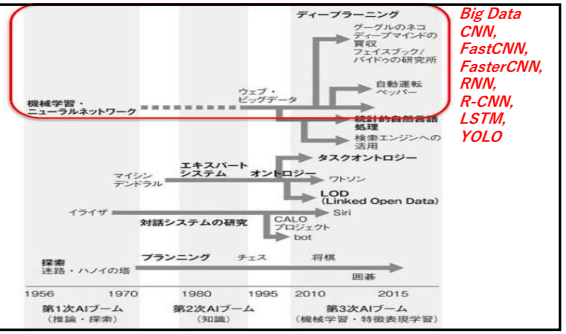
深層学習、CNN:畳み込みニューラルネットワーク、RNN、LSTM、R-CNN、FastCNN、FasterCNN、Yolo

佐賀大学　名誉教授、客員研究員
久留米工業大学・AI応用研究所　客員教授、非常勤講師
西九州大学　非常勤講師、アリゾナ大学　客員教授
新井康平

1



Big Data
CNN,
FastCNN,
FasterCNN,
RNN,
R-CNN,
LSTM,
YOLO

2

AI関連学問分野

- 高度AI、機械学習
  - 自律運転
  - オントロジー（タスクOntology）
  - リンクされたオープンデータ
  - ダイアログシステム（シリ、チャットボットなど）
- Industry4.0、 Society5.0
  - BigDataプラットフォーム
  - クラウド＆フォグコンピューティング
  - IoT、IoX（IoE、IoM, etc.）
  - ブロックチェーン
  - 深層学習
  - マシンインテリジェンス
- BigData分析(データサイエンス)
  - ストレージ＆管理
  - データ収集
  - データクリーニング
  - データマイニング
  - データ分析
  - データの視覚化
  - データ統合
  - データ言語

3

Singularity

- ナウツ（1952年）
- チェッカー（1994）
- ディープ・ブルー・チェス（1997年）
- ボナンザ将棋（2015）
- AlphaGo Go(2016)：次のステップの数は、チェスに比べて10^{100}倍
- Ray Kurtzwile：AIは2045年に人間を超える→Singularity

人工知能の変遷

- 音声認識（InterSpeech2011）
- アウディ自動自律運転（2009年）
- 行動ベースのニューラルネットワーク（2015年）
- Google Deep Dream
- 2千万の論文入力→白血病の最適治療
- MIT：単語による動画作成（2016）

4

概要

- AIは学習する→プログラミング通じて学習理論を理解
- 学習理論：教師あり学習、教師なし学習、強化学習
- クラスタリング、SVM、遺伝的アルゴリズム(GA)、EMアルゴリズム、ニューラルネットワーク、最適化理論(最急降下法、共役勾配法、シミュレーテッドアニーリング)、反復法、
- 深層学習(Deep Learning)、CNN:畳み込みニューラルネットワーク、RNN、LSTM、R-CNN、FastCNN、FasterCNN、YOLO

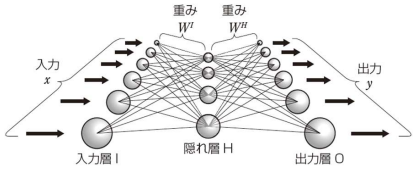
5

6

7

ニューラルネットワーク

- 人の脳で行われている情報処理を**模倣した**情報処理モデルである.
- 回帰問題にも分類問題にも用いることができる.
- 誤差逆伝搬：BP法（勾配法の効率的計算方法）によって学習する.



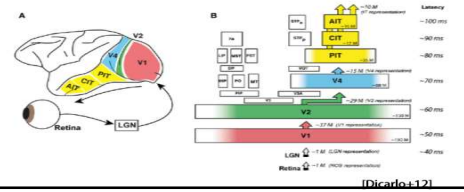
8

脳の視覚情報処理

DiCarlo, Zoccolan, Rust, How does the brain solve visual object recognition?, Neuron, 2012

視覚野（腹側皮質視覚路）の構造

- フィードフォワードで伝播
- 階層性：単純な特徴抽出 → 複雑なものへ
- IT野での処理：「おばあさん細胞」仮説と「population code」仮説

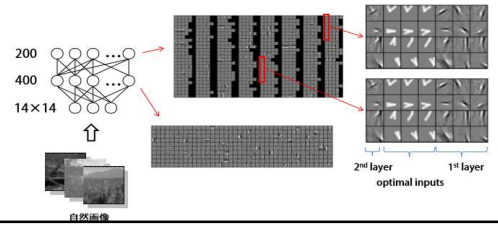


9

V2 のモデル

Lee et al., Sparse deep belief net model for visual area V2, NIPS08

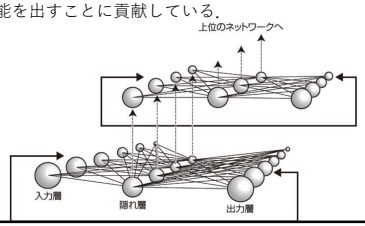
- スパースRBM を使った V2 のモデル
  - 現実のV2ニューロンの応答[Ito-Komatsu04]を再現
  - V1より複雑な形（アングルやジャンクション）に反応



10

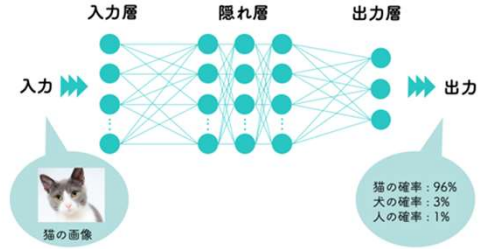
深層学習(deep learning)

- 深層学習(deep learning) は2010 年代に入ってから急速に注目されている低次元化手法であり，主にパターン認識のための特徴ベクトル抽出に用いられている．音声認識や画像認識で非常に高い性能を出すことに貢献している．



11

ディープラーニング(画像認識の例)

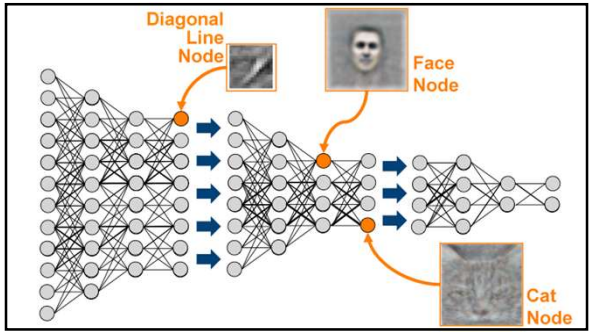


12

Deep Learning

- ニューラルネットワークの中間層を、画像のように増やして複数列を作り、多層化することで、情報伝達と処理の増加、特徴量・汎用性・予測の精度を向上させたのがディープラーニング
- 前図のように猫の画像を入力すると、これまで蓄積された多くのデータをもとに、何の写真なのかを判断する→また、その特徴はニュートラルネットワーク自身が考えるため、人間による性質の入力を必要としない
- ディープラーニングは多層化によって何層もの隠れた層を持つ→複雑な多重構造によって、より人間の脳に近いニューラルネットワークの構造を構築することが可能

13



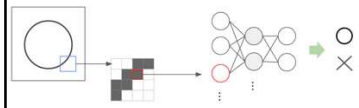
14

CNN（畳み込みニューラルネットワーク）

- 最もよく使用されているネットワーク構造です。主に画像認識や動体検知に使用されています。
- 「畳み込み層」と「プーリング層」で構成され、畳み込み層は画像の局所的な特徴を抽出して際立たせ、プーリング層は局所的な特徴をまとめてフィルタリングし、分析します。
- これによって、画像の特徴を抽象化し、パターン学習を進めていきます。
- 動画や音声の認識ができないというデメリットはありますが、画像認識と識別の速さがあり、幅広い分野で応用されています。

15

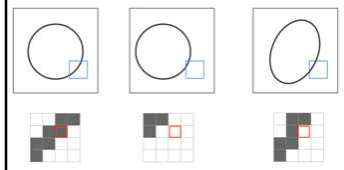
NN：図形が〇か×かを判定するタスク



- 画像の1ピクセルが1入力に対応
- 10x10の画像であれば、入力はサイズ100のベクトルになる(なお、RGB表現の場合x3)
- 円の淵の黒い部分が入力として渡っていく様子を示しているが、少し位置がずれていたりすると判定に大きな影響が出る

16

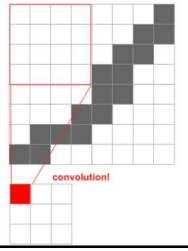
CNNのアイデア



- 図の青い四角の中は概ね「右上から左下にかけて黒」という傾向がある
- 1ピクセルではなくある程度の広さの領域をまとめて入力にすることができれば、より精度の高い判定が可能
- このアイデアを実現するのが、CNN

17

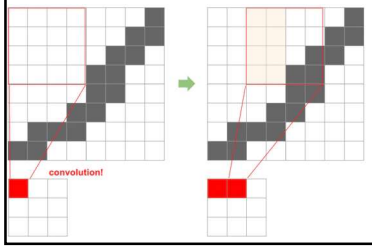
CNNの例



- 画像上にフィルタと呼ばれる小領域(左図では赤枠の4x4のエリア)をとり、これを1つの特徴量として圧縮する(=畳み込み)

18

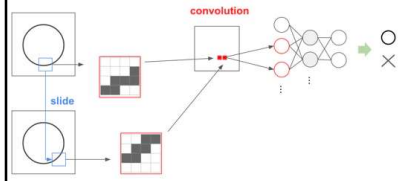
Convolution Layer



- この処理を、領域をスライドさせながら繰り返す
- この結果作成されるのが、フィルタ内の情報が畳み込まれて作成されたレイヤ、Convolution Layerになる

19

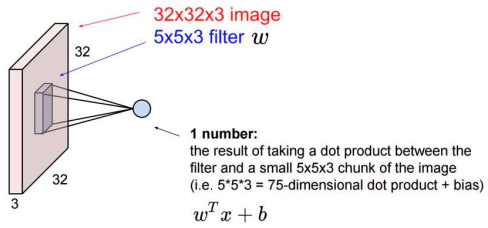
先ほどのNeural Networkの図をCNNにすると以下のようなイメージ



- このフィルタを使った「畳み込む」という処理は、具体的には「フィルタ内の画像のベクトル」と「畳み込みに使用するベクトル」との間の掛け算、内積になる

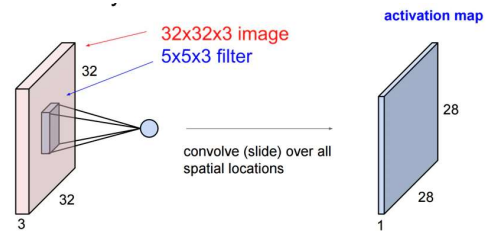
20

32x32x3の画像(32x32のRGB画像)に対して、5x5x3のフィルタを適用した例



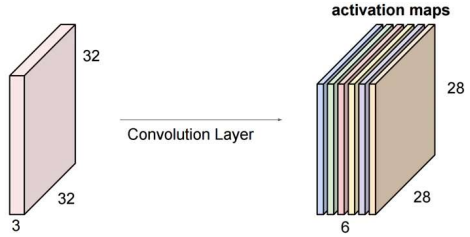
21

これにより、(スライド幅が1の場合)最終的には28x28x1のレイヤが作成



22

フィルタの種類を増やせばその分Convolution Layerも増える。以下では6つのフィルタで6つのレイヤを作成

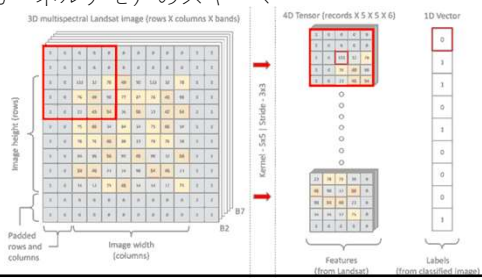


23

- これは、ちょうど畳み込みによって「新しい画像」を作っている
- こうして作った畳み込み層を通常のNeural Network同様、活性化関数でつないでいったものがConvolutional Neural Networkである(活性化関数としては、ReLUがよく使われる)
- CNNは、フィルタ内の領域の情報を畳み込んで作成するConvolution Layerを導入した、Neural Networkのこと
- Convolution Layerはフィルタを移動させながら適用することで作成し、フィルタの数だけ作成される。これを重ねて活性化関数(ReLU等)で繋いでいくことで、ネットワークを構築
- 畳み込みにより、点ではなく領域ベースでの特徴抽出が可能になり、画像の移動や変形などに頑健になる。また、エッジなど領域ベースでないとわからない特徴抽出も可能
- このCNNを特徴づけるのが、フィルタの設定とレイヤ構成

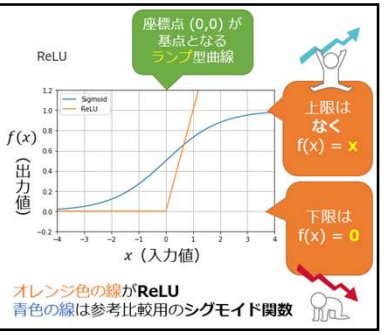
24

CNNモデルのトレーニングチップ生成（5＊5カーネルデモ）のスキーマ



25

ReLU（Rectified Linear Unit）／ランプ関数：「0」を基点として、0以下なら「0」、0より上なら「入力値と同じ値」を返す、ニューラルネットワークの活性化関数



26

ReLUの特徴

- ニューラルネットワークの基礎となっている情報処理モデル「パーセプトロン」では「ステップ関数」という活性化関数が用いられ、「バックプロパゲーション」が登場してからは「シグモイド関数」が活性化関数
- シグモイド関数の微分係数（＝導関数の出力値）の最大値が0.25（範囲は0.0~0.25）であり、そのシグモイド関数を重ねれば重ねるほど勾配の値は小さくなっていく
- 計算式がシンプルなので、処理が速い
- 0以下は常に0となるので、ニューロン群の活性化がスパース（sparse： 疎、スカスカ）になり、発火しないニューロン（＝生体ニューロンに近い動作）も表現できることで精度が向上しやすい

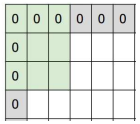
27

畳み込みに使用するフィルタについて、設定しなければならないパラメータ

- フィルタの数(K): 使用するフィルタの数。大体は2の累乗の値がとられる(32, 64, 128 ...)
- フィルタの大きさ(F): 使用するフィルタの大きさ
- フィルタの移動幅(S): フィルタを移動させる幅
- パディング(P): 画像の端の領域をどれくらい埋めるか

28

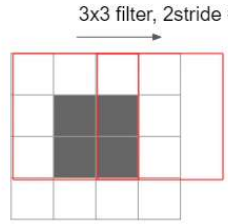
パディングは、以下のように画像の端の領域を0で埋める処理



- 普通に畳み込みを行うと端の領域はほかの領域に比べて畳み込まれる回数が少なくなる
- このように画像の端を0で埋め、そこからフィルタをかけていくことで端もほかの領域と同様に反映

29

フィルタの大きさと移動幅については、画像の大きさに適合するよう調整する



- 画像をはみ出てしまうようなフィルタの大きさ・移動幅は設定できない
- これらのパラメーターの値から、Convolutional Layerのサイズを計算することが可能

30

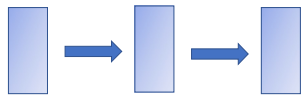
32x32x3の画像に5x5x3のフィルタを、移動幅1、パディング2で適用する例

- 32x32x3の画像に5x5x3のフィルタを、移動幅1、パディング2で適用する
- まず、パディングを加味すると画像のサイズは32+2*2=36となる
- ここから幅5のフィルタを移動幅1でとる場合、36-5+1で32となる
- つまり、最終的には32x32x3の層ができることになる
- これらのパラメーターはCaffeなどのライブラリを使用する際にも設定が必要なため、その意味とサイズの計算方法を頭に入れておくとよい

31

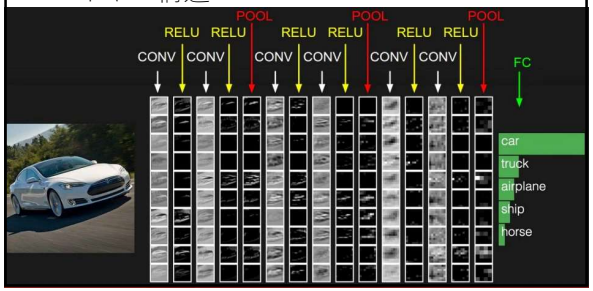
CNNにおけるレイヤの種類としては、Convolutional Layerも含めて以下の3つ

- Convolutional Layer: 特徴量の畳み込みを行う層
- Pooling Layer: レイヤの縮小を行い、扱いやすくするための層
- Fully Connected Layer: 特徴量から、最終的な判定を行う層



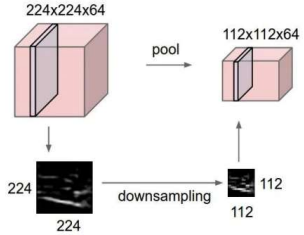
32

レイヤー構造



33

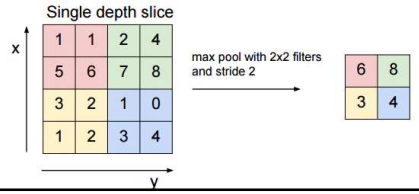
Pooling Layerは画像の圧縮を行う層(画像サイズを圧縮して、後の層で扱いやすくできるメリットがある)→ダウンサンプリング



34

Poolingを行う手段のMax Poolingは、各領域内の最大値をとって圧縮を行う方法

- Fully Connected Layerは、前レイヤのすべての要素と接続するレイヤ(主に、最後の判定などを行う層で使用される)



35

CNNの基本的な構成
(Convolution * N + (Pooling)) * M + Fully Connected * K

- **N**は~5くらいで、これを**M**層重ねて(Mは結構大きな値)、最後に判定のためのFCを**K**層(0<=K<=2)設ける(分類問題を扱うため、これにSoftMax関数を使った層をつけることもある)。
- 活性化関数としてはReLUが使用されることが多い。
- CNNはとても複雑そうに見えるが、重みをかけて伝播していくというNeural Networkの基本は外していないため、Neural Networkと同様Backpropagationによって学習させることが可能

36

CNNによる画像識別の特徴

- 画像が識別できるCNNは、画像の特徴をよくとらえられる
- 識別の層を外したCNNは、入力された画像をその特徴を(識別が可能なほど)よく表すベクトルに変換するプロセス
- 応用例の幾つかはこの特徴を利用しており、特に画像に対してキャプションを付与するといった応用例は、CNNから抽出した画像の特徴量とテキスト情報を組み合わせている

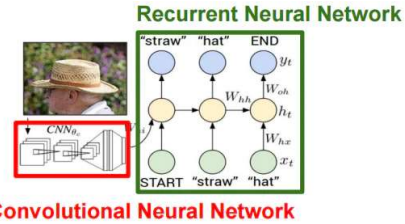
37

RNN（再帰型ニューラルネットワーク）

- 音声データのような可変長の時系列データをニューラルネットワークで扱うため、隠れ層の値を再び隠れ層に入力するというネットワーク構造を持つ
- 時系列データの学習や、自然言語処理分野（機械翻訳、文章生成、音声認識など）で使われているが、長い系列データを学習させると、勾配消失が発生し、上手く学習できない問題があるため、短時間のデータしか処理できません

38

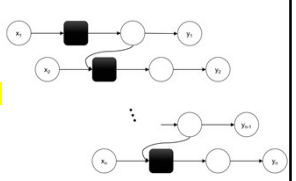
RNN：画像→CNN→RNNの隠れ層でstraw→hat→と再帰的に使用



39

Recurrent Neural Network: RNN

- 通常のニューラルネットワークでは、ある層の出力は、次の層の入力に利用されるのみである。しかしRNNでは、ある層の出力は、次の層の入力として利用されるだけでなく、一般的なニューラルネットワークの最後の層のような（中間データではないユーザーが利用可能な）出力としても利用される。また、各層の入力として、前の層の入力のみではなく、時系列のデータポイントも入力とする

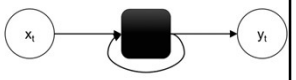


40

時系列予測

- 隠れ層同士の結合が時系列に沿って直線的であり、かつその隠れ層が同一構造のものであるような場合を「RNN」という。
- RNNでは、再帰的に出現する同一のネットワーク構造（右図中では黒四角で表現される）のことをセル（cell）と呼ぶ。

RNNの大きな特徴の一つは「ある時点の入力が、それ以降の出力に影響を及ぼす」ということである。言い換えれば、「過去の情報を基に予測できる」



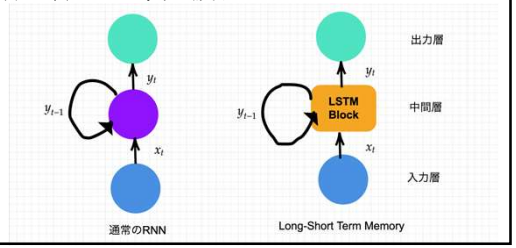
41

LSTM法（Long Short Term Memory）

- RNNのデメリットを解消した、長期の時系列データを学習することができるモデル
- 1997年に提唱された古いモデルですが、RNNの欠点を解消し、自然言語分野での処理に活用されている
- このモデルによって画像からのキャプション生成や自然なテキストの読み上げが可能になった→また動画をリアルタイムで解析して字幕を追加するなど、今までは人が行っていた分野も自動化された。

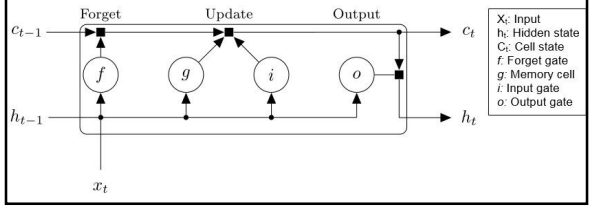
42

LSTMはRNNの一種：通常のRNNが情報をそのまま次に引き継ぐのに対し、LSTMでは中間層を噛ませて次に渡す



43

LSTMブロックは、従来のRNNにおける隠れ状態に加え、メモリセル、入力ゲート、出力ゲート、忘却ゲートを有しています。



44

- 入力ゲートへの重みとバイアスは、新しい値のセルへの流入量を制御
- 同様に、忘却ゲートと出力ゲートに対する重みとバイアスは、それぞれセル内にどの程度値が保持されるかと、そのセル内の値がどの程度LSTMブロックの出力の活性化状態を計算するために用いられるかを制御
- LSTMネットワークは、ゲートを用いて、関連する情報を選択的に保持し、関連しない情報を忘却することで、勾配消失問題（バニシング・グラジエント）の問題を解決→時間差に対する感度が低いため、LSTMネットワークは単純なRNNよりも時系列データの解析に適する

45

R-CNN: Regions with CNN

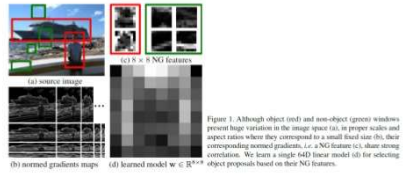
- R-CNN を使用したオブジェクトの検出用モデルは、次の 3 つのプロセスに基づく
  - オブジェクトが含まれている可能性のあるイメージ内の領域を検出→これらの領域は *"領域提案"* と呼ばれる
  - 領域提案から CNN 特徴量を抽出
  - 抽出した特徴量を使用してオブジェクトを分類
- R-CNN 検出器はまず、Edge Boxesなどのアルゴリズムを使用して領域提案を生成→提案領域はイメージからトリミングされ、サイズ変更→その後、トリミングされてサイズ変更された領域は、CNN によって分類→最後に、CNN 特徴量を使用して学習したサポート ベクター マシン (SVM) によって、領域提案境界ボックスが調整される

46

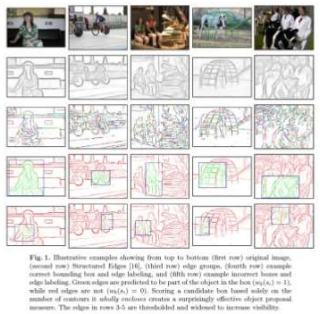
- 画像中の物体が存在しそうな場所を box として複数提案する (1,000～10,000 boxes 程度)
- できる限り少ない提案 Box 数で画像中に存在する全ての物体をカバーするように Box を提案する手法が望ましい
- 応用例として，物体検出の前処理があげられ，Sliding Window で多数の窓を調べる代わりに，Object Proposal で提案された Box だけ調べることで効率的に物体を検出できる

47

- 画像を様々なサイズ・アスペクト比に変換して勾配振幅を計算 → 8x8 画素の Box の値が対応する Window の 64 次元勾配特徴量（NG Feature, $\mathbf{g_l}$）
- フィルタースコア $s_l = \mathbf{w} \cdot \mathbf{g_l}$
- Objectness スコア $o_l = v_i \cdot s_l + t_i$（$i$ は Window のサイズ）
- Non-Maximal Suppression (NMS) で重複する Window を除去



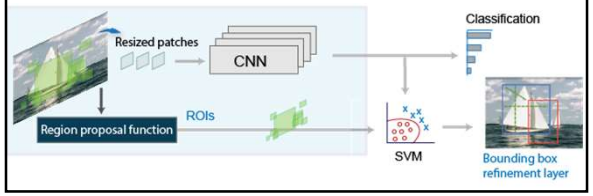
48



49

R-CNN アルゴリズムを使用したオブジェクトの検出

関数を使用して R-CNN オブジェクト検出器を学習→この関数は、イメージ内のオブジェクトを検出するオブジェクトを返す。

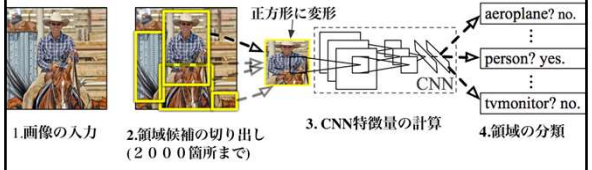


50

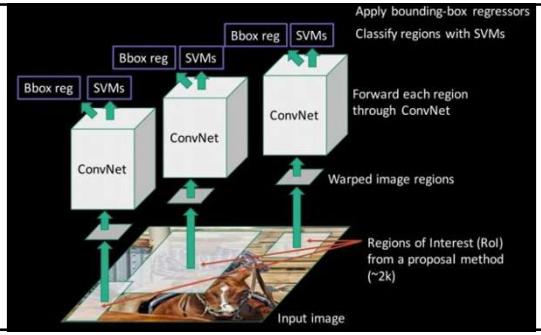
Fast R-CNN

- R-CNN 検出器と同様に、Fast R-CNN検出器も Edge Boxes などのアルゴリズムを使用して領域提案を生成→領域提案をトリミングしてサイズ変更する R-CNN 検出器とは異なり、Fast R-CNN 検出器ではイメージ全体を処理
- R-CNN 検出器は各領域を分類しなければならないが、Fast R-CNN は各領域提案に対応する CNN 特徴量をプーリングする
- Fast R-CNN 検出器ではオーバーラップする領域の計算を共有するので、Fast R-CNN は R-CNN よりも効率的

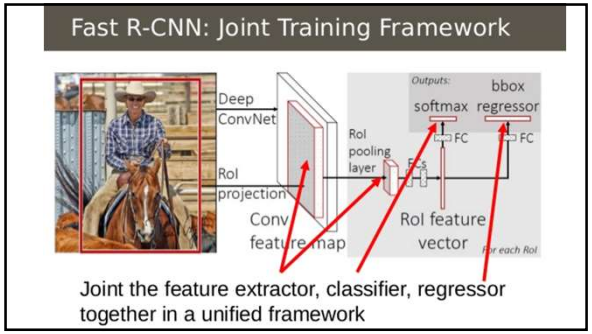
51



52

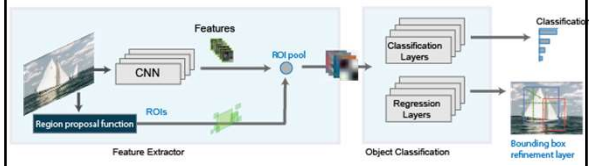


53



54

Fast R-CNN



55

Faster R-CNN

- Faster R-CNN検出器では、Edge Boxes などの外部アルゴリズムを使用する代わりに、領域提案ネットワーク (RPN) を追加してネットワーク内で領域提案を直接生成→RPN はアンカー ボックスによるオブジェクトの検出を使用→ネットワーク内で領域提案を生成する方が高速であり、データに合わせてより適切に調整される
- “アンカー ボックス”は、特定の高さと幅の事前定義された境界ボックスのセット→これらのボックスは、検出する特定のオブジェクト クラスのスケールおよび縦横比を取得するために定義され、通常、学習データセットに含まれるオブジェクト サイズに基づいて選択される→検出中、事前定義されたアンカー ボックスはイメージ全体でタイル配置される→ネットワークは、確率や、すべてのタイル配置されたアンカー ボックスの背景、Intersection over Union (IoU) およびオフセットなどのその他の属性を予測：予測は、個々のアンカー ボックスを調整するために使用される→別個のオブジェクト サイズのアンカー ボックスを複数定義できる→アンカー ボックスは、固定された初期境界ボックスの推定

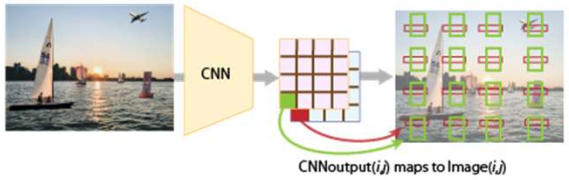
56

アンカーボックス

- アンカー ボックスを使用する場合、すべてのオブジェクトの予測を一度に評価可能
- アンカー ボックスにより、すべての考えられる位置で個別の予測を計算するスライディング ウィンドウを使用してイメージをスキャンする必要がなくなる
- スライディング ウィンドウを使用する検出器の例には、集約チャネル特徴 (ACF) または勾配ヒストグラム (HOG) 機能に基づく検出器がある
- アンカー ボックスを使用するオブジェクト検出器は、イメージ全体を一度に処理でき、リアルタイムのオブジェクト検出システムを可能にする

57

CNN→イメージマッピング



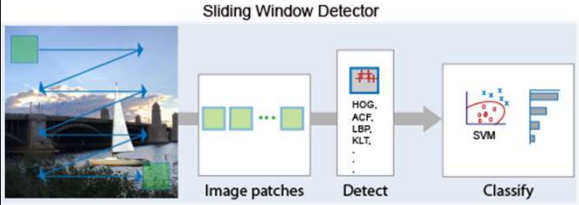
58

タイリング

- アンカー ボックスの位置は、ネットワーク出力の位置を入力イメージに戻してマッピングすることで決定される→プロセスはすべてのネットワーク出力に対して複製される→その結果として、イメージ全体でタイル配置されたアンカー ボックスのセットが生成される→各アンカー ボックスは、クラスの特定の予測を表す。たとえば、次の図のイメージ内の位置ごとに 2 つの予測を行うために、2 つのアンカー ボックスがある
- 各アンカー ボックスはイメージ全体でタイル配置されている→ネットワーク出力の数はタイル配置されたアンカー ボックスの数と等しくなる：ネットワークはすべての出力の予測を生成

59

スライディングウィンド検知



60

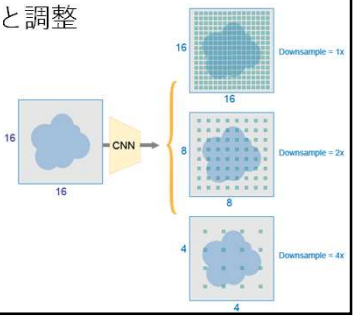
スライディング ウィンドウ

- 畳み込みニューラル ネットワーク (CNN) が畳み込みの方法で入力イメージを処理できるため、入力内の空間的な位置は、出力内の空間的な位置に関連している可能性がある→この畳み込みの対応は、CNN がイメージ全体のイメージ特徴を一度に抽出できることを意味する→次に、抽出された特徴はイメージ内の位置に戻して関連付けることができる
- アンカー ブロックを使用することで、イメージの特徴を抽出するためのスライディング ウィンドウ手法のコストが置き換えられ、大幅に削減される→アンカー ボックスを使用すると、スライディング ウィンドウ ベースのオブジェクト検出器の 3 つの段階 (検出、特徴の符号化および分類) すべてを含む、効率的な深層学習のオブジェクト検出を設計可能

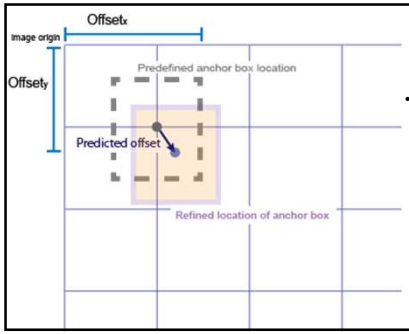
61

位置推定エラーと調整

- タイル配置されたアンカー ボックス間の距離 (“ストライド”) は、CNN 内に存在するダウンサンプリング量の関数：4 から 16 の間のダウンサンプリング係数が一般的
- これらのダウンサンプリング係数により粗くタイル配置されたアンカー ボックスが生成され、これが位置推定エラーにつながる可能性がある



62

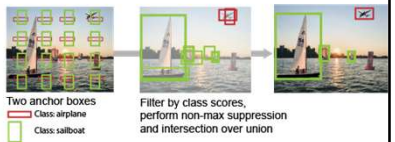


- 位置推定エラーを修正するために、深層学習オブジェクト検出器はオフセットを学習し、タイル配置された各アンカー ボックスに適用して、アンカー ボックスの位置とサイズを調整する

63

オブジェクトの検出の生成

- 最終オブジェクトの検出を生成するために、背景クラスに属するタイル配置されたアンカー ボックスが削除され、残ったアンカー ボックスは信頼度スコアによってフィルター処理される→非最大抑制 (NMS) を使用して信頼度スコアが最も高いアンカー ボックスが選択される



64

Mask R-CNN 対応 Detectron 互換 物体検出モデル

- 物体検出の手法としては Fast R-CNN, Faster R-CNN, YOLO 更には SSD などが良く知られている、最新技術としては Mask R-CNN が有名
- Mask R-CNN は物体検出した領域についてセマンティック・セグメンテーションも遂行

65

Yolo

- **YOLO (You Only Look Once)**はCVPR 2016で発表された**You Only Look Once: Unified, Real-Time Object Detection**で提案された手法。
- **YOLO**の最大の特長は、スライディングwindowや**region proposal**といった領域スキャンのアプローチを使わずに、畳み込みニューラルネットワークで画像全体から直接物体らしさと位置を算出する点
- YOLOは予め画像全体をグリッド分割しておき、各領域ごとに物体のクラスとbounding boxを求める、という方法を採用

66

You-Only-Look-Once (YOLO) v2

- You-Only-Look-Once (YOLO) v2 オブジェクト検出器は、単一ステージのオブジェクトの検出ネットワークを使用
- YOLO v2 は、畳み込みニューラル ネットワーク (Faster R-CNN) を含む領域などの、他の 2 段階深層学習オブジェクト検出器より高速
- YOLO v2 モデルは、入力イメージに対して深層学習 CNN を実行し、ネットワーク予測を生成→オブジェクト検出器は予測を復号化し、境界ボックスを生成

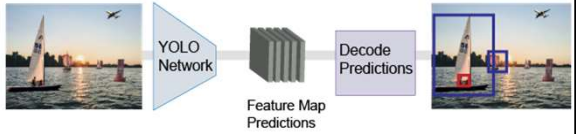
67

YOLO v2 ネットワーク

- 事前学習済みネットワークを YOLO v2 ネットワークに変換する手順は、イメージ分類の転移学習手順に似ている
- 事前学習済みのネットワークを読み込む
- 事前学習済みのネットワークから特徴抽出に使用する層を選択
- 特徴抽出層の後のすべての層を削除
- オブジェクト検出タスクをサポートする新しい層を追加

68

YoloNetwork



69

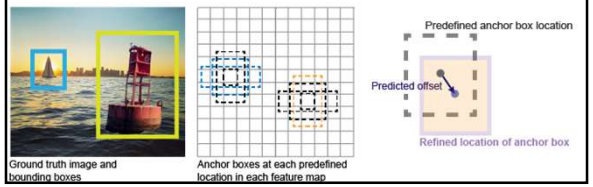
イメージ内のオブジェクトの予測

- YOLO v2 は、アンカー ボックスを使用してイメージ内のオブジェクトのクラスを検出→詳細については、アンカー ボックスによるオブジェクトの検出を参照
- YOLO v2 は、各アンカー ボックスの次の 3 つの属性を予測
- Intersection over Union (IoU) — 各アンカー ボックスの物体らしさのスコアを予測
- アンカー ボックスのオフセット — アンカー ボックスの位置を調整
- クラス確率 — 各アンカー ボックスに割り当てられるクラス ラベルを予測

70

転移学習を使用すると、YOLO v2 検出ネットワークで事前学習済みの CNN を特徴抽出器として使用できます

- 特徴マップ内の各位置の事前定義されたアンカー ボックス (破線) と、オフセットの適用後の調整された位置を示す。クラスと一致したボックスは色付きで表示される



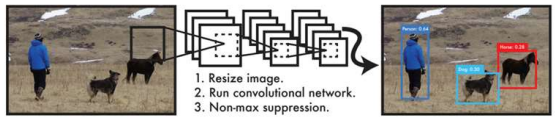
71

事前学習済みの深層ニューラル ネットワーク

- 自然イメージから強力で情報量の多い特徴を抽出するよう既に学習させてある事前学習済みのイメージ分類ネットワークを用意し、新しいタスクを学習させるための出発点として、そのネットワークを使用可能→事前学習済みのネットワークの大部分は、ImageNet データベースのサブセットで学習：このデータベースは ImageNet Large-Scale Visual Recognition Challenge (ILSVRC)で使用される
- これらのネットワークは、100 万枚を超えるイメージで学習しており、イメージを 1000 個のオブジェクト カテゴリ (キーボード、マグ カップ、鉛筆、多くの動物など) に分類可能
- 通常は、転移学習によって事前学習済みのネットワークを使用する方が、ネットワークにゼロから学習させるよりもはるかに簡単で時間がかからない

72

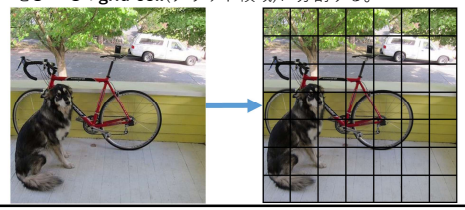
- **YOLO**では、まず入力画像を正方形(論文の例では448×448)にリサイズし、それを畳み込みニューラルネットワークの入力とする。



73

出力①

- **YOLO**は候補領域検出を行わない代わりに、正方形の画像全体を**S** × **S**の**grid cell**(グリッド領域)に分割する。

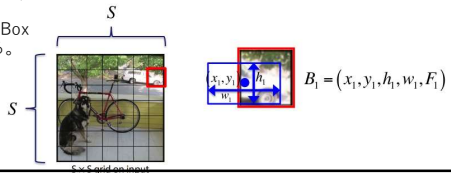


74

出力 ②

- それぞれの矩形領域は中心座標 (x,y) 、矩形の大きさ (h,w) 、信頼度 F を保持する

- 分割した各**grid cell**に対して、**B**個のBounding Boxを推定する。



75

信頼度スコア

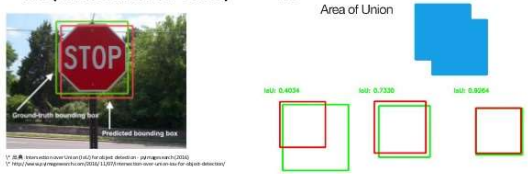
- 1つのBounding Boxにつき、Bounding Boxの座標値(x, y, w, h)と、そのBounding Boxが物体である**信頼度(confidence)スコア**の計5つの値が出力される。
- 座標値のx, yは**grid cell**の境界を基準にしたBounding Boxの中心座標、幅wと高さhは画像全体のサイズに対する相対値。**信頼度スコア**はそのBounding Boxが物体か背景かの確率を表す。(物体なら1, 背景なら0)
- 物体領域の推定精度を測る指標として、正解Bounding Boxと推定Bounding Boxの一致具合を表す**IoU (Intersection over Union)**がある。

76

領域の一致具合の指標 IoU

IoU (Intersection over Union)

$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$

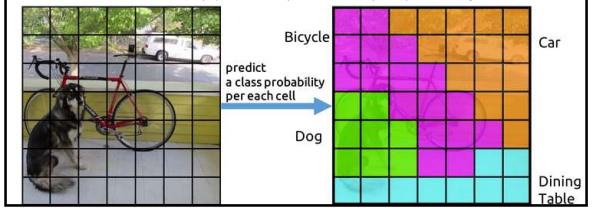


原論文では IoU threshold = {0, 0.1, …, 0.5} で認識率をスキャンし、結局 threshold = 0.3 に設定

77

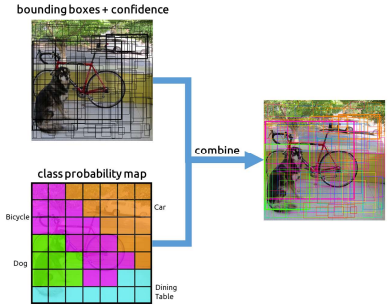
物体の種類の推定

- **YOLO**ではBounding Boxの**信頼度スコア**が**IoU**を表している。各**grid cell**単位で物体の種類も推定する。**C**種類の分類クラスで、**grid cell**が物体である場合にどのクラスに属するかの確率、つまり条件つき確率を推定する。



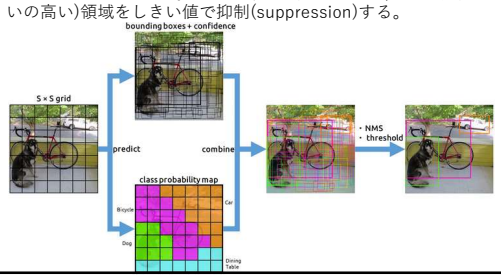
78

- Bounding Boxとclass probability mapの結合
- ここで推定したクラス確率を先ほどのBounding Boxと合わせると、何の物体であるかを示す複数のBounding Boxが得られる。



79

- 重複領域も含んだこれらのBounding Boxは、**信頼度スコア**の高いBounding Boxを基準に**NMS(Non-Maximum Suppression)**という手法で選別する。**NMS**は、**IoU**値が大きい(重なり度合いの高い)領域をしきい値で抑制(suppression)する。



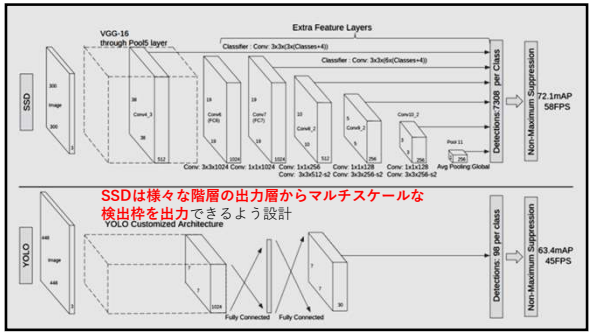
80

81

SSD: Single Shot Detector

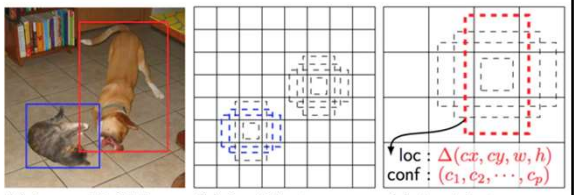
- YOLOより高速で、Faster R-CNNと同等の精度を実現
- 小さなフィルタサイズのCNNを特徴マップに適応することで、物体のカテゴリと位置を推定
- マルチスケール特徴マップ：様々なスケールの特徴を利用し、アスペクト比ごとに識別することで、高精度の検出率を達成
- 比較的低解像度でも高精度に検出できる

82



83

SSDはまた、Faster R-CNN に類似した様々なアスペクト比でアンカーボックスを使用し、ボックスを学習するのではなくオフセットを学習



84

性能比較

85

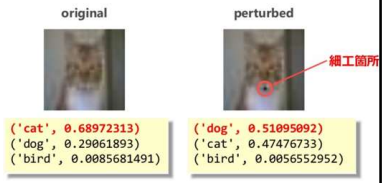
AIのリスク

- **脆弱性**
  - 標識に白と黒のステッカーを貼り付けることで「止まれ」の標識を「制限時速45マイル（72キロ）」とAIに誤認識
  - 人間には聞こえない周波数で音声アシスタントを操作できる「ドルフィンアタック」
  - アメリカでAIの研究を行なっている非営利団体の「Open AI」は、元々ある画像データにモザイクのようなノイズデータを加えることでAIの画像認識を簡単に狂わせてしまう方法
- Garbage in, Garbage out.ガラクタを入れればガラクタが出てくる(データサイエンスの重要性)
- **ブラックボックス**
- **バイアス問題**:人間の偏見が学習される
- ディープフェイク

86

CNNが1画素のノイズで誤分類

- モデルは細工なしのオリジナル画像（左）を約68%の確率で猫「cat」と正しく分類
- 顎下の1pixelを黒くした細工画像（右）は、約51%の確率で犬「dog」と誤分類しています。



87

まとめ

- ディープラーニングはニューラルネットワークによる機械学習の手法の一つであり、人間が判断基準を教えなくても機械自身が特徴を抽出することで自動学習ができる技術→複数の手法があり、目的によって使い分けが必要ですが、各分野でパフォーマンスを発揮
- 応用例は幅広く、多くのビジネスモデルが生まれており、気づかないうちに身近な場所にディープラーニングによって得られたデータが活用
- かつては最終的に人間がAIの導き出したデータを修正する必要があったが、現在では今まで人間が判断していた曖昧な基準や、人間では到底解析しきれなかった大量のビッグデータから、より正確に多角的なアプローチをすることが可能
- AIのリスクに留意し、活用(OECDの8原則)

88