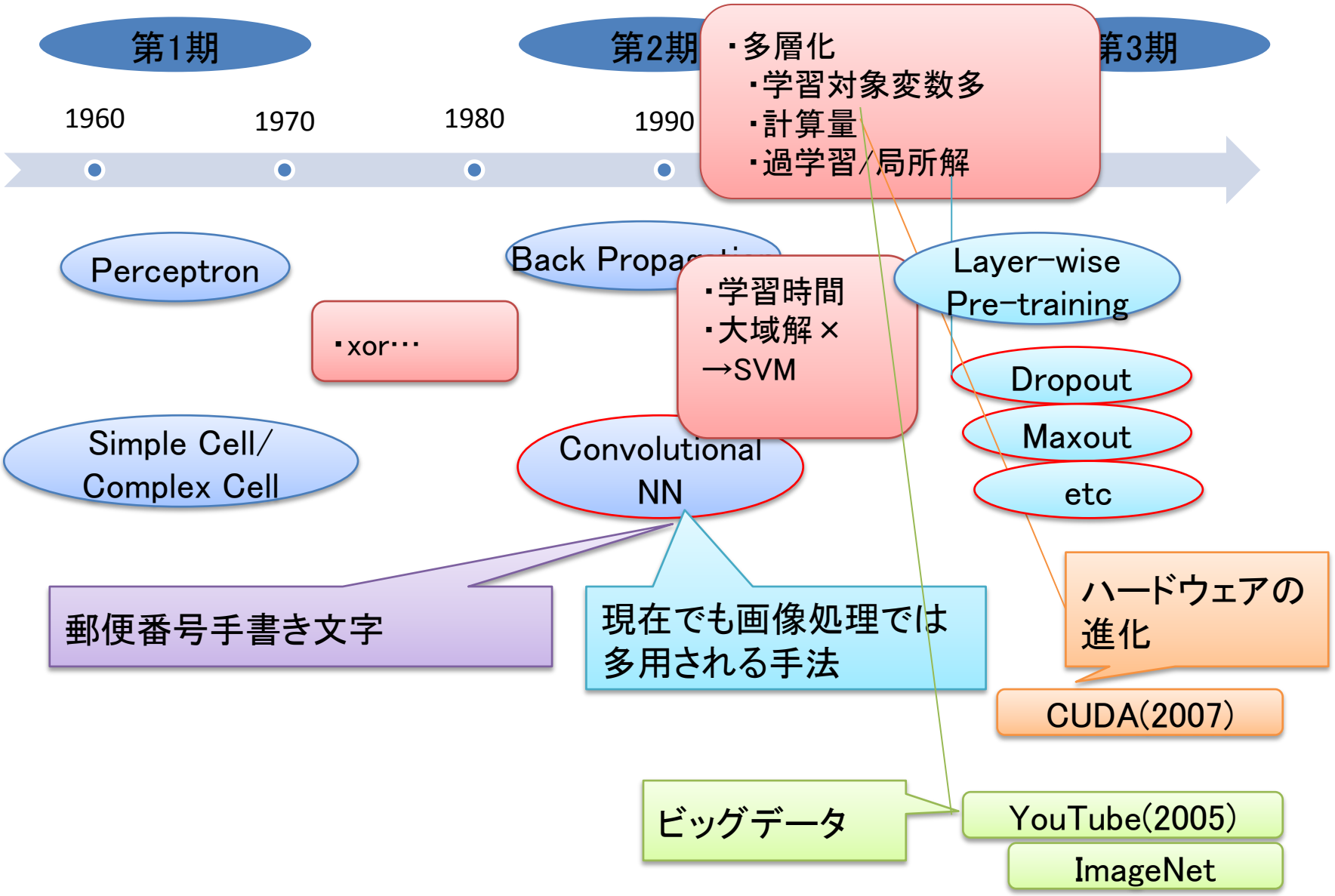
Deep Learningのアルゴリズム

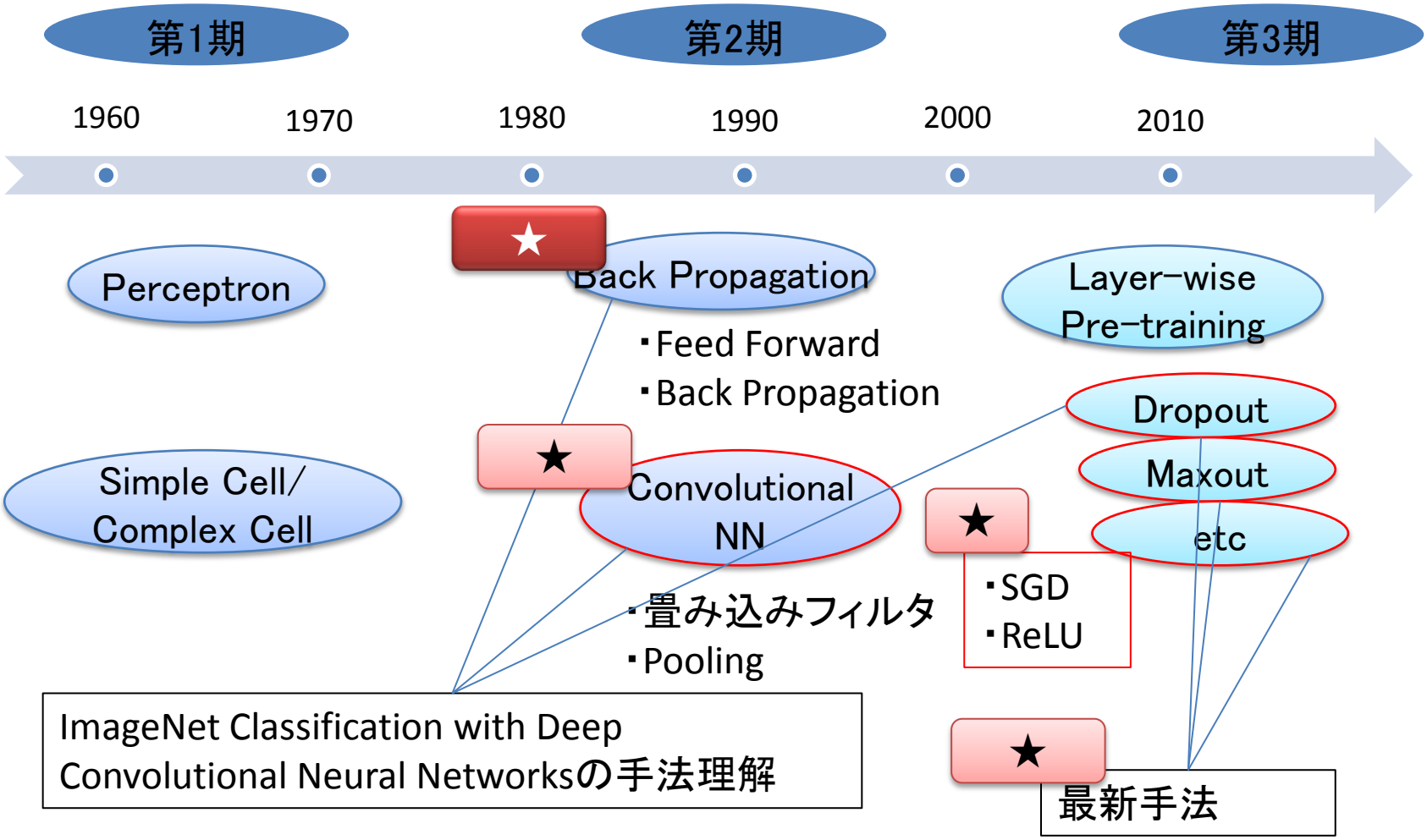
2014年9月5日

G-DEPソリューションパートナー
株式会社 システム計画研究所
奥村 義和

Neural Networkの進化

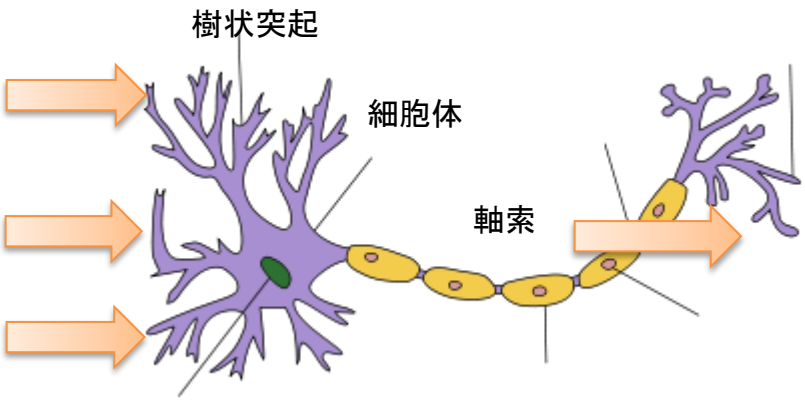


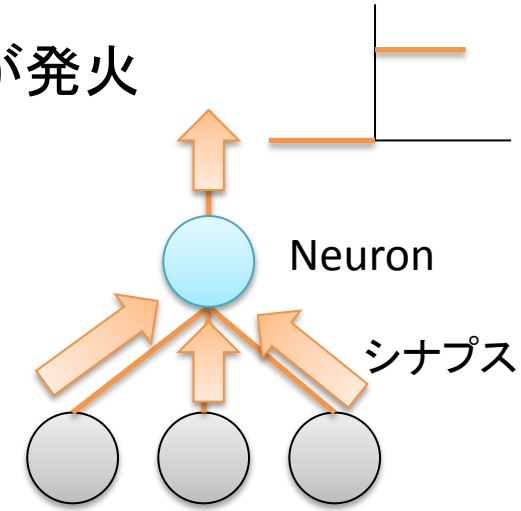
本日の話題


第1期
第2期
第3期
1960
1970
1980
1990
2000
2010
Perceptron
Back Propagation
・Feed Forward
・Back Propagation
Layer-wise
Pre-training
Dropout
Maxout
etc
Simple Cell/
Complex Cell
Convolutional
NN
・畳み込みフィルタ
・Pooling
・SGD
・ReLU
ImageNet Classification with Deep
Convolutional Neural Networksの手法理解
最新手法

ニューラルネットワークのモデル

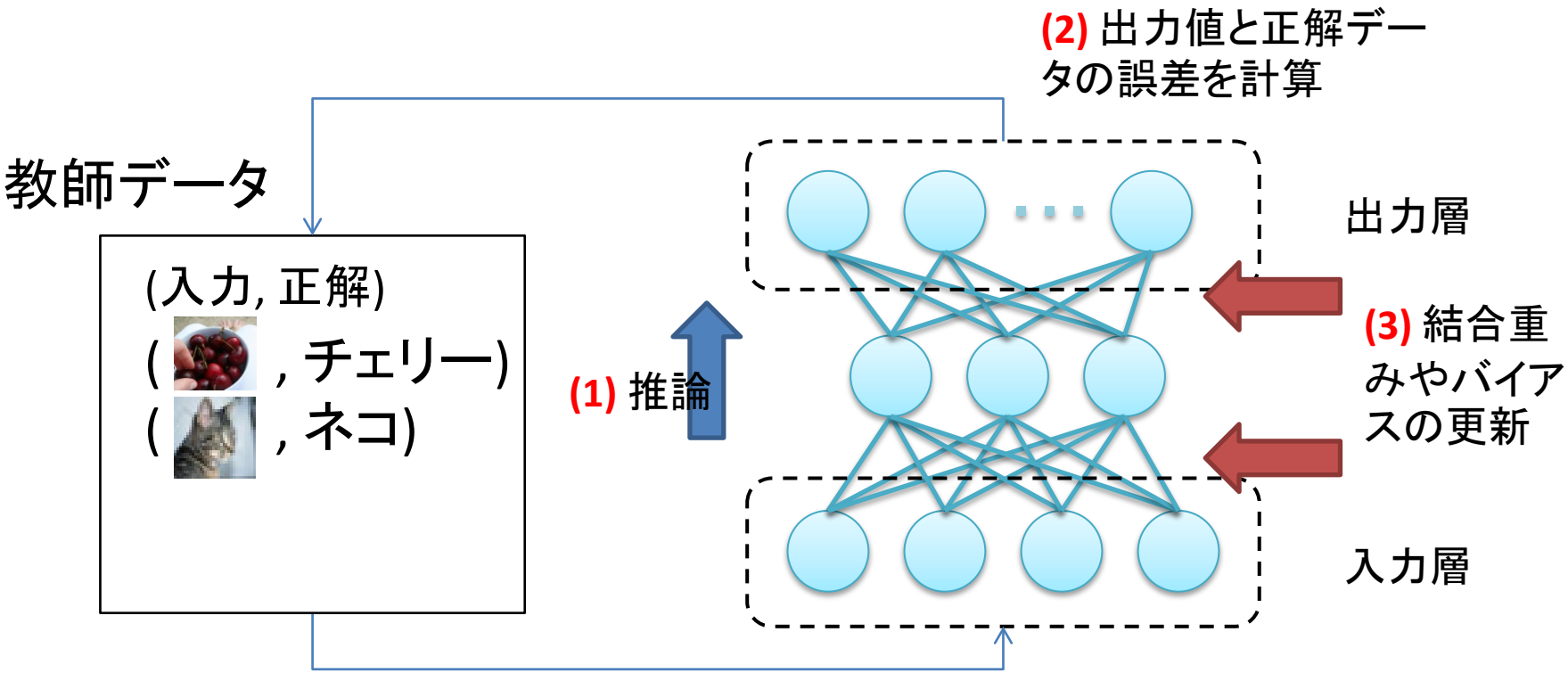
- ニューロンの動作原理を模式化
  - 樹状突起から他のニューロンの刺激を受信
    →閾値を超えるとニューロンが発火
    →他のニューロンへ刺激を伝達
- 人工ニューロン
  - 0, 1を出力(初期モデル)
  - ニューロンが結合。結合には重み。
  - 入力がある閾値を超えるとニューロンが発火





http://ja.wikipedia.org/wiki/%E7%A5%9E%E7%B5%8C%E7%B4%B0%E8%83%9E

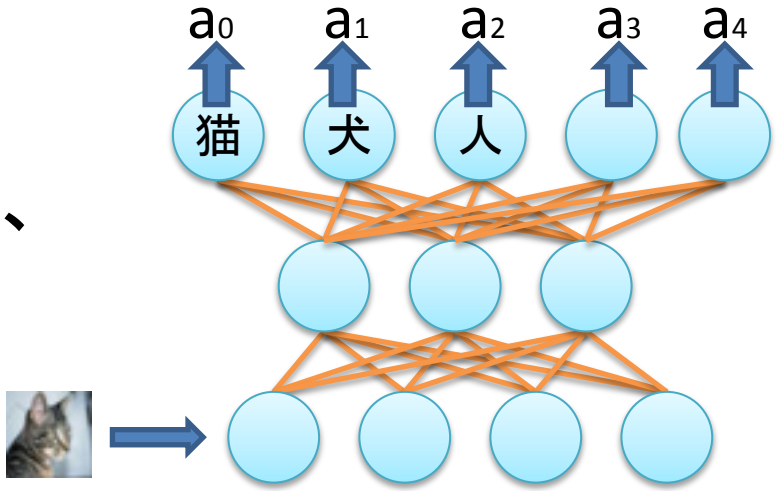
学習の全体フロー

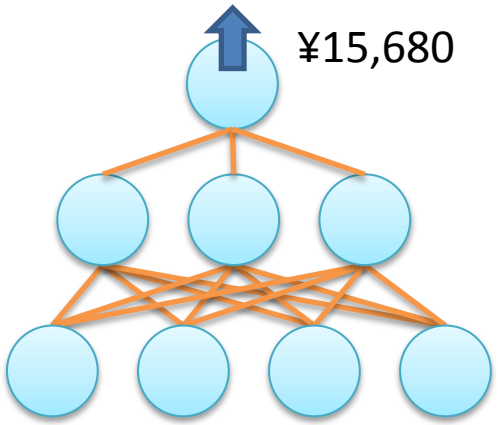


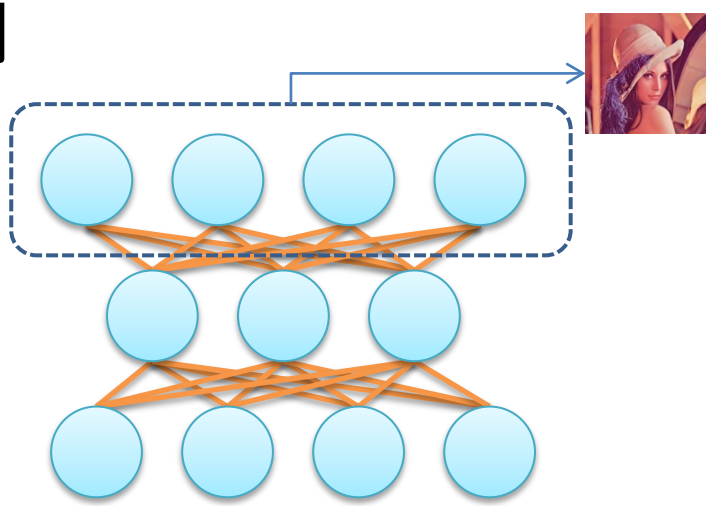
- 誤差が収束するまで(1)～(3)の学習を繰り返す

ニューラルネットワークの基本
適用領域と出力

- 分類問題
  - 出力層にSoftmaxを適用、確率に。
    $p(i) = \exp(a_i) / \sum \exp(a_i)$
- 回帰問題
  - →出力値をそのまま利用







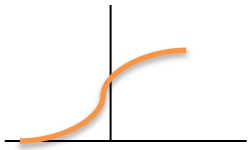
ニューラルネットワークの基本
Feed Forward

- $Y_i = f(\sum_j w_{i,j} * x_j + b_i), \quad f(u) = \frac{1}{1+\exp(-u)}$

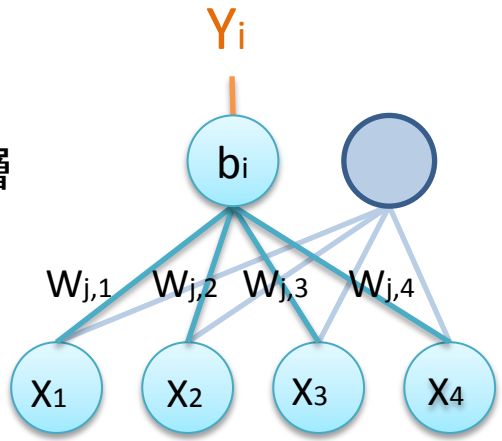
  h

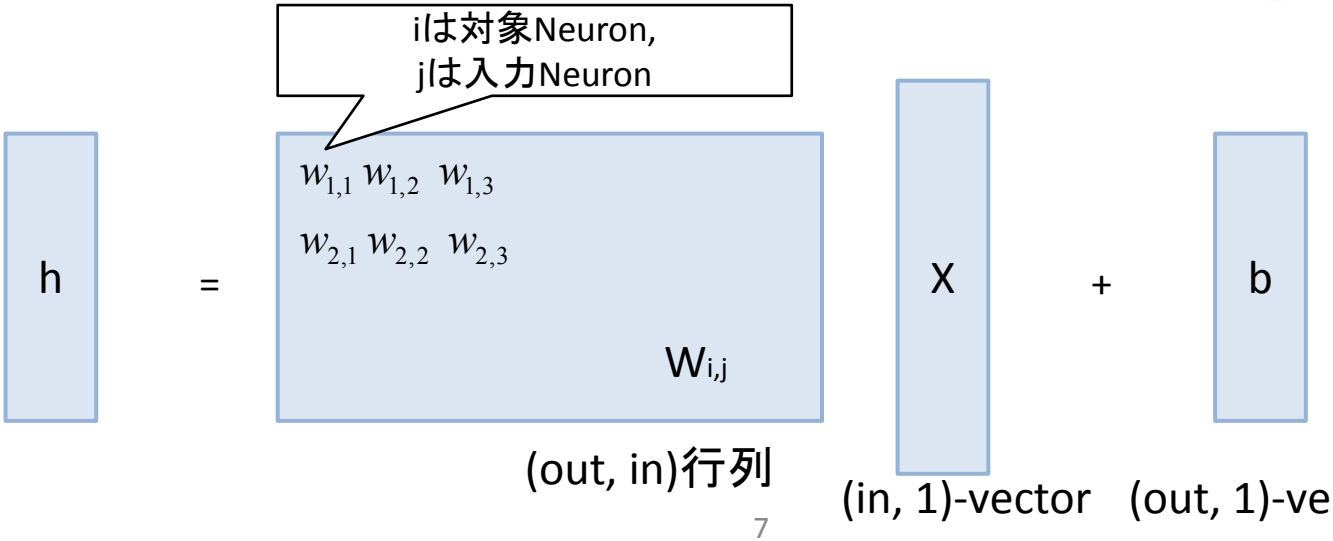
  f: 非線形関数

  (step, logistic, tanh)



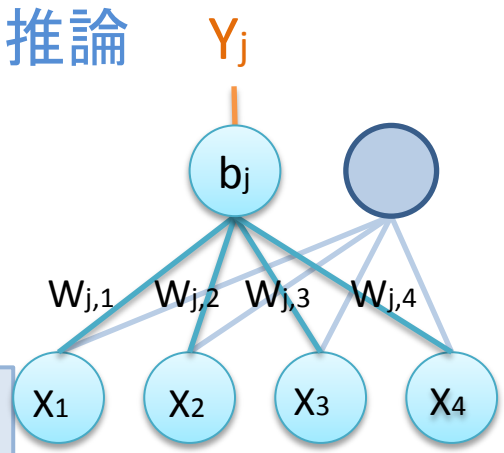
- Xを縦ベクトルと考えると、以下の様な計算イメージ

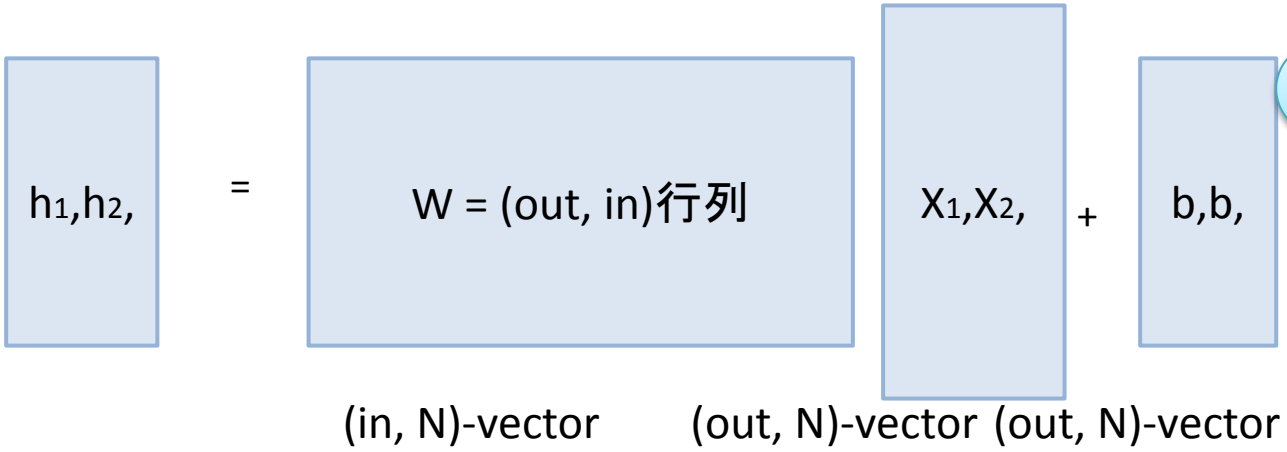




ニューラルネットワークの基本
Feed Forward計算イメージ

- 学習時は複数の教師データを
  まとめて扱う
  →Xは行列のイメージ



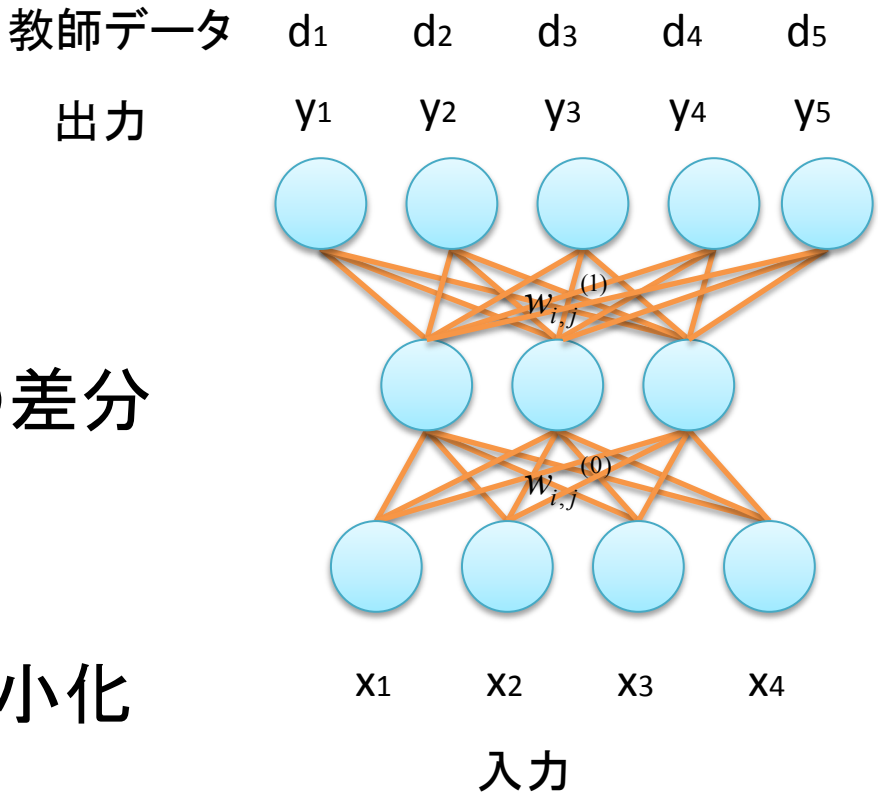


- CUDAやBLASによる高速計算が可能

ニューラルネットワークの基本
学習の定式化

- 学習＝
  教師データに対し
  正解を答える
- 定式化
  E: コスト関数
  　＝出力と教師データの差分
  x, y: 入力・出力
  d: 教師データ
  W, b: 重み・バイアス
- E(x,d,W,b)をW,bについて最小化

$$E(x,d,W,b)=\frac{1}{2}\| y-d \|^2=\frac{1}{2}\sum_i (y_i-d_i)^2$$



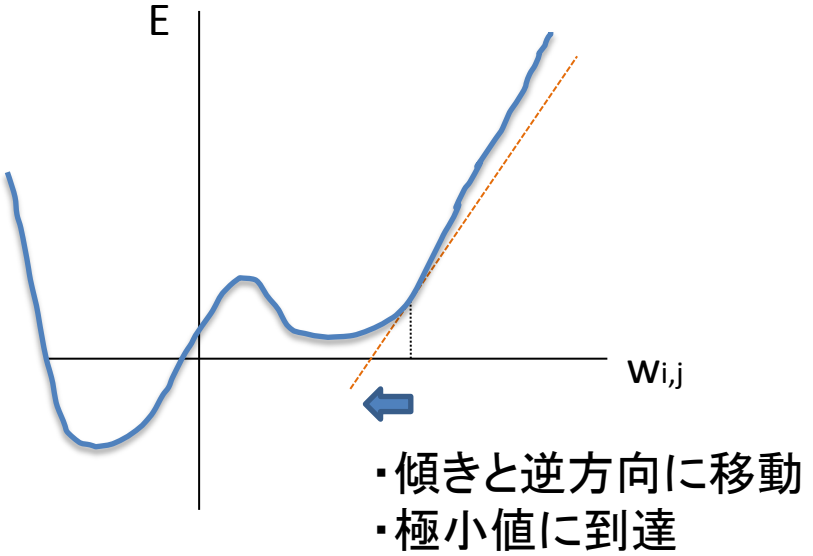
ニューラルネットワークの基本
Back Propagation

- 形式的に解くのは難しいため、最急降下法を用いて近似計算。

$$w_{i,j} = w_{i,j} - \rho \frac{\partial E}{\partial w_{i,j}}$$

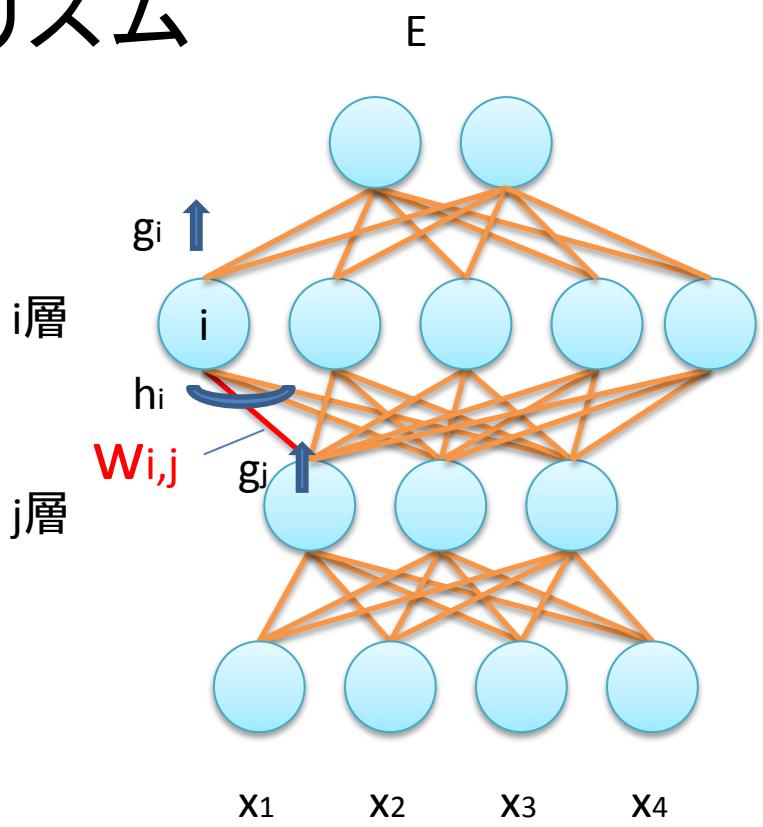
ρは正定数(学習係数)

- 局所解に陥る可能性あるが、妥当な結果が得られればOK。
- 赤線部を求めよ



ニューラルネットワークの基本
Back Propagation 記号の導入

- $\frac{\partial E}{\partial w_{i,j}}$ を求める為のアルゴリズム



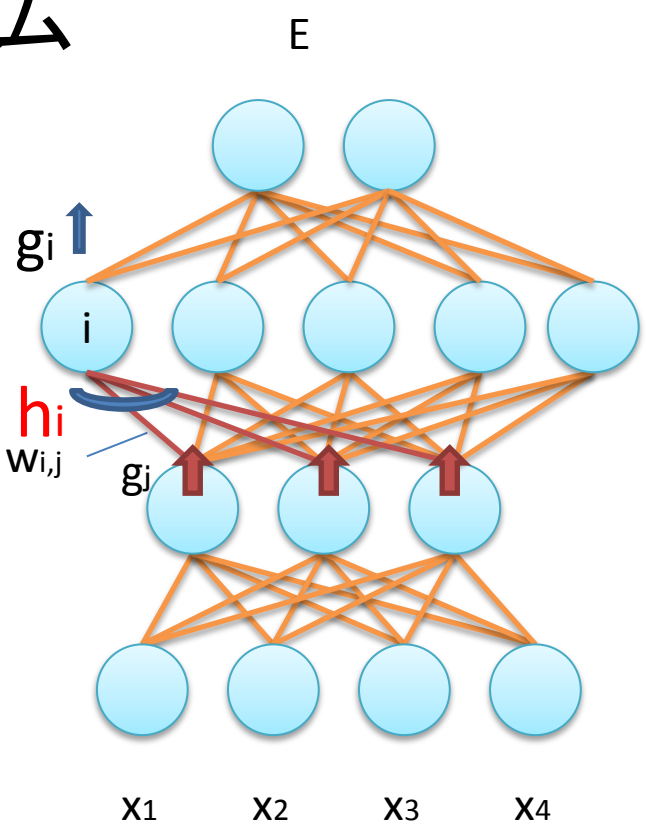
ニューラルネットワークの基本
Back Propagation 記号の導入

- $\frac{\partial E}{\partial w_{i,j}}$ を求める為のアルゴリズム

・記号の導入

ニューロンiへの入力：$h_i = \sum_j w_{i,j} g_j$

ニューロンiの出力：$g_i = f(h_i)$



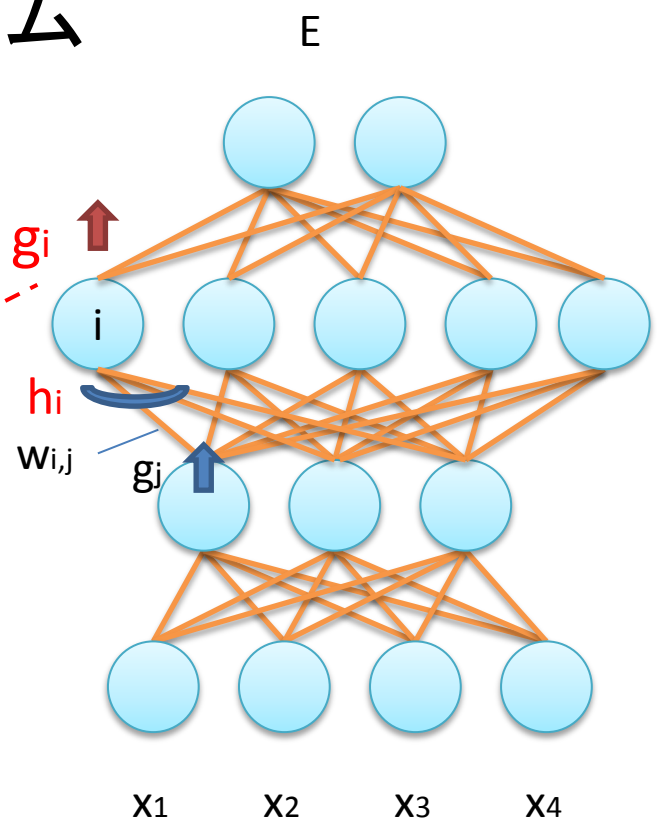
ニューラルネットワークの基本
Back Propagation 記号の導入

- $\frac{\partial E}{\partial w_{i,j}}$ を求める為のアルゴリズム

・記号の導入

ニューロンiへの入力：$h_i = \sum_j w_{i,j} g_j$

ニューロンiの出力：$g_i = f(h_i)$



ニューラルネットワークの基本
Back Propagation 記号の導入

- $\frac{\partial E}{\partial w_{i,j}}$ を求める為のアルゴリズム

・記号の導入

ニューロンiへの入力：$h_i = \sum_j w_{i,j} g_j$

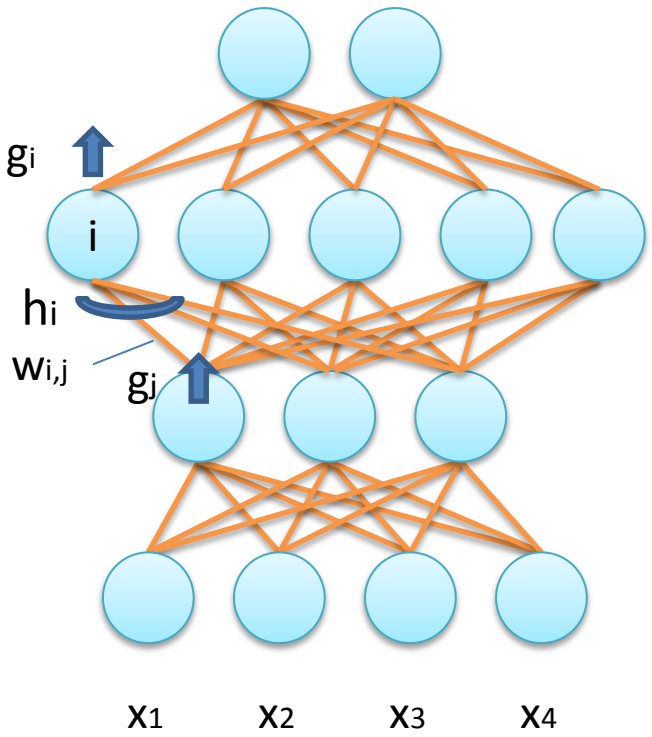
ニューロンiの出力：$g_i = f(h_i)$

・式変形

$$\frac{\partial E}{\partial w_{i,j}} = \frac{\partial h_i}{\partial w_{i,j}} \frac{\partial E}{\partial h_i} = g_j \frac{\partial E}{\partial h_i}$$

$\frac{\partial E}{\partial h_i}$を求めることに問題が帰着

Neuron iの誤差



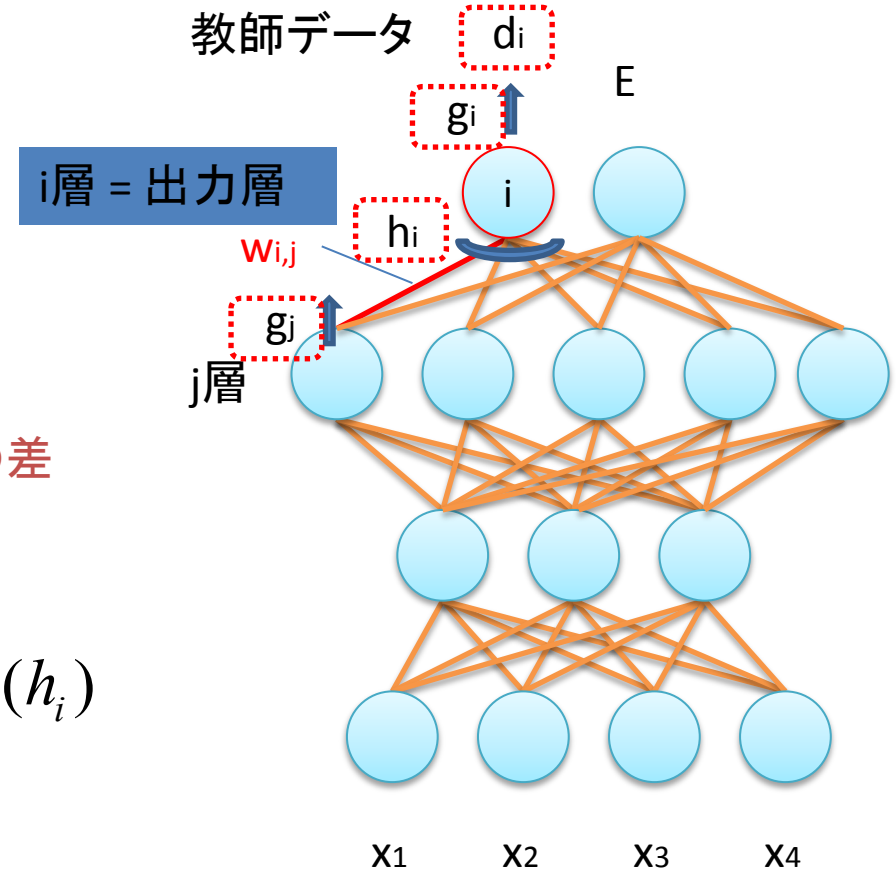
ニューラルネットワークの基本
Back Propagation 出力層

$\frac{\partial E}{\partial h_i}$ を求める

(1)iが出力層

$$\frac{\partial E}{\partial h_i}=\frac{\partial E}{\partial g_i}\frac{\partial g_i}{\partial h_i}=(g_i-d_i)f'(h_i)$$

出力と教師データの差



- 勾配計算のアルゴリズム

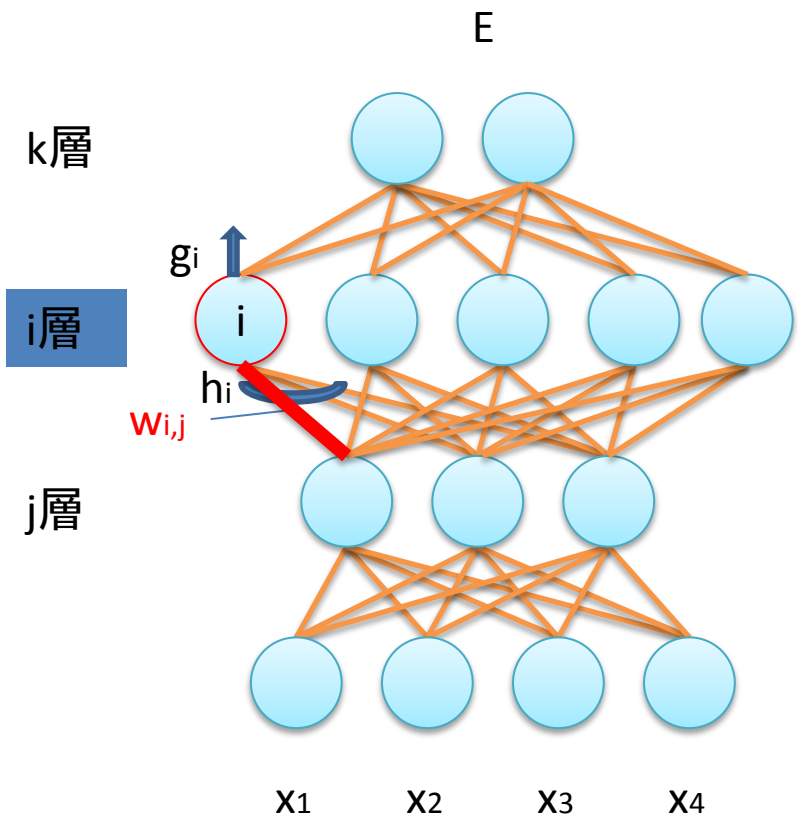
$$\frac{\partial E}{\partial w_{i,j}}=g_j\frac{\partial E}{\partial h_i}=g_j(g_i-d_i)f'(h_i)$$

ニューラルネットワークの基本
Back Propagation (出力層-1) 層

$\frac{\partial E}{\partial h_i}$を求める

(2)iが出力層の一つ手前

$$\frac{\partial E}{\partial h_i} = \frac{\partial E}{\partial g_i}\frac{\partial g_i}{\partial h_i}$$



ニューラルネットワークの基本
Back Propagation (出力層-1) 層

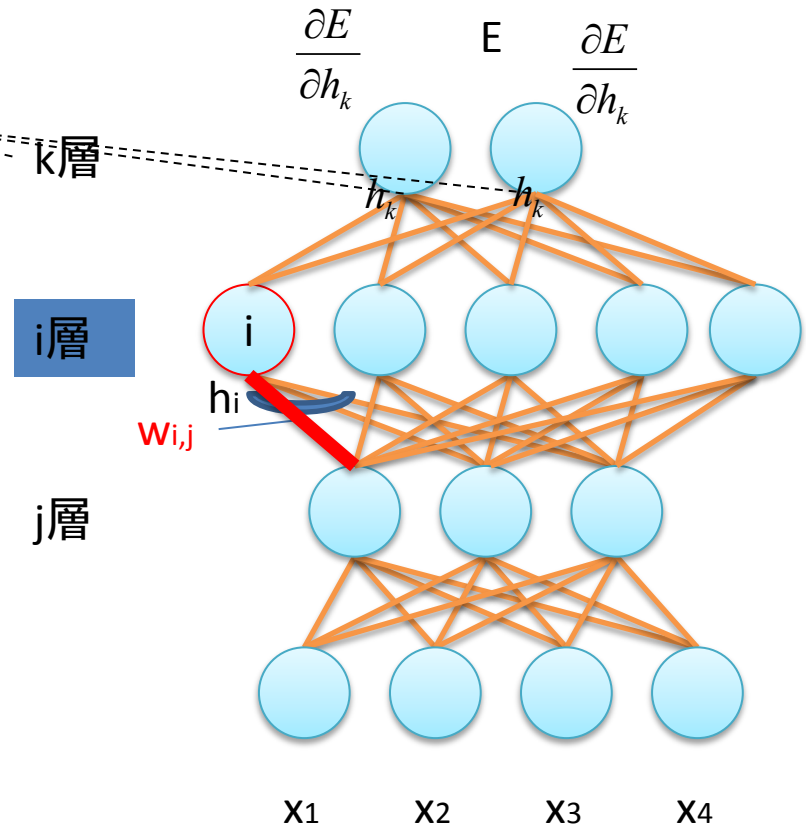
$\frac{\partial E}{\partial h_i}$を求める

$$E \propto E(h_{k_1}, h_{k_2}, \ldots, h_{kn})$$

$$h_k = h_k(g_i)$$

(2)iが出力層の一つ手前

$$\frac{\partial E}{\partial h_i} = \frac{\partial E}{\partial g_i}\frac{\partial g_i}{\partial h_i} = \left(\sum_{\mathrm{k}} \frac{\partial \mathrm{E}}{\partial \mathrm{h_k}}\frac{\partial h_k}{\partial g_i}\right)\frac{\partial g_i}{\partial h_i}$$



ニューラルネットワークの基本
Back Propagation (出力層-1) 層

$\frac{\partial E}{\partial h_i}$を求める

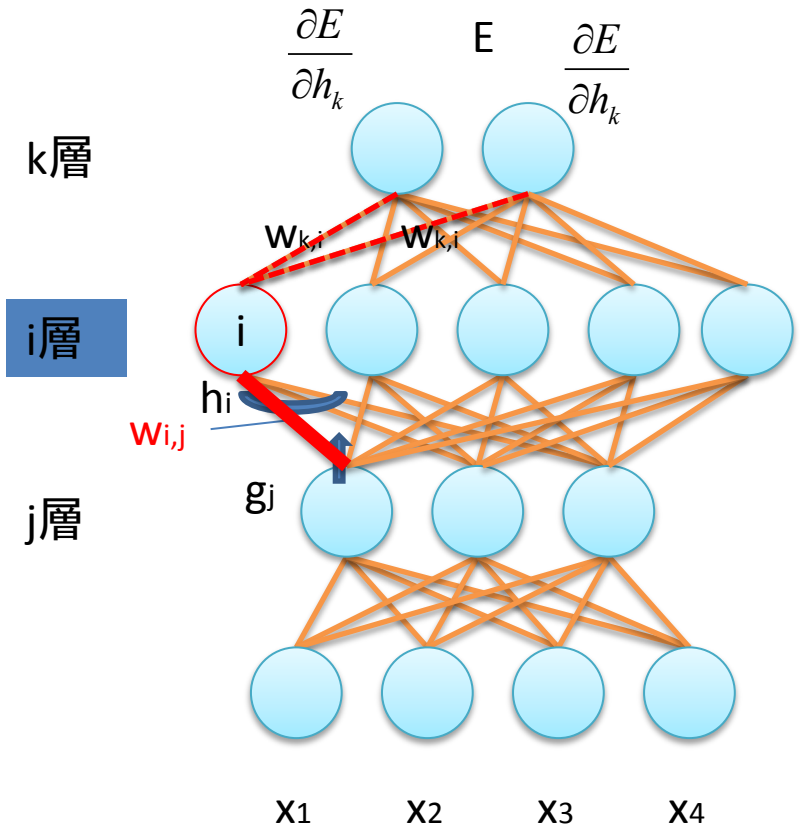
(2)iが出力層の一つ手前

$$\frac{\partial E}{\partial h_i} = \frac{\partial E}{\partial g_i}\frac{\partial g_i}{\partial h_i} = (\sum_k \frac{\partial \mathrm{E}}{\partial \mathrm{h_k}}\frac{\partial h_k}{\partial g_i})\frac{\partial g_i}{\partial h_i}$$

$$h = \sum wg \qquad = (\sum_k \frac{\partial \mathrm{E}}{\partial \mathrm{h_k}} w_{k,i}) f'(h_i)$$

$$g = f(h)$$

一つ上の層のニューロンの誤差
→すでに計算済み



ニューラルネットワークの基本
Back Propagation (出力層-1) 層

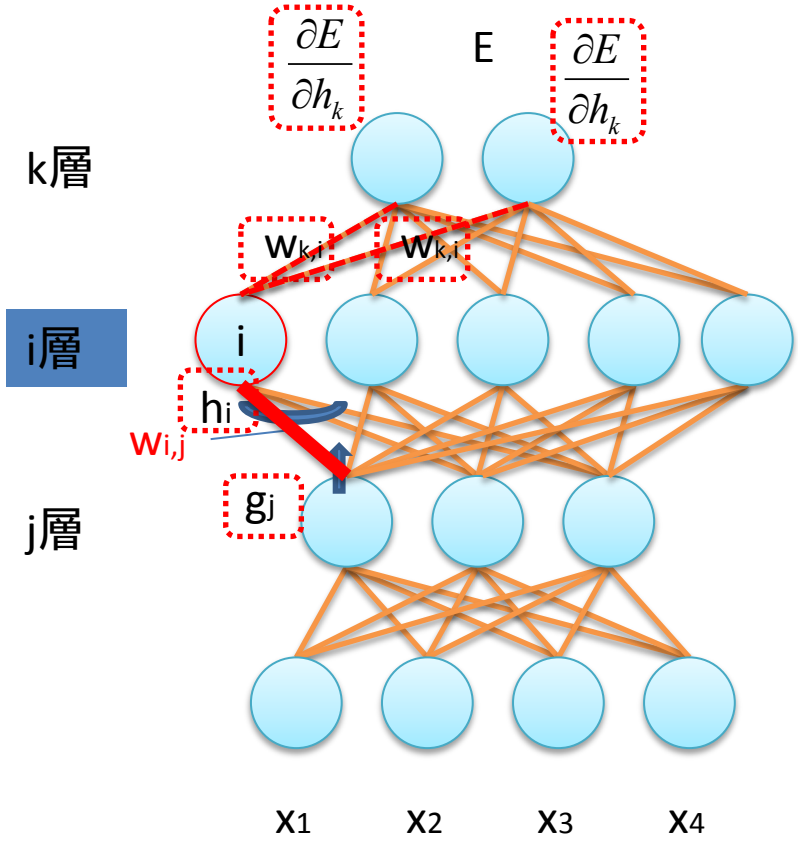
$\frac{\partial E}{\partial h_i}$を求める

(2)iが出力層の一つ手前

$$\frac{\partial E}{\partial h_i}=\frac{\partial E}{\partial g_i}\frac{\partial g_i}{\partial h_i}=(\sum_k \frac{\partial \mathrm{E}}{\partial \mathrm{h_k}}\frac{\partial h_k}{\partial g_i})\frac{\partial g_i}{\partial h_i}$$

$$=(\sum_k \frac{\partial \mathrm{E}}{\partial \mathrm{h_k}} w_{k,i}) f'(h_i)$$

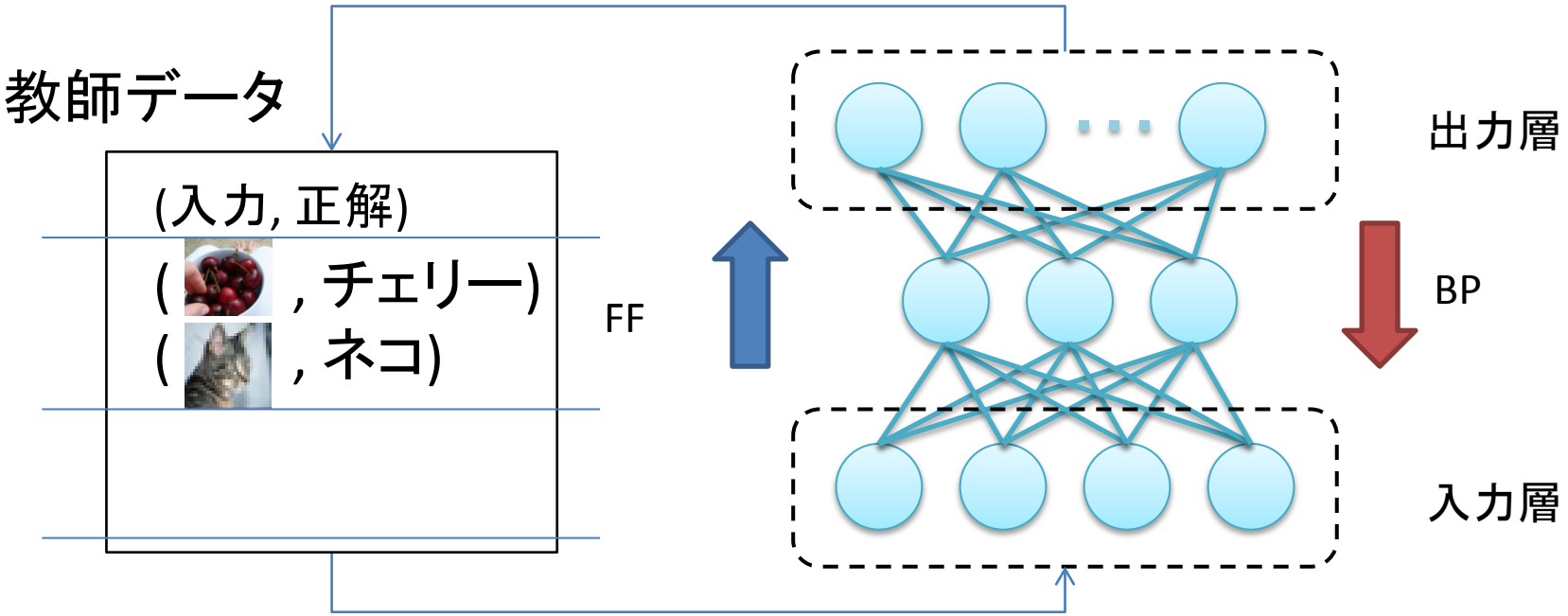
一つ上の層のニューロンの誤差
→すでに計算済み



•勾配計算のアルゴリズム

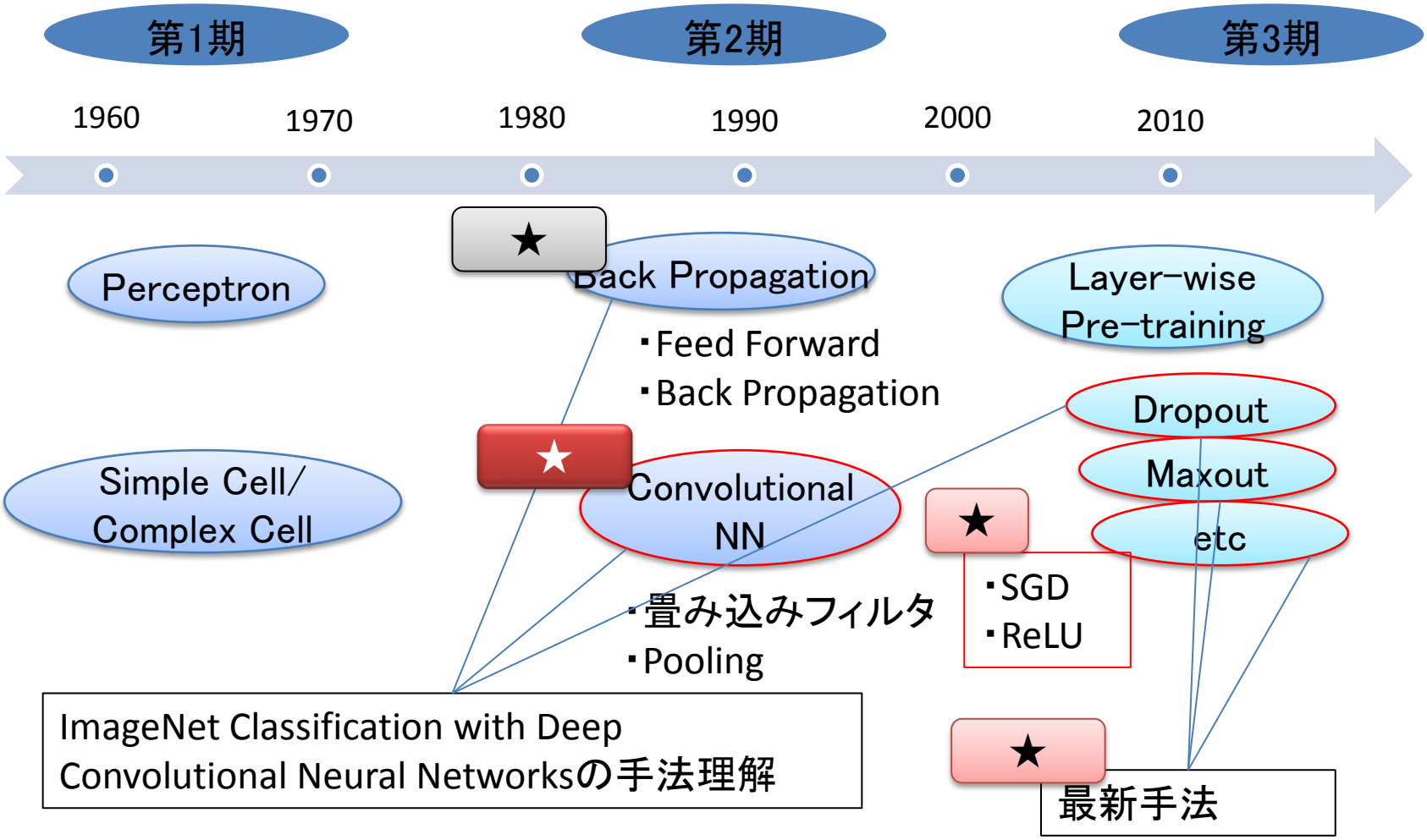
$$\frac{\partial E}{\partial w_{i,j}}=\frac{\partial h_i}{\partial w_{i,j}}\frac{\partial E}{\partial h_i}=g_j\frac{\partial E}{\partial h_i}$$

ニューラルネットワークの基本
Back Propagation補足



- 全教師データを同時に与えて重みを更新
- 教師データを1つ1つ与えて重みを更新
- 教師データをいくつかに分割し、その単位で処理
  - 確率的勾配法(SGD)、平均

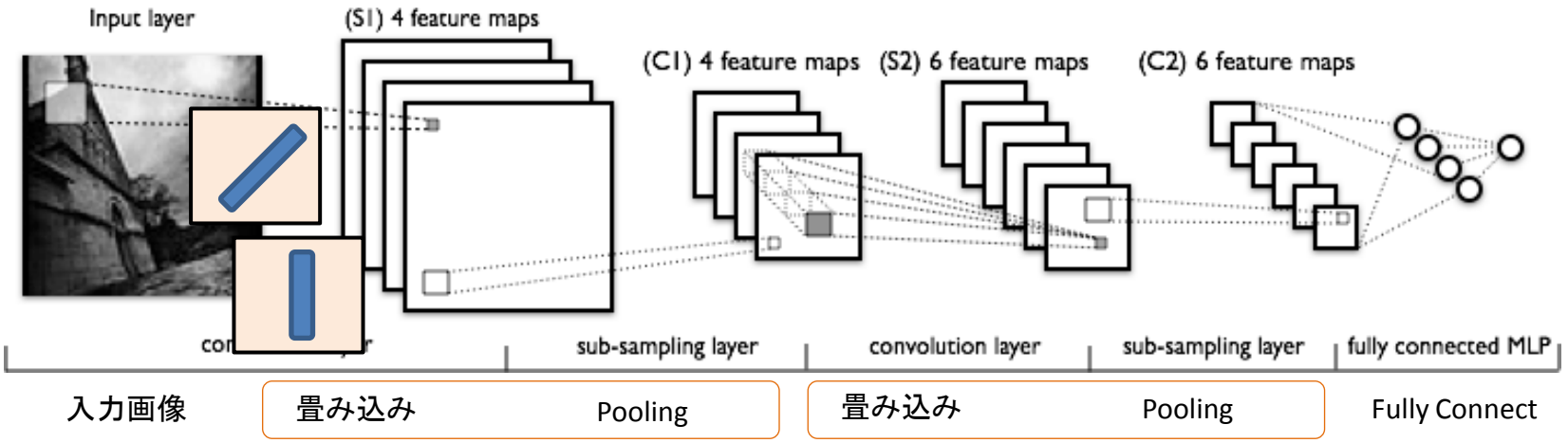
本日の話題


第1期
第2期
第3期
1960
1970
1980
1990
2000
2010
Perceptron
Back Propagation
・Feed Forward
・Back Propagation
Layer-wise Pre-training
Dropout
Maxout
etc
Simple Cell/ Complex Cell
Convolutional NN
・畳み込みフィルタ
・Pooling
・SGD
・ReLU
ImageNet Classification with Deep Convolutional Neural Networksの手法理解
最新手法

畳み込みニューラルネット(CNN)概要

- 視覚野は階層構造
- Simple Cell
  - 局所的
  - 単純(棒状・角度)
- Complex Cell
  - 複数Simple Cellの出力を足し合わせ
- Complex Cellの出力をより高次な細胞に入力

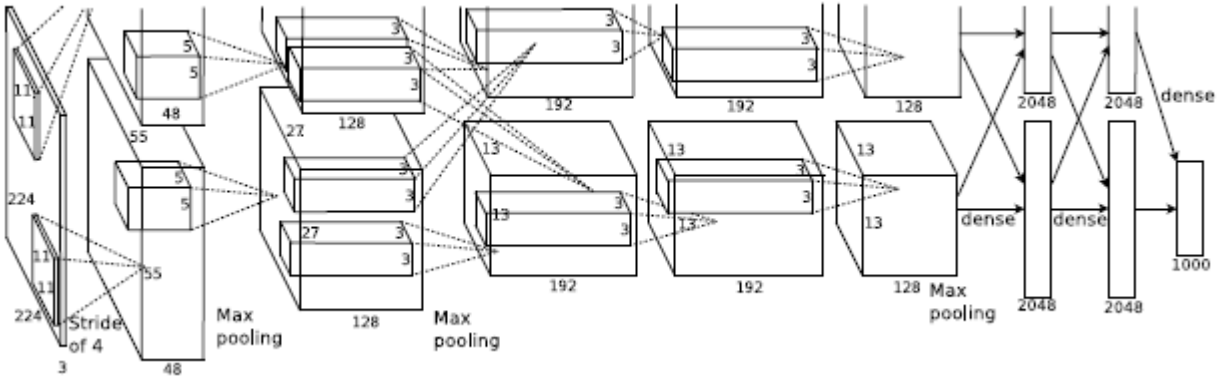
畳み込みニューラルネット(CNN) 全体像



- 畳み込み＋プーリング層を何層か用意
  - 畳み込み ≒ Simple Cell
  - Pooling ≒ Complex Cell
- Fully-Connectな層を挟み、出力層につなげる

http://deeplearning.net/tutorial/lenet.html#lenet

ImageNetでの例

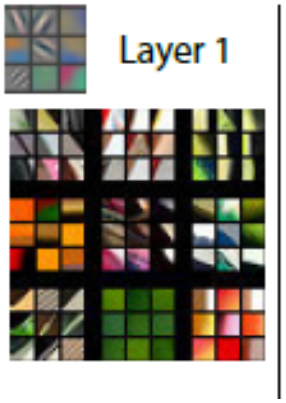


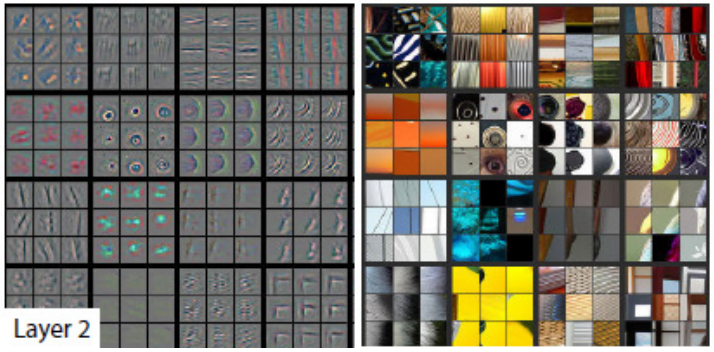
入力画像 CNN+Pooling CNN+Pooling CNN CNN CNN+Pooling FC FC Output

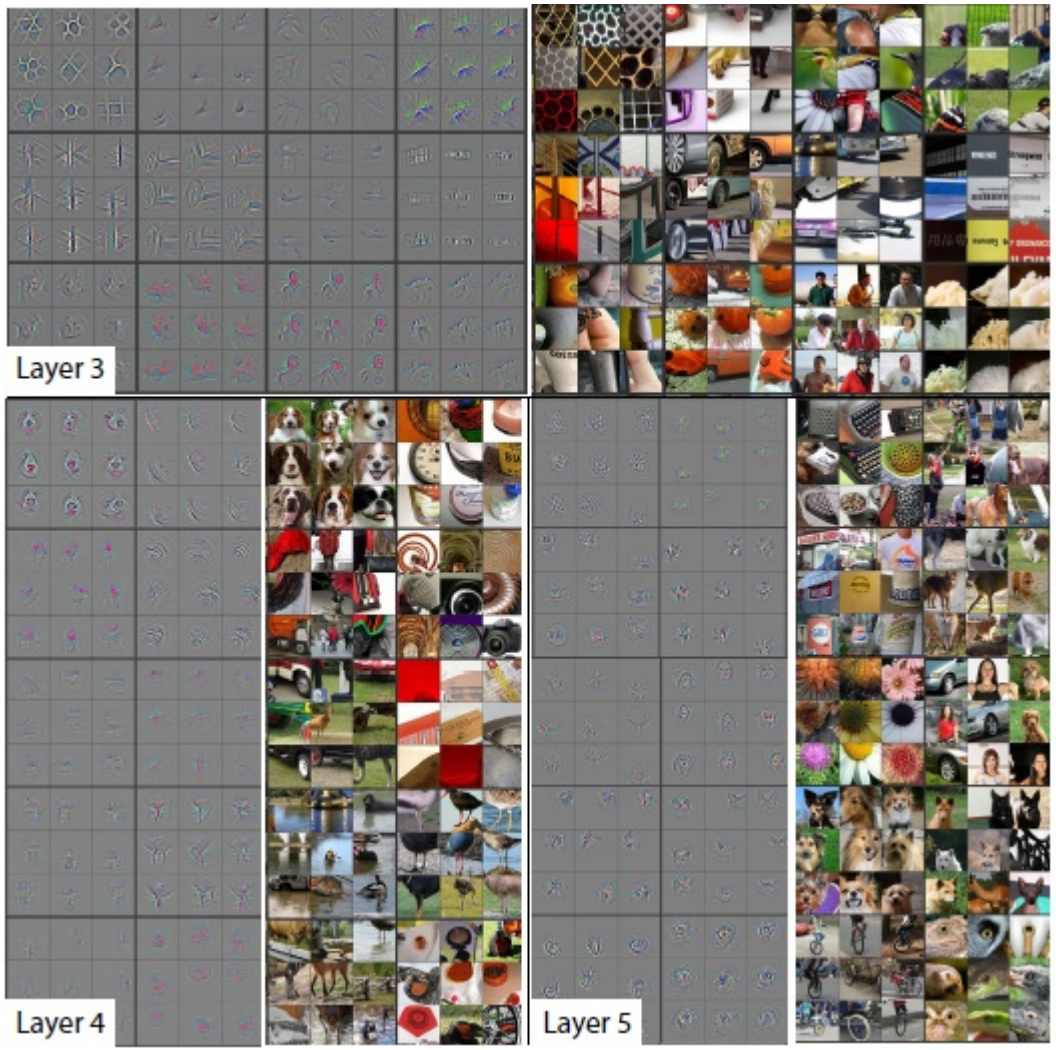
- 畳み込み層5層、FC 2層、出力層
- Feature Mapを直方体の厚みとして表現
  - 上下に分かれているのはMulti GPUの為

ImageNet Classification with Deep Convolutional Neural Networks

各層で反応する形状





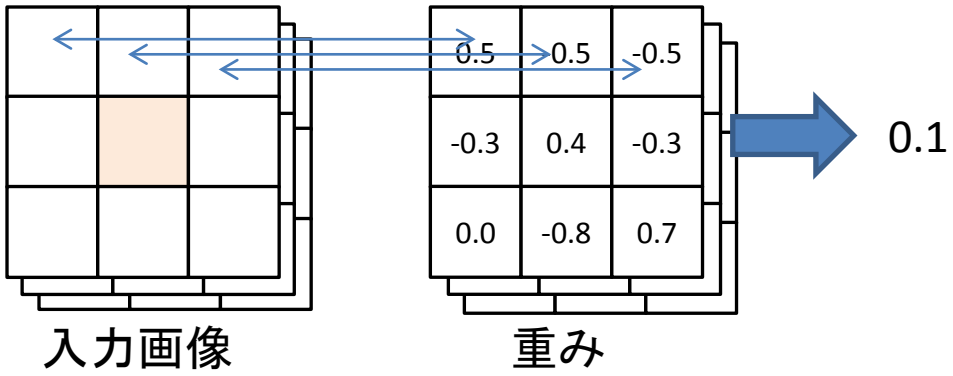


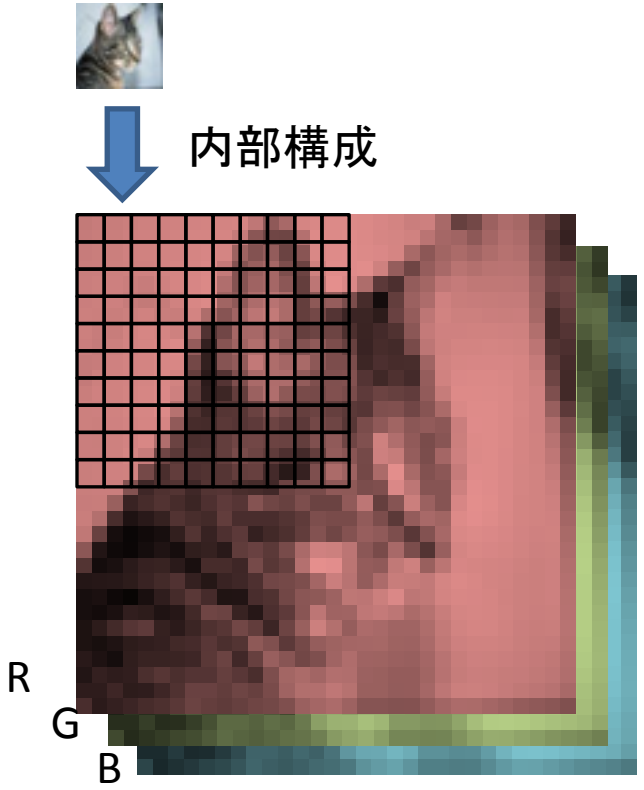
畳み込みニューラルネット(CNN)
畳み込みフィルタ

- 畳み込みフィルタ

$$f * g(i,j) = \sum_{n,m} f(i-m, j-n) g(n,m)$$

  - 画像フィルタと同じ処理:
    入力画像と重みの同じ位置をかけ、合計を取る
  - 近傍画素から値を計算
- 3x3フィルタ(カーネル)のイメージ

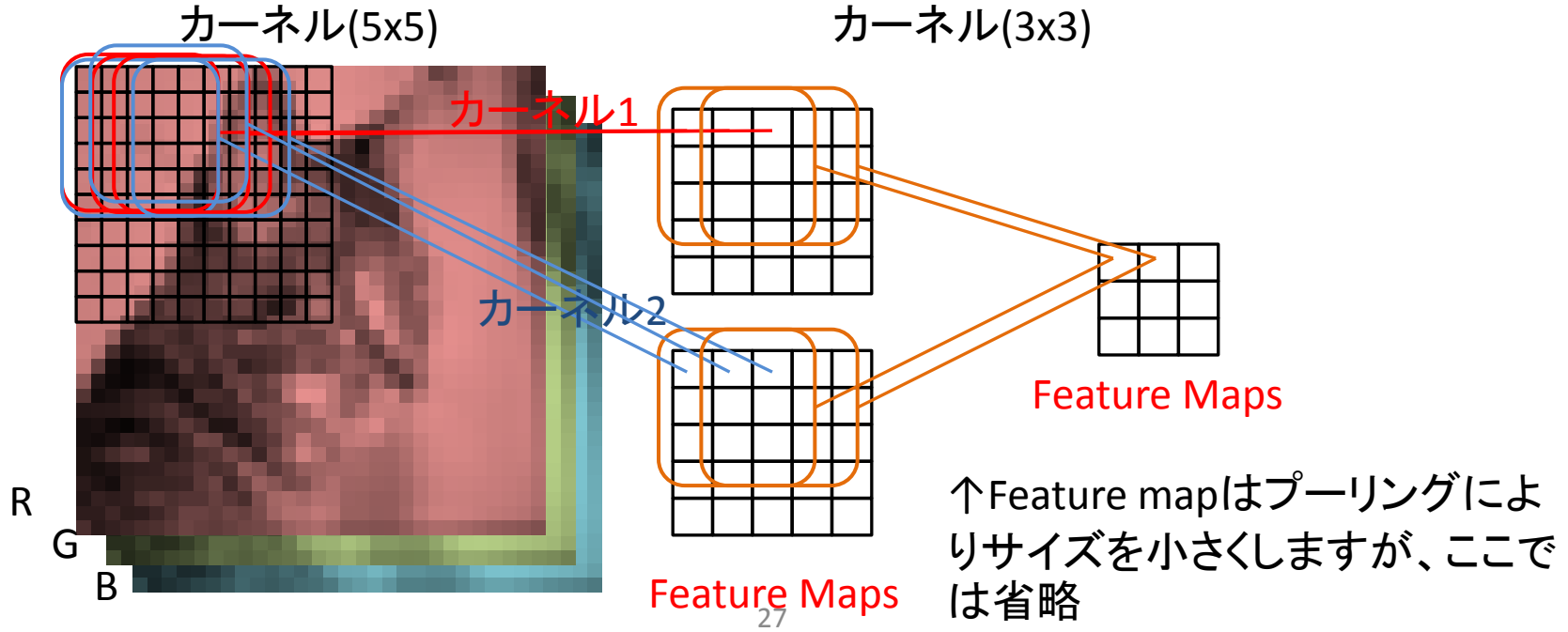




※RGBの場合、画像は3チャネルになる。
四角というより、チャネルを"深さ"に持つ直方体のイメージ。

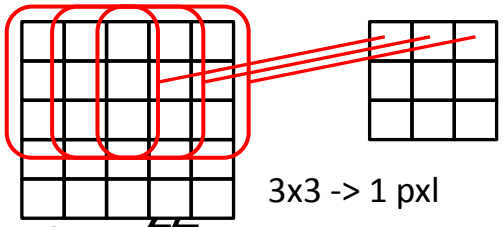
畳み込みニューラルネット(CNN)
畳み込みフィルタ

- カーネルをスライドさせながら画像に畳み込む。
  - オーバラップすることで微小変動に対応
- カーネルは一般には複数。Feature mapが複数得られるので、これを入力にさらに畳み込む。
- カーネルの数・サイズ・スライド量・参照するチャネル数等がハイパーパラメタになる。

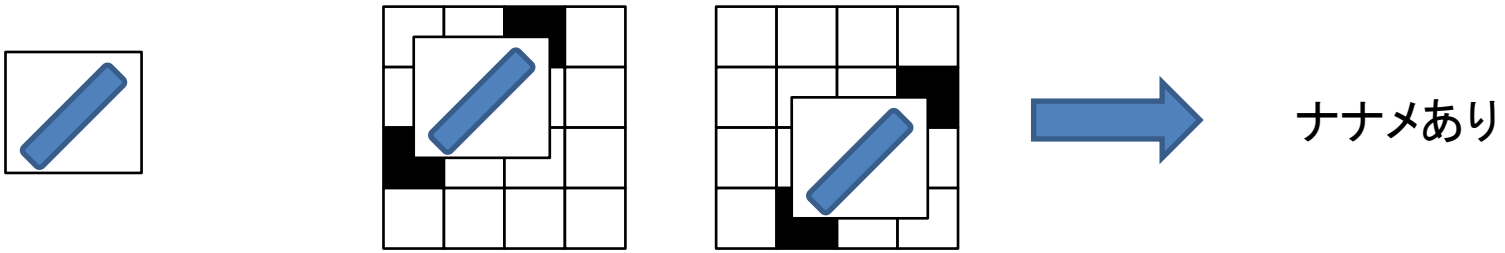


畳み込みニューラルネット(CNN)
プーリング

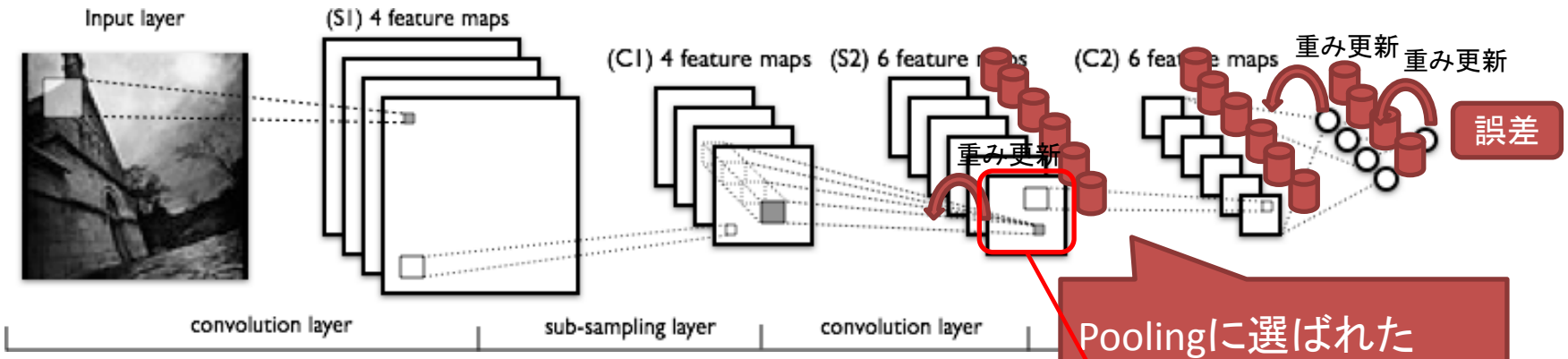
- プーリング
  - Feature Mapの小領域から代表値を算出
  - 解像度を落とす様なイメージ
  - 値の取り方でバリエーション
    - Max Pooling、Average Pooling、L2 Pooling等



- 畳み込みフィルタ＋プーリングの効果
  - 移動不変性

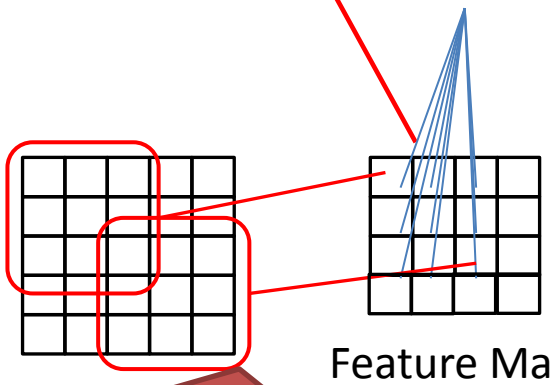


CNNでの学習概要



Poolingに選ばれたNeuronに誤差を伝達

Neuron毎に誤差

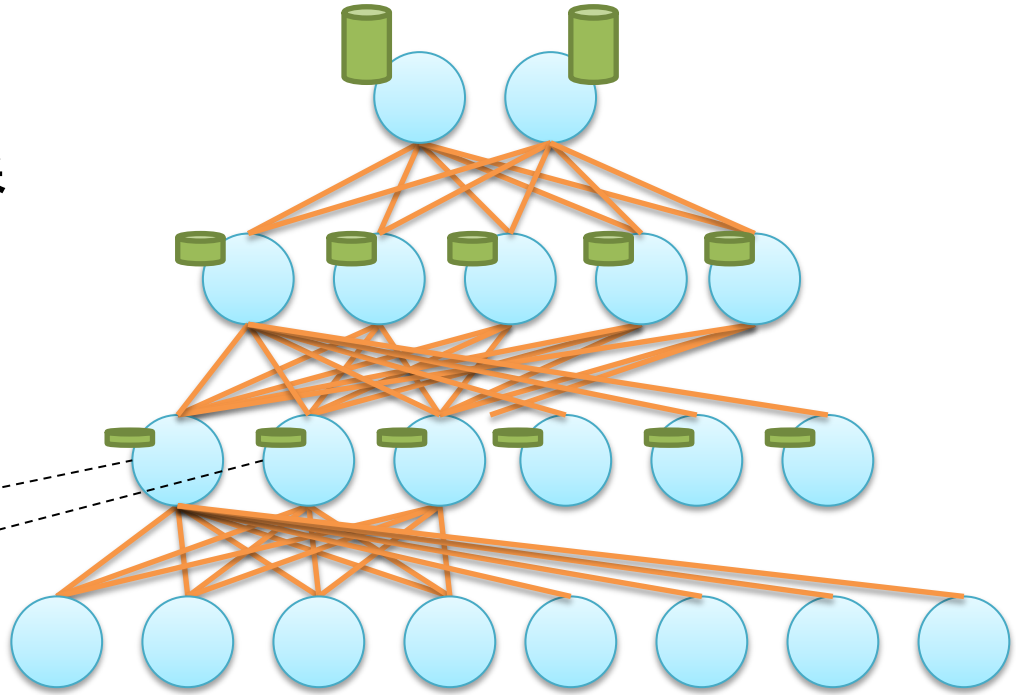


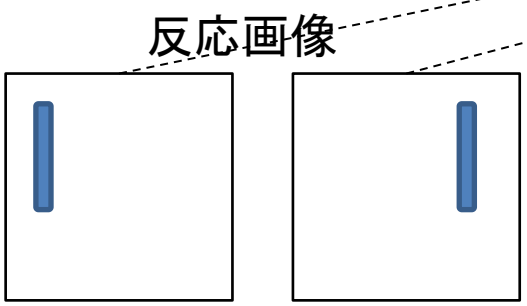
Feature Mapの値はそれぞれ同じカーネルを畳み込んだ結果
→個別に重みを学習して異なる値になってはダメ。共有重みにする必要あり。
→それぞれ重みを計算し、平均を取る

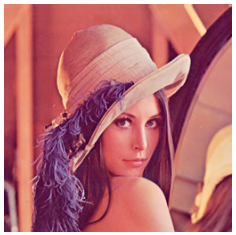
共有重みにする為、それぞれ重みを計算し平均

何故CNNは早くから多層にできたか

- 全結合
  - 誤差が広く浅く伝播
- 画像全体に反応
  →移動に弱い



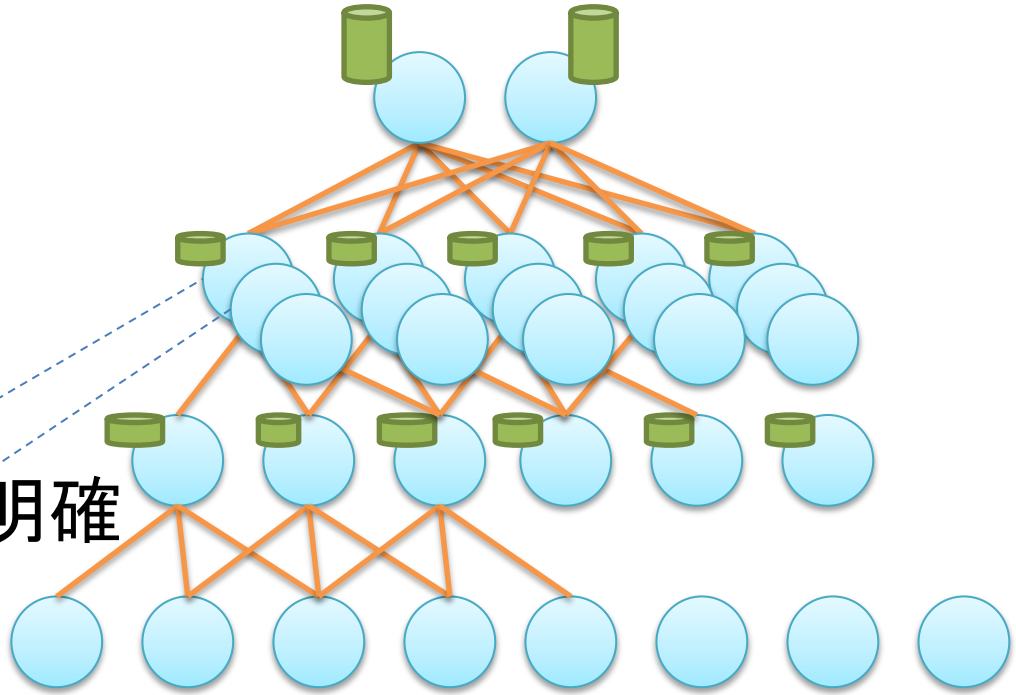




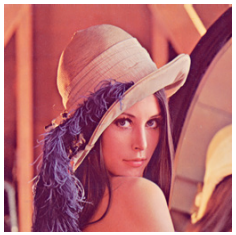
入力画像

何故CNNは早くから多層にできたか

- CNN
  - 誤差が局所的
  - Feature Map単位で重みを共有。
  - カーネルの役割が明確



反応画像

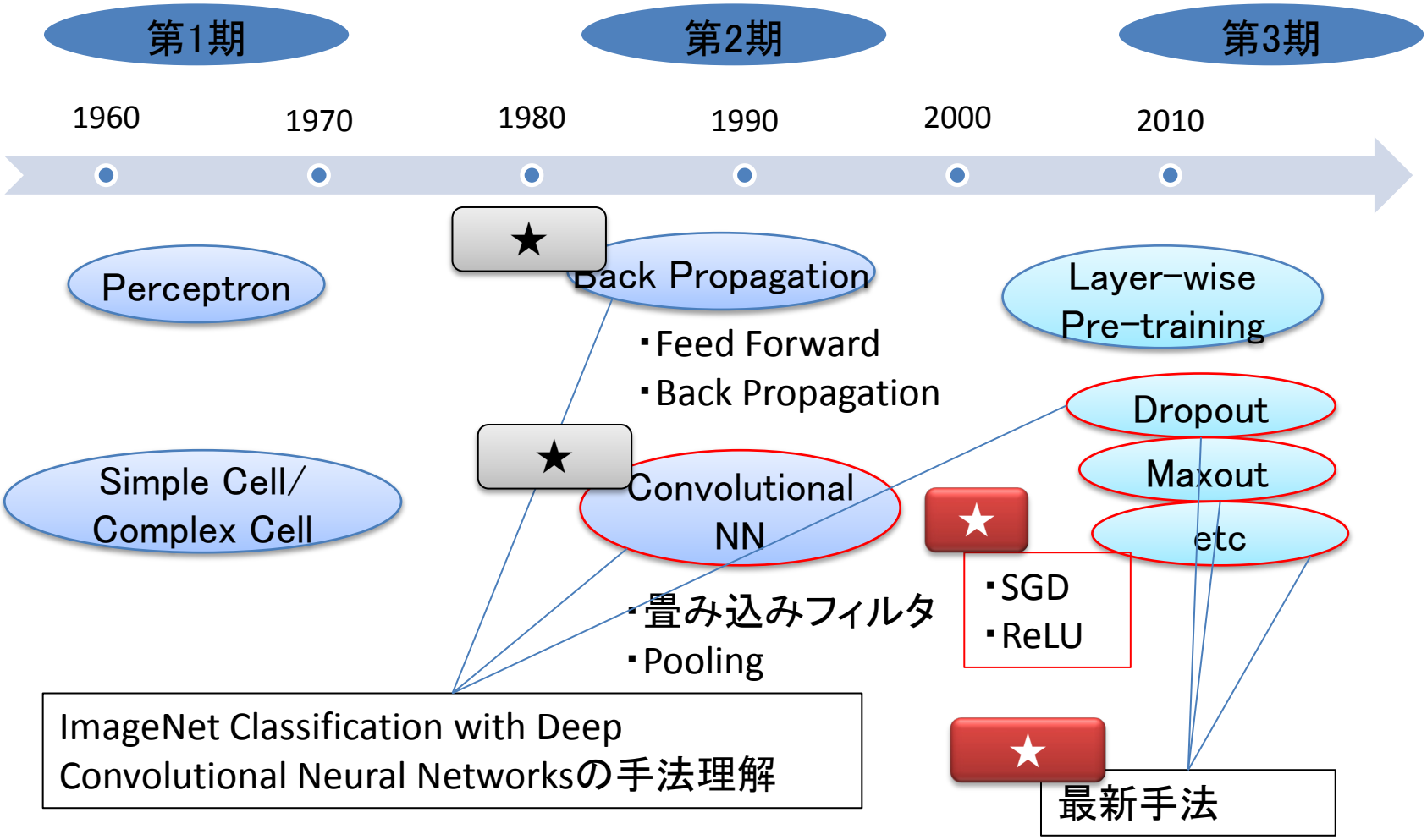


入力画像

畳み込みニューラルネット(CNN)
補足

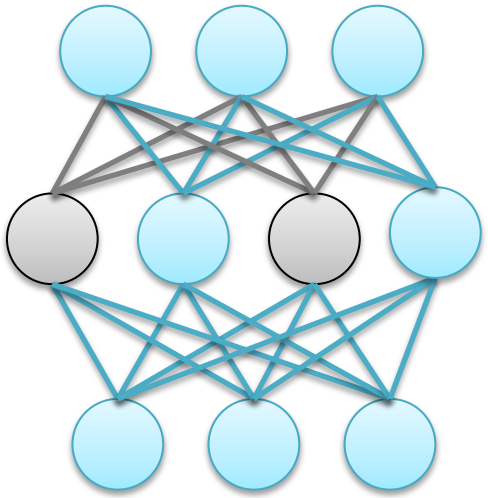
- カーネルサイズや深さは試行錯誤
- 共有重みの必然性
  - カーネルを複数用意するのは
    移動不変性の為。
  - 位置が大体決まっていれば共有重みに
    する必要無し？
  - 実際FacebookのDeep Faceでは前処理で位置を正規化。非共有重みカーネルでFeature mapを生成。

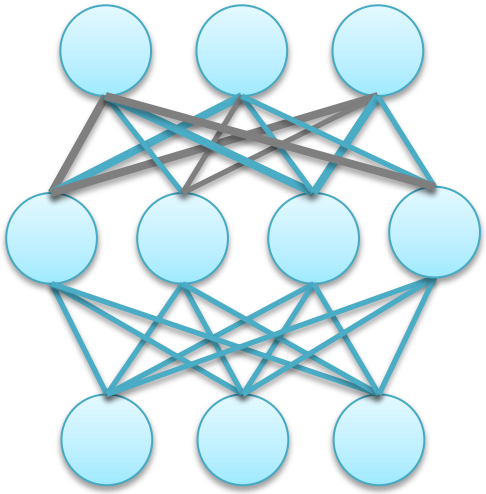
本日の話題


第1期
第2期
第3期
1960
1970
1980
1990
2000
2010
Perceptron
Back Propagation
・Feed Forward
・Back Propagation
Layer-wise
Pre-training
Dropout
Maxout
etc
Simple Cell/
Complex Cell
Convolutional
NN
・畳み込みフィルタ
・Pooling
・SGD
・ReLU
ImageNet Classification with Deep
Convolutional Neural Networksの手法理解
最新手法

過学習を防ぐための仕組み (Dropout, Dropconnect)

- Fully-Connected層の自由度を下げることで過学習を防ぐ
  - (Dropout)FC層のニューロンの出力をランダムに非活性化(0)にする。
  - (Dropconnect)FC層の一部の結合を非活性化(0)にする





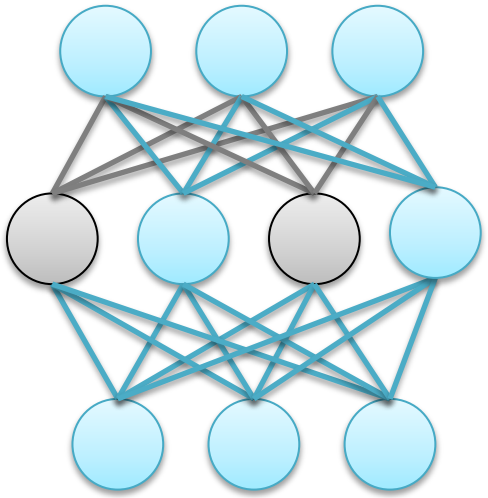
過学習を防ぐための仕組み (Dropout, Dropconnect)

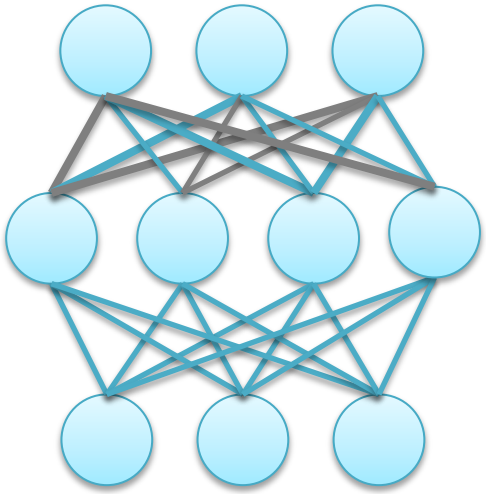
- 取り除く位置はMini-batch毎にランダム
  - 50%程度

$$Wx+b \rightarrow (W*M)x+b$$

$$(w_{i,j})*(m_{i,j})=(w_{i,j}*m_{i,j}),$$

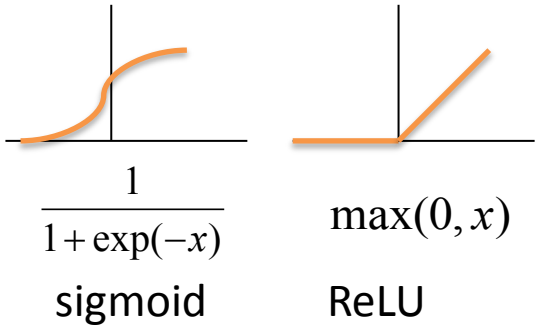
  - MはWと同じ形の0,1行列
- 学習終了後(実行時)はDropoutせず平均を取る。





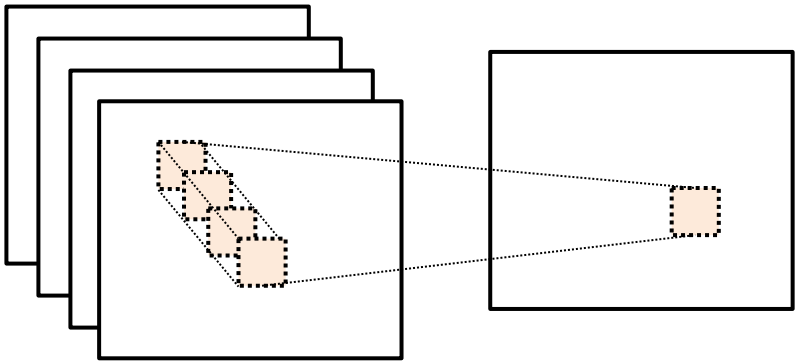
過学習を防ぐための仕組み（ReLU）

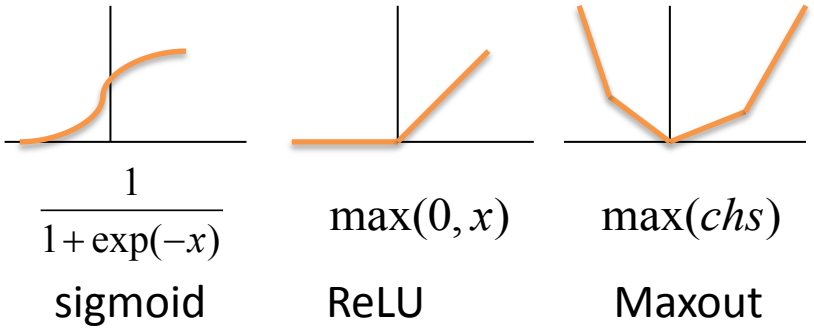
- 非線形関数にはsigmoidが一般的だったが、Saturationの問題
- Rectified Linear Unit
  - max(0, x)
  - 正の値なら勾配が1



過学習を防ぐための仕組み（Maxout）

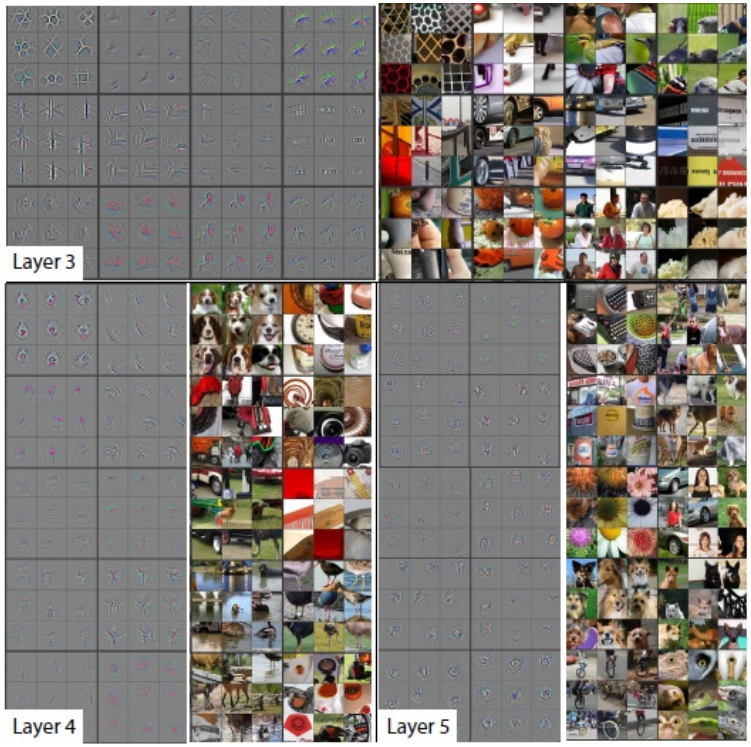
- 非線形関数をチャネル削減で代替
  - 個々のニューロンは恒等写像
  - 複数Feature Mapの同じ位置から最大値を採用
    - Poolingが縦横の解像度を下げる
    - Maxoutはchannel数を減らす

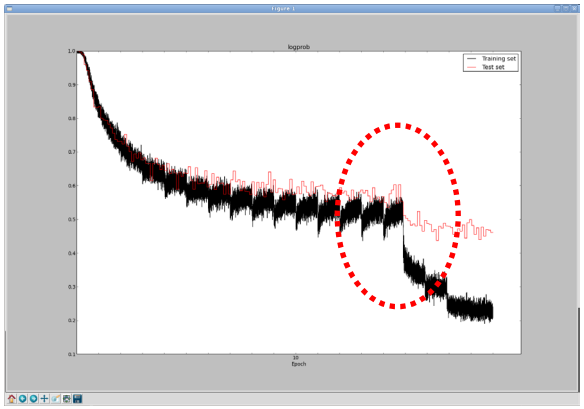




可視化の重要性

- 学習した係数の可視化
  - ニューロンが最も反応する画像・入力値
  - Deconvolution
  - gradient-based visualization
- 学習プロセスの可視化
  - コスト遷移と学習係数





その他の話題

- アルゴリズム的な進歩は著しい。
  - 一方、ImageNetの結果を使っての研究も盛ん。
- 学習データ数万～数百万の規模で必要。
  - 数千枚程度で利用する為の研究
  - Data Augmentation、Elastic Distortion
- 特徴抽出はDeep Learningで自動学習してくれるが、データの前処理による性能向上も。
  - 平均を0にする、分散を1にする、etc
  - 極端な値が悪影響

References

Krizhevsky et al., ImageNet Classification with Deep Convolutional Neural Networks, 2012

Zeiler et al., Visualizing and Understanding Convolutional Networks, 2013