

高度ITを活用したビジネス創造プログラム 講師マニュアル

－AI基礎講座 Vol.2－

プログラム概要

高度IT 技術を活用したビジネス創造プログラムの概要

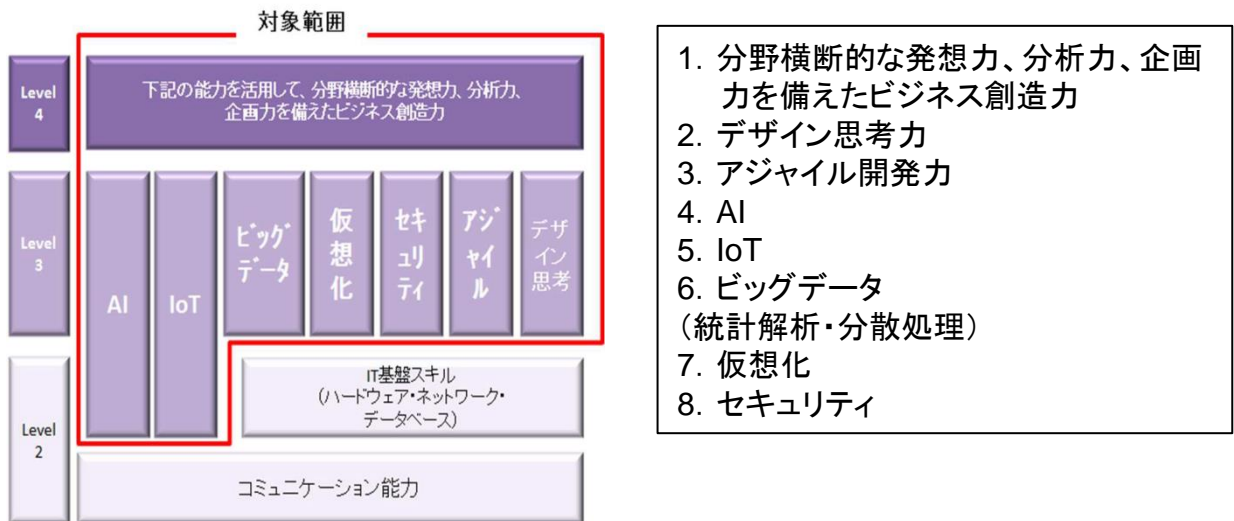
1. プログラム開発の目的と背景

■プログラムの目的

第4 次産業革命において必須であるIoT、AI やビッグデータに代表されるIT 系の技術を駆使し、新たな発想（サービス企画・デザイン思考）でビジネスを創造できる高度IT エンジニアを育成する。

■修得すべき能力とその理由

修得すべき能力を図で表すと下記のようなイメージになる。今回は一定のスキル（レベル2～3程度）を修得しているエンジニアを受講者に想定しているので、学習対象範囲の各能力についての教育訓練プログラムを作成する。



(図1)

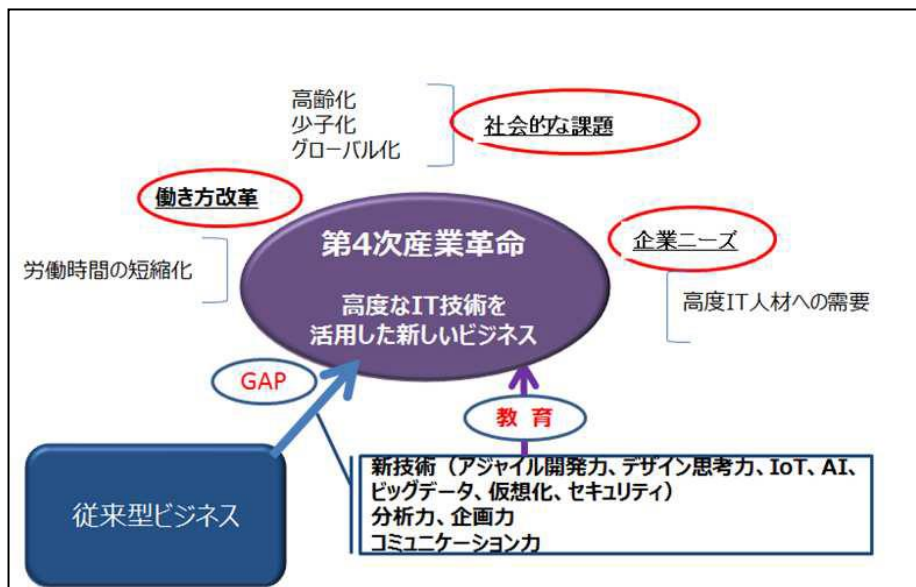
2. プログラム開発の背景

1) 「第4次産業革命」における必要性

現代社会において高齢化・少子化・グローバル化等の社会的な課題が山積する中で、それらの社会的な課題を解決することが求められている。それらの課題の解決には、新しい発想のビジネスが求められていて、それを実現するためには、高度なIT技術の活用が不可欠である。企業も社会的な課題を解決するための新たなビジネスチャンスを探ってはいるが、現状はほとんどの技術者が従来型ビジネスに対応した人材であり、高度なIT技術をもった人材へのニーズがある。

また、「働き方改革」という労働時間の短縮化という中で、労働時間が削減されても経済成長を促すには、単位時間あたりの労働生産性の向上が欠かせない。労働生産性の向上には、より高度なIT技術の修得が不可欠である。

このように、「社会的必要性」「企業ニーズ」「働き方改革」という3つの要因で上記に挙げた8つの能力が必要である。上述の内容を表したのが図2である。

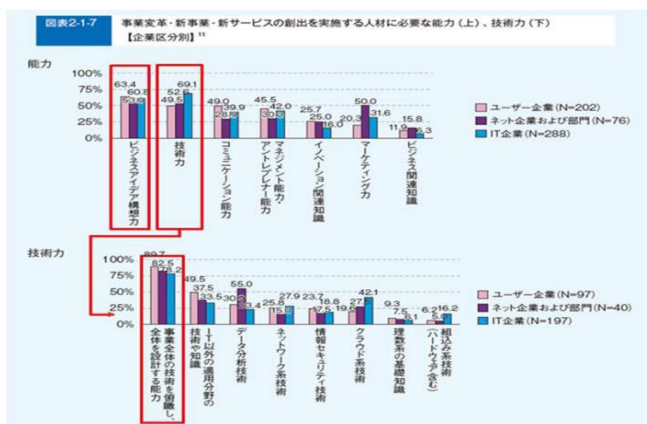


(図2)

2.プログラム開発の背景

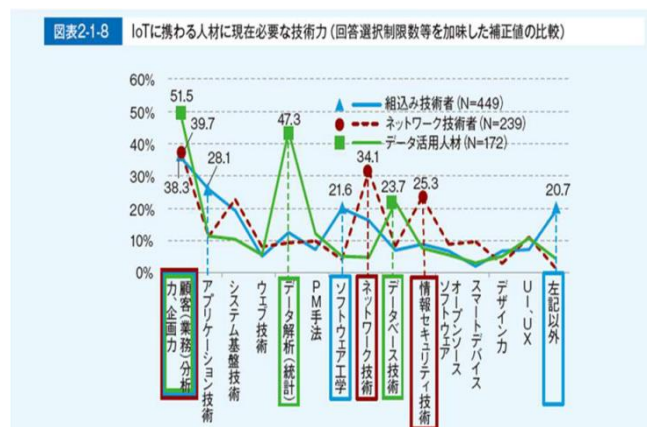
2)8 つの能力の根拠

図2のように、企業が新しいビジネスを創造するためには、従来型ビジネスとのGAPを埋めるための能力が必要であり、IPAの「IT人材白書2016」(資料1)によると「ビジネスアイデア構想力」「技術力」が求められる



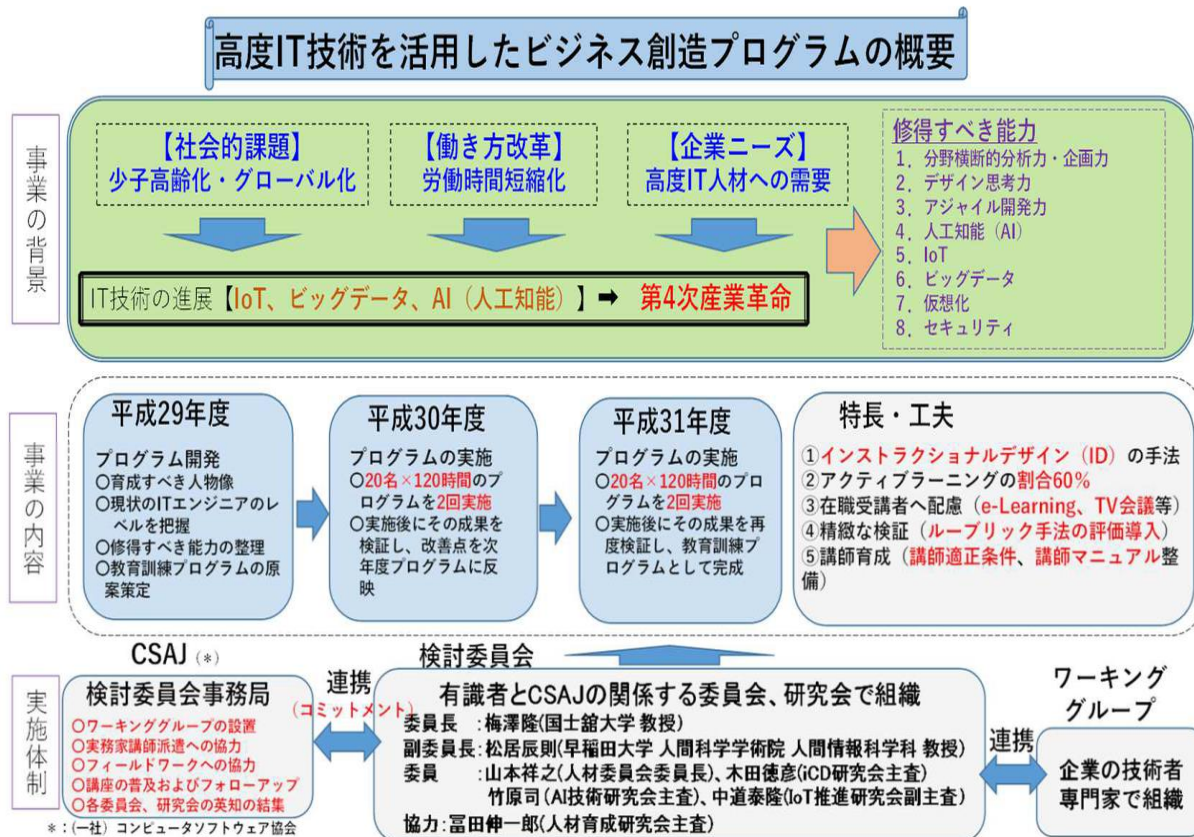
(資料1)
『新事業・新サービス創出に必要な能力・技術力とは？』
(IPA「IT人材白書2016」)

「ビジネスアイデア構想力」「技術力」とは何かを詳細に見てみると、例えば「IoTに関わる人材に必要な能力は」というIPAのアンケート調査「IT人材白書2016」(資料2)では「顧客分析・企画力」「データ解析」「ソフトウェア工学」「ネットワーク技術」「データベース技術」「情報セキュリティ」があげられる。



(資料2)
『IoT人材に必要な技術力とは？』
(IPA「IT人材白書2016」)

3.実施組織



4.高度IT 技術を活用したビジネス創造プログラムの構成

■各講座の時間割

コース名	講義	演習	小計
1. オリエンテーション	2:00	-	2:00
2. デザイン思考講座	4:00	6:00	10:00
3. 仮想化講座	4:00	4:00	8:00
4. ビッグデータ講座	7:10	7:50	15:00
5. AI基礎講座	8:35	7:25	16:00
6. IoT活用講座	8:50	7:10	16:00
7. セキュリティ講座	6:30	4:30	11:00
8. アジャイル講座	5:40	6:20	12:00
9. 顧客分析・企画力養成講座	4:50	13:10	18:00
10. フィールドワーク（セキュリティ・アジャイル開発・AI）	-	12:00	12:00

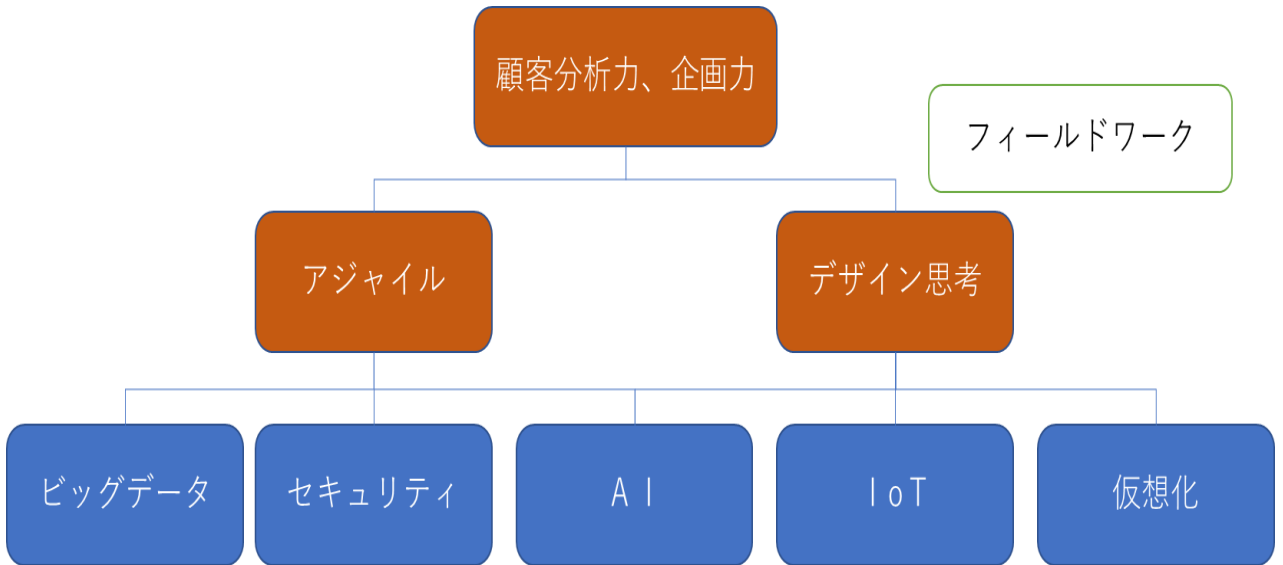
■フィールド・ワーク

	関連講座	訪問先企業（予定）	学習概要
1	セキュリティ	株式会社ラック様	場所：本社ビル 永田町 内容：1.ラック様のセキュリティ業務の実際 2.JSOCS見学 3.体験ゲーム
2	アジャイル開発	KDDI株式会社様	場所：新宿 内容：1.開発現場の見学 2.KDDI様のアジャイルへの取り組み 3.体感ゲーム
3	AI基礎	日本電気株式会社様	場所：三田 内容：1. NEC Future Creation Hub 見学 2:AI企画の概要と演習説明 3:【演習】データ・バリューチェーンの作成 4:まとめ

5.高度IT 技術を活用したビジネス創造プログラムの構成

講座の関係図

- ・赤枠：思考法・発想法
- ・青枠：IT技術知識・スキル



6. 教育訓練プログラム受講修了及び評価

■ 受講修了条件

● 終了時間：120 時間

● 終了時の能力像：「第4 次産業革命において、IT 系の必須技術を駆使し、新たな発想を持ってビジネスを創造できる知識・スキル」

【各講座の時間数】

No	講座名 (モジュール名)	時間	No	講座名 (モジュール名)	時間
1	オリエンテーション	2	6	IoT活用講座	16
2	デザイン思考講座	10	7	セキュリティ講座	11
3	仮想化講座	8	8	アジャイル講座	12
4	ビッグデータ講座	15	9	顧客分析・企画力養成講座	18
5	AI基礎講座	16	10	フィールドワーク	12
総合計					120

■ 評価指標と基準点

下記の指標に対する基準点をすべてクリアする。

	実施タイミング	指標	合格基準
1	—	出席率	80%以上
2	各講座	e-learning テスト	100%
3	各講座	理解度確認テスト	80%
4	各講座	ルーブリック仕様のアンケート	～ができる(下記に例示)
5	各講座	成果物評価	講師による評価：講師が正常稼働及び理解度を前提に総合的な判定を行う

ルーブリック仕様のアンケート (例)

自己評価		合格基準		
		1	2	3
理解度	内容を理解している	メソッドのメリット・デメリットを理解している。	メリット・デメリットをメンバーに説明できる。	メリット・デメリットについて自分の考えを説明できる
応用力	現場で活用できる	状況に応じてメリット・デメリットを説明できる。	状況にあったメソッドを選択できる。	状況にあったメソッドで課題解決に取り組める。

7. A I 基礎講座 概要

ねらい	最新の A I 技術である機械学習、DeepLearning を学び、実際の AI アプリケーションを作成する。				
開催日程	16 時間 (e-Learning 2 時間含む)				
受講条件	IT 技術者としての経験が 3 年以上、ICT の基礎知識を持っていること				
学習目標	DeepLearning をはじめとした様々な AI 技術について学び、これらを活用して自社ソリューションを構築し、実際に自ら動かすことができることを想定。				
	時間	講義	演習	学習概要	学習詳細
カリキュラム 概要	1:00	1:00	0:00	人工知能概論	・従来手法と機械学習は何が違うのか？
					・人工知能とは
					・人工知能概論の参考文献
					・深層学習 Deep Learning とは？
					・ニューラルネットワークと
					・機械学習の例
					・人工知能の世間的な分類
					・弱い人工知能と強い人工知能
					・人工知能の利用動向
	・今後の展望				
	・制度的課題：知的財産の観点				
	1:00	1:00	0:00	ニューラルネットワークと ディープラーニング	・ニューラルネットワーク
・ロジスティック回帰					
・ディープラーニング（深層学習）とは					
・従来の機械学習とディープラーニングの比較					
・代表的なディープラーニング					
・できていることと今後できそうなこと					
4:00	1:30	2:30	ディープラーニングの処理	・ディープラーニングの処理概要	
				・準備するデータセット	
				・ニューラルネットワーク内の計算	
				・活性化関数（Activation Function）	
				・損失関数（Loss Function）	
				・One-hot ベクトル	
				・誤差最小化手法	
				・誤差逆伝播法	
				・バッチ学習とミニバッチ学習	
・勾配消失問題					

					<ul style="list-style-type: none"> ・過学習、過剰適合 ・分類の評価指標 	
	3:00	50	2:10	Convolutional Neural Network	<ul style="list-style-type: none"> ・CNN の基礎知識 ・画像情報の弱点 ・畳み込み (Convolution) ・プーリング (Pooling) ・パディング (Padding) ・畳み込みの意義 ・高度なモデル: VGG ・高度なモデル: GAN 	
	1:00	1:00	0:00	RNN,LSTM,AE	<ul style="list-style-type: none"> ・RNN: Recurrent Neural Network ・LSTM: Long Short-Term Memory ・LSTM による予測 ・時間軸と自然言語 ・Auto Encoder 	
	1:00	0:15	0:45	学習画像生成と分類演習	<ul style="list-style-type: none"> ・画像データの水増し ・Keras による学習画像生成と画像分類演習 	
	1:00	1:00	0:00	転移学習	<ul style="list-style-type: none"> ・転移学習 ・学習済みモデルの利用 ・転移学習のメリット ・転移学習と Fine tuning ・ドメイン適応 	
	2:00	0:00	2:00	総合演習	<ul style="list-style-type: none"> ・演習 	
	合計時間	14:00	6:35	7:25	教材 1 章を事前実施	

8. A I 基礎講座 詳細カリキュラム

時間	学習項目	学習項目の狙い	詳細内容
1:00	人口知能概論	<p>目的 1 : この研修における目標を明確にし、研修への意欲を高める</p> <p>目的 2 : 人工知能を実現する多様な手法について理解する。</p> <p>ゴール : 統計的手法、推論と探索、エキスパートシステム、機械学習など従来手法の特徴を理解する。</p>	<p>・講師自己紹介</p> <p>・コースの目的 (AI を実現する多様な手法の知識と現在注目されているディープラーニングの知識と技術を習得する。自社で独自に簡単な検証できるようになってもらいたい)</p> <p>★注意点</p> <ul style="list-style-type: none"> ・すべての技術背景をこの研修で取り扱うのは不可能 ・研修では画像を入力とする CNN を集中的に実施 ・試験は研修が 10 問と e-learning が 20 問ずつ ・配布資料の確認 <p>【講義】</p> <p>①e-Learning で事前学習としている事項について簡単に振り返り学習を行う。</p> <p>②統計的手法 (回帰分析、判別分析)、推論と探索 (探索木)、エキスパートシステム (IF-THEN ルール)、機械学習 (k-means 法、自己組織化マップ、k-近傍法、決定木、ランダムフォレスト、サポートベクターマシン) について仕組みを簡単に解説する。</p>
1:00	ニューラルネットワークとディープラーニング	<p>目的 :</p> <p>従来手法の一つであるニューラルネットワークの発展版ともいえるディープラーニングについて、従来の機械学習との特徴の違いを理解する。</p>	<p>【講義】</p> <p>③ニューラルネットワークを用いた回帰や分類の仕組みを解説し、ロジスティック回帰で出力を確率としてみなすことができることを説明する。</p>

			<p>④ディープラーニングとは多層のニューラルネットワークを用いた機械学習であることを解説し、CNN、RNN、AE について主な用途と今後で実現されそうな分野を紹介する。</p> <ul style="list-style-type: none"> ・CNN、RNN、AE は代表的なモデルだが、単体で用いられることはほとんどない。 ・動きが早いため、情報の更新に注意。
1:30	ディープラーニングの処理	<p>目的：</p> <ul style="list-style-type: none"> ・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。 ・NNC (Neural Network Console)の使い方を理解する。 ・NNC を用いてシンプルなディープラーニングを実装する。 <p>ゴール：</p> <ul style="list-style-type: none"> ・ディープラーニングにおける処理要素である、活性化関数や損失関数、誤差最小化手法、誤差逆伝播法などについて、その働きを理解する。 ・NNC の使い方を習得する。 ・NNC により多次元ベクトルを入力とする分類モデルおよび回帰モデルを構築する。 	<p>[講義]</p> <p>①ディープラーニングにおける処理要素（各層におけるニューラルネットワークの計算、活性化関数、損失関数、One-hot ベクトル、誤差最小化手法、誤差逆伝播法、ミニバッチ学習、勾配消失問題、過学習・過剰適合、分類の評価指標）について解説する。</p> <ul style="list-style-type: none"> ・数学をかなり多用するので、受講者の反応を見ながら内容を進める。 ・勾配消失の原因が、誤差逆伝播の課程で最大絶対値が 1 未満となる活性化関数の微分を多重にかけることにあることに特に注意する。 ・誤差最小化手法における学習率により、誤差収束の挙動が変わることに注意する。 ・評価指標（再現率、適合率）の違いと、目的によって重視する指標が異なることに注意する。
2:30			<p>[演習]</p> <p>①実際に NNC を操作し、各種設定を行う。</p>

・データセットファイルの書式、学習データセットの指定、ニューラルネットワークの構築、最小化手法の指定など各種設定を行う。

[以下、ニューラルネットワークのノード間の接続を意識しながら演習を行う]

② 次元点の線形 2 クラス分類を単純パーセプトロンとロジスティック回帰で行い、分類結果を確認する。

・分類境界付近では、ロジスティック回帰の出力が 0.5 に近く、境界から離れた部分では 0 または 1 に近くなることを確認し、ロジスティック回帰の出力がそれぞれのクラスに属する確率のように振舞うことを確認する。

③ 次元点の非線形 2 クラス分類をロジスティック回帰とディープニューラルネットワーク (DNN) で行い、分類結果を確認する。

・非線形分類問題はロジスティック回帰では分類できず、DNN では可能となることを確認し、層を増やすことで分類能力 (分類モデルの表現力) が向上することを確認する。

④ DNN による多次元データの分類

・デンバーの気象データ (Kaggle のオープンデータを改変したもの) を使い、湿度、気圧、気温、風向、風速のデータから天気 (晴れ、曇り、雨、雪) を予測する分類モデルを構築する。

・3 クラス以上分類を行うために活性化関数は Softmax、損失関数は CategoricalCrossEntropy を使う。

			<ul style="list-style-type: none"> ・入力データの値の大きさにばらつきがある場合に精度が悪くなるため、値の大きさを揃える正規化を行うと精度が向上することを説明する。あわせて BatchNormalization も説明する。 <p>⑤DNN による多次元データの回帰</p> <ul style="list-style-type: none"> ・ボストンの住環境と住宅価格データ (Kaggle のオープンデータ) を使い、犯罪率や税率などの住環境データから住宅価格を予測する回帰モデルを構築する。 ・回帰モデルであるため、最終出力直前には活性化関数を使わず (恒等写像と同じ)、損失関数には SquaredError を使う。
50	CNN (Convolutional Neural Network)	<p>目的： 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。</p> <p>ゴール： 単純なニューラルネットワークとの違いを理解し、NNC で CNN を実装できる。</p>	<p>【講義】</p> <p>①ディープラーニングの典型である畳み込みネットワーク (CNN;(Convolutional Neural Network)について説明する。</p> <ul style="list-style-type: none"> ・畳み込み (Convolution) 、プーリング (Pooling) についてその機能を解説する。 ・ストライド、パディングと生成される特徴マップのサイズの関係について解説する。 <p>②演習後に、精度の改善方法の補足として ILSVRC の上位モデルを紹介する。(VGG モデル、ResNet など)</p>
2:10			<p>【演習】</p> <p>①NNC で CNN を実装する。</p> <ul style="list-style-type: none"> ・MNIST データを用いた CNN による 2 クラス分類と 10 クラス分類。 ・中間層におけるノードのウェイトと特徴マップの可視化。

			<ul style="list-style-type: none"> ・10 クラス MNIST を対象に、中間層、ノード数を調節することで、より精度の高い分類を試みる。 ★注意点 ・CUDA、GPU のドライバのインストールバージョンによって動作に支障が出るがあるので、演習前にバージョンを確認する。 ・学習時間が単純ロジスティック回帰より数倍に増加するため、演習時間の確保を心がける。 ・画像の畳み込み、プーリング、パディングは初めて学ぶ場合にはやや理解しがたい部分があるため、具体的な例を出して説明を行う。
1:00	RNN (Recurrent Neural Network)	<p>目的：</p> <ul style="list-style-type: none"> ・時系列データを扱う RNN (Recurrent Neural Network) について特徴を理解する。 <p>ゴール：</p> <ul style="list-style-type: none"> ・RNN の概要を理解する。 ・自然言語をディープラーニングで扱う手法を理解する。 	<p>【講義】</p> <p>①RNN (Recurrent Neural Network) と LSTM (Long Short-Term Memory) 、GRU(Gated Recurrent Unit)について簡単に特徴を説明する。</p> <ul style="list-style-type: none"> ・RNN の概念自体はそれほど難しくないが、実装する技術は複雑なので、詳細を全てこの講義で説明することは時間的に難しい。LSTM、GRU も同様。 ・ここでは概念と、RNN はその構造により時系列データに対して有効であることを確認する。 <p>②自然言語をディープラーニングの入力データであるベクトルに変換する手法について解説する。</p> <ul style="list-style-type: none"> ・単語の ID 化、文章の ID 列化、日本語の場合は形態素解析による文章の単語列化について解説する。また、単語をベクトル化する word2vec について解説する。

			<p>★注意点</p> <p>※時間の関係上、RNN は簡単な紹介のみ行い、時系列データに有効なモデルであること、自然言語は単語の時系列データであること、現在のディープラーニングは入力をテンソルにする必要があることを説明。</p> <p>[演習]</p> <p>なし</p>
0:15	学習画像生成と分類演習	<p>目的：</p> <ul style="list-style-type: none"> ・学習データの水増し（Image Augmentation）について理解する。 <p>ゴール：</p> <ul style="list-style-type: none"> ・学習データの水増しで行われる画像処理について理解する。 ・NNC による Image Augmentation を実行し、生成される（水増しされる）学習用データを確認する。 	<p>[講義]</p> <p>① 学習データの水増し（Image Augmentation）についての概要を解説する。</p>
0:45			<p>[演習]</p> <p>① 学習データの水増し（Image Augmentation）で行われる画像処理について説明を交えながら NNC で演習を行い、生成される画像を確認する。</p> <ul style="list-style-type: none"> ・画像データはラスタ画像であるため、画像に写っている対象物の位置、サイズ（画像内で対象が占める割合）、アスペクト比、明るさなどが異なれば異なる画像とみなされるが、学習に必要な正解画像を全て準備することは困難であるため、画像処理によって生成することで大量の学習データを準備できることを説明する。 ・画像のシフト、スケーリング（アスペクト変換）回転・反転、せん断変形、明度・コントラスト・彩度変更など。
1:00	転移学習	<p>目的：</p> <ul style="list-style-type: none"> ・転移学習について理解する。 <p>ゴール：</p> <ul style="list-style-type: none"> ・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。 	<p>[講義]</p> <p>① 転移学習についての概要を解説する。</p> <ul style="list-style-type: none"> ・幅広い機械学習の枠組みに対して使われる転移学習について説明する。 <p>② 音声認識などのドメイン対応について解説する。</p>

2:00	総合演習	<p>目的：</p> <ul style="list-style-type: none"> ・自作データを用いて CNN による分類を行う。 <p>ゴール：</p> <ul style="list-style-type: none"> ・MNIST の手書き文字分類モデルをベースとして、自作の手書き文字の分類精度を向上させる。 ・ASL (American Sign Language) データ (Kaggle のオープンデータより抜粋) を用いて、手話画像を分類するモデルを構築する。 	<p>【講義】</p> <p>なし</p> <p>【演習】</p> <p>①フォトレタッチソフト (XnView) とカメラアプリを用いて画像データを自作する手順について説明を交えながら練習する。</p> <ul style="list-style-type: none"> ・撮影、トリミング、リサイズ、カラーとグレイスケールの変換、レベル補正など <p>②精度の改善のヒントについて改めて触れる</p> <ul style="list-style-type: none"> ・CNN の表現力の増加、勾配消失の回避、過剰適合の回避、BatchNormalization など <p>③既存の MNIST 分類モデルに対して、自作手書き数字画像を作成して分類を行う。分類精度が悪い場合は向上を試みる。</p> <p>④ASL データの中から Blank を含め 4 種類程度の手話画像を用いて分類モデルを構築する。90%以上の分類精度を目指す。</p> <p>注意点</p> <ul style="list-style-type: none"> ・あまり規模の大きいものを実験すると学習時間が足りなくなるため規模を小さくする。 ・自作データ作成には思った以上に時間がかかる。そのため、Kaggle のデータセットから少数を抜粋して学習データとし残りのデータを分類するなど、自作データ作成はあまり多くないようにする。
------	------	---	---

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

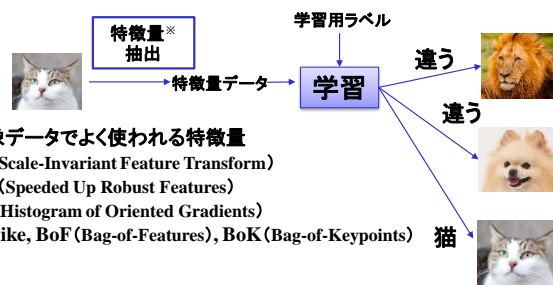
第7章 転移学習

- 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)



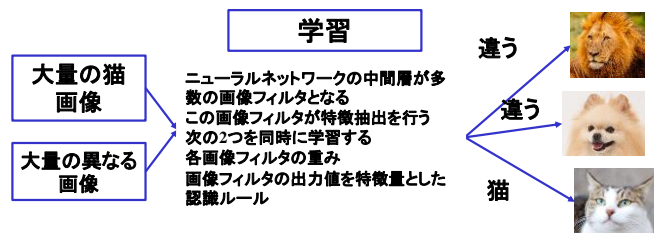
※ 画像データでよく使われる特徴量
SIFT (Scale-Invariant Feature Transform)
SURF (Speeded Up Robust Features)
HOG (Histogram of Oriented Gradients)
Haar-like, BoF (Bag-of-Features), BoK (Bag-of-Keypoints) 猫
etc.

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

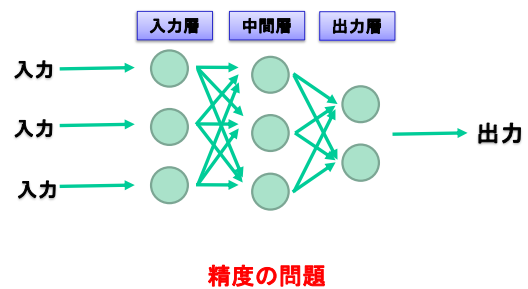
例) 画像の物体認識

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

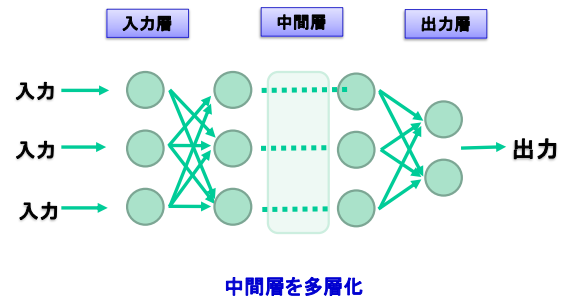
処理能力の限界から、入力層、中間層、出力層の3層構造が多く精度の問題があった



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

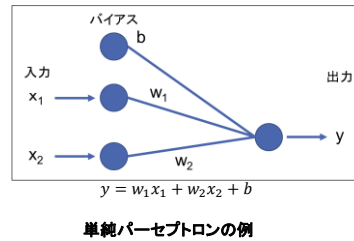
シナプスの結合によりネットワークを形成した人工ニューロン（ノード）が、学習によってシナプスの結合強度を変化させ、問題解決能力を持つようなモデル（Wikipediaより）

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

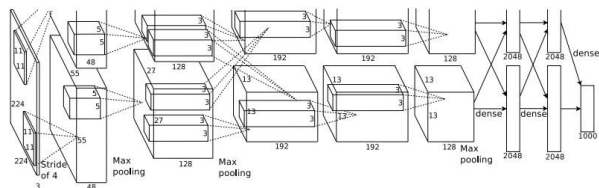
G.H. ヒントン(Toronto Univ., Google)の2006年の論文
“Reducing the Dimensionality of Data with Neural Networks”, G. E.
Hinton, R. R. Salakhutdinov, Science 28 Jul, Vol. 313, Issue 5786,
pp. 504-507,(2006)

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

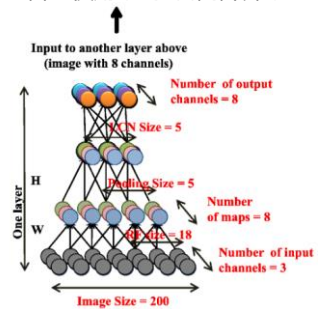
流行したきっかけ: 3

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。

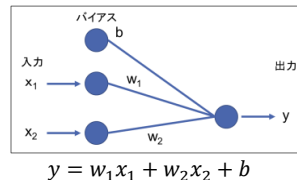
著作権についてここで説明した後、次のスライドにも続きます。

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。

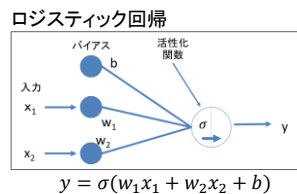
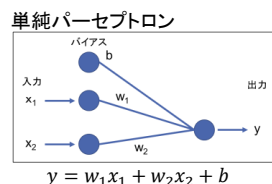
このモデルは線形分割しかできないことが知られています。

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。

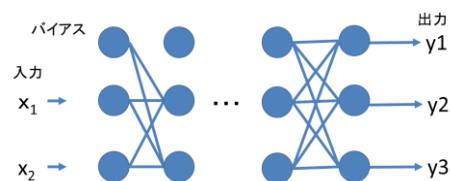
・ポイント

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)

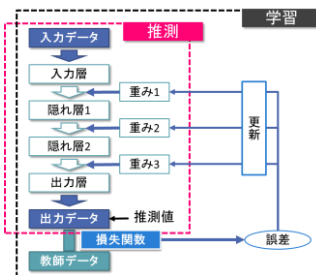
場合によっては、一度ここで時間を取って、ここまでの内容を再度整理してください。

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。

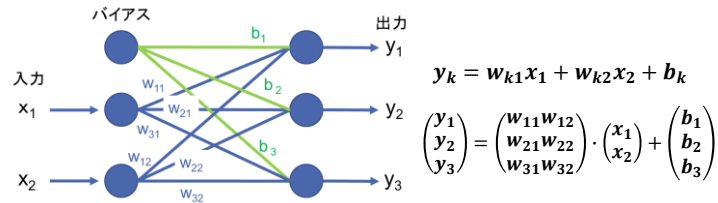
必要なデータとしてこの3つのデータセットがあることを説明してください。

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



$$y_k = w_{k1}x_1 + w_{k2}x_2 + b_k$$

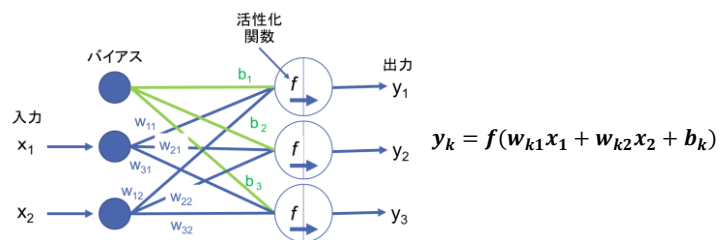
$$\begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix} = \begin{pmatrix} w_{11} & w_{12} \\ w_{21} & w_{22} \\ w_{31} & w_{32} \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix}$$

・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)

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

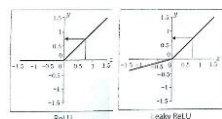
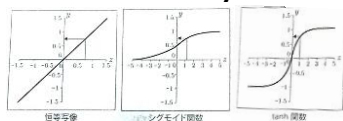
□ Leaky ReLU

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

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

□ 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となりません。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)

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

- クロスエントロピー

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

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

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

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。

。

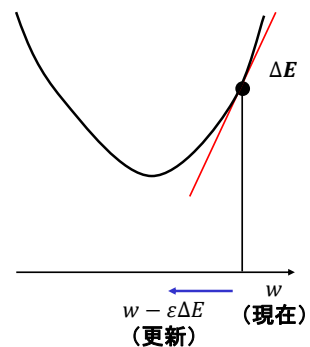
スライド47,48,49は参考程度でよいでしょう。

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

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

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

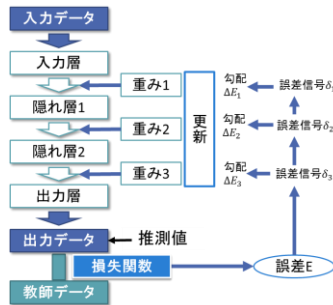
・補足説明

誤差最小化手法として、こ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になります。

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？

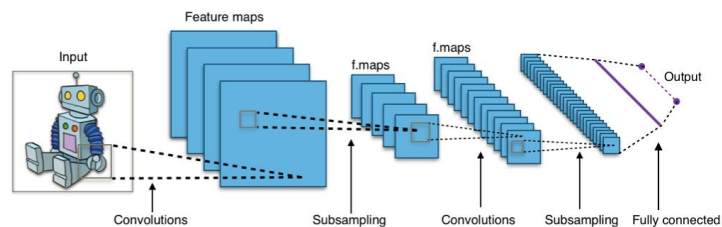
あやしいものは全てを見つけないのか？

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

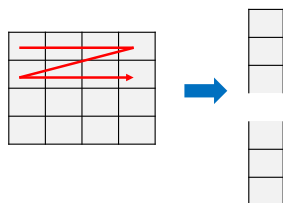
ディープラーニングの中でもよく使用されているのが畳み込みネットワークであるCNNです。特に画像データの分類によく使用されます。

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

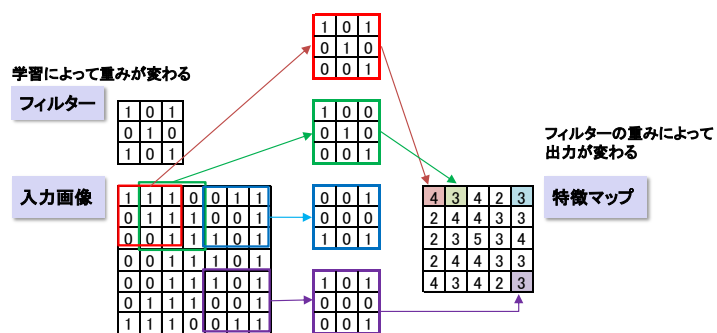
・説明の流れ

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



・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

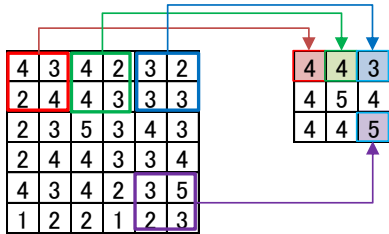
畳み込みの次に行われるのがプーリング処理です。

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



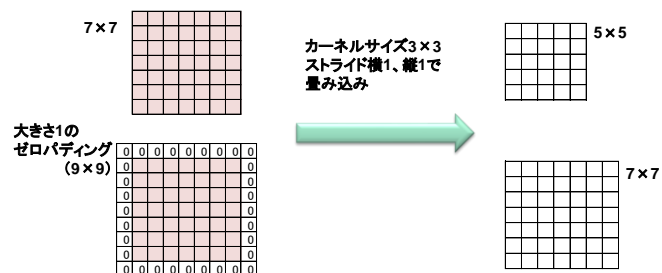
・説明の流れ

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

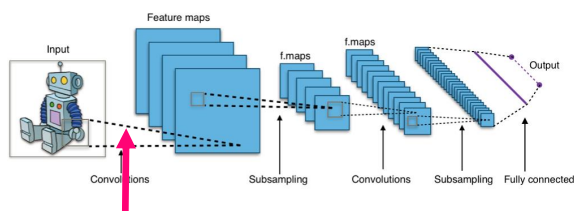
畳み込みの時に、画像サイズを過度に小さくしたくない場合にパディングを使用します。

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

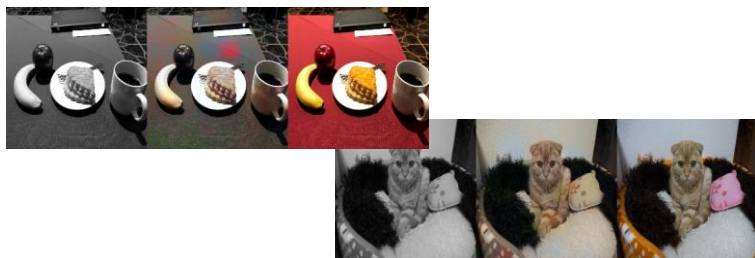
https://en.wikipedia.org/wiki/Convolutional_neural_network

・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 卷 (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

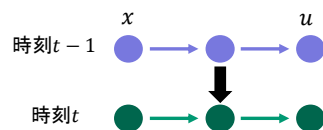
再帰型ニューラルネットワーク

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

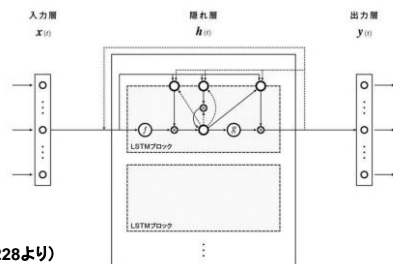
過去の間層の出力 $u(t-1)$ には、再帰的に過去の状態が全て反映されていることになる



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

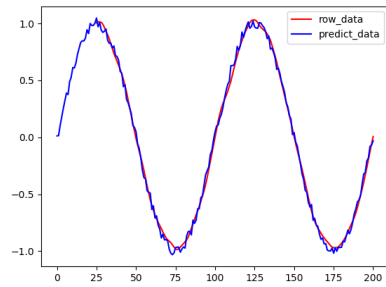
LSTMはRNNの拡張型で、その名の通り長短期記憶(Long Short-Time Memory)に特徴があります。

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

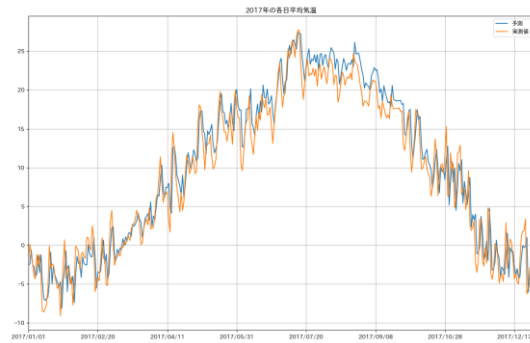
教師信号の正解データ：1976/1/1

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

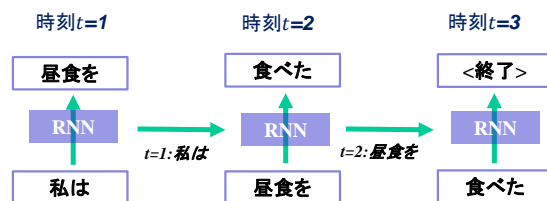
結果：
平均 1.9度
最小 0.0度
最大 7.3度



5-4. 時間軸と自然言語

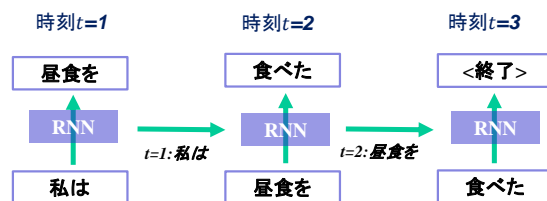
RNNは時系列データであれば何でも扱える。例) 株価
自然言語 (文章) もまた、1種の時系列データである

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

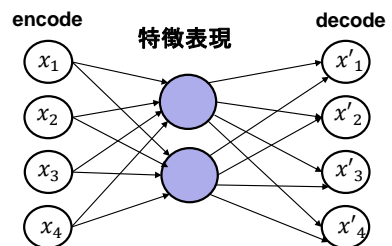


5-5. Auto Encoder

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

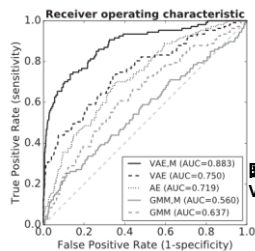
教師データがない = ラベル付けがない

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

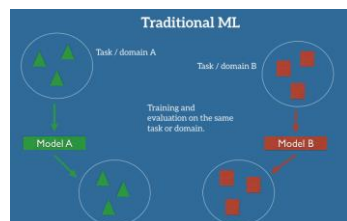
- ディープラーニングに限らず、幅広い機械学習の枠組みに対して使われる言葉
- あるタスクを効果的かつ効率的に解決するために、別のタスクで学習された知識を得て、それを適用する問題
 - “the problem of retaining and applying the knowledge learned in one or more tasks to efficiently develop an effective hypothesis for a new task.”
- 転移学習のワークショップ (NIPS 2005 Workshop - Inductive Transfer: 10 Years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

説明委の流れ

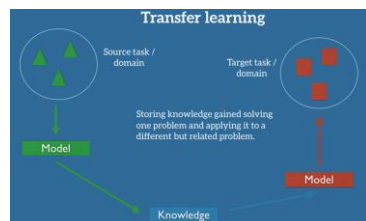
転移学習のメリットを理解し、機械学習のなかでの有効性を知る。

転移学習についての概要を解説する。

- 従来の機械学習



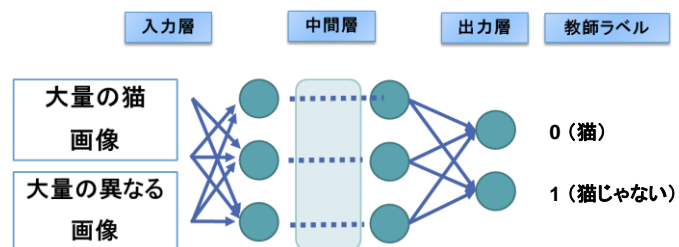
- 転移学習



(<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

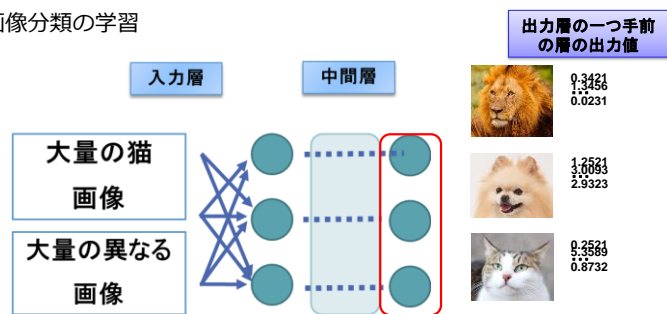


0.05
0.95



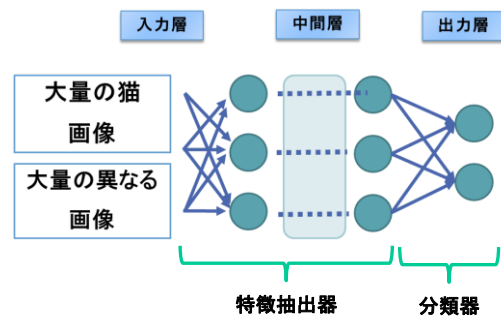
0.93
0.07

画像分類の学習

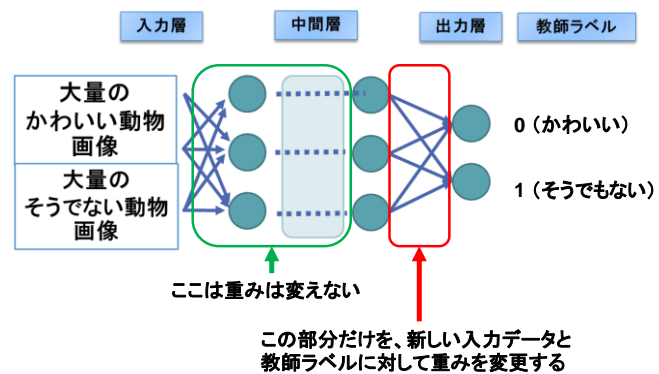


出力層の一つ手前の層(全結合層の場合が多い)で、
入力はベクトルデータに変換される。
そのベクトルデータは、入力の特徴を表す特徴ベクトルとなっている

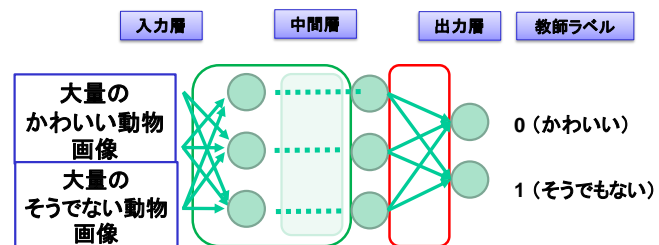
ニューラルネットワーク全体を、特徴抽出器と分類器に分けて考える



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
0.95



0.87
0.13



0.62
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

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: VGG64
4-8. 高度なモデル: GAN65

目次

第5章 Recurrent Neural Network	
5-1. RNN: Recurrent Neural Network	68
5-2. LSTM: Long Short-Term Memory	69
5-3. LSTMによる予測	70
5-4. 時間軸と自然言語	74
5-5. Auto Encoder	76

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

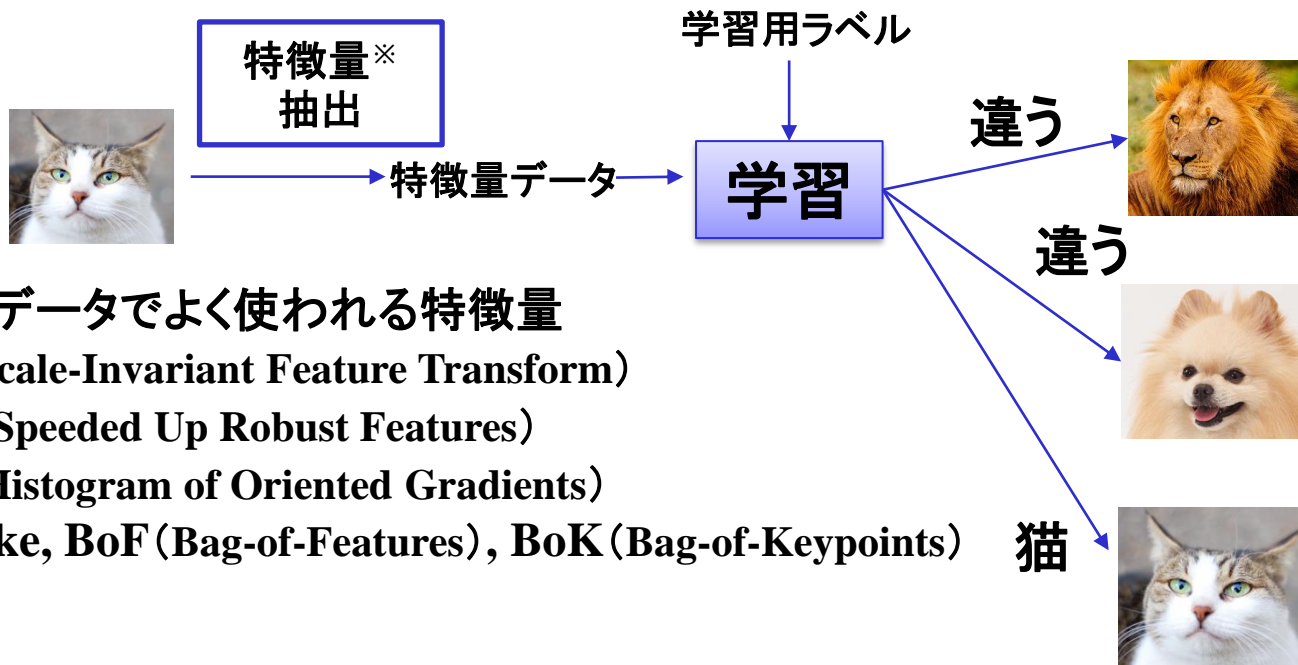
第7章 転移学習

- 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)

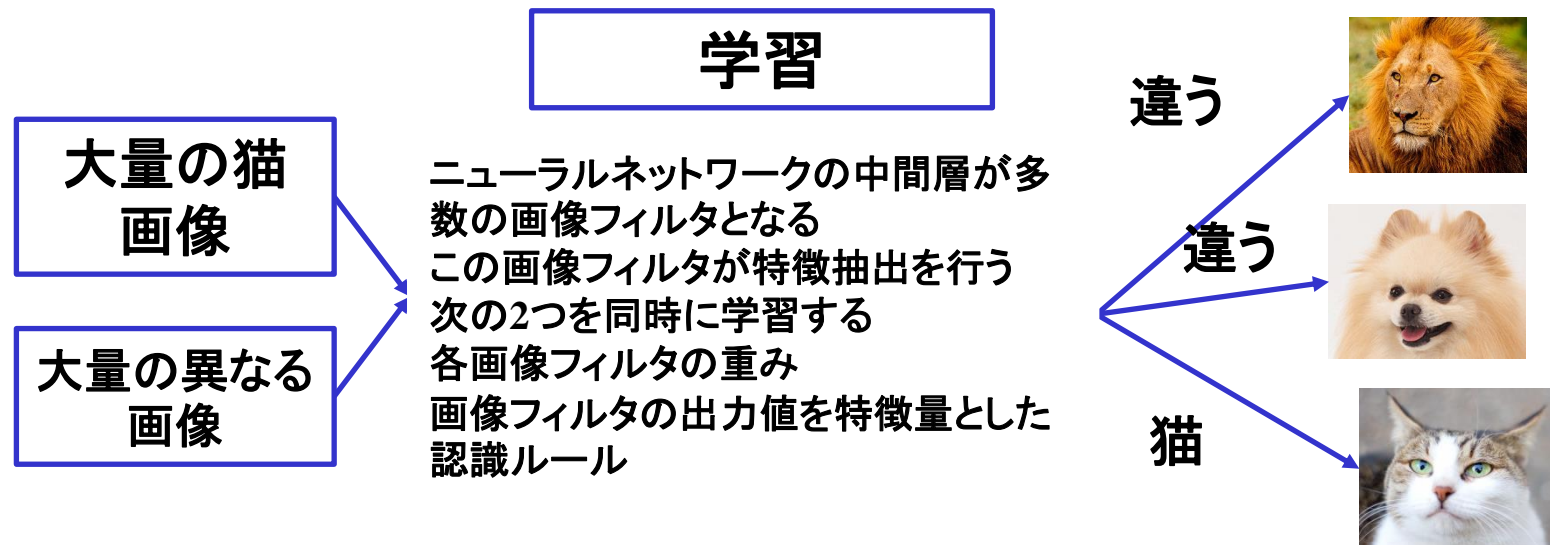
etc.

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

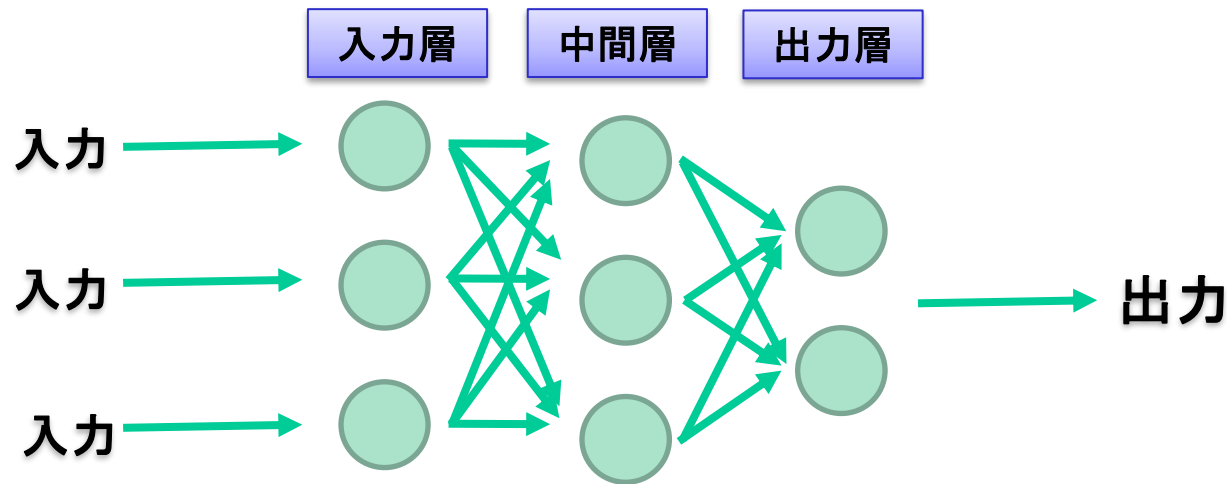
例) 画像の物体認識

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

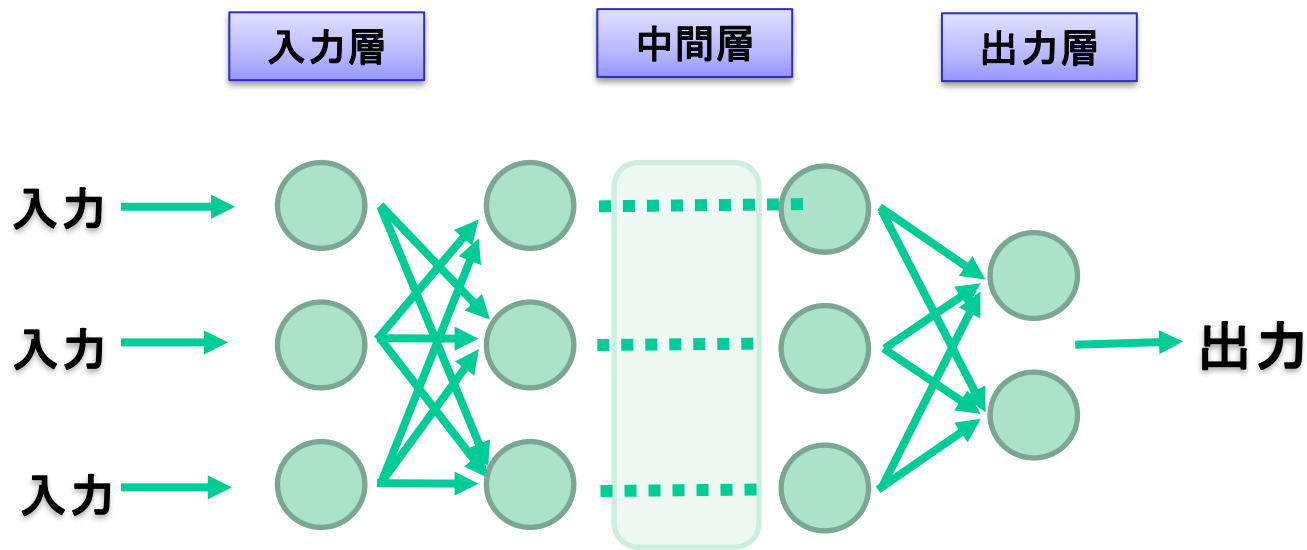


精度の問題

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



中間層を多層化

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。

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

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

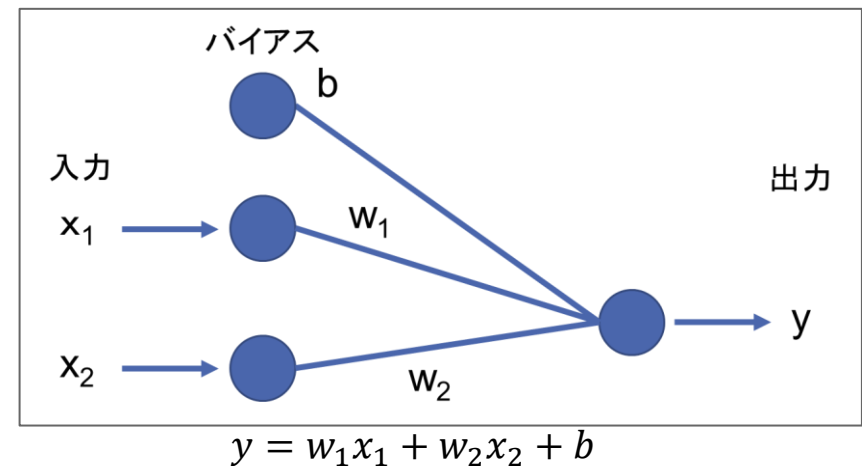
シナプスの結合によりネットワークを形成した人工ニューロン（ノード）が、学習によってシナプスの結合強度を変化させ、問題解決能力を持つようなモデル（Wikipediaより）

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



単純パーセプトロンの例

流行したきっかけ: 1

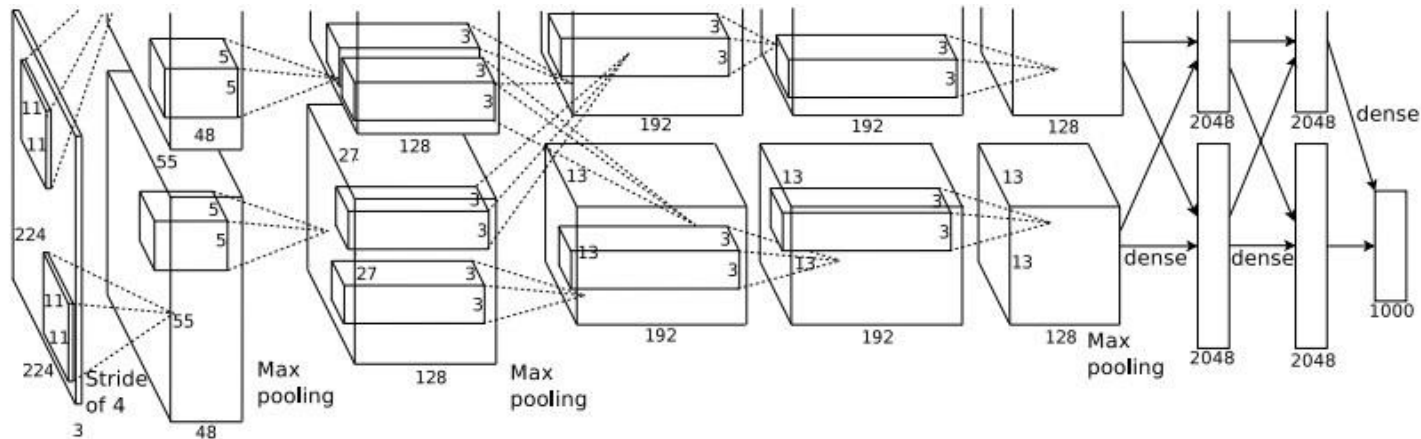
G.H.ヒントン(Toronto Univ., Google)の2006年の論文

“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

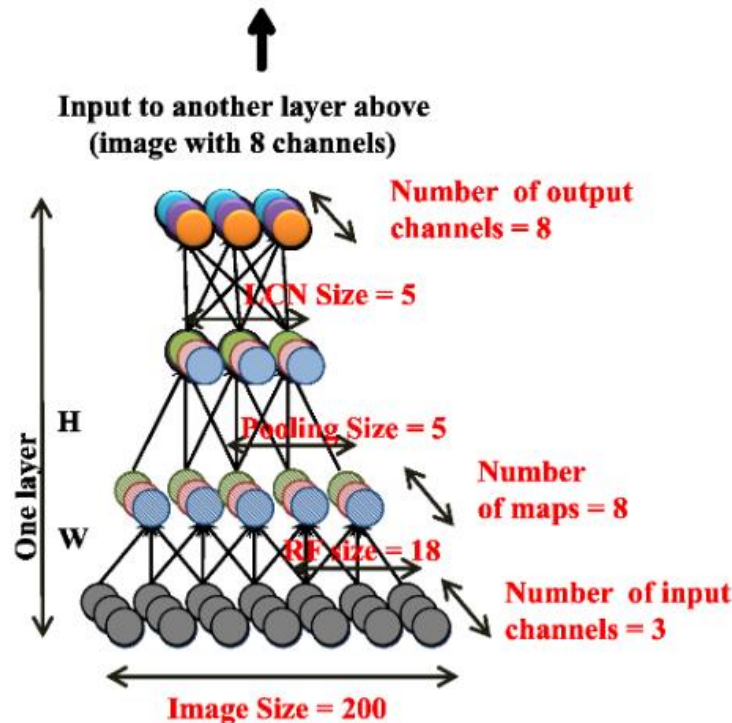


AlexNet: 5つの畳み込み層、3つの全結合層、65万ニューロン

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に投入した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



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）

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

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)

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)

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

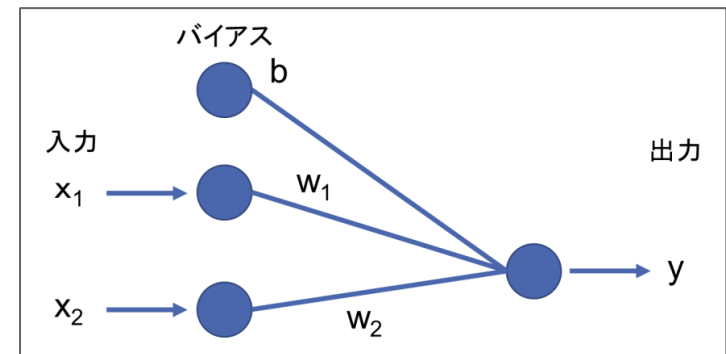
学習用データの活用と保護

- 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）

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



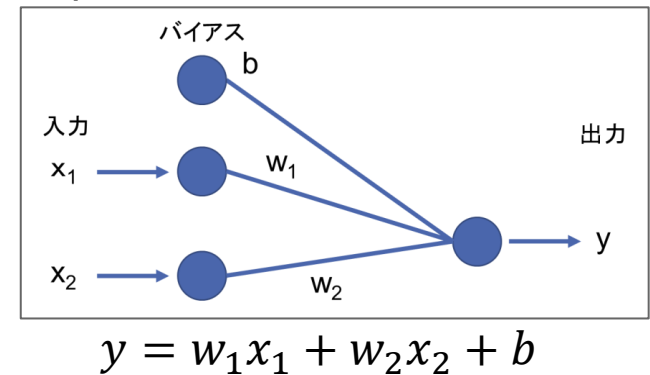
$$y = w_1x_1 + w_2x_2 + b$$

単純パーセプトロンの例

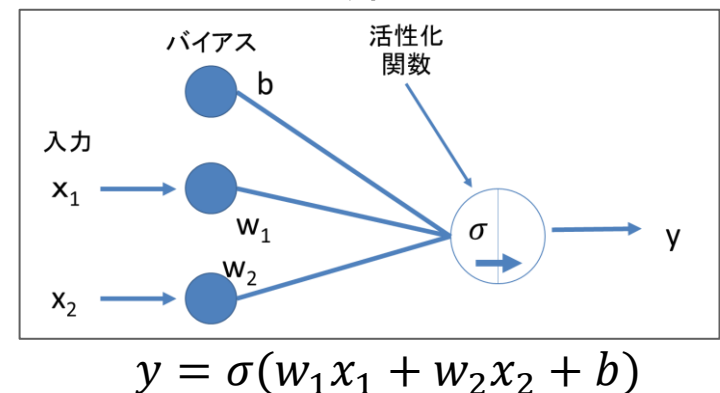
2-2. ロジスティック回帰

- 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

単純パーセプトロン

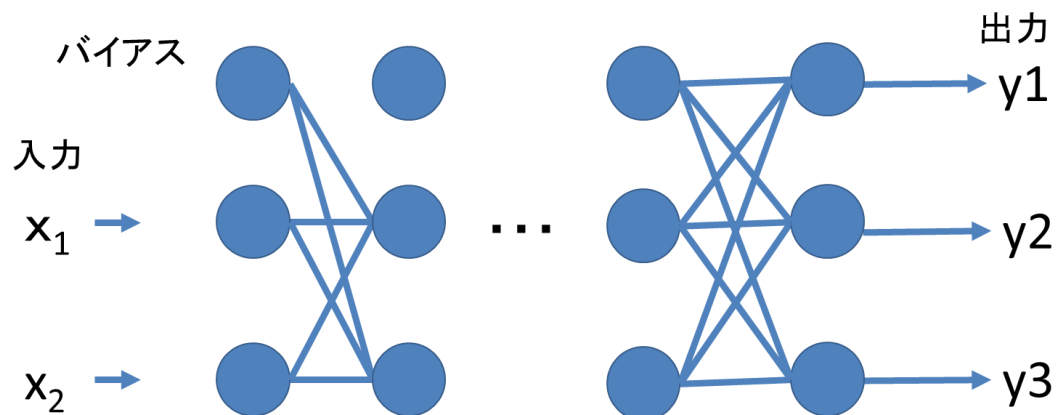


ロジスティック回帰



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



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

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

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）

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、

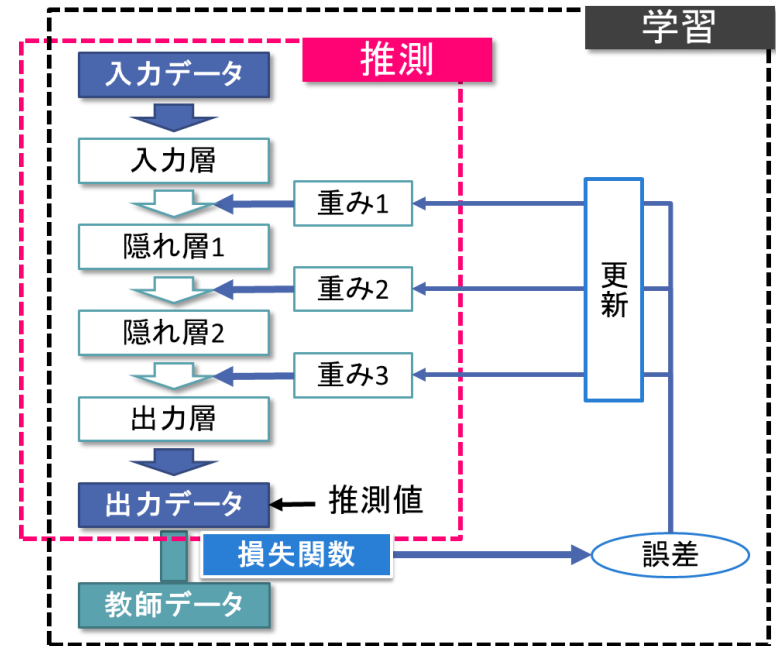
- CNN、RNN、AEの組合せ
- 強化学習
- Generative Adversarial Network（敵対的生成ネットワーク）

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

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



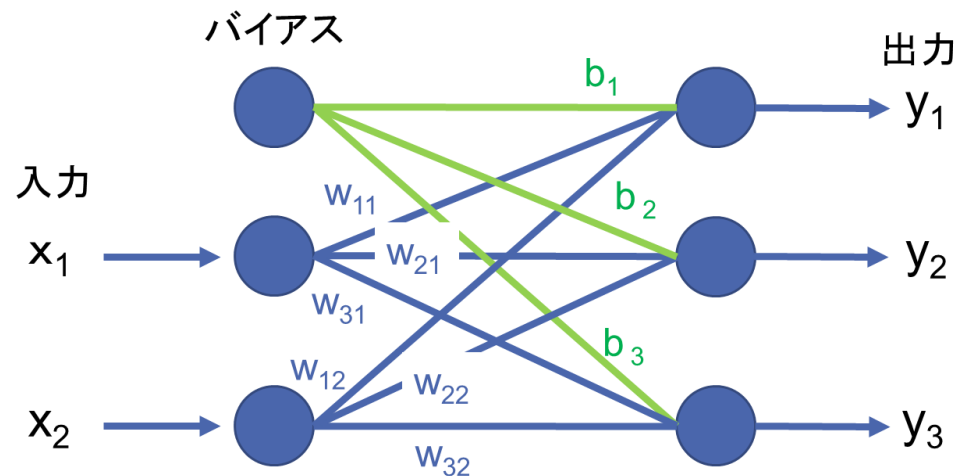
3-2. 準備するデータセット

- 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

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

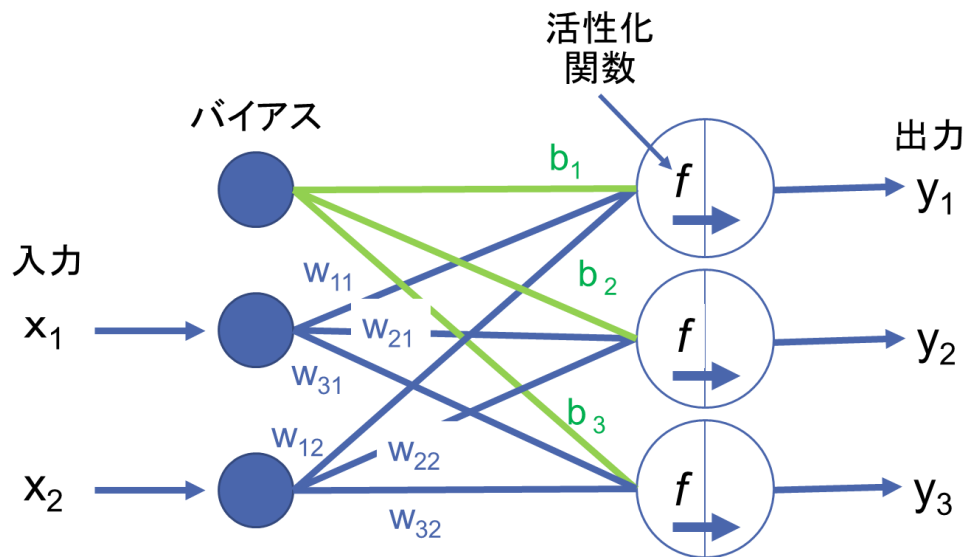


$$y_k = w_{k1}x_1 + w_{k2}x_2 + b_k$$

$$\begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix} = \begin{pmatrix} w_{11} & w_{12} \\ w_{21} & w_{22} \\ w_{31} & w_{32} \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix}$$

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



$$y_k = f(w_{k1}x_1 + w_{k2}x_2 + b_k)$$

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

➤ $y = x$

□ シグモイド関数 (Sigmoid)

➤ $y = \frac{1}{1+e^{-x}}$

□ tanh関数

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

□ ReLU (Rectified Linear Unit)

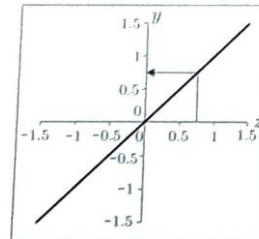
➤ $y = \max(x, 0)$

□ Leaky ReLU

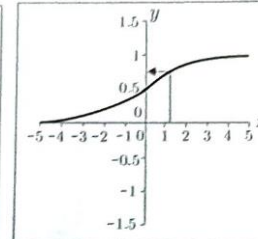
➤ $y = x$ if $x \geq 0$
 $y = ax$ if $x < 0$

□ ソフトマックス関数 (Softmax)

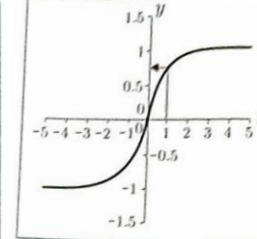
➤ $y_i = \frac{e^{x_i}}{\sum_{j=1}^K e^{x_j}}$ K: 出力層のユニット数



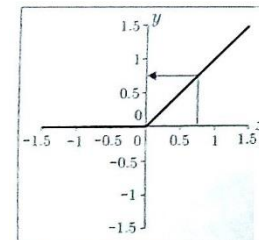
恒等写像



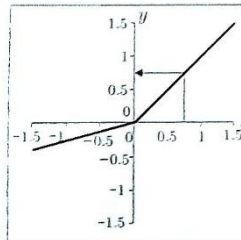
シグモイド関数



tanh 関数



ReLU



Leaky ReLU

(藤田、高原、「実装ディープラーニング」、p.52)

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

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

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

- 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

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

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

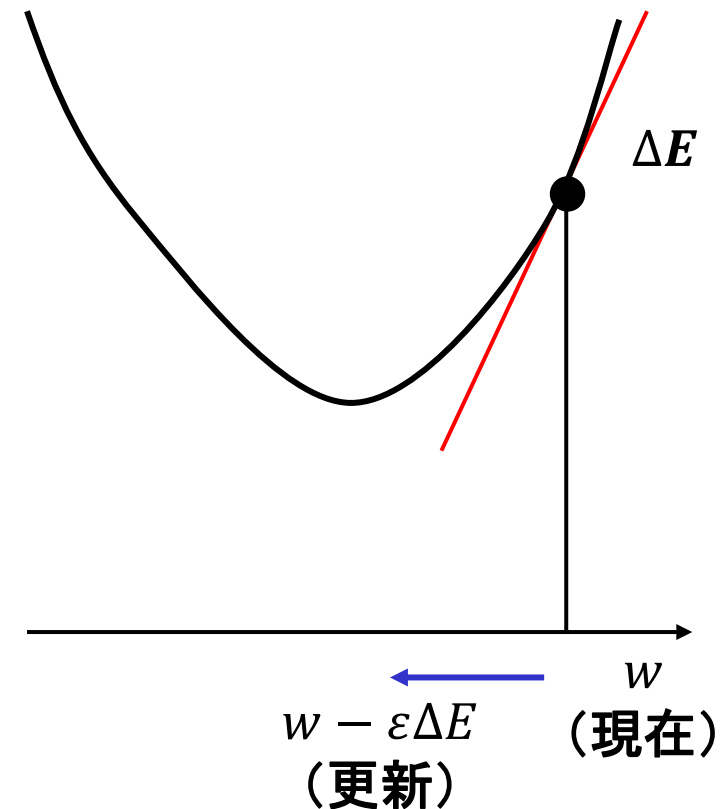
- 分類問題の教師ラベル表現として使われる
 - 分類問題では出力層のノード数を総ラベル数にし、Softmax関数を活性化関数とする
 - 教師データをOne-hotベクトルで与えると、正解ラベルに相当するノードの出力が1、それ以外のノードの出力が0に近づくようにノードの重みを更新する
 - Softmax関数を使うので、それぞれの出力ノードの値は、そのラベルに属する確率とみなすことができる

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

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
 - 収束点を通り過ぎることがある
(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）

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

- RMSProp

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

- 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）

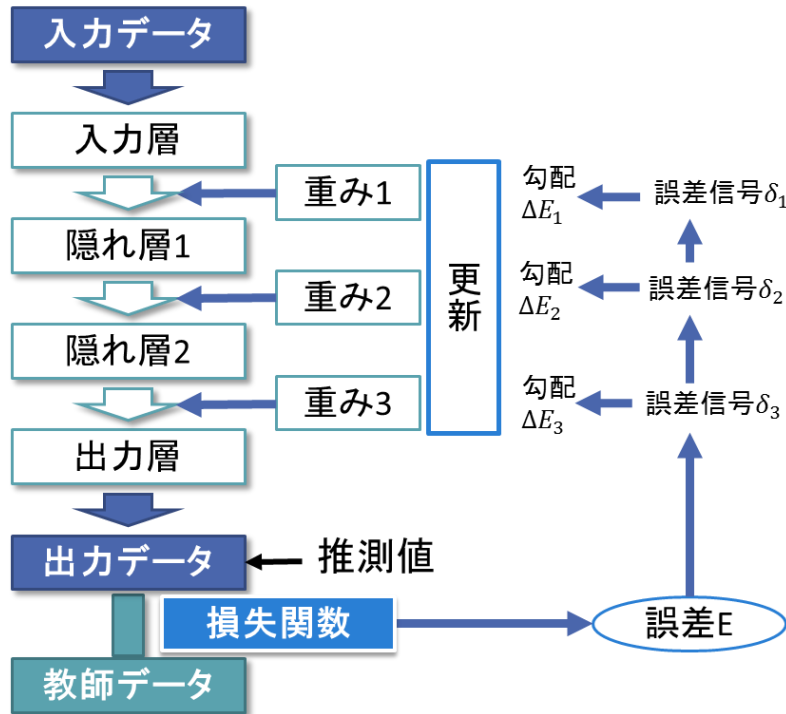
$$m \leftarrow \beta_1 m + (1 - \beta_1) \Delta E, \quad v \leftarrow \beta_2 v + (1 - \beta_2)(\Delta E)^2$$

m は重みの平均値、 v は重みの分散

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

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

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

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

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

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

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

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

。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

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

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って行列計算を行う

□ 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回終わることを1エポック (Epoch) と呼ぶ

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

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

3-12. 分類の評価指標

教師データ

予測結果

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

Confusion Matrix:

	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	

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

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

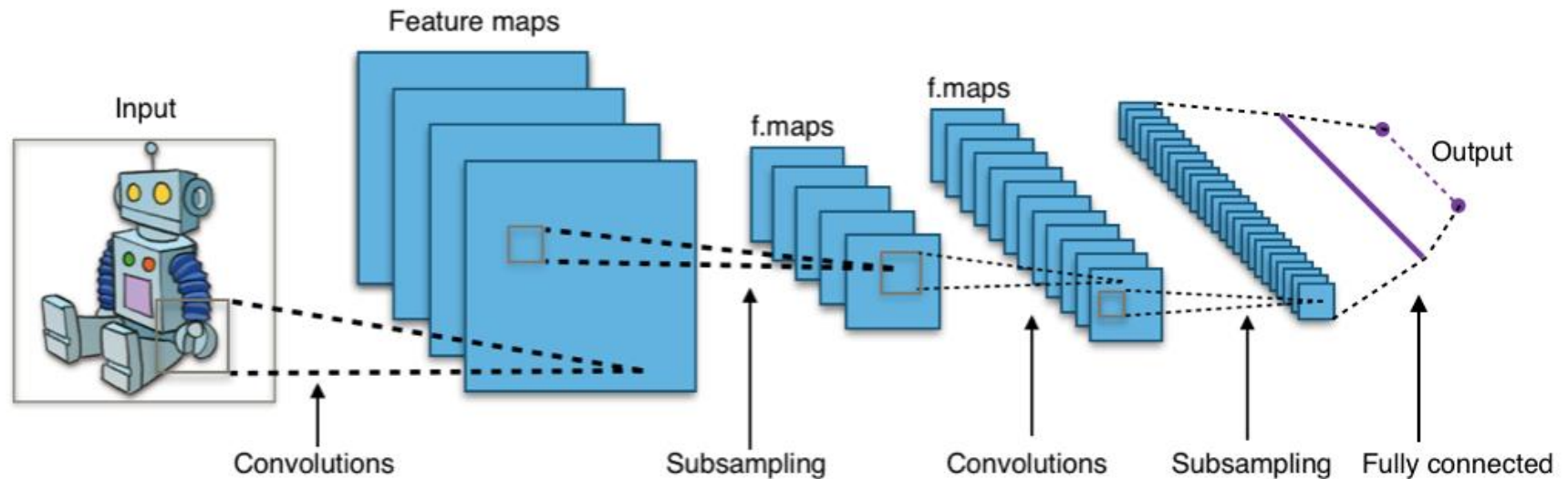
F-Measure (F尺度):
RecallとPrecisionの調和平均
(逆数の算術平均の逆数)

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

第4章 Convolutional Neural Network

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

□ CNNの一般的な構造



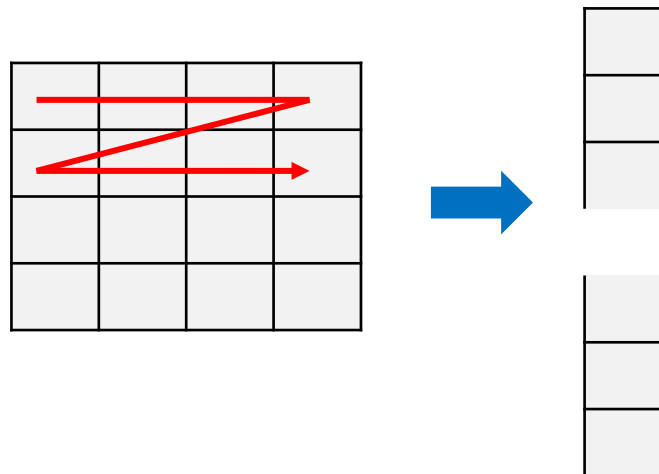
https://en.wikipedia.org/wiki/Convolutional_neural_network

□ 画像データの処理

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

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

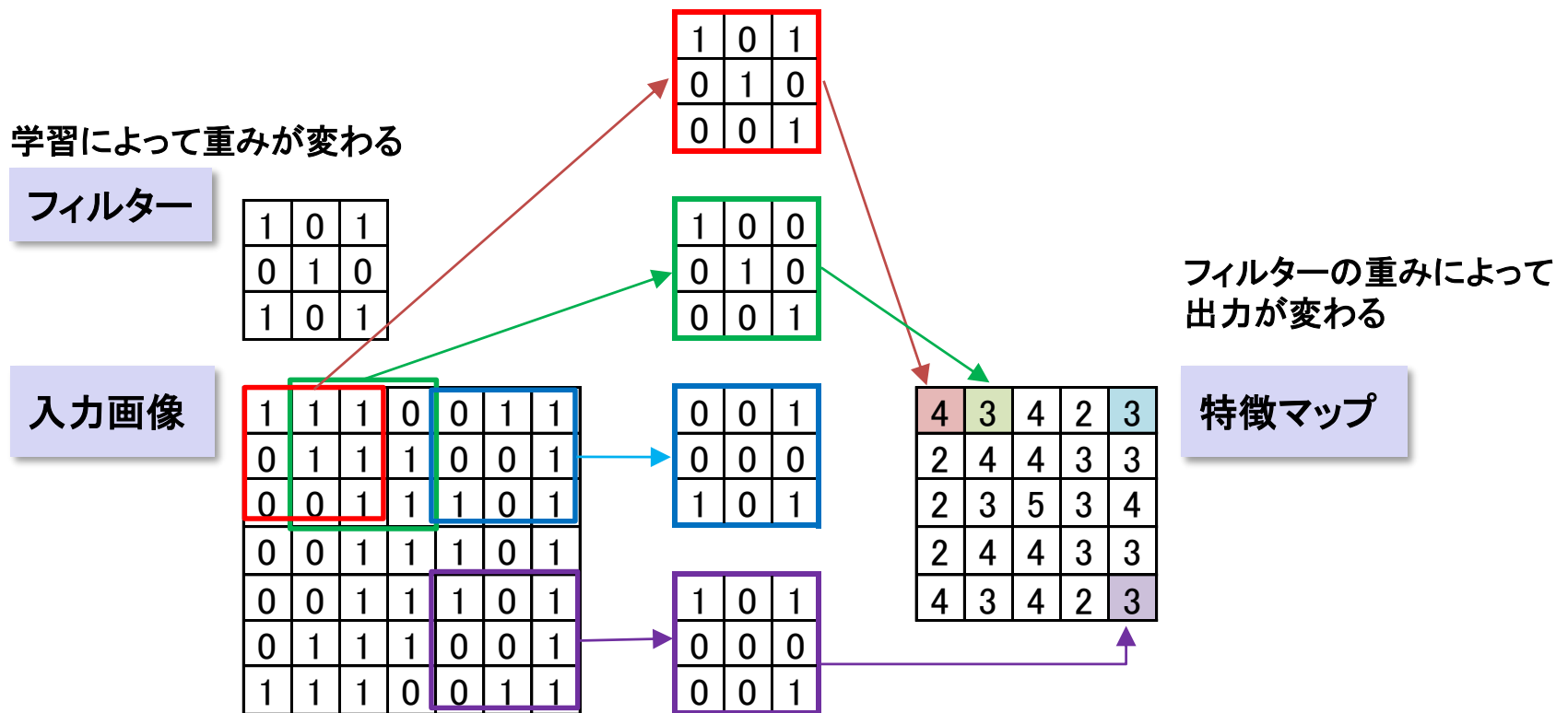


4-3. 畳み込み (Convolution)

- 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

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

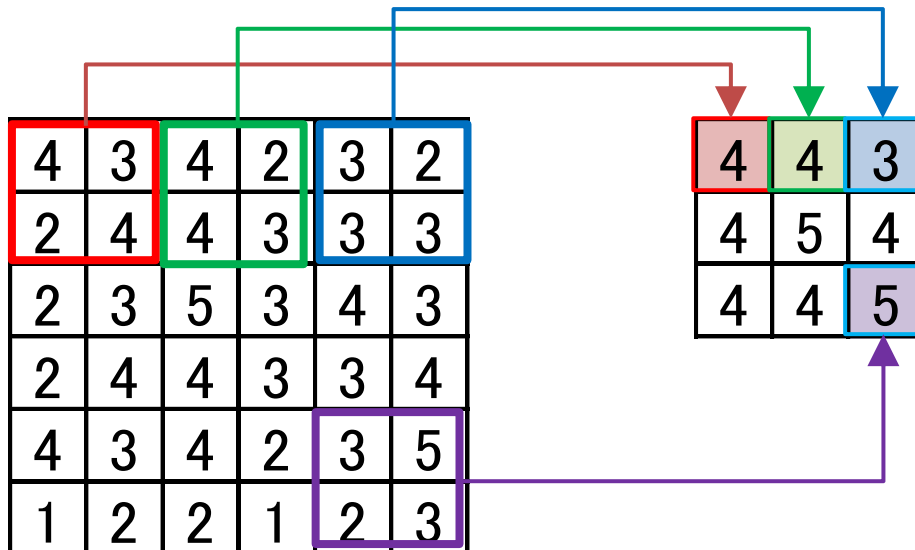


4-4. プーリング (Pooling)

- 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。

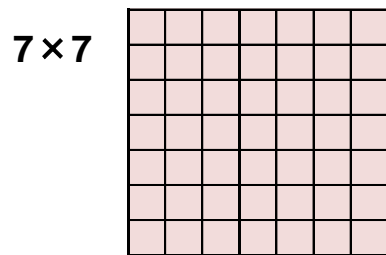
プーリング (Pooling)

- 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

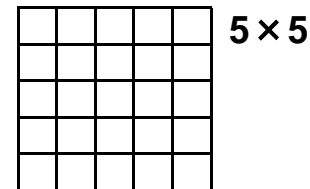


4-5. パディング (Padding)

- 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

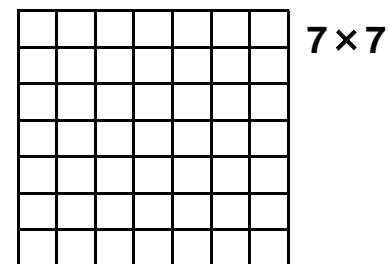


カーネルサイズ3×3
ストライド横1、縦1で
畳み込み



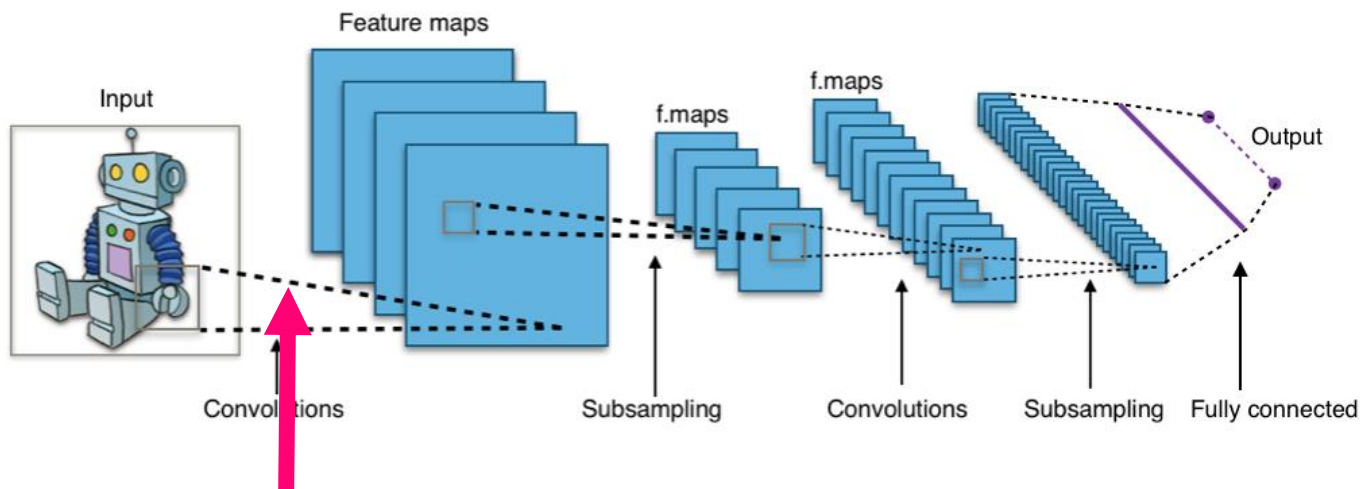
大きさ1の
ゼロパディング
(9×9)

0	0	0	0	0	0	0	0	0
0								0
0								0
0								0
0								0
0								0
0								0
0								0
0	0	0	0	0	0	0	0	0



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

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、最も有名なものが、Automatic Image Colorization



4-8. 高度なモデル: GAN

Generative **A**dversarial **N**etworks: 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 卷 (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.**

第5章 RNN, LSTM, AE

5-1. RNN: Recurrent Neural Network

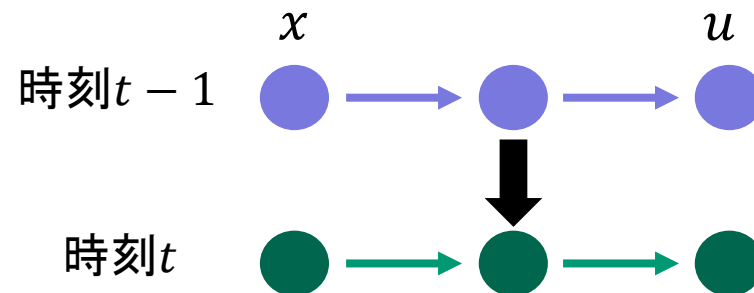
再帰型ニューラルネットワーク

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

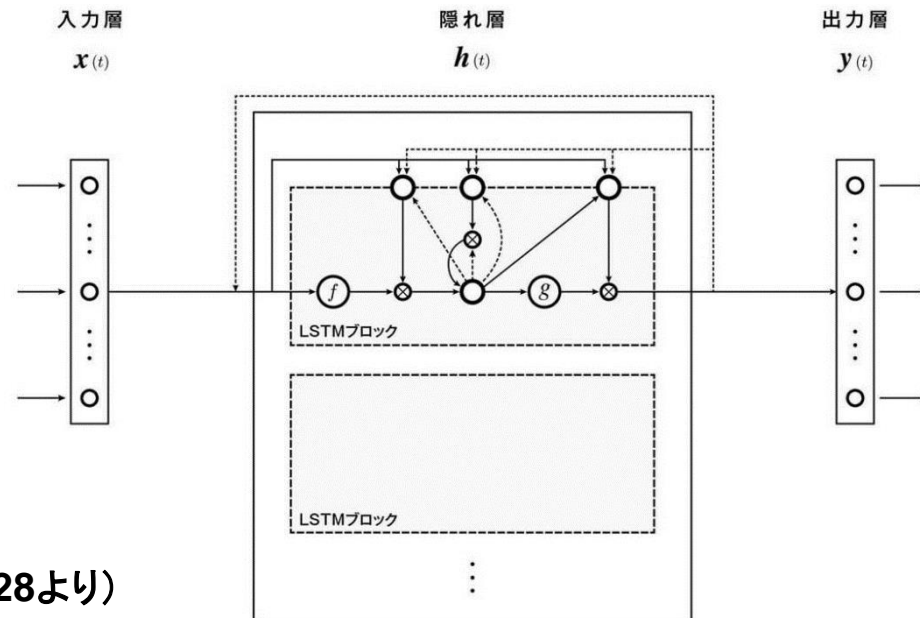
過去の間層の出力 $u(t-1)$ には、再帰的に過去の状態が全て反映されていることになる



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

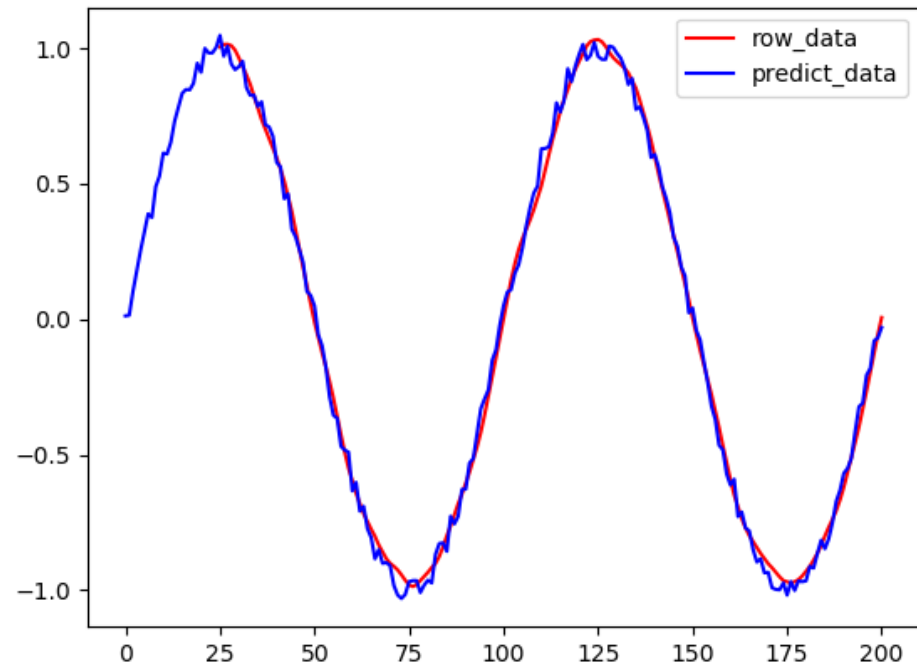


(巢籠, “詳解ディープラーニング”, p.228より)

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

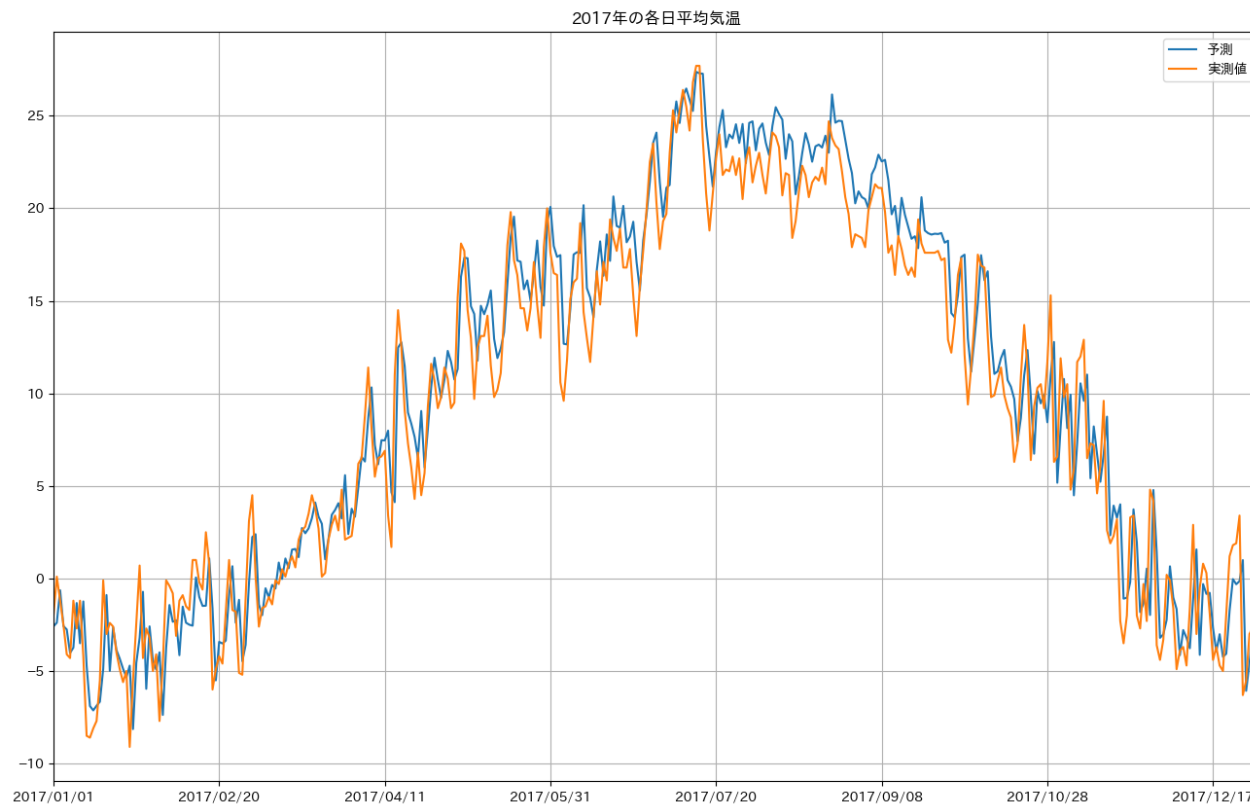
教師信号の正解データ：1976/1/1

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

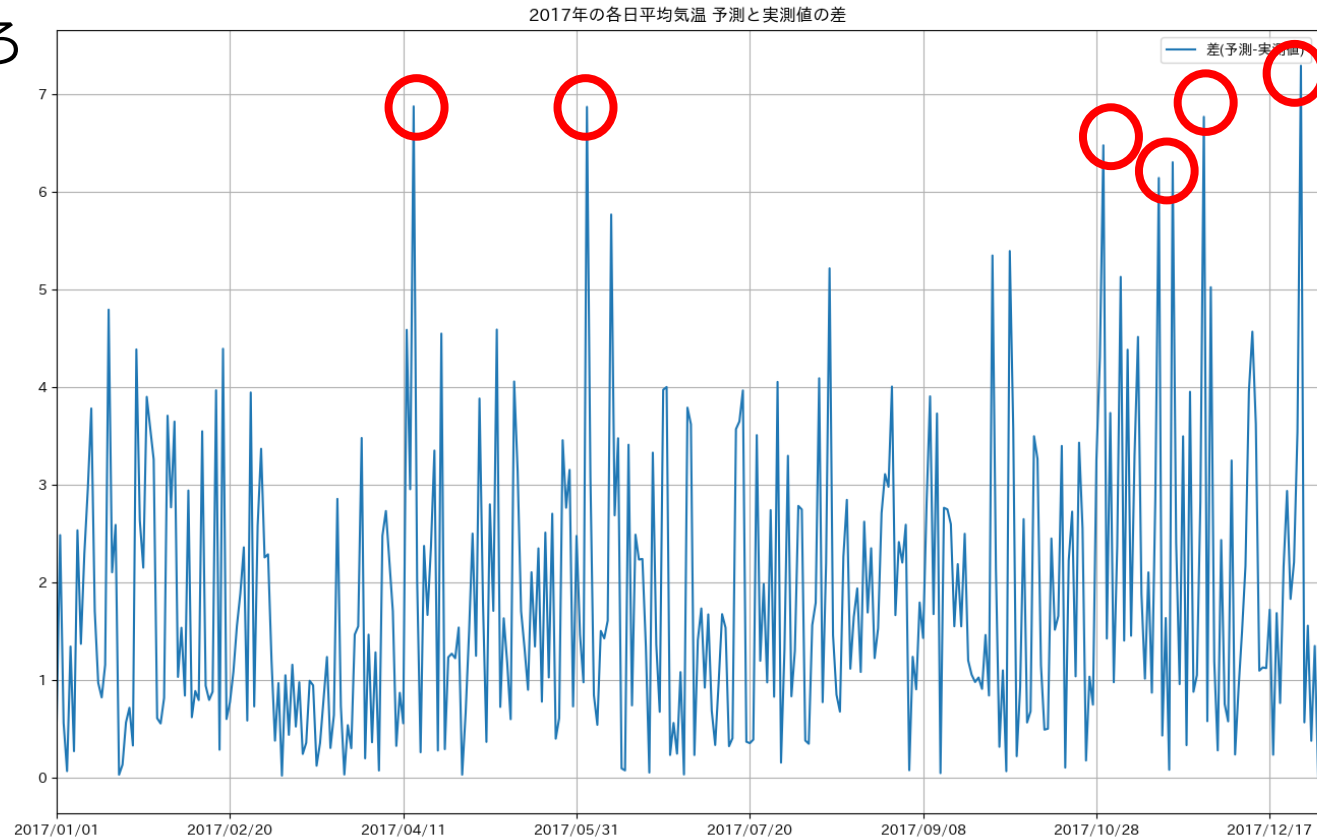
一見よさそうだが……



気温の予測：予測と実際の差異

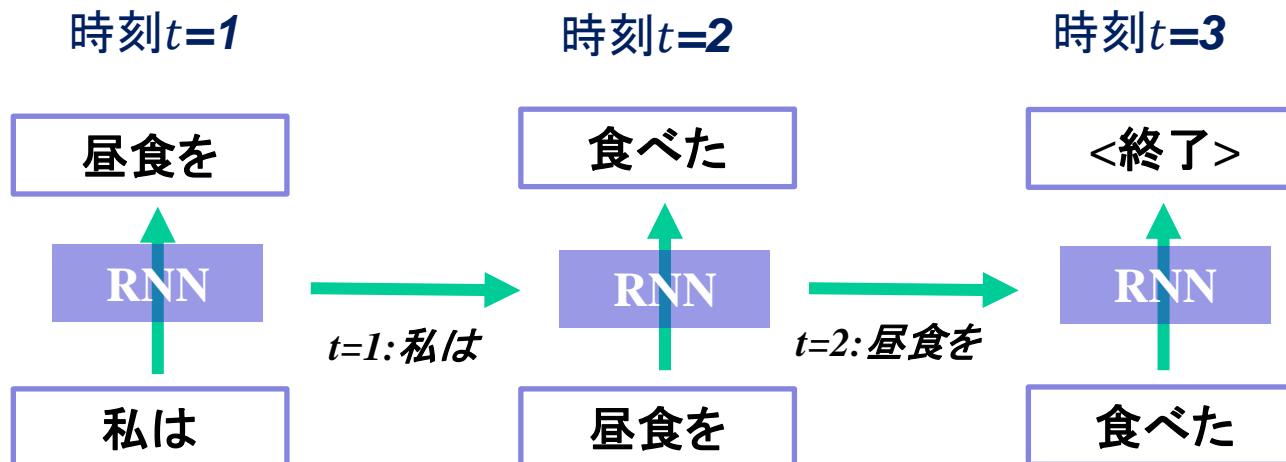
ところどころ
差が大きい

結果：
平均 1.9度
最小 0.0度
最大 7.3度



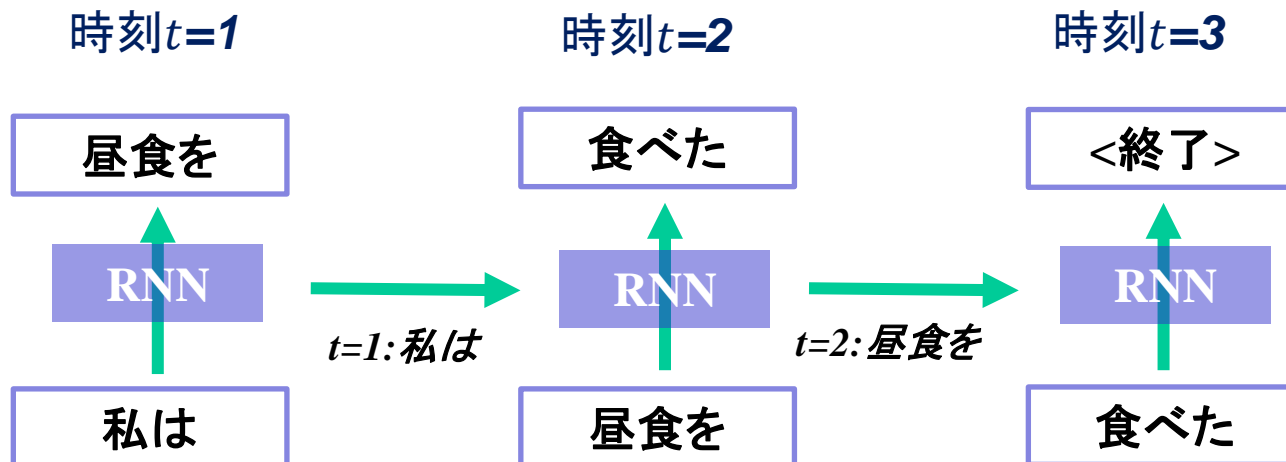
5-4. 時間軸と自然言語

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

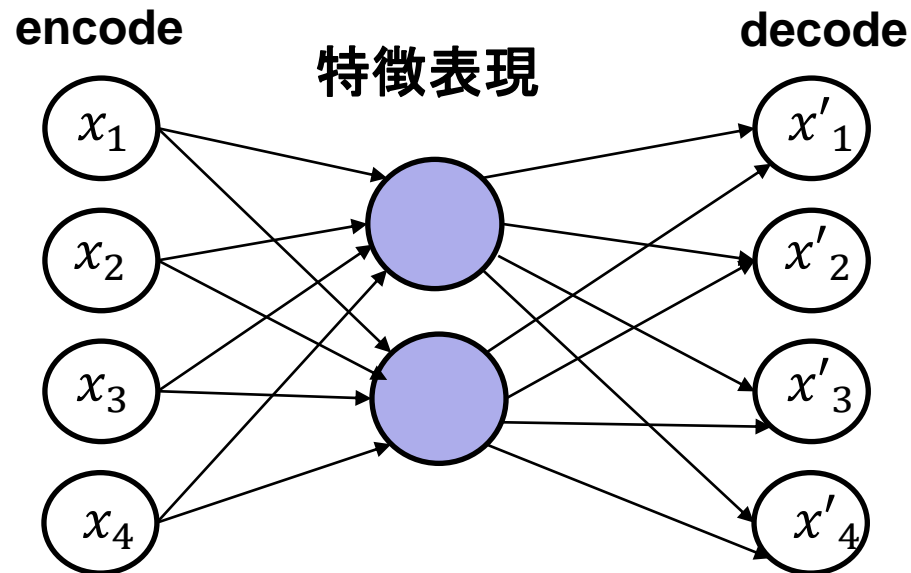


5-5. Auto Encoder

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

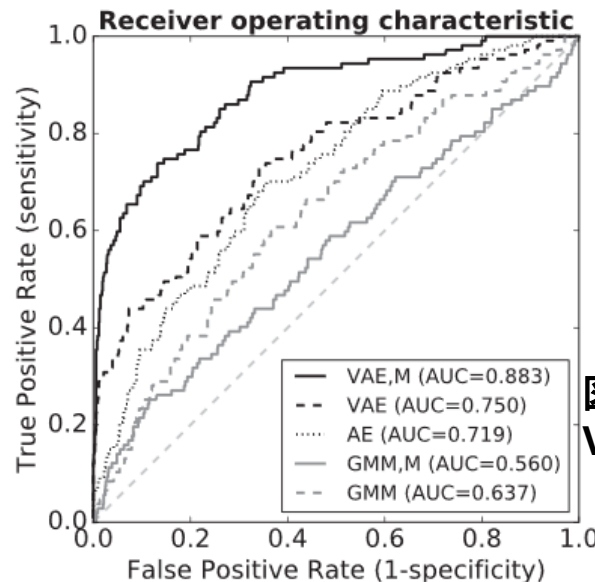
教師データがない = ラベル付けがない

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

6-1. 画像データの増強 (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

6-2. Kerasによる学習画像生成と画像分類演習

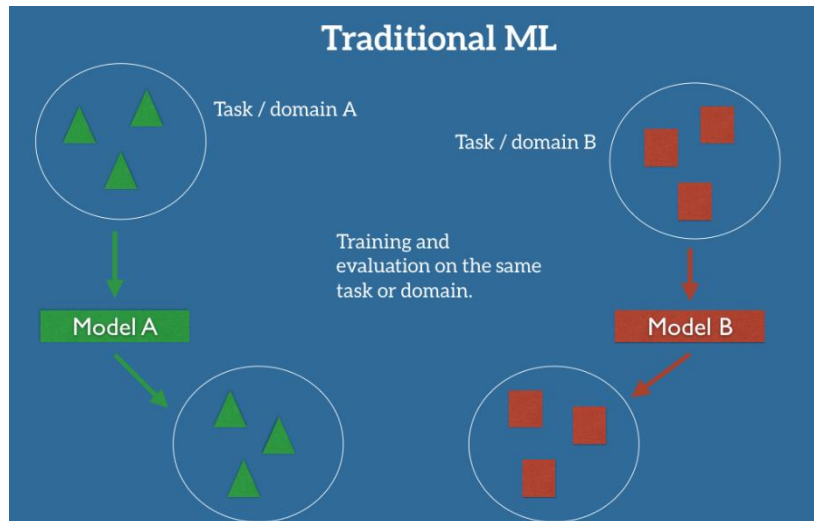
- 内容は講座当日に配布する別紙を参照してください

第7章 転移学習

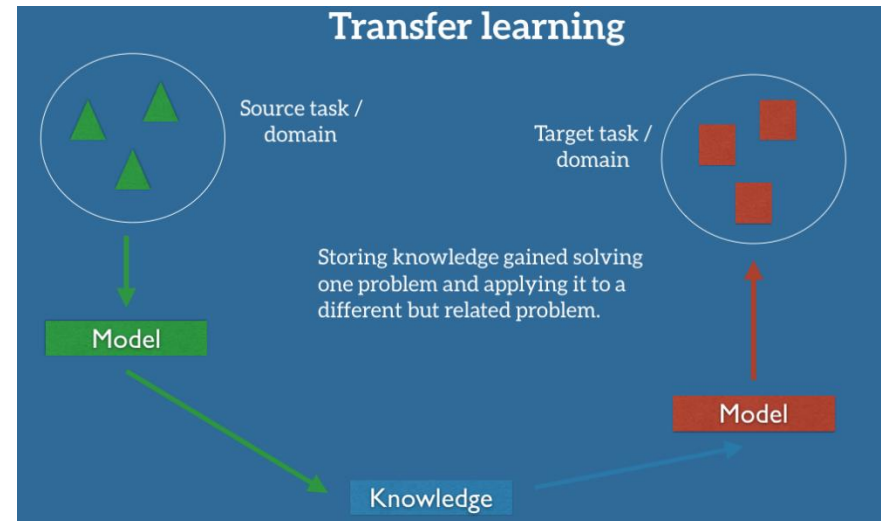
7-1. 転移学習

- 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

- 従来の機械学習



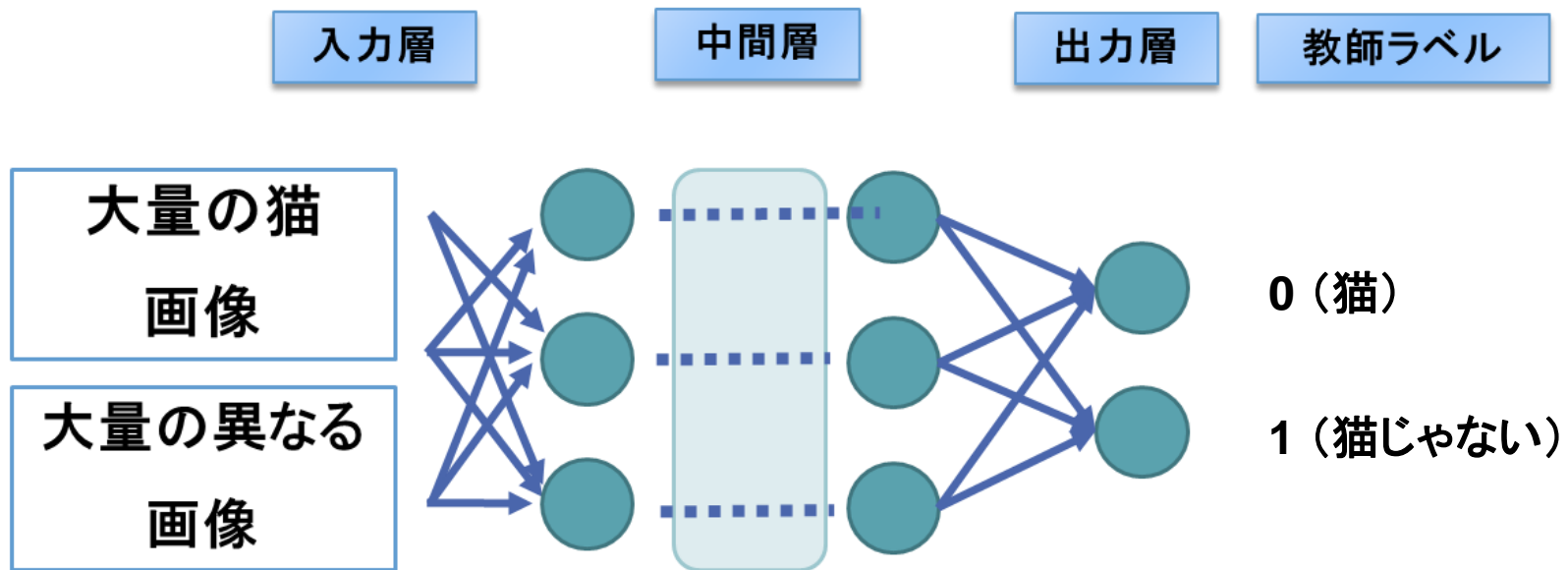
- 転移学習



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

7-2. 学習済みモデルの利用

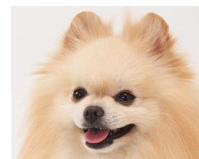
画像分類の学習



予測時の
出力層の値



0.2
0.8

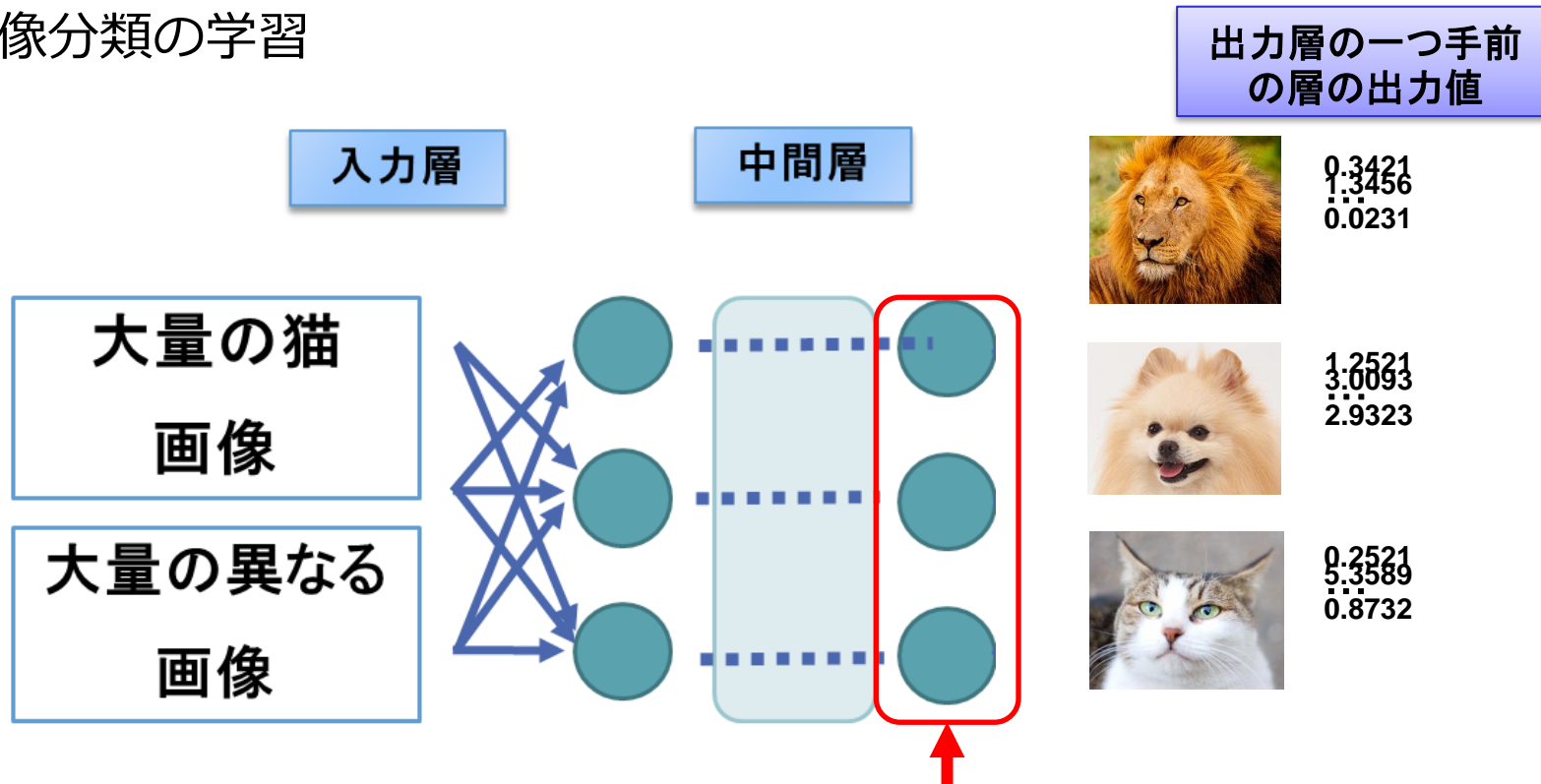


0.05
0.95



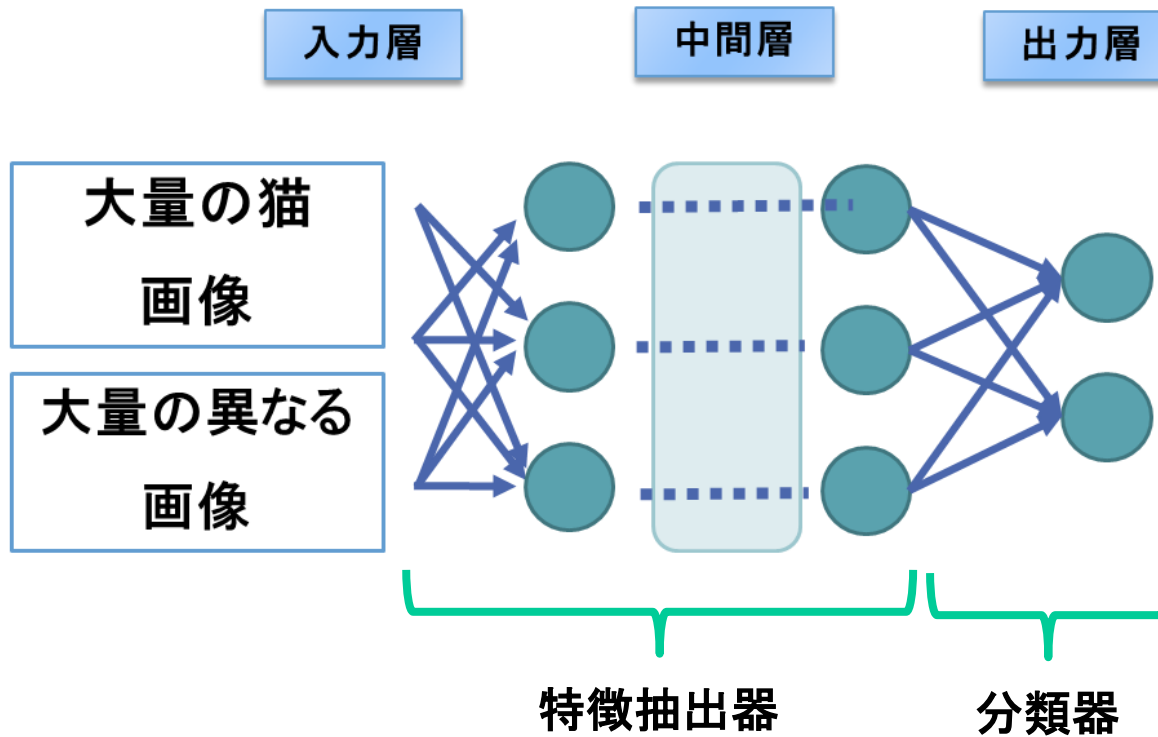
0.93
0.07

画像分類の学習

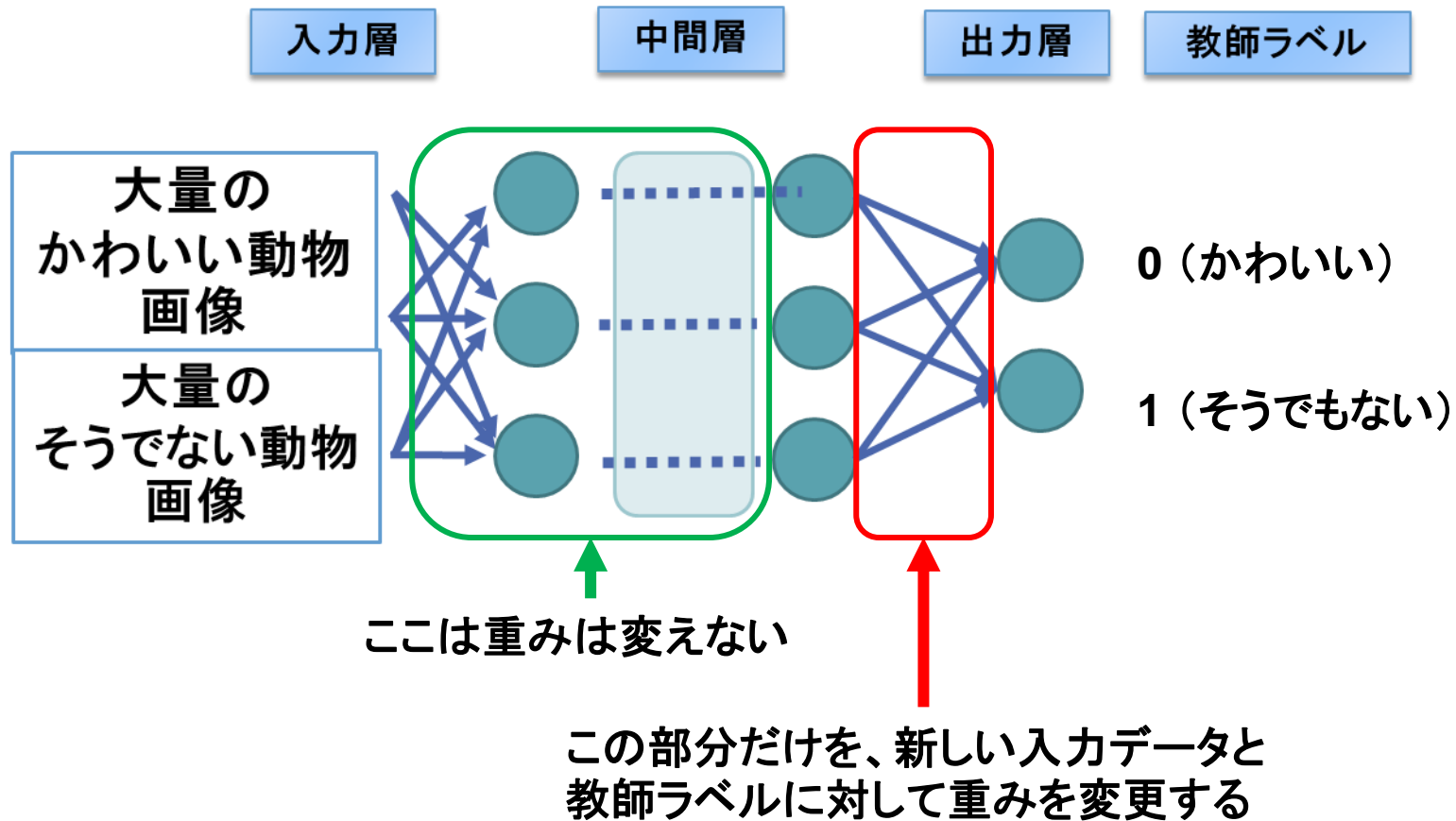


出力層の一つ手前の層(全結合層の場合が多い)で、
 入力はベクトルデータに変換される。
 そのベクトルデータは、入力の特徴を表す特徴ベクトルとなっている

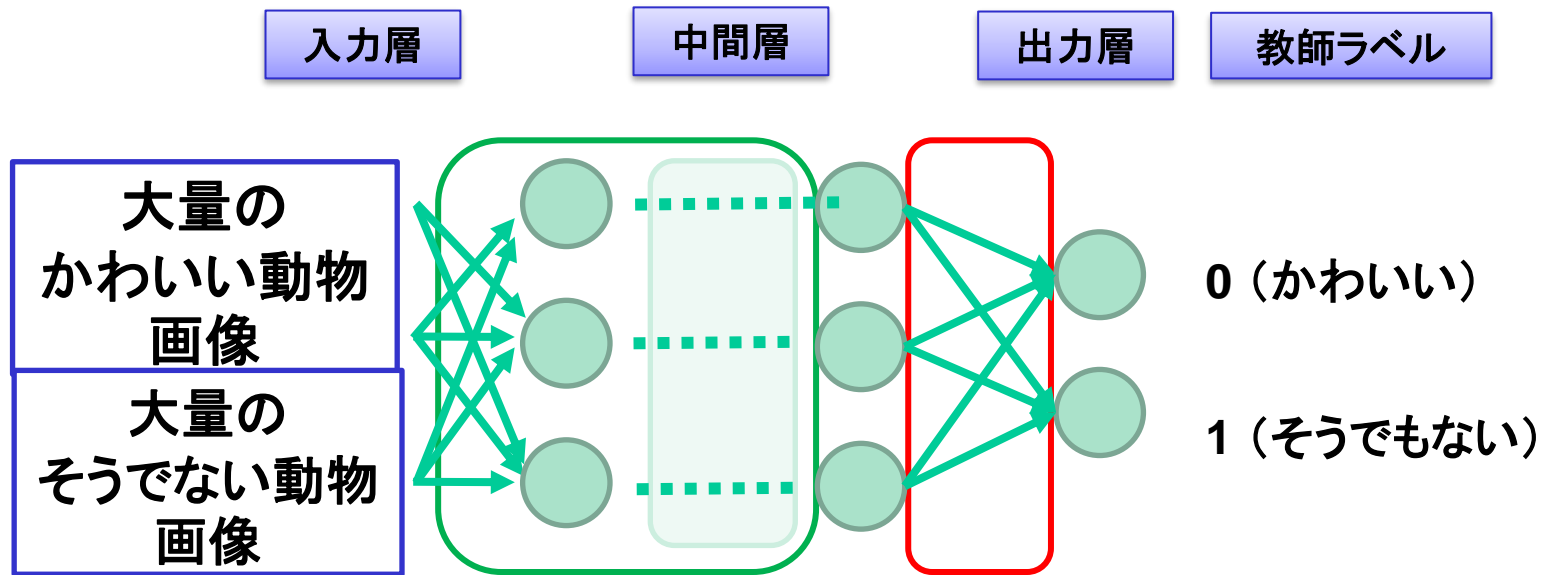
ニューラルネットワーク全体を、特徴抽出器と分類器に分けて考える






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 0.95	0.87 0.13	0.62 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



AI基礎講座

★第1章：人工知能概論

- 1-1. 人工知能とは
- 1-2. ディープラーニング（深層学習）
- 1-3. 弱い人工知能（AI）と強い人工知能（AI）
- 1-4. 人工知能の利用動向
- 1-5. 今後の展望
- 1-6. 制度的課題：知的財産の観点

★第2章：ディープラーニングの基礎知識

- 2-1. 人工知能の手法一覧
- 2-2. 統計的手法
- 2-3. 推論と探索
- 2-4. エキスパートシステム
- 2-5. 機械学習



第1章：人工知能概論

■ 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。

人工知能学会は、人工知能の研究には二つの立場があると述べています。立場の違いをこのように定義してよいか、また、これらの立場は本当に異なるのかということについて議論の余地がありますが、1つは人間の知能そのものを持つ機械を作ろうとする立場、もう1つは人間が知能を使ってすることを機械にさせようとする立場です。そして、実際の研究のほとんどは後者の立場で進められています。ですので、人工知能の研究といっても、人間のような機械を作っているわけではありません。それでは、実際にどのような研究が行われているのか、その例をみていきましょう。

[エキスパートシステム]

専門家の知見をルールとして蓄積し、推論の手法を用いて問題を解決するシステムです。

[音声認識]

マイクに向かって話した内容をコンピュータに理解させる研究です。カーナビゲーションなどのシステムで実用化されています。車内などの限定された状況以外での認識を可能にしたり、誰が話しているのかを特定する研究などに発展しています。

[画像認識]

カメラなどで撮った内容をコンピュータに理解させる研究です。コンピュータ内にある絵の内容を理解させる画像理解と、絵の明るさや色調（例えば、デジタルカメラのセピア調など）を変えたりする画像処理とに大きく分けられます。画像処理は実用化されていますが、画像理解はまだ研究段階です。

[機械学習]

観測センサーやその他の手段で収集されたデータの中から一貫性のある規則を見つけ出そうとする研究です。数学の統計の分野と強い関連があります。また、機械学習はAIの他のほとんどの分野で利用されています。

[自然言語処理]

文章に何が書かれているか、その意味内容をコンピュータに理解させる研究です。音声認識や情報検索の分野に応用されています。

[情報検索]

蓄積されたデータの中から人間が必要とするものを見つけ出すための技術です。WWWの検索エンジンなどで活用されています。

[推論]

いろいろなルールを統合して矛盾のない答えを導き出すための手法です。最も基本になるのはアリストテレスの三段論法というものです。これは「ソクラテスは人間である。人間は死ぬ。よって、ソクラテスは死ぬ」という

三段階で結論を出すものです。

[探索]

データの集まりから条件に合うものを見つけ出す手法です。データの数が多く、条件が複雑なので様々な工夫が必要になります。機械学習や推論の基盤となる技術です。

[データマイニング]

データベース技術と機械学習が結びついた技術で、大量の整理されていないデータから役に立つと思われる情報を見つけ出す手法です。例えば、ネット上で買い物をする際、あなたの趣味に合ったおすすめ品が示されることがあると思います。これは、今までの買い物のデータをもとに顧客の好みをデータマイニングによって調べています。

[ニューラルネット]

生物の神経をもとにした手法です。機械学習の有力な手法として発展し、AIの各分野で活用されています。

人工知能とは

- 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

(松尾豊：『人工知能は人間を超えるか』より)

レベル1の単純な制御プログラムは、自ら学習するような機能を持っておらず、従来からある単純な制御プログラムです。例えば、気温の変化に合わせてエアコンのON、OFFを制御するようなプログラムはレベル1に分類されます。制御できる範囲も狭く、実際には人工知能ではありませんが、人工知能搭載等の謳い文句で発売されます。

レベル2の古典的な人工知能は、大量のデータや情報などを読み込んでおき、条件分岐により多岐に渡る制御を行うことができるプログラムを指します。これも、人工知能が自ら学習するような機能はありません。なので、条件分岐については、人間がこのような条件ならこのような制御を行うと予めプログラムしておかなければなりません。エキスパートシステムと呼ばれる人工知能を利用して専門的な知識を扱えるようにしたシステムもここに該当します。

レベル3の機械学習を取り入れた人工知能は、Googleに代表される検索エンジンや、ビッグデータの解析、手書き文字の認識などに利用されています。この人工知能では、人間が特徴のある入力（学習データ）を用意する必要があり、事前にその特徴のある入力と出力の関係からパターンやルールを学習します。そして学習したパターンやルールを使用し、新しい入力に対する出力を導き出します。

レベル4のディープラーニングを取り入れた人工知能は、入力の特徴を数値化した特徴量自体を学習することができます。これにより、レベル3で人間が入力を用意していた工程を機械が担うこととなります。人の顔から年齢や感情を判定する技術や、車の自動運転などの技術に活用されています。

人工知能とは

■ 人工知能 (Artificial Intelligence, AI) とは、人工的にコンピュータ上などで人間と同様の知能を実現させようという試み、あるいはそのための一連の基礎技術を指す (wikipediaより)

■ 様々なアルゴリズムで上記のことを実現しようとしている

機械学習

- ニューラルネットワーク (今回のテーマはここ)
- 強化学習、遺伝的アルゴリズム、サポートベクターマシン (SVM)

データマイニング

- 主成分分析、クラスタ分析

ゲーム理論

その他

今回のAI講座ではニューラルネットワークについて学習しますが、ニューラルネットワークにかかわる機械学習の分類である、教師あり学習、教師なし学習、強化学習について説明します。

教師あり学習は、入力に対して正しい出力が用意されている学習データを使用し、入力と出力の関係を学習するものです。学習した入出力の関係性から予測や分類を行う際には、教師あり学習が利用されます。教師なし学習は、入力のみ与えられ正しい出力は用意されていません。それは正解が分からないことを意味しますから、入力から特徴量を求めることで、正解の分からない未知のデータに対する予測や、クラスタリングに利用されています。ニューラルネットワークは教師あり学習、教師なし学習どちらを行う手法としても活用できます。

強化学習は、機械自身が成功と失敗を繰り返しながら学習していく方法です。囲碁で有名になった人工知能「AlphaGo」は、この強化学習を採用しています。どのように機械が学習していくかという、例えば、ロボットを歩かせたい場合、今までは人間がロボットの姿勢を制御するプログラムや体の動きを制御するプログラムを作成し、失敗したら人間が再度プログラムを作り直すというように繰り返していました。しかし強化学習では、ロボットを歩かせてその動きによって、ロボットに対して報酬を与えるようにします。1m歩いたときより、10m歩いたときの報酬を高くします。これを繰り返すことでロボットはより大きな報酬を得られる方法を探します。このように、機械自身が価値を最大化するような学習を強化学習といいます。

■ ニューラルネットワーク (wikipediaより)

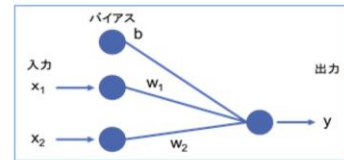
人工神経を相互接続したもので計算を構造化し、コネクショニズム的計算技法で情報を処理する

入力と出力の間の複雑な関係をモデル化する

データのパターン認識や観測された変数間の未知の同時分布における統計的構造を捉える

■ ディープラーニング

多層のニューラルネットワークによる機械学習手法 (詳細は後述)



$$y = w_1x_1 + w_2x_2 + b$$

単純パーセプトロンの例

コンピュータは非常に高速に計算をすることができますが、人間のように見たものが一体何なのかを認識することは不得意です。この問題を解決するために、脳の神経回路網を模してニューラルネットワークが生まれました。

生物の脳の神経細胞であるニューロンは、ネットワークを形作りニューロン同士の間で信号を伝えています。1つのニューロンに焦点を当ててみると、あるニューロンは複数のニューロンから信号を受け取ります。このとき、受け取る信号は大きさがまちまちであるため、同じ大きさの信号を複数受け取るわけではありません。受け取った信号の和が、ある一定の大きさを超えた時、他の複数のニューロンに対して信号を送ります。このプロセスが何度も繰り返されることで、脳では色々な情報を認識されています。このようなニューロンの集合体としての働きを

、数学的モデルにしたのがニューラルネットワークです。また、脳の神経回路網をコンピュータ上でモデルとして再現しようという考え方をコネクショニズムといいます。

- 入力と出力の間の複雑な関係をモデル化する
 - 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
- ディープラーニングの欠点
 - 大量の学習データが必要
 - 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

機械学習を用いると、入力と出力の間の複雑な関係をモデル化することができます。分類とは、例えば、写真に写っている動物が猫なのか犬なのか分類することを指します。分類では、学習データに正解データ（ラベル）が付いており、コンピュータは入力から割り出した出力を正解データと比べながら、入力と出力の関係をモデル化していきます。つまり、前述した教師あり学習に分類されます。正解データは教師データともいわれます。回帰とは、明日の天気や株価等、未知のデータを予想することを指します。回帰で用いられる学習データは連続したデータであり、天気や株価であれば過去の時系列データを使用します。この連続した学習データから入力と出力の関係をモデル化し、未知のデータを予想します。学習に用いられる連続したデータには答えがあるため、こちらも教師あり学習に分類されます。

ディープラーニングでは「データの前処理が必要ない」ということが利点です。人間の手によって行われる作業が機械に置き換わり、前処理のためにかける時間や専門知識が以前よりは少なくなったことを意味しています。このように人間の手間が少なくなった一方で、今度は膨大な量のデータが必要になりました。近年ではIoT等を含めセンサーデバイスが普及し、以前よりデータは取得しやすくなってきています。

■ きのこの山とたけのこの里の分類（Interface 2017年8月号より）

■ 従来の機械学習（SVM）

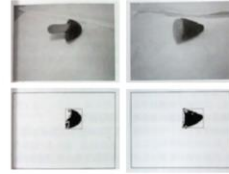
入力対象の何を抽出し、どうやって解析するかを人間が考える

1. きのこの山は柄の部分が細いので白い部分が多そう
2. きのこの山は傘の部分に凹凸が多そう

画像取得→2値化（チョコの部分抽出）

1. バウンディングボックス内の白と黒の面積比の計算
2. 黒い部分の頂点を検出し頂点数を求める

分類対象を1.と2.の数値データで表している



(Interface 2017年8月号 p.33)

■ ディープラーニング

取得した画像を入力データとする

- 画像の画素値が入力データ



(Interface 2017年8月号 p.33)

従来の機械学習と、ディープラーニングの違いに、人間が特徴量の抽出をしなくてよくなったことが挙げられます。図の例のように、入力対象の何を抽出し、どうやって解析するかというのは、人間の経験と知識により決めなければなりませんでした。このような技術の特徴量エンジニアリングといいますが、これはとても専門的な知識を必要とする技術であるため、時間と手間がかかり、機械学習の普及にも悪影響を及ぼし、人工知能の実用化が進まない原因にもなっていました。しかし、ディープラーニングでは、膨大な量のデータから入力対象の何を抽出し、どのように解析するかというパターンやルールをコンピュータ自身が見つけ出します。つまり、特徴量をコンピュータ自身が学習から見つけ出します。その結果、未知のデータからも答えを導き出すことが可能になるのです。

ディープラーニング（深層学習）

- 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

All Rights Reserved, Copyright© UHD2018

10

ニューラルネットワークの詳しい仕組みは後述しますが、ここでは代表的なディープラーニングの種類について紹介します。

ニューラルネットワークの中でも、特に近年脚光を浴びているのが畳み込みニューラルネットワークです。その特徴として、ネットワークの仕組み自体は簡潔であるにもかかわらず、複雑なパターン認識問題に対応することが可能であることが挙げられます。ネットワークの仕組みが簡潔であるということは、入力から出力を判断するまでの計算が少ないことを意味します。

一般的なニューラルネットワークの手法は、ある任意の独立した時間において、入力から出力を判断します。しかし、未来予測等を行うには任意の時間だ

けでなく、時系列を考慮して判断する必要があります。そのために開発された手法がリカレントニューラルネットワークです。リカレントニューラルネットワークでは、前の時間に得た情報を次の時間に反映させることで最終的な出力を得ることができます。

オートエンコーダとは、与えられた入力の特徴を獲得するための手法です。オートエンコーダでは、入力の次元を減らすことで特徴を抽出します。例えば、4つの入力を受け、2つの出力を得たとします。今度はその2つの出力を入力とし、4つの出力を得たとします。この時、最初の4つの入力と、最後の4つの出力がほぼ似たような値を得るようであれば、中間で得た2つの出力は、最初の4つの入力の特徴を表しているものといえます。この特徴を表す2つの出力の部分抽出するのがオートエンコーダです。

■ 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低い）
- 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）

CNNの実用例としては、コンピュータやセキュリティにおける顔認証システムや、写真に写ってしまった第三者の顔や車のナンバープレート等に自動でボカシ処理を入れるシステムなどが挙げられます。実用例からも分かるように、画像の解析や分類に強みを発揮しています。今後の発展として、リアルタイムで映像を解析・認識するシステム（交通事故防止のために道路標識を自動認識するなど）の研究や、農業において収穫物の出来を判定するシステムの研究も行われています。

RNNでは、音声認識を時系列に行うことで、過去に使用された単語や文脈から話の内容を認識するシステムの研究や、大規模な人流の計測を行うことで、災害時の安全な避難・誘導などに役立てるような研究も進んでいます。

■ 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、強化学習など今後も様々な手法があらわれると思います。

AEの活用には、入力された低解像度の画像について局所的な特徴を抽出し、それをもとの高解像度の画像に復元する技術や、入力として与えられた画像の特徴から、似たような特徴を持つ画像の生成技術などが挙げられます。このような技術をより発展させることで、過去の芸術家が用いた手法の特徴を抽出し、芸術品としての特徴を持った作品の生成も期待されています。

CNN、RNN、AEと3つに分けて紹介しましたが、これらを組み合わせた研究も進んでいます。

ディープラーニング（深層学習）

- 実用的な計算速度が可能となった
軽い学習アルゴリズムが登場した
GPUの利用による計算能力の向上
- 大量の学習データ取得が容易になった
センサーデバイスの普及
オープンデータの普及
- オープンソースなフレームワークの登場でソフトウェアの実装が簡単になった
ディープラーニング用のライブラリが公開され、簡単に実装することができるようになった
しかも実際にGoogleやSonyが社内で行っているもの
個人レベルでディープラーニングをすぐに扱うことができる

ディープラーニングの普及には、大量の学習データを取得できるか、適したアルゴリズムを作成できるか、対応できるハードウェアを用意できるか、などのハードルがありました。それらが近年解決されようとしているため、ディープラーニングが注目され始めたのです。

計算速度の向上例として、CPUではなくGPUの活用が挙げられます。GPUは、Graphics Processing Unit の略で、画像処理演算に特化しています。ディープラーニングに必要な計算能力は画像処理の計算能力と似ているため、GPUの利用が注目されるようになりました。また、クラウドコンピューティングにより同時に複数台のコンピュータを用いて計算できるようになったことも、計算速度の向上に寄与しています。

大量の学習データを収集する際、人間の手では限界がありますし、膨大な時間がかかってしまいます。これに対して、センサーを備えたデバイスが自動でデータを収集するようになったことで、学習データの収集効率は以前より格段に向上しました。また、大学や研究機関等がオープンデータとして誰でも使用できるデータセットを公開し始めたことも、ディープラーニングを活用しやすくなった一因といえるでしょう。

以前は一から作成しなくてはいけなかったプログラムも、ライブラリの充実によって比較的簡単に作成することができるようになりました。3章で改めて説明しますが、実際に企業が使用しているものも個人で扱うことができる場合があります。

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

- 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)

- 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

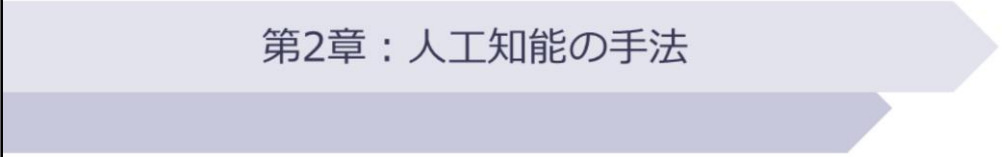
(AI白書2017より)

- 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さかん

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。

学習済みモデルの保護

- 学習済みモデルはプログラムとパラメータで構成されており、著作権法上の「プログラムの著作物」に該当するかの議論がある
- 著作物に該当しなくても、特許法の要件を満たすならば発明として保護されるとされている
- 著作物や発明に該当しなくても、不正競争防止法上の秘密管理性、有用性、非公知性などの要件を満たせば、「営業秘密」として保護されるとされている
- 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」は著作権による保護が困難
→特許権や契約による保護が議論されている



第2章：人工知能の手法

2-1. 人工知能の手法一覧

- 統計的手法
- 推論と探索
 - 探索木
- エキスパートシステム
 - If-Thenルール
- 機械学習
 - k-近傍法（教師あり）
 - 決定木、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）
 - その他にもいろいろとあります
 - ✓ 遺伝的アルゴリズム
 - ✓ ベイジアンネットワーク
- 深層学習（ディープラーニング）・・・後に詳述

扱う問題によっては、
従来手法も十分に有効

知能をより具体的に、「何かを判断するために場合分けする」ことと考えます。

2-2. 統計的手法

- 回帰・重回帰分析（回帰）
 - 回帰式により説明変数から目的変数を予測
- 判別分析（分類）
 - 判別式により目的変数を判別
- 主成分分析（次元圧縮）
 - 変数の共分散行列や相関行列の固有ベクトルを使って新たな変数を生成する
 - 変数の数が多いデータに対し、変数を統合してデータの傾向を表す
 - 新たな変数を生成することで、より少ない次元でデータを表す

回帰・重回帰分析と判別分析は教師あり、主成分分析は教師なしの手法です。

2-3. 推論と探索

□ 推論

- 人間の思考過程を記号で表現し、実行する

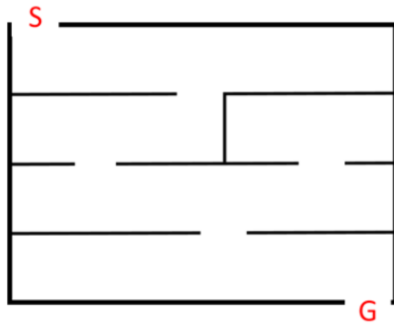
□ 探索

- 取り得る可能性がある選択肢を木構造（探索木）で表現し、目的のノードまでの経路を調べる

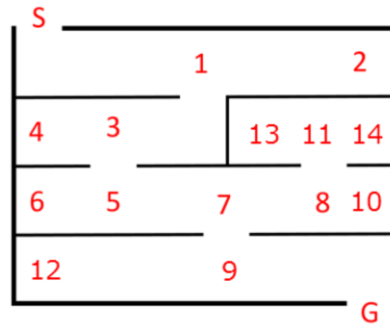
□ 迷路の問題表現

迷路を探索問題として表現する場合、分岐地点と行き止まりに記号を割り当て、経路を表す

迷路

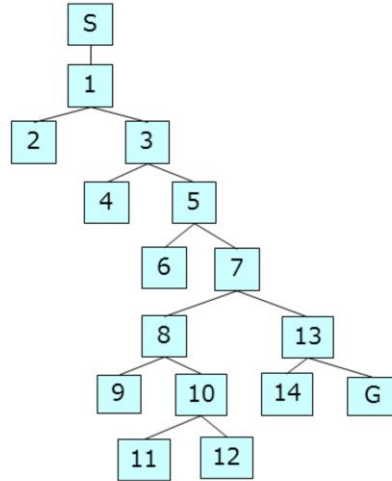
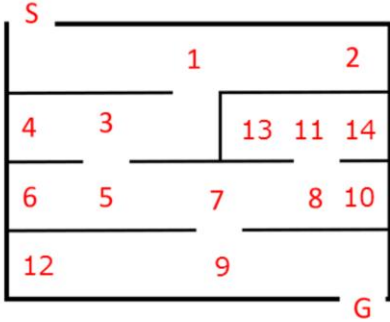


迷路の問題表現



■ 迷路の探索木

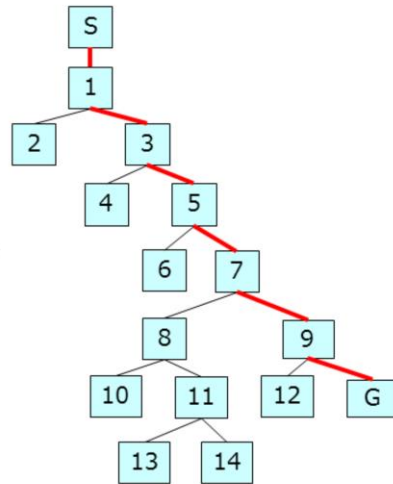
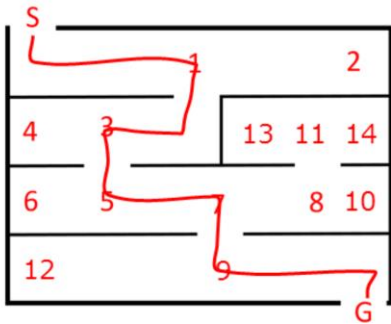
迷路の問題表現



探索木

- 幅優先探索、深さ優先探索などでゴールを探索する。

迷路の問題表現



- 盤面を評価するスコアを定義し、指せる可能性のある盤面すべてをノードとする探索木を生成する
- 次の指し手のスコアが良くなるようにノードを探索する
 - ミニマックス法：相手は自分のスコアが最小になるように、自分は自分のスコアが最大になるように指し手を探索する
- 序盤は可能性のある盤面が膨大な数になる
 - 枝刈り法：一定のルールに従ってそれ以上探索しないようにする
- 後半は盤面の可能性が少なくなるため、指し手の決定手法を変更する
 - モンテカルロ法：ランダムに指し手を決定することを繰り返し勝負を終了させる。勝負を終了させることを繰り返して、最も勝つ可能性が高い指し手を選択する。
- 盤面評価（スコア計算）手法など、他にも様々な工夫をしている

ここで挙げたものは代表的なものの一部であり、他にも様々な手法を組み合わせて実現しています。

盤面評価（スコア計算）は人間が工夫して計算方法を考えるため、対象となるゲームの知識が必要です

。

□ 盤面評価の手法の改良

- 2つの駒の位置関係（王将と金など）から、3つの駒の位置関係（王将と金と銀など）の方がより良く評価できる
- 盤面評価手法 = 特徴量表現の改良

□ 後半の指し手決定方法の変更

- 盤面評価スコアによる最善手探索よりも、ランダムな指し手でとにかく終了させることをたくさん繰り返し、勝率の高い指し手を選んだ方が効果的
- 指し手の選択肢が少なくなることで可能となる

2-4. エキスパートシステム

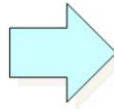
□ 専門分野の知識を取り込み、推論を行うことでその分野のエキスパートのように振る舞うシステム

□ If-Thenルール

➤ 条件を定義し、条件を満たせばその条件における判断を下す

□ 例

もし、
警報が発令されている
午前7時より前である
発令されているのは県内である



午前の授業は休講とする



これらの知識は手動で設定

エキスパートシステム

- 知識をコンピュータに与えるために膨大なコストが必要
 - 専門家にヒアリングをして知識を取り出すなど
- 知識（条件）の適切な管理が必要
 - 知識（条件）の数が増えるとお互いに矛盾したり一貫していなかったりする
- あいまいな条件を扱うための広い範囲の知識の記述が難しい
 - 「お腹に違和感がある」
 - お腹とはどの部分なのか？
 - ✓ 人間の身体構造の知識（お腹には胃、小腸、大腸などがある）が必要
 - 違和感とはどんな症状か
 - ✓ 鋭い痛み、鈍い痛み、張りがあるなど

処理としては単純な条件分岐だが、知識をどのように作るかが問題となっています。

ただし、処理が単純で、蓄積されたルールを人間が理解しやすいため、現在でも使われることがあります。

2-5. 機械学習

□ 教師なし学習

- k-means法 (クラスタリング)
- 自己組織化マップ (次元圧縮)

□ 教師あり学習

- k近傍法 (分類、回帰)
- 決定木、ランダムフォレスト (分類、回帰)
- サポートベクターマシン (分類、回帰)
- ニューラルネットワーク (分類、回帰)

□ 強化学習 (時間の関係で今回は扱いません)

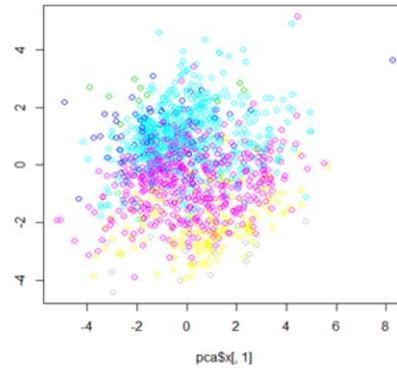
- 教師あり学習に似ているが、与えられた正解に対する価値を最大化するのではなく、その時点での環境における価値を定義し最大化する
- ある時点における環境における価値が大きくても正解に対する価値が大きいとは限らない

2-5-0. (参考) 主成分分析

- 主成分分析は教師無し学習と同じ目的の統計的手法
- 下の表のようなワインの成分とワインの品質のデータに対して、成分データでワインを特徴をあらわすことを考える

属性 (単位)
酒石酸濃度 (g/dm ³ g/dm ³)
酢酸濃度 (g/dm ³ g/dm ³)
クエン酸濃度 (g/dm ³ g/dm ³)
残留糖分濃度 (g/dm ³ g/dm ³)
塩化ナトリウム濃度 (g/dm ³ g/dm ³)
遊離亜硫酸濃度 (mg/dm ³ mg/dm ³)
総亜硫酸濃度 (mg/dm ³ mg/dm ³)
密度 (g/dm ³ g/dm ³)
pH
硫酸カリウム濃度 (g/dm ³ g/dm ³)
アルコール度数 (% vol.)
ワインの味 (グレード)

第1主成分と
第2主成分で
プロット



点の色は品質の違いを表す

2-5-1. k-means法

- 教師なし学習
- 特徴空間の中で距離が近いものをおしを同じクラスとしてグループ化する
- 結果は初期クラス重心にかなり依存する
- 処理手順
 1. クラスタ数を決め、初期クラスタを設定する
 2. 各クラスタの重心を求める
 3. 各データと各クラスタ（の重心）との距離を計算し、各データに最も近いクラスタを割り当てる
 4. 2と3を繰り返す
- Rのkmeansなどで実装されている
- *k-means clustering* (wikipediaより) https://en.wikipedia.org/wiki/K-means_clustering

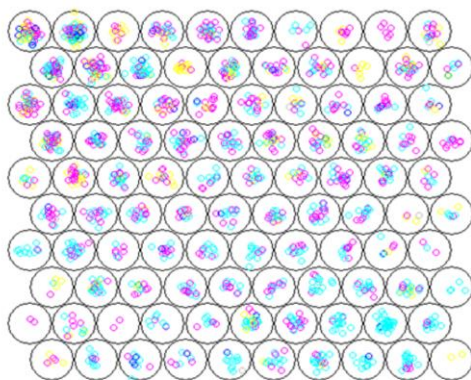
2次元での描画しかできないため高次元の実データではイメージを持ちにくいですが、英語版wikiの動画がわかりやすいので参照してください。

2-5-2. 自己組織化マップ

- 教師なし学習
- 入力データの類似度（ユークリッド距離など）の計算を繰り返して、類似度を基にデータをグループ分けする
- 入力層と出力層の2層で構成されるニューラルネットワーク
- 処理手順
 - 入力ノードと出力ノードに類似度が定義されている
 - 出力層のノードには近傍ノードが定義されている
 - 1. 出力ノードの重みベクトルをランダム化する
 - 2. ある一つの入力に対し、最も類似した出力ノード（BMU、Best Matching Unit）を見つける
 - 3. BMUとその近傍ノードの重みベクトルを、入力ノードの重みベクトル近づけるように変更する
 - 4. 2と3を繰り返す
- Rのsomパッケージ、kohonenパッケージなどで実装されている

■ 自己組織化マップの出力例（ワインの成分）

Positioning map



点の色は品質の違いを表す

2-5-3. k-近傍法 (k-Nearest Neighbor法)

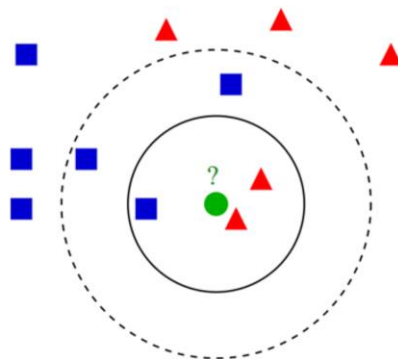
- 教師あり学習
- 特徴空間上で最も近い距離にある訓練データに基づいて分類や回帰を行う
- 線形分離が難しい場合でも分類可能
- 処理手順
 1. 正解がわかっているデータを基に、正解がわからないデータを分類する
 2. 未分類のデータに対して、k個の分類済みの近傍データを見つける
 3. 分類済みの近傍データの多数決などにより未分類のデータを分類・回帰する
- Rのclassパッケージなどで実装されている

おおまかに言うと、対象がなんらかの多変量データで表されていて、分類したいデータの近くに最も多い訓練データのクラスを採用することです。

回帰の場合は、近傍の訓練データのクラスの代わりに近傍データの平均や加重平均などを使います。

□ k-近傍法の例

- 青と赤は教師データ
- 緑はラベルを持っていないデータ
- この例では新しい緑のデータは赤と同じラベルと予測される



2-5-4. 決定木

- 教師あり学習
- 段階的にデータを分割していき、木のような分析結果を出力する
- 分析結果の解釈や実装が容易
- 複数の決定木を組み合わせることで、ランダムフォレストなどの強力なアルゴリズムに発展させることが可能
- 処理手順
 1. データを分割する基準を決定する
 - 最もきれいにデータを分割できる基準を選ぶ
 - 「情報利得」が最大となる基準 = 「不純度」が最小
 2. データを分割する
 3. 設定した基準になるまで1と2を繰り返す
 - 終了条件は、木の深さ、終端ノード数、ノードに含まれるデータ数、誤り率などを考慮する
- Rのrpartパッケージなどで実装されている

前述の決定木を自動的に作成する方法です。

決定木はルールが人間に理解しやすく、ルールの変更をすぐに反映できるという利点があります。

決定木

- **不純度** : ジニ不純度、交差エントロピーなど
- ノード t におけるジニ不純度 $J(t) = 1 - \sum_{i=1}^N p(i|t)^2$ 、
- $p(i|t)$: ノード t におけるクラス i の割合
- N : クラス数
- 一つのノードに様々なクラスが含まれていると不純度が高くなる

- **情報利得** : 分割のよさ (分割前の不純度 - 分割後の不純度)
- ノード X をノード $Y1$ と $Y2$ に分割した場合の情報利得を $IG(X, Y1, Y2)$ とする
- $IG(X, Y1, Y2) = E(X) - p(Y1) * E(Y1) - p(Y2) * E(Y2)$
- E : 不純度、 p : 割合 ($p(Y1) + p(Y2) = 1$)

2-5-5. ランダムフォレスト

- ランダムサンプリングされたトレーニングデータによって学習した多数の決定木を使用する方法
- 処理手順の概要
 1. ランダムにデータを抽出する
 2. 抽出したデータで決定木を生成する
 3. 1. と2. を繰り返す
 4. 予測結果の多数決で分類しきい値を決定する
- RのrandomForestパッケージなどで実装されている

2-5-6. サポートベクターマシン (SVM)

- 教師あり学習
- 訓練データから、各データ点との距離が最大になるマージン最大化超平面を求める（これだけでは線形分離可能な場合のみ）
- カーネル法を使い、特徴空間上で線形分離することで、実空間での非線形分離が可能
 - カーネル関数と呼ばれる関数で入力データをより高次元の特徴空間に写像し、特徴空間上で分割する
- Rのkernlabパッケージなどで実装されている

AI 演習資料

転移学習の演習

- 背景画像と 2 人の顔画像を使って 3 分類の画像分類を行います。
- 学習には Colaboratory を使い、予測はノート PC のローカル環境で行います。
- ローカルの実行環境については、後ほど行う学習済みモデルの利用のために、ベータ版ですが TensorFlow2.0.0-beta1 を使用します。

■学習用画像データセット作成

「画像の水増し.py」、「学習用データセット作成.py」を参照。ローカル環境で実行することを想定していますが、Colaboratory でも実行できます。

1. 各分類の画像を数枚ずつ用意する。
2. Image Augmentationにより分類ラベルごとに画像を水増しする（画像の水増し.py）。
 - デフォルトでは「FaceImage」フォルダに画像を置く。
 - 後で生成した画像の一部を学習評価用データにすることを考慮して枚数を決める。
 - 入力画像のサイズはここで決定する。
 - ✓ 100x100 を目安に。
 - 生成された画像は「AugImgs.npy」という名前での一つの npy ファイルとして出力される。
 - ✓ 分類ラベル毎に出力ファイル名を変更する。
3. 全画像データを統合して一つの npy ファイルにする（学習用データセット作成.py）。
 - 先に作成した各分類の npy ファイルを一つの npy ファイルに統合する。
 - 統合する際にラベルデータを一緒に作成する。
 - デフォルトでは各分類の 20%を学習評価用データとしている。
 - 学習用データセット
 - ✓ training_imgs.npy（入力画像データ）
 - ✓ training_labels.npy（正解ラベルデータ）

- 学習評価用データセット
 - ✓ validation_imgs.npy (入力画像データ)
 - ✓ validation_labels.npy (正解ラベルデータ)

■ シンプルな CNN による分類モデルの作成

「CSAJ2019Nov 顔認識 CNN. ipynb」を参照。まずはシンプルな CNN で分類モデルを作ってみます。Colaboratory で実行します。

1. 作成した学習用データセット、学習評価用データセットを GoogleDrive のフォルダに置く。
2. 「CSAJ2019Nov 顔認識 CNN. ipynb」の data_dir を修正し、データを置いたフォルダへのパスを通す。
3. NN の定義部分を修正してニューラルネットワークを構築し分類モデルを学習させる。
4. デフォルトではデータを置いたフォルダに「CNN. h5」というファイルが作成される。

■ 作成した分類モデルによる画像分類

「detection_roi. py」を参照。ローカル環境で実行します。

1. 作成した学習モデルのファイル（デフォルトでは CNN. h5）をソースファイルと同じフォルダにダウンロードする。
2. ソースコード内の「img_rows」と「img_cols」の値を、学習データの画像サイズと一致させる。
3. 実行するとノート PC のカメラ映像が表示される。カメラ映像の中の枠内が分類の対象となる。
4. 分類結果によって枠の色が変わります。

※背景と顔は比較的正確に分類できます。しかし、演習内で試すことができる学習時間（10 分以内程度）で 2 人のうちどちらの顔かの分類が可能になる程度の学習モデル生成は難しいかもしれません。

■ 転移学習による分類モデル作成

転移学習の概要についてはスライド資料参照。

学習用データの作成は、先の顔分類モデル演習と同じ要領です。学習済みモデルで使用する画像サイズはモデルによって決まっていますが、モデルを呼び出すときに画像サイズを指定することで任意の画像サイズを使うことができます。

「CSAJ2019Nov 転移学習.ipynb」参照

12~46 行目 学習用データ読み込みや前処理など

```
---
data_dir = "./drive/My Drive/CSAJ_AI2019Nov/Data/Transfer/"

train_images = np.load(data_dir+"training_imgs100x100.npy")
train_labels = np.load(data_dir+"training_labels100x100.npy")
valid_images = np.load(data_dir+"validation_imgs100x100.npy")
valid_labels = np.load(data_dir+"validation_labels100x100.npy")

IMG_SIZE = 100
IMG_SHAPE = (IMG_SIZE, IMG_SIZE, 3)

# 画像の解像度とクラス数
img_rows = IMG_SIZE
img_cols = IMG_SIZE
num_classes = 3
num_channels = 3

# データ数, 行数, 列数, 3 (色チャンネル数) の 4 次元のテンソルに変形
train_images = train_images.reshape(train_images.shape[0],img_rows,
img_cols, num_channels)
valid_images = valid_images.reshape(valid_images.shape[0],img_rows,
img_cols, num_channels)

# データ形式をそろえる
train_images = train_images.astype('float32')
valid_images = valid_images.astype('float32')
train_labels = train_labels.astype('int32')
valid_labels = valid_labels.astype('int32')

# 正解データは整数値になっているので, One-hot ベクトルに変換
train_labels = tf.keras.utils.to_categorical(train_labels,num_classes)
valid_labels = tf.keras.utils.to_categorical(valid_labels,num_classes)
```

```

#画像の画素値を MobileNet V2 用に正規化
train_images =
tf.keras.applications.mobilenet_v2.preprocess_input(train_images)
valid_images =
tf.keras.applications.mobilenet_v2.preprocess_input(valid_images)
---

51 行目 学習済みモデル読み込み
---
base_model =
tf.keras.applications.mobilenet_v2.MobileNetV2(input_shape=IMG_SHAPE,
                                                include_top=False,
                                                weights='imagenet')

# base_model の重みは変更しない
base_model.trainable = False
---

62~67 行目 読み込んだ学習済みモデルに分類器にあたる層を追加
---
x = base_model.output
x = tf.keras.layers.GlobalAveragePooling2D()(x)
predictions = tf.keras.layers.Dense(num_classes,
activation=tf.nn.softmax)(x)

# ネットワーク定義
model = tf.keras.Model(inputs = base_model.input, outputs = predictions)

# ニューラルネットワークの構造を表示
model.summary()
---

73~86 行目 モデル構築と学習
---
base_learning_rate = 0.0001
model.compile(optimizer=tf.keras.optimizers.RMSprop(lr=base_learning_rate
),
              loss='categorical_crossentropy',
              metrics=['accuracy'])

#一度に学習するバッチサイズ
batchsize=100

# 学習
history = model.fit(train_images, train_labels, epochs=30,
batch_size=batchsize,
                    validation_data=(valid_images,valid_labels),
verbose=1)

# TensorFlow 2 系は compile 情報も含めてファイルに保存する。
model.save(data_dir+"transfer_imagenet.h5")

```


keras で利用できる ImageNet (<http://www.image-net.org>) のデータによる学習済みモデルは次のとおりです。 (<https://keras.io/ja/applications/> より)

- Xception (François Chollet, Google, keras の作者)
- VGG16 (Visual Geometry Group, Univ. Of Oxford)
- VGG19 (Visual Geometry Group, Univ. Of Oxford)
- ResNet50 (Kaiming He, 当時 Microsoft Research Asia, 現 Facebook AI Research)
- InceptionV3 (Christian Szegedy, Google)
- InceptionResNetV2 (Christian Szegedy, Google)
- MobileNet (Andrew Howard, Google)
- DenseNet (Gao Huang, 精華大学)
- NASNet (Barret Zoph, Google Brain)
- MobileNetV2 (Mark Sandler, Google)

以上。

Computer Software Association of Japan

ローカル PC における実装環境構築手順

- グラフィックスボードのドライバを最新にする
- CUDA (TensorFlow のバージョンに合わせたもの) のインストール
- cuDNN (CUDA のバージョンに合わせる) のインストール

ここまでの部分は、GPU 版の TensorFlow (tensorflow-gpu) を使う場合です。
CUDA、cuDNN、と TensorFlow のバージョン合わせには注意してください。
今回は CPU 版を使うので必要ありません。

- Miniconda のインストール
以降は Miniconda のプロンプトから作業を行う
- Python3.7 の仮想環境構築 (tf2_py37 は仮想環境名、変更可)
> conda create -n tf2_py37 python=3.7
- 仮想環境の起動
> activate tf_py37
- pip のアップグレード
> python -m pip install --upgrade pip
- Matplotlib のインストール
> pip install matplotlib
- OpenCV のインストール
> pip install opencv-python
- Pandas のインストール
> pip install pandas
- SciPy のインストール
> pip install scipy
- TensorFlow のインストール (今回は 2.0.0-beta1 をインストールする)
> pip install tensorflow==2.0.0b1

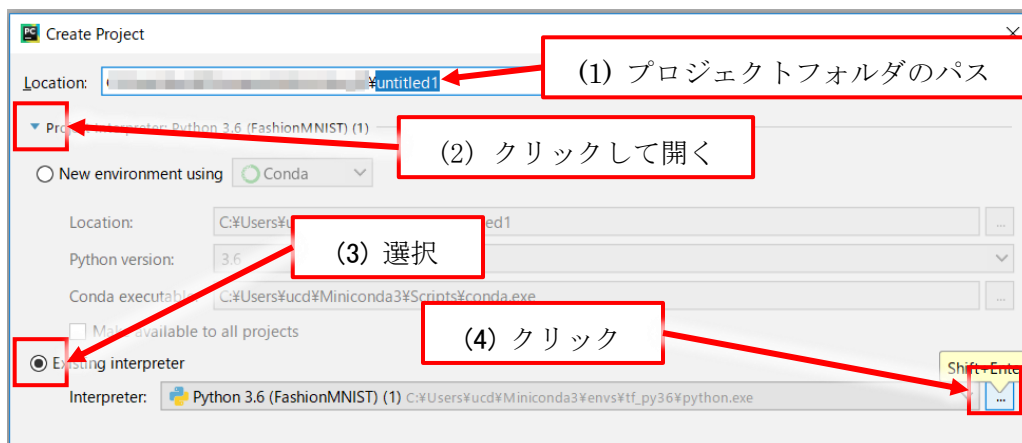
PyCharm 操作

プロジェクト新規作成

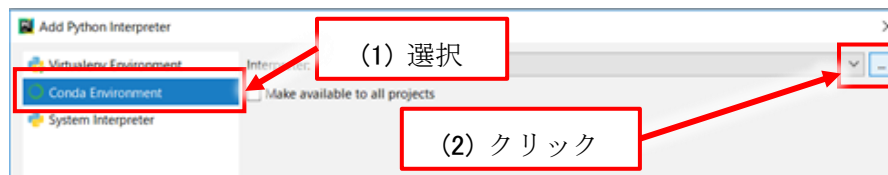
「File」から「New Project」

プロジェクトの設定 (Python インタープリタの指定)

※Miniconda で作成した仮想環境の Python インタープリタを指定します。

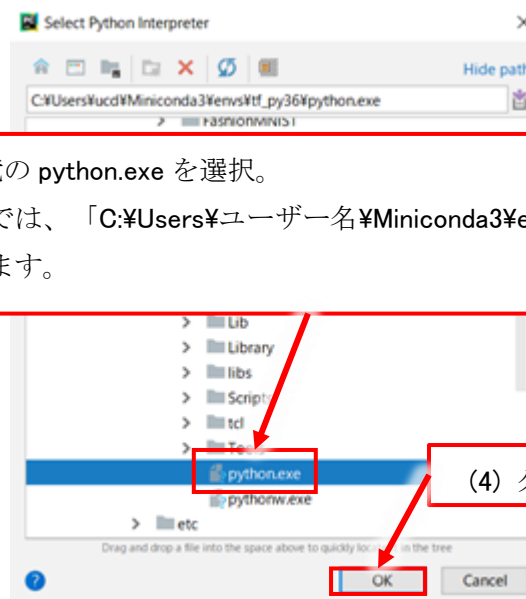


conda の仮想環境を指定



(3) 仮想環境の python.exe を選択。

デフォルトでは、「C:\Users\ユーザー名\Miniconda3\envs\仮想環境名」の中にあります。



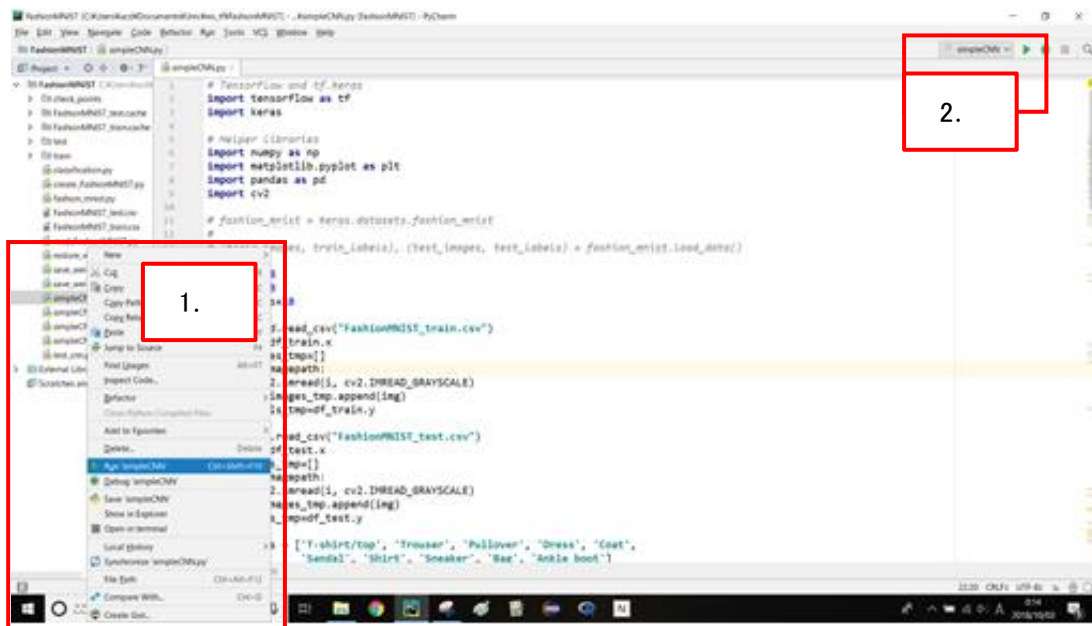


プログラムの実行方法

※ファイルは全てオートセーブなので「保存」メニューはありません。

実行方法は複数あります。

1. 実行したいソースファイル名の上で右クリックし、「Run ファイル名」をクリックする
2. 実行したいソースファイルを選択して実行ボタンをクリックする



以上。