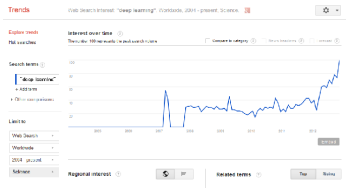
# 情報意味論 (第15回)
# Deep Learning

慶應義塾大学理工学部
櫻井 彰人

## 機械学習に関する最近のバズワード

- Big Data
- Deep Learning いえいえ、バズワードではありません。非常に真っ当な専門用語です



## なぜ注目されるのか

- Deep Learningが各分野のコンペティションで優勝し話題になっています。Deep Learningは7、8段と深いニューラルネットを使う学習手法です。すでに、画像認識、音声認識、最も最近では化合物の活性予測で優勝したり、既存データ・セットでの最高精度を達成しています。

岡野原 大輔氏のブログ
http://research.preferred.jp/2012/11/deep-learning/

## 画像認識では

- ILSVRC 2012 (ImageNet Large Scale Visual Recognition Challenge)

Task 1 (classification)
Task 2 (localization)
Task 3 (fine-grained classification)

| Team name | Error (5 guesses) | Description |
|---|---|---|
| SuperVision | 0.15315 | Using extra training data from ImageNet Fall 2011 release |
| SuperVision | 0.16422 | Using only supplied training data |
| ISI | 0.26172 | Weighted sum of scores from each classifier with SIFT+FV, LBP+FV, GIST+FV, and CSIFT+FV, respectively |

| Team name | Error (5guesses) | Description |
|---|---|---|
| SuperVision | 0.334563 | Using extra training data for classification from ImageNet Fall 2011 release |
| SuperVision | 0.341905 | Using only supplied training data |
| ISI | 0.500342 | Re-ranked DPM detection over Mixed selection from High-Level SVM scores and Baseline Scores, decision is performed by looking at the validation performance |

## 画像認識

- Googleによる巨大な Neural Net を利用した画像認識
- ImageNet の画像データ分類
  - 22,000 categories
  - 14,000,000 images
- 特徴の学習
  - 10,000,000 images (200x200 pixels)
  - 1.15 billion parameters
  - 2000 machines (16000 cores) を一週間

使用したモデルは
deep auto-encoder with pooling and local contrast normalization

## 22000 categories

smoothhound, smoothhound shark, Mustelus mustelus
American smooth dogfish, Mustelus canis
Florida smoothhound, Mustelus norrisi
whitetip shark, reef whitetip shark, Triaenodon obseus
Atlantic spiny dogfish, Squalus acanthias
Pacific spiny dogfish, Squalus suckleyi
hammerhead, hammerhead shark
smooth hammerhead, Sphyrna zygaena
smalleye hammerhead, Sphyrna tudes
shovelhead, bonnethead, bonnet shark, Sphyrna tiburo
angel shark, angelfish, Squatina squatina, monkfish
electric ray, crampfish, numbfish, torpedo
smalltooth sawfish, Pristis pecNnatus
guitarfish
roughtail stingray, Dasyatis centroura
butterfly ray
eagle ray
spotted eagle ray, spotted ray, Aetobatus narinari
cownose ray, cow-nosed ray, Rhinoptera bonasus
manta, manta ray, devilfish
Atlantic manta, Manta birostris
devil ray, Mobula hypostoma
grey skate, gray skate, Raja batis
little skate, Raja erinacea

Sting ray

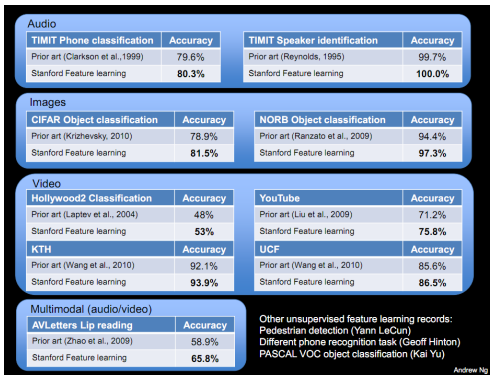
Manta ray

0.005% Random guess
9.5% State-of-the-art (Weston, Bengio '11)
15.8% Feature learning From raw pixels

ImageNet 2009 (10k categories): Best published result: 17% (Sanchez & Perronnin '11), Our method: 20%
Using only 1000 categories, our method > 50%

# 音声認識他

- マイクロソフト
  - Microsoft Audio Video Indexing Service (MAVIS)
    - http://research.microsoft.com/en-us/projects/mavis/
- 音声とテキスト
  - WSJ CSR corpus
    - http://aclweb.org/anthology-new/W/W12/W12-2703.pdf
- 確率言語モデル
    - http://www.gatsby.ucl.ac.uk/~amnih/papers/ncelm.pdf
- 化合物の活性予測コンテスト
  - Merck Molecular Activity Challenge
- residue-residue contact predictor
  - Predicting protein residue-residue contacts using deep networks and boosting
    - Jesse Eickholt and Jianlin Cheng, Bioinformatics (2012)

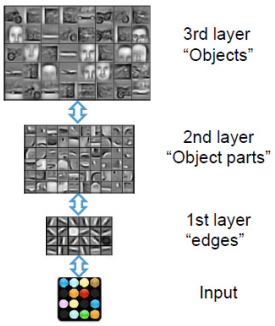



# Deep Learning

- 定義(直接的な定義は見つからないが)
  - Deep learning methods aim at learning feature hierarchies with features from higher levels of the hierarchy formed by the composition of lower level features.

# 大枠

- 特徴量を学習する
  - Hand-craftはしない
- 抽象度が低い特徴から抽象度の高い特徴までを階層的に学習する
  - 抽象度の低い特徴は、類似タスクで利用可能
- 主な手法
  - Deep belief networks (Hinton)
  - Deep autoencoder (Bengio)
  - Deep neural networks
    etc.

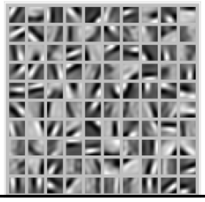



# つまり

- 多層のネットワーク( "shallow" netとは層数が2個ぐらい)を学習させる
- この「多層」により、よりよい(人間が作るよりよい)特徴空間を学習する
  - (画像認識の場合)第一層は、いわば、一次の特徴(エッジ等)を学習する
  - 第二層はより高次の特徴(ex. エッジの連なり)を学習する
  - 初期層は、通常、教師なしで学習する。複数のタスクに使えるような、共通的な、一般的な特徴を学習する
  - 最終層は、通常、教師付き学習を行う
    - ネットワーク全体で教師付き学習を行うこともある(教師なし学習で得た荷重を初期値に用いるわけである)
  - 勿論、完全に(つまり最初から)教師付き学習を行ってもよい(もっともそれがうまくいかないものだから、工夫をすることになった)

# したがって

- 通常は、入力空間(つまり、説明変数値がなす空間)が局所的な構造を持つ場合に、Deep Learning はうまくいく。
  - 局所構造: 空間的・時間的のいずれでも。従って、画像、音声が最適であるが、言語や遺伝子・化合物も適しているであろう。
  - 仮に局所構造があっても、各構造について十分な学習データ数がなければ、Deep Learning はできない
- 局所構造の例: 初期視覚



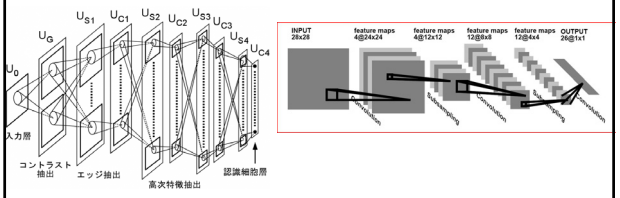
左図は学習例だが、初期視覚において、動物もこれに類した反応をすることが知られている。

# なぜ？

- 生物学的に首肯できる – 例えば、視覚皮質
- Håstad の定理(?): k 層あれば、多項式オーダーのノード数のネットワークで表現できる対象のなかには、k-1層では、ノード数の指数関数オーダーのノード数でないと表現できないものがある（例: パリティ関数）
- 変動が激しい関数は、deep architecture を用いれば、効率よく（つまり少ないノード数で）表現できる場合がある
  - 学習時の更新も、shallowな表現に比べれば、少ない回数・個数の更新で済む
- 特徴量のうち、タスクに共通な（対象には依存する）特徴量は、タスク間で共通に用いることができる。画像認識や音声認識では、多くの特徴はタスク（顔認識、文字認識、一般オブジェクト認識等）独立であろう

# 初期の研究

- Fukushima (1980) – ネオコグニトロン
- LeCun (1989) – Convolutional Neural Networks




# 初期の研究

- 多層パーセプトロン（MLP）
  - 構造はほぼ同じ。違うのは学習方法。
  - PDPの時代から、試みられた（当然！）。しかし
    - 遅い。とにかく遅い。
    - 出力層から離れるほど学習が進まない（diffusion of gradient と言われる）
  - ごく最近の研究では、（初期値が悪くなければ）辛抱強くBPで学習を続けると、deep MLP でも精度の改善が図られることが分かった

(unsupervised) pre-learning

# Deep network の学習

- 特徴空間の構築
  - 機械学習を行うときは、まず、行うこと。しかし、今回は、deep architecture を用いて、特徴（それもよい特徴）を作らせるのである
  - 特徴には、抽象度の違いがある。抽象度の低い特徴が学習できたら、それをもとに、より抽象度の高い特徴が学習できないか？

*Lee et al. ICML'2009*

# Deep network の学習

- Deep network を教師付き学習させることには困難が伴う
  - MLP の初期層（入力に近い層）の学習は進まない
    - Gradient（結合荷重の修正量を計算するもと）が（逆伝播の途中で）拡散してしまい、初期層には届きにくい
    - その結果、学習が非常に遅くなる
    - 直観的には：出力に近い層は、一般にどんなタスクもそこそこなすだけの学習力がある。そのため、出力に近い層が少しでも学習してしまうと、初期層に戻すエラーの量（つまり修正すべき量）が急速に減少してしまう。
    - credit assignment 問題を解決していることは間違いないのだが、局所最適解がたくさんあることとあいまって、「出力に近い層が学習してしまい、そこそこの正しさに満足して、初期層の方に修正を要求する必要が小さくなってしまう」というわけである。
    - 何らかの方法によって、初期層の学習をさせる必要がある
  - （教師付きデータが膨大にあれば学習させることもできようが）教師付きデータはコスト高であるため、十分な個数があることはまれである。
    - 教師なし、または、半教師付き学習を行うことはできないか？
  - Deep networks は、shallow なものに比べ、局所最適解の個数が多いことが推測される

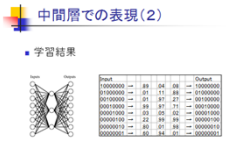
# Greedy かつ層ごとに学習を行う

- 一つの方法は、greedy かつ layer-wise の学習

1. 第一層を教師なし学習させる
   - 教師付き学習も可能であるが、通常はラベルなしデータを用いる。
2. 次に、第一層のパラメータは固定して、第二層の学習を行う。第一層の出力をラベルなしデータとして再び教師なし学習を行う
3. 必要なだけ、上記の学習を繰り返す
   - 抽象度の階層をもった、特徴量が得られる
4. 最後の層の出力を、教師付き学習を行う層（NNではない他の学習モデルでもよい）の入力として、学習を行う（他の層のパラメータは固定しておく）
5. 微調整を行うこともできる。すなわち、全層のパラメータを対象に学習を行う

## auto-encoder

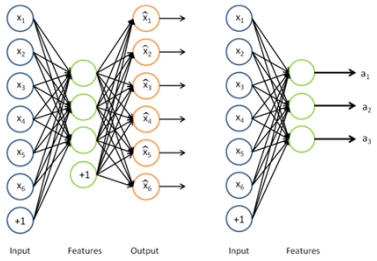
- データ中の特徴を見出す方法の一つ
  - 恒等関数を学習させる（砂時計型のNNです）。
    - 実際には、砂時計型にしなくても、よいことが知られている。
  - 情報圧縮である




## auto-encoder



## auto-encoder




## sparse encoder

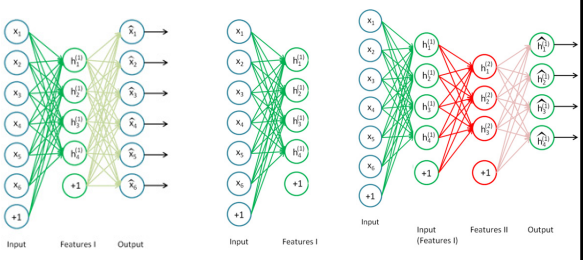
- Auto-encoder は一般に次元削減（dimensionality reduction）を行う
  - PCA-類似である。ただし、非線形な次元削減である
- この結果、"dense" な表現が得られる。これはこれで、目的にかなった結果である
  - 全ての特徴は一般にnon-zeroな値を持ち、その組み合わせは入力値に関する情報を十分に持っている（そして、変数の数は少ない）
- しかし、この分散表現は絡み合っており（特徴量間に何らかの意味での相関がある）、後段の auto-encoder で抽象度の高い特徴を得るのが難しくなる
- "sparse" な表現が得られれば、この問題は解決する。"sparse" な表現とは、どの時点でも（どのような入力値に対しても）、ほとんどの特徴の値は0であるような表現。ただ一つだけ non-zero というのは、よくone-hot 表現と言われるが、今回は、それは sparse 過ぎる

## sparse auto-encoder の作り方

- encoder に多くの隠れ素子を配置すればよい
- sparseness を誘導するような正則化項を（損失関数に）付け加える。
  - non-zero ノードの個数に応じて大きくなる penalty 項を入れる
  - Weight decay
  - etc.
- De-noising Auto-Encoder
  - 学習データに、確率的に、ノイズを加える。auto-encoder は（ノイズが加わった入力値ではなく）ノイズを加える前の値を教師信号として、学習させる。データ間の条件付き独立性を強制する方法でもある
  - 実験結果はよい

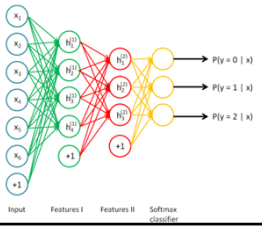
## stacked auto-encoder

- Bengio (2007) – Deep Belief Networks (2006) の後で提案
- sparse auto-encoder を積み上げる。それぞれは、greedy layer-wise に学習させる
- なお、decoder 部分は、廃棄していく（学習時以外使わない）



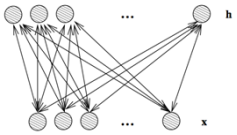
## stacked auto-encoder

- 出力層では、教師付き学習を行う。
- ネットワーク全体に対して、微調整する
- Stacked Auto-Encoders は DBN (Deep Belief Networks) に性能が劣る
  - (De-noising auto-encoders を用いれば, stacked auto-encoders は DBN と同等になる)
  - 統計的モデルとしては、DBN のような生成モデルではない




## Deep Belief Networks (DBN)

- Geoffrey Hinton (2006)
- Greedy layer-wise 学習. ただし、各層は RBM (Restricted Boltzmann Machine)
- RBM はBoltzmann machine に次の制約を加えたもの
  - 同じ層内の結合はない. 層は hidden (**h**) と visible (**x**) node 層
  - 結合は対称
  - annealingは用いない（temperatureはない）. これは、それぞれのRBMでは、大域最適解は求めないからである。むしろ、特徴空間を次々と変換していく
  - 多くの場合、logistic 関数を用いる。他の関数も可能である




## RBMのsamplingと学習

- 初期状態は学習データで
  - example **x** (実数値も可)
- Sampling は前進・後退の繰り返し
  - $P(h_i = 1 | \mathbf{x}) = \text{sigmoid}(W_i\mathbf{x} + c_i) = 1/(1+e^{-\text{net}(h_i)})$ // $c_i$ is hidden node bias
  - $P(x_i = 1 | \mathbf{h}) = \text{sigmoid}(W'_i\mathbf{h} + b_i) = 1/(1+e^{-\text{net}(x_i)})$ // $b_i$ is visible node bias
- Contrastive Divergence (CD-*k*): 次のパラメータを得るのに、本来は、MCMCを十分繰り返す必要があるのだが、それを短いステップ(k steps)で打ち切りなおかつある近似を行う方法
- Boltzmann machine と同様に荷重の更新を行う
- 多くの場合 CD-1 (経験的には十分良い結果が得られる「)
  - 学習係数が小さいので、k を大きくして CD-k を行っても、CD-1 を多く行っても結果に大きな違いはない。
  - なお、bias は残る。理論的には最尤推定にならない。しかし、実際上は影響がないようである。
  - CD-1 は傾き方向が正しければよい。それはたいてい正しい。次に学習係数に従い、荷重の変化量を決める

**RBMupdate**$(\mathbf{x}_1, \epsilon, W, \mathbf{b}, \mathbf{c})$
*This is the RBM update procedure for binomial units. It can easily adapted to other types of units.*
$\mathbf{x}_1$ is a sample from the training distribution for the RBM
$\epsilon$ is a learning rate for the stochastic gradient descent in Contrastive Divergence
$W$ is the RBM weight matrix, of dimension (number of hidden units, number of inputs)
$\mathbf{b}$ is the RBM offset vector for input units
$\mathbf{c}$ is the RBM offset vector for hidden units
Notation: $Q(\mathbf{h}_{2\cdot} = 1|\mathbf{x}_2)$ is the vector with elements $Q(\mathbf{h}_{2i} = 1|\mathbf{x}_2)$

**for all** hidden units $i$ **do**
- compute $Q(\mathbf{h}_{1i} = 1|\mathbf{x}_1)$ (for binomial units, $\text{sigm}(\mathbf{c}_i + \sum_j W_{ij}\mathbf{x}_{1j})$)
- sample $\mathbf{h}_{1i} \in \{0, 1\}$ from $Q(\mathbf{h}_{1i}|\mathbf{x}_1)$

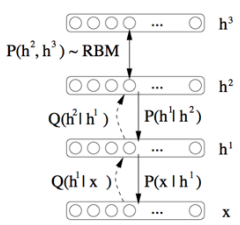
**end for**
**for all** visible units $j$ **do**
- compute $P(\mathbf{x}_{2j} = 1|\mathbf{h}_1)$ (for binomial units, $\text{sigm}(\mathbf{b}_j + \sum_i W_{ij}\mathbf{h}_{1i})$)
- sample $\mathbf{x}_{2j} \in \{0, 1\}$ from $P(\mathbf{x}_{2j} = 1|\mathbf{h}_1)$

**end for**
**for all** hidden units $i$ **do**
- compute $Q(\mathbf{h}_{2i} = 1|\mathbf{x}_2)$ (for binomial units, $\text{sigm}(\mathbf{c}_i + \sum_j W_{ij}\mathbf{x}_{2j})$)

**end for**
- $W \leftarrow W + \epsilon(\mathbf{h}_1\mathbf{x}'_1 - Q(\mathbf{h}_{2\cdot} = 1|\mathbf{x}_2)\mathbf{x}'_2)$
- $\mathbf{b} \leftarrow \mathbf{b} + \epsilon(\mathbf{x}_1 - \mathbf{x}_2)$
- $\mathbf{c} \leftarrow \mathbf{c} + \epsilon(\mathbf{h}_1 - Q(\mathbf{h}_{2\cdot} = 1|\mathbf{x}_2))$

## Deep Belief Network の学習

- Greedy layer-wise
- 最初に初期層 RBM ($h^0 - h^1$) を RBM 更新規則にて更新する
- 荷重を固定して、より上位のRBM層の学習をする
- 次に、出力層を教師付き学習モデルに接続し、学習する
- 最後に, 全ての荷重を自由にし、教師付き学習を行い、微調整を行う




## まとめ

- 発展著しい
  - びっくりするくらい良い結果が出ている
  - 研究者急増中（？）
  - Hinton, LeCun らの地道な研究と Bengio の広い視野
- 技術のポイント
  - 恒等写像を作る。入力情報を（ほぼ可逆）圧縮させる。
  - Sparse coding
  - 階層的化
- 課題
  - もっと広い範囲の特徴量に適用できるのか？
  - 人間の領域知識を組み込むことはできないのか？