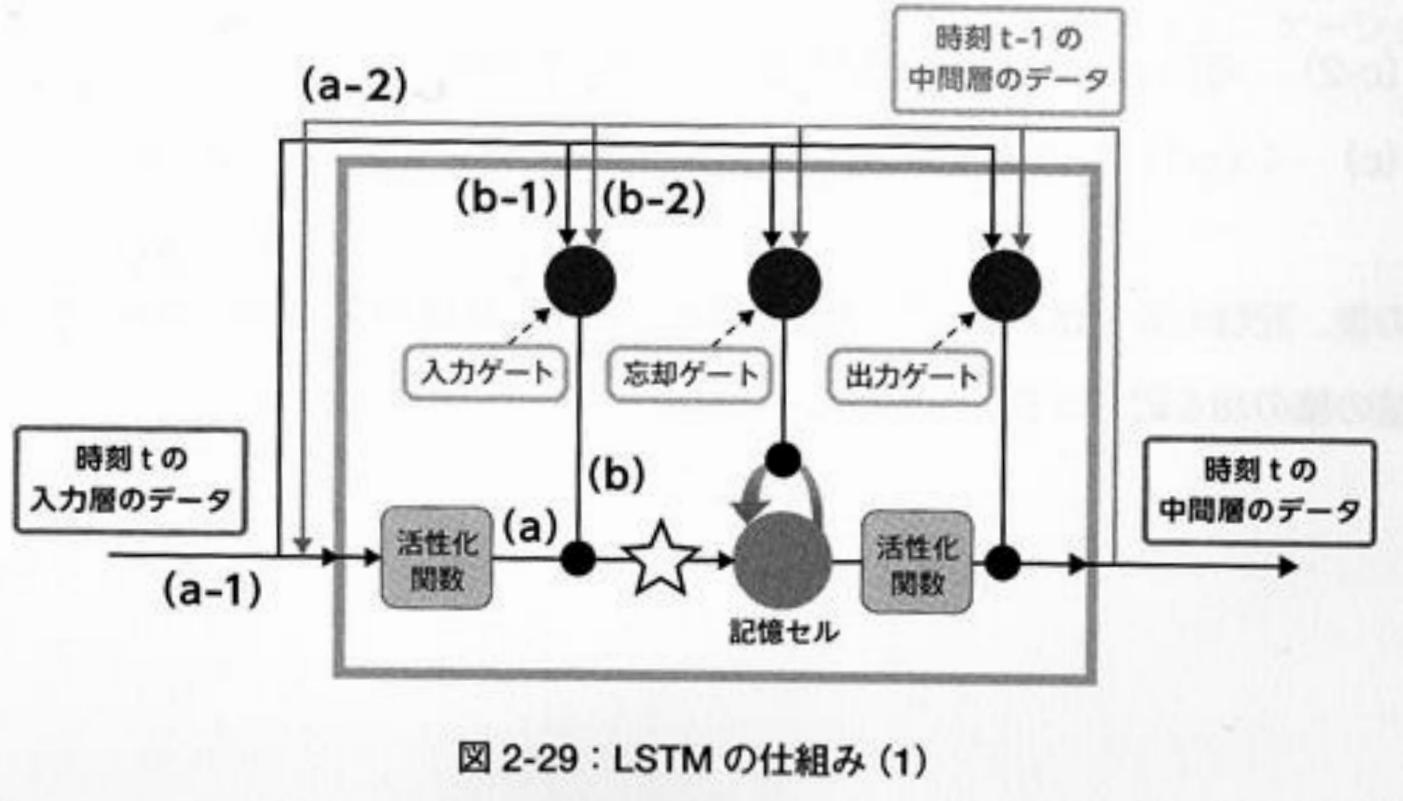


図 2-29 : LSTM の仕組み (1)

- ▶▶▶ (a-1) : 時刻 t の入力層のデータ
- ▶▶▶ (a-2) : 時刻 t の 1 つ前の時刻である ($t-1$) の中間層のデータ
- ▶▶▶ (a) : (a-1) と (a-2) の和を活性化関数で変換した値
- ▶▶▶ (b-1) : 時刻 t の入力層のデータ
- ▶▶▶ (b-2) : 時刻 t の 1 つ前の時刻である $t-1$ の中間層のデータ
- ▶▶▶ (b) : (b-1) と (b-2) の和を入力ゲート内部の活性化関数で変換した値

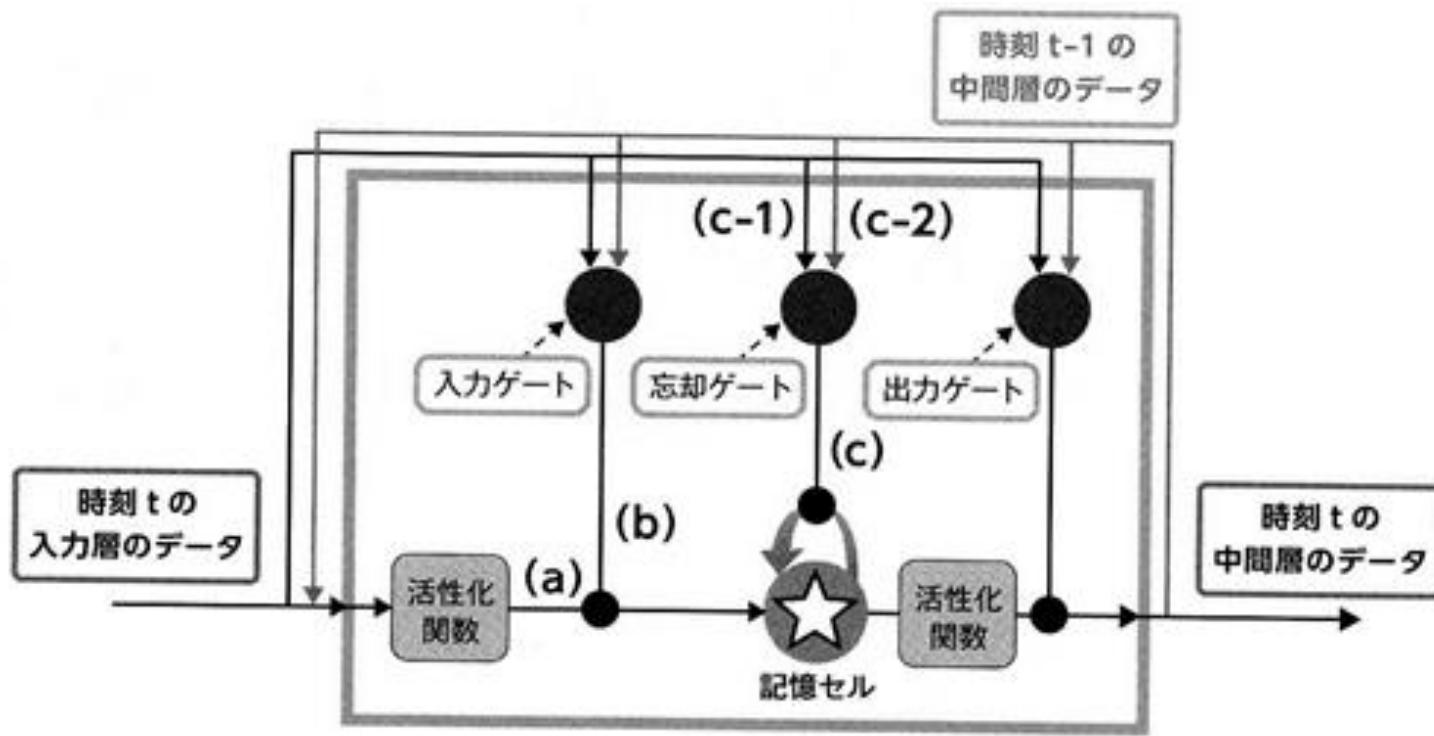


図 2-30 : LSTM の仕組み (2)

- ▶▶▶ (c-1) : 時刻 t の入力層のデータ
- ▶▶▶ (c-2) : 時刻 t の 1 つ前の時刻である $t-1$ の中間層のデータ
- ▶▶▶ (c) : (c-1) と (c-2) の和を忘却ゲート内部の活性化関数で変換した値

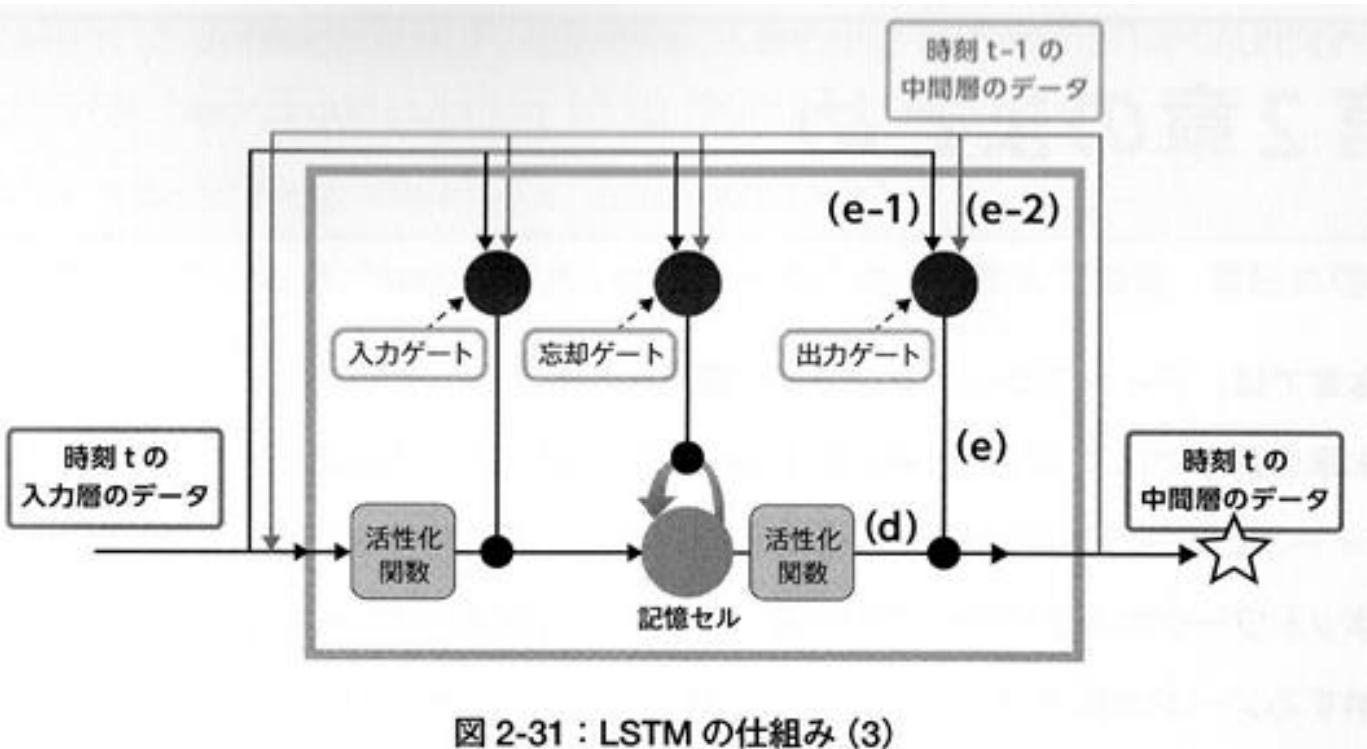


図 2-31 : LSTM の仕組み (3)

- ▶▶▶ (d) : 記憶セルから出力したデータを、活性化関数で変換した値
- ▶▶▶ (e-1) : 時刻 t の入力層のデータ
- ▶▶▶ (e-2) : 時刻 t の 1 つ前の時刻である t-1 の中間層のデータ
- ▶▶▶ (e) : (e-1) と (e-2) の和を出力ゲート内部の活性化関数で変換した値

4.瀧 雅人著「これならわかる深層学習」

9 再帰型ニューラルネット

文章に代表される時系列データを学習する. **自然言語処理**についても議論する.

9.1 時系列データ

長さ T の時系列データを $\mathbf{x}^1, \mathbf{x}^2, \dots, \mathbf{x}^T$ とする.

文章は有限の長さなので、他の場合と違い、いつも全長を対象にするのだと思う。

会話: $\mathbf{x}^1 = \text{"This"}, \mathbf{x}^2 = \text{"is"}, \mathbf{x}^3 = \text{"an"}, \mathbf{x}^4 = \text{"apple"}$

$$\mathbf{x}^t = (0 \ 0 \ \cdots \ 0 \ 1 \ 0 \ \cdots \ 0)^\top$$

↑
aah
applause
apple
applet
zzz

a

図9.1 単語「apple」の1-of-Kベクトル表示

1-of-Kベクトルとは、辞書中の総単語数だけの次元を持ち、対応する単語の成分だけが1で他は0である。
図9.1は”apple”のベクトル。

時系列データは時間方向に強い相関を持つ。“This is an”ときたら、前後のつながり(文脈)から次は何かを求められることが多い。

これを可能にするのが再帰型である。

9.2 再帰型ニューラルネット

9.2.1 ループと再帰

図9.2に一般的なネットワーク、グラフを示す。

このグラフには階層型グラフと大きな違いが2つある。

- (1) 一番左のグラフでは濃い灰色のユニットには一周して戻ってくる再帰回路がある.
- (2) ユニットから出た矢印が自分自身に戻ってくるケースがある.

このような回路には**過去のデータを保持**する機能がある.

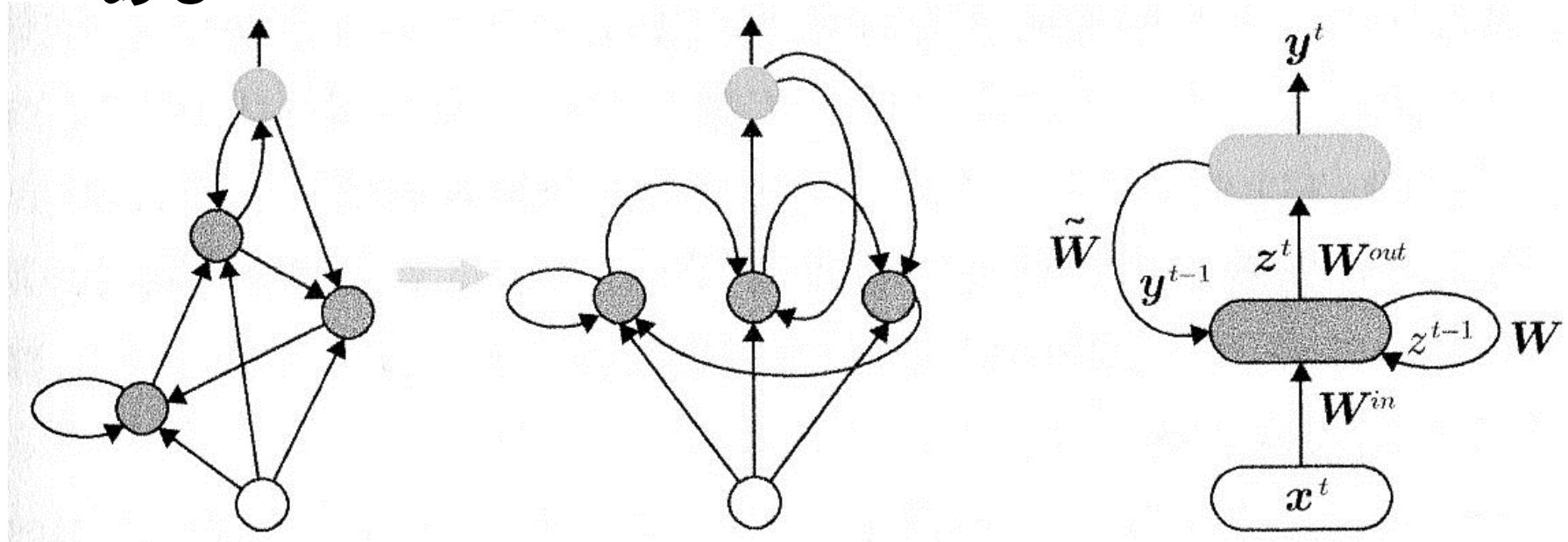


図9.2 ループを持つグラフとしての再帰型ニューラルネット

図9.2右は全体では3層のNNに出力層から中間層. 中間層から中間層に向かう帰還回路を付け加えたものである.これを再帰型NN (recurrent neural network, RNN)と呼ぶ.

RNNの順方向については通常のNNと同様に信号が即時に伝わるが, 帰還ループに沿っては, 1単位時間遅れて伝わる.

式(9.1)は中間層ユニットの, 式(9.2)は出力層の入出力を示す. 入出力はベクトル, ウェイトはマトリクスである.

$$\mathbf{u}^t = \mathbf{W}^{in} \mathbf{x}^t + \mathbf{W} \mathbf{z}^{t-1} + \tilde{\mathbf{W}} \mathbf{y}^{t-1}, \quad \mathbf{z}^t = f(\mathbf{u}^t) \quad (9.1)$$

$$\mathbf{v}^t = \mathbf{W}^{out} \mathbf{z}^t, \quad \mathbf{y}^t = f(\mathbf{v}^t) \quad (9.2)$$

9.2.2 実時間リカレント学習法

RNNの実時間リカレント学習 (real time recurrent Learning, RTRL) を示す。

時刻 t での誤差関数：

$$E^t(\mathbf{w}) = \sum_{n=1}^N E_n^t(\mathbf{x}_n^1, \dots, \mathbf{x}_n^t; \mathbf{w}) \quad (9.3)$$

回帰の場合であれば

$$E_n^t(\mathbf{w}) = \frac{1}{2} \sum_k \left(y_{nk}^t(\mathbf{x}_n^1, \dots, \mathbf{x}_n^t; \mathbf{w}) - t_{nk}^t \right)^2 \quad (9.4)$$

ソフトマックス回帰ならば

$$E_n^t(\mathbf{w}) = \sum_k t_{nk}^t \log y_{nk}^t(\mathbf{x}_n^1, \dots, \mathbf{x}_n^t; \mathbf{w}) \quad (9.5)$$

以下でこの勾配を計算する。

添え字の使い分け:

$i \leftrightarrow$ 入力層, $j \leftrightarrow$ 中間層, $k \leftrightarrow$ 出力層,

$r \leftrightarrow$ 中間層あるいは出力層, $s \leftrightarrow$ すべての層,

すべての層間の重みは w_{kj} のように w で表すので

$$w_{ji} = w_{ji}^{in}, \quad w_{kj} = w_{kj}^{out}, \quad w_{j'j} = w_{j'j} \quad (9.6)$$

簡単のため, $\tilde{\mathbf{W}}$ のないRNNを考える.

$$\frac{\partial E^t(\mathbf{w})}{\partial w_{rs}} = \sum_k \frac{\partial E^t(\mathbf{w})}{\partial y_k^t(\mathbf{w})} \frac{\partial y_k^t(\mathbf{w})}{\partial w_{rs}} \quad (9.7)$$

$\partial y_k^t(\mathbf{w})/\partial w_{rs}$ は

$$p_{rs}^k(t) \equiv \frac{\partial y_k^t(\mathbf{w})}{\partial w_{rs}} = f^{out'}(v_k^t) \frac{\partial}{\partial w_{rs}} \sum_j w_{kj}^{out} z_j^t \quad (9.8)$$

書き直すと

$$p_{rs}^k(t) = f^{out'}(v_k^t) \left(\delta_{r,k} z_s^t + \sum_j w_{kj}^{out} p_{rs}^j(t) \right) \quad (9.9)$$

ここで $p_{rs}^j(t) \equiv \partial z_j^t(\mathbf{w}) / \partial w_{rs}$ で式(9.1)を用いると

$$\begin{aligned} p_{rs}^j(t) &\equiv \frac{\partial z_j^t(\mathbf{w})}{\partial w_{rs}} = f'(u_j^t) \frac{\partial u_j^t}{\partial w_{rs}} \\ &= f'(u_j^t) \frac{\partial}{\partial w_{rs}} \left(\sum_i w_{ji}^{in} x_i^t + \sum_{j'} w_{jj'} z_{j'}^{t-1} \right) \end{aligned} \quad (9.10)$$

であるので、以下の式を得る。

$$p_{rs}^j(t) = f'(u_j^t) \left(\delta_{r,j} x_s^t + \delta_{r,j} z_s^{t-1} + \sum_{j'} w_{jj'} p_{rs}^{j'}(t-1) \right) \quad (9.11)$$

$p_{rs}^j(0) = 0$ から初めて、これらを解く。RTRL法では各時刻で勾配降下法を行う。

$$\Delta w_{rs}^{(t)} = -\eta \sum_k \frac{\partial E^t(\mathbf{w})}{\partial y_k^t(\mathbf{w})} p_{rs}^k(t) \quad (9.12)$$

この手法は、各時間ごとにRNNを学習させる。そのため、過去の情報をあまり保持する必要がなくメモリ消費の少ない手法である。

ただし、 p_{rs}^k という3階のテンソルを計算するため、どうしても計算量が増えてしまう。後述のBPTTはメモリを消費するが、学習速度は速い

9.2.3 ネットワークの展開

RNNにおける情報の流れを見やすくするための展開と呼ばれる方法がある。

図9.3では横方向に時間を展開している.

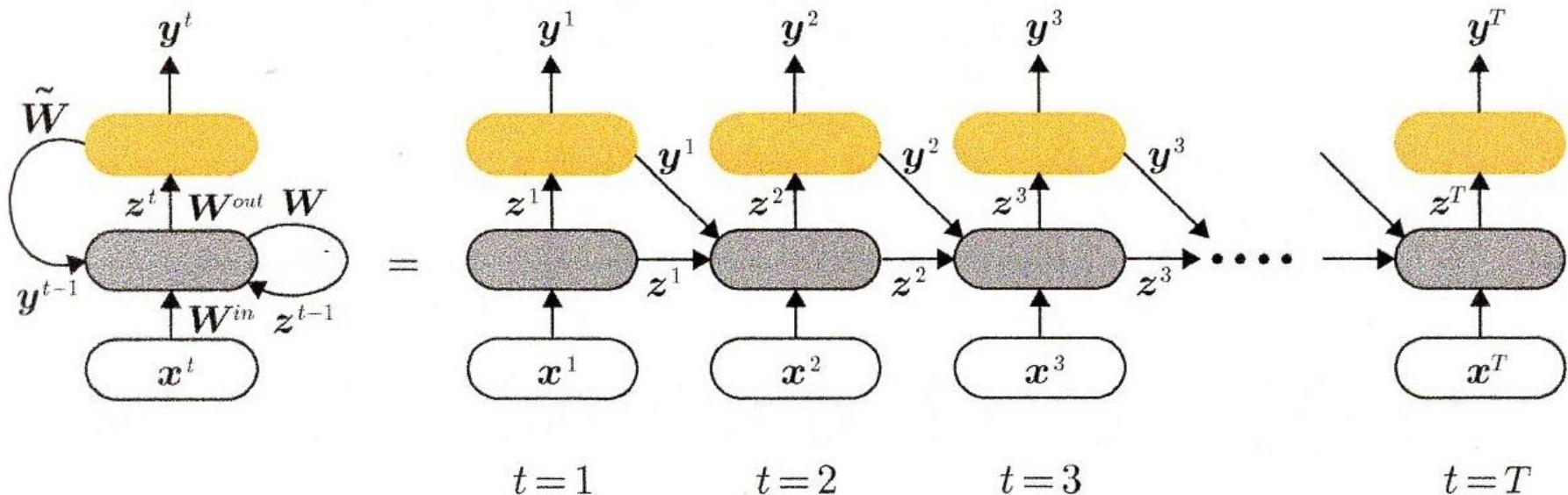


図9.3 RNNの実時間方向への展開

9.2.4 通時的誤差逆伝搬法

RNNを展開すると、順伝播型のときと同じように誤差逆伝播法が定式化できる。これが**通時的誤差逆伝播**(back propagation through time, BPTT)である。

BPTTでは時刻全体にわたる誤差を考える:

$$E(\mathbf{w}) = \sum_{n=1}^N \sum_{t=1}^T E_n^t(\mathbf{w}) \quad (9.13)$$

勾配を考えるために、中間と出力ユニットのデルタ:

$$\delta_j^t = \frac{\partial E}{\partial u_j^t}, \quad \delta_k^{out,t} = \frac{\partial E}{\partial v_k^t} \quad (9.14)$$

を定義し、チェインルールを用いる。

$$\delta_j^t = \sum_k \frac{\partial v_k^t}{\partial u_j^t} \frac{\partial E}{\partial v_k^t} + \sum_{j'} \frac{\partial u_{j'}^{t+1}}{\partial u_j^t} \frac{\partial E}{\partial u_{j'}^{t+1}} \quad (9.15)$$

$$\delta_k^{out,t} = \sum_n \frac{\partial E_n^t}{\partial v_k^t} + \sum_{j'} \frac{\partial u_{j'}^{t+1}}{\partial v_k^t} \frac{\partial E}{\partial u_{j'}^{t+1}} \quad (9.16)$$

書き直すと

$$\delta_j^t = \left(\sum_k w_{kj}^{out} \delta_k^{out,t} + \sum_{j'} w_{j'j} \delta_{j'}^{t+1} \right) f'(u_j^t) \quad (9.17)$$

$$\begin{aligned} \delta_k^{out,t} &= \sum_n \left(y_k^t(\mathbf{x}_n^{1,\dots,t}) - t_{nk}^t \right) f^{out'}(v_k^t) \\ &\quad + \sum_{j'} \tilde{w}_{jk} \delta_j^{t+1} f^{out'}(v_k^t) \end{aligned} \quad \text{要check} \quad (9.18)$$

この逆伝播は $\delta_j^T = 0$ として解ける.

次に勾配を求める. 展開されたRNNは重み共有されたNNとみなせるので, 例えば

$$\frac{\partial E}{\partial w_{ji}^{in}} = \sum_{t=1}^T \frac{\partial E}{\partial u_j^t} \frac{\partial^+ u_j^t}{\partial w_{ji}^{in}} \quad (9.19)$$

ここで, $\partial^+ u_j^t / \partial w_{ji}^{in}$ はある時刻 t の瞬間だけを考えて, 順伝播しか起こらないとした際の微分係数である. ループを回る再帰的な伝播は考えない.

$$\text{結局 } \frac{\partial E}{\partial w_{ji}^{in}} = \sum_{t=1}^T \delta_j^t x_i^t, \quad \frac{\partial E}{\partial w_{kj}^{out}} = \sum_{t=1}^T \delta_k^{out,t} z_j^t \quad (9.20)$$

$$\frac{\partial E}{\partial w_{j'j}} = \sum_{t=1}^T \delta_{j'}^t z_j^{t-1}, \quad \frac{\partial E}{\partial \tilde{w}_{jk}} = \sum_{t=1}^T \delta_j^t v_k^{t-1} \quad (9.21)$$

この方法は, 正確には**エポックごとのBPTT法**と呼ばれる. 通常単にエポック法と呼ぶ場合は現時刻での誤差関数のみを考慮して以下のように考える.

$$E(\mathbf{w}) = \sum_{n=1}^N E_n^T(\mathbf{w}) \quad (9.19)$$

9.3 機械翻訳への応用

RNNにはさまざまな応用があるが、現在熱心に研究されているものは**自然言語への応用**である。

まず入力文を $\mathbf{x}^1, \mathbf{x}^2, \dots, \mathbf{x}^T$ のように **T個の単語** に分けて、RNNに入力する。

RNNの出力層は、辞書中の単語数だけのユニットがあるソフトマックス層とする。

出力最大のユニットから出力単語 y^t が予測される。
 y^1, y^2, \dots, y^T が翻訳文になる。

9.4 RNNの問題点

RNNは中間層が1層のNNに帰還路が加わった、**一見浅いNN**のように見える。実際には信号がループされるので深いNNである。

深いNNは学習において問題を引き起こす.

ループを回るたびに同じ重み W が何度もかかるために、容易に勾配が消失したり爆発する。

RNNの問題に対処するための研究は現在でも活発。

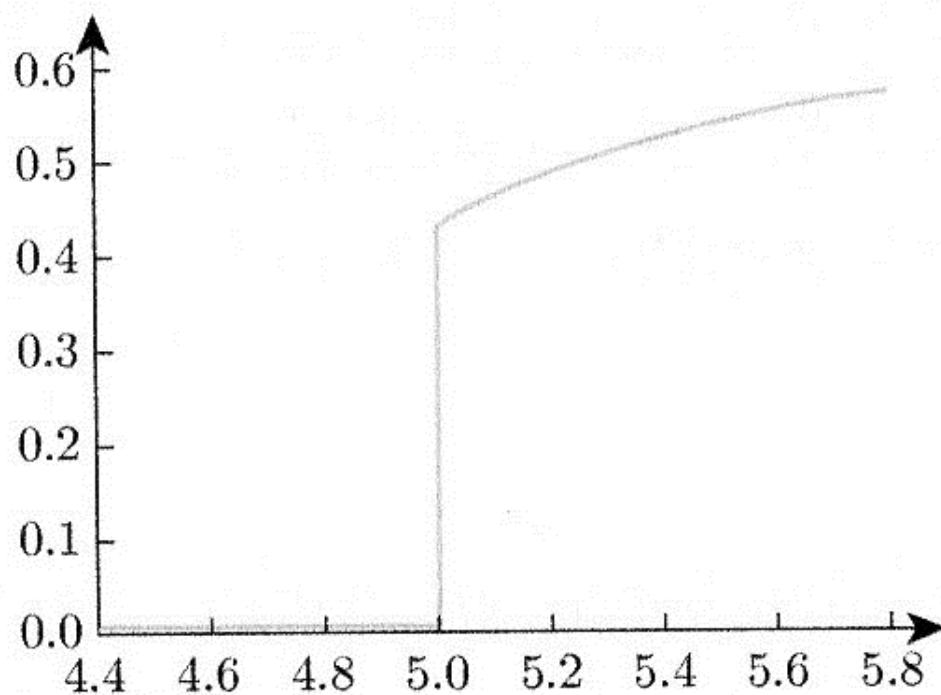


図9.4 横軸は w 、縦軸はRNNの模型 $z^t = \sigma(wz^{t-1} + b)$ 。 (ただし $b=-2.5$, $z^1=0.5$) の時刻 $t=50$ における二乗誤差 $E^{50}=(y^{50}-0.2)^2$ 。

9.5 長・短期記憶

RNNの目的は長期間に渡る時系列情報の蓄積.

長時間になると勾配消失を起こすので長時間蓄積が不能になる.

この状況の改善するために長・短期記憶 (long short-time memory, LSTM) が生まれた.

不要になった過去の記憶は捨てる方がよい.

この目的のために忘却ゲートを設ける.

LSTMの理論的詳細は本講義のレベルを越える.

以下の諸節を省略する

9.5.1 メモリー・セル

9.5.2 ゲート

9.5.3 LSTM

9.5.4 LSTMの順伝搬

9.5.5 LSTMの逆伝搬

9.5.6 ゲート付き再帰的ユニット

9.6 再帰的ニューラルネットと自然言語処理

9.6.1 Seq2Seq学習

9.6.2 ニューラル会話モデル

5. 深層学習ニューロンの応用

浅里さんの宿題

高精度三次元地図がマーケットに出始めているので、これと組み合わせることが考えられないか？

知能化と測位技術分科会の活動を今回の講義終了をもって止めるか？それとも**浅里さんの宿題**に応えることを目標とする活動を継続するか？そういう形で結果を出せるか？

毛利さんに頂いたヒントをもとに

A. GPSの性能向上:

- (1) NNでカルマンフィルタを作る. 近未来予測性能の向上ができないか?
- (2) 自己符号化器を使って
 - (2-1) 受信信号のノイズ除去
 - (2-2) 受信信号の異常検出して多重反射などの出ている信号を測位に使わない.
 - (2-3) ジャミングの検出.
 - (2-4) 受信衛星の選択
 - (2-5) 受信アンテナの指向性をAIでコントロール

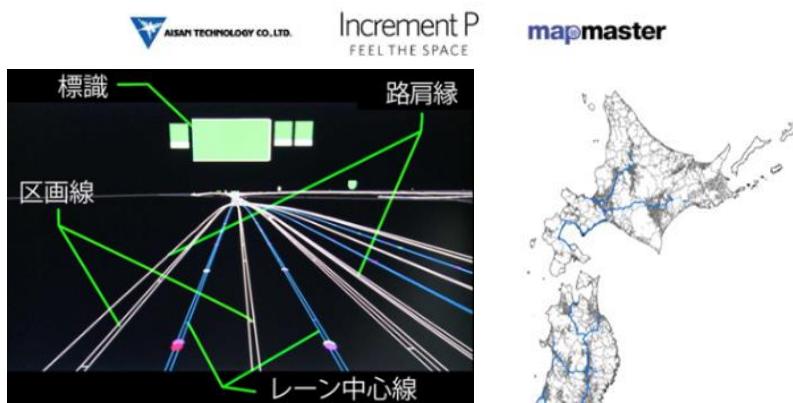
B. GPSのアプリケーション：

ほとんどすべての車がGPSとカメラを備えている。
外国人のAIを駆使した迷子対策. 特に救急医療.
Yes/Noで答えを出せるようにする.
目標の方向に矢印を示す.

画像認識による測位(浅里さんのご提供)

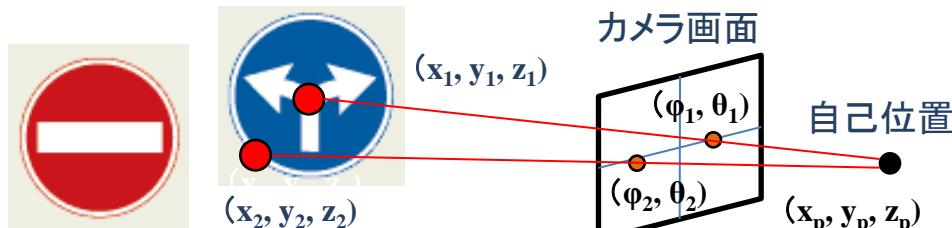
自動車業界から普及しつつある高精度三次元地図と、カメラによる画像認識で測位が可能

DYNAMIC MAP PLATFORM



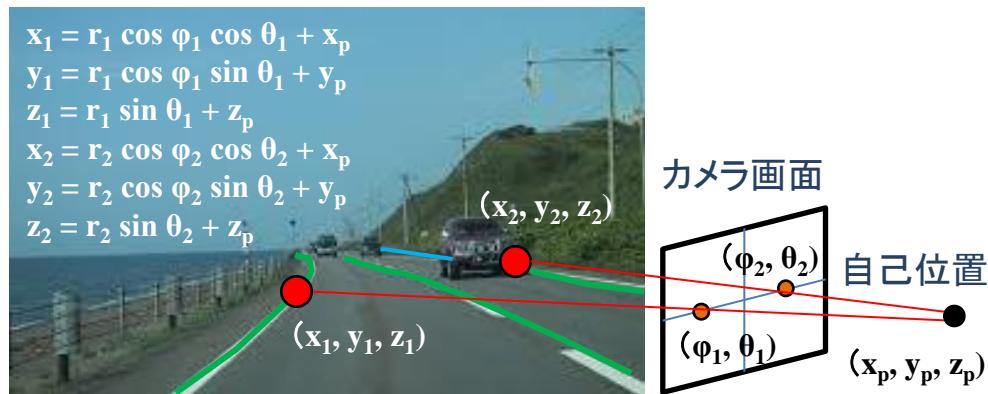
● 標識等を認識してマップと照合し位置を検知

$$\begin{aligned}x_1 &= r_1 \cos \varphi_1 \cos \theta_1 + x_p \\y_1 &= r_1 \cos \varphi_1 \sin \theta_1 + y_p \\z_1 &= r_1 \sin \theta_1 + z_p\end{aligned}\quad \begin{aligned}x_2 &= r_2 \cos \varphi_2 \cos \theta_2 + x_p \\y_2 &= r_2 \cos \varphi_2 \sin \theta_2 + y_p \\z_2 &= r_2 \sin \theta_2 + z_p\end{aligned}$$



● 白線等を認識してマップと照合し位置を検知

$$\begin{aligned}x_1 &= r_1 \cos \varphi_1 \cos \theta_1 + x_p \\y_1 &= r_1 \cos \varphi_1 \sin \theta_1 + y_p \\z_1 &= r_1 \sin \theta_1 + z_p \\x_2 &= r_2 \cos \varphi_2 \cos \theta_2 + x_p \\y_2 &= r_2 \cos \varphi_2 \sin \theta_2 + y_p \\z_2 &= r_2 \sin \theta_2 + z_p\end{aligned}$$



講演者からのコメント：

ビルの谷間で使えないでしょうか？

地下街での測位に使えないでしょうか？

シームレス測位の実現

6. 学習の手引き

東京女子大学の浅川伸一先生から頂いた情報

Summary of ImageNet

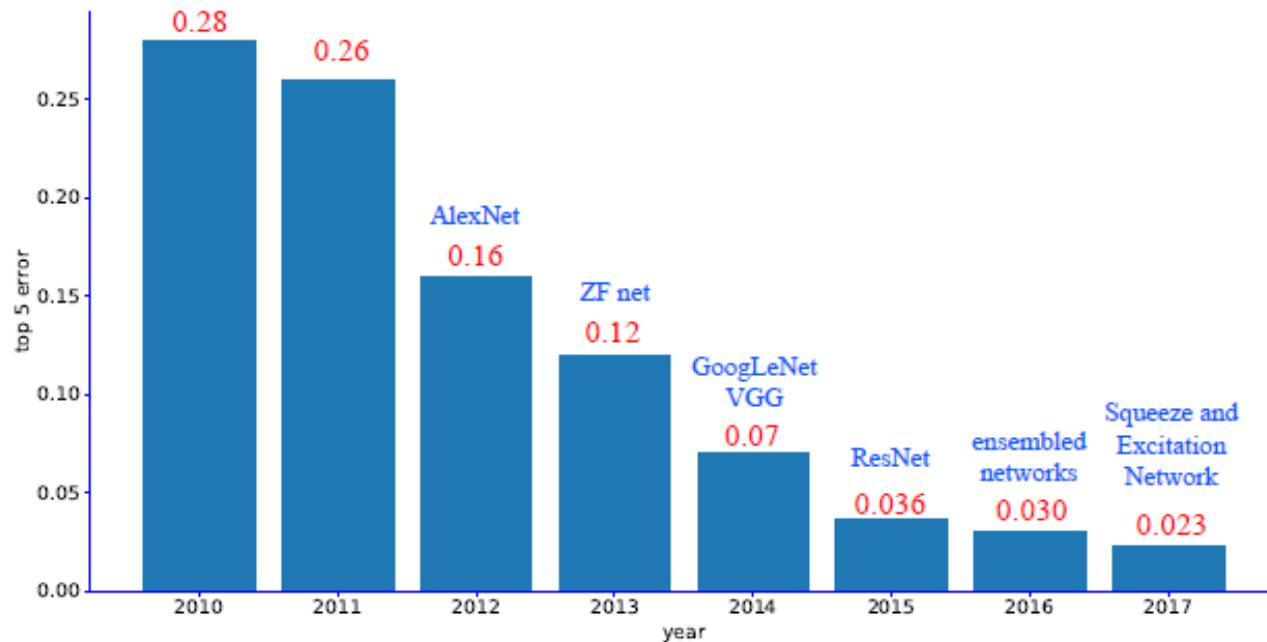


Figure: イメージネットの優勝チームの成績年次変化

フレームワーク毎の GitHub 星数

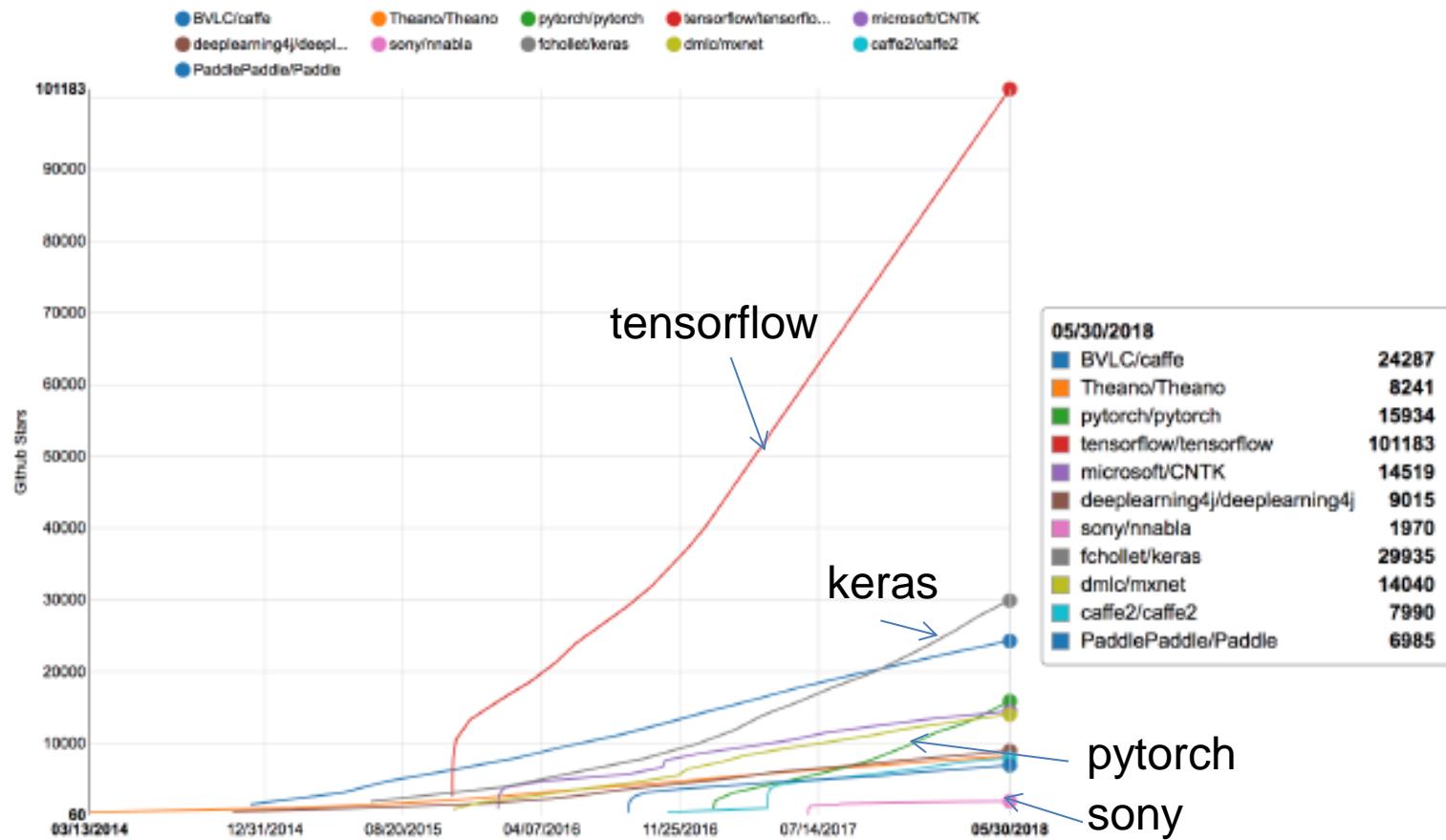


Figure: フレームワーク毎の GitHub 星数の推移¹
<http://jorditorres.org/deep-learning-frameworks/>

フレームワーク毎の GitHub 星数 (2)

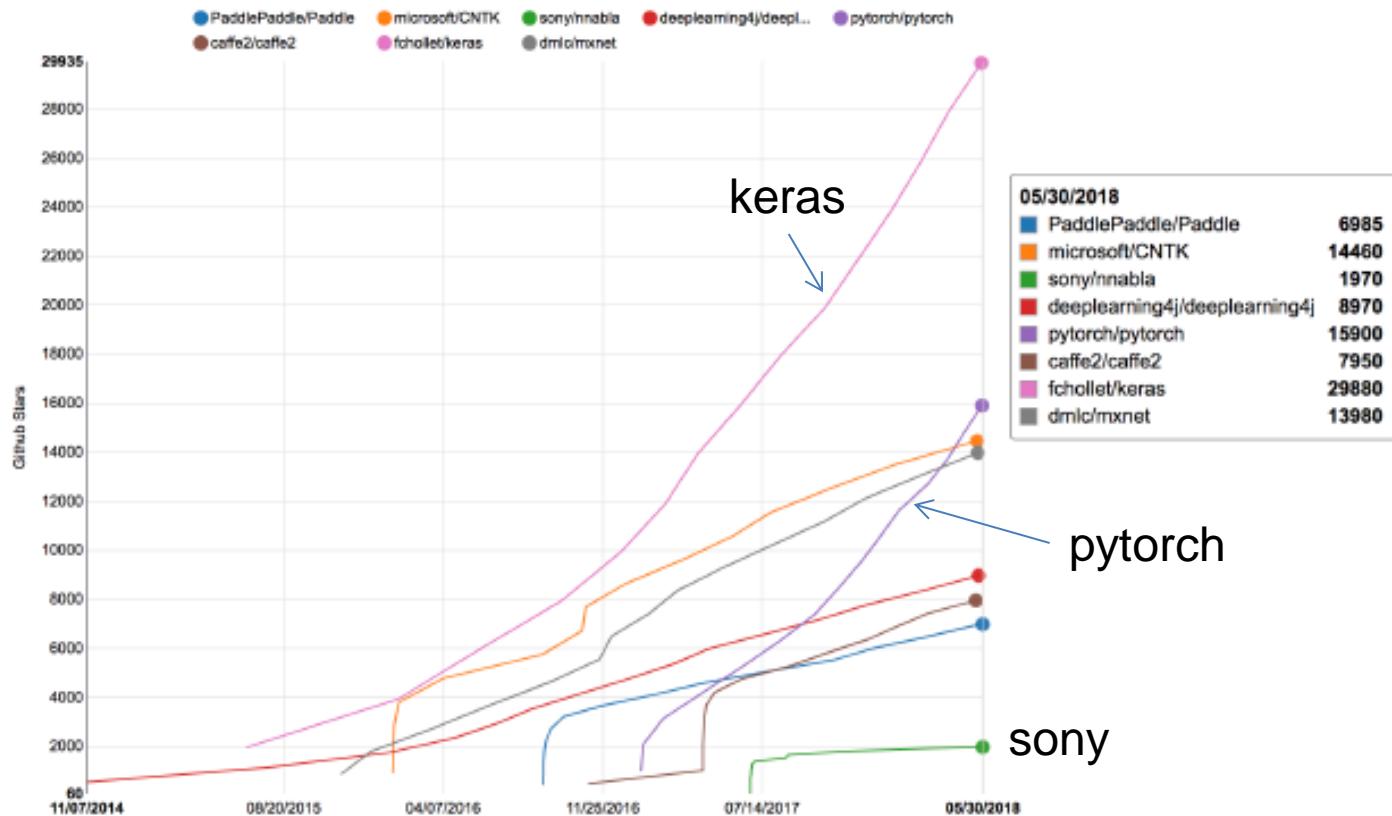


Figure: フレームワーク毎の GitHub 星数の推移 without Tensorflow and Theano

浅川伸一先生に頂いたコメント：

- ・継続学習の機会についてのご質問ですが、英語が苦にならない方であれば MOOC のコンテンツを利用されるのが良いかと存じます。たとえば fast.ai のMOOC コースは無料ですが良質です。

<http://www.fast.ai/>

- ・日本はこの分野で周回遅れ
- ・手前味噌になりますが資格取得を目指すことにはれば、目標が設定されるので学習意欲を高く保つことができます。日本ディープラーニング協会の G 検定、E 資格を目指して勉強するという選択肢もあります。

<http://www.jdla.org/>

<http://www.fast.ai/> の感想：

- ・ビデオレクチャで、7レッスンからなる。
- ・レッスン1を聞いてみました。英語の発音もクリアでとてもよい教材と思います。

fast.aiの内容

Our courses (all are free and have no ads):

Deep Learning Part 1: [Practical Deep Learning for Coders](#)

Deep Learning Part 2: [Cutting Edge Deep Learning for Coders](#)

Computational Linear Algebra: [Online textbook](#) and [Videos](#)

Deep Learning Part 1: [Practical Deep Learning for Coders](#)

のLesson 1は研究開発者は必見だと思います。このビデオレクチャの最初の部分にGPUをクラウドでレンタルする方法が解説されています。自分のPCを端末にしてクラウド上のGPUを使う方法で、**使用時間で課金**されます。安いものから高いものまであります。

Deep Learning Part 1の内容

1. Image Recognition
RECOGNIZING CATS AND DOGS
2. CNNs
CONVOLUTIONAL NEURAL NETWORKS
3. Overfitting
IMPROVING YOUR IMAGE CLASSIFIER
4. Embedding
STRUCTURED, TIME SERIES, & LANGUAGE MODELS
5. NLP
COLLABORATIVE FILTERING; INSIDE THE TRAINING LOOP
6. RNNs
INTERPRETING EMBEDDINGS; RNNs FROM SCRATCH
7. CNN Architecture
RESNETS FROM SCRATCH

日本人向きの独学用教材としては

アイシア梅田社長のブログ

https://products.sint.co.jp/aisia/blog/tag/ai_blog_basics

株式会社システムインテグレータの梅田社長のブログ.

数式は皆無といってよいが、深層学習ニューロの原理と応用の全体像を的確に解説している。非常にユニークで的確。小説を読む感じで読める。短時間で概要を知りたい人には最適と思われる。

MathWorksのビデオ・Webセミナー

MATLABによるディープ・ラーニング

<https://jp.mathworks.com/videos/deep-learning-with-matlab-119998.html>

MathWorks社の製品を導入することを前提にしているが、よくまとまっていて、非常に分かり易い。

Michael Nielsen

ニューラルネットワークと深層学習

オンライン書籍(日本語訳)

https://nnadl-ja.github.io/nnadl_site_ja/ 第1-4章

<http://yusuke-ujitoko.hatenablog.com/entry/2017/02/18/235231> 第5-6章

日本語に読み易く翻訳されている。丁寧に書かれているので、
時間を掛けられる人が、腰を据えて取り組むのに向いている。

瀧正人

これならわかる深層学習

講談社, 2017

深層ニューラルネットワークの理論を包括的に、分かり易く説明している。深く理解したい人には最適と思うが、応用についての解説は十分ではないと思われる。この本を読んだからと言ってすぐにコーディングできたり、すぐにライブラリを活用できるとは思えないが、研究開発者の必読書であることに間違いない。

計算機アルゴリズム研究会

C言語による実用ニューロ・コンピューティング

ラッセル社, 1992

講演者が同僚と共に著で書いたものである。C言語によるコーディングもあるので、階層型ニューロの学習の基礎である誤差逆伝搬学習を確実に理解するための一助となると思われる。

磯蘭水，藤永奈保子，鈴木悠(翻訳)(Jason R. Briggs)

楽しいプログラミング Pythonではじめよう！

(Snake Wrangling For Kids Learning to Program
with Python)

オーム社

(https://www.utrgv.edu/cstem/_files/documents/snake-wrangling.pdf)

深層学習NN学習の言語としてはPythonが必須である。このテキストは親が子供にプログラミングを教える際のテキストとして書かれているので、大変分かり易い。

英語が苦手でない人の独学用教材としては

Francois Chollet

DEEP LEARNING with Python

Manning Publications, 2018

深層学習ニューロの使い方の解説書である。Keras(MITのライセンス)をframework(基盤、ライブラリ)としおり、使用言語はPythonである。**難しい式は皆無で、アプリケーション作成方法が詳しく解説されている。**すぐに深層学習ニューロを応用したい人には最適といってよからう。深層学習ニューロのほぼ全般にわたり解説しているが、Convolution NN(画像)とRecurrent NN(言語、時系列)が中心で AutoEncoderには触れていない。

MOOC (Massive Open Online Courses, 大規模公開オンライン講座)

例えば, [fast.ai](#) のDeep Learning Part 1: Practical Deep Learning for Codersがある. ビデオレクチャで, 7レッスンからなる. 英語の発音もクリアでとてもよい教材と思う. Pytorchをframework (基盤, ライブラリ)としている. 実際にPCを動かしながら, さらに補助教材を読みながら聞かないと理解できないと思われる.

7. Sony Neural Network Console

今年の6月から本格稼働したが、最近web上の記事が急増している。

Neural Network Console - Sony

<https://dl.sony.com/ja/>

2018/06/18 - Neural Network Consoleはニューラルネットワークを直感的に設計でき、学習・評価を快適に実現するディープラーニング・ツール。グラフィカルユーザーインターフェイスによる直感的な操作で、ディープラーニングをはじめましょう。[Docs](#) · [Cloud](#) · [Community](#) · [Neural Network Console](#)

すごいぞ！Neural Network Console！ - クresco

<https://www.cresco.co.jp> › ホーム › AI/Watson

2017/11/22 - これをニューラルネットワークコンソールで表現すると、こうなります。レイヤーの繋がりかたも直感的にわかりやすいです … 気になったところ。なんだか褒めすぎてSONYさんの回し者だと思われそうなので、気になったところもあげておきます。

ソニー Neural Network Console と エクセルで始める ニューラルネットワーク

<https://www.ecomott.co.jp> › ホーム › ネタ

2018/03/28 - エコモット社内では、昨年から技術者向けに何度もニューラルネットワークや分析の勉強会をしています。現実的には、技術者より非技術者(管理とか営業とか企画とかとか、プログラマーやSEではないという意味)の方が何らかの分析すべき ...

「簡単にディープラーニングができる」と噂の、SONY「Neural Network ...

ai-kenkyujo.com/2017/12/05/neural-network-console/

2017/12/05 - 今年夏頃にサービスが開始された、「誰でも簡単にディープラーニングができる」と噂の、SONY「Neural Network Console」を ... 層の数や活性化関数や誤差関数を選ぶだけで、とっても簡単にニューラルネットワークが組めてしまいました！

20171206 Sony Neural Network Console 活用テクニック - SlideShare

https://www.slideshare.net/Sony_Neural.../20171206-sony-neuralsnetworkconsole

2018/01/16 - ソニーのNeural Network Console大勉強会ソニーネットワークコミュニケーションズ株式会社 / ソニー株式会社シニア ... 用意したデータセットで、設計したニューラルネットワークを学習する基本的にはこの3ステップで画像認識機を作成すること ...

深層学習 - ソニーネットワークコミュニケーションズ

<https://www.sonymetwork.co.jp> › 2017年

2017/08/17 - プログラムエンジニアやデザイナーは、本格的なGUIを持つディープラーニング統合開発環境であるコンソールソフトウェアを用いることで、直感的なユーザーインターフェースで、ニューラルネットワークの設計、学習、評価などを効率的に行い ...

ソニー開発のNeural Network Console入門 一数式なし、コーディングなし ...

数式を一切使わず、1全結合型・2畳み込み型・3再帰型のニューラルネットワークの手法を図で紹介します。そして、ソニーが開発した注目のディープラーニングツール「Neural Network Console」を使用。ドラッグ&ドロップ操作でニューラルネットワークを構築し、…

ソニーのニューラルネットワークコンソールのクラウド版を試しに使ってみる ...

arakans-pgm-ai.hatenablog.com › ニューラルネットコンソール

2017/11/09 - ニューラルネットワークコンソール(Neural Network Console)に、インストールしなくても使えるクラウド版ができたみたいですね。早速、試してみます。

【無料化速報レビュー】SONYのディープラーニングクラウドサービスを触った感想

<https://www.codexa.net> › MLaaS

2017/11/13 - 無料枠を利用してSony Neural Network Console(ソニー・ニューラル・ネットワーク・コンソール)のクラウドβ版を使って見ました！サンプルプロジェクトが提供されていましたので、そちらをひとまず触ってみることにします。

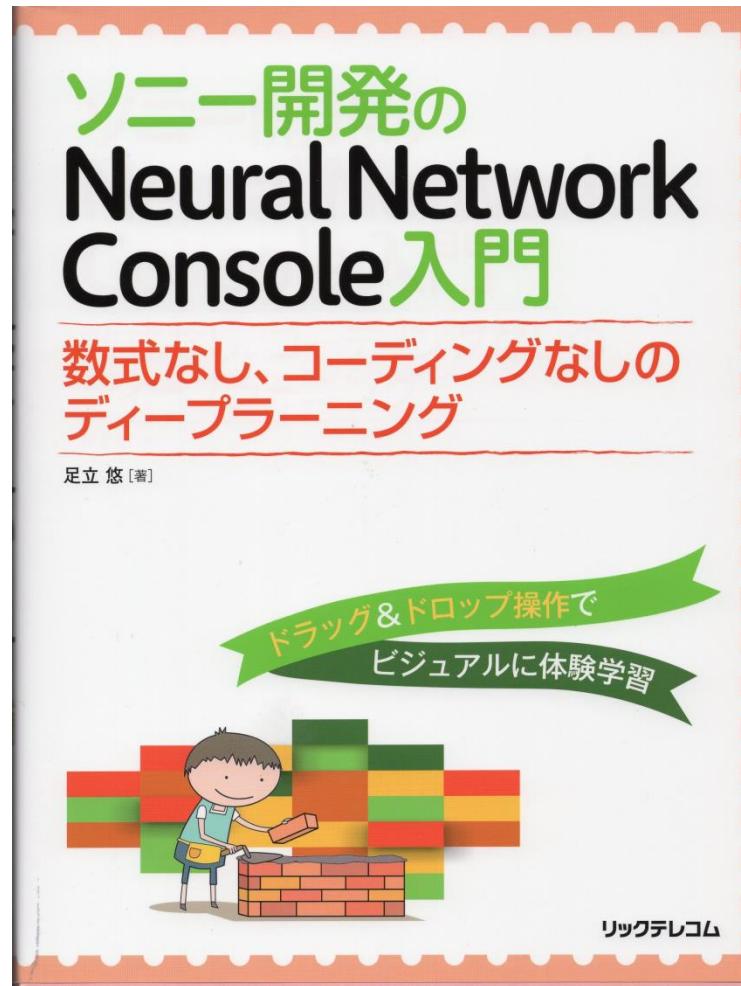
SONY Neural Network Consoleで日経平均株価をディープラーニング ...

cedro3.com › AI(人工智能)

2018/03/10 - このCalcを使うと、SONY Neural Network Consoleで読める数値データファイルが作成できます。しかも、なんと無料なん.... 60回くらいトライさせ、評価をチェックした結果、最も良いニューラルネットワークの構成は、こんな形になりました。

解説本

足立悠, 「ソニー開発のNeural Network Console入門」, リックテレコム (2018) ¥2,400+税



CONTENTS

| | |
|---|----|
| はじめに..... | 3 |
| Chapter 1 / 第1章 AI の世界へようこそ | |
| 1.1 AI とデータサイエンス..... | 8 |
| 1.2 機械学習..... | 15 |
| 1.3 ニューラルネットワークからディープラーニングへ..... | 24 |
| 第1章のまとめ..... | 27 |
| Chapter 2 / 第2章 ディープラーニングの手法 | |
| 2.1 ニューラルネットワーク..... | 30 |
| 2.2 ディープラーニング..... | 38 |
| 2.3 置き込みニューラルネットワーク..... | 42 |
| 2.4 再帰型ニューラルネットワーク..... | 46 |
| 第2章のまとめ..... | 52 |
| Chapter 3 / 第3章 AI ツールと Neural Network Console | |
| 3.1 世界で普及する AI ツール..... | 56 |
| 3.2 Neural Network Console..... | 59 |
| 3.3 NNC のインストール..... | 63 |
| 3.4 NNC の操作画面..... | 74 |
| 第3章のまとめ..... | 78 |

● Chapter 4 / 第4章 初級：サンプルプロジェクトを実行してみよう！

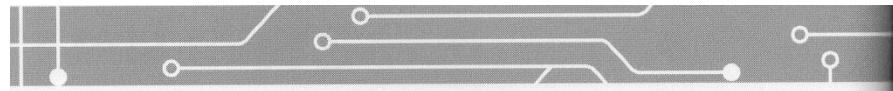
| | |
|--------------------------------|-----|
| 4.1 ニューラルネットワークによる画像分類(1)..... | 80 |
| 4.2 CNN による画像分類(1)..... | 100 |
| 第4章のまとめ..... | 118 |

● Chapter 5 / 第5章 中級：新規プロジェクトを実行してみよう！

| | |
|--------------------------------|-----|
| 5.1 ニューラルネットワークによる画像分類(2)..... | 120 |
| 5.2 CNN による画像分類(2)..... | 140 |
| 5.3 ネットワーク構造の最適化..... | 152 |
| 第5章のまとめ..... | 156 |

● Chapter 6 / 第6章 上級：オリジナル画像で実装してみよう！

| | |
|--------------------|-----|
| 6.1 データセットの作成..... | 158 |
| 6.2 ネットワークの作成..... | 166 |
| 6.3 データセットの選択..... | 173 |
| 6.4 学習条件の設定..... | 176 |
| 6.5 学習の実行..... | 179 |
| 6.6 評価の実行..... | 181 |
| 第6章のまとめ..... | 184 |



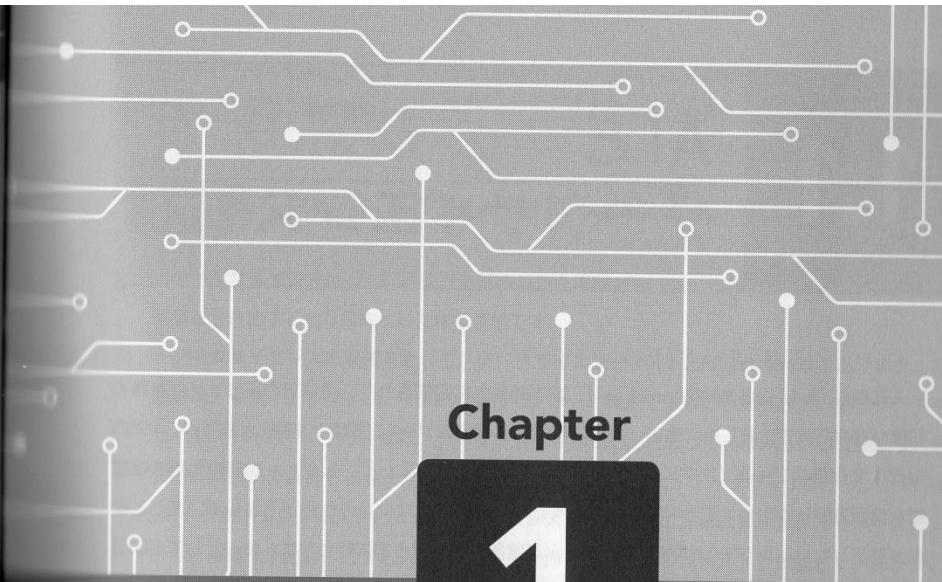
● Chapter 7_{第7章} 上級：オリジナルデータで実装してみよう！

| | |
|--------------------|-----|
| 7.1 データの前処理..... | 186 |
| 7.2 データセットの登録..... | 232 |
| 7.3 ネットワークの作成..... | 234 |
| 7.4 データセットの選択..... | 237 |
| 7.5 学習条件の設定..... | 239 |
| 7.6 学習の実行..... | 241 |
| 7.7 評価の実行..... | 243 |
| 第7章のまとめ..... | 246 |

● Appendix_{付録}

| | |
|--|-----|
| A.1 NNC 非対応 OS の PC に NNC をインストール..... | 248 |
| A.2 機械学習を使って分類問題を解いてみよう！..... | 268 |

| | |
|-----------------|-----|
| おわりに..... | 280 |
| 参考文献・お勧め書籍..... | 281 |
| 索引..... | 283 |



Chapter

1

AI の世界へようこそ



ここ数年、AIは新聞の一面記事やニュースの話題などで取り上げられ、急速に注目を集めています。そしてすでに、AIを搭載したロボットや家電、チャットボットなどが登場し、社会に広く普及しつつあります。本章ではまず、皆さんのがこれからAIを活用していくために、AIとそれを取り巻く技術を中心に、必要な知識を順に学んでいきましょう。