

AI (人工知能)

目次

第1章 人工知能概論	
1-1. 従来手法と機械学習は何が違うのか？	14
1-2. 人工知能とは	16
1-3. 人工知能概論の参考文献	18
1-4. 深層学習 Deep Learningとは？	19
1-5. ニューラルネットワークと	20
1-6. 機械学習の例	21
1-7. 人工知能の世間的な分類	26
1-8. 弱い人工知能（AI）と強い人工知能（AI）	27
1-9. 人工知能の利用動向	28
1-10. 今後の展望	29
1-11. 制度的課題：知的財産の観点	30

目次

第2章ニューラルネットワークとディープラーニング

2-1. ニューラルネットワーク	31
2-2. ロジスティック回帰	32
2-3.ディープラーニング（深層学習）とは	33
2-4. 従来の機械学習とディープラーニング の比較	34
2-5. 代表的なディープラーニング	35
2-4. できていることと今後できそうなこと	36

目次

第3章 ディープラーニングの処理	
3-1. ディープラーニングの処理概要	39
3-2. 準備するデータセット	40
3-3. ニューラルネットワーク内の計算	41
3-4. 活性化関数 (Activation Function)	42
3-5. 損失関数 (Loss Function)	44
3-6. One-hotベクトル	45
3-7. 誤差最小化手法	46
3-8. 誤差逆伝播法	50
3-9. バッチ学習とミニバッチ学習	51
3-10. 勾配消失問題	52
3-11. 過学習、過剰適合	53
3-12. 分類の評価指標	54

目次

第4章 Convolutional Neural Network

4-1. Convolutional Neural Network (CNN) の基礎知識	56
4-2. 画像情報の弱点	57
4-3. 畳み込み (Convolution)	58
4-4. プーリング (Pooling)	60
4-5. パディング (Padding)	62
4-6. 畳み込みの意義	63
4-7. 高度なモデル: VGG	64
4-8. 高度なモデル: GAN	65

目次

第5章 Recurrent Neural Network	
5-1. RNN: Recurrent Neural Network68
5-2. LSTM: Long Short-Term Memory69
5-3. LSTMによる予測70
5-4. 時間軸と自然言語74
5-5. Auto Encoder76

目次

第6章 学習画像生成と分類演習

- 6-1. 画像データの増し
(Image augmentation) 79
- 6-2. Kerasによる学習画像生成と画像分類演習 80

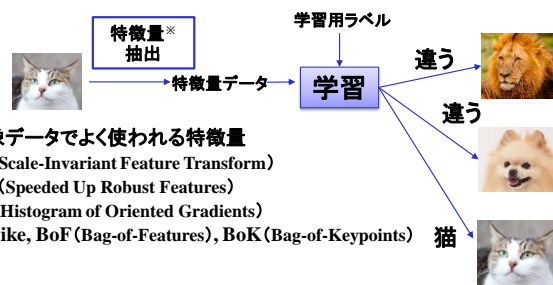
第7章 転移学習

- 7-1. 転移学習 82
- 7-2. 学習済みモデルの利用 84
- 7-3. 転移学習のメリット 89
- 7-4. 転移学習とFine tuning 90
- 7-5. ドメイン適応 91

第1章 人工知能概論

今日の話をざっくり：従来の機械学習

例) 画像の物体認識 (OpenCVなど)



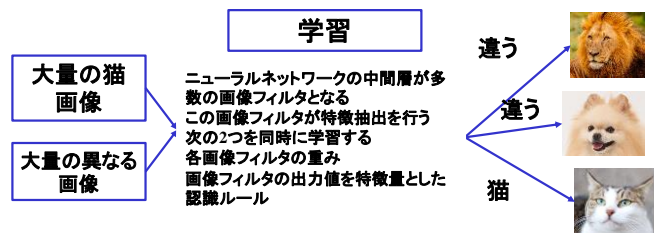
※ 画像データでよく使われる特徴量
SIFT (Scale-Invariant Feature Transform)
SURF (Speeded Up Robust Features)
HOG (Histogram of Oriented Gradients)
Haar-like, BoF (Bag-of-Features), BoK (Bag-of-Keypoints) 猫
etc.

機械学習では、認識すべき対象の着目点(=入力データ)は人間が考える

今日の話をざっくり： Deep Learningの学習

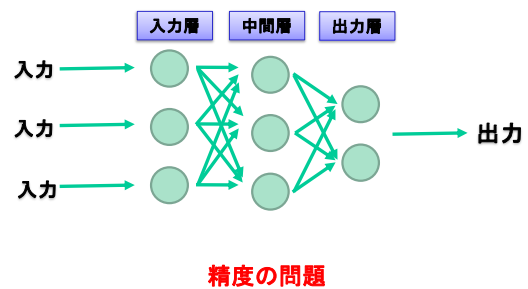
例) 画像の物体認識

大量の正解画像と不正解画像（の画素値）をニューラルネットワークの入力とし、特徴と認識ルールを自動的に学習させる



今日の話をざっくり: 従来のニューラルネットワーク

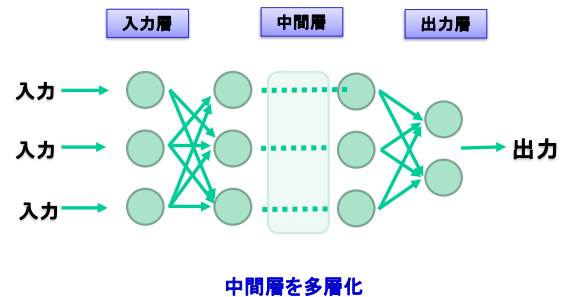
処理能力の限界から、入力層、中間層、出力層の3層構造が多く精度の問題があった



今日の話をざっくり: Deep Learning (深層学習)

中間層を多層化することで精度の問題を解決

ただし、GPUを使用してもやはり処理時間はそれなりにかかってしまう



1-1. 従来手法と機械学習は何が違うのか

従来手法

人間が、特徴量を構築する

機械学習（正確には深層学習、AI）

コンピュータによって、特徴量を自動的に獲得する

機械学習とは？

Arthur Samuel (1959)

"Field of study that gives computers the ability to learn without being explicitly programmed"

明示的にプログラムしなくても学習する能力をコンピュータに与える研究分野



Photo by https://en.wikipedia.org/wiki/Arthur_Samuel

1-2. 人工知能とは

人工知能のFAQ（人工知能学会のサイトより抜粋）

Q. 人工知能とは何でしょうか？

- A. 知的な機械、特に、知的なコンピュータプログラムを作る科学と技術です。人の知能を理解するためにコンピュータを使うことと関係がありますが、自然界の生物が行っている知的手段だけに研究対象を限定することはありません。

Q. では、知能とは何でしょうか？

- A. 知能とは、実際の目標を達成する能力の計算的な部分です。人間、動物、そして機械には、種類や水準がさまざまな知能があります。

Q. AIは人の知能をまねようとしているのではないのですか？

- A. ときにはそうしますが、いつもというわけではありません。あるときは、機械に問題を解決させることについて、他人や自分自身がどうするかを調べます。一方、AIのほとんどの研究は、人間や動物について研究するよりも、知的に解決しなければならない問題そのものについて研究しています。AI研究者は、人間がやらないような方法や、人間ができるよりも多くの計算を伴う方法を用いることもできます。

・説明の流れ

AI、人工知能の学会である人工知能学会で紹介されているFAQです。学術的な定義として紹介します。

AIは何の略ですか？

Artificial Intelligence : 人工的な 知性

教師あり学習

- 人間（教師）があらかじめ用意したデータとパターンを学習し、ここから新しいデータに対して何らかの結果を出力する

教師なし学習

- 与えられたデータやパターンから、何らかの規則性をAI自らが発見し、見いだす

強化学習

- どのような行動を取れば最大限の利益が得られるかを学習する（囲碁やチェスなど）

1-3. 人工知能概論の参考文献

人工知能は人間を超えるか ディープラーニングの先にあるもの：松尾豊、KADOKAWA/中経出版、2015

※日本でもトップレベルのAI研究者による紹介本。内容は若干古いが、非常にわかりやすく、かつコンパクトにまとめられている

AI白書 2017：独立行政法人情報処理推進機構 AI白書編集委員会 編、KADOKAWA、2017

AI白書 2019：独立行政法人情報処理推進機構 AI白書編集委員会 編、KADOKAWA、2018

※国内・国外のAI動向をまとめたもの。松尾先生も委員会に加わっている

・説明の流れ

AI、機械学習に関する書籍は山のように出ていますが、2018年現在、初心者にもわかりやすく、また業界についてコンパクトにまとまっているのがこの2冊です。

特に1冊目は、おそらく日本ではAIでは最先端の東大 松尾先生が書かれたもので、出版年はやや古いものの、まだまだ通用します。

2冊目のAI白書は毎年出版されていて、その年の業界の流れがよくまとまっています。

1-4. 深層学習 Deep Learningとは？

ニューラルネットワークを多層化したもの
概念自体は昔からあったが、莫大な計算コストによって実現不可能
ニューラルネットワークは「冬の時代」

私が大学院の学生だった1997年から2002年ごろには、人工知能の研究をしていると言うと、怪訝な顔をされることが多かった。(中略)なぜなら「人工知能」という言葉自体が、あるいは「人工知能ができる」と主張すること自体が、ある種のタブーとなっていたからだ。

(中略、研究費を獲得する面接で手厳しく批判され)

「あなたたち人工知能研究者は、いつもそうやって嘘をつくんだ」案の定、その提案は落選した。

(「人工知能は人間を超えるか」 はじめに、より)

1-5. ニューラルネットワークとは

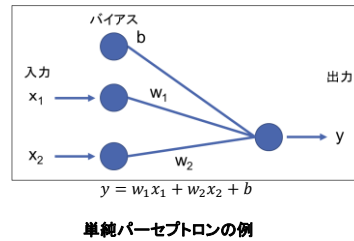
シナプスの結合によりネットワークを形成した人工ニューロン（ノード）が、学習によってシナプスの結合強度を変化させ、問題解決能力を持つようなモデル（Wikipediaより）

教師あり学習、教師なし学習の両方がある

単純パーセプトロン

入力層と出力層の2層のみの
ニューラルネットワーク

入力Xに対して重みとバイアスを用いた
計算により得られた値と教師データY
の誤差を計算し、誤差が小さくなる
ように重みとバイアスを更新して
学習する



ディープラーニングの基本となるモデルです。

このモデルは線形分割しかできないことが知られています。

分類の場合は出力にステップ関数という、しきい値を超えれば1、そうでなければ0を出力する関数を使います。

ディープラーニングはこのモデルを多層にしたもので、多層にすることで非線形な分割が可能となります。

流行したきっかけ: 1

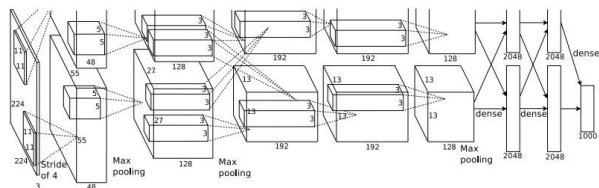
G.H. ヒントン(Toronto Univ., Google)の2006年の論文
“Reducing the Dimensionality of Data with Neural Networks”, G. E.
Hinton, R. R. Salakhutdinov, Science 28 Jul, Vol. 313, Issue 5786,
pp. 504-507,(2006)

要旨：小さな中心層を持つ多層ニューラルネットワークを訓練することによって高次元の入力ベクトルを再構成し、高次元のデータを低次元の符号に変換することができる。このような“Auto Encoder” ネットワークでは、グラディエント降下を使用してウェイトを微調整できるが、これは初期の重みが適切なソリューションに近い場合にのみ効果がある。

本論文では、データの次元性を低減するツールとして主成分分析よりもはるかに良く働く低次元符号をDeep Auto Encoder ネットワークが学習することを可能にする重みを初期化する効果的な方法を述べる。

流行したきっかけ: 2

物体認識率コンテストである IMAGENET Large Scale Visual Recognition Challenge 2012 (ILSVRC2012)において、G.H. ヒントンのチームがDeep Learning手法で従来26%のエラー率を17%まで改善させた



AlexNet: 5つの畳み込み層、3つの全結合層、65万ニューロン

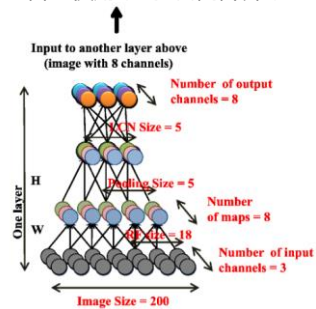
流行したきっかけ: 3

Googleによる猫の「認識」(2012年)

YouTubeにアップロードされている動画から、ランダム抽出した200x200ピクセルサイズの画像を1000万枚用意し、Deep Learningに入力した

3%前後の画像に人間の顔が含まれており、猫が含まれる画像も多数

Googleクラウドの1,000台を使用して3日間計算した



- 22 -

猫の認識ニューロン

ニューラルネットワークで自動的に構築した猫の特徴



このニューロンで選択された猫の画像

<https://googleblog.blogspot.com/2012/06/using-large-scale-brain-simulations-for.html>

1-6. 機械学習の例

k-近傍法（教師あり）

決定木（教師あり）

ランダムフォレスト（教師あり）

自己組織化マップ（教師なし）

サポートベクターマシン（教師あり）

ニューラルネットワーク（教師あり）

遺伝的アルゴリズム

ベイジアンネットワーク

1-7. 人工知能の世間的な分類

レベル1：単純な制御プログラム

- マーケティング的に「人工知能」を名乗っているだけ

レベル2：古典的な人工知能

- 推論、探索、知識ベースなどにより、組み合わせが極端に多い入力と出力を関係付ける方法を定義する

レベル3：機械学習を取り入れた人工知能

- サンプルやデータをもとに入力と出力の関係であるルールや知識を自ら学習する
- 入力は、目的に応じて入力対象の特徴をあらわすもの（特徴量）である必要がある

レベル4：ディープラーニングを取り入れた人工知能

- 特徴量自体を学習する機械学習

※それぞれの手法は後程説明

(松尾豊、「人工知能は人間を超えるか」より)

・説明の流れ

参考文献として紹介した松尾先生の本からです。おそらくこのような講座で紹介するのはレベル3でしょう。

クラウドがはやった時もそうでしたが、マーケティングに都合が良いからといって何でも「AI」「人工知能」を名乗っているのは、レベル1に該当します。

1-8. 弱い人工知能 (AI) と強い人工知能 (AI)

ジョン・サール (アメリカの哲学者) の造語

□ 弱い人工知能

- 人間の全認知能力を必要としない程度の問題解決や推論を行うソフトウェアの実装や研究
- チェスや囲碁など限定された範囲の問題について、一見知的に見える問題解決ができるもの

□ 強い人工知能

- 人間の知能に迫るようになるか、人間の仕事をこなせるようになるか、幅広い知識と何らかの自意識を持つもの

□ 汎用人工知能 (≒強い人工知能)

- 人間レベルの知能を実現するもの
- 限定された問題を解決する特定型人工知能ではなく、一般的な知能を実現するもの

・説明の流れ

昔から、知能に関する研究は哲学や人間科学の面からもいろいろと言及されてきました。ここでは哲学方面からの人工知能に関する考え方を紹介します。

1-9. 人工知能の利用動向

- 人工知能の効果を最大化するためには、質の高い学習用データを基に付加価値を生み出す学習済みモデルを生成することが重要
- 学習済みモデルを初期状態とし再利用することで、比較的少数の学習データから優れた性能を持つ派生データを得ることができる
- 画像認識や音声認識の領域ではすでに実用性の高い技術として応用が進められている
 - 自動走行における車外走行環境認識、医用画像からの疾病等の診断支援など
 - 家庭や自動車内での音声対話や音声アシスタント、工場での異常音検知など

(AI白書2017より)

・説明の流れ

なにかと人工知能を使うことで夢物語のように言われていますが、現在のところこのような研究が進められています。

ここに書かれている内容は代表的な分野(特に自動車など)ですが、コールセンターの応答を機械学習で行うなどもよく報道されることがあり、ご存じの方も多いと思います。(質問してもよいでしょう)

1-10. 今後の展望

海外では検索サービスやSNSなどのインターネット空間での活動から得られるデータに対して適用を進めている

海外企業がすでに圧倒的なシェアを持つインターネット空間を中心とした人工知能利用に、今後対抗することは容易ではない

先行する企業はAIの機能を組み込んだ機械やロボットを普及させることで実空間における消費者との接点も押さえつつある

健康情報、自動車の走行データ、工場の稼働データなど、個人や企業の実世界における活動から得られる実空間データへの適用は今後の競争課題

(AI白書2017より)

・説明の流れ

人工知能の定義にもよりますが、特に海外ではデータを大量に集め、そこから様々な学習を行い、すでにかんりの実用化がされています。

1-11. 制度的課題：知的財産の観点

人工知能生成物の知財制度上の扱い

- 学習済みモデルの利用者の創作的意図が認められない場合は著作物とはみなされない→人工知能の創作物は著作物ではない
- 学習済みモデル利用者に創作的意図と生成物を得るための創作的寄与があれば著作物とみなす

学習用データの活用と保護

- 日本ではインターネット上の著作物を元に学習用データを作成・解析することは営利目的も含めて著作権侵害にならない
- ただし、そのような学習用データの一般への提供は難しい
- 海外では国によって異なるが、学習用データをオープンデータとして提供する活動が盛ん（例：KaggleやGoogleの一連のデータなど）

・説明の流れ

スライドにあるように、日本では現在のところ人工知能に関する著作権が非常にゆるい、天国のような状態になっています。

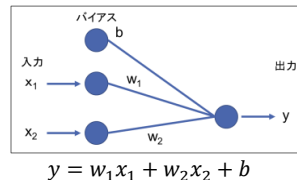
著作権についてここで説明した後、次のスライドにも続きます。

既存の学習済みモデルに学習データを追加してモデルを生成する「蒸留モデル」は元のモデルへの依拠性の認定が難しいため、著作権による保護が困難という状況です。

第2章 ニューラルネットワーク とディープラーニング

2-1. ニューラルネットワーク

- シナプスの結合によりネットワークを形成した人工ニューロン（ノード）が、学習によってシナプスの結合強度を変化させ、問題解決能力を持つようなモデル（Wikipediaより）
- 教師あり学習、教師なし学習の両方がある
- 単純パーセプトロン
 - 入力層と出力層の2層のみのニューラルネットワーク
- 入力Xに対して重みとバイアスを用いた計算により得られた値と教師データYの誤差を計算し、誤差が小さくなるように重みとバイアスを更新して学習する



単純パーセプトロンの例

ディープラーニングの基本となるモデルです。

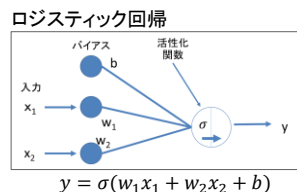
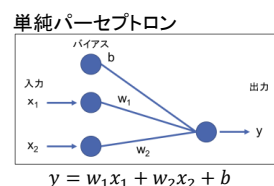
このモデルは線形分割しかできないことが知られています。

分類の場合は出力にステップ関数という、しきい値を超えれば1、そうでなければ0を出力する関数を使います。

ディープラーニングはこのモデルを多層にしたもので、多層にすることで非線形な分割が可能となります。

2-2. ロジスティック回帰

- 単純パーセプトロンのモデルに活性化関数としてシグモイドを使う
- シグモイド関数 $y = \frac{1}{1+e^{-x}}$
- 出力 y は $y = \frac{1}{1+e^{-(w_1x_1+w_2x_2+b)}}$ となる
- $0 \leq y \leq 1$ となるので出力を確率とみなすことができる



・説明の流れ

ここで扱う単純パーセプトロンは、ディープラーニングの基本となるモデルです。このモデルは線形分割しかできないことが知られています。

分類の場合は出力にステップ関数という、しきい値を超えれば1、そうでなければ0を出力する関数を使います。

ディープラーニングはこのモデルを多層にしたもので、多層にすることで非線形な分割が可能となります。

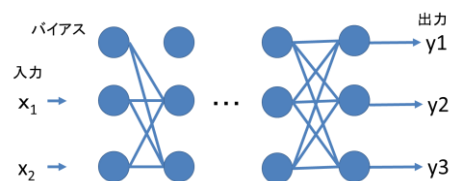
・ポイント

線形分類とは何か、ここで押さえておきます。線形分類とは、2次元平面にデータをプロットしたとき、2分類すると $y=ax$ で表すような直線ではっきりと分類できることを指します。

複雑に入り組んだ非線形分類は、単純パーセプトロンでは分類できません。

2-3. ディープラーニング（深層学習）とは

- 多層のニューラルネットワークによる機械学習手法
- 層の数に特に厳密な定義は無い
 - 入力層と出力層以外に層があればディープと呼ぶこともある
 - 以前から2層の中間層は使われていたので、3層以上の中間層とすることもある



・説明の流れ

単純パーセプトロンとロジスティック回帰のモデルをもう1度振り返り、間の層を増やすとどうなるか、と考えます。

深層学習、ディープラーニング自体にこれという定義はありません。とはいえ中間層があればディープと見なす、というのはやや乱暴で、現在の一般的な深層学習は、64層など、非常に多くの中間層を持っています。

2-4. 従来の機械学習と ディープラーニングの比較

機械学習は入力と出力の間の複雑な関係を学習してモデル化する

- 分類（出力がグループのラベル）
 - 判別、領域抽出など
- 回帰（出力が連続値）
 - 未来予測など

ディープラーニングの利点

- データの前処理（特徴量への変換）が従来の機械学習ほど必要でない
- 従来の機械学習は目的に応じた特徴量を入力としないと精度が上がらないことが多い

ディープラーニングの欠点

- 大量の学習データが必要
 - しかし、近年ではセンサーデバイスの普及により、大量のデータが取得しやすくなった

スライド番号10番を振り返ってみると、人工知能の手法としては、

レベル1:単なる制御

レベル2(推論、探索、エキスパートシステム):人間が手動でルールを作成する

レベル3(機械学習):ルールをデータから学習する。データは単なる座標や計測値ではなく、ルールに適した特徴量が望ましい。

レベル4(ディープラーニング):ルールとルールに適した特徴量をデータから学習する。

と言えます。ただし、ディープラーニングでは大量の学習データと学習時間が必要ですし、得られた特徴量は必ずしも人間が見て理解できるものとは限りません。

2-5. 代表的なディープラーニング

畳み込みニューラルネットワーク

- Convolutional Neural Network、CNN
- 教師あり学習
- 画像データの扱い、認識、分類に強い

リカレントニューラルネットワーク

- Recurrent Neural Network、RNN
- 教師あり学習
- 時系列データの扱い、未来予測や自動作文に強い

オートエンコーダ

- Auto Encoder、AE
- 教師無し学習
- ノイズ除去や画像生成に強い

今回は時間の関係上、オートエンコーダは扱いません。オートエンコーダは、教師なしで次元削減する主成分分析と似ています。

実際、活性化関数に恒等写像を使うと主成分分析と同じ結果となることが知られています。

2-6. できていることと今後できそうなこと

CNN

- できていること
 - 画像認識、画像の説明テキスト生成、運転支援、個人特定
- 今後できそうなこと
 - リアルタイムの検査（現在は処理できる画像の解像度は高くない）
 - 競技の採点
 - 農業など、自然を対象とする仕事の自動化

RNN

- できていること
 - 翻訳、小説作成、電力予測、自動記事作成、
- 今後できそうなこと
 - 話し相手、防犯、教育目的（学習管理）

・説明の流れ

このスライドと次のスライドでは、各ネットワークごとにできていること、できそうなことをまとめています。

略称についてここで覚えておいてください。それぞれ、事例を聞いたことがありますか、と参加者に質問します。

できていることと今後できそうなこと

AE

- できていること
 - 低解像度から高解像度への復元、画像生成（画風模倣、筆跡模倣など）
- できそうなこと
 - 芸術作品の生成
 - 合成音声による声優

その他、

- CNN、RNN、AEの組合せ
- 強化学習
- Generative Adversarial Network（敵対的生成ネットワーク）

など様々な手法が開発されています。

第3章 ディープラーニングの 処理

・説明の流れ

ここからいよいよ深層学習(ディープラーニングに入ります)

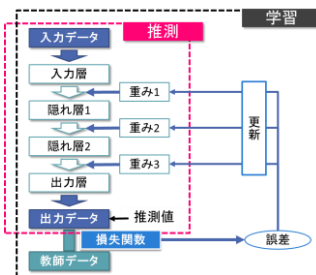
場合によっては、一度ここで時間を取って、ここまでの内容を再度整理してください。

また、単純パーセプトロンやロジスティック回帰といった、単純なニューラルネットワークについても再度確認してください。

ディープラーニングにおける処理の一連の流れを理解する。

3-1. ディープラーニングの処理概要

- 入力をディープニューラルネットワークで計算し、出力を得る
- 出力を推測値として、教師データとの誤差を用いて各層の重みを更新する
- 推測と重みの更新を繰り返し、誤差が少なくなるように重みを適正な値に収束させる



・説明の流れ

ディープニューラルネットワークの定義は何だったでしょう？とここで問いかけ復習します。(明確な定義はないですが)

・ポイント

後ほどの誤差逆伝播法のところで触れますが、重みの更新は層ごとに行います。ここでは流れだけ把握してください。

3-2. 準備するデータセット

- 学習（訓練）用データセット
 - トレーニングデータセット (Training data set)
 - 入力と教師データのセット
- 学習評価用データセット
 - バリデーションデータセット (Validation data set)
 - 入力と教師データのセット
 - 学習（重みの更新）には使わないデータを評価として使う
 - 汎化性能のチェック
- 推測（予測）用データセット
 - テストデータセット (Test data set、Prediction data set)
 - 入力のみ

・説明の流れ

ここから実際にニューラルネットワークを使用して予測を行う処理について説明します。

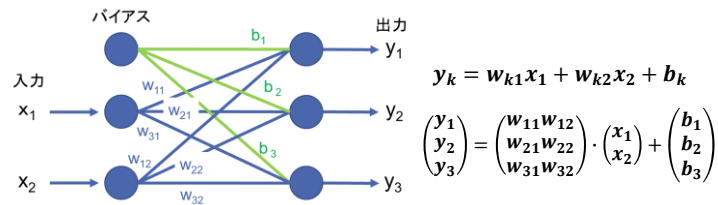
必要なデータとしてこの3つのデータセットがあることを説明してください。

最後の推測データは、実際にニューラルネットワークに入力し、出力として分類や判別を行うデータ(本当に欲しいデータのラベル)のことです。

3-3. ニューラルネットワーク内の計算

下の図は最も基本的な全結合ニューラルネットワーク
(第1層は2ユニット、第2層は3ユニット)

それぞれの y の値は、行列計算として表すことができる

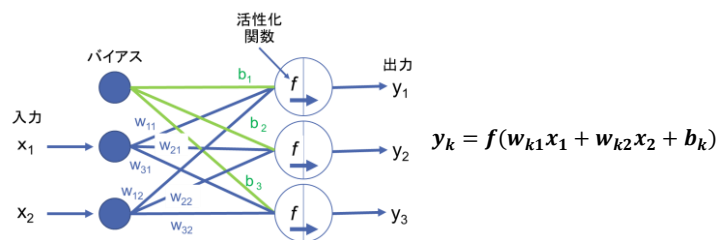


・説明の流れ

スライド43のモデルと比較してください。基本的にスライド31と同じで、ニューラルネットワークの数学的な意味合いを確認しています。

3-4. 活性化関数 (Activation function)

- 各ノードの計算結果に活性化関数で変換して出力とし、次の層の入力にする
- 脳のシナプスがある閾値を超えると発火するという動きを模倣している



・説明の流れ

前スライドの出力に、活性化関数を付け足しました。

活性化関数の種類 (入力をz、出力をyとしたとき)

□ 恒等写像、線形関数

$$\text{> } y = x$$

□ シグモイド関数 (Sigmoid)

$$\text{> } y = \frac{1}{1+e^{-x}}$$

□ tanh関数

$$\text{> } y = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

□ ReLU (Rectified Linear Unit)

$$\text{> } y = \max(x, 0)$$

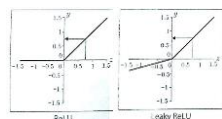
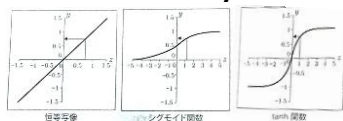
□ Leaky ReLU

$$\text{> } y = x \text{ if } x \geq 0$$

$$y = ax \text{ if } x < 0$$

□ ソフトマックス関数 (Softmax)

$$\text{> } y_i = \frac{e^{x_i}}{\sum_{j=1}^K e^{x_j}} \quad K: \text{出力層のユニット数}$$



(藤田、高原、「実装ディープラーニング」、p.52)

$0 \leq y_i \leq 1$ 、 $\sum_{i=1}^K y_i = 1$ となるため、
確率変数とみなせる

恒等写像、線形関数は入力と出力が同じものです。

シグモイド関数は出力が0～1の範囲となるので、確率とみなすことができます。また、出力は次の層の入力となるので、値が大きくなりすぎることを防ぐことができます。

また、正の値のみをとるので、次の層の合計が0になりにくくなります。ノードでの計算が0になるとその重みの意味が弱くなります。

tanh(ハイパボリックタンジェント)は出力が-1～1となります。2値分類などに適しています。

ReLUはCNNでよく使われる活性化関数で、入力が正の場合は恒等写像、負の場合は0の出力になります。出力

に負の値をとらず、正の値は入力のままなので、
画像の畳み込み計算で画素値が0になりにくく、誤差が消失しにくいという利点があります。

ソフトマックス関数は、主に分類の時に使われます。それぞれの値は0～1で、その層のユニット(ノード)の合計が1になるような出力のため、確率変数とみなすことができます。

そこで、単なる分類ではなく、分類結果の確からしさを表すことができます。

3-5. 損失関数 (Loss Function)

- 推測した出力データと教師データとの間の誤差を計算する関数
- n : サンプル番号、 y_n : 出力データ、 t_n : 教師データとする
- 平均二乗誤差 (Mean Squared Error)

$$E_n = \frac{1}{2}(y_n - t_n)^2, \quad E = \sum_{n=1}^K E_n$$

- クロスエントロピー

$$\square E_n = -\sum_{k=1}^K t_{nk} \log y_{nk}, \quad E = \sum_{n=1}^K E_n$$

t_{nk} : n サンプル目のクラス k の教師データ

y_{nk} : n サンプル目のクラス k の出力データ

- 2値クロスエントロピー

$$E_n = -\{t_n \log y_n + (1 - t_n) \log(1 - y_n)\}, \quad E = \sum_{n=1}^K E_n$$

教師データや予測値の出力が連続値である回帰は平均二乗誤差が使われます。

出力が離散値である分類の場合はクロスエントロピーが使われます。

2値クロスエントロピーは2値分類の場合の損失関数です。

3-6. One-hotベクトル

- ラベルデータのベクトル表現
- 総ラベル数Nの i 番目のラベルを、N次元ベクトルの i 番目の要素だけを1とし、その他は0のベクトルで表す
- 例：総ラベル数5 (0~4) でラベル3は[0、0、0、1、0]となる

- 分類問題の教師ラベル表現として使われる
 - 分類問題では出力層のノード数を総ラベル数にし、Softmax関数を活性化関数とする
 - 教師データをOne-hotベクトルで与えると、正解ラベルに相当するノードの出力が1、それ以外のノードの出力が0に近づくようにノードの重みを更新する
 - Softmax関数を使うので、それぞれの出力ノードの値は、そのラベルに属する確率とみなすことができる

・補足説明

「One-hotベクトル」と呼ぶと難しそうですが、実際は1次元の配列データです。

この例でもありますが、総ラベル数5ならば、以下のようになります。ホワイトボードなどに記述してください。

ラベル0 [1 0 0 0 0]、ラベル1 [0 1 0 0 0]、ラベル2 [0 0 1 0 0]、ラベル3 [0 0 0 1 0]、ラベル4 [0 0 0 0 1]

※0はじまりなので、ラベルは0からです。

また、実際の出力として、ラベル1に近いと判定される場合は、例えば[0.001 0.982 0.123 0.005 0.034] などになります

。

3-7. 誤差最小化手法

- 勾配降下法：
 - 重み w に対する誤差 E の勾配 ΔE (E を w で微分したもの) を求め、誤差 E が小さくなる方向に w を更新する
 - $w = w - \varepsilon \Delta E$ 、 ε : 学習係数 (0.01や0.001など小さな数)
 - 仮に誤差を二乗誤差 $E = \frac{1}{2}(Y - t)^2 = \frac{1}{2}(wX - t)^2$ とすると、
 E の w での微分は $\Delta E = (wX - t)X^T$ となる
 - 誤差をクロスエントロピーとすると、
 $\Delta E = -\sum_{n=1}^K (y_n - t_n) X$
- 重要なことは、勾配 ΔE が入力データ X と誤差信号 δ : $(Y - t)$ から求められること
 - バイアス b についても同様

・説明の流れ

ここから高等数学の話が続きます。難しいようなら、補足説明のように、概念をある程度把握した程度でも大丈夫です。

。

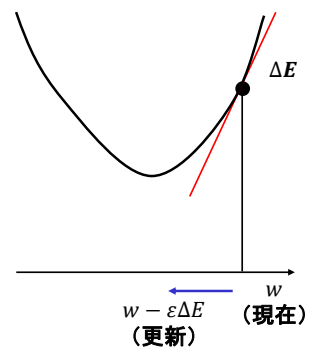
スライド47,48,49は参考程度でよいでしょう。

数学的には、勾配の計算、つまり誤差関数の微分の計算が難しいところですが、

ここでは、スライドの書かれているように、損失関数の勾配が、その時点での出力と教師データの差(誤差信号)と入力データの積で求められることを把握してください。

勾配降下法のイメージ

- あるウェイト w の時点での損失関数の勾配を求め、損失が減少する方向に w を更新する
- 勾配 ΔE が正なら w を減少させ、負なら w を増加させる
- 学習係数 ε が大きい場合
 - 早く収束点に向かう
 - 収束点を通過することがある (収束せずに振動する)
- 学習係数 ε が小さい場合
 - 多くの学習回数が必要
 - 着実に収束点に向かう (収束しやすい)
 - 極小解で収束する場合がある (最小解ではない)
- 多次元でもあり、一概に決めることが難しい



誤差最小化手法

□ 確率的勾配降下法

- 一度に重みを更新する計算をせずに、データを1つずつ（または一部を）ランダムに選んで更新する
- 一部を選ぶ方法をミニバッチ学習と呼ぶ

□ モメンタム（Momentum）を使った更新

- 前回の重み更新の差分 $\Delta w^{(t-1)} = w^{(t)} - w^{(t-1)}$ をつかって、重みの更新を $\Delta w^{(t+1)} = w^{(t)} - \varepsilon \Delta E + \mu \Delta w^{(t-1)}$ とする
- 勾配の正負が逆転する場所でも収束の良いNesterov momentumが良く使われる

誤差最小化手法

□ 学習率（学習係数 ϵ ）の調整法

➤ AdaGrad（学習回数が進むほど g が大きくなり更新量が少なくなる）

$$g \leftarrow g + (\Delta E)^2, \quad w \leftarrow w - \frac{\epsilon}{\sqrt{g}} \Delta E$$

➤ RMSProp

$$g \leftarrow \alpha g + (1 - \alpha)(\Delta E)^2, \quad w \leftarrow w - \frac{\epsilon}{\sqrt{g}} \Delta E$$

➤ Adam（過去の勾配の平均と分散を考慮しながら重みを更新する）

$$m \leftarrow \beta_1 m + (1 - \beta_1) \Delta E, \quad v \leftarrow \beta_2 v + (1 - \beta_2)(\Delta E)^2$$

m は重みの平均値、 v は重みの分散

$$\bar{m} \leftarrow \frac{m}{1 - \beta_1^n}, \quad \bar{v} \leftarrow \frac{v}{1 - \beta_2^n}$$

$$w \leftarrow w - \epsilon \frac{\bar{m}}{\sqrt{\bar{v}}}, \quad \beta_1 \text{と} \beta_2 \text{は} 0 \sim 1$$

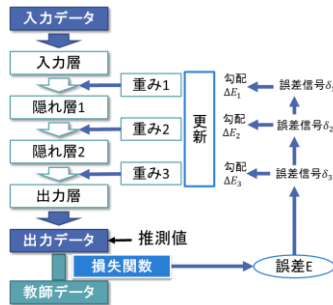
・補足説明

誤差最小化手法として、どういったデータでどの手法を使用する、という明確決まりはなく、経験的に決めているのが現状です。

とりあえず最初はAdamから始めるのが無難とされています。演習で使用するSony NNCでもデフォルトはAdamです。

時系列データ(NNCでは扱えません)にはRMSPropがよいとされていますが、これも決定的ではありません。

3-8. 誤差逆伝播法



多層のニューラルネットワークでは誤差を伝播させて各層で重みを更新する

勾配は誤差信号 δ と入力から求められ、誤差信号は次のように求められる

$$\delta_3 = (Y - t) \circ f'_3(Z_3)$$

f' は活性化関数の微分

$$\delta_2 = ((W_3)^T \delta_3) \circ f'_2(Z_2)$$

$$\delta_1 = ((W_2)^T \delta_2) \circ f'_1(Z_1)$$

$$\Delta E_3 = \delta_3 X_2^T$$

$$\Delta E_2 = \delta_2 X_1^T$$

$$\Delta E_1 = \delta_1 X_0^T$$

誤差信号が逆方向に伝播しながら勾配を求めるようになっている

○はアダマール積で、同じサイズの2つの行列の要素ごとの積

現在の重みに対してニューラルネットワークの計算を行い最終的な出力を得て、

出力層と教師データとの誤差(スライドの δ_3)を求め、 δ_3 と出力層への入力 X_2^T から勾配を求め、

出力層の重みを、誤差が小さくなるように更新します。

スライドの隠れ層2の出力(Z_2)と出力層の重み(W_3)と出力層と教師データの誤差(δ_3)から

隠れ層2における誤差(δ_2)が得られるので、 δ_2 と隠れ層2への入力 X_1^T から勾配を求め、

隠れ層2の重みを誤差が小さくなるように更新します。

これを繰り返して全ての層(入力層は除く)の重みを更新

します。

3-9. バッチ学習とミニバッチ学習

- バッチ学習
 - 用意した学習データや学習評価データを一度に使うて行列計算を行う
- ミニバッチ学習
 - データを小分けにして学習を繰り返す
- 行列計算はデータをGPUに展開して計算を行うので、一度に計算できるデータ量はGPUのメモリに依存する
 - ミニバッチのサイズを大きくすれば学習が速くなるが、GPUのメモリ容量に応じてミニバッチのサイズを調整する必要がある
- 全サンプルの学習を1回終えることを1エポック (Epoch) と呼ぶ

・説明の流れ

一気にデータを投入して学習する方法をバッチ学習、小分けにして学習する方法をミニバッチ学習と呼びます。

・補足説明

コンシューマレベルのGPUだと数ギガ程度のGPUメモリを搭載していますが、このサイズは案外小さく、画像データを大量に投入するとメモリがオーバーフローしてしまうため、ミニバッチで調整する必要があります。

3-10. 勾配消失問題

- 誤差逆伝播法では、誤差が出力層側から入力層側に伝播しながら各層の重みを更新する
- 活性化関数によっては、活性化関数の微分が小さな値になり、それが層を重ねることで誤差がほぼ0になる
 - シグモイド関数の微分の最大値は0.25
 - 3層分逆伝播すると3層目は 0.25^3 となり、最初の誤差の約0.016倍となる
- ReLUを使うことでこの問題を軽減することができる
 - ReLUは入力が正の時の微分は常に1

誤差逆伝播法で触れたように、重みの更新で各層における誤差信号を求めるために、活性化関数の微分を複数回掛け合わせます。

そのため、活性化関数の微分値の絶対値が1未満の場合、誤差が大きくても誤差信号が小さくなることとなります。

微分値の絶対値が1以上になる活性化関数を使うことでこの問題を軽減できますが、大きすぎると収束しなくなります。

3-11. 過学習、過剰適合

- ディープラーニングでは、層やユニットが増えるとパラメータ数が増え、学習データの情報を全て保持できる
 - 学習データ自体の推測はほぼ100%にできる
- しかし、学習データ以外のデータは未知のデータなので推測精度が悪くなる場合がある
 - 学習データに特化したパラメータになる
- 過学習を抑える方法
 - 学習評価データを使ったエポック数の設定
 - 正則化
 - ✓ 学習時のパラメータの更新幅に制約をつける
 - ドロップアウト
 - ✓ 学習時にランダムにユニットを無効化してネットワークの構造を変える

・説明の流れ

過学習と過剰適合は、ディープラーニングに関わらず、機械学習には常につきまとう問題です。

学習データばかりに適合するようになると、他のデータにはまったく適合しなくなってしまい、結局悪いモデルになってしまいます。

様々な手法がありますが、解析的に解ける問題ではないので、試行錯誤や経験則が重要になります。

3-12. 分類の評価指標

Elapsed: 00:00:00:01 Remaining: 00:00:00:00

教師データ

予測結果

Confusion Matrix

Recall(再現率): $\frac{A}{A+B}、\frac{D}{C+D}$

Precision(適合率): $\frac{A}{A+C}、\frac{D}{B+D}$

	y'=0	y'=1	Recall
y=0	50 A	0 B	1
y=1	0 C	50 D	1
Precision	1	1	
F-Measures	1	1	

F-Measure(F尺度):
RecallとPrecisionの調和平均
(逆数の算術平均の逆数)

$\frac{2 \cdot \text{Recall} \cdot \text{Precision}}{\text{Recall} + \text{Precision}}$

・説明の流れ

これは、演習で使用するNeural Network Consoleでの出力結果です。ここでは教師データが0であるものが50あり、実際に0として予測された結果が50なので、100%の再現率です。1についても同様です。ここではたまたますべてうまくいっていますが、もちろん常にこうなるとは限りません。

一般に再現率や適合率が限りなく1(100%)に近くなることを目指しますが、どの指標を参考にするのかは、目的によって異なります。

確実なものだけを見つけないのか？

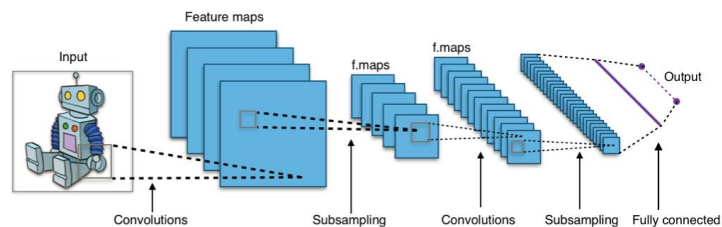
あやしいものは全てを見つけないのか？

第4章 Convolutional Neural Network

画像を入力とするモデルで主に使用される
CNN(Convolutional Neural Network)、畳み込み及びプー
リングの概念について理解する。

4-1. Convolutional Neural Network (CNN) の基礎知識

□ CNNの一般的な構造



https://en.wikipedia.org/wiki/Convolutional_neural_network

□ 画像データの処理

- 畳み込み (Convolution)
- プーリング (Pooling) (Subsamplingの一種)

・説明の流れ

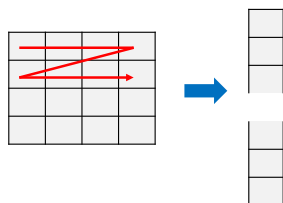
ディープラーニングの中でもよく使用されているのが畳み込みネットワークであるCNNです。特に画像データの分類によく使用されます。

この図はWikipediaからの引用ですが、特徴マップ (Feature map) を複数作成するのが大きな特徴です。

特徴マップを作る時に、畳み込みとプーリングという処理を行います。

4-2. 画像情報の弱点

ニューラルネットワークへの入力は一hotベクトルの形にするため、2次元の画像情報が1次元になってしまう → 画像の特徴が失われる
特徴を活かす形で入力できないといけない → 解決策の1つがCNN



4-3. 畳み込み (Convolution)

- 画像に小さなフィルターをかけ、その値をピクセル値として新たな画像を生成する
- 参考：
<https://ja.wikipedia.org/wiki/%E7%95%B3%E3%81%BF%E8%BE%BC%E3%81%BF> (Wikipedia: 畳み込み)
- フィルターの重みを学習によって変化させることで、特徴を表す画像を生成する
 - 畳み込みで生成された画像は特徴マップとも呼ばれる
- 後述するパディング、ストライドの大きさによって生成される画像のサイズが異なる

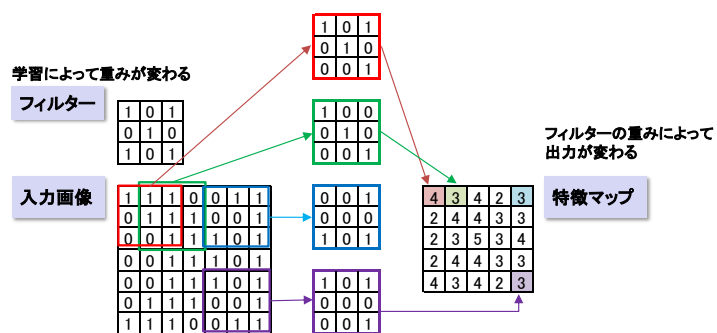
・説明の流れ

画像にフィルターをかけることで、先ほどの図のように特徴マップを作ります。

その方法について概要はスライドの通りですが、細かく見ていきます。

畳み込み (Convolution)

- 入力データに対してフィルターをかけることで、特徴マップを作成する
- 例：カーネルサイズ：3×3、ストライド：横1、縦1



- 5 -

CSAJ

・説明の流れ

画像処理が未経験の人にはわかりにくいかもしれないので、丁寧に説明してください。

入力画像は、単純化のために0,1で表記していますが、これは画素の値です。CNNではモノクロ画像を扱うことが多いため、最も単純な画像だと2値モノクロになります。

そこに、フィルターをかけていきます。このフィルターの大きさがカーネルサイズです。

入力画像のうち、最初のカーネルサイズ分にフィルターをかけます。このとき、フィルターで1の部分のみ、元の画像の値を残します。フィルターをかけた結果が一番上の赤で囲った3x3のマスですが、この結果をすべて足し合わせて、特徴マップの1分とします。そのため、最初の3x3分は、結果として4になります。

次に、フィルターをストライド分ずらし、同じ処理を行います。これを画像の全領域に対して行った結果が特徴マップです。

4-4. プーリング (Pooling)

- サブサンプリングの一種
- ある領域ごとに代表値を求め、その値をピクセル値として新たに画像を生成する
- 結果として画像サイズが小さくなる
 - Max pooling : 最大値を代表値とする
 - Average pooling : 平均値を代表値とする
 - Sum pooling : 合計値を代表値とする
- 領域の大きさによって、縮小率が変わる
 - 領域が 2×2 、ストライド (後述) が2の場合、サイズは縦 $1/2$ 、横 $1/2$ の $1/4$ になる。

・説明の流れ

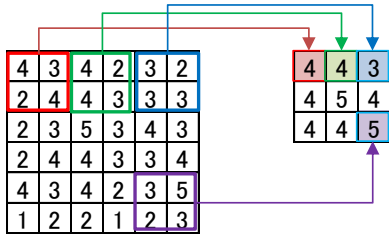
畳み込みの次に行われるのがプーリング処理です。

プーリングは、特徴マップの特徴を残しつつさらに小さな画像を生成する処理です。元の特徴を残しつつ画像を小さくすることで、次元削減になり、計算量が削減されます。

プーリングの方法として、ある領域ごとに最大値を取る (Max pooling)か、平均値を取る (Average pooling)か、合計値を取る (Sum pooling)か、などがあります。

プーリング (Pooling)

- Max poolingの例
- ウィンドウサイズ (カーネルサイズ) : 2×2
- スライド : 横2、縦2



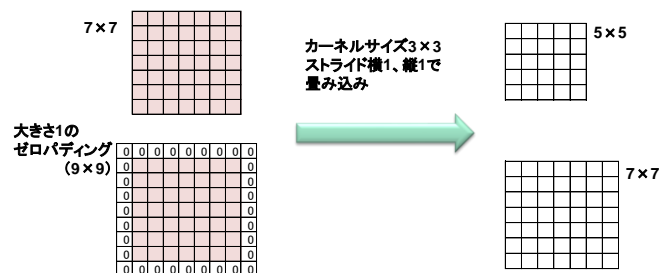
・説明の流れ

ここでは最大値を取るMax poolingの例を取り上げています。

畳み込みと同じように、カーネルサイズを決め、その領域内の最大値を取り出します。1つ終わると、ストライド分ずらし、また同じ処理を繰り返します。

4-5. パディング (Padding)

- 畳み込みを行うと、生成される画像サイズは、元の画像よりも小さくなる
- 元の画像の外側を仮に何らかの値で埋め「ふち」をつくることで、生成される画像サイズを調整する
- 0で「ふち」を埋めるゼロパディングが良く使われる



・説明の流れ

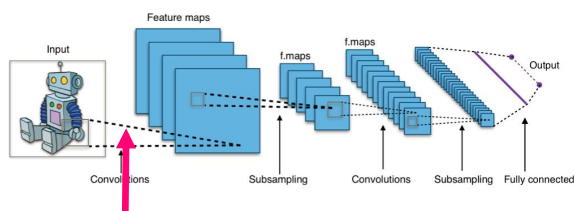
畳み込みの時に、画像サイズを過度に小さくしたくない場合にパディングを使用します。

基本的に0でふちを言うなれば無理矢理作り出してサイズ調整を行います。

特に、畳み込み層が多すぎると、最終的には画像が1x1になってしまいます。そのため、必要に応じてパディングを行い、過度な縮小を防ぎます。

4-6. 畳み込みの意義

- 入力画像から目的にマッチした特徴量を学習によって生成する
 - 畳み込みフィルターの重みを学習によって決定する
- 中間層が全結合ではないので、計算量が少なくなる



もし全結合にすると、入力画素数と特徴マップの画素数×特徴マップ数の結合が必要

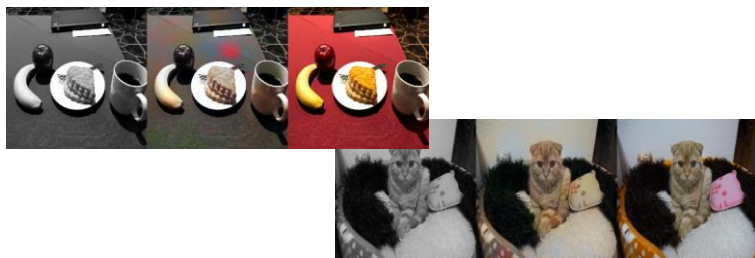
https://en.wikipedia.org/wiki/Convolutional_neural_network

・説明の流れ

ここで、これまで説明した畳み込み、特徴マップ、プーリングなどを再度説明して整理してください。

4-7. 高度なモデル: VGG

Visual Geometry Groupチームによる、畳み込み13層+全結合層3層=16層のニューラルネットワーク→VGG-16、他に19層のVGG-19など
物体検出などに使用されたが、最も有名なのが、Automatic Image Colorization



4-8. 高度なモデル: GAN

Generative Adversarial Networks: 敵対的生成ネットワーク
教師なし学習であり、入力データから特徴を学習し、実在しないデータを生成したり、存在するデータの特徴に沿ってデータを変換できるのが特徴

既存の顔から存在しない顔や表情の合成
動物同士の合成 (シマウマ柄の馬など)

感情をさらに付与する研究

感情推定を利用した感性的な画像説明文自動生成システム

(三由 裕也, 萩原 将文, 日本感性工学会論文誌 18 巻 (2019) 2 号)

要旨：画像中に人が存在する場合と、存在しない場合で2通りの感情推定を行う。画像中に人存在する場合は画像中の人の表情から感情推定を行い、存在しない場合は、画像の背景となる風景画像から感情推定を行う。画像中に人が存在する場合は直喩表現を生成し、画像中に人が存在しない場合は 隠喩の擬人化表現を生成する



a man in a suit and a tie
happily as if he were
success at business.



a cat sitting on the back
of a car disgustingly.

disgustingly: うんざりする

第5章 RNN, LSTM, AE

第6章は紹介のお話だけです。

5-1. RNN: Recurrent Neural Network

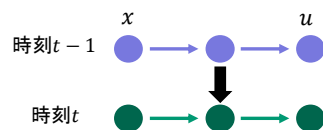
再帰型ニューラルネットワーク

時系列データ予測や画像生成、自動作曲、自然言語処理などに用いられている

時間の概念をニューラルネットワークに取り入れる

時刻 t における入力 $x(t)$ の他に、時刻 $t-1$ における中間層の出力 $u(t-1)$ を保持しておき、 $x(t)$ とともに中間層への入力とする

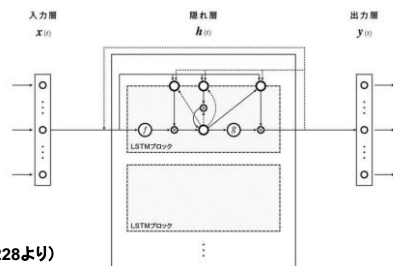
過去の間層の出力 $u(t-1)$ には、再帰的に過去の状態が全て反映されていることになる



5-2. LSTM: Long Short-Term Memory

RNNの拡張型、短期記憶(short-term memory)を長期(Long)に渡って活用するニューラルネットワーク

RNNの中間層のユニットをLSTM blockと呼ばれるメモリと3つのゲートを持つブロックに置き換えたもの



(奥籠, “詳解ディープラーニング”, p.228より)

・説明の流れ

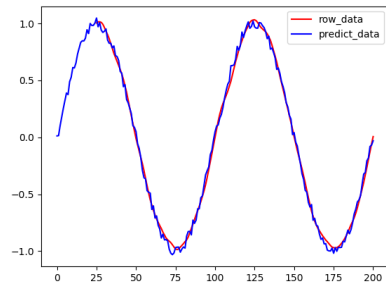
LSTMはRNNの拡張型で、その名の通り長短期記憶(Long Short-Time Memory)に特徴があります。

RNNは短期記憶のみで、長期記憶を利用すると勾配消失問題が発生しましたが、「記憶を忘れる」ための忘却ゲートを導入してこれを抑えるようにうまく工夫されています

。

5-3. LSTMによる予測

三角関数の予測
気温の予測



気温の予測

気象庁の公開している過去のデータを使用

訓練用時系列データ：1975/1/1～1975/12/31

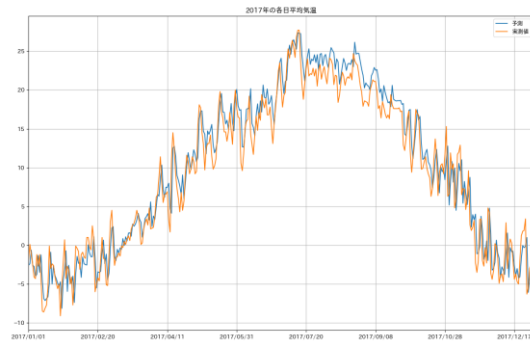
教師信号の正解データ：1976/1/1

この組み合わせを1セットとし、日付を一日ずらして次の1セット、これを繰り返して学習用のデータセットを作成

<https://qiita.com/nvtomo1029/items/689c0a19880d1dc41d43> より

気温の予測: 結果

一見よさそうだが……



気温の予測: 予測と実際の差異

ところどころ
差が大きい

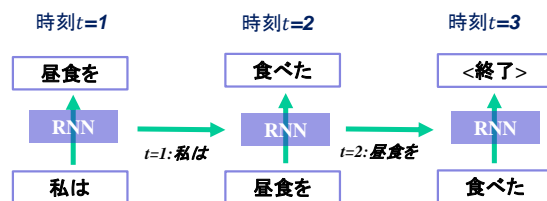
結果：
平均 1.9度
最小 0.0度
最大 7.3度



5-4. 時間軸と自然言語

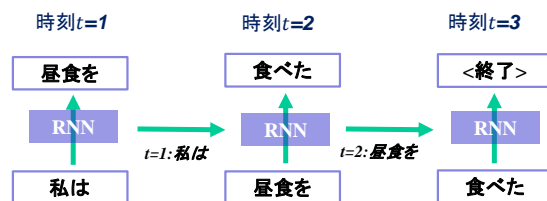
RNNは時系列データであれば何でも扱える。例) 株価
自然言語 (文章) もまた、1種の時系列データである

例) 私は昼食を食べた → 文節分解 → 私は/昼食を/食べた
これをRNNに置き換えると



RNNの自然言語処理

自然言語はあるルールに基づいた時系列データとみなせる
文章を大量に学習することで、私は→昼食を、となるようなRNNの重み学習を行う

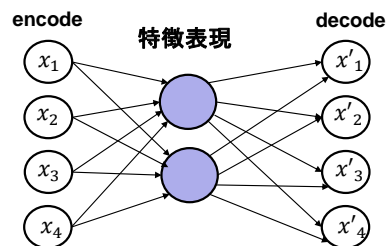


5-5. Auto Encoder

AE: 自己符号化器

教師なし（訓練データなし）のニューラルネットワーク

入力されたデータの次元削減を行い、コンパクトな特徴表現を行うのが特徴



Auto Encoderの応用

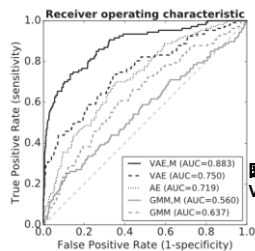
教師データがない = ラベル付けがない

製造業からの需要：ラベル付けのないデータからの異常検知

応用事例：深層生成モデルによる非正規化異常度を用いた工業製品の異常検知

立花亮介(神戸大)他, 人工知能学会全国大会2018年

AEをさらに改良したVariational AEを用いて、ネジ画像データセットを入力し、異常なネジを検出



図は発表予稿より
VAEの手法が最も検出率が高い

第6章 学習画像生成と 分類演習

学習データの増強 (Image Augmentation) について理解する。

6-1. 画像データの水増し (Image augmentation)

- 学習には、画像内の対象の差異がある大量の画像データが必要
- 対象の差異の種類 (手書き数字の場合)
 - 文字の形
 - 画像内での位置
 - 画像内での文字が占める割合 (サイズ)
 - アスペクト比
 - 回転、反転
 - その他変形
 - 明るさ、コントラスト、彩度
- 傾きやサイズ、アスペクト比などが異なる学習データが無い場合精度が落ちる
- 画像処理で生成できるものも多い
 - 一つの画像に様々な画像処理を組合せて適用し、学習データを増やす
 - 予測の安定性につながる

・説明の流れ

ディープラーニングで画像の学習を行うには大量の画像が必要です。数百枚程度ではまったく少なく、数万枚は必要だと言われています。

しかし、十分な画像数を用意できないときにはどうすればよいか。そこで役立つのが、これから紹介する画像の水増しです。

ここに挙げたように、画像を少し回転させたり、拡大、縮小させるなどして、少しずつ異なった画像を生成し、数を増やします。

6-2. Kerasによる学習画像生成と画像分類演習

- 内容は講座当日に配布する別紙を参照してください

第7章 転移学習

7-1. 転移学習

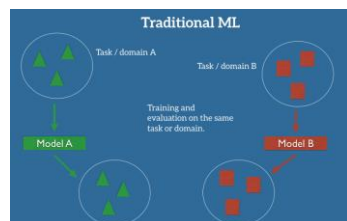
- ディープラーニングに限らず、幅広い機械学習の枠組みに対して使われる言葉
- あるタスクを効果的かつ効率的に解決するために、別のタスクで学習された知識を得て、それを適用する問題
 - “the problem of retaining and applying the knowledge learned in one or more tasks to efficiently develop an effective hypothesis for a new task.”
- 転移学習のワークショップ (NIPS 2005 Workshop - Inductive Transfer: 10 Years Later) のCFP中の定義

説明委の流れ

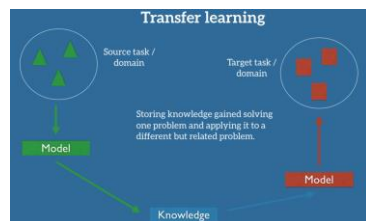
転移学習のメリットを理解し、機械学習のなかでの有効性を知る。

転移学習についての概要を解説する。

- 従来の機械学習



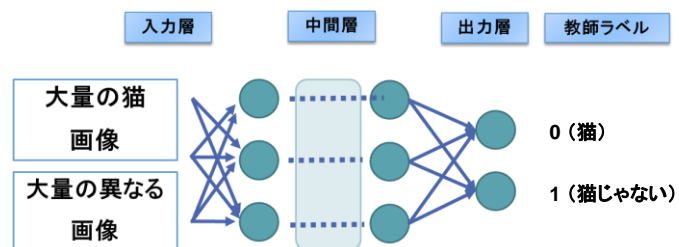
- 転移学習



(<http://ruder.io/transfer-learning/>)

7-2. 学習済みモデルの利用

画像分類の学習



予測時の出力層の値



0.2
0.8

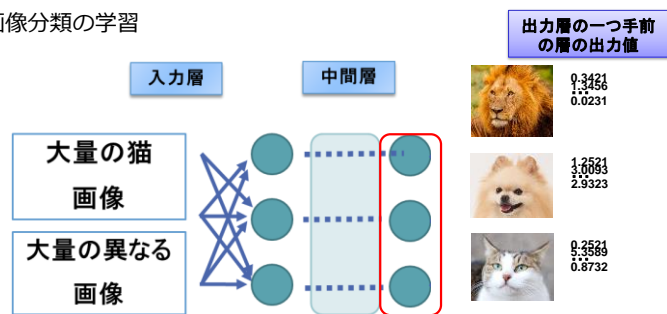


0.05
0.95



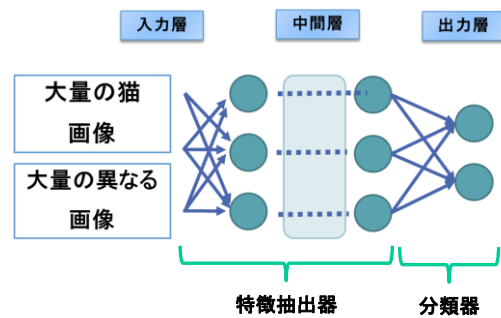
0.93
0.07

画像分類の学習

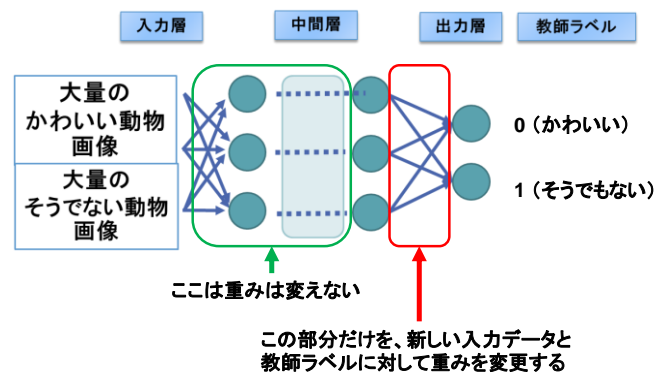


出力層の一つ手前の層(全結合層の場合が多い)で、
入力はベクトルデータに変換される。
そのベクトルデータは、入力の特徴を表す特徴ベクトルとなっている

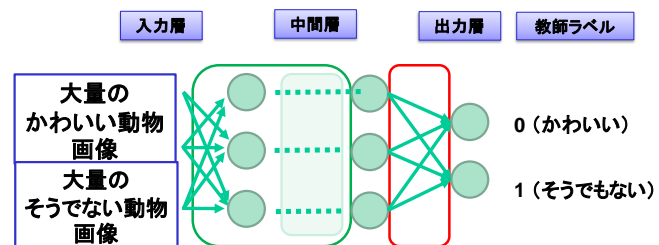
ニューラルネットワーク全体を、特徴抽出器と分類器に分けて考える





他の分類問題を解決するために、新しいデータで学習する



他の分類問題を解決するために、新しいデータで学習する



予測時の出力層の値	0.05	0.87	0.62
	0.95	0.13	0.38
			
			

7-3. 転移学習のメリット

少ないデータで高い精度を実現できる

- 汎用性を高めるためには膨大な量のデータが必要
- 既に特徴ベクトルが得られているため、少ないデータでも汎用性の高いモデルを学習できる

計算量が少ない

- 分類器の部分のみ重みを変更する
- 特徴抽出器部分を高火力マシンで学習し、分類器だけユーザーのマシンで学習することで、個々のユーザーに最適化されたモデルの開発も可能

7-4. 転移学習とFine tuning

転移学習

- 学習済みモデルの分類器部分のみ再学習する
- 分類器は必ずしも同じ手法である必要はない
 - ディープラーニングで得られた特徴ベクトルを使ってSVMで分類器を作る等

Fine tuning

- 学習済みモデルの一部、またはすべての重みを再学習する
- つまり、学習済みモデルを初期値として学習すると見なせる

7-5. ドメイン適応

ドメイン適応

- 解決するタスクは同じだが、入力データの分布が異なる場合に適応させるために、入力層に近い層を再学習する
- 例：音声認識
 - 音声->単語->文章と生成する場合、単語->文章は共通だが、音声->単語は話者によって異なる発音を認識する必要があるため、その部分を再学習することで対応できる