

深層学習で一変した ニューロと人工知能の衝撃

一色 浩

測位技術振興会 知能化と測位技術分科会長

((有)数理解析研究所 代表)

2018.07.27

第9回 測位技術懇話会

目 次

1. はじめに...日本の現状
2. 知識を獲得する 2 つの方法
3. ニューロの歴史
4. 学習とは
5. 深層学習とは何か？
6. 深層学習ニューロの応用
7. 終わりに...学習の手引き

1. はじめに...日本の現状

政府が如何に経済政策を実施しても、**日本経済が発展しない**. 国債が増えるばかり。何故？

日本はモノづくりに成功して、**太平の夢**を貪った.
江戸時代

気がついたら、AIを始めとする**ITで周回遅れ**
幕末...黒船

産業構成の見直し(単純なモノづくりは主役でない)
明治維新

ハイリスク・ハイリターン時代に、なぜ情報が重要？
情報がビジネスチャンスを見つけ、リスクを評価.

原子力
電子計算機
アポロ計画
GPS
インターネット
携帯電話
AI(人工知能)
...

...

革新的な科学技術
日本から生まれない

革新技術～荒唐無稽な夢 ➡ 夢を育てる社会

日本人の国民性
モノづくり(農業...縦社会)に向いているが, IT(狩猟
...横社会)などに向いていない.

日本社会の閉鎖性

大学はかなり改善, 理研などはひどい

小保方事件の温床

研究上のことで直接コンタクトできない

コンタクトに必要な個人のメールアドレスなどは非公開
まず広報が出てくる

夢やコミュニケーションで生まれる発想の芽を潰す
スピードにブレーキを掛ける

日本の社会がもっとオープンで風通しの良い社会にならないと行けない. そうしないと産業の衰退を招く.

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)

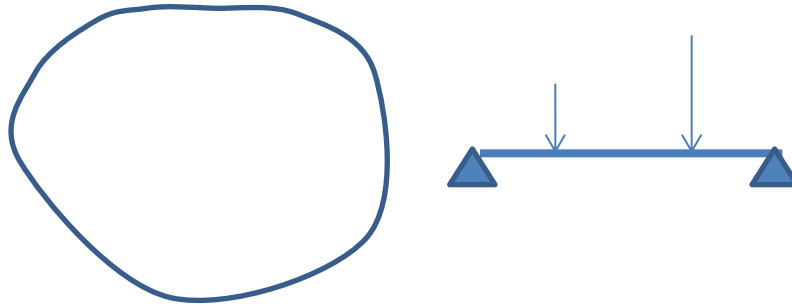
ニュートン力学の大成功:現代科学の原動力

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

文法重視

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

ラジオ少年

電子工学の技術者

音声認識, 翻訳

最近著しく性能が向上

(Google翻訳, スマホ, AIスピーカ)

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はすべてのケースを覚える.
- ・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 の 28×28 乗で全数学習は不可能.

DBA...2015年にNNが人間を越える.

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

経験的方法は訓練に時間が掛かる

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. ニューロの歴史

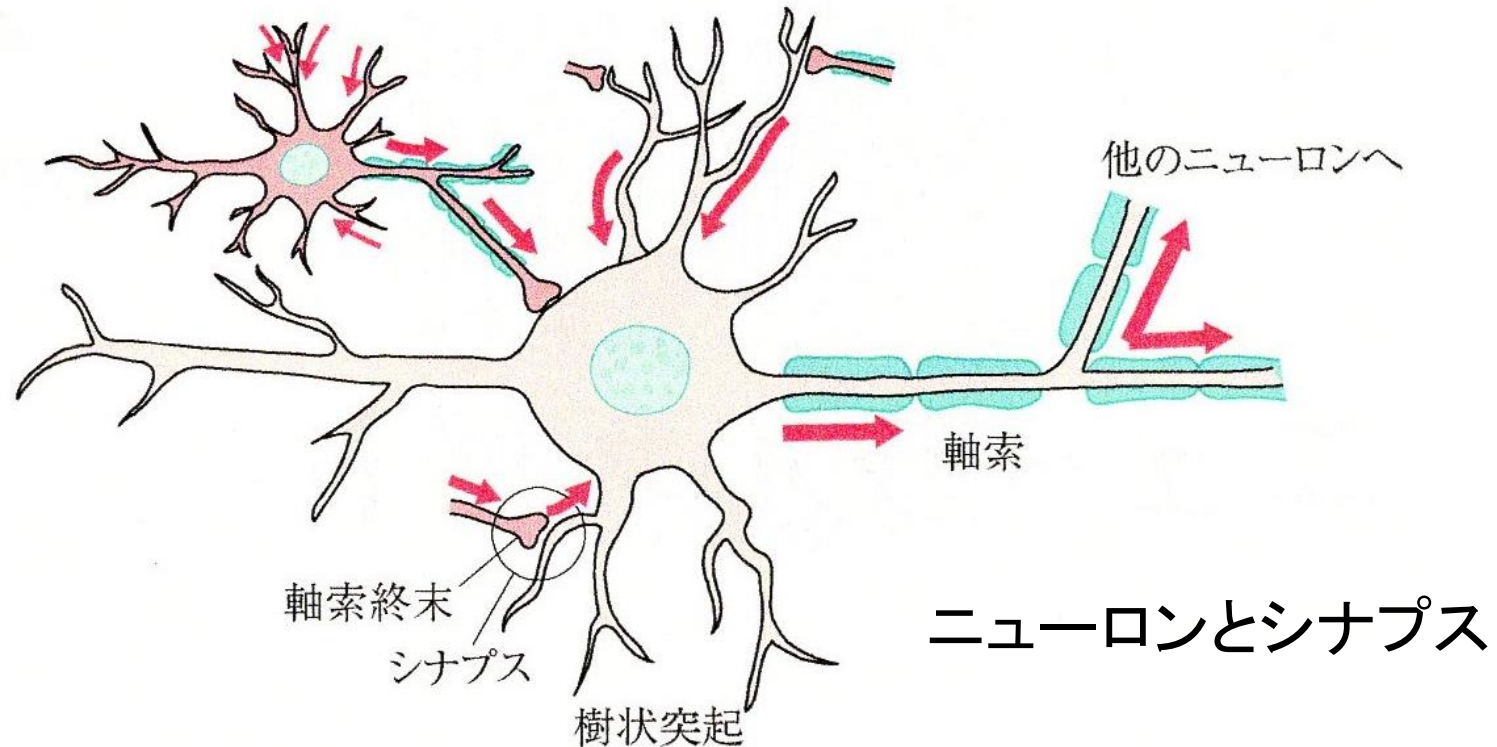
3.1 大脳の構造 ... ゴルジとカハーンの研究

ニューロンの形状を明らかにしたのはイタリア人のカミッロ・**ゴルジ**で、1873年に銀とクロムを用いた細胞の**染色法**を編み出して**神経細胞の観察**に成功する。

スペインのサンティアゴ・ラモン・イ・**カハール**は、**シナプスでは軸索終末と樹状突起は直接つながっていない**で狭い隙間があることを主張した。ゴルジは継っていると考えた。光学顕微鏡では直接確認できないため見解が別れたが、結果的にはカハールが正しかった。

ニューロン同士が直接継っていない事には、信号処理上、深い意味がある。

人間の脳には、1000億個以上の下図のような神経細胞（ニューロン）がネットワークを形成している。



神経細胞の数は大脳で数百億個、小脳で1000億個、脳全体では千数百億個にもなる。

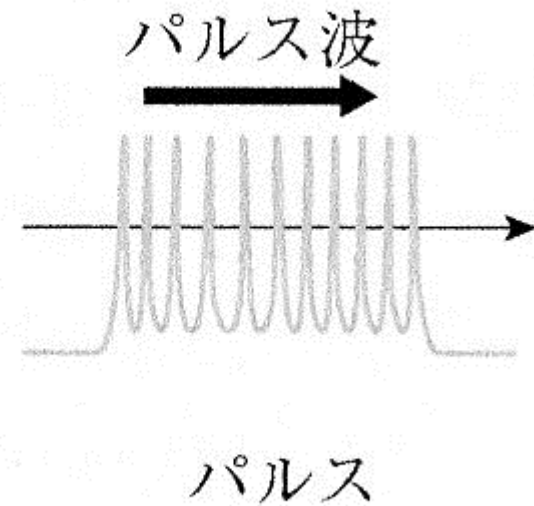
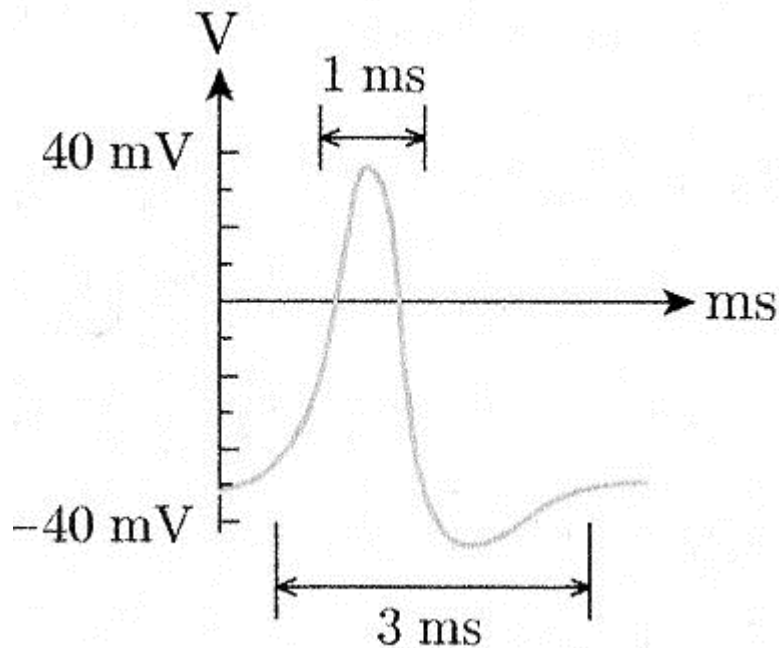
一つの神経細胞からは、信号を出力する長い「**軸索**」と、信号を入力する木の枝のように複雑に分岐した短い「**樹状突起**」が伸びている。

軸索終末と樹状突起との間には**シナプス**があり、信号が伝達される。

シナプスでは軸索を伝わってきた電気信号を受けると、**シナプス小胞**というカプセルに詰め込まれた**神経伝達物質**と呼ばれる化学物質を放出する。

樹状突起の**レセプタ**が受けると**電気信号**が細胞体（中央の膨らみ）に向けて出される。

細胞体では樹状突起から上がってきた全信号を合算し, それがしきい値を超えると軸索に向けて電気信号を発する.



活動電位と電気パルスの伝播

電気信号は連続パルスからなる. パルスの振幅は決まっており, 信号の強さはパルスの密度である.

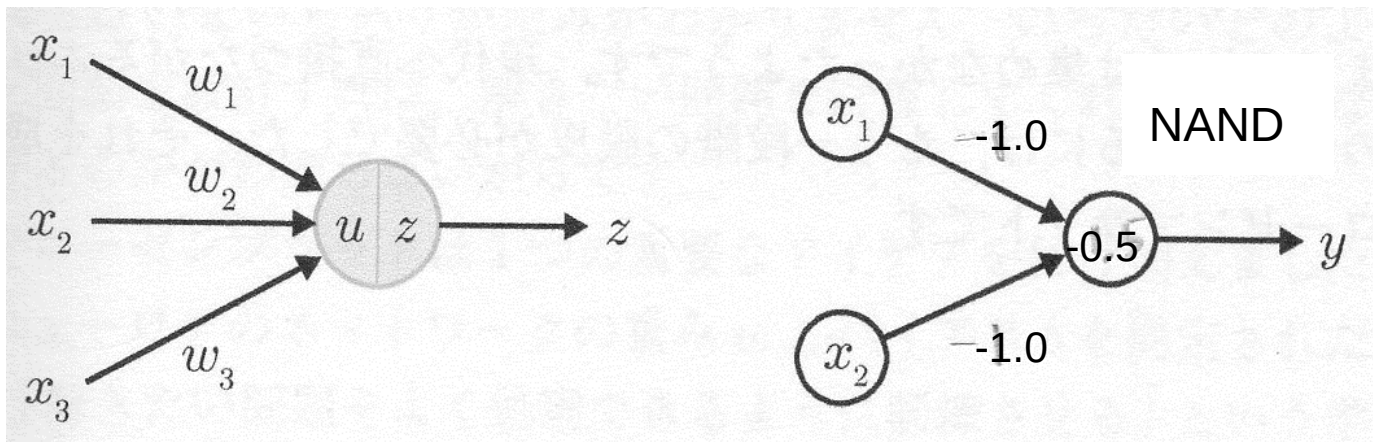
3.2 ニューラルネットワークの数理モデル

3.2.1 マカロックとピッツのモデル

NNの起源は1943年のマカロックとピッツの研究.

神経生理学者で外科医のマカロックは, チューリングの影響を受けて, 思考の計算モデルを作ろうと考えた.

神童のピッツを共同研究者として, 脳の論理処理の計算モデル(形式ニューロンと呼ばれる素子からなる回路)を考え出した.



形式ニューロン(左)と簡単なニューロン回路の例(右)

形式ニューロンには他の多数の形式ニューロンから0か1の信号 x_i が入る. 重みを w_i として総入力 u を

$$u = \sum_i w_i x_i \quad (3.1)$$

とする. このニューロンの出力 z を

$$z = \theta(u + b) = \theta\left(\sum_i w_i x_i + b\right) \quad (3.3)$$

とする. ここで θ はヘビサイドの階段関数:

$$\theta(u + b) = \begin{cases} 1 & (u \geq -b) \\ 0 & (u < -b) \end{cases} \quad (3.2)$$

入力 u を出力 z に変換する関数を**活性化関数**という.

図の右はNAND. **NNでいかなる論理演算も可能.**

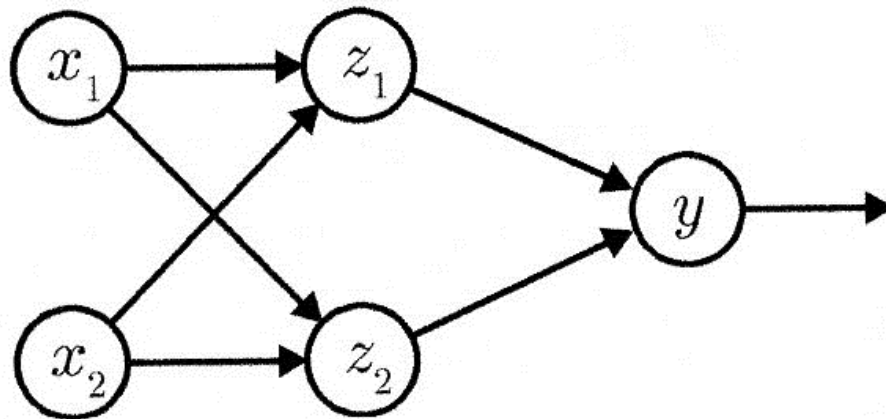
3.2.2 ローゼンブラットのパーセプトロン

前節ではパラメータ w や b は固定値であった.

ローゼンブラットはパラメータ w や b を「問題の解を与えるように訓練する」という**学習**の概念を導入した.

つまり, 図のようなNNに**教師あり学習**をさせる.

ローゼンブラットはこのようなNNを**パーセプトロン**と呼んだ.

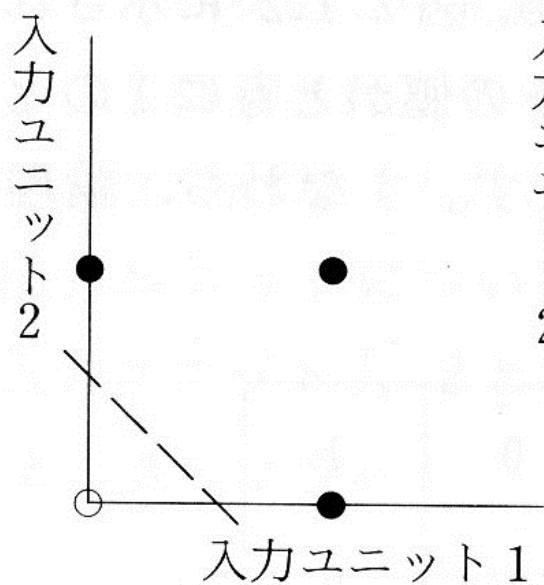


3層からなるパーセプトロンの例

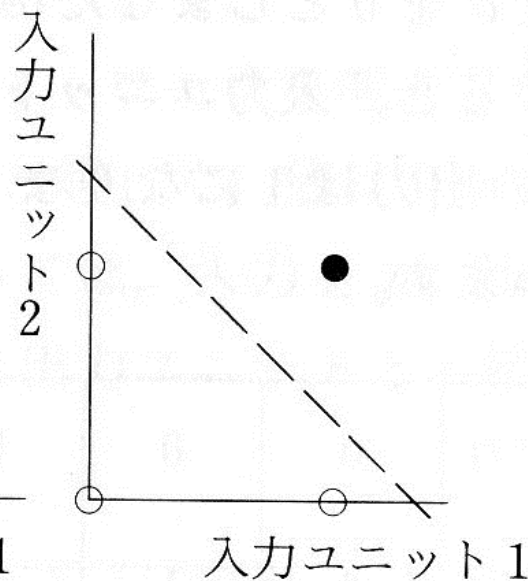
人工知能と呼ばれる研究分野を立ち上げた**ミンスキー**（ローゼンブラットの高校の同級生）は、入力と出力からなる**2層パーセプトロン**では**XORのような線形分離不可能問題は解けない**ことを示した。

これが一因となり60年代の終わりに第1次ニューラルネットブームが終焉した。

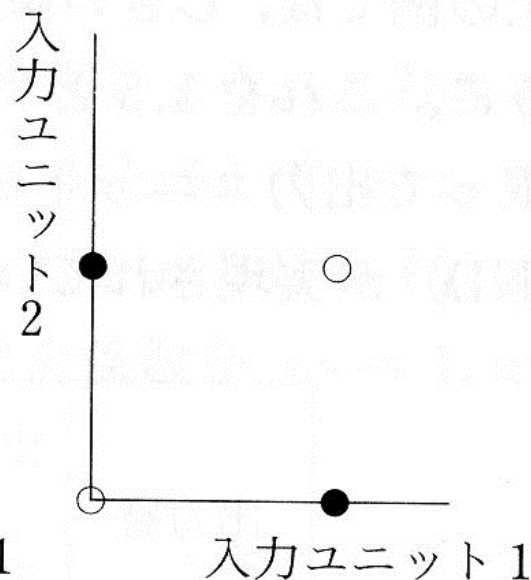
3層パーセプトロンならばXORを解ける。



OR



AND



XOR

- 出力ユニットの値が 0
- 出力ユニットの値が 1

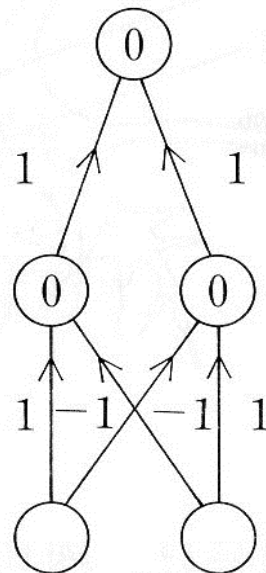
線型分離できる例とできない例

出力層	出力	0	1	1	0
	入力	0	1	1	0
中間層	出力	0 0	1 0	0 1	0 0
	入力	0 0	1 -1	-1 1	0 0
入力層	出力	0 0	1 0	0 1	1 1

出力層

中間層

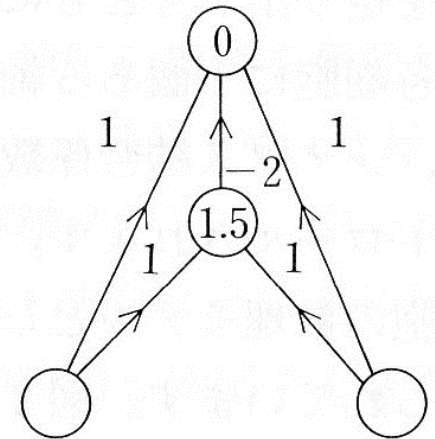
入力層



出力層

中間層

入力層

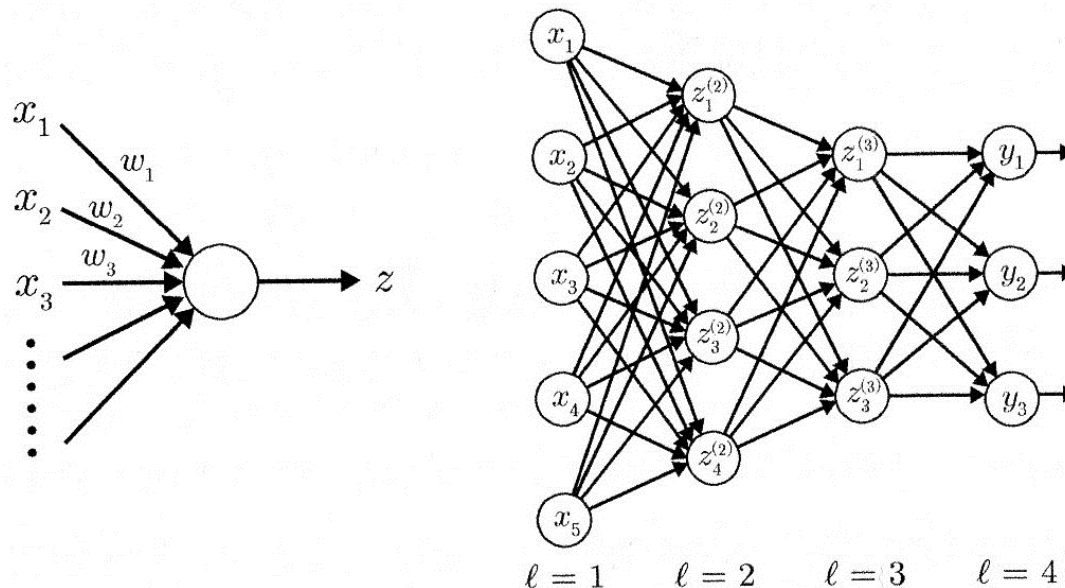


排他的論理和
(XOR)を実現するネット
ワーク2

排他的論理和
(XOR)を実現するネット
ワーク1

3.3 現在の階層型ニューラルネットワークの構造

図に階層ニューラルネットワークを示す. パーセプトロンの入出力は0, 1であったがNNでは実数値.



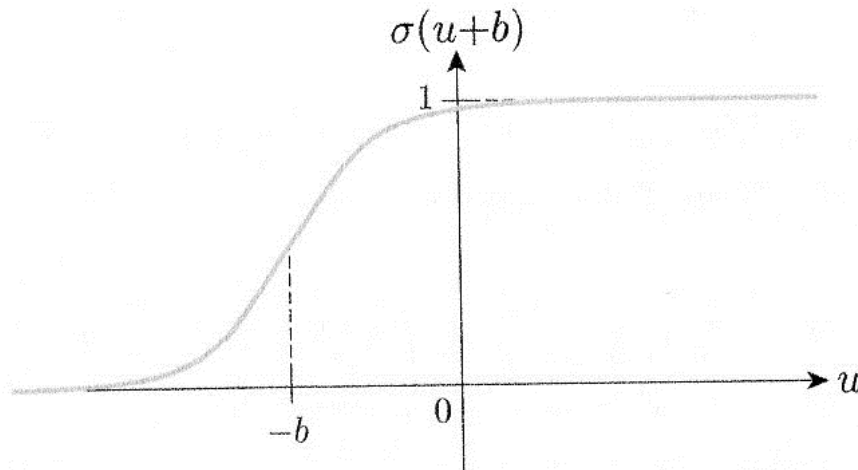
左はユニットの構造. 右は4層順伝播型NNの例.
丸の中の変数は各ユニットの出力値を意味する.

出力と入力, 入力と出力の関係はパーセプトロンと同じ. 改めて書くと

$$u = \sum_i w_i x_i \quad (3.6)$$

$$z = f(u + b) = f\left(\sum_i w_i x_i + b\right) \quad (3.7)$$

パーセプトロンとの違いは**入出力が実数**になっていること, 活性化関数には**連続関数**が使われる.



シグモイド関数

3.4 深層学習ニューラルネットワークの登場

最近NNが大きな話題

ブレークスルーのキーワードは**深層学習**

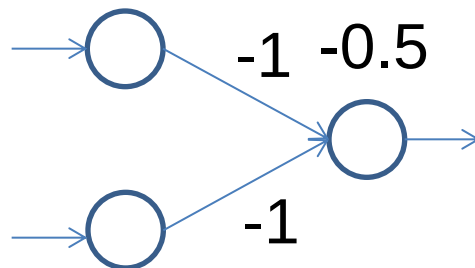
人工知能の中核技術になる勢い

これまでのNN

汎化能力(未学習のデータに正解する能力)が
非常に低い

NNはいかなる問題も解けることが数学的に保証

NNはNAND回路を作れる. NAND回路を用いればいかなる論理回路も構成できる.



NAND回路

汎化能力を高めるために多層化すると、ローカル・ミニマムに捕まって学習を継続できない



NNの冬の時代

2006年にHintonらが深層ボルツマン・マシンに成功

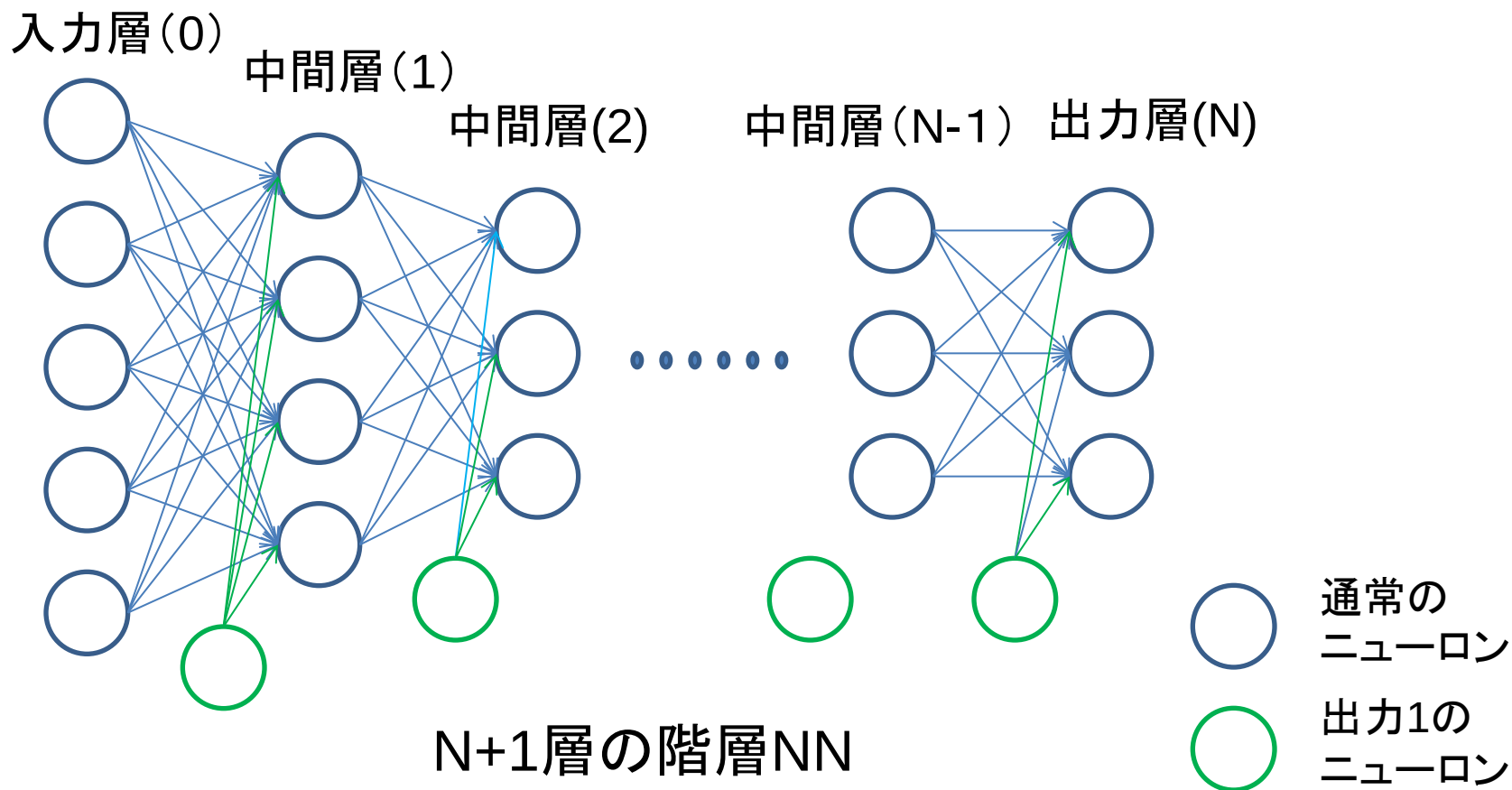
2012年にHintonらが深層NNが画像認識で画期的な誤り率をたたき出す

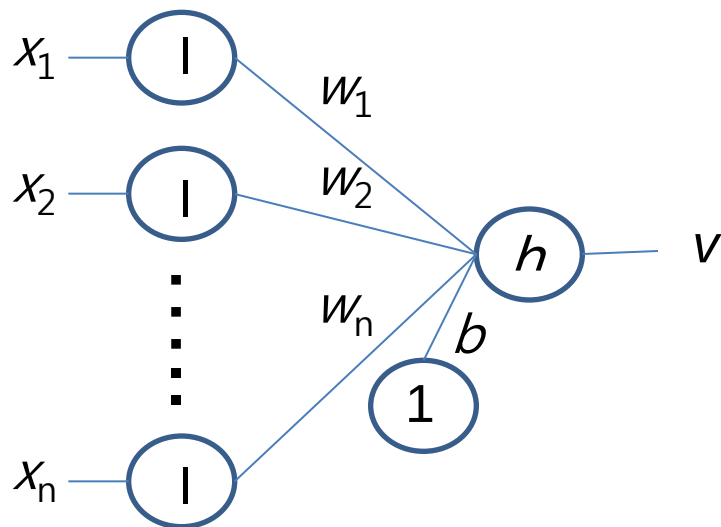


深層NNの時代到来

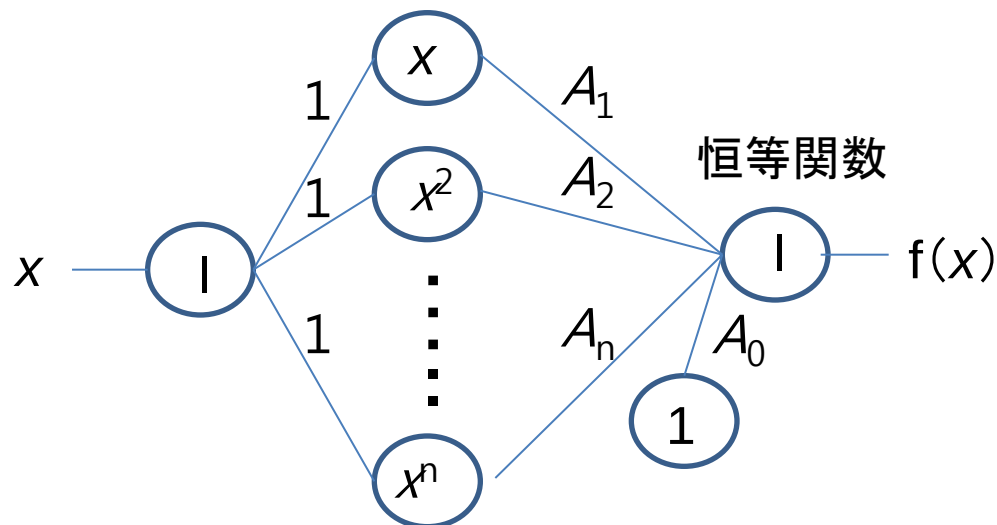
4. 学習とは

誤差逆伝播学習： 誤差（教師データと出力データの差）
をゼロにする学習をする





2層NN
(Iは恒等関数)



多項式補間のNN的表現

$$f(x) = A_0 + A_1 x + A_2 x^2 + \dots + A_n x^n$$

出力ユニット入力: $u = \sum_i w_i x_i + b$

出力ユニット出力: $v = h(u) = h\left(\sum_i w_i x_i + b\right)$

未知数は $w_1, w_2, \dots, w_n$ と b の $n+1$ 個.

学習方法を $v = t$ or $h\left(\sum_i w_i x_i + b\right) = t$

とすると, 必ずしも解けない.

簡単な例： 2入力のANDは, 3つの未知数 w_1, w_2, b に対して、4つの入力がある.

$$x_1 = 0, x_2 = 0 \rightarrow 0 = h(b)$$

$$x_1 = 0, x_2 = 1 \rightarrow 0 = h(w_2 + b)$$

$$x_1 = 1, x_2 = 0 \rightarrow 0 = h(w_1 + b)$$

$$x_1 = 1, x_2 = 1 \rightarrow 1 = h(w_1 + w_2 + b)$$

h を恒等関数とすると, 上の3つの方程式から

$b = w_1 = w_2 = 0$ となり4番目の方程式を満足できない.

しかし, h を **非線形関数** であるヘビサイドのステップ関数とし, $b = -1.5$ とすると, 解になっている.

そこで、出力関数 h は非線形とする。さらに、誤差関数を定義し、誤差が最少となる学習方法に変更する:

$$E = \frac{1}{2} (t - z)^2 = \frac{1}{2} \left(t - h \left(\sum_i w_i x_i + b \right) \right)^2 = \min$$

これより
$$\frac{\partial E}{\partial w_i} = -(t - z) \frac{\partial z}{\partial w_i} = -(t - z) h'(u) x_i$$

$$\frac{\partial E}{\partial b} = -(t - z) \frac{\partial z}{\partial b} = -(t - z) h'(u) \cdot 1$$

最急降下法によるウェイト w_i とオフセット b の修正量:

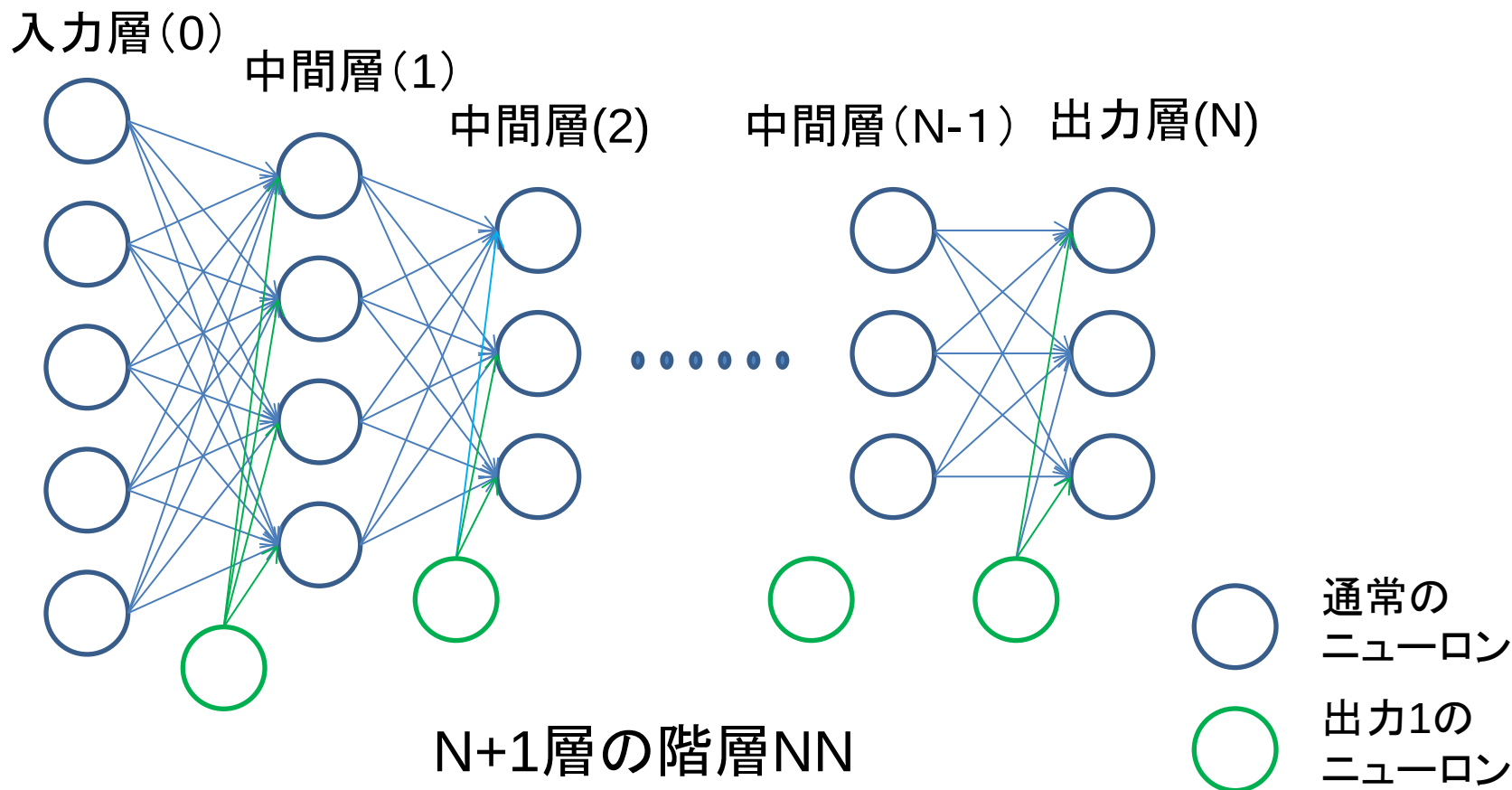
$$\Delta w_i = -\frac{\partial E}{\partial w_i} \alpha \quad \Delta b = -\frac{\partial E}{\partial b} \beta$$

ウェイト w_i とオフセット b の修正:

$$(w_i)_{New} = (w_i)_{Old} + \Delta w_i \quad (b)_{New} = (b)_{Old} + \Delta b$$

誤差逆伝播学習 (NNの理論ではこれが一番難しい)

入力層が第0層, 中間層が第1,2,..., N-1層, 出力層が第N層の場合に一般化すると



$n+1$ 層の j ユニットへの入力:

$$u_j^{(n+1)} = \sum_i w_{ji}^{(n)} v_i^{(n)} + b^{(n)}, \quad n = 0, 1, \dots, N-1$$

$n+1$ 層の j ユニットの出力:

$$v_j^{(n+1)} = h(u_j^{(n+1)}) = h\left(\sum_i w_{ji}^{(n)} v_i^{(n)} + b^{(n)}\right)$$

誤差関数:

$$\begin{aligned} E &= \frac{1}{2} \sum_j (t_j - v_j^{(N)})^2 \\ &= \frac{1}{2} \sum_j \left(t_j - h\left(\sum_i w_{ji}^{(N-1)} v_i^{(N-1)} + b^{(N-1)}\right) \right)^2 \end{aligned}$$

$$\frac{\partial E}{\partial w_{ji}^{(N-1)}} = -(t_j - v_j^{(N)}) \frac{\partial v_j^{(N)}}{\partial w_{ji}^{(N-1)}}$$

$$\frac{\partial v_j^{(N)}}{\partial w_{ji}^{(N-1)}} = h'(u_j^{(N)}) v_i^{(N-1)}$$

したがって

$$\frac{\partial E}{\partial w_{ji}^{(N-1)}} = -(t_j^{(N)} - v_h^{(N)}) h'(u_j^{(N)}) v_i^{(N-1)}$$

同様に

$$\frac{\partial E}{\partial b^{(N-1)}} = -(t_j^{(N)} - v_h^{(N)}) h'(u_j^{(N)}) \cdot 1$$

誤差の定義と計算法:

$$\begin{aligned}\delta_j^{(N)} &= -\frac{\partial E}{\partial u_j^{(N)}} = -\frac{\partial E}{\partial z_j^{(N)}} \frac{\partial z_j^{(N)}}{\partial u_j^{(N)}} \\ &= (t_j - z_j^{(N)}) h'(u_j^{(N)}) \\ \delta_j^{(n+1)} &= -\frac{\partial E}{\partial u_j^{(n+1)}} = -\frac{\partial E}{\partial z_j^{(n+1)}} \frac{\partial z_j^{(n+1)}}{\partial u_j^{(n+1)}} \\ &= -h'(u_j^{(n+1)}) \sum_k \frac{\partial E}{\partial u_k^{(n+2)}} \frac{\partial u_k^{(n+2)}}{\partial z_j^{(n+1)}} \\ &= h'(u_j^{(n+1)}) \sum_k \delta_k^{(n+2)} w_{kj}^{(n+2)}\end{aligned}$$

誤差 $\delta_j^{(n)}$ は上の階層から決まるこれが**誤差逆伝搬法**の名前の由来.

勾配の計算法:

$$\begin{aligned}\frac{\partial E}{\partial w_{ji}^{(n)}} &= \frac{\partial E}{\partial u_j^{(n+1)}} \frac{\partial u_j^{(n+1)}}{\partial w_{ji}^{(n)}} = -\delta_j^{(n+1)} v_i^{(n)} \\ \frac{\partial E}{\partial b^{(n)}} &= \frac{\partial E}{\partial u_j^{(n+1)}} \frac{\partial u_j^{(n+1)}}{\partial b^{(n)}} = -\delta_j^{(n+1)} \cdot 1\end{aligned}$$

最急降下法によるウェイト w_j とオフセット b の修正量:

$$\Delta w_{ji}^{(n)} = -\frac{\partial E}{\partial w_{ji}^{(n)}} \alpha \Delta b^{(n)} = -\frac{\partial E}{\partial b^{(n)}} \beta$$

勾配は誤差逆伝搬法によらなくても**数値微分**でも可能であろう.

RTINY.c による計算結果1...ほぼ同じものを第8回講義で配布済

AND, OR, XORの計算

計算条件

システムデータ

IU = 2

HU = 2

OU = 3

NT = 4

計算パラメータ

u0 0.6

inival_wgt 0.3

inival_off 0.2

eta_wgt 0.8

eta_off 0.6

alpha 0.3

dalpha 0.01

alpha_lmt 0.9

error_lmt 0.005

iteration_lmt 1000

入力データ

(1) 0, 0

(2) 1, 0

(3) 0, 1

(4) 1, 1

教師データ

(1) 0, 0, 0

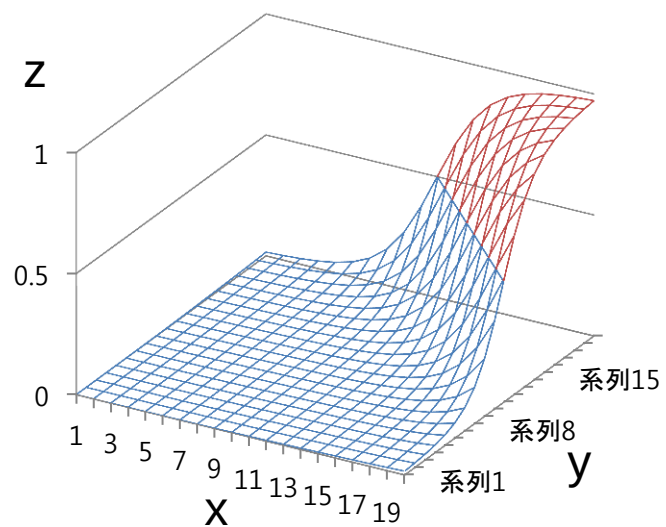
(2) 0, 1, 1

(3) 0, 1, 1

(4) 1, 1, 0

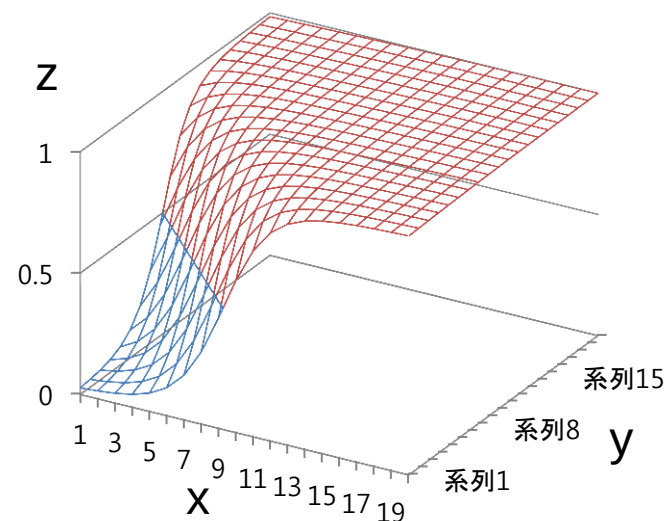
出力結果

入力値	入力値	出力値	出力値	出力値
ユニット0	ユニット1	ユニット0	ユニット1	ユニット2
		AND	OR	XOR
0	0	0	0.02	0.04
1	0	0.02	0.98	0.97
0	1	0.02	0.98	0.97
1	1	0.98	1	0.04



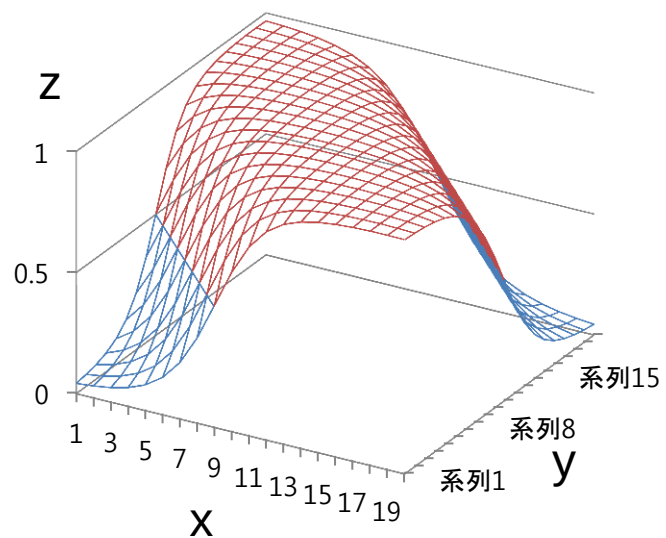
(a) AND

0.5-1
0-0.5



(b) OR

0.5-1
0-0.5



(c) XOR

0.5-1
0-0.5

AND, OR, XOR で
できる曲面

RTINY.c による計算結果2(シグモイド→ReLU)

入力-出力間の関係が線形変換になるので
XOR(線形不分離)が解けなくなる.

入力値 入力値 出力値 出力値 出力値

ユニット0 ユニット1 ユニット0 ユニット1 ユニット2

AND

OR

XOR

0	0	0	0.29	0
1	0	0.19	0.73	0
0	1	0.04	0.84	0
1	1	0.88	1.21	0

5. 深層学習とは何か？

最近NNが大きな話題

ブレークスルーのキーワードは**深層学習**

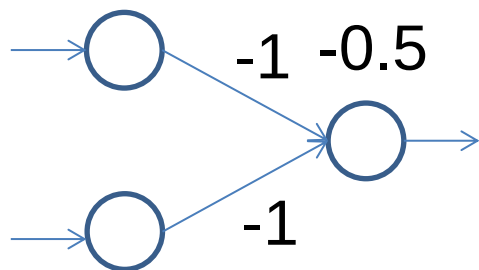
人工知能の中核技術になる勢い

これまでのNN

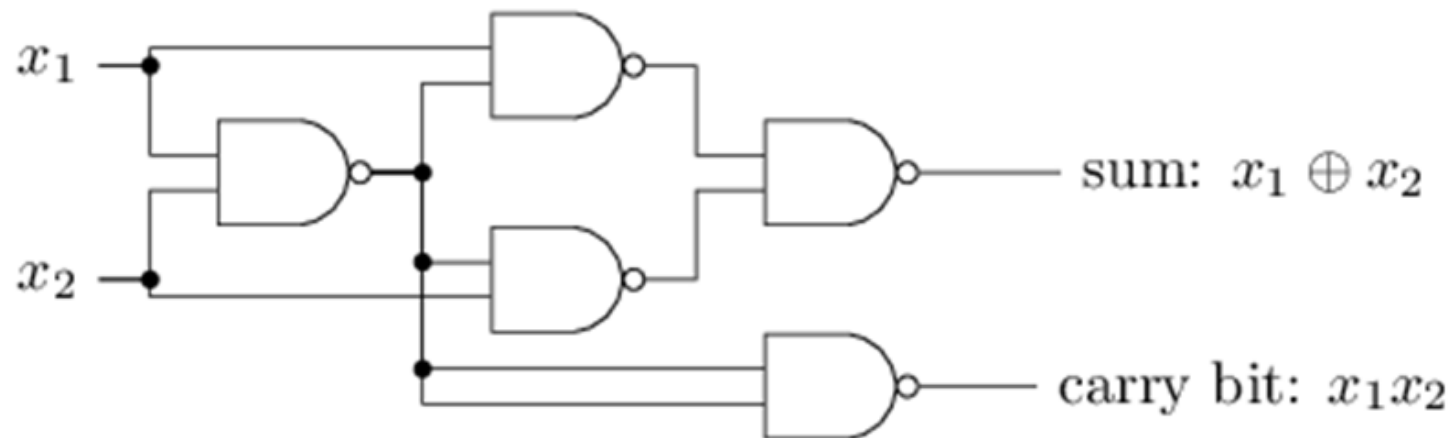
汎化能力(未学習のデータに正解する能力)が非常に低い

NNはいかなる問題も解けることが数学的に保証

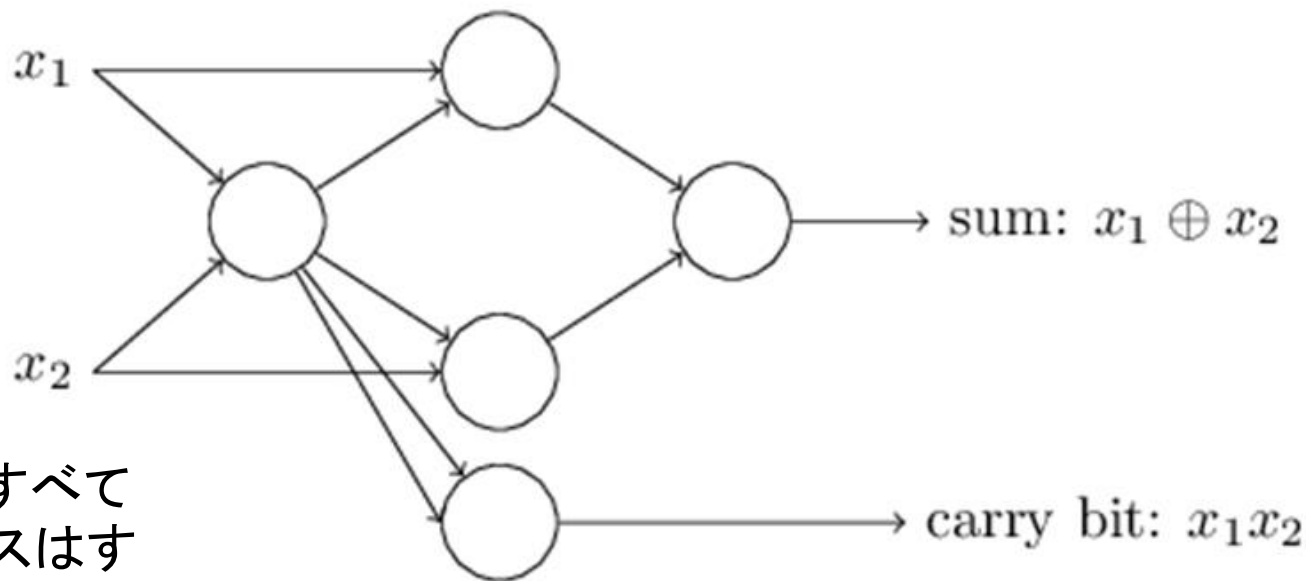
NNはNAND回路を作れる. NAND回路を用いればいかなる論理回路も構成できる.



NAND回路



NANDゲートによる2進数の和の実装(電子回路)



ウェイトはすべて
-2、バイアスはす
べて3

ニューロによる2進数の和の実装

汎化能力を高めるために**多層化すると、ローカル・ミニマムに捕まって学習を継続できない**



NNの冬の時代

2006年にHintonらが**深層ボルツマン・マシン**に成功

2012年にHintonらが**深層NN**が画像認識で画期的な誤り率をたたき出す



深層NNの時代到来

2015年に人間を凌駕する結果が出る！

従来のNNの問題点と対策

汎化能力が低い、しかし多層化できない
十分に学習できない

シグモイド以外の活性化関数の導入

例えばReLU (Rectified Linear Unit)

前処理ネットワークの導入

自己符号化器 (Autoencoder)

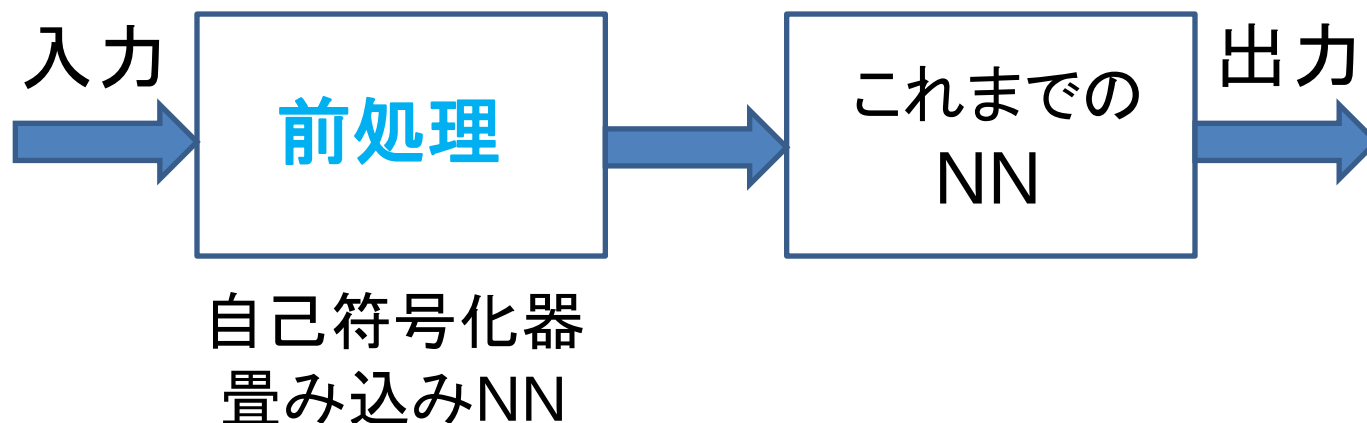
畳み込みNN (Convolution NN)

学習し過ぎて過学習になる

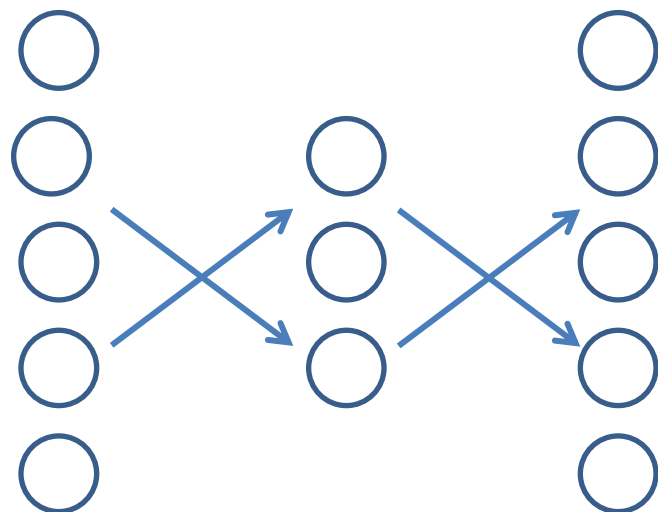
学習の制限

正則化

新しいアーキテクチャの導入:



自己符号化器 (Autoencoder)



ニューロン数:

入力層 = 出力層 > 中間層

学習:

教師データ = 入力データ

中間層に圧縮されたデータ

自己符号化器の利用

ノイズ除去:

- (1) 正常データで学習させた自己符号化器にノイズありデータを入力するとノイズが落ちる.
- (2) 入力データにノイズありデータ, 教師データにノイズなしデータを与えて学習させると, ノイズ除去フィルタができる.

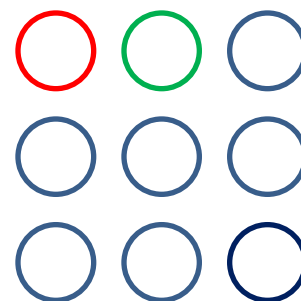
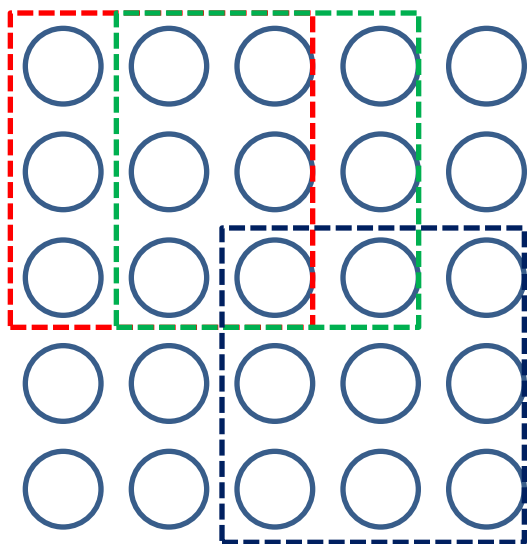
故障, 異常診断:

正常データで学習させた自己符号化器に異常データを見せると, 正常データを出力するので, 入力データと出力データの差を取ると, 異常が検知される.

畳み込みNN (Convolution NN)

瀧 雅人, 「これなら分かる
深層学習入門」, 講談社,
2017, p. 159

データを
2次元の
に考える



$$26 = 5 \times 2 + 1 \times 0 + 0 \times (-2) \\ + 1 \times 0 + 6 \times 2 + 1 \times 0 \\ + 4 \times (-2) + 1 \times 0 + 6 \times 2$$

5 1 0 1 9

1 6 1 7 0

4 1 6 3 5

1 8 0 1 0

2 2 8 0 4

*

2 0 -2

0 2 0

-2 0 2

=

26 6 -6

0 -4 8

24 -14 -4

畳み込みニューラルネット

この構造を開発する**ヒント**になったのは、**動物の視覚野の構造**である。

ヒューベルとウィーゼルの階層仮説

人間にはパターンを認識する高い能力が備わっている。

網膜で受け取られた**視覚情報**の電気信号は、**画像としての平面構造**を保ったまま、外側膝状体を経由して、**1次視覚系**に入力する。1次視覚系は**後頭野**にある。ここで、**パターン認識プロセス**の重要な**第1段階**が行われる。

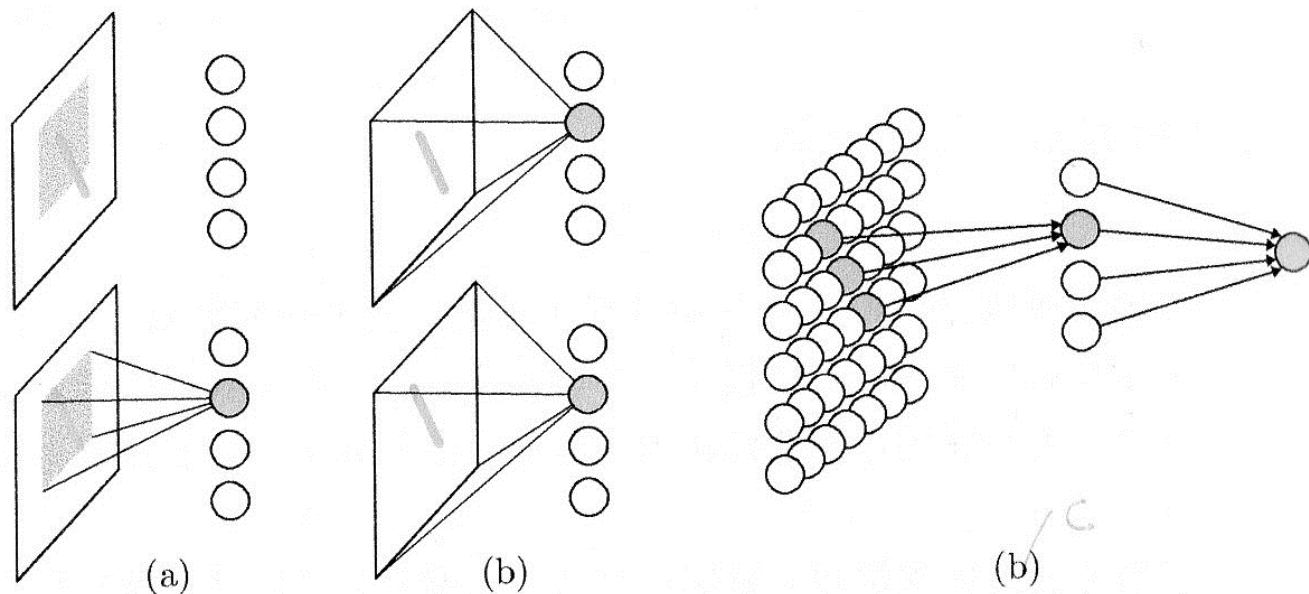
1958年ヒューベルとウィーゼルは、猫の視覚野に特定の傾きをもつ線分を見せたときにだけ反応する細胞があることを発見した.

このような細胞は単純型細胞と複雑型細胞に大別され、これらの違いは図の受容野の概念で説明される

図(a)(b)の4つのケースそれぞれは、左の四角い領域(網膜)から視覚信号を受け取った際に、縦に並んだ4つのニューロンがどう変化するかを示す概念図である.

4つのうち上から2つ目のニューロンに注目する.

単純型細胞(a)では、パターンが灰色で塗りつぶされた領域にぴったり現れた時にのみニューロンが発火する。
(a)の下の方では発火するが、上の方では発火しない。
複雑型細胞(b)はパターンがずれた時でも発火する。



(a) 単純型細胞、(b) 複雑型細胞. 単純型細胞は、パターンの形状に対応した網膜型細胞とのみ結合を持つ. ヒューベルとウィーゼルの説では、複雑型細胞はそのような単純型細胞を束ねる形で実現している.

ヒューベルとウィーゼルの階層仮説によると、単純型細胞はパターン内の受容野とのみ結合している。刺激が受容野とぴったりあった時に最も大きく反応する。

(c)の3層目のニューロンは複雑型細胞で単純型細胞を束ねている。

したがって複雑型細胞は網膜上のより広い範囲にパターンがあれば反応する。

階層仮説が大きなヒントとなって生まれたのが畳み込みニューラルネットワーク (convolutional network CNN) である。

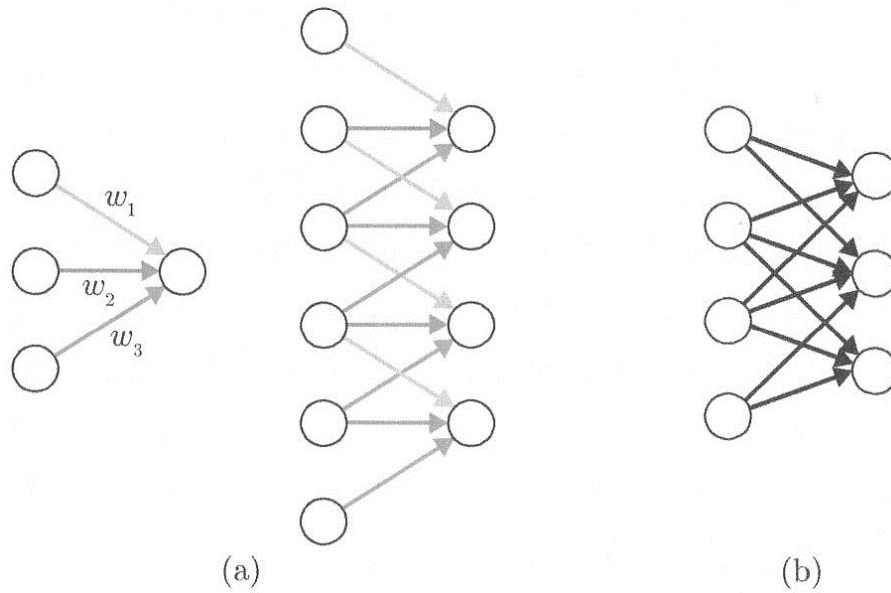
CNNの起源は福島邦彦のネオコグニトロン (1979) に遡る。1989年にカルカンのグループが同じネットワークに誤差逆伝播法を導入して高い性能を実現した (LeNet)。

ニューラルネットと畳み込み

畳み込みニューラルネットは2種類の層からなる.

1つ目の層は単純型細胞からなり図の(a)のように全体を覆う. (a)の右図のように次層と結合されている. どこにパターンが現れても次層のユニットが活性化する.

1層と2層の間はスパースな結合である.



(a) 左は3つのユニットからなる局所受容野から入力を受ける単純型細胞、(b) 単純型細胞の層から入力を受ける複雑型細胞の層.

パターンの現れる位置で認識能力が変わらないためには、結合がすべて同じ重みを持っていればよい。

結合の多さにもかかわらずパラメータ数は少ない。

畳み込みそうには重みのスパース化と共有という2つの正規化が課せられていると看做せる。

プーリング層

複雑型細胞に対応するプーリング層 (pooling layer) の説明をする.

プーリング層の役割は単純型細胞の出力を合算して入力位置の平行移動に対してロバストにすること.

$W \times W \times K$ 入力画像 $z_{i,j,k}^{(l-1)}$ の各チャンネルに対し, 各画素位置 (i,j) を中心とした $H \times H$ の領域 P_{ij} を考える.

ただし画像はパディングしておき, どの点に対しても P_{ij} の領域を取れるようにしておく.

P_{ij} 中の画素から, その領域での代表的な値 $u_{i,j}^{(l)}$ を決定する.

最大プーリングでは, 領域中の最大の画素値を代表値とする.

$$u_{ijk}^{(l)} = \max_{(p,q) \in P_{ij}} z_{pqk}^{(l-1)}$$

最大プーリング以外にも、**平均プーリング**：

$$u_{ijk}^{(l)} = \frac{1}{H^2} \sum_{(p,q) \in P_{ij}} z_{pqk}^{(l-1)}$$

この2つのプーリングを拡張した方法として、 L^P **プーリング**：

$$u_{ijk}^{(l)} = \left(\frac{1}{H^2} \sum_{(p,q) \in P_{ij}} \left(z_{pqk}^{(l-1)} \right)^P \right)^{\frac{1}{P}}$$

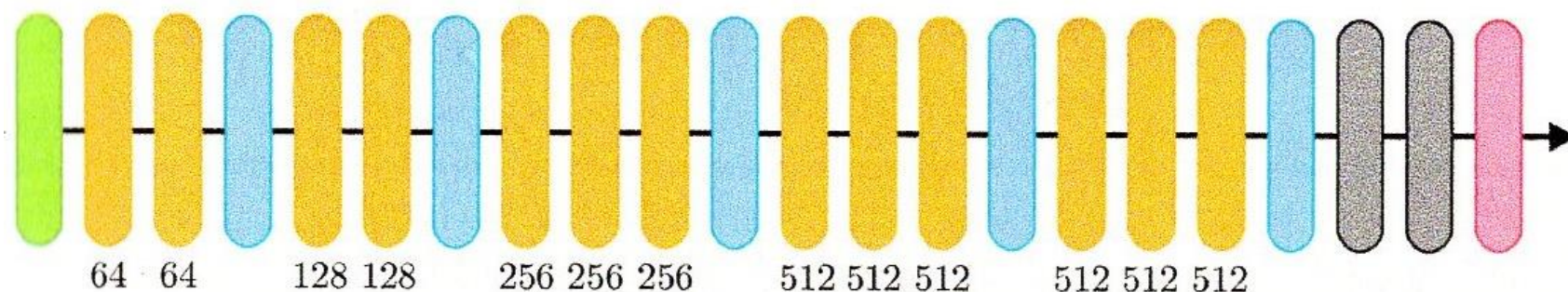
$P=1$ の場合は明らかに平均プーリングになるが、対極の $P \rightarrow \infty$ では、最大プーリングになる。（証明省略）

確率的プーリングというものもある。

プーリング層の役割は畳み込み層の出力を束ねるだけだから重み学習はしない。学習中もプーリングの式は変更されない。

ネットワーク構造

典型的なCNNの例を図に示す(オックスフォード)。



VGGの一例。オレンジ色は畳み込み層，青はマックスプーリング層。灰色は全結合層，そして赤は出力のソフトマックス層。数字は各畳込み層のフィルタの幅。

学習済みモデルと転移学習

ImageNetなどの大きなデータで学習したモデルは、他の機械学習とは比較にならないほどの高い画像分類性能を達成できる。

しかし多層のCNNを用いた自然画像の学習には多くのコストと計算時間を要する。

もしも、学習済のモデルがImageNetデータの分類以外には全く役立たないならば、コストが高すぎて、魅力が半減する。

しかし、ニューラルネットには高い汎化性能があり、深いネットワーク構造では、中間層にさまざまな特徴量を獲得していると期待される。

そこで、学習済のモデルの出力側の層を取り除いて、**中間層からの出力を入力画像の表現**として用いることが考えられる。

出力に近い側は出力層に設定した学習タスクに強く影響を受けていて、他のタスクへの応用に向かない。

出力層だけを置き換えて他のタスクのために学習し直す方法も用いられる。

学習済みパラメータが初期値となって、**学習がスムーズに進行**する。

この方法を**転移学習**という。

CNNはどのようにパターンを捉えているのか

CNNの高い学習能力の詳細は多く分かっていない。しかし、CNNの中を覗き見る技術はさまざま開発されている。

図の(1), (2), (3)はそれぞれVGG16のある特定の画像で、(1)から(3)の順に出力寄りから入力寄りの順に並んでいる。

これを見ると、低層から高次の層に行くに連れ、各フィルタがより複雑なパターンを捉えている。ただ、高次の層でのパターンを直感的に理解できるものでない。

図の(a)から(f)は、出力層から選んだ6個のユニットに対応した画像である.

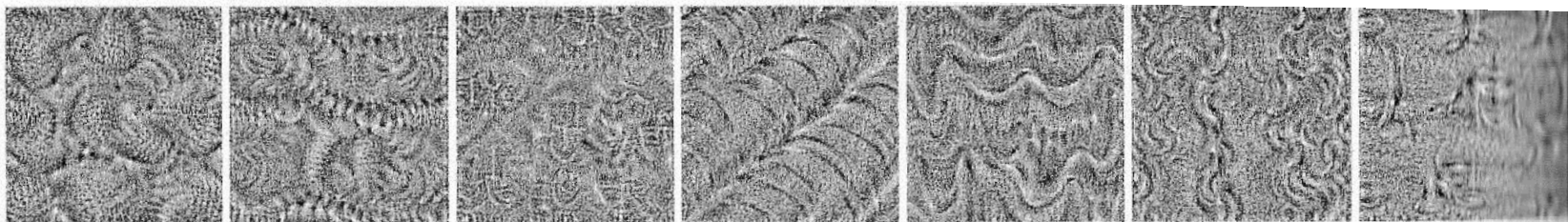
(a)三葉虫, (b)シベリアンハスキー, (c)シマウマ, (d)タクシー, (e)図書館, (f)エスプレッソである.

ランダムノイズを初期値として作成した入力画像である.
99.9%という高い確率で、各カテゴリに分類される.

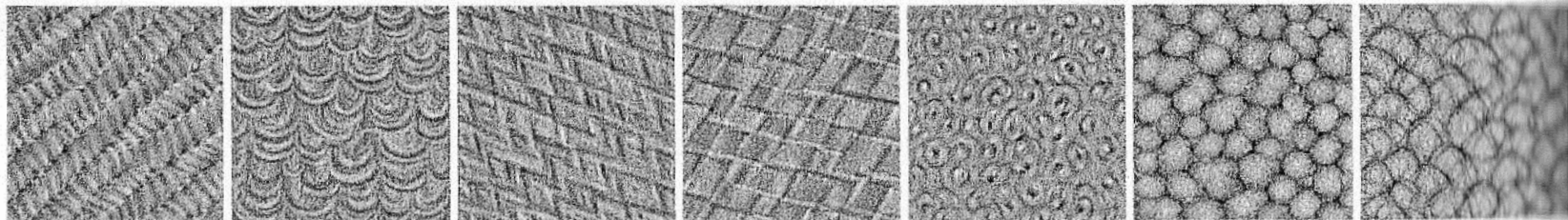
人間がこの画像を見ても、全く認識できない.

ノイズに対するこの謎めいた挙動はadversarial examplesという現象と類似のもので、現在でも活発に研究されている.

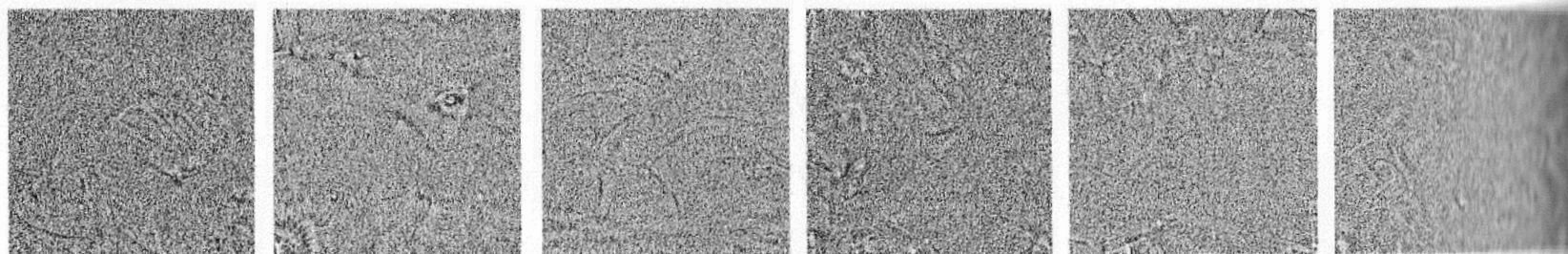
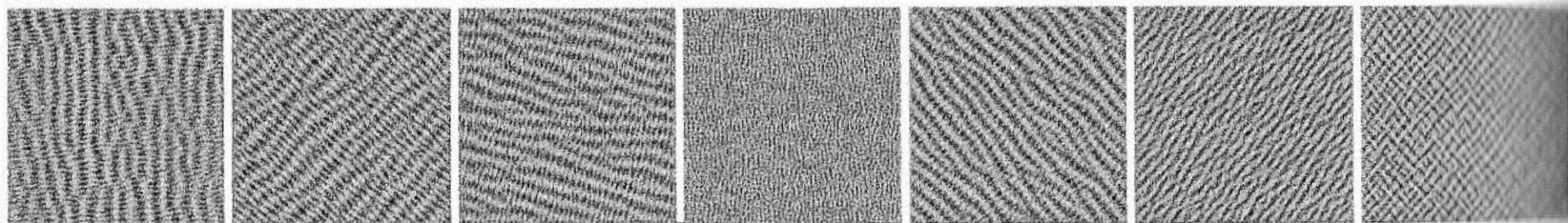
(1)



(2)



(3)



(a)

(b)

(c)

(d)

(e)

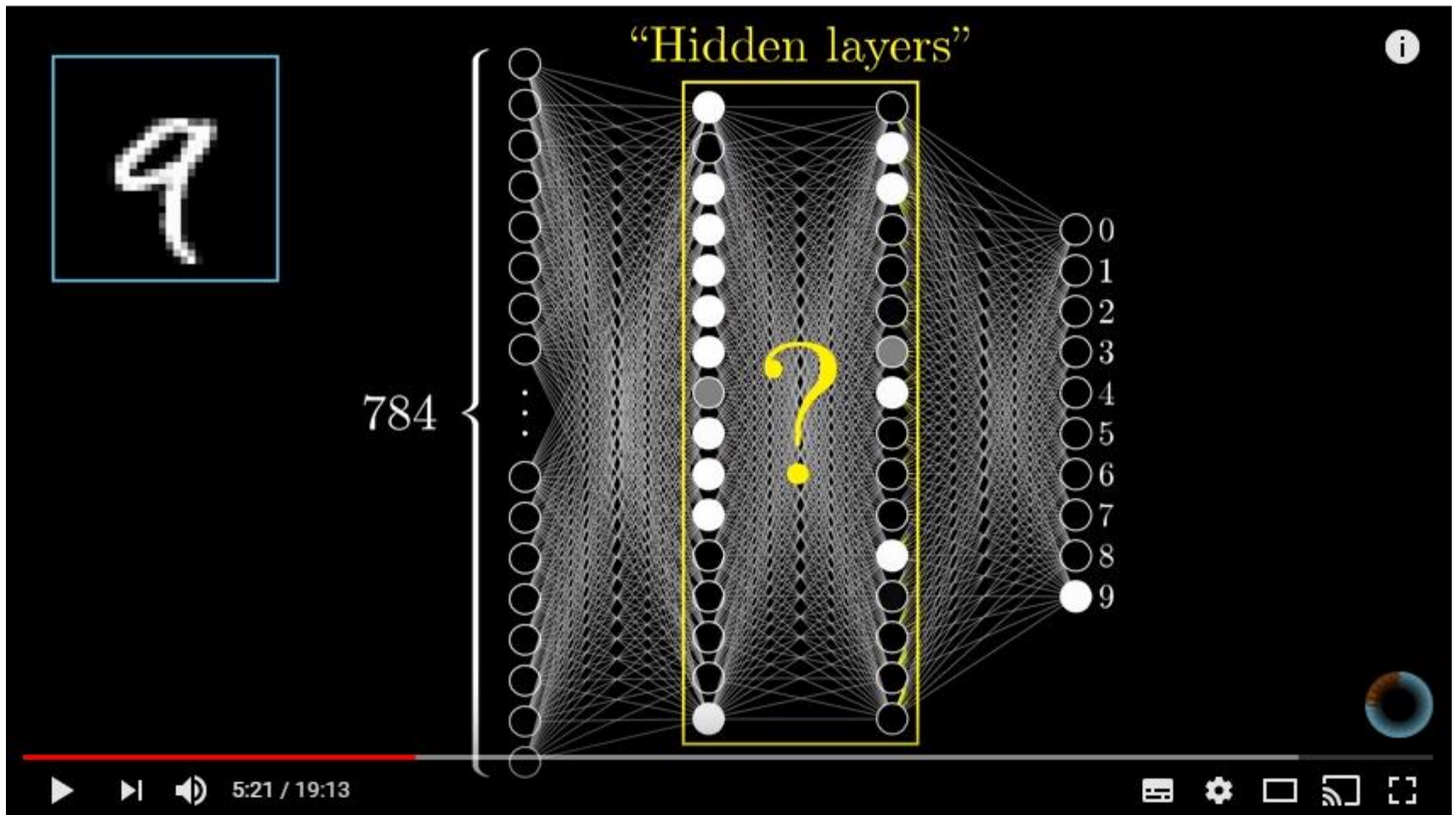
(f)

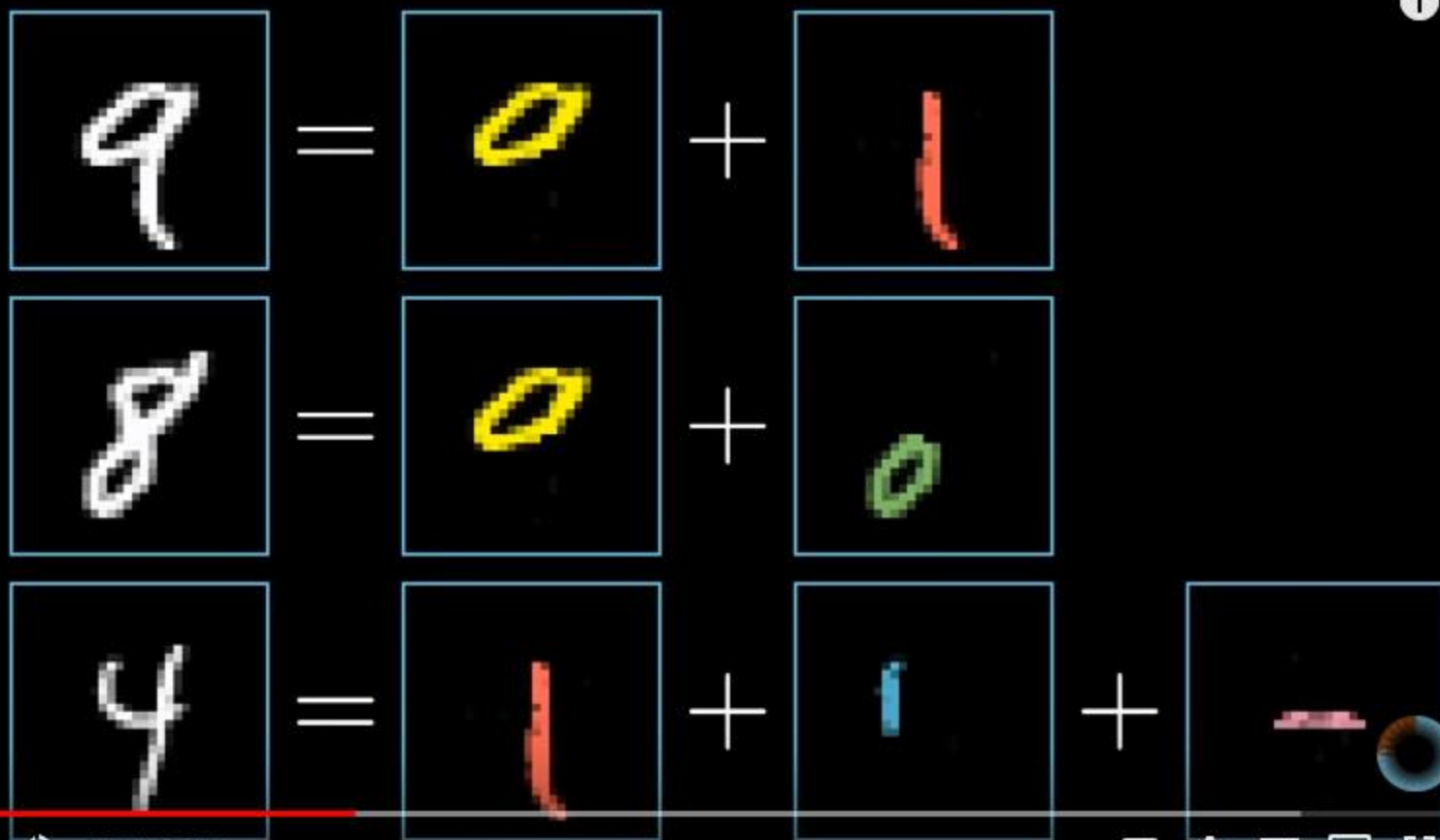
VGG16のいくつかの層のユニット活性を極大化させるように「訓練」した入力画像たち

But what **is** a Neural Network? |

Chapter 1, deep learning

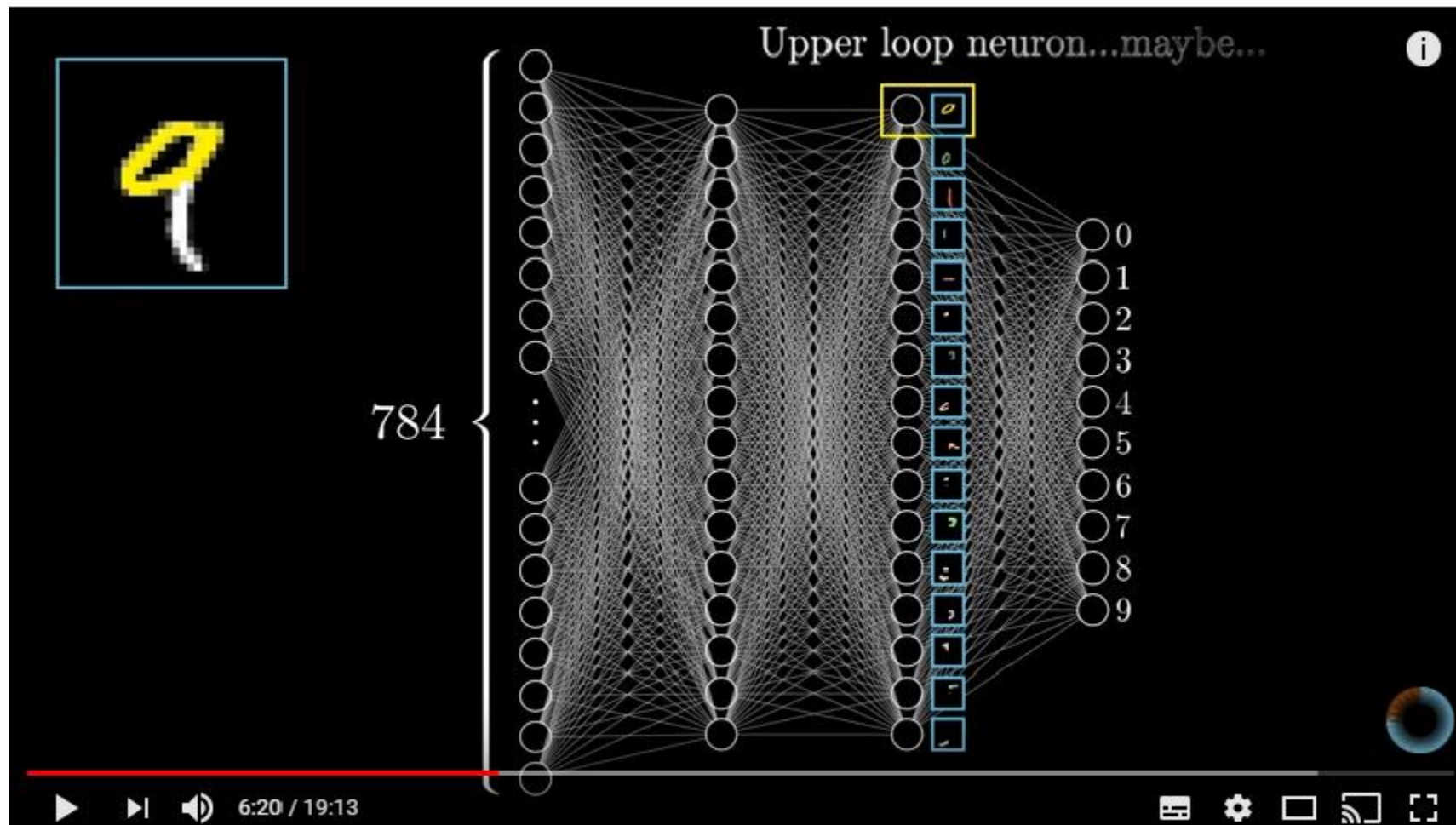
<https://www.youtube.com/watch?v=aircAruvnKk>





6:06 / 19:13





単純型細胞の出力
細かい特徴

複雑型細胞の出力
複合された特徴

正則化

過学習の防止： 過剰な自由度を除く

重み減衰 (Weight Decay) 項を導入：

$$E_{WD}(\mathbf{w}) = E(\mathbf{w}) + \lambda \mathbf{w}^T \mathbf{w} = \text{minimum}$$

誤差関数 $E(\mathbf{w})$ だけでなく, $\mathbf{w}^T \mathbf{w}$ も最小化されるので, 不要な重みは出来るだけゼロに近づく.

実際的な方法： 訓練データを学習データと評価データに分け, 評価データが増加に転じた時点で学習をやめる.

6. 深層学習ニューロの応用

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して

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

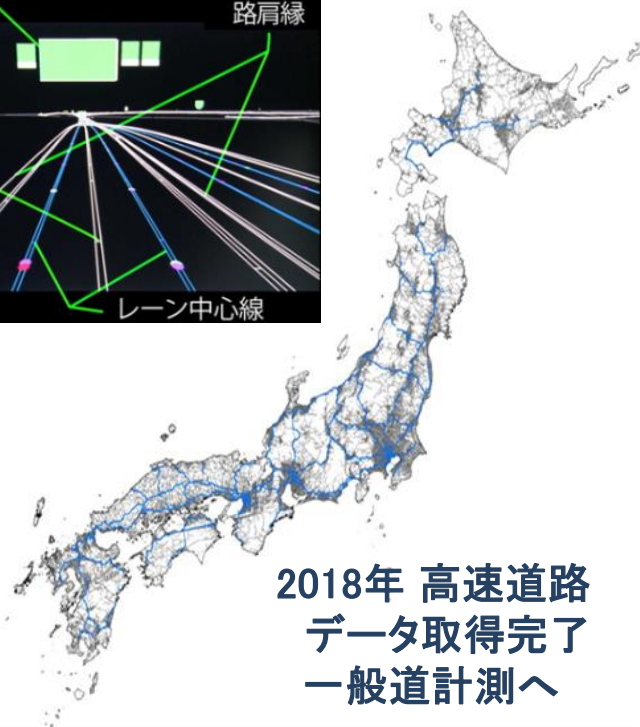
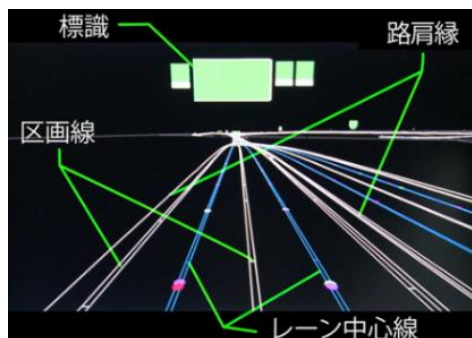
B. GPSのアプリケーション:

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

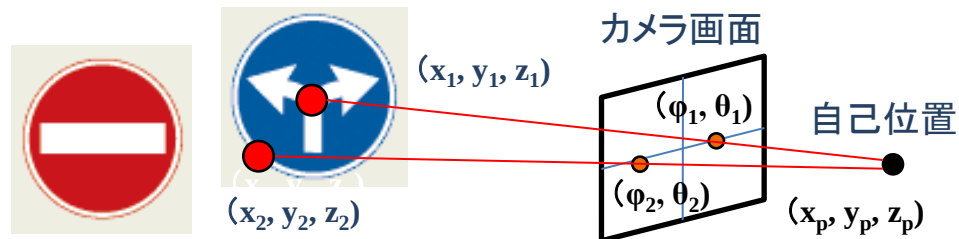


2018年 高速道路
データ取得完了
一般道計測へ

● 標識等を認識してマップと照合し位置を検知

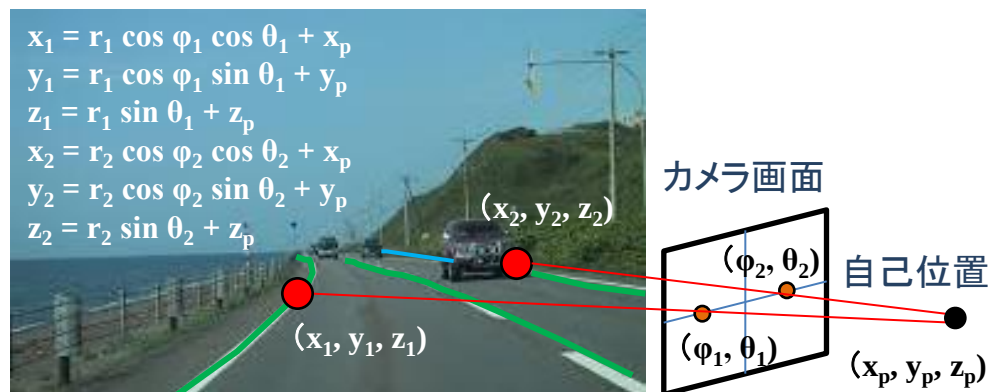
$$\begin{aligned}x_1 &= r_1 \cos \phi_1 \cos \theta_1 + x_p \\y_1 &= r_1 \cos \phi_1 \sin \theta_1 + y_p \\z_1 &= r_1 \sin \theta_1 + z_p\end{aligned}$$

$$\begin{aligned}x_2 &= r_2 \cos \phi_2 \cos \theta_2 + x_p \\y_2 &= r_2 \cos \phi_2 \sin \theta_2 + y_p \\z_2 &= r_2 \sin \theta_2 + z_p\end{aligned}$$



● 白線等を認識してマップと照合し位置を検知

$$\begin{aligned}x_1 &= r_1 \cos \phi_1 \cos \theta_1 + x_p \\y_1 &= r_1 \cos \phi_1 \sin \theta_1 + y_p \\z_1 &= r_1 \sin \theta_1 + z_p \\x_2 &= r_2 \cos \phi_2 \cos \theta_2 + x_p \\y_2 &= r_2 \cos \phi_2 \sin \theta_2 + y_p \\z_2 &= r_2 \sin \theta_2 + z_p\end{aligned}$$



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

7. 終わりに...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

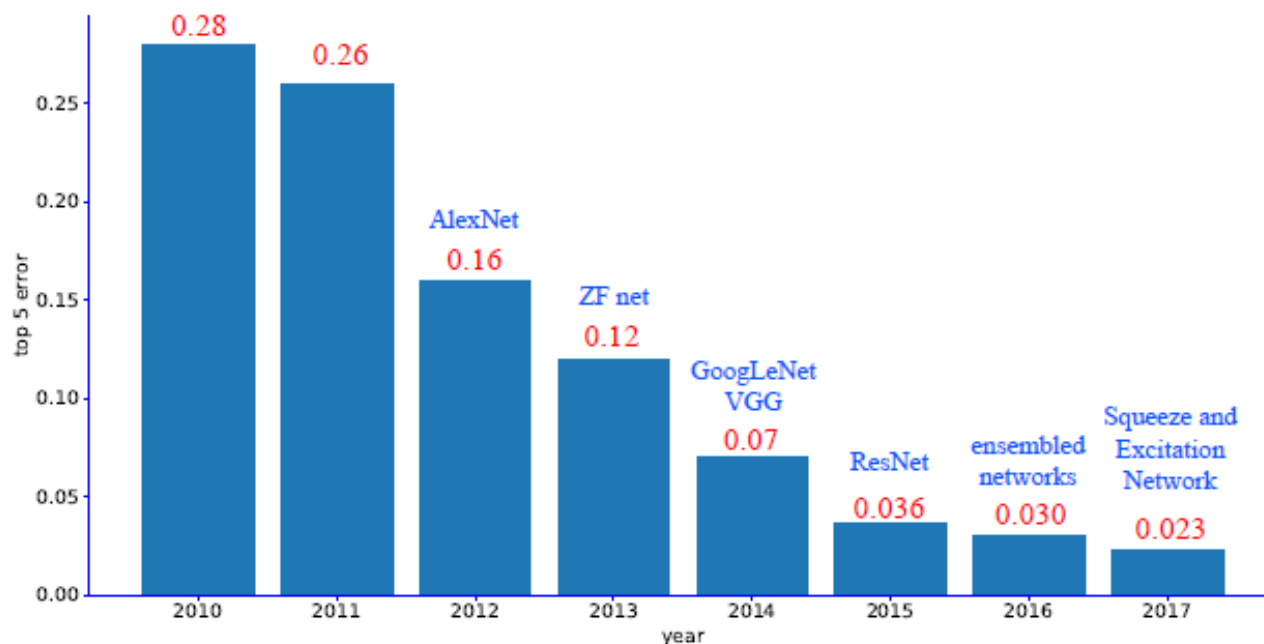


Figure: イメージネットの優勝チームの成績年次変化

フレームワーク毎の GitHub 星数

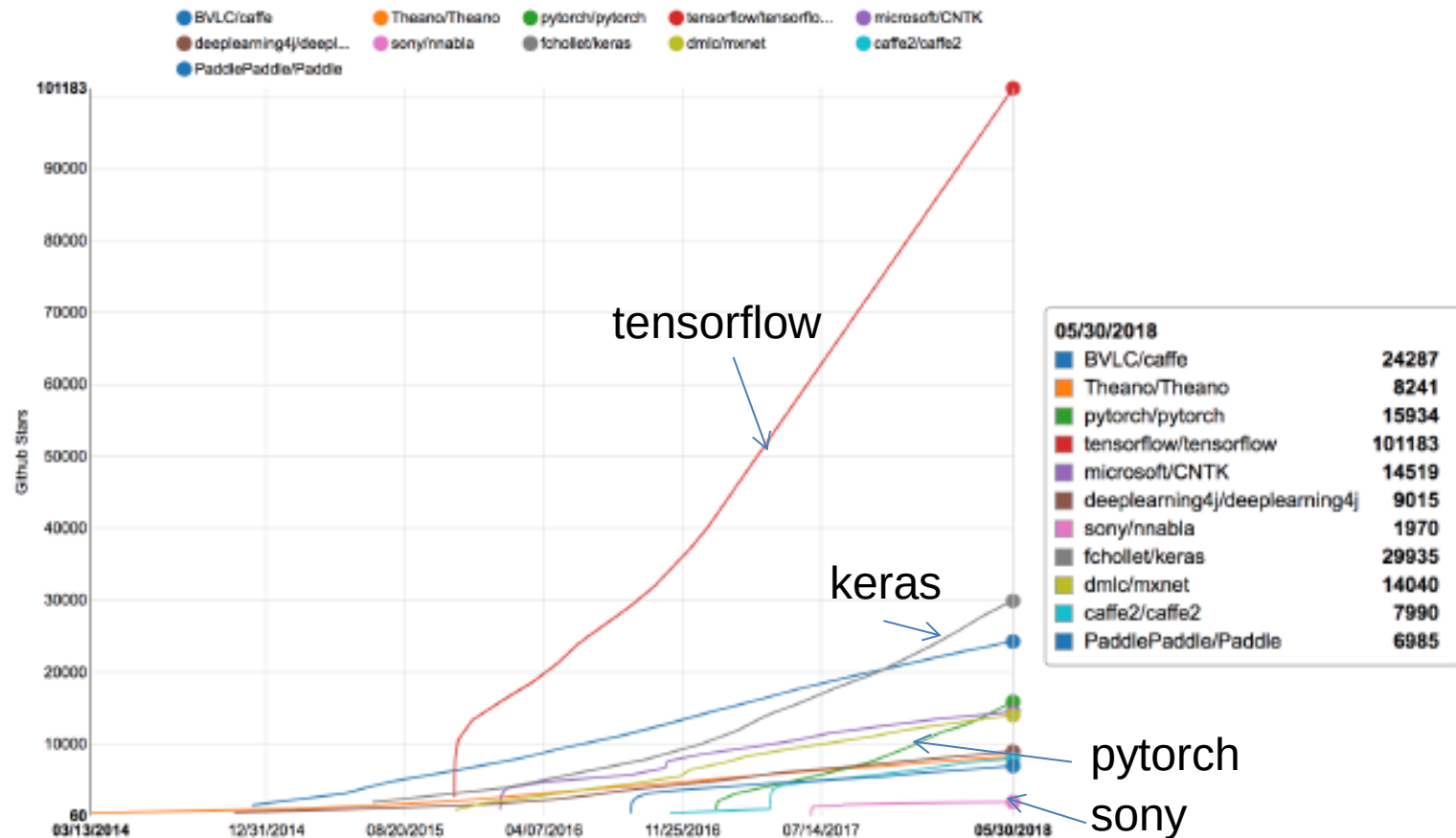


Figure: フレームワーク毎の GitHub 星数の推移¹

<http://jorditorres.org/deep-learning-frameworks/>

フレームワーク毎の GitHub 星数 (2)

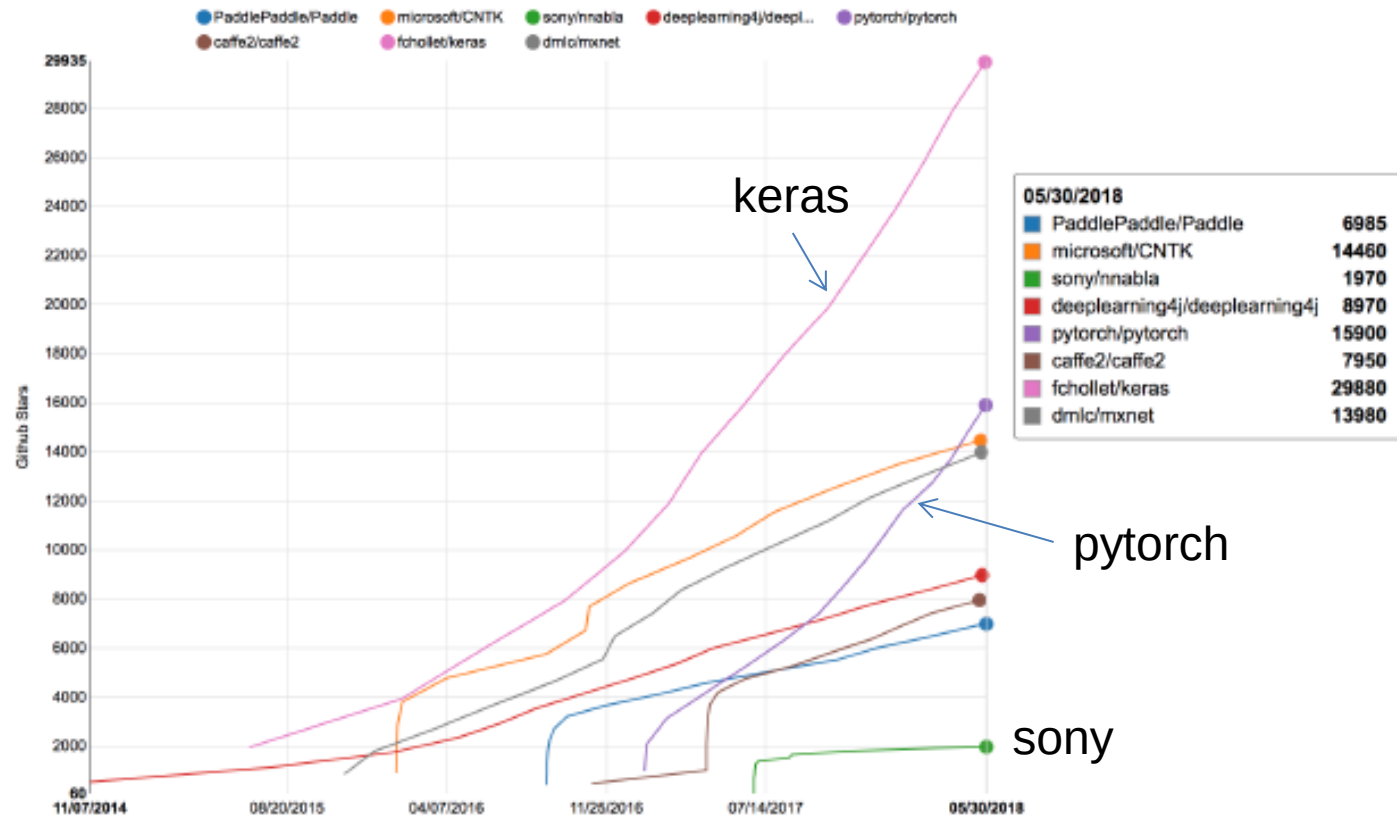


Figure: フレームワーク毎の GitHub 星数の推移 without Tensorflow and Theano

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**を目指すことにすれば、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.

MathWoks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ろう。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.