ゼロから作るDeep Learning
Chap. 8: Deep Learning

Akiyama Lab. M1
Hiroki Watanabe (渡辺 紘生)

Overview

これまで

- Neural Networkを構成するさまざまな層
- Neural Networkの学習テクニック
  - パラメータの更新
  - 重みの初期化
  - Batch Normalization
  - 正則化
  - ハイパーパラメータの検証
- Convolutional Neural Network

今回

- Deep Learning（networkをより深く）
- Deep Learningの概要

Outline

1. ネットワークをより深く
2. ディープラーニングの小歴史
3. ディープラーニングの高速化
4. ディープラーニングの実用例
5. ディープラーニングの未来
6. まとめ

ネットワークをより深く

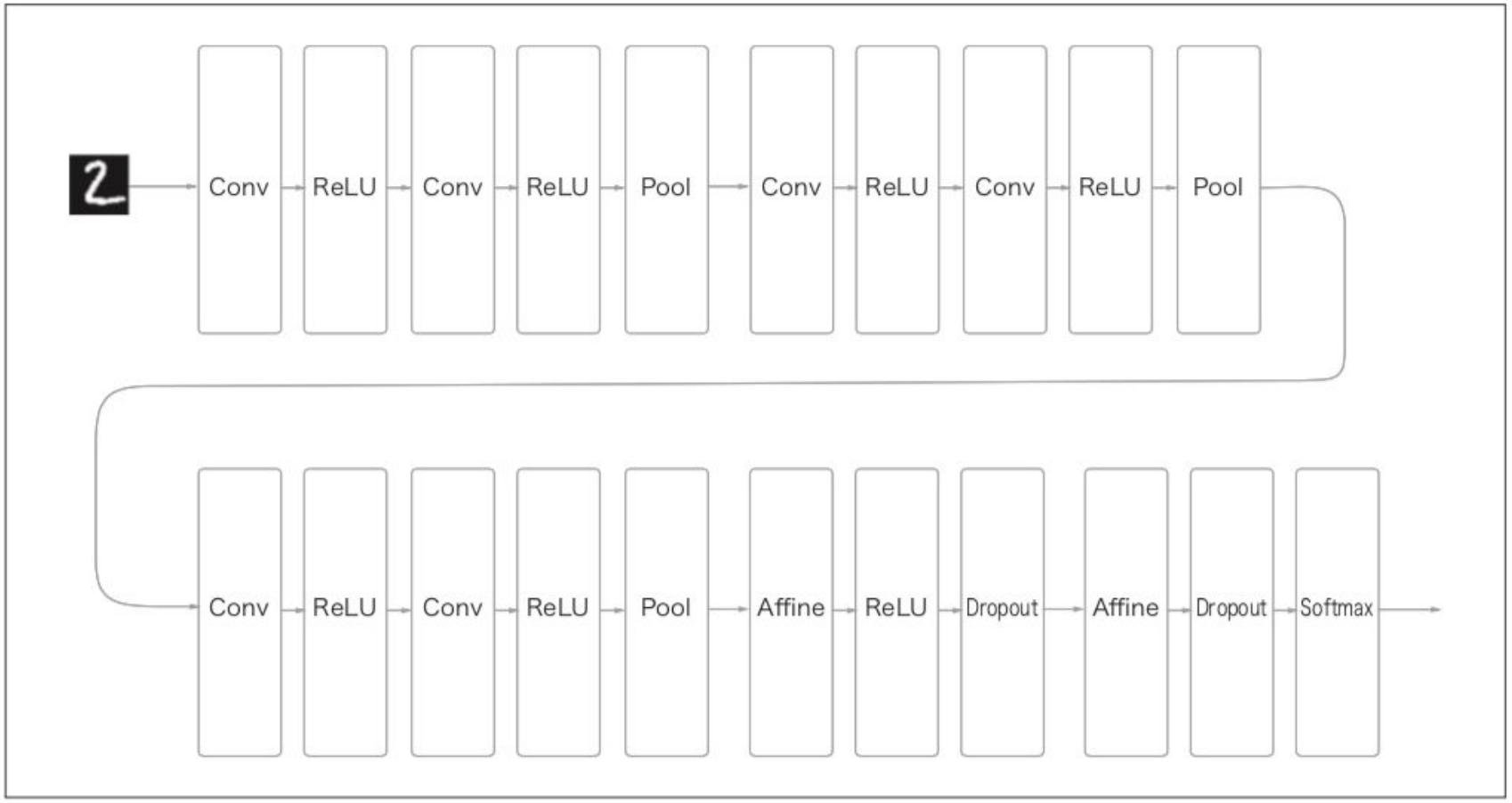
Deep CNN for MNIST

- 前章で手書き数字認識を行うCNNを実装した
- よりdeepなnetworkを構築（VGGを参考）

networkの特徴

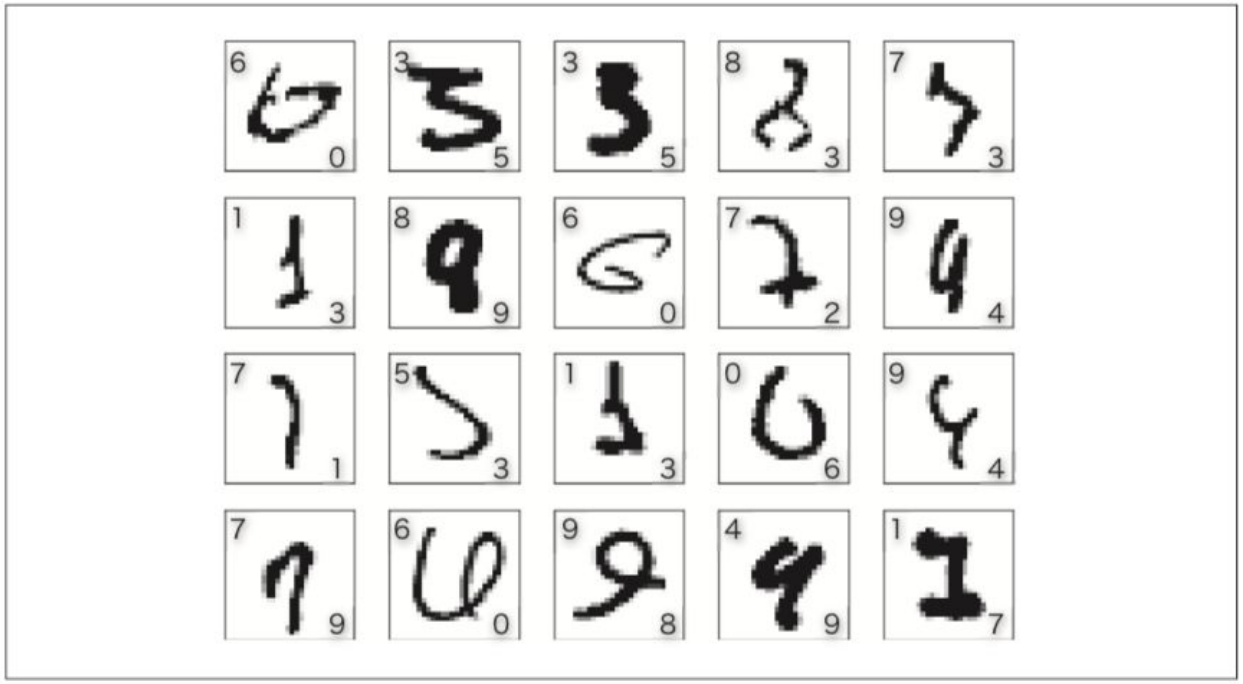
- 3 × 3の小さなfilterによる畳み込み層、層が深くなるにつれてチャンネル数が増加 (16, 16, 32, 32, 64, 64)
- 活性化関数はReLU
- 全結合層の後にDropout layerを使用
- Adamによる最適化
- 重みの初期値としての「Heの初期値」（初期値に分散2/nのガウス分布）を使用

Overview of the Layers



Result

構築したnetworkの誤認識率は0.62%



人間にとっても、わかりにくい

For Higher Recognition Accuracy...

“What is the class of this image?” (http://rodrigob.github.io/are_we_there_yet/build/classification_datasets_results.html) に、様々なデータを対象にこれまで論文などで発表されてきた手法の認識精度がランキング形式で掲載されている

- 現時点（2018年7月）では、CNN baseの手法がトップ
  - 畳み込み層2層
  - 全結合層2層

MNIST

who is the best in MNIST ?

MNIST 50 results collected

Units: error %

Classify handwriten digits. Some additional results are available on the original dataset page.

Result	Method	Venue	Details
0.21%	Regularization of Neural Networks using DropConnect	ICML 2013	
0.23%	Multi-column Deep Neural Networks for Image Classification	CVPR 2012	
0.23%	APAC: Augmented PAttern Classification with Neural Networks	arXiv 2015	
0.24%	Batch-normalized Maxout Network in Network	arXiv 2015	Details
0.29%	Generalizing Pooling Functions in Convolutional Neural Networks: Mixed, Gated, and Tree	AISTATS 2016	Details
0.31%	Recurrent Convolutional Neural Network for Object Recognition	CVPR 2015	

CNN for Simple/Complicated Image

- MNIST（単純な画像）に対しては、層を深くしなくても（現時点の）最高精度が
  得られる
  - 単純な問題では層を深くして、networkの表現力を高める必要がない
- 一般的物体認識（複雑な画像）に対しては、層を深くすると良い
  - 複雑な問題ではnetworkの表現力を高める

Data Augumentation

入力画像（訓練画像）をアルゴリズムによって人工的に拡張する

- 回転や縦横方向の移動などの微小な変化を与え、画像枚数を増やす
- データセットの枚数が限られている場合には特に有効
- crop（一部切り出し）やflip（左右反転）、輝度やスケールの変更も有効

Why Are NNs Becoming Deeper?

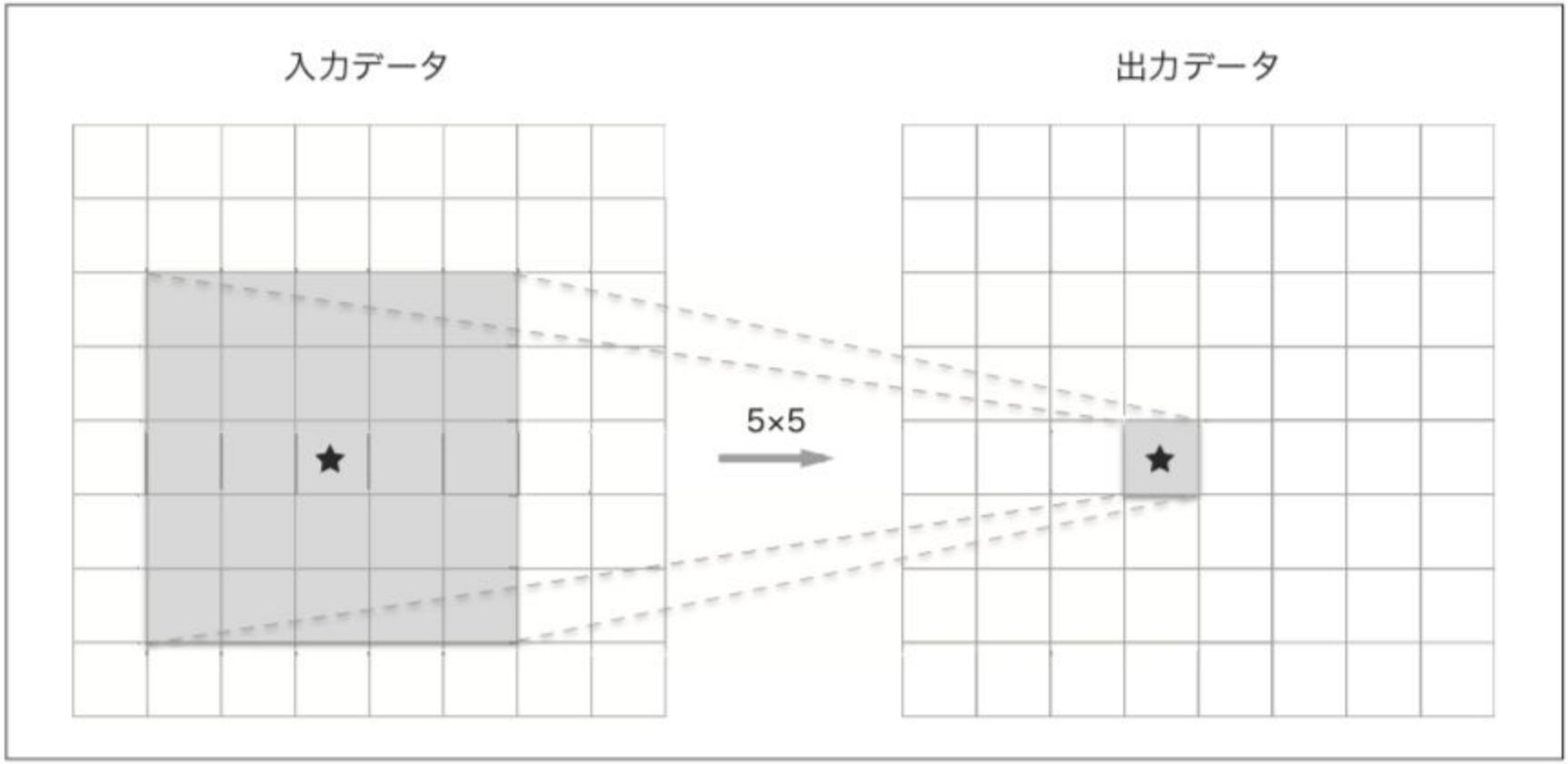
「層を深くすること」の重要性について、理論的にはそれほど多くのことが分かっていない

層を深くする利点

- networkのパラメータ数を少なくできる
  - 層を深くしたnetworkは、層を深くしなかった場合に比べて、より少ないパラメータで同レベル（もしくはそれ以上）の表現力を達成できる
- 学習の効率がよくなる

5 × 5 Filter

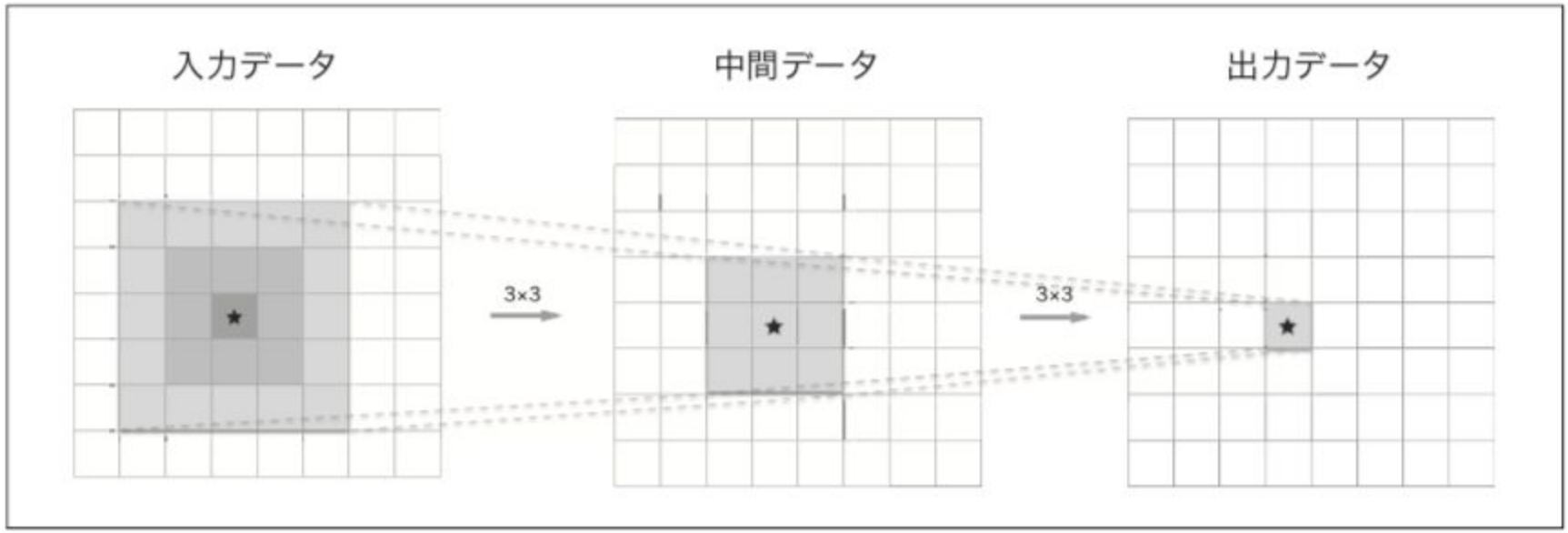
パラメータ数は5 × 5 = 25



3 × 3 Filter

パラメータ数は3 × 3 × 2 = 18

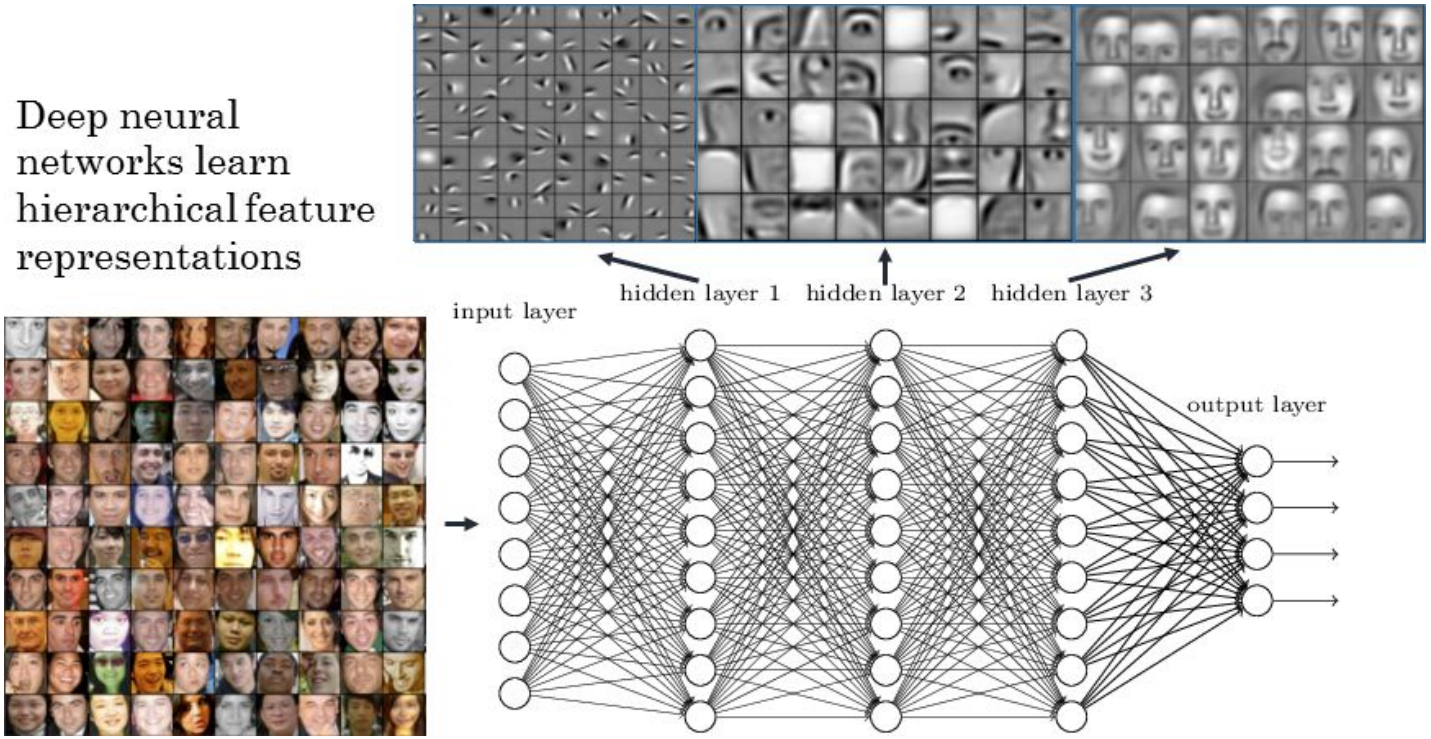
→ 5 × 5のfilterによる畳み込み層1層よりも、パラメータが少ない



Learning Efficiency

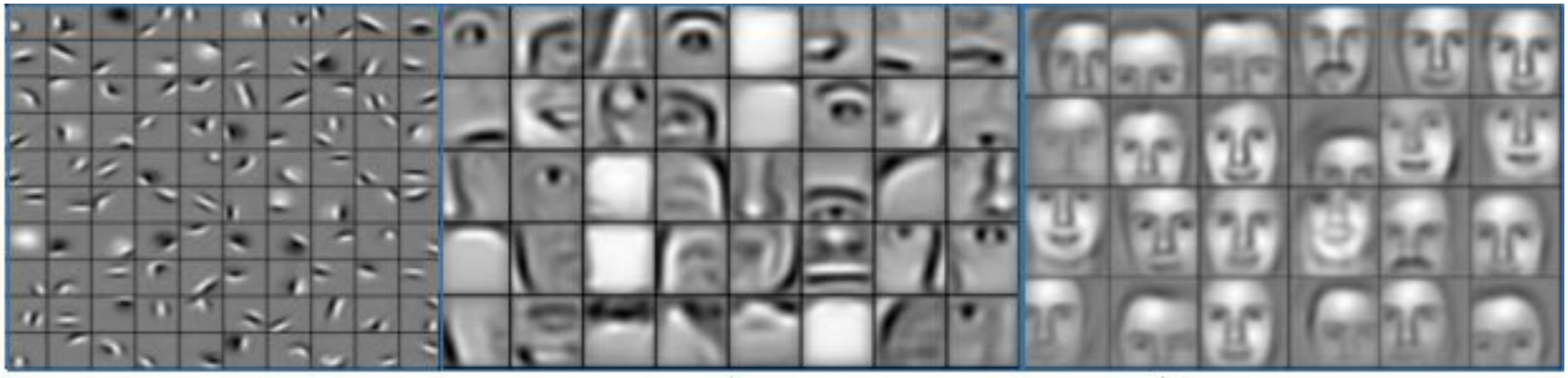
CNN の畳み込み層は階層的に情報を抽出している

- 前層の畳み込み層 → エッジなどの単純な形状に反応
- 層が深くなる → テクスチャや物体のパーツなどの複雑な形状に反応



Hierarchical Feature Extraction

- networkが浅いと、一つの畳み込み層で多くの特徴を理解する必要がある
  → 学習データのバリエーションが多い必要があり、時間がかかる
- networkが深いと、学習すべき問題を階層的に分解できる
  → 各層は「解きやすいシンプルな問題」を解けば良い
- networkが深いと、階層的に情報を渡せる
  - エッジの情報を次の層に渡せる



ディープラーニングの小歴史

ImageNet

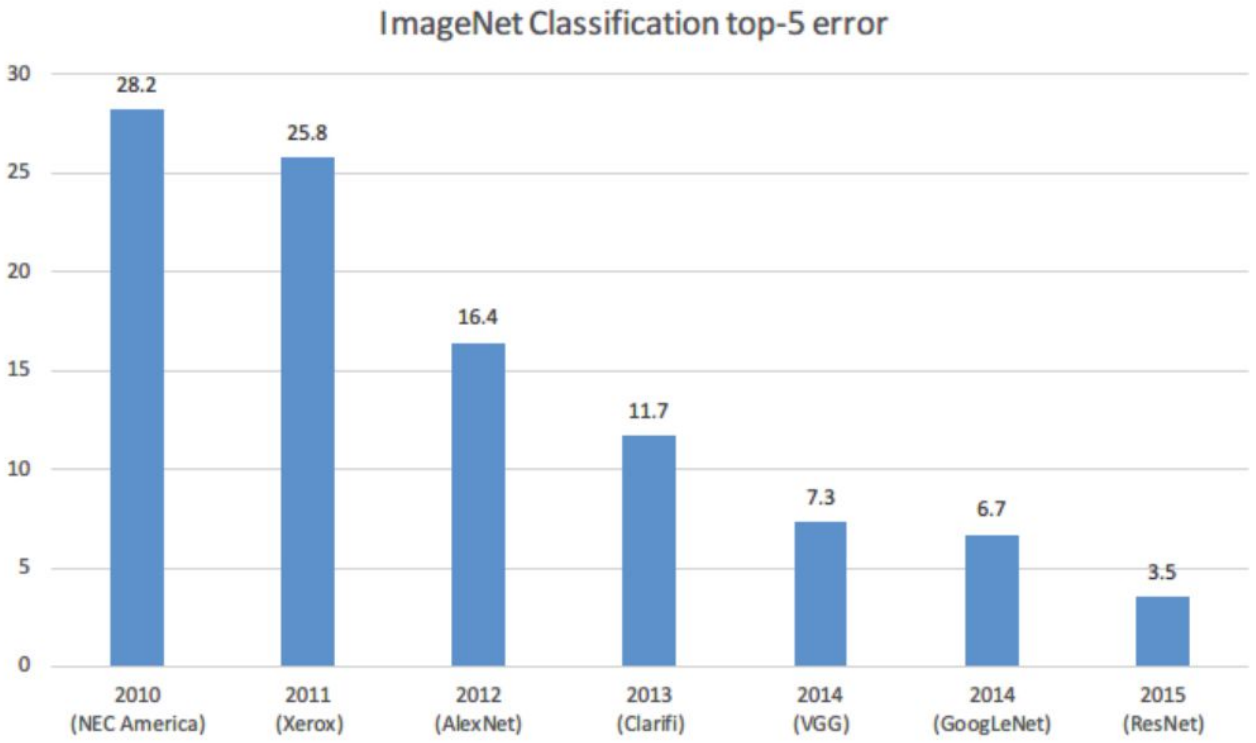
100万を超える画像のデータセット

- 様々な種類の画像
- それぞれの画像にはラベル（クラス名）が紐づけられている
- http://www.image-net.org/

ILSVRC

ImageNetを用いて行われる、画像認識のコンペティション

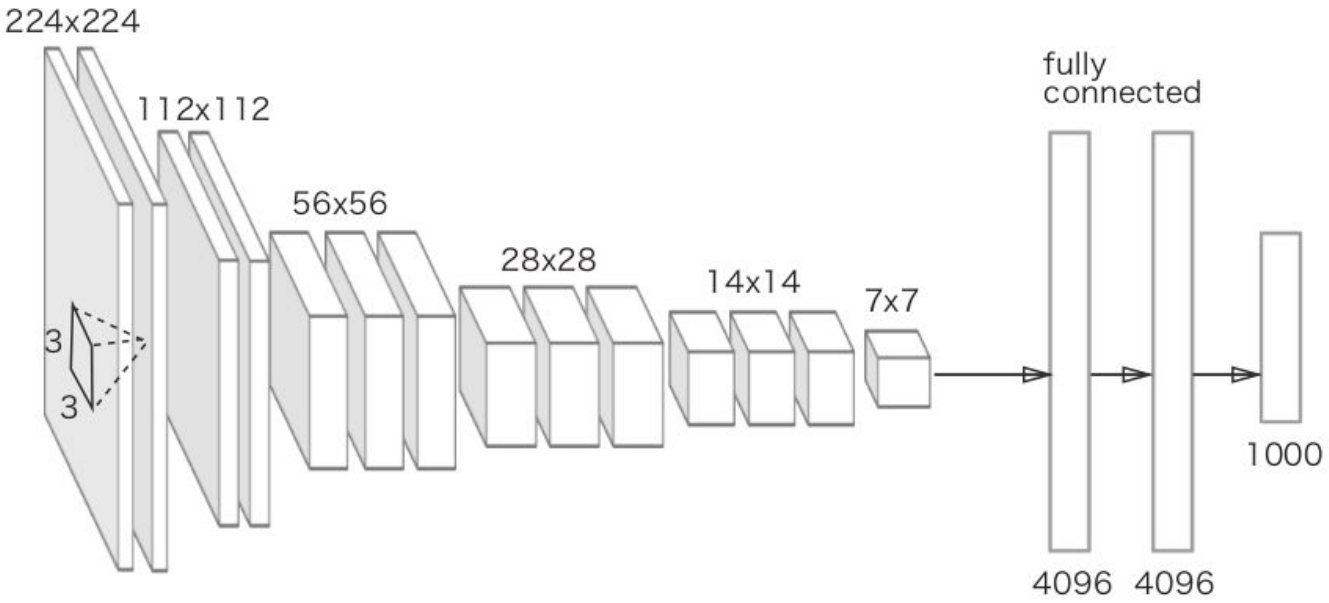
- classification部門（1,000クラスの分類の認識精度を競う）の結果（2015年までの結果）



VGG

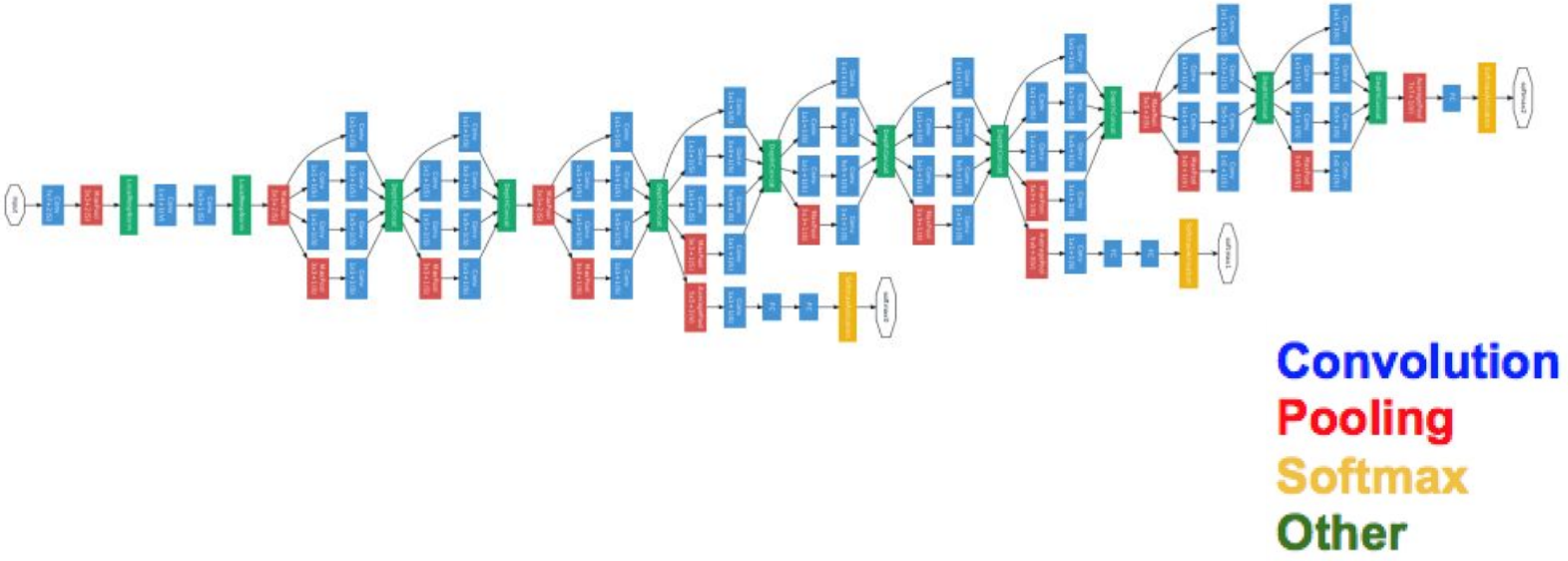
convolutional layerとpooling layerからなる基本的なCNN

- 重みのある層 (convolutional layer, fully-connected layer) を全部で16（または19）層まで重ねている
- 3 × 3の小さなfilterによる畳み込み層を連続して行っている



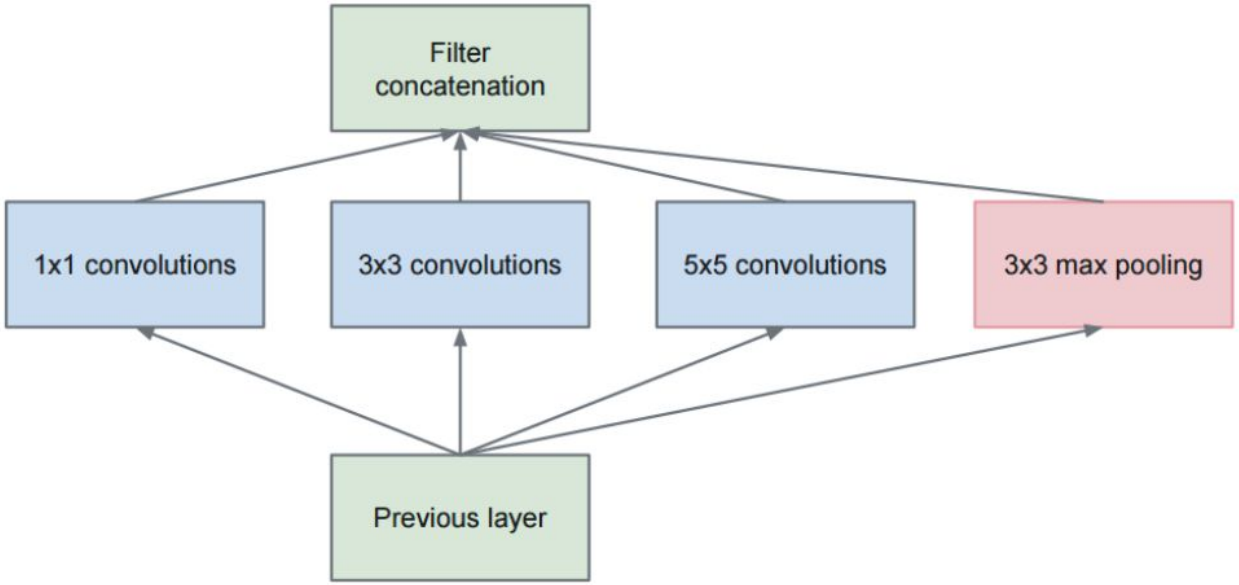
GoogLeNet

- 基本的にはこれまでのCNNと同じ構成
- networkが横方向へ広がりを持っている (inception module)
- 1 × 1のfilterのconvolutional layerを多くの場所で使用
  - チャンネル方向にサイズを減らし、パラメータの削減や処理の高速化に貢献



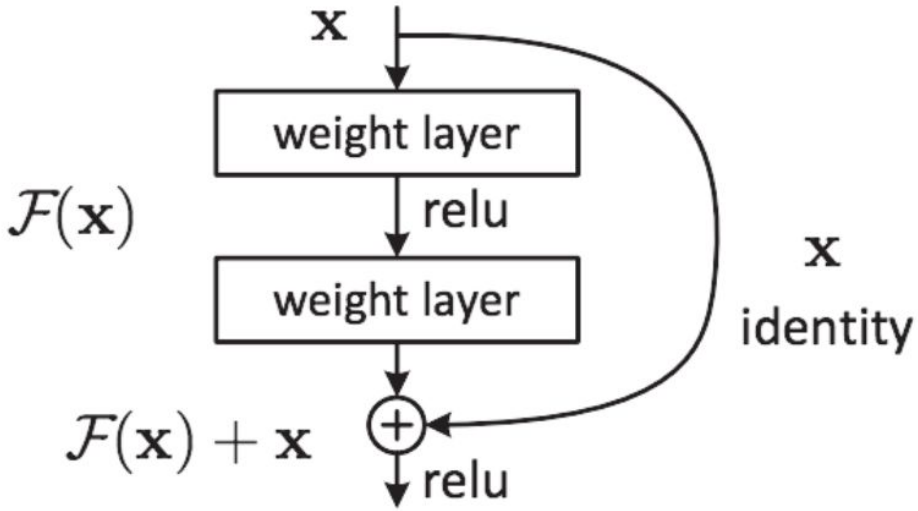
Inception Module

- サイズの異なるfilterとpooling layerを複数適用し、その結果を結合
- inception moduleを一つのbuilding blockとして使用するのがGoogLeNetの特徴



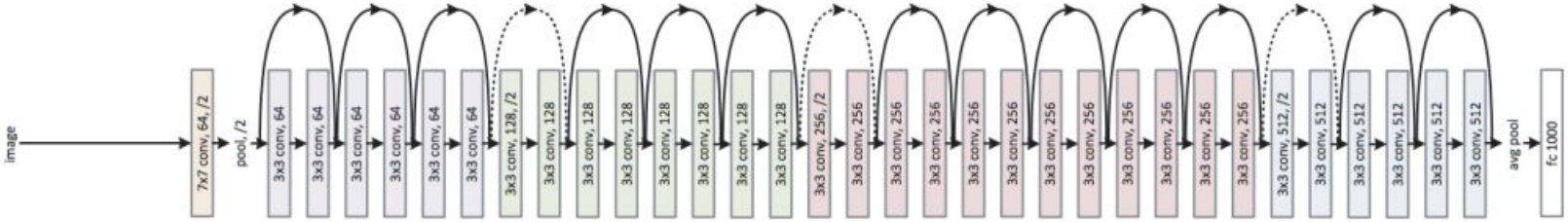
ResNet

- Microsoftのチームによって開発されたnetwork
- 層を深くする工夫がある
  - Deep Learningでは、層を深くしすぎると性能が劣ることがよくある
- スキップ構造 (shortcut, bypass)
  - 入力データの畳み込み層をスキップして出力に合算する構造
  - 逆伝搬の際に、スキップ構造によって信号が減衰せず伝わる



Layers of ResNet

- 畳み込み層を2層おきにスキップしてつなぎ、層を深くしていく
- 150層以上に深くしても認識精度は向上し続ける
- ILSVRCのコンペティションでは、誤認識率（上位5クラス以内に正解が含まれる精度の誤認識率が3.5%

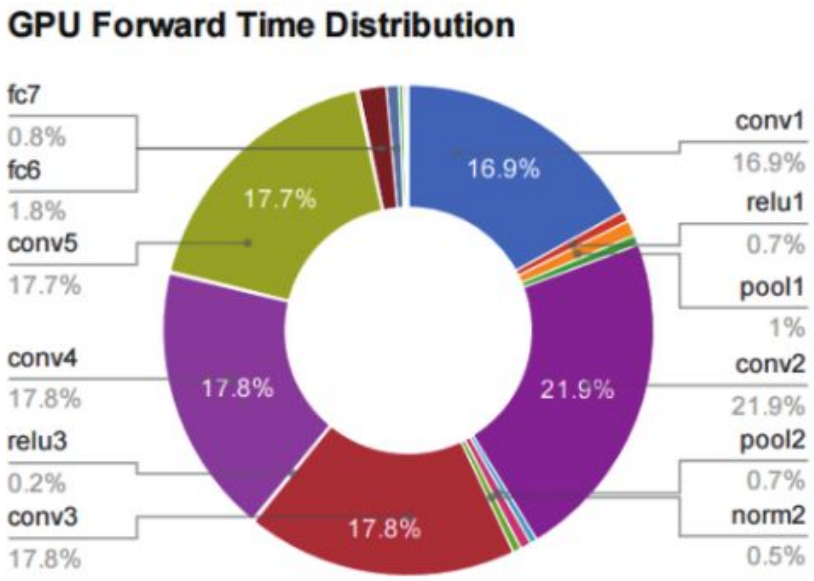


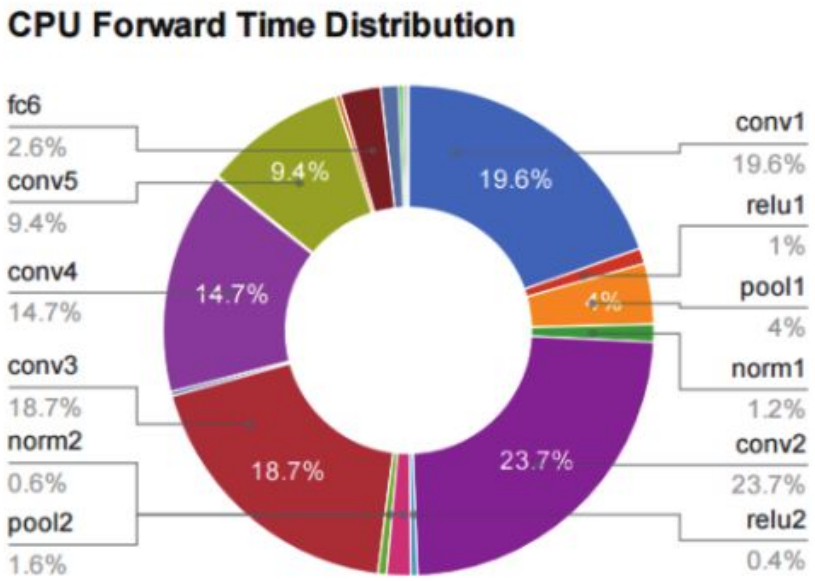
ディープラーニングの高速化

Forward Processing Time of AlexNet

AlexNet（現在のDeep Learningブームのきっかけ）のforward処理における各層の時間比率をグラフに表す

- 多くの時間が畳み込み層に費やされる (GPU: 95%, CPU: 89%)
- 畳み込み層で行われる演算をいかに高速化するか





Speed Up Using GPU

- Deep Learningでは、大量の積和演算（または大きな行列の積）を行う必要がある
- GPUを用いることで高速化可能

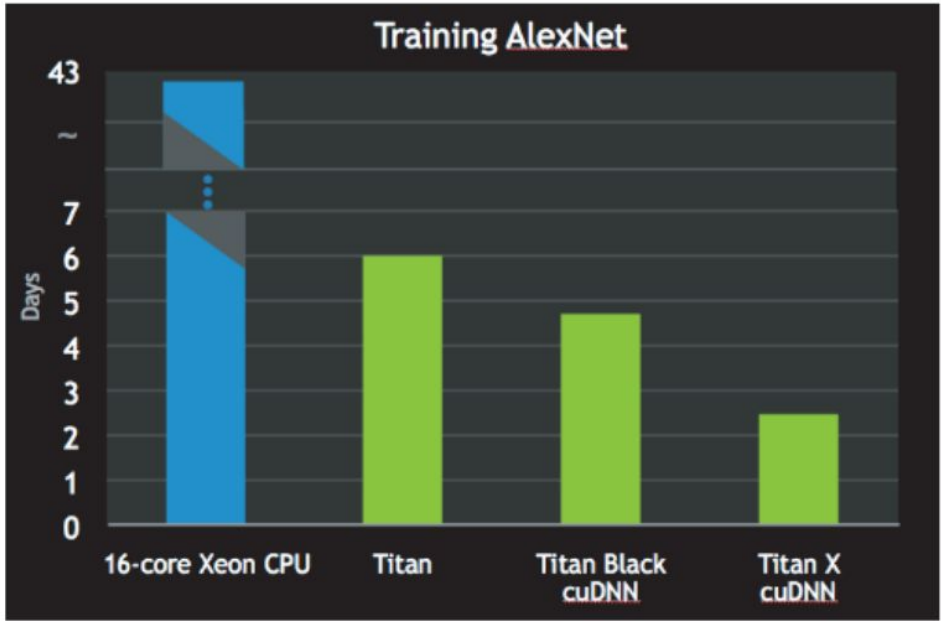


図8-15　AlexNet の学習に要する時間を、CPU の「16-core Xeon CPU」と GPU の「Titan シリーズ」で比較した結果（文献 [27] より引用）

Distributed Learning

- GPUを用いてもDeep Learningは学習のために数日・数週間必要
- Deep Learningでは多くのtry and errorを伴う

分散学習

複数のGPUや複数台のマシンで分散して計算を行う

- TensorFlow (Google), CNTK (Microsoft) は分散学習を重要視
- マシン間でのデータ通信やデータの同期などの問題がある

Distributed Learning on TensorFlow

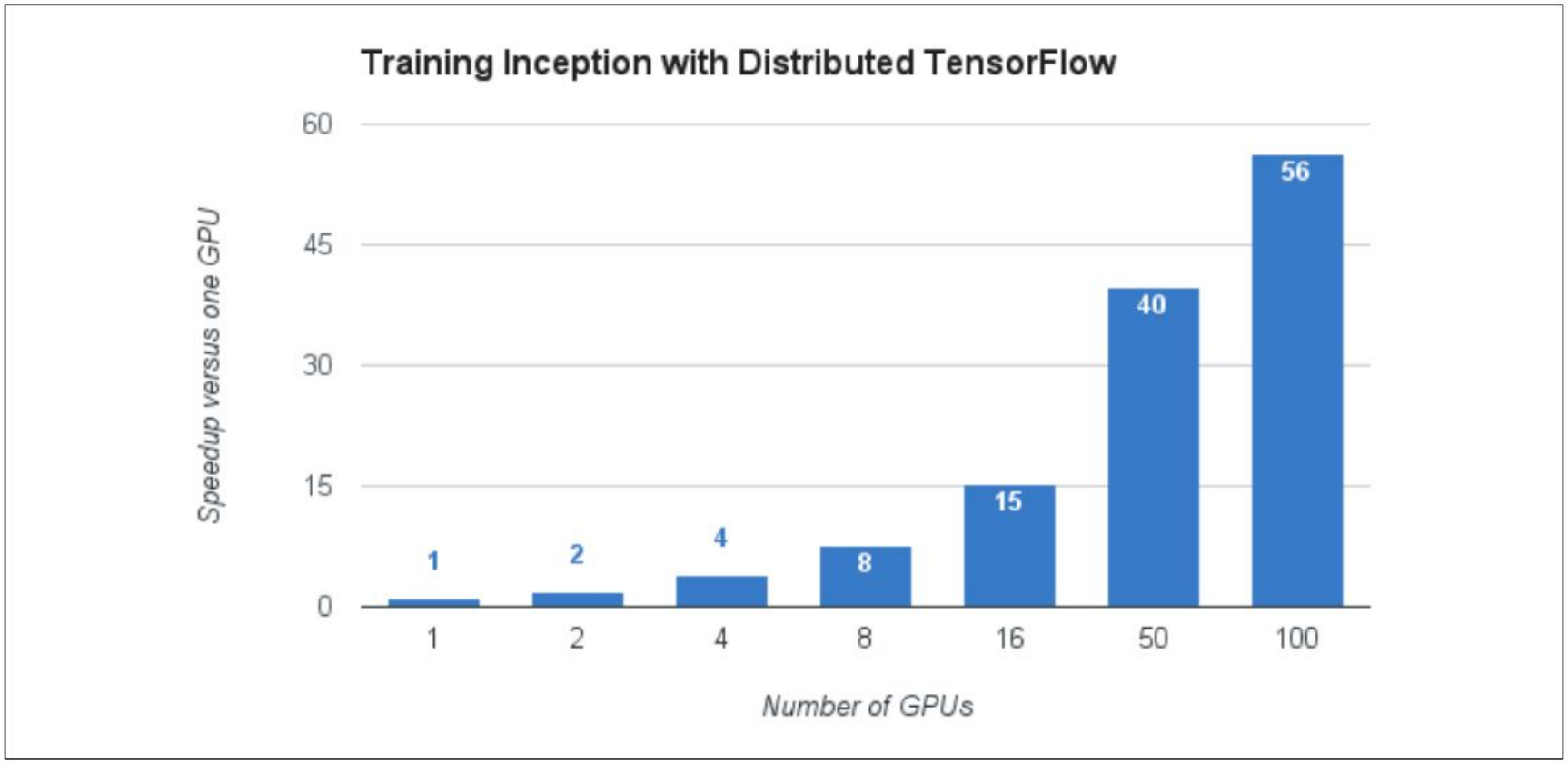


図8-16　TensorFlow による分散学習の効果：横軸は GPU の個数、縦軸は GPU ひとつのときと比べた高速化率（文献 [28] より引用）

Speed Up by Low Bit NN

- Deep Learningの高速化では、メモリ容量やバス帯域などがボトルネックになり得る
  - 大量の重みパラメータや中間データをメモリに格納
  - GPU, CPUのバスを流れるデータが増加してある制限を超える

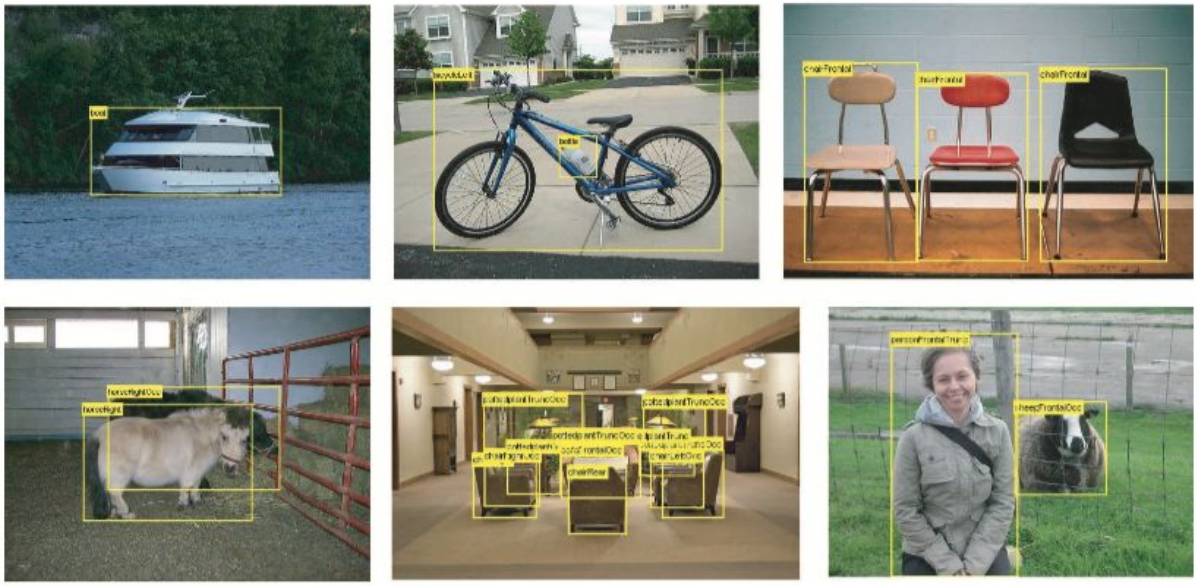
ビット数を減らす

- Neural Networkは、入力画像に小さなノイズがのってしまっても、出力結果が変わらない (robustness)
- 半精度浮動小数点数 (half float) でも問題なく学習できる
- 重みや中間データを1ビットで表現する手法も提案されている (Binarized Neural Network)

ディープラーニングの実用例

Object Detection

画像中から物体の位置の特定を含めてクラス分類を行う問題

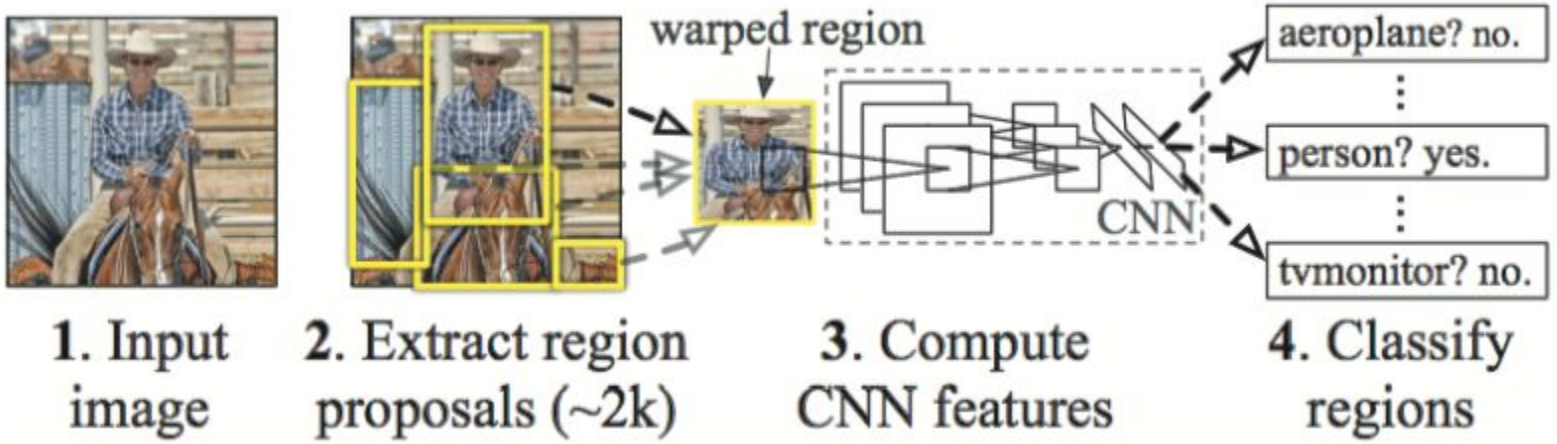


- 物体認識より難しい
  - 物体認識 → 画像全体を対象、単一の物体
  - 物体検出 → クラスの位置まで特定、複数個の物体

R-CNN

CNNを用いて物体検出を行う手法の一つ

- 2. でオブジェクトらしい領域を探し出す
- 3. でCNNを行う
- 実際のフローは、画像を正方形に変形したり、分類にSVMを用いたりと複雑



Segmentation

画像に対してピクセルレベルでクラス分類を行う問題

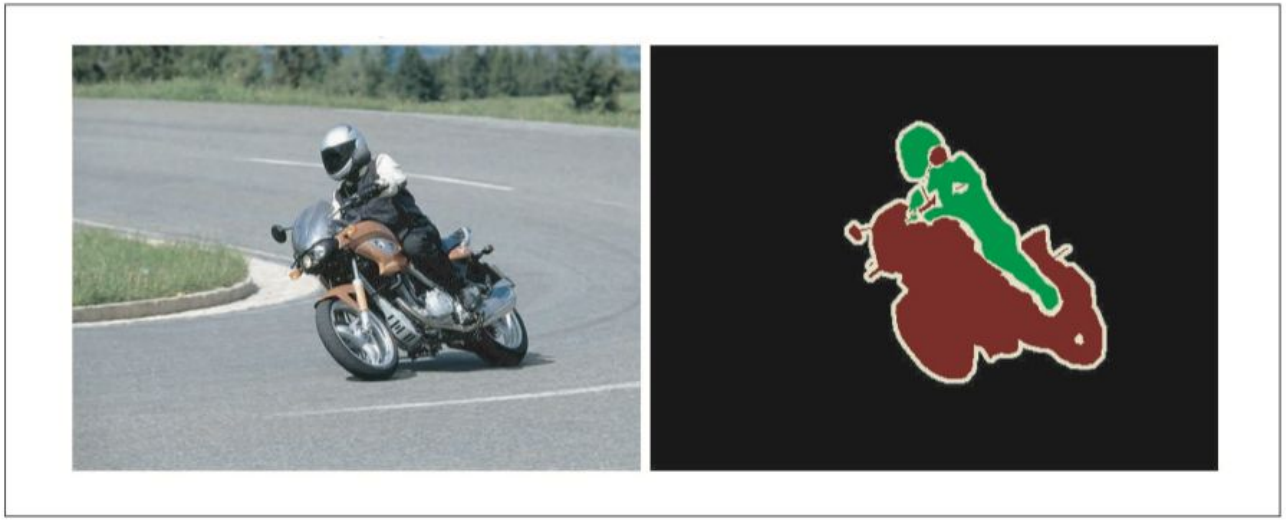


図8-19　セグメンテーションの例（文献 [34] より引用）：左が入力画像、右が教師用のラベリング画像

FCN (Fully Convolutional Network)

- 1回のforward処理で全てのピクセルに対してクラス分類を行い、多くの領域を再計算する無駄を改善する

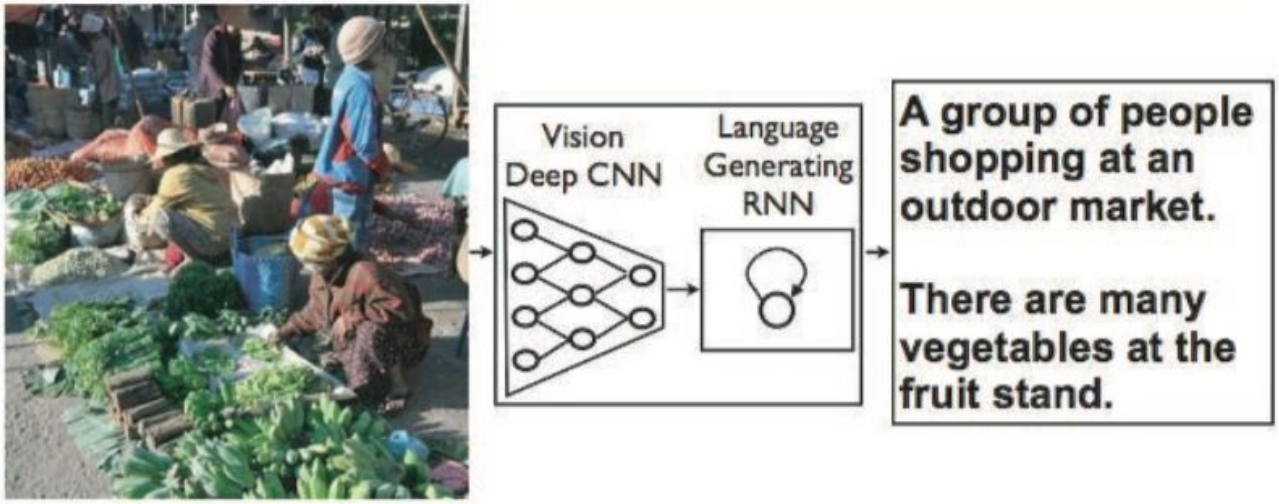
Image Caption Generator

画像を与えると、その画像を説明する文章を自動で生成する

NIC (Neural Image Caption)

画像キャプションを生成する代表的な方法

- CNNとRNN (Recurrent Neural Network) から構成される
  - RNNは再帰的なつながりをもつnetwork
  - RNNは自然言語や時系列データなどの連続性のあるデータに対してよく用いられる



ディープラーニングの未来

Image Style Transfer

Deep Learningを使って、アーティストのような絵を”描かせる”

- 2つの画像を入力し、新しい画像を生成する

Image Generator

新しい画像を「ゼロから」描く

- 先に大量の画像を使って学習する
- 下の図はbedroomの画像

DCGAN

先のbedroomの画像はDCGANによって生成された

- 以下の2つのNNを利用している
  - Generator : 本物そっくりの画像を生成
  - Discriminator : Generatorが生成した画像が本物かどうか判定
- 両者が互いに切磋琢磨しながら成長していくのがGAN (Generative Adversarial Network) の特徴

Automated Driving

- 通行ルートを決めるpath plan、カメラやレーザーなどのセンシング技術など、様々な技術が必要
- SegNet（CNNベースのnetwork）は高精度に走路環境を認識できる（下図）

図8-25　ディープラーニングによる画像のセグメンテーションの例：道路や車、建物や歩道などが高精度に認識されている（文献 [43] より引用）

Reinforcement Learning

コンピュータにtry and errorの過程から自律的に学習させる (reinforcement learning)

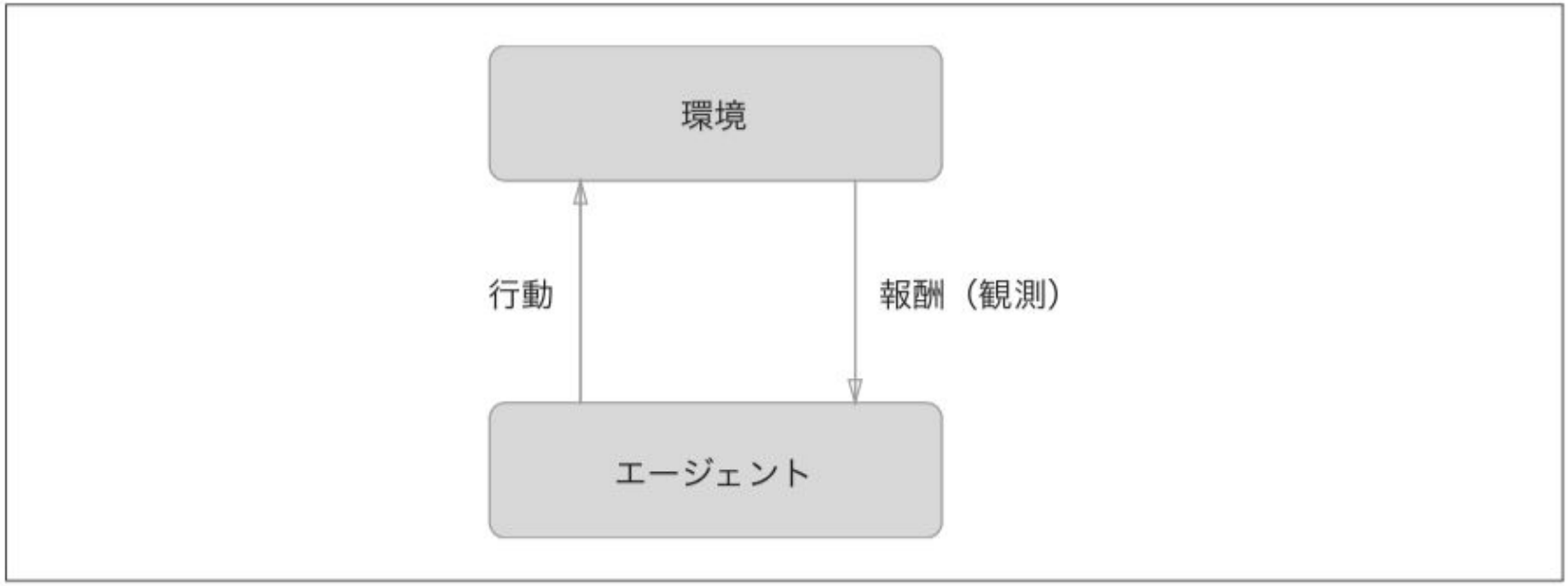


図 8-26　強化学習の基本的な枠組み：エージェントは、より良い報酬を目指して自立的に学習する

Deep Q-Network (DQN)

強化学習において、報酬は「見込み」である

- 「スーパーマリオブラザーズ」では、マリオ右に動かすことによってどれくらいの報酬を得るかということは必ずしも明確ではない
  → ゲームのスコアやゲームオーバーなどの明確な指標から逆算して、「見込みの報酬」を決める必要がある

Deep Q-Network

- Q学習と呼ばれる強化学習のアルゴリズムをベースとする
- Q学習における最適行動価値関数と呼ばれる関数を近似するためにCNNを用いる

Overview of DQN

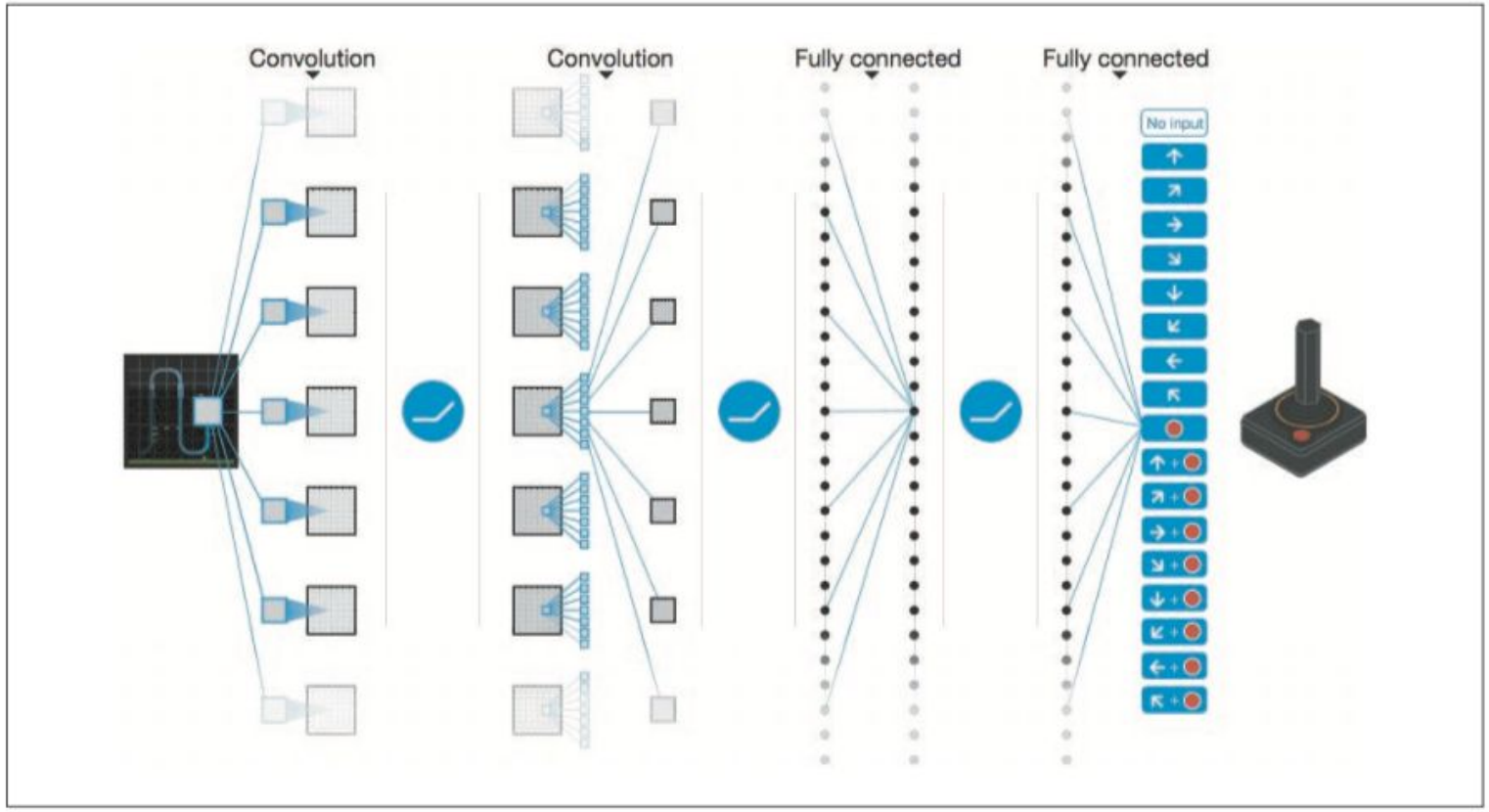


図 8-27　Deep Q-Network によってテレビゲームの操作を学習する。入力はテレビゲームの画像であり、試行錯誤を経て、プロ顔負けのゲームコントローラー（ジョイスティック）の手さばきを学習する（文献 [44] より引用）

まとめ

Conclusion

- ややディープなCNNを実装
- networkを深くする利点は、主に2つある
  - networkのパラメータ数を減らせる
  - 学習の効率をあげることができる
- これまでILSVRCのコンペティションで優れた成績をあげた手法にVGG, GoogLeNet, ResNetがある
- Deep LearningとGPUは相性が良く、GPUによりDeep Learningの学習速度を向上できる
- Deep Learningの適用例は数多く存在し、また新たな応用例も研究されている