# 畳み込みニューラルネットワーク CNN

- 畳込みニューラルネットワークによる日本語古典籍くずし字 kminst の認識実習
- 姿勢推定デモ
- Style-based GAN

CNN の特徴として，次の 7 つを上げることができます 。

1. 非線形活性化関数 (nonlinear activation functions)
2. 畳込み演算 (convolutional operation)
3. プーリング処理 (pooling)
4. データ拡張 (data augmentation)
5. バッチ正規化 (batch normalization)
6. ショートカット(shortcut)
7. GPU の使用

上記 7 つの特徴を説明するのは専門的になりすぎるので省略します。一つだけ説明するとすれば最後の GPU とは高解像度でしかも処理速度を必要とするパソコンゲームで用いられるグラフィックボードのことです。 詳細な画像を高速に画面に表示する必要から開発されたグラフィックボードですが，大規模なニュールネットワークの計算でも用いられる数学が同じです。 そのため，ゲーム用に開発されたグラフィックボードがニューラルネットワークにも用いられるようになりました。

## 畳込みニューラルネット(CNN)とは何か

本節では深層学習，特に CNN と呼ばれるニューラルネットワークについて解説します。

最初に画像処理の概略を述べる CNN が，それまで主流であった従来の手法の性能を 凌駕したことはすでに述べました。CNN の特徴の一つに **エンドツーエンド** と 呼ばれる考え方があります。エンドツーエンドとは，従来手法によるパターン認識 システムでは，専門家による手の込んだ詳細な作り込みを必要としていたことと異 なり，面倒な作り込みをせずとも性能が向上したことを指します。

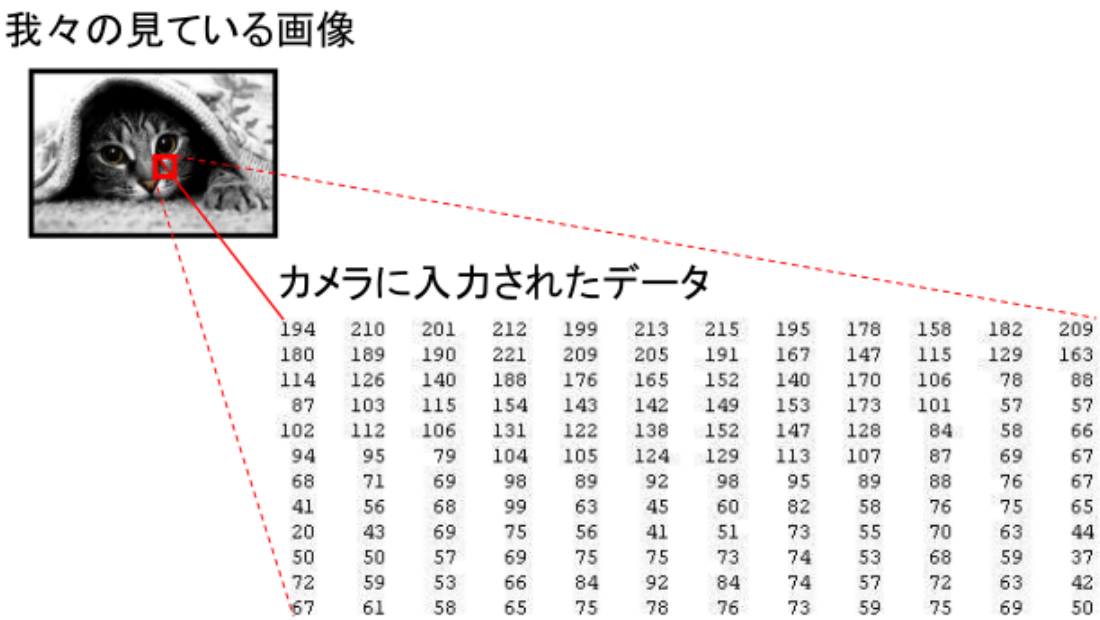
エンドツーエンドなニューラルネットワークにより，次のことが実現しました。

- ニューラルネットワークの層ごとに，特徴抽出が行われ，抽出された特徴がより高次の層へと伝達される
- ニューラルネットワークの各層では，比較的単純な特徴から次第に複雑な特徴へと段階的に変化する
- 高次層にみられる特徴は低次層の特徴より大域的，普遍的である
- 高次層のニューロンは，低次層で抽出された特徴を共有している

このことを簡単に説明してみます。

我々人間は，外界を認識するために必要な計算を，生物種としての発生の過程と， 個人の発達を通しての経験に基づく認識システムを保持していると見ることができ ます。従って我々の視覚認識には化石時代に始まる光の受容器としての眼の進化の 歴史と発達を通じた個人の視覚経験が反映された結果でもあります。人工知能の目 標は， この複雑な特徴検出過程をどうやったらコンピュータが獲得できるかという ことでもあります。外界を認識するために今日まで考案されてきたモデル（例えば， ニューラルネットワークサポートベクターマシンなどは）は複雑です。ですがモデ ルを訓練するための学習方法はそれほど難しくありません。この意味で画像認識課 題が正しく動作するためのポイントは，認識システムが問題を解く事が可能なほど 複雑であるかどうかではなく，十分に複雑が視覚環境，すなわち画像認識の場合， 外部の艦橋を反映するために十分な量の像データを容易すことができるか否かにあ ります。今日の CNN による画像認識性能の向上は，簡単な計算方法を用いて複雑な 外部環境に適応できる認識システムを構築する方法が確立したからであると言うこ とが可能です。

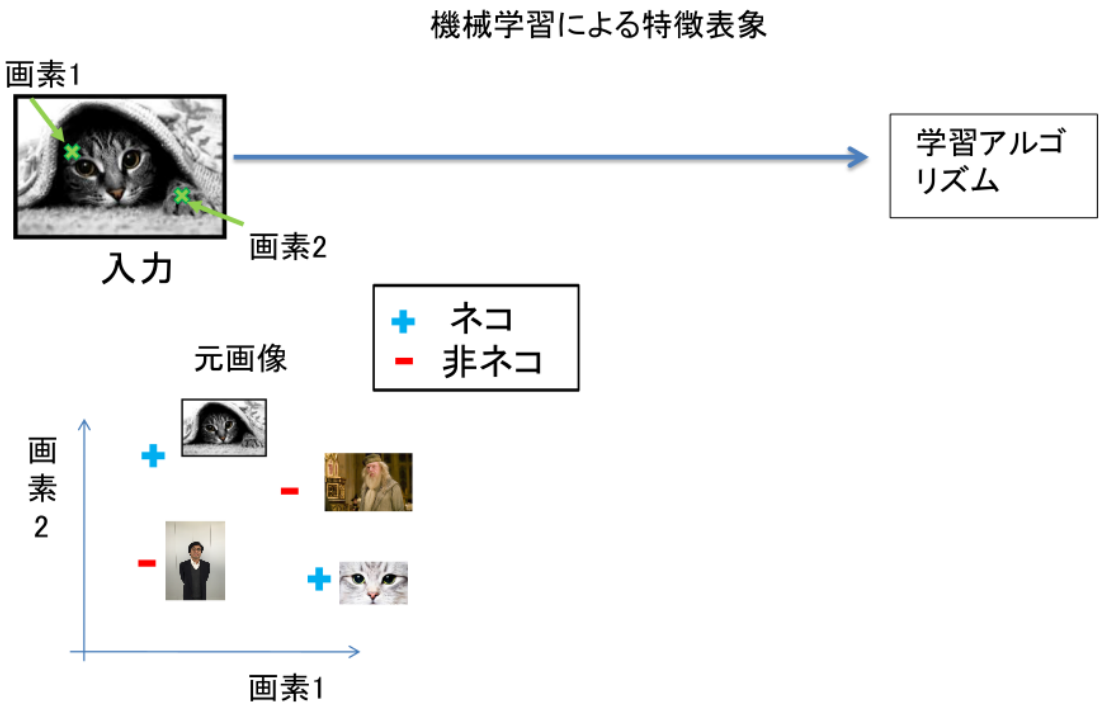
下図 に画像処理の例を挙げました。



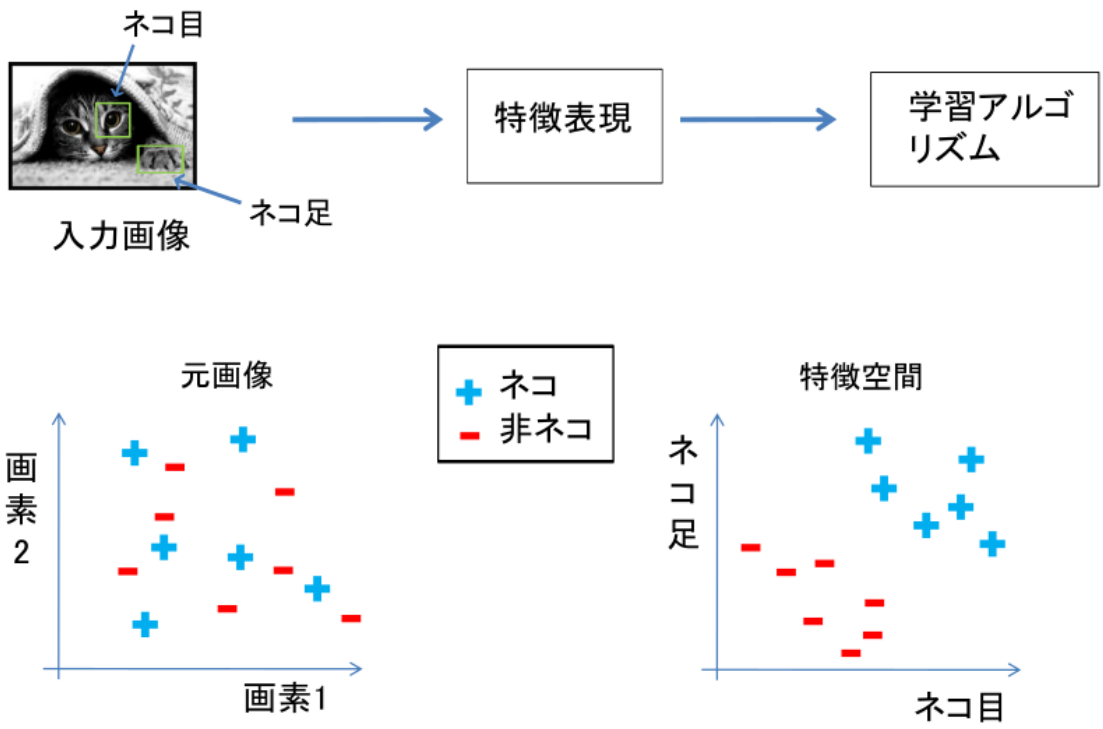

では入力画像がネコであるか否かを判断する画像認識であるとしました。我々はネコの画像を瞬時に判断できます。ですが画像認識の難しさは， 入力画像が上図 に示されているように入力信号の数字の集まりでしか無いことです。 このようなデータを何度も経験することでネコを識別できるようにする必要があります。 コンピュータに入力される画像は数字の塊に過ぎません。

状況ごとにとるべき操作を命令として逐一コンピュータに与える指示する手順の集まりのことをコンピュータプログラムと呼びます。人間がコンピュータに与えることができる操作や命令によって画像認識システムを作る場合，命令そのものが膨大になったり，そもそも説明することが難しかったりします。例を挙げれば，お母さんを思い浮かべてくださいと言われれば誰でも，それぞれ異なるイメージであれ思い浮かべることができます。また，提示された画像が自分の母親のものであるか，別の女性であるかの判断は人間であれば簡単です。ところがコンピュータには難しい課題となります。 加えて母親の特徴をコンピュータに理解できる命令としてプログラムすることも難しい課題です。つまり自分の母親の特徴を曖昧な言葉でなく明確に説明するとなるととても難しい課題となります。 というのは，女性の顔写真であればどの写真も似ていると言えるからです。顔の造形や輪郭，髪の位置などはどの画像も類似していることでしょう。ところがコンピュータにはこの似ている，似ていいないの区別が難しいのです。

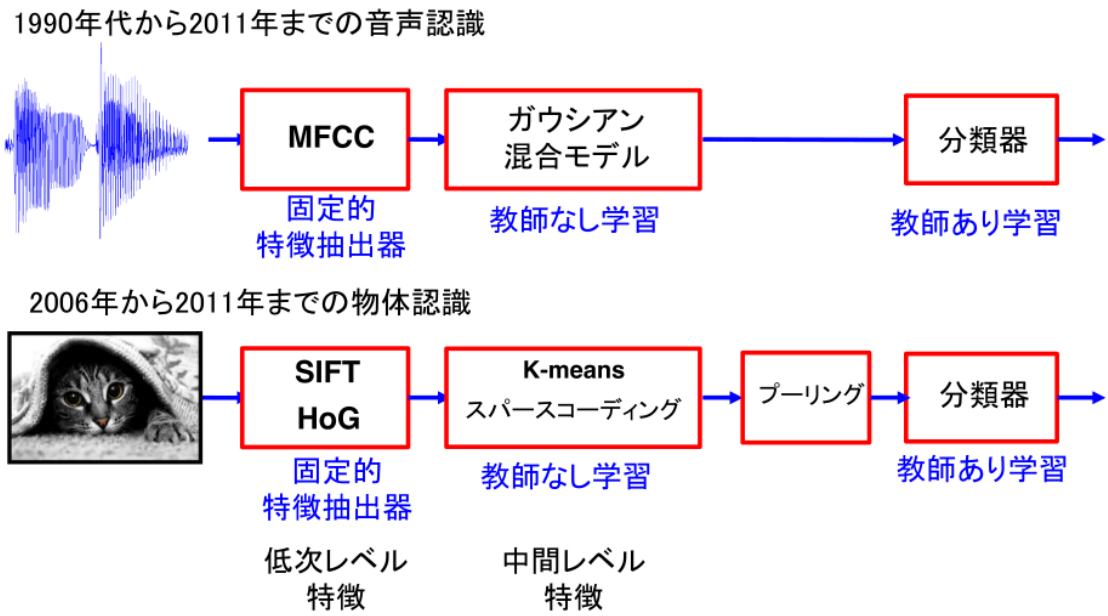
加えて，同一ネコの画像であっても，被写体の向き視線の方向や光源の位置や撮影条件が異なれば画像としては異なります。下図に示したように入力画像の中の特定の値だけを調べてみても，入力画像がネコであ る，そうではないかを判断することは難しい課題になります。



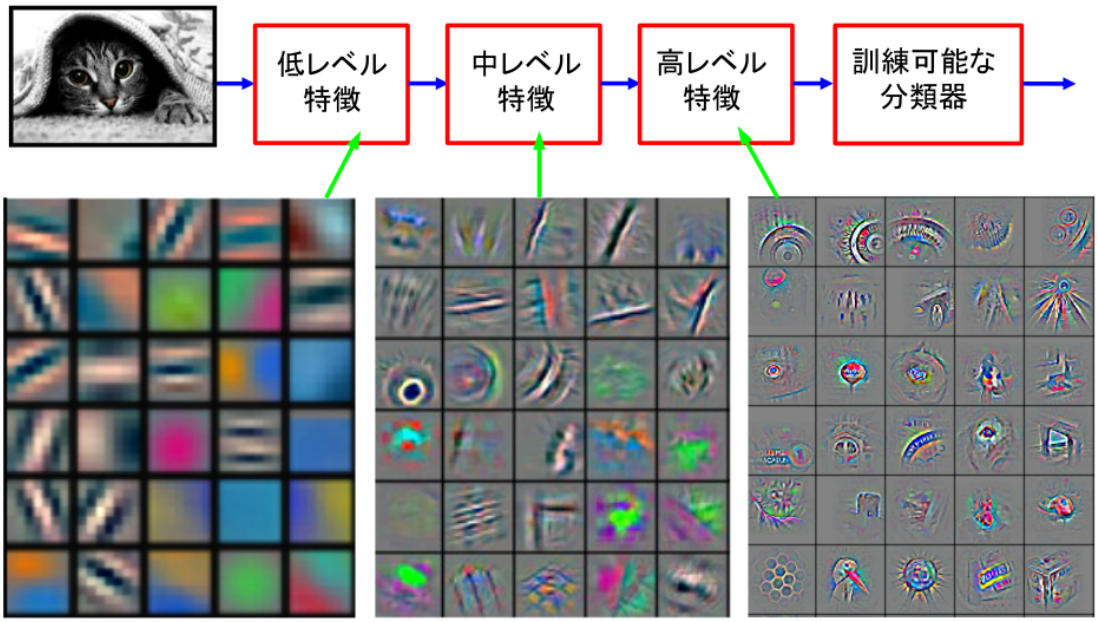

現在の画像認識では，特定の画素の情報に依存せずに，入力画像が持っている特徴 をとらえるように設計されます。たとえば，ネコを認識するために必要ことは，ネ コに特徴的な「ネコ目」や「ネコ足」を検出することであると考えます。入力画像 から，ネコの持つ特徴を抽出することができれば，それらの特徴を持っている入力 画像はネコであると判断して良いことになります(下図)。



下図 は，音声認識と画像認識の両分野においてCNN が用いられる以前の従来手法 をまとめたものです。



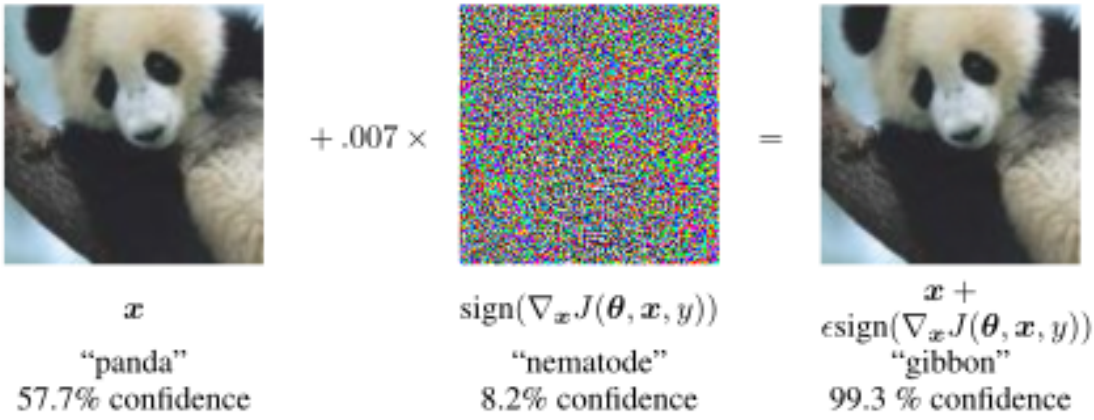

上図 のような従来手法に対して，CNN ではエンドツーエンドな特徴抽出を多層多段に重ねることによって複雑な特徴を抽出(検出)しようとしています (下図)。




コンピュータにはネコ目特徴検出器，ネコ足特徴検出器は備わっていません。そこで画像認識研究では，画像の統計的性質に基づいて特徴検出器を算出する方法を探す努力が行われてきました。しかし，コンピュータにネコ目特徴やネコ足特徴を教えるは容易なことではありません。このことは画像処理の分野だけに限りません，音声認識でも言語情報処理でもそれぞれの特徴器を一つ一つ定義し，チューニングするのは時間がかかり，専門的な知識も必要で困難な作業でした。

まとめると，1950 年代後半以来:固定的，手工芸的特徴抽出器と学習可能な分類器を用いた認識システムを作ることが試みられてきたといえます。これに対して CNN が主流となった現在はエンドツーエンドで学習可能な特徴抽出器を多数重ね合わせることで性能が向上しました。

夢のような話が続きましたが，本節の最後に逆に CNN は簡単に騙すことができる例 を挙げておきます。




図では，左の画像が入力画像で，CNN は確信度 57.7パーセントでパンダである認識しました。ところがこの画像に 0.007だけ意味のない画像(図中央)を加えるた画像(図右)を CNN は 99.3パーセントの確信度でテナガザル (gibbon)と判断しました。この例はここでは詳しく触れることはしませんが **敵対的学習**と呼ぶ訓練手法を説明する際に用いられた例です 。

この例からも分かることは以下のようにまとめられるでしょう。 すなわち，人間の脳を模したニューラルネットワークである CNN が大規模化像認識チャレンジにおいて人間の認識性能を越えたと報道されました。 ですが，人間の視覚認識を完全に実現したと考えるのは早計で，解くべき課題は未だ多数あるということです。 この状況は，音声認識や言語情報処理でも同様であると言えます。

- ドロップアウト，データ拡張，各種正規化: cnn.md
- 有名なモデル LeNet，Alex Net，Inception，VGG，ResNet
- R-CNN，ハイウェイネット，YOLO，SSD
- セマンティックセグメンテーション
- 転移学習，事前学習，ファインチューニング

---

## 活性化関数 activation functions

```
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline

## [Original](https://github.com/alrojo/tensorflow-tutorial/\
## blob/master/lab1_FFN/lab1_FFN.ipynb)

# PLOT OF DIFFERENT OUTPUT USNITS
x = np.linspace(-6, 6, 100)
relu = lambda x: np.maximum(0, x)
leaky_relu = lambda x: np.maximum(0, x) + 0.1*np.minimum(0, x)
elu = lambda x: (x > 0)*x + (1 - (x > 0))*(np.exp(x) - 1)
sigmoid = lambda x: (1+np.exp(-x))**(-1)
def softmax(w, t = 1.0):
    e = np.exp(w)
    dist = e / np.sum(e)
    return dist
x_softmax = softmax(x)

plt.figure(figsize=(6,6))
plt.plot(x, relu(x), label='ReLU', lw=2)
plt.plot(x, leaky_relu(x), label='Leaky ReLU',lw=2)
plt.plot(x, elu(x), label='Elu', lw=2)
plt.plot(x, sigmoid(x), label='Sigmoid',lw=2)
plt.legend(loc=2, fontsize=16)
plt.title('Non-linearities', fontsize=20)
plt.ylim([-2, 5])
plt.xlim([-6, 6])

# softmax
# assert that all class probablities sum to one
print(np.sum(x_softmax))
assert abs(1.0 - x_softmax.sum()) < 1e-8
```

デモファイル 2019si_activation_functions.ipynb

---

## 最終層（あるいは最終２層）は全結合層

ドット積を実行し、得られた結果に非線形活性化関数により出力を計算
ネットワーク全体は、一方の生画像ピクセルから他方のクラススコアまで、単一の微分可能なスコア関数を依然として表現している。 最上位層の全結合層には損失関数（SVM/Softmax）がある。通常のニューラルネットワークの学習と同じ手法を用いる

---

## CNN の詳細

通常のニューラルネットワークでは，直下層のニューロンとそのすぐ上の層の全ニューロンと結合を有する。一方 CNN ではその結合が部分的である。 各ニューロンは多入力一出力の信号変換機とみなすことができ，活性化関数に非線形な関数を用いる点は通常のニューラルネットワークと同様。

画像処理を考える場合，典型的には一枚の入力静止画画像は３次元データである。次元は幅w, 高さh, 奥行きd であり，入力画像では奥行きが３次元，すなわち赤緑青の三原色。出力ニューロンへの入力は局所結合から小領域に限局される。

## 1. CNNの構成

CNN は以下のいずれかの層から構成される：

1. **畳込み層**
2. **プーリング層**
3. **完全結合層**（通常のニューラルネットワークと正確に同じもの，CNN では最終 1 層または最終 1,2 層に用いる）

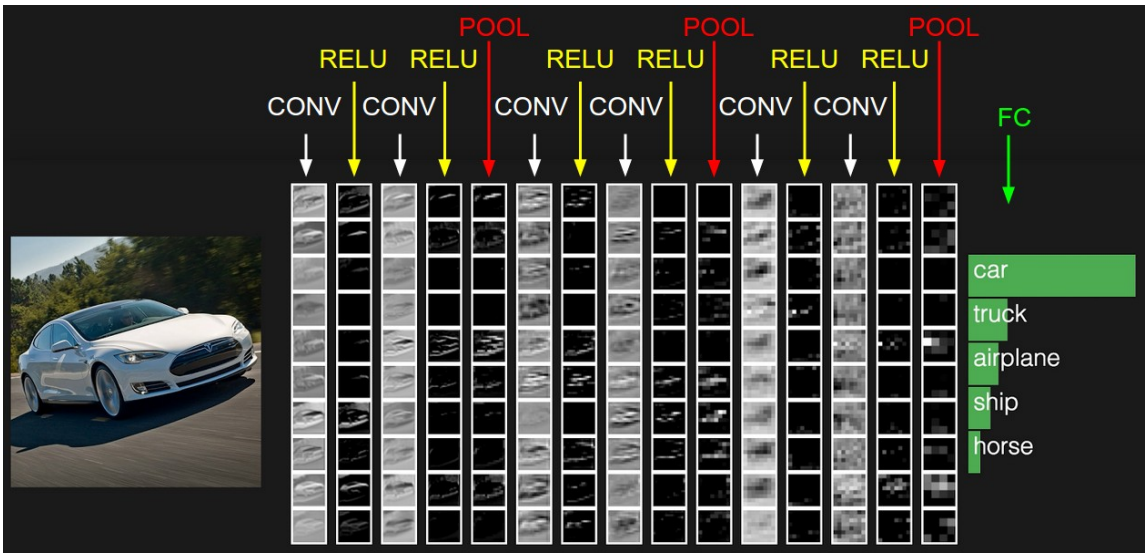
入力信号はパラメータの値が異なる活性化関数によって非線形変換される。 畳込み層とプーリング層と複数積み重ねることで多層化を実現し，深層ニューラルネットワークとなる。

---

**例：**

- 画像データを出力信号へ変換
- 各層は別々の役割（畳込み，全結合，ReLU, プーリング）
- 入力信号は 3 次元データ，出力信号も 3 次元データ
- 学習すべきパラメータを持つ層は畳込み層，全結合層
- 学習すべきパラメータを持たない層は ReLU 層とプーリング層
- ハイパーパラメータを持つ層は畳込み層, 全結合層, プーリング層
- ハイパーパラメータを持たない層は ReLU層

---




CNN アーキテクチャ: 入力層は生画像の画素値(左)を格納、最後層は分類確率(右)を出力。処理経路に沿った活性の各ボリュームは列として示されている。3Dボリュームを視覚化することは難しいため、各ボリュームのスライスを行ごとに配置してある。最終層のボリュームは各クラスのスコアを保持するが、ソートされた上位5スコアだけを視覚化し、それぞれのラベルを印刷してある。

---

- 入力層[32x32x3]: 信号は画像の生データ（画素値）幅w(32)，高さh(32)、色チャネル3(R, G, B)
- 畳込み層: 下位層の限局された小領域のニューロンの出力の荷重付き総和を計算(内積，ドット積）。12個のフィルタを使用すると[32x32x12]となる。
- ReLU層の活性化関数は ReLU (Recutified Linear Unit) $max(0, x)$
- プーリング層: 空間次元（幅,高さ）に沿ってダウンサンプリングを実行。[16x16x12]のようになる。

- 全結合層はクラスに属する確率を計算: 10 の数字のそれぞれが CIFAR-10 の 10 カテゴリーの分類確率に対応するサイズ[1x1x10]に変換。通常のニューラルネットワーク同様、全結合層のニューロンは前層の全ニューロンと結合する。

CNN は元画像（入力層）から分類確率（出力層）へ変換。学習すべきパラメータを持つ層（畳込み層，全結合層）とパラメータを持たない層（ReLU層）が存在。畳込み層と全結合層のパラメータは勾配降下法で訓練

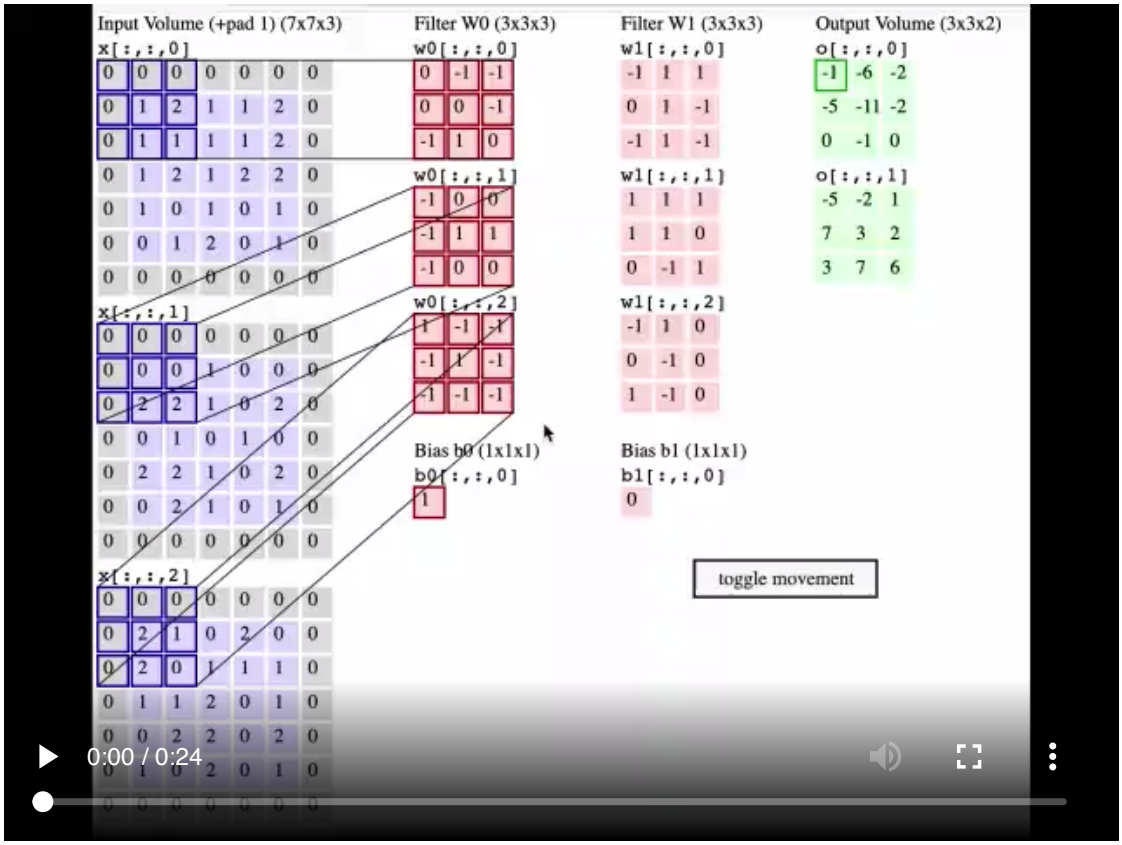
---

## 2. 畳込層

- 畳込み層のパラメータは学習可能なフィルタの組
- 全フィルタは空間的に（幅と高さに沿って）小さくなる
- フィルタは入力信号の深さと同一
- 第1層のフィルタサイズは例えば 5×5×3（5 画素分の幅，高さ，と深さ 3（３原色の色チャンネル）
- 各層の順方向の計算は入力信号の幅と高さに沿って各フィルタを水平または垂直方向へスライド
- フィルタの各値と入力信号の特定の位置の信号との内積（ドット積）。
- 入力信号に沿って水平，垂直方向にフィルタをスライド
- 各空間位置でフィルタの応答を定める 2 次元の活性化地図が生成される
- 学習の結果獲得されるフィルタの形状には、方位検出器，色ブロッブ，生理学的には視覚野のニューロンの応答特性に類似
- 上位層のフィルタには複雑な視覚パタンに対応する表象が獲得される
- 各畳込み層全体では学習すべき入力信号をすべて網羅するフィルタの集合が形成される
- 各フィルタは相異なる 2 次元の活性化地図を形成
- 各フィルタの応答特性とみなすことが可能な活性化地図
- フィルタの奥行き次元に沿って荷重総和を計算し、出力信号を生成

---

- 畳込み演算のデモ




---

**局所結合**: 画像のような高次元の入力を処理する場合，下位層の全ニューロンと上位層の全ニューロンとを接続することは **責任割当問題回避** の観点からもパラメータ数の増加は現実的ではない。代わりに各ニューロンを入力ボリュームのローカル領域のみに接続。空間的領域はニューロンの **受容野** と呼ばれるハイパーパラメータ（フィルタサイズとも言う）。深さ次元に沿った接続性＝入力層の深さ次元。 空間次元（幅と高さ）と深さ次元をどのように扱うかにより，この非対称性を再び強調することが重要です。ニューロン間の結合は空間次元（幅と高さ）にそって限局的。入力次元の深さ全体を常にカバーする。

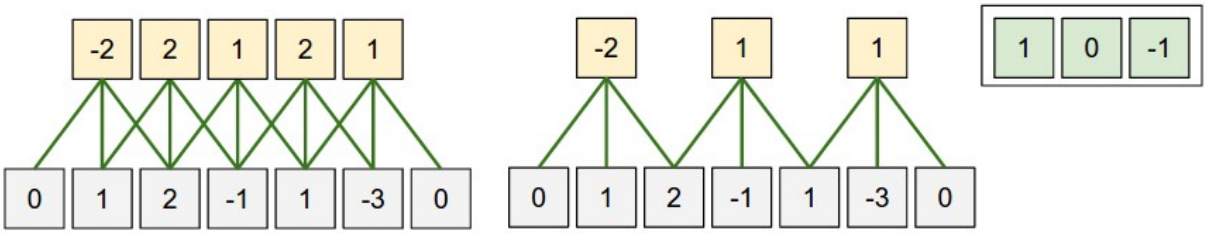
- 例1: 入力層のサイズが[32x32x3]（RGB CIFAR-10画像データセットなど）であれば受容野（フィルタサイズ）が 5x5 とすれば，畳込み層内の各ニューロンは入力層の [5x5x3] 小領域への結合係数を持つ。各小領域毎に 5x5x3=75 の重み係数と 1 つのバイアス項が必要である。深さ次元に沿った上層のニューロンから下位層のニューロンへの結合は下位層の深さ(色チャンネル数)と等しく 3 である。
- 例2: 入力ボリュームのサイズが[16x16x20]であるとすると 3x3 の受容野サイズで畳込層の全ニューロンの合計は 3x3x20=180 接続。接続性は空間的に局在する（3x3）が，入力深度（20）に沿っては完全結合

**空間配置**: 出力層ニューロンの数と配置については 3 つのハイパーパラメータで出力ニューロン数が定まる。

1. **深さ数(フィルタ数)**
2. **ストライド幅**
3. **ゼロパディング**
4. 出力層ニューロン数のことを出力層の **深さ** 数と呼ぶハイパーパラメータである。深さ数とはフィルタ数（カーネル数）とも呼ばれる。第 1 畳込み層が生画像であれば，奥行き次元を構成する各ニューロンによって種々の方位を持つ線分(エッジ検出細胞)や色ブロッブのような特徴表現を獲得可能となる。入力の同じ領域を **深さ列** とするニューロン集団を **ファイバ** ともいう。

5. フィルタを上下左右にずらす幅を **ストライド幅** と呼ぶ。ストライド幅が 1 ならフィルタを 1 画素ずつ移動することを意味する。ストライドが 2ならフィルタは一度に 2 画素ずつジャンプさせる。ストライド幅が大きければ入力信号のサンプリング間隔が大きく広がることを意味する。ストライド幅が大きくなれば上位層のニューロン数は減少する。
6. 入力の境界上の値をゼロで埋め込むことがある。これを **ゼロパディング** という。ゼロパディングの量はハイパーパラメータである。ゼロパディングにより出力層ニューロンの数を制御できる。下位層の空間情報を正確に保存するには入力と出力の幅，高さは同じである必要がある。

入力層のニューロン数を$W$，上位にある畳込み層のニューロン数を$F$，とすれば出力層に必要なニューロン数$S$は，周辺のゼロパディング を$P$とすれば $(W-F+2P)/S+1$ で算出できる。たとえば下図でストライド 1 とゼロパディング 0 であれば入力 7x7 でフィルタサイズが 3x3 であれば 5x5(=S=(7-3+2x0)/1+1=5) の出力である。ストライド 2 ならば 3x3=(S=(7-3+2x0)/2+1=3) となる。




空間配置の例：入力空間の次元（x軸）が1つで受容野サイズ F=3 の場合，入力サイズ W=5, ゼロパディング P=1 であれば,
**左図：**出力層ニューロン数は (5-3+2)/1+1=5 の出力層ニューロン数となる。ストライド数 S=1 の場合。
**右図：**s=2，出力層ニューロン数 (5-3+2)/2+1=3 となる。ストライド S=3 ならばボリューム全体にきちんと収まらない場合もでてくる。数式で表現すれば \\((5-3+2)=4\\) は 3 で割り切れないので、整数の値として一意に決定はできない。
ニューロン結合係数は（右端に示されている）[1,0,-1]でありバイアスはゼロ。この重みはすべての黄色ニューロンで共有される。

---

*ゼロパディング*: 上例では入力次元が 5，出力次元が 5 であった。これは受容野が 3 でゼロ埋め込みを1としたためである。ゼロ埋め込みが使用されていない場合、出力ボリュームは、どれだけの数のニューロンが元の入力に「フィット」するのであろうかという理由で、空間次元がわずか3であったであろう。ストライドが $S=1$ のとき、ゼロ埋め込みを $P=(F-1)/2$ に設定すると、入力ボリュームと出力ボリュームが空間的に同じサイズになる。このようにゼロパディングを使用することは一般的である。CNNについて詳しく説明している完全な理由について説明する。

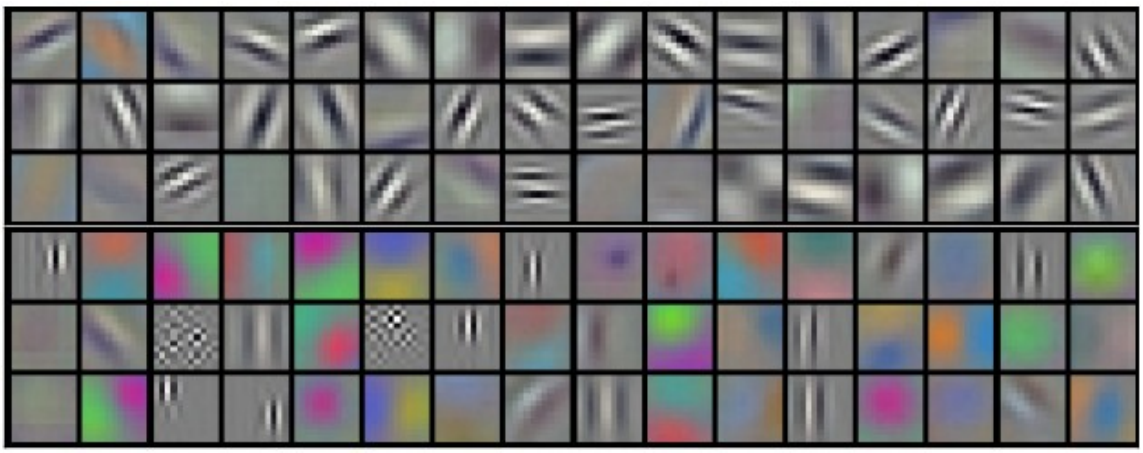
*ストライドの制約*: 空間配置ハイパーパラメータには相互の制約があることに注意。たとえば入力に$W=10$というサイズがあり、ゼロパディングは$P=0$ ではなく、フィルタサイズは$F=3$, $(W-F+2P)/S+1=(10-3+0)/2+1=4.5$よりストライド $S=2$ を使用することは不可能である。すなわち整数ではなくニューロンが入力にわたってきれいにかつ対称的に "適合" しないことを示す。

*AlexNet* の論文では，第一畳込層は受容野サイズ $F=11$，ストライド$S=4$，ゼロパディングなし$P=0$。
畳込層 $K=96$ の深さ $(227-11)/4+1=55$。畳込層の出力サイズは [55x55x96]。55x55x96 ニューロンは入力領域 [11x11x3] と連結。全深度列 96 個のニューロンは同じ入力領域[11×11×3]に繋がる。論文中には(224-11)/4+1 となっている。パディングについての記載はない。

**パラメータ共有** パラメータ数を制御するために畳み込み層で使用される。上記の実世界の例を使用すると、最初の畳故意層には 55x55x96=290,400のニューロンがあり、それぞれ 11x11x3=363 の重みと1のバイアスがある。これにより CNN 単独の第 1 層に最大 290400x364=105,705,600 のパラメータが追加される。

**パラメータ共有** により学習すべきパラメータ数が減少する。 例えば [55x55x96] のフィルタでは深さ次元は 96 個のニューロンで，各深さで同じ結合係数を使うことにすれば ユニークな結合係数は計 96x11x11x3=34,848 となるので総パラメータ数は 34,944 となる(バイアス項 +96)。各深さで全ニューロン(55x55)は同じパラメータを使用する。逆伝播での学習では，全ニューロンの全結合係数の勾配を計算する必要がある。各勾配は各深さごとに加算され 1 つの深さあたり一つの結合係数集合を用いる。

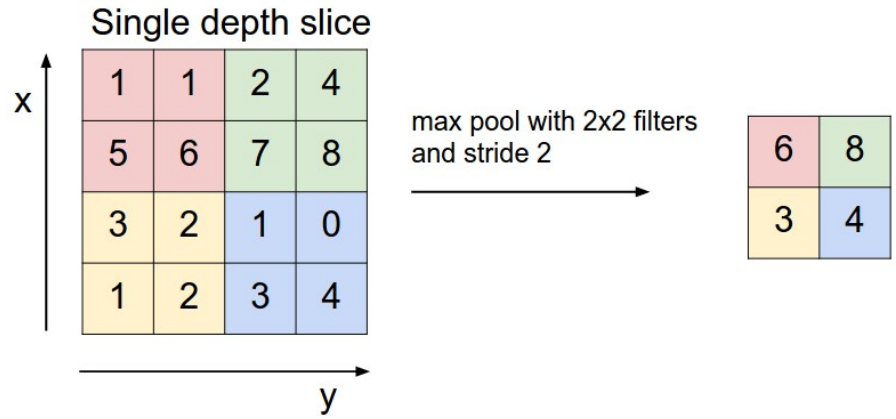
ある深さの全ニューロンが同じ重み係数ベクトルを共有する場合，畳込み層の順方向パスは各深さスライス内で入力ボリュームとのニューロンの重みの **畳み込み** として計算できることに注意。結合荷重係数集合のことを **フィルタ** または **カーネル** と呼ぶ。入力信号との間で畳込み演算を行うこととなる。



AlexNet の学習済フィルタ例：図の 96 個のフィルタは サイズ[11x11x3]。それぞれが 1 つの深さ内の 55×55 ニューロンで共有されている。画像の任意の位置で水平エッジ検出が必要な場合，画像の並進不変構造 translationall-invariant structure 仮定により画像中の他の場所でも有効である。 畳込み層の出力ニューロン数は 55x55 個の異なる位置すべてで水平エッジの検出を再学習する必要はない。

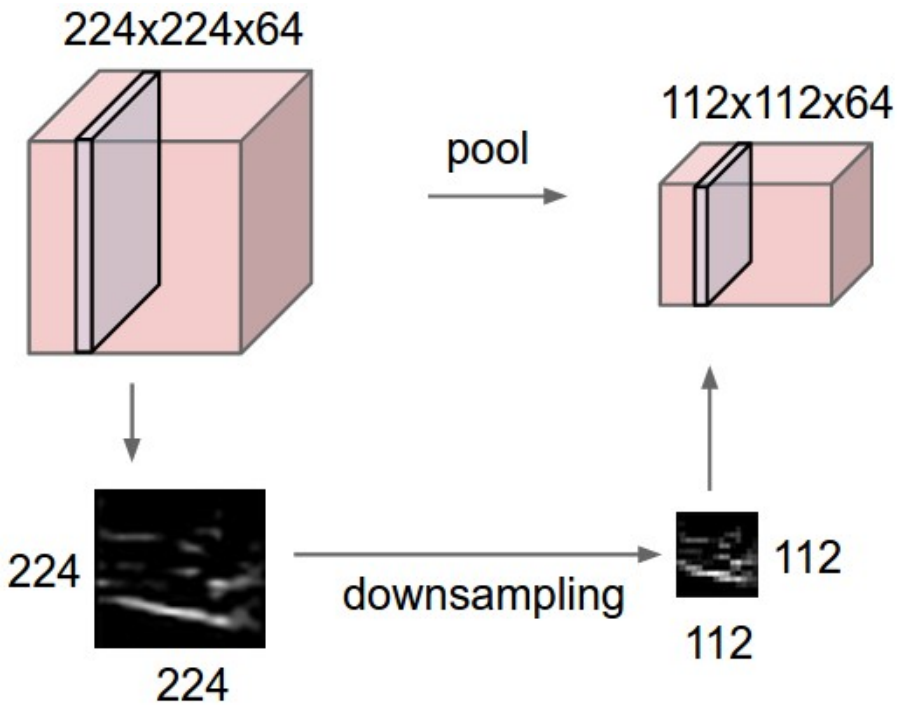
## 3. プーリング層

CNN では，連続する畳込み層間にプーリング層を挿入するのが一般的。プーリング層の役割は，空間次元の大きさに減少させることである。パラメータ，すなわち計算量を減らし，過学習を制御できる。プーリング層は入力の各深さ毎に独立して動作する。最大値のみをとり他の値を捨てることを **マックスプーリング** と呼ぶ。サイズが 2x2 のフィルタによるプーリング層では，入力の深さごとに 2 つのダウンサンプルを適用し、幅と高さに沿って2ずつ増やして75%の情報を破棄する。この場合 4 つの数値のうち最大値を採用することになる。




一般的なダウンサンプリング演算は **マックスプーリング** である。図では ストライド 2 すなわち 4 つの数値の中の最大値

**平均プーリング**. マックスプーリングではなく *L2正則化プーリング* を行う場合もある。平均プーリングは歴史的な意味あいがあるがマックスプーリングの方が性能が良いとの報告がある。ある画像位置には物理的に一つの値だけが存在するという視覚情報処理が仮定すべき外界の物理的制約を反映していると文学的に解釈することも可能である。




プーリング層では，入力層ニューロン数の各深さについて空間的ダウンサンプリングを行う。この例は サイズ[224x224x64]の入力層ニューロン数がフィルタサイズ 2 でプールされ，サイズ 2 の出力ニューロン数 [112x112x64] は 2 倍である。奥行き数が保持されている。

## 4. 全結合層

全結合層のニューロンは、通常のニューラルネットワークと同じ
前層の全ニューロンと結合を持つ

## 5. CNN アーキテクチャ

1. 畳込層
2. プーリング層
3. 全結合層

層は以上 3 種類が一般的。

## 6. CNN の層構造

入力層 → [[畳込層 → ReLU]×$N$ → プーリング(?)]× M → [全結合層 → ReLU] × K → 全結合層

最近のトレンドとしては大きなフィルタより小さなフィルタが好まれる傾向にある。
[3x3] が好まれる理由はど真ん中がある奇関数を暗黙に仮定しているためだと思われる（浅川の妄想）。その代わり多段にすれば [3x3] が２層で ［5x5], ３層で[7x7]の受容野を形成できるから受容野の広さを層の深さとして実装しているとも解釈できる。１層で[7x7]の受容野より３層で[7x7]の受容野を実現した方が the simpler, the better の原則に沿っているとも（文学的）解釈が可能である（またしても浅川妄想）。

バックプロパゲーションの計算時に広い受容野を作るより層を分けた方が GPU のメモリに乗せやすいと言う計算上の利点もある。