ディープラーニングによる画像生成の最前線（基礎）

飯塚里志
早稲田大学

画像関連学会連合会 第3回秋季大会
2016/11/18

自己紹介

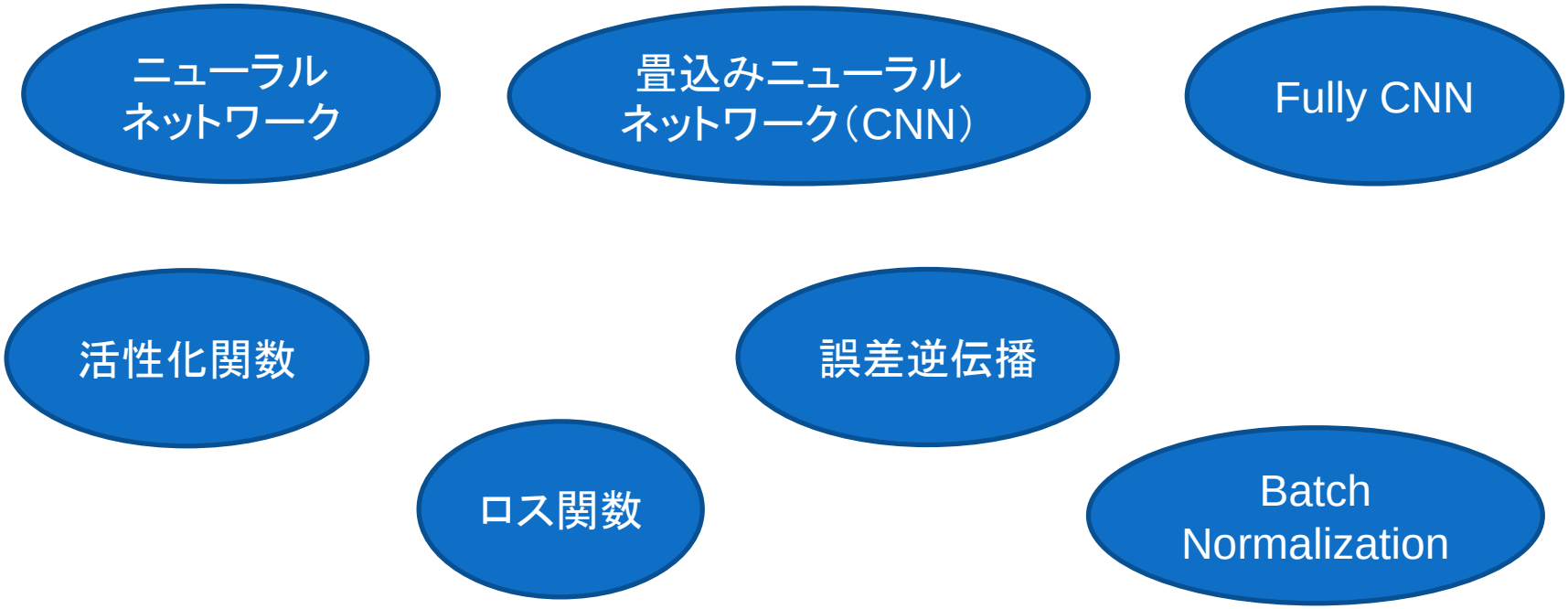
- 飯塚里志（IIZUKA Satoshi）
  - 2015年3月　　筑波大学で博士号取得
  - 2015年4月～　早稲田大学 研究院助教
- 研究分野
  - コンピュータグラフィクス、画像生成など

経年変化の再現
（Eurographics 2016）

白黒写真の自動色付け
（SIGGRAPH 2016）

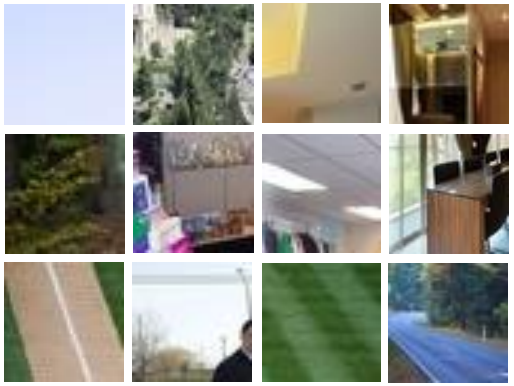
お話しする内容

- ディープラーニングによる画像生成に重要な基礎知識
  - 広い意味での画像生成



機械学習（教師あり）

顔画像



顔でない画像

教師（訓練）データ

学習

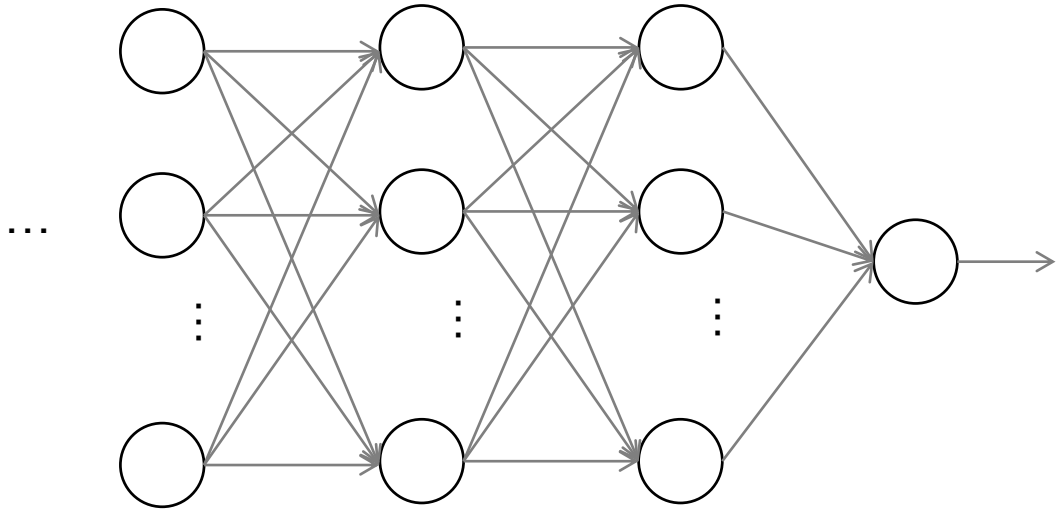
$\mathbf{x}$

$f(\mathbf{x})$

“顔” or “顔でない”

ディープラーニング

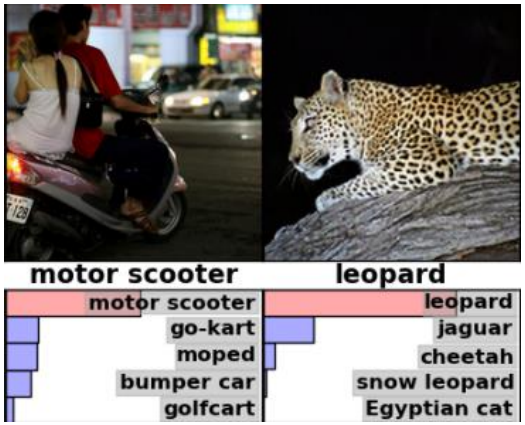
- 多層のニューラルネットワークを用いた機械学習
  - 脳の神経回路を模倣
  - 様々な分野でトップレベルの性能



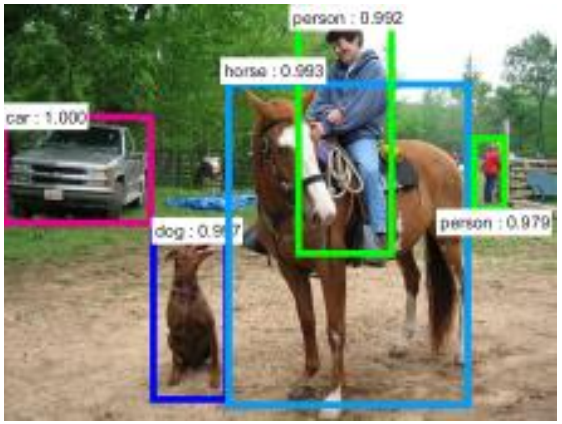
多層ニューラルネットワーク

ディープラーニングの応用例

- 画像分類
- 物体検出
- セグメンテーション
- 画像処理・生成



画像分類



物体検出

セグメンテーション

今どんなことができている？

超解像

白黒写真の自動色付け

ラフスケッチの線画化

画風の転写

画像を生成

動画を生成

研究事例①：超解像

- 高解像度画像へ変換
- 低解像度から高解像度への変換をニューラルネットワークによって学習
  - ニューラルネットワークの構造は研究によって様々

元画像

バイキュービック

提案手法

[Dong+ ECCV ’14]

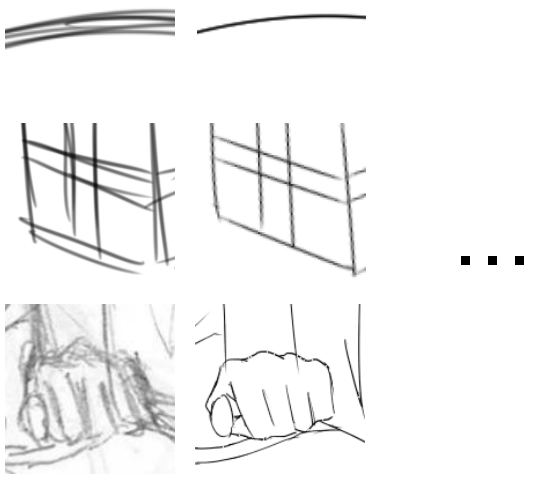
研究事例②：白黒写真の自動色付け

- 白黒写真をカラー写真に自動で変換
- 大量の写真データからディープラーニングによって写真の色を学習
- 自然な色付けを高精度で実現

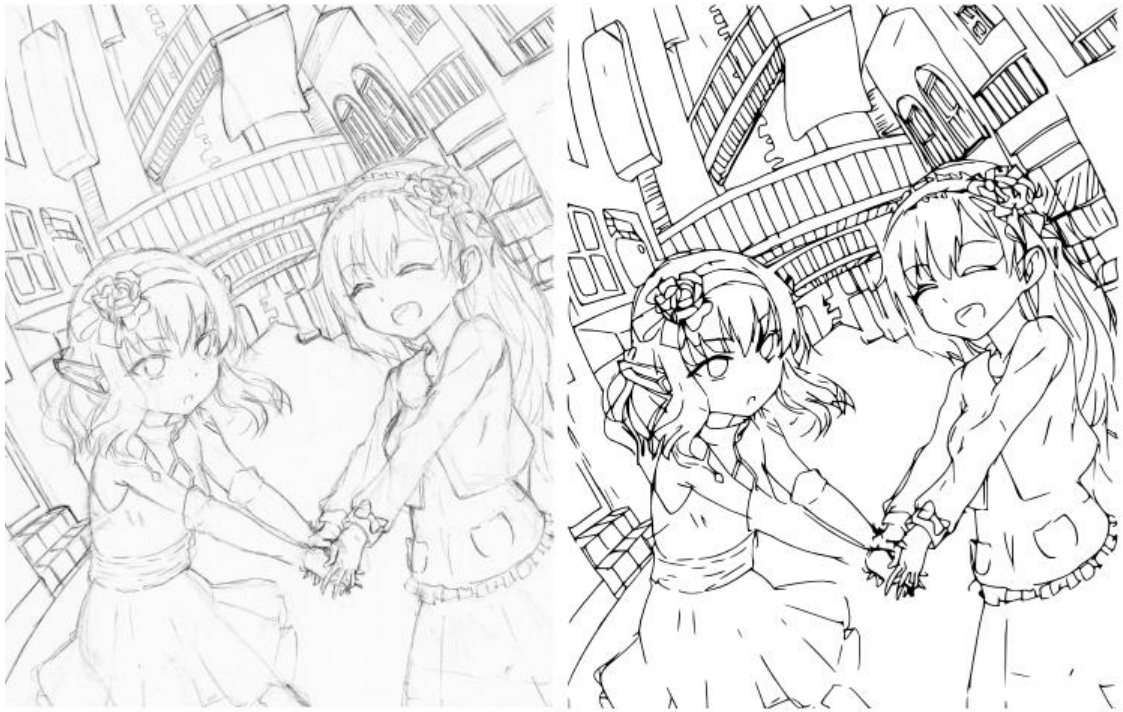
[Iizuka and Simo-Serra+ SIGGRAPH ’16]

研究事例③:ラフスケッチの自動線画化

- ラフスケッチを自動できれいな線画に
- ラフと線画の様々な対応関係をディープラーニングによって学習



対応関係の学習



[Simo-Serra and Iizuka+ SIGGRAPH '16]

研究事例④：画風転写

- 別の画像の「スタイル」を転写
- 学習済みのCNNを利用して入力画像を更新
  - 新しく学習はしない

コンテンツ画像

スタイル画像

出力画像

[Gatys+ CVPR ’16]

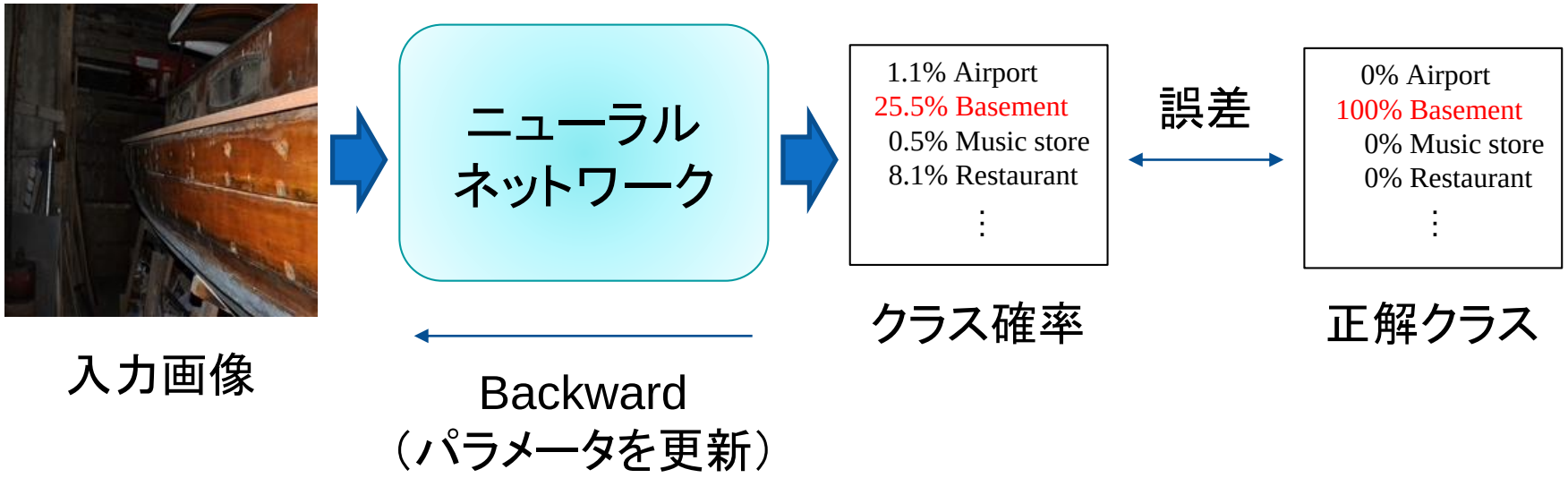
ディープラーニングは何が難しい？

- どんなモデル構造にするか
- どうやって学習させるか
- データセットをどうするか

ハイパーパラメータが多く、経験と試行錯誤が必要

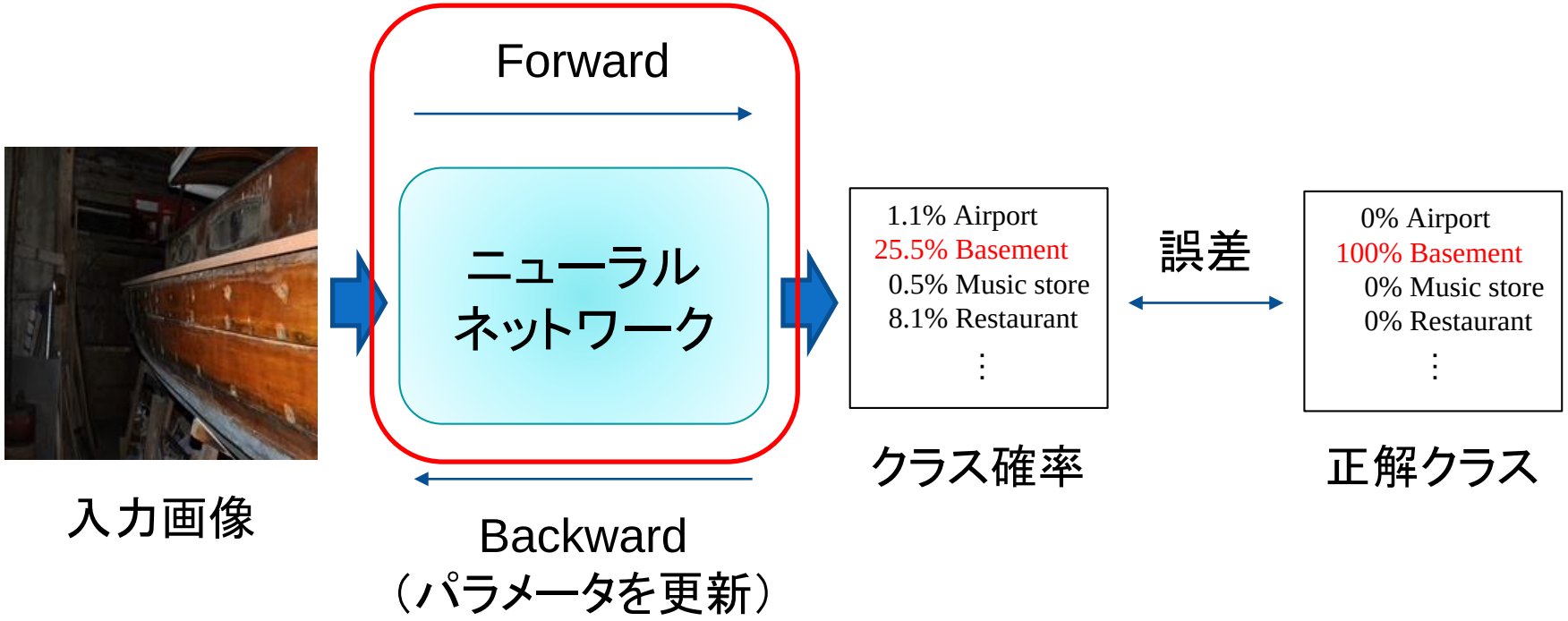
ディープラーニングの流れ

- 多クラス分類の場合

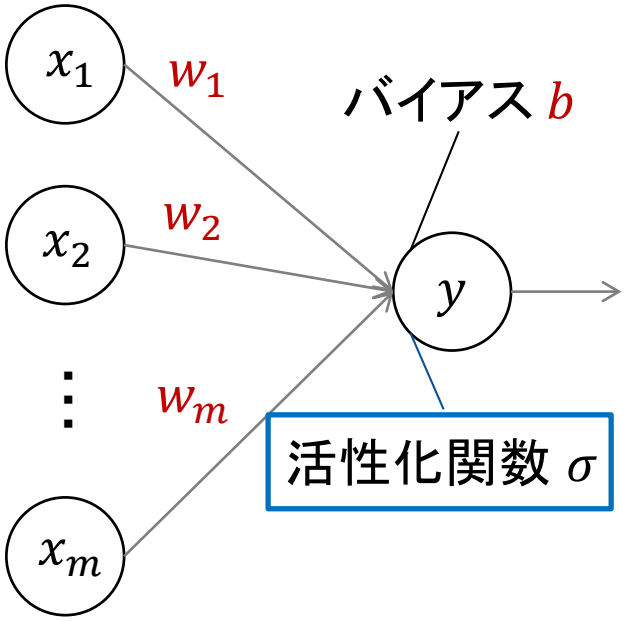


ディープラーニングの流れ

- 多クラス分類の場合



ニューラルネットワーク

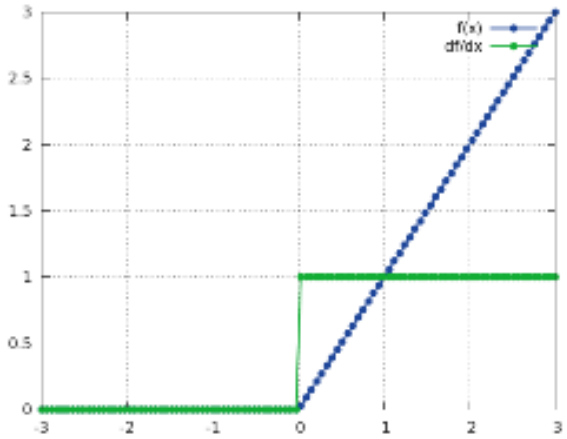


$$y = \sigma(w_1 x_1 + w_2 x_2 + \cdots + w_m x_m + b)$$

$$= \sigma\left(\sum_{j=1}^{m} w_j x_j + b\right)$$

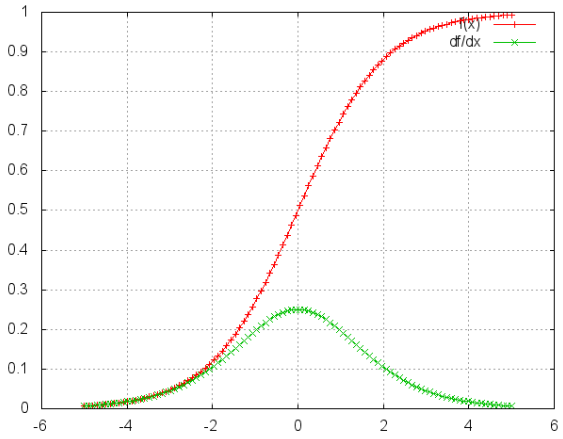
活性化関数

- ReLUとSigmoid（またはtanh）が画像生成では主流
  - SigmoidとtanhでOutputを[0,1]または[-1,1]に



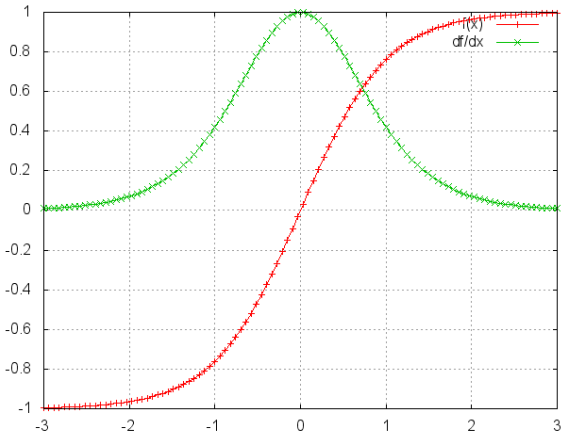
$$f(x) = \max(0, x)$$

ReLU



$$f(x) = \frac{1}{1 + e^{-x}}$$

Sigmoid

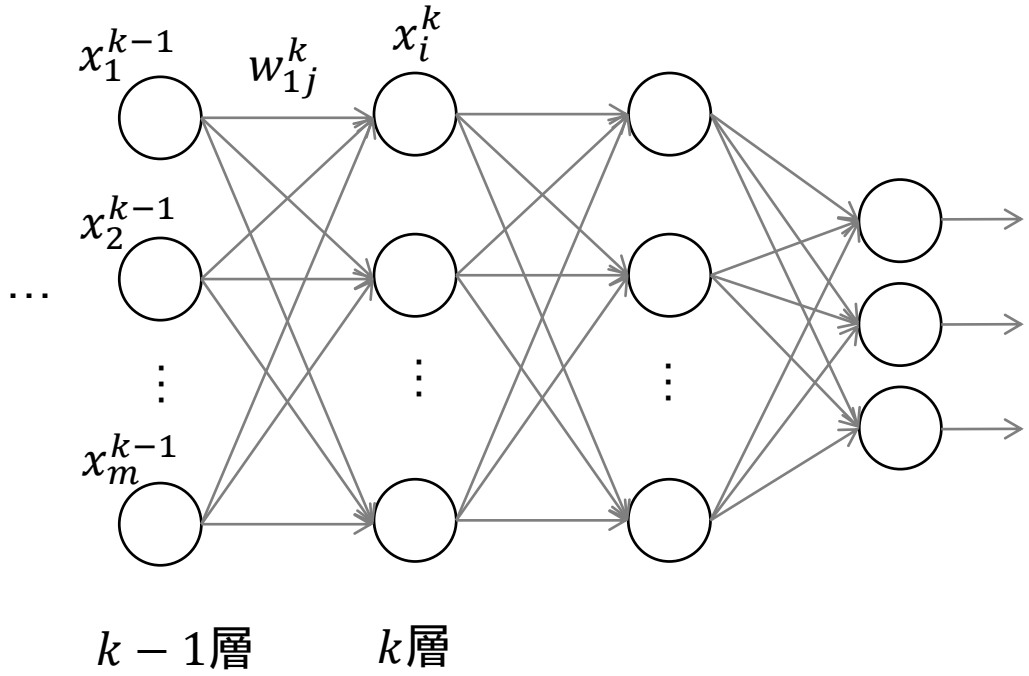


$$f(x) = \tanh(x)$$

tanh

https://github.com/torch/nn/blob/master/doc/transfer.md#nn.transfer.dok

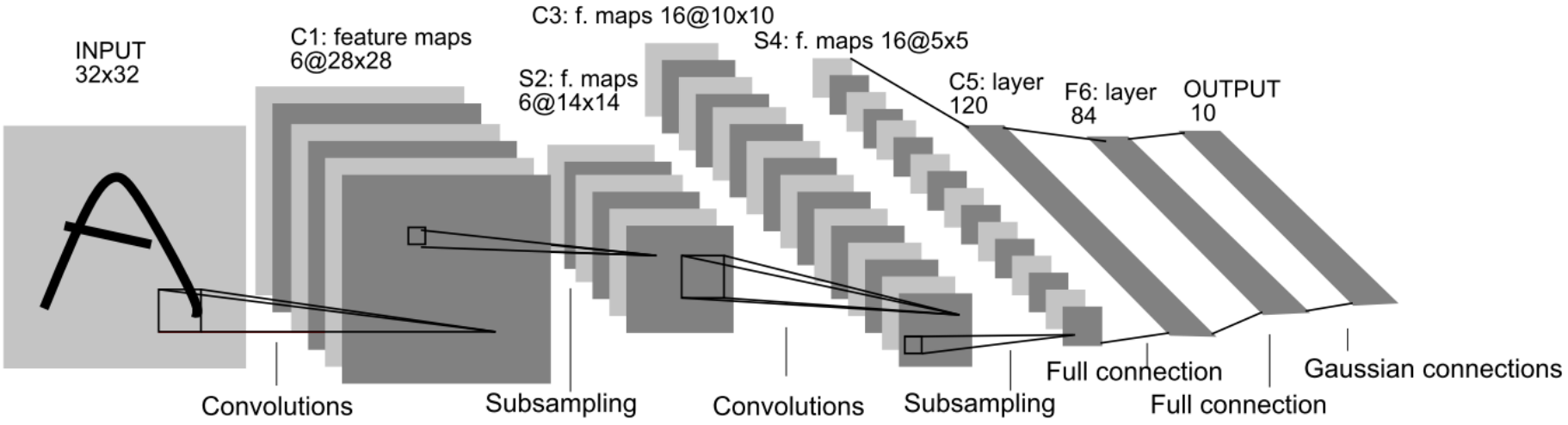
多層ニューラルネットワーク



$$x_i^k = \sigma\left(\sum_{j=1}^{m} w_{ij}^k x_j^{k-1} + b_i^k\right)$$

畳込みニューラルネットワーク（CNN）

- 畳込み層を取り入れたニューラルネットワーク
  - 神経科学の知見が基礎



LeNet-5 [LeCun+ ’98]

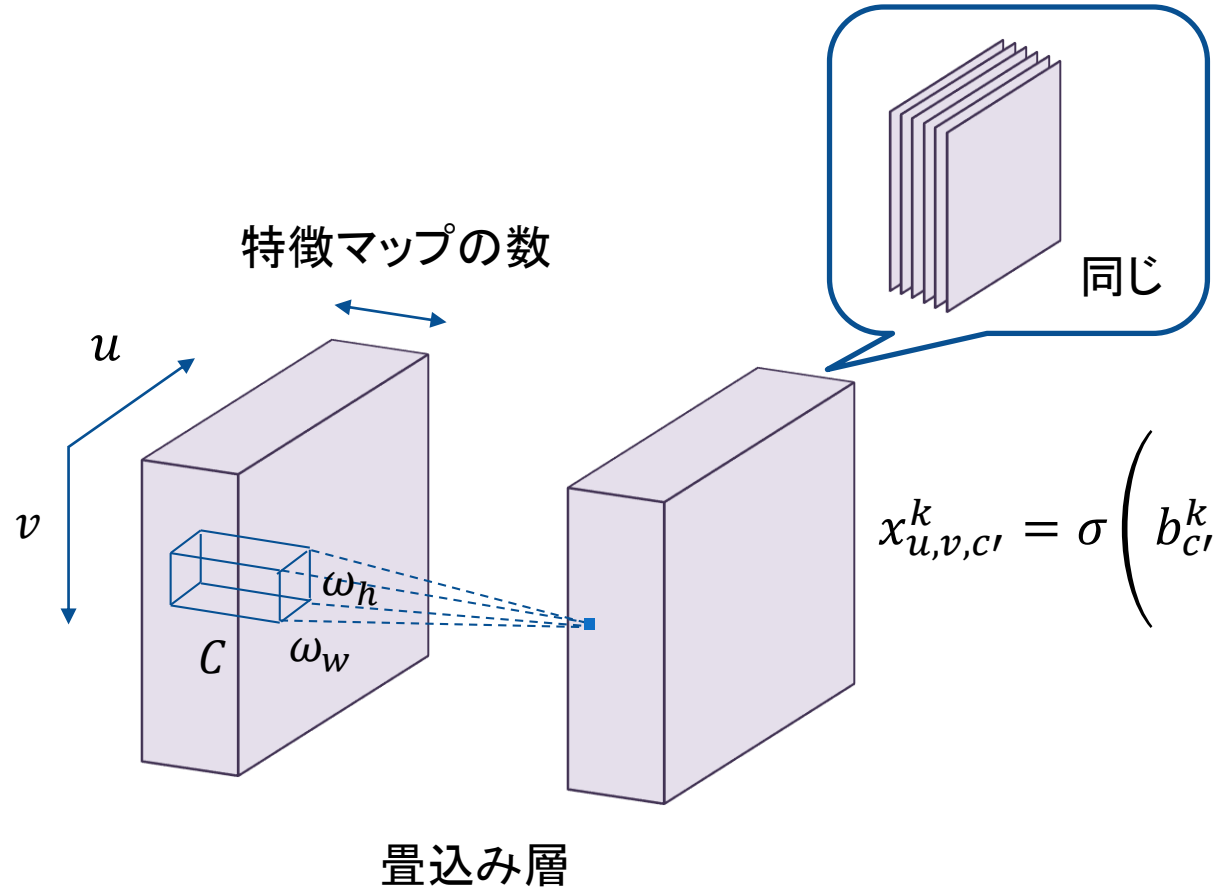
一般的なCNNの構造

- 入力サイズは固定（全結合層のため）

多クラス分類のネットワークの例

畳込み層

- 畳込み演算を行う層



$$x^k_{u,v,c'} = \sigma\left(b^k_{c'} + \sum_{c=1}^{C} \sum_{p=1}^{\omega_h} \sum_{q=1}^{\omega_w} x^{k-1}_{u+p,v+q,c} w^k_{p,q,c} \right)$$

畳込み演算

1	0	1
0	1	0
1	0	1

フィルタ
（この重みを学習する）

$1_{\times 1}$	$1_{\times 0}$	$1_{\times 1}$	0	0
$0_{\times 0}$	$1_{\times 1}$	$1_{\times 0}$	1	0
$0_{\times 1}$	$0_{\times 0}$	$1_{\times 1}$	1	1
0	0	1	1	0
0	1	1	0	0

入力画像

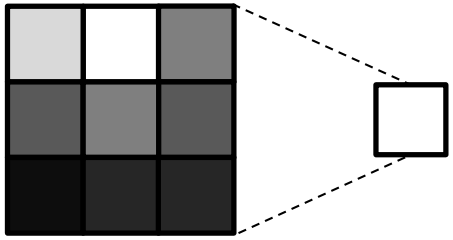
4		

出力画像

引用元：
http://deeplearning.stanford.edu/wiki/index.php/Feature_extraction_using_convolution

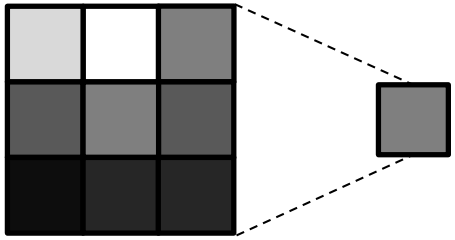
プーリング層

- 特徴マップのサイズを縮小
- シフト不変



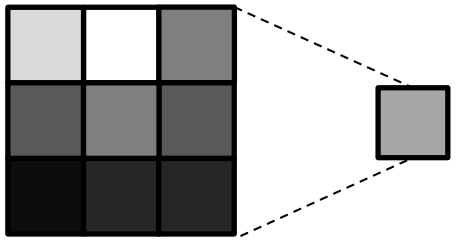
$$h_i = \max_{j \in P_i} h_j$$

マックスプーリング



$$h_i = \frac{1}{|P_i|} \sum_{j \in P_i} h_j$$

平均プーリング

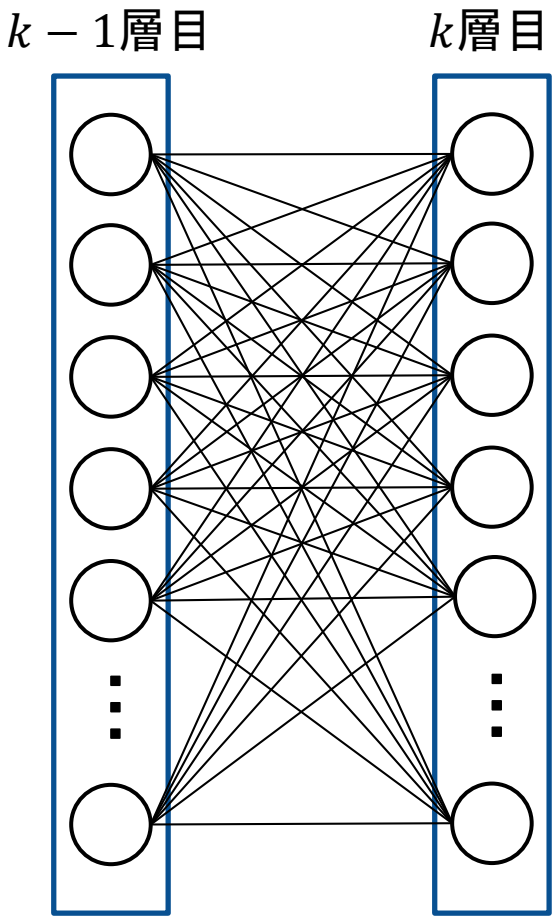


$$h_i = \left(\frac{1}{|P_i|} \sum_{j \in P_i} h_j^P \right)^{\frac{1}{P}}$$

Lpプーリング

全結合層

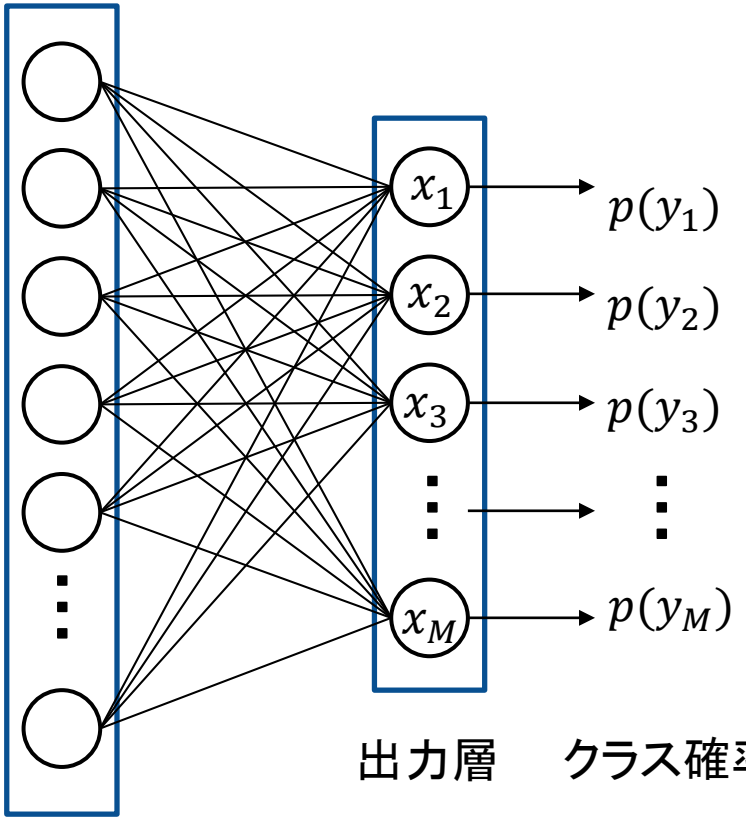
- すべてのノードが結合



$$x_i^k = \sigma\left(\sum_j w_{ij}^k x_j^{k-1} + b_i^k\right)$$

出力層

- 出力層のノード数＝目的のクラス数
- 分類にはSoftmax関数を利用



$$P(y_i) = \frac{e^{x_i}}{\sum_{j=1}^{M} e^{x_j}}$$

Softmax関数

ディープラーニングの流れ

- 多クラス分類の場合

入力画像

Forward

ニューラル
ネットワーク

1.1% Airport
25.5% Basement
0.5% Music store
8.1% Restaurant
⋮

クラス確率

誤差

0% Airport
100% Basement
0% Music store
0% Restaurant
⋮

正解クラス

Backward
（パラメータを更新）

ロス関数の例

- 平均二乗誤差（Mean Squared Error, MSE）

$$L = \frac{1}{|N|}\sum_{i \in N}(\hat{y}_i - y_i)^2$$

出力値

正解値

- クロスエントロピー誤差
  - 分類

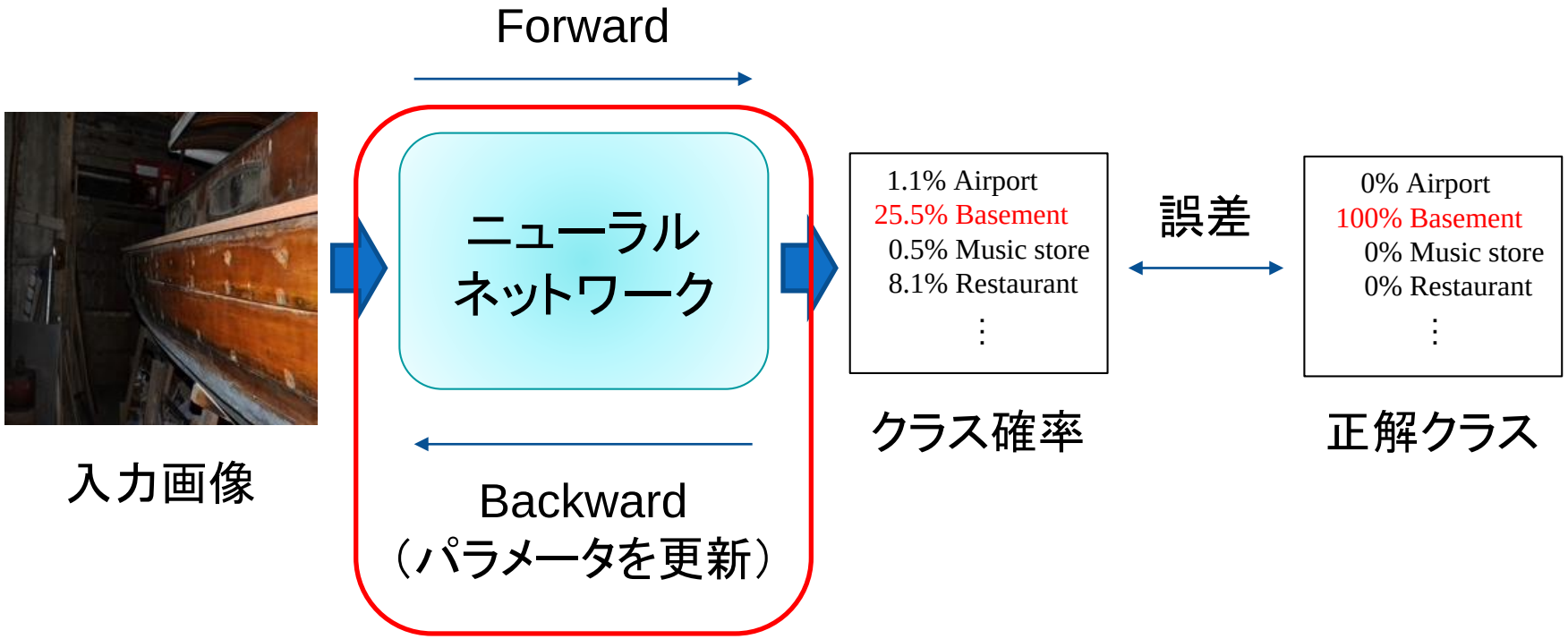
$$L = -\sum_{i} d_i \log p_i$$

出力されたクラス確率

正解のクラス確率

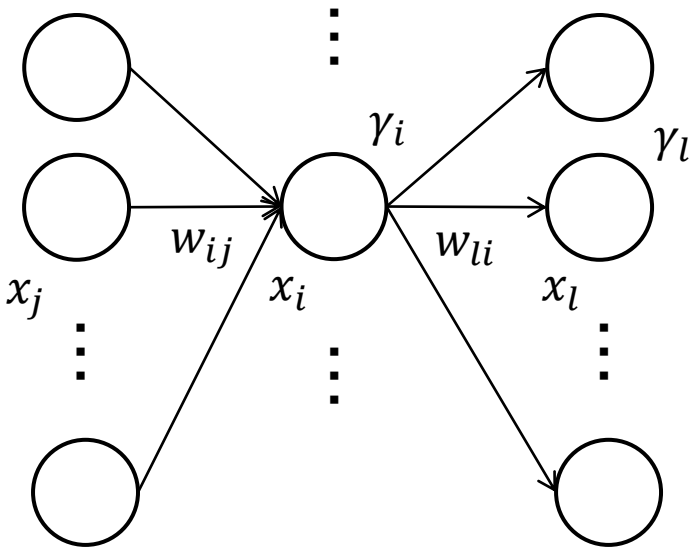
ディープラーニングの流れ

- 多クラス分類の場合



誤差逆伝播（Back Propagation）

- 重みを更新 $w_{ij} \leftarrow w_{ij} - \epsilon \dfrac{\partial L}{\partial w_{ij}}$
- 出力層から遡って計算



$$\frac{\partial L}{\partial w_{ij}} = \frac{\partial L}{\partial x_i}\frac{\partial x_j}{\partial w_{ij}}$$

$\sigma(x_j)$

$$\gamma_i = \sum_l \frac{\partial L}{\partial x_l}\frac{\partial x_l}{\partial x_i} = \sum_l \gamma_l \sigma'(x_i) w_{li}$$

$$= \sigma'(x_i) \sum_l \gamma_l w_{li}$$

$$\boxed{\frac{\partial L}{\partial w_{ij}} = \gamma_i \sigma(x_j)}$$

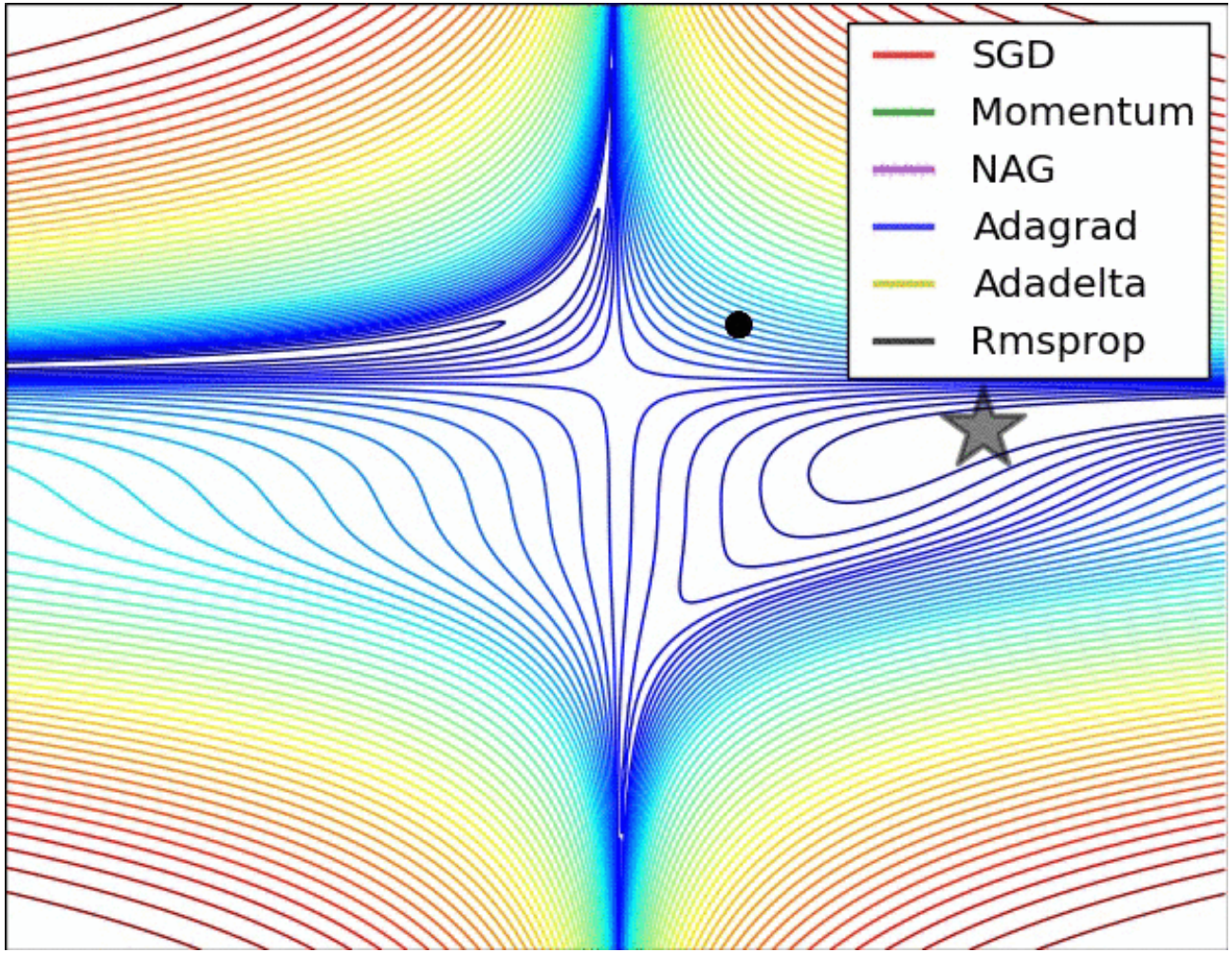
主な更新手法

- 確率的勾配降下法（SGD）
- Adagrad
  - 学習係数をパラメータによって変化
- RMSprop
  - Adagradの学習率の変化を改良
- Adam
  - Adagradにモメンタムとバイアス補正を追加
- Adadelta
  - 学習係数を自動で決定

$$w_{ij} \leftarrow w_{ij} - \epsilon \frac{\partial L}{\partial w_{ij}}$$

学習係数

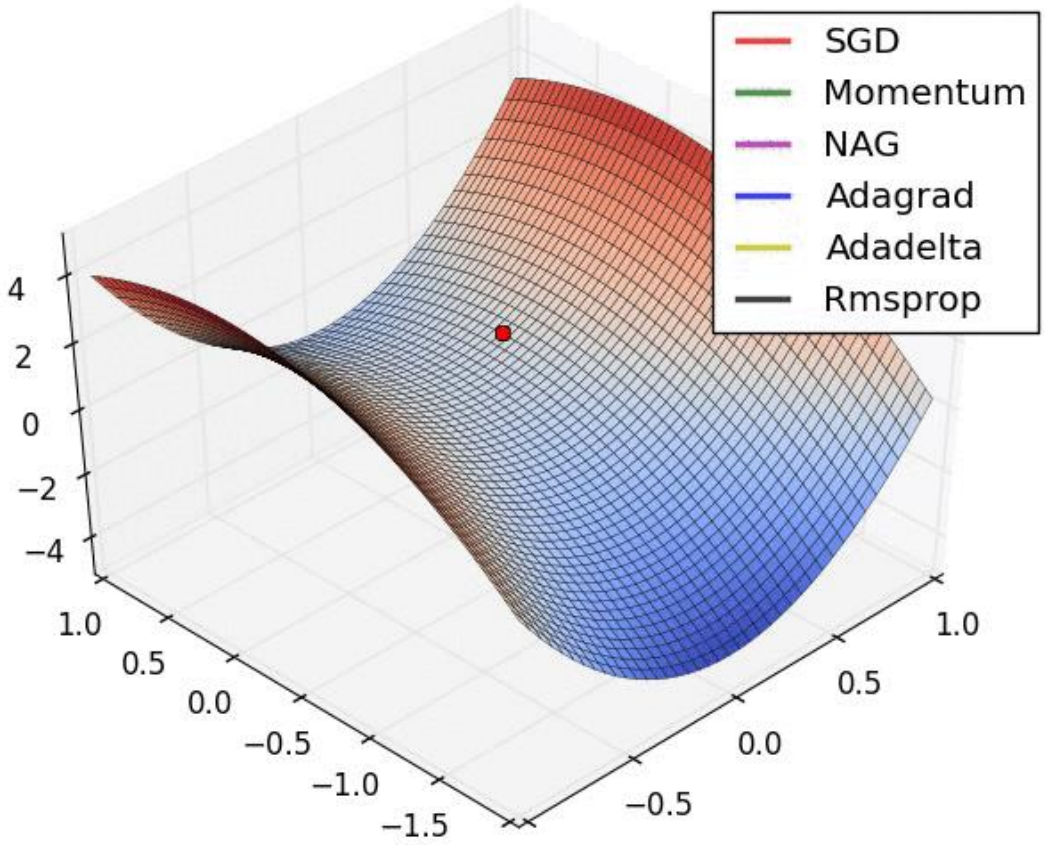
最適解への更新



Alec Radford's animation

鞍点での動作

- 鞍点：見る方向によって極小値にも極大値にもなる点



Alec Radford’s animation

結局どれがいい？

- 実はどれを使ってもあまり変わらない（場合が多い）
  - 適切な学習係数を使った場合の話
  - 個人的にはAdadeltaがおすすめ

SGD

超解像
[Dong+ ’14]

SGD+モメンタム＋荷重減衰

セマンティックセグメンテーション
[Nor+ ’15]

Adadelta

自動色付け
[Iizuka+ ’16]

Adam

画像生成
[Radford+ ’16]

汎化能力・学習効率の向上

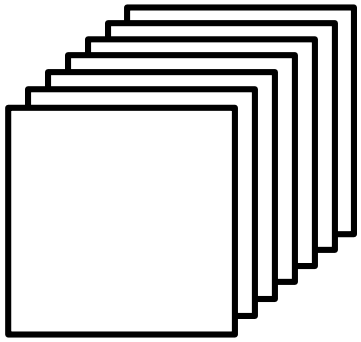
- Dropout
  - ニューロンの出力をランダムでゼロに
- Data Augmentation（データ拡張）
  - データを「無理やり」増やす
- Batch Normalization [Ioffe+ NIPS ’15]

入力画像　反転　切り抜き＋回転　切り抜き＋拡大

正解ラベルは
すべて同じ

Batch Normalization

- バッチごとに各層の特徴マップを正規化
  - 平均を0、分散を1
- 大規模なモデルの学習に非常に重要



バッチ

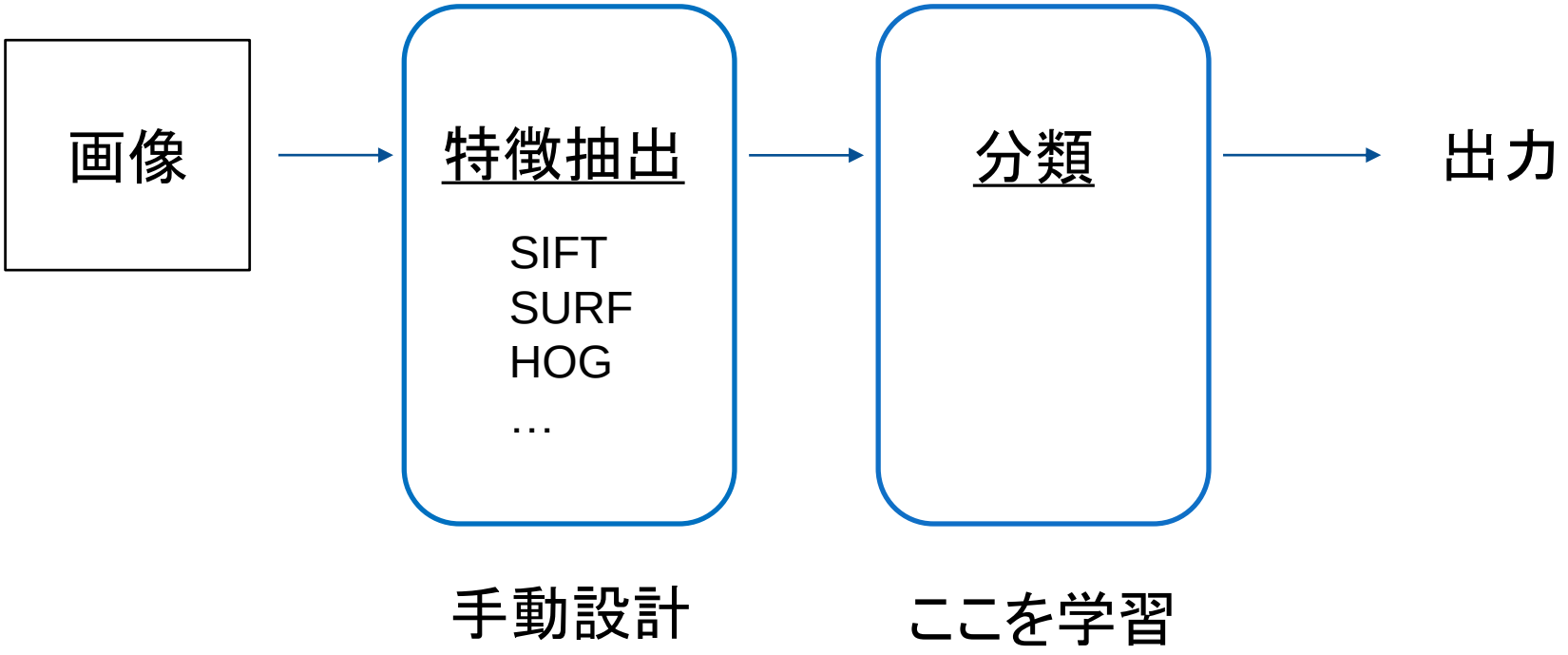
バッチの平均

$$y_i \leftarrow \gamma \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}} + \beta$$

バッチの分散

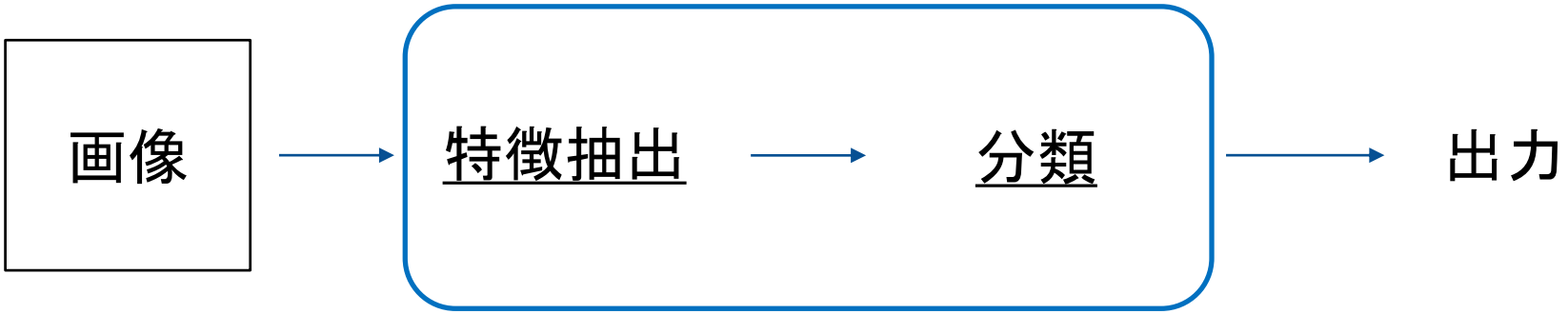
ディープラーニングのすごいところ

- 従来の機械学習



ディープラーニングのすごいところ

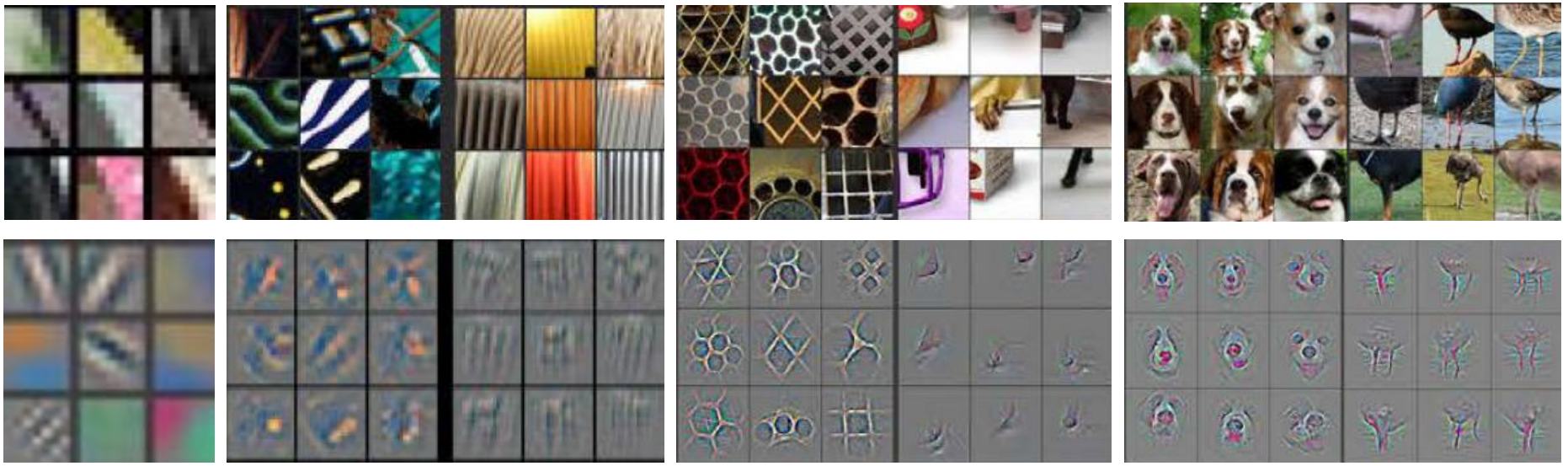
- ディープラーニング



すべて学習！

なぜ画像生成にCNN？

- 画像処理・生成には中・高レベルの特徴が重要
  - 手動設計するのが困難
  - CNNは複雑な特徴を抽出可能



1層目
コーナー、エッジ、色
（低レベル特徴）

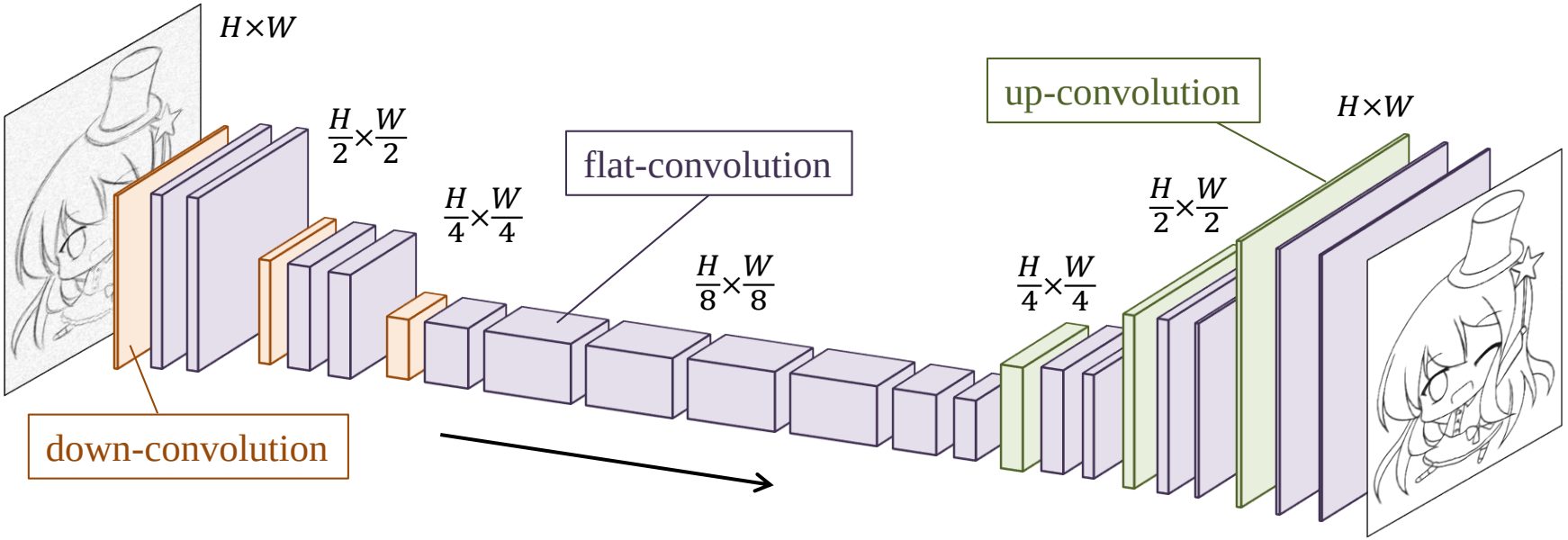
2層目
3層目
模様
（中レベル特徴）

4層目
犬の顔、鳥の足
（高レベル特徴）

Visualizing and UnderstandingConvolutional Networks [Zeiler+ ECCV ’14]

Fully Convolutional Neural Network

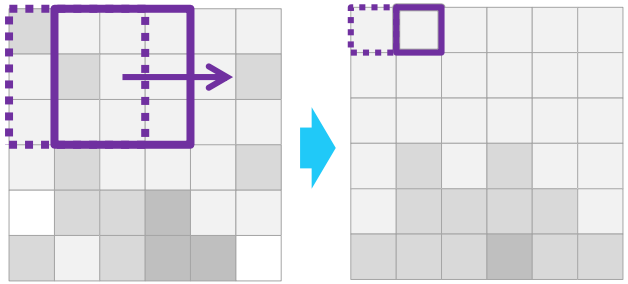
- すべての層が畳込み層
- 画像生成で非常に重要なモデル



[Simo-Serra and Iizuka+ SIGGRAPH ’16]

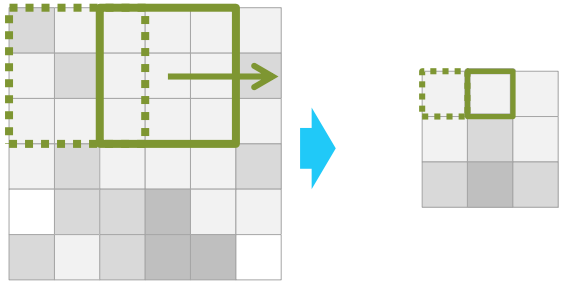
畳込みの種類

ストライド 1



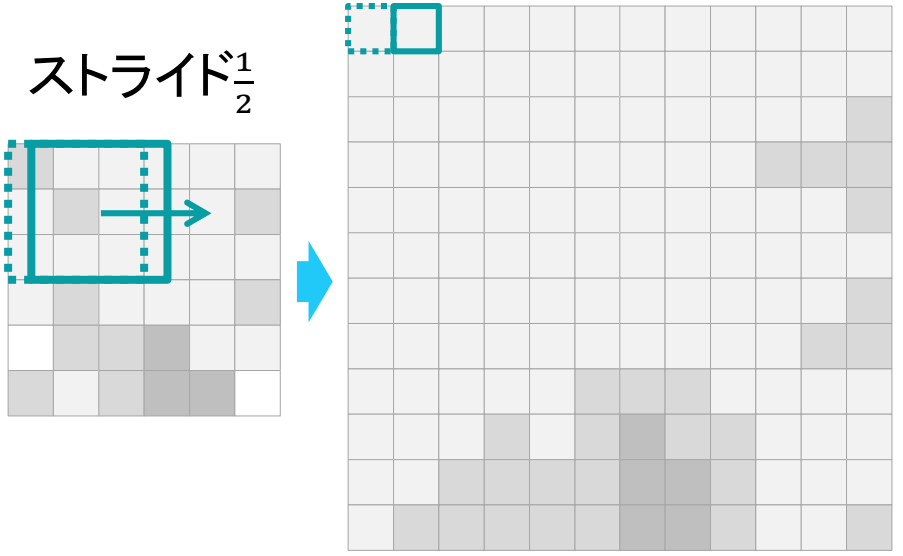
Flat-convolution

ストライド 2



Down-convolution

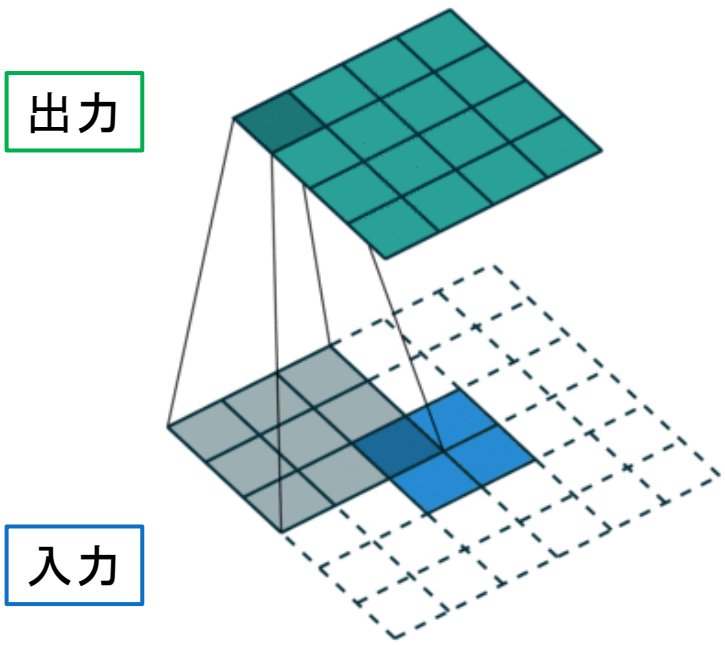
ストライド$\frac{1}{2}$

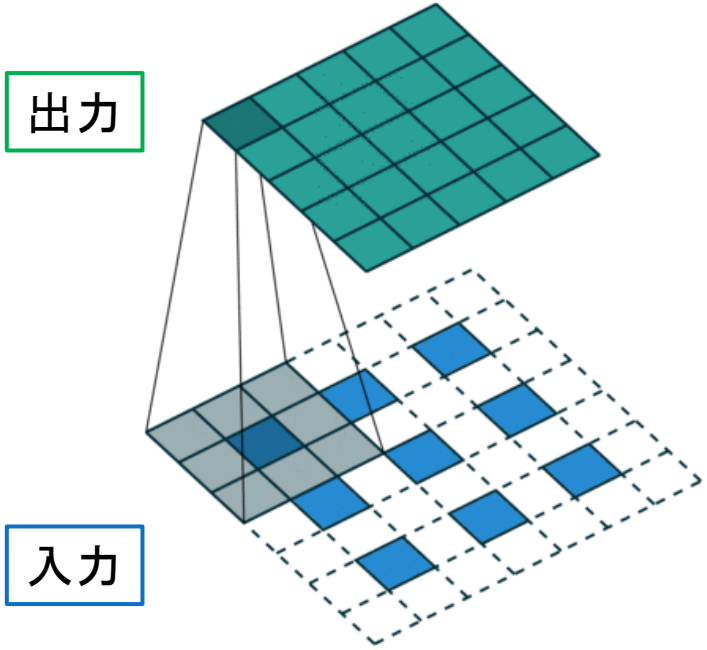


Up-convolution

Up Convolution

- 特徴マップのサイズを拡大
  - やり方はいろいろ



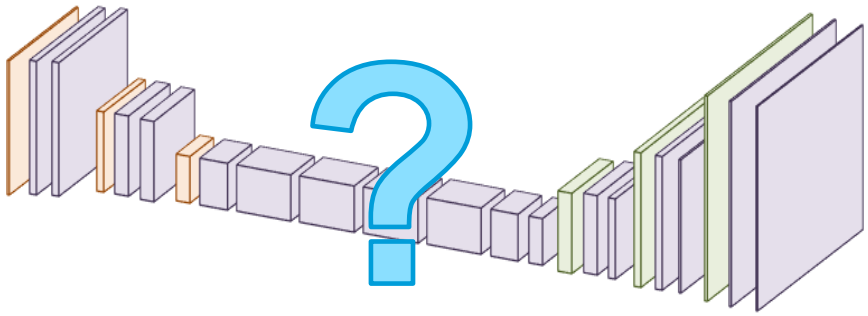


Fully CNNの利点

- 入力画像と同じサイズの画像を出力できる
  - どんなサイズの画像でも扱える
  - ピクセル単位の推定
- 学習と推定を画像全体で一度にできる
- 重みを減らせる
  - オーバーフィッティングしにくい

CNNの設計

- モデル構造
  - 層の数
  - フィルタのサイズ
  - フィルタの数
  - 活性化関数
  - プーリングの方法
- 学習
  - バッチサイズ
  - ロス関数
  - 最適化の手法



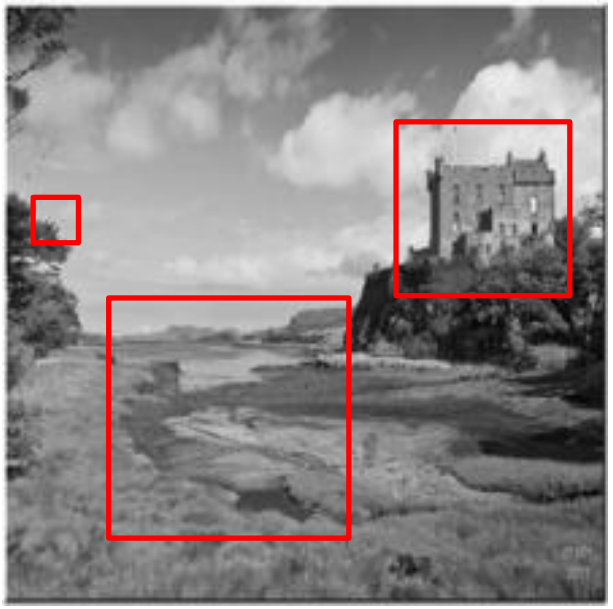
問題によって異なる ☹

モデルの表現力と学習の難しさ

	表現力	学習	必要データ数	利用例
層やフィルタの数：少	小	簡単	少	2クラス分類 文字認識
層やフィルタの数：多	大	困難	多	多クラス分類 画像生成

空間サポート

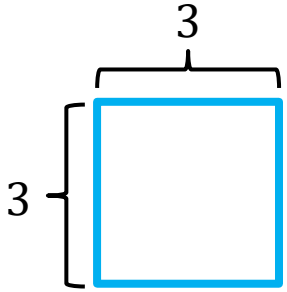
- どの程度の範囲の情報が必要か
  - モデルが「見る」範囲は決まっている
  - 特にFully CNNで重要

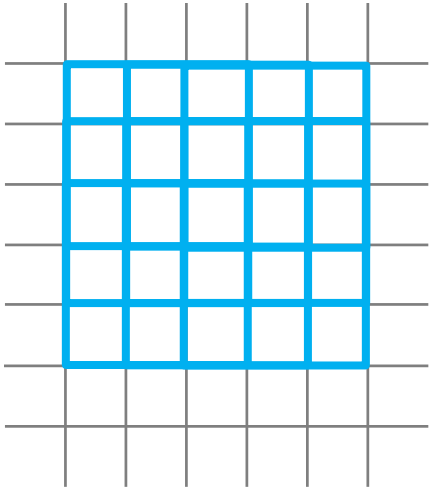


?

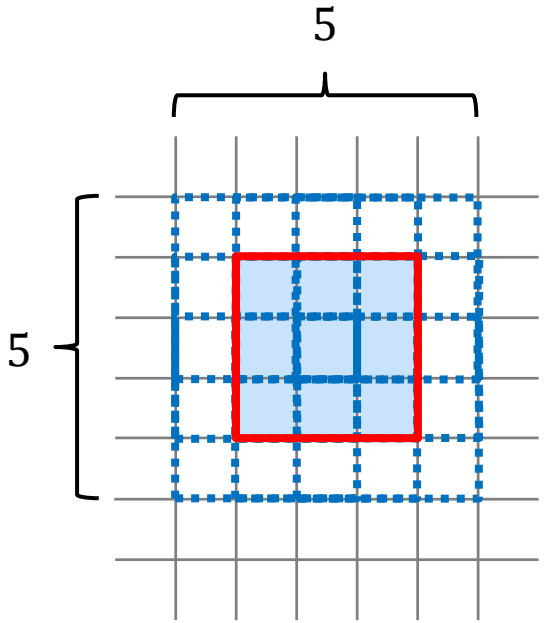
モデルが見られる範囲

- 3 × 3 カーネルで2回畳込み→ 5 × 5 カーネル
  - 非線形性が増す
  - 重みの数を減らせる





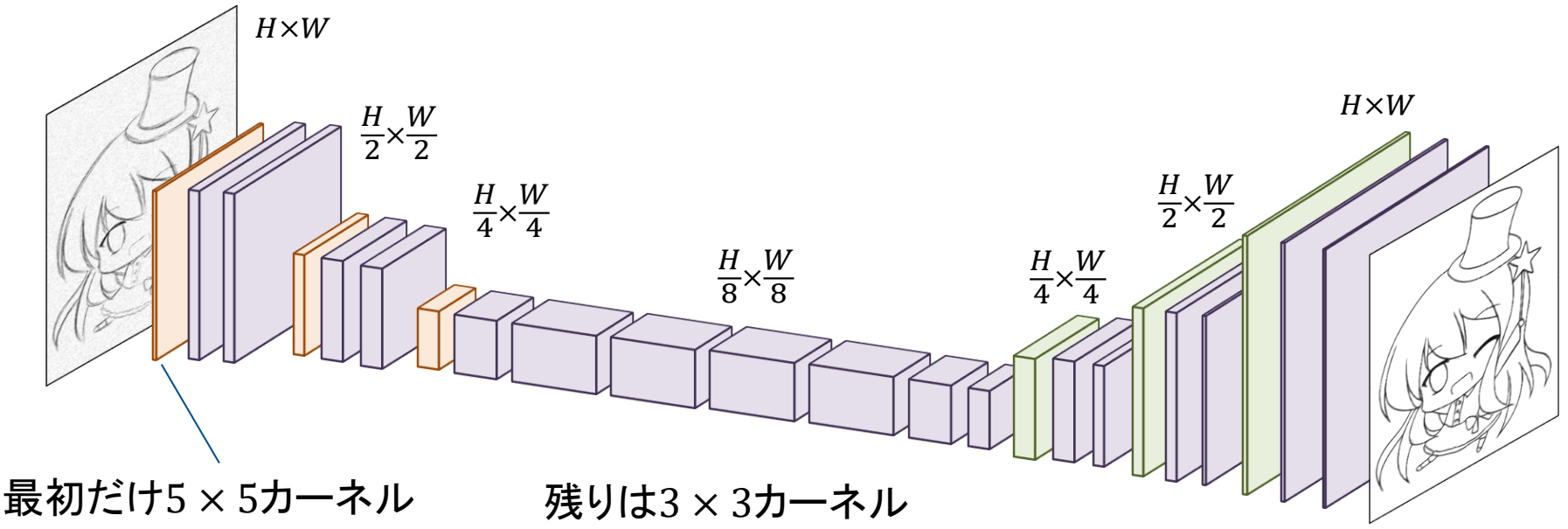
1層目



2層目

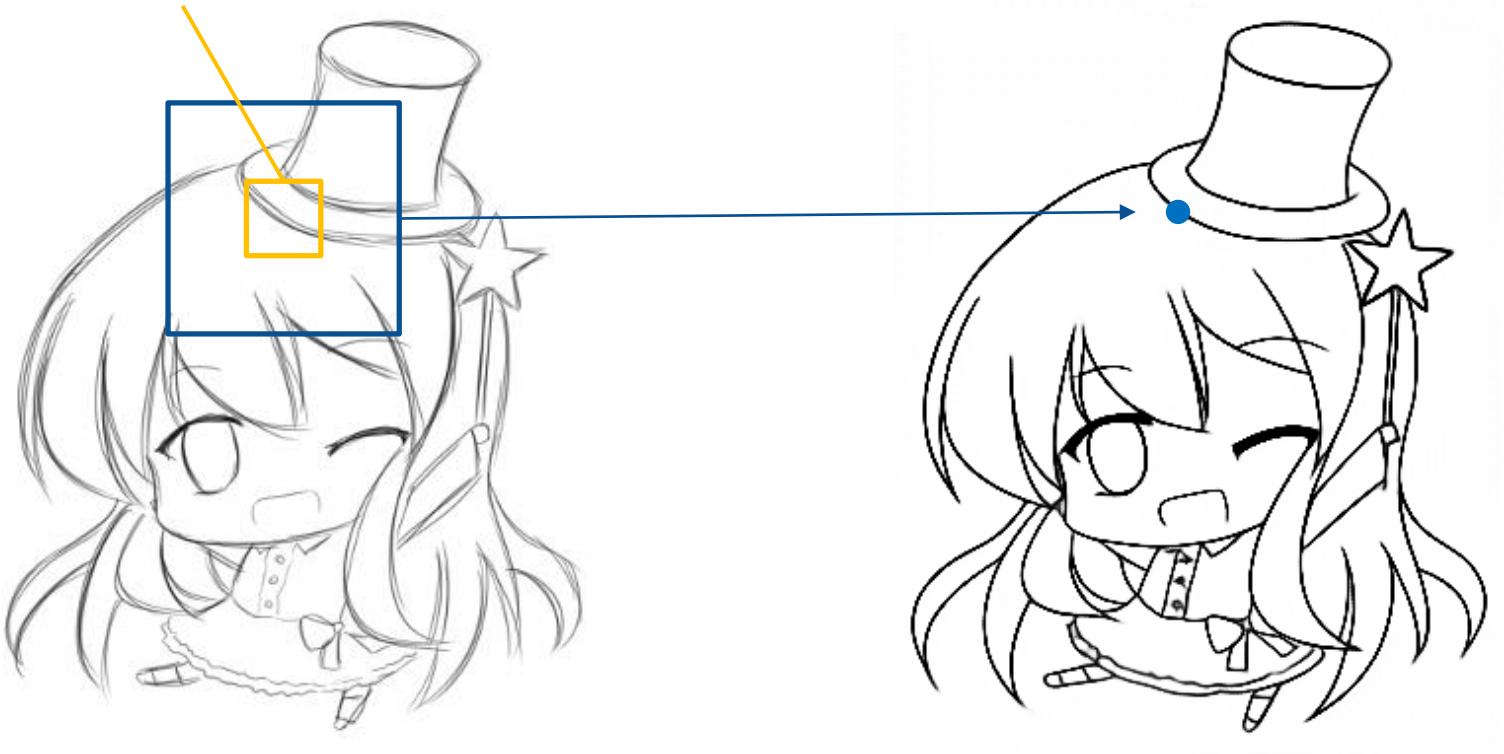
例えば

- このモデルは約200 × 200ピクセルを見ている
  - 実際はもっと小さい



つまり…

中心に近いほど影響が強い



ラフスケッチの自動線画化

まとめ

- ディープラーニングを用いた画像生成に必要な基礎知識を紹介
  - CNNとは
  - モデルの学習
  - モデルの設計

- 問題に応じて最適なモデルと学習方法を選択
  - 非常に重要

ご清聴ありがとうございました

- 質問や感想などありましたらお願いします
  - Email: iizuka@aoni.waseda.jp
  - ホームページ: http://hi.cs.waseda.ac.jp/~iizuka/