卒業演習発表
深層学習を用いた
ネコ科判定アプリケーション
の作成

日本大学文理学部　情報科学科　谷聖一研究室　天野晃介
2019/1/31

目次

１．はじめに

1–1動機

私天野晃介はネコアレルギーなので
ネコ科に近づくと危ない。

だが、動物がとても好きなのである。

1–1動機

画像がネコ科であるのかどうかを手軽に判定する
アプリケーションが欲しい

↓

触れるかすぐにわかる！

1-2目的

・画像を入力しその画像をネコ科の属ごとに分類
・どの属にも分類されない画像はネコ科ではないと判定

・アプリケーション化し手軽に行えるようにする

1-2目的

機械学習を用いてネコ科かどうか判定する
モデルを獲得し
ネコ科判定アプリケーションを作成していく

1-3演習内容

演習の流れ

深層学習を用いたネコ科判定アプリケーションの作成

学習に用いる画像を収集

↓

収集した画像を整形

↓

深層学習を用いて学習モデルを生成

↓

学習モデルを用いてアプリケーションを作成

2.準備

2-1機械学習

「明示的にプログラムしなくても学習する能力をコンピュータに与える研究分野」“Field of study that gives computers the ability to learn without being explicitly programmed”

(*Arthur Lee Samuel 1959*)

明示的に表現することが困難なものを、
具体的な事例などから計算機に帰納的に学習させること

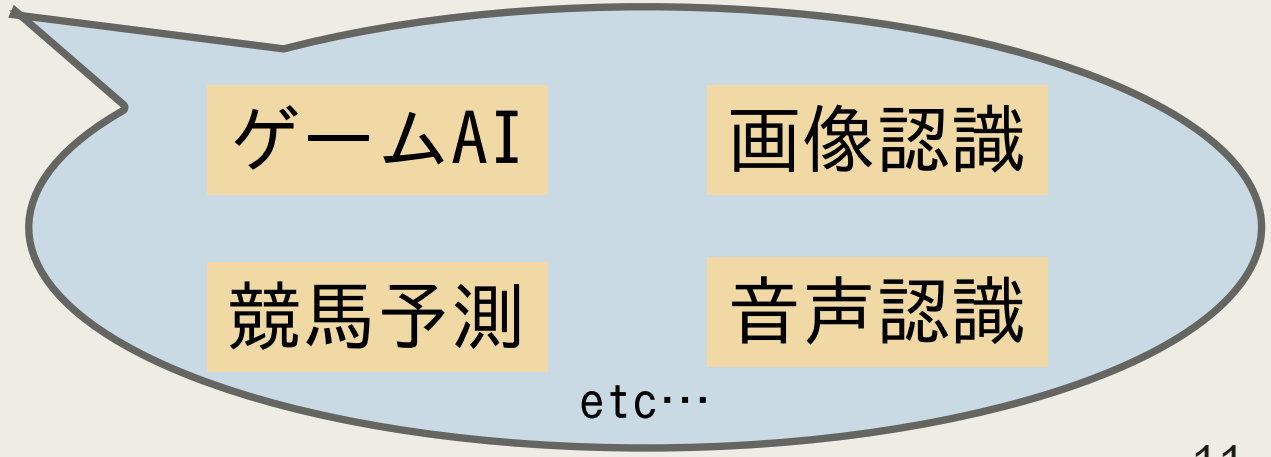
(「深層学習」, 人工知能学会, 2016, 4p)

2-2 深層学習

・機械学習の一種
・多層構造のニューラルネットワークを用いた機械学習

（「深層学習」,人工知能学会,2016,3p）

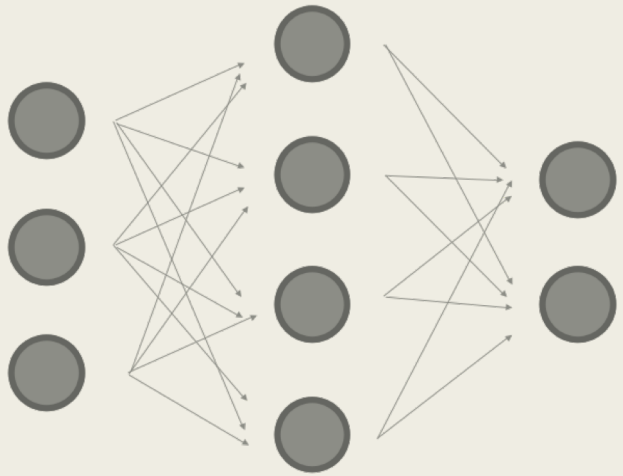
・様々な分野に応用される

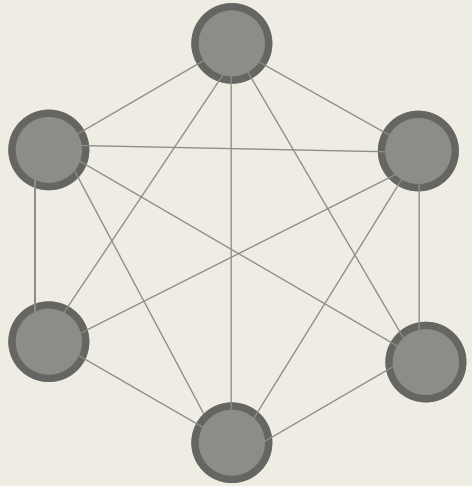


2–2 深層学習

ニューラルネットワークとは

脳神経系の情報処理機構を模倣した数理モデル





2-2 深層学習

ニューラルネットワーク

ラベル付きのデータのデータを
入力とし、出力値を算出

二乗誤差

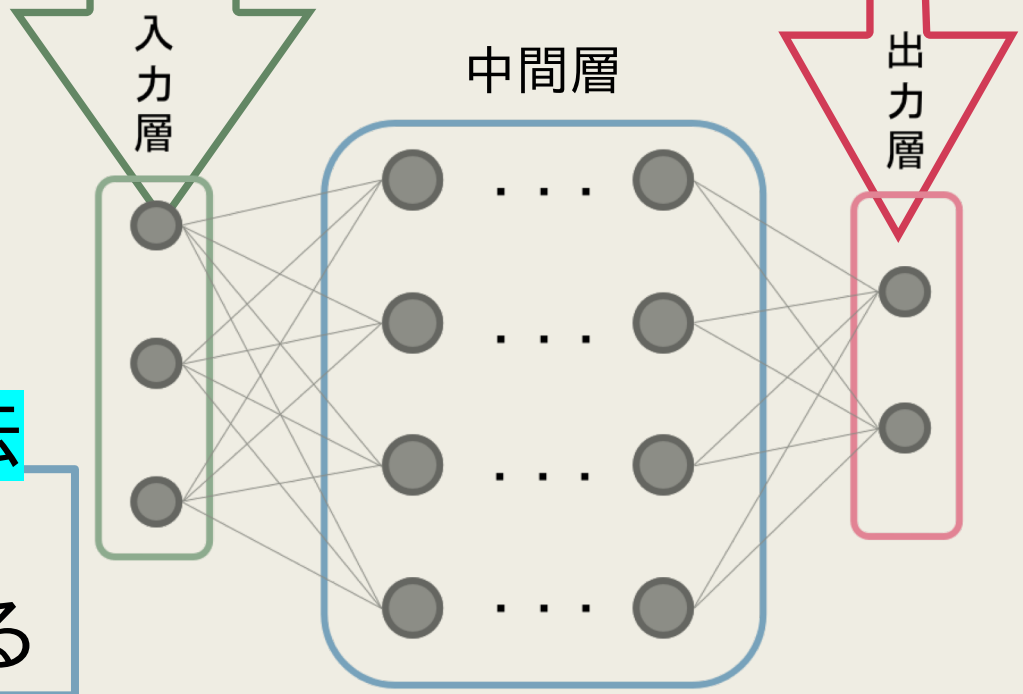
ラベルと出力値の誤差を計算

確率的勾配降下法

ラベルと出力値の誤差が
小さくなるように重みを修正する

ラベル付きデータの例

番号	データ	ラベル
1	[01001‥‥11]	[1,0,0,‥‥,0]
2	[01111‥‥11]	[0,1,0,‥‥,0]
⋮	⋮	⋮
n	[01000‥‥01]	[0,0,0,‥‥,1]



2-3 CNN(Convolutional Neural Network)

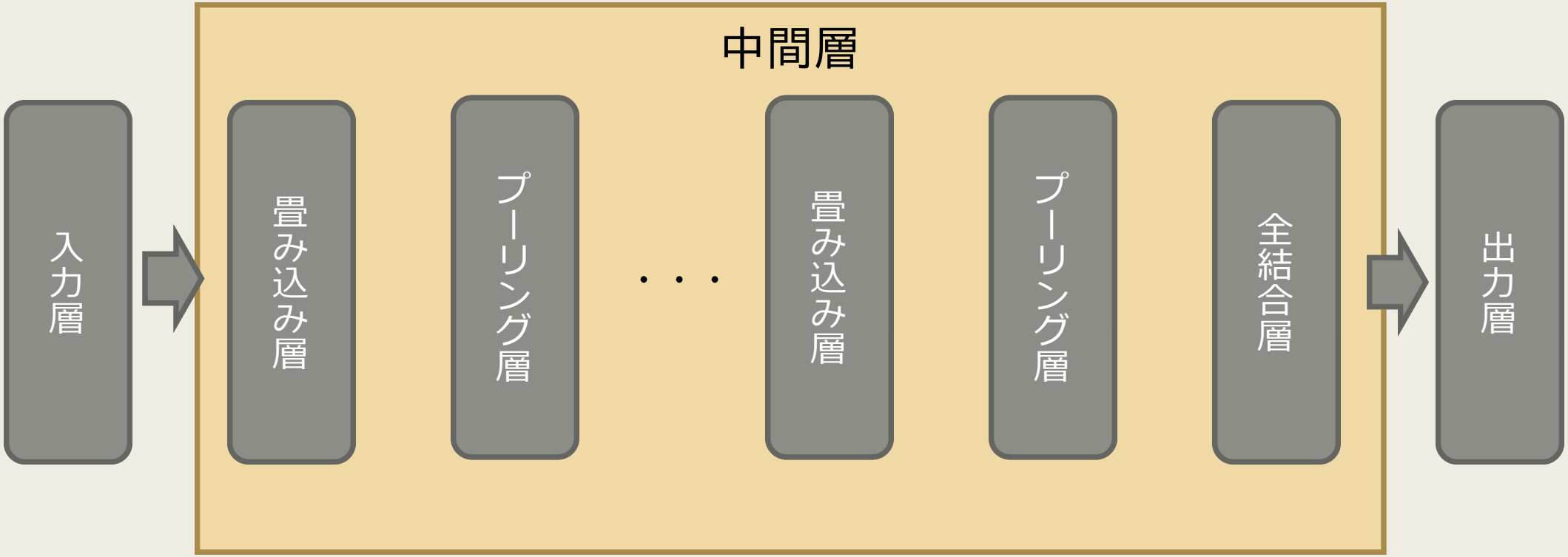
・深層学習における応用的な手法の一つ

・CNNは、画像認識や音声認識などよく使われている

特に画像認識では優れた実績を収めている

移動不変性をもたせることができる

2-3 CNN(Convolutional Neural Network)



2-3 CNN(Convolutional Neural Network)

畳み込み層

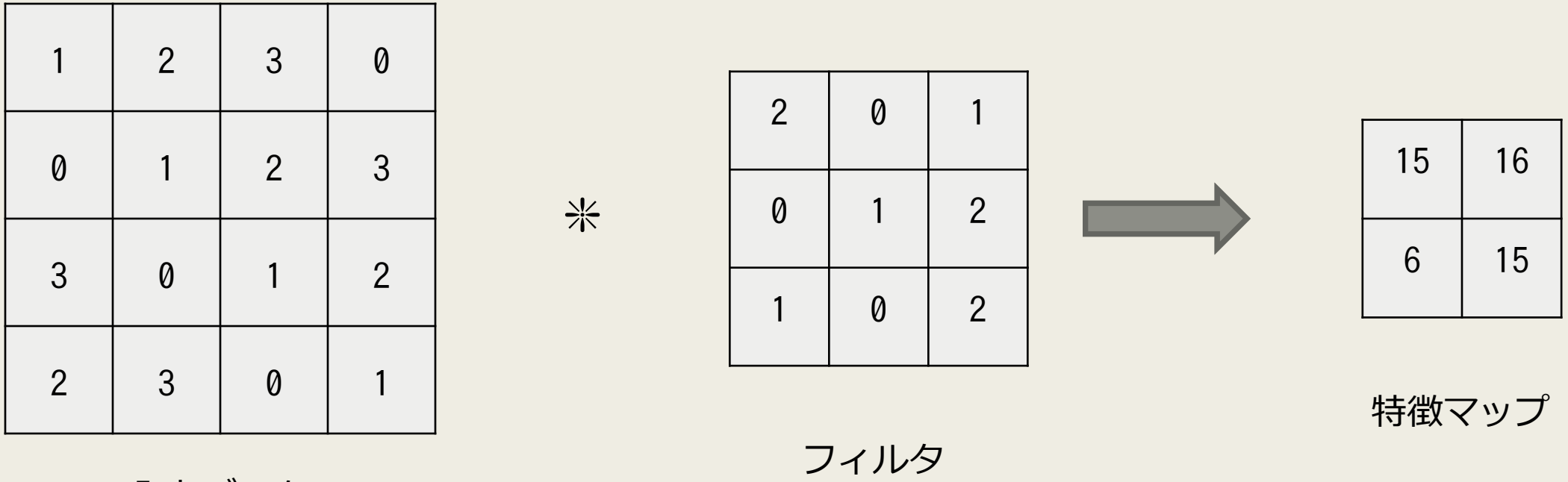
入力のあらゆる位置で特徴の比較と一致点の検出を
試み画像全体にわたって
特徴の一致件数を計算し特徴マップを作成する

2-3 CNN(Convolutional Neural Network)

畳み込み層

入力に対してフィルタとの内積を計算し特徴マップを作成

計算例



2-3 CNN(Convolutional Neural Network)

畳み込み層

計算例

1	2	3	0
0	1	2	3
3	0	1	2
2	3	0	1

$*$

2	0	1
0	1	2
1	0	2

15	

2-3 CNN(Convolutional Neural Network)

畳み込み層

計算例

1	2	3	0
0	1	2	3
3	0	1	2
2	3	0	1

＊

2	0	1
0	1	2
1	0	2

15	16

2-3 CNN(Convolutional Neural Network)

畳み込み層

計算例

1	2	3	0
0	1	2	3
3	0	1	2
2	3	0	1

＊

2	0	1
0	1	2
1	0	2

15	16
6	

2-3 CNN(Convolutional Neural Network)

畳み込み層

計算例

1	2	3	0
0	1	2	3
3	0	1	2
2	3	0	1

$*$

2	0	1
0	1	2
1	0	2

15	16
6	15

2-3 CNN(Convolutional Neural Network)

プーリング層

プーリング層は畳み込み層から出力された
特徴マップを縮小する

大きな画像を重要な情報は残しつつ縮小することができる

2-3 CNN(Convolutional Neural Network)

プーリング層

MaxPooling

1	2	3	0
0	1	2	3
3	0	1	2
2	3	0	1

MaxPooling(2, 2)

2	3
3	2

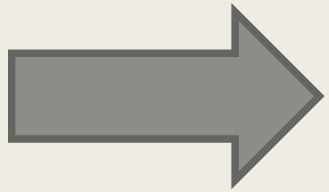
他にもaveragePloolingなどもある

2-3 CNN(Convolutional Neural Network)

パディング

畳み込み層の前に入力データの周囲に固定のデータを埋めること

1	2	3	0
0	1	2	3
3	0	1	2
2	3	0	1



0	0	0	0	0	0
0	1	2	3	0	0
0	0	1	2	3	0
0	3	0	1	2	0
0	2	3	0	1	0
0	0	0	0	0	0

2-3 CNN(Convolutional Neural Network)

ストライド

フィルタを適用する位置の間隔のこと

7	12	10	2
4	15	16	10
10	6	15	6
8	10	4	3

1	0
1	1

ストライド　1

7	12	10	2
4	15	16	10
10	6	15	6
8	10	4	3

7	12	10	2
4	15	16	10
10	6	15	6
8	10	4	3

2-3 CNN(Convolutional Neural Network)

ストライド

フィルタを適用する位置の間隔のこと

7	12	10	2
4	15	16	10
10	6	15	6
8	10	4	3

1	0
1	1

ストライド　2

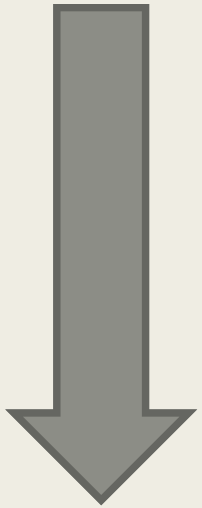
7	12	10	2
4	15	16	10
10	6	15	6
8	10	4	3

7	12	10	2
4	15	16	10
10	6	15	6
8	10	4	3

３.学習モデル生成

3-1学習モデル生成手順

データの収集
データの整形
モデルの構築
学習モデルの評価



3-1学習モデル生成手順

・演習環境

機器	MacBook Air 11-inch Mid 2012
OS	macOS High Sierra
CPU	Intel Core i7
クロック	2 GHz
メモリ	8 GB

・開発言語

Python	3.6

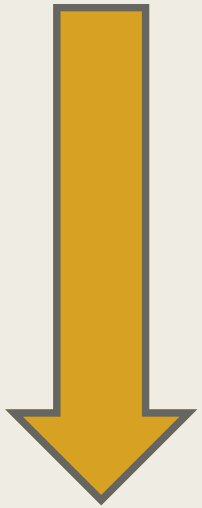
3-1学習モデル生成手順

使用したライブラリ

ライブラリ		バージョン
Keras	深層学習ライブラリ	2.1.6
Pillow	画像処理ライブラリ	4.0.0
NumPy	データ分析	1.14.5
sckit-learn	機械学習ライブラリ	0.18.1
TensorFlow	計算ライブラリ	1.5.0
coremltools	フレームワーク	2.0

3-1学習モデル生成手順(再掲)

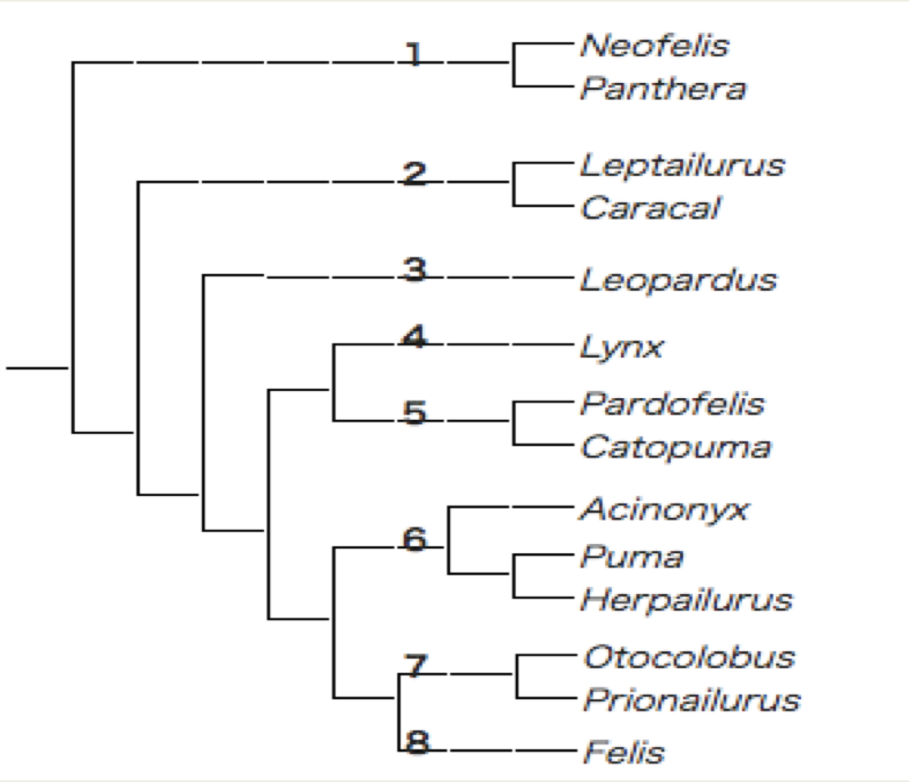
データの収集
データの整形
モデルの構築
学習モデルの評価



3-2データの収集

画像の収集

番号	系統	枚数
1	Panthera lineage	4000
2	Caracal lineage	4000
3	Ocelot lineage	4000
4	Lynx lineage	4000
5	Bay cat lineage	4000
6	Puma lineage	4000
7	Leopard cat lineage	4000
8	Domestic cat lineage	4000
9	other	3000



Wikipedia ネコ科
https://ja.wikipedia.org/wiki/%E3%83%8D%E3%82%B3%E7%A7%91
2019年1月19日参照

ネコ科でない動物の画像には
犬，牛，馬，熊，狐，さる，象，かに，ワニ，鳥，人が含まれる

3-2データの収集

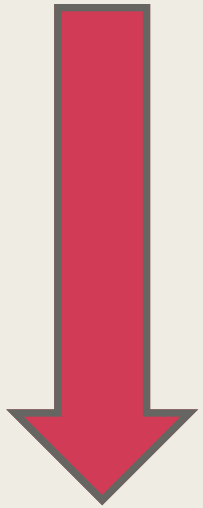
画像収集はExcite画像検索からスクレイピングで行う

本演習では github 上のコードを利用させてもらった

excite-image-scraping
https://github.com/Doarakko/scraping-challenge/tree/master/excite-image-scraping
2019年1月28日参照

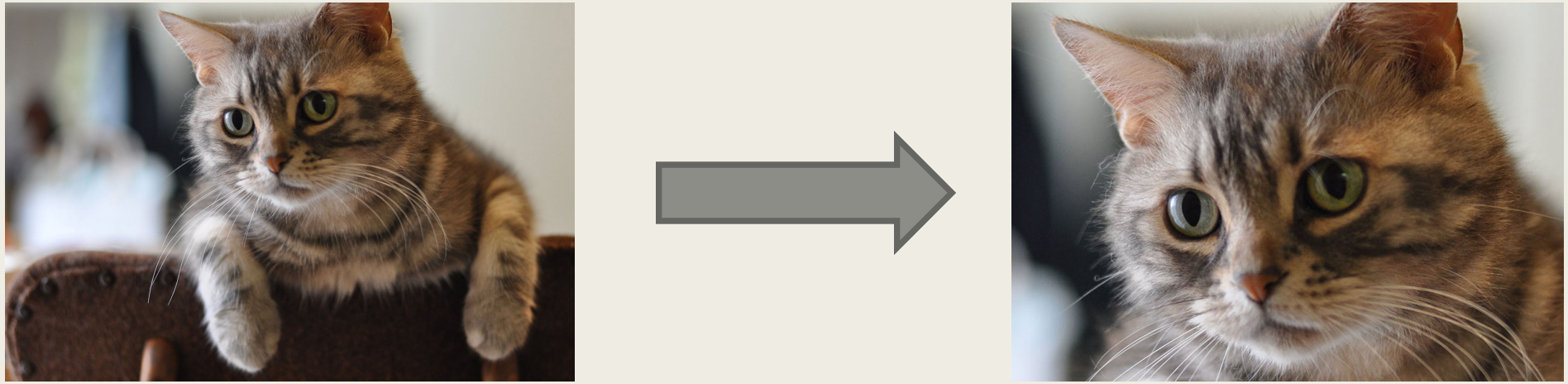
3-1学習モデル生成手順(再掲)

データの収集
データの整形
モデルの構築
学習モデルの評価



3-3データの整形

収集してきた各画像において手作業で顔部分をトリミング



猫休み展　http://tgs.jp.net/event/neko-break-2017winter　2019年1月28日参照

3-3データの整形

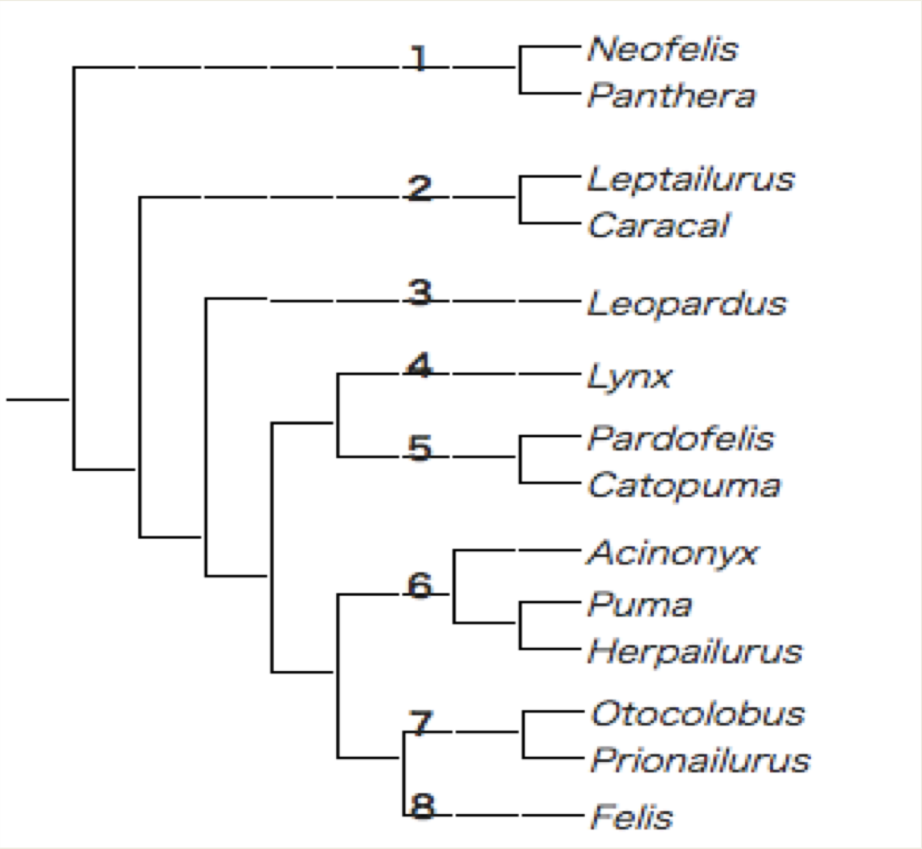
以下の画像は除去

- 該当の系統に当てはまらない画像
- 画像のサイズが極端に小さい画像
- 顔が写っていない画像

3-3データの整形

結果

番号	系統	枚数
1	Panthera lineage	2000
2	Caracal lineage	1740
3	Ocelot lineage	1982
4	Lynx lineage	1889
5	Bay cat lineage	1887
6	Puma lineage	1764
7	Leopard cat lineage	1785
8	Domestic cat lineage	1846
9	other	2012



Wikipedia ネコ科
https://ja.wikipedia.org/wiki/%E3%83%8D%E3%82%B3%E7%A7%91
2019年1月19日参照

3-3データの整形

データの前処理

・Pillow を用いて画像データのカラーモードを RGB に変換し
画像サイズを 50×50 に統一

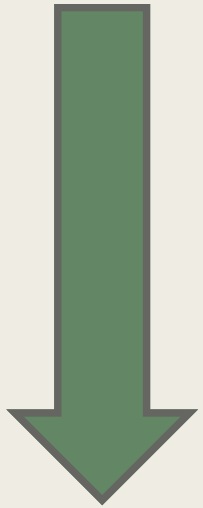
・NumPy を用いて画像データを
ニューラルネットワークに入力できる形式に変換

NumPyとは
“NumPy is the fundamental package
for scientific computing with Python.”
（NumPyはPythonを用いた科学計算のための基本的なパッケージ）

NumPy NumPy http://www.numpy.org/ 2019年1月17日参照

3-1学習モデル生成手順(再掲)

データの収集
データの整形
モデルの構築
学習モデルの評価



2-4モデルの構築

モデルの構築は Keras を用いて行う

Keras とは

「Python で書かれた，TensorFlow または CNTK , Theano 上で実行可能な高水準のニューラルネットワークライブラリ」

ニューラルネットワークを構築する

Keras Pythonの深層学習ライブラリ
https://keras.io/ja/
2019年1月17日参照

2-4モデルの構築

本演習では Keras を TensorFlow 上で実行

TensorFlowとは

「データフローグラフを使用して数値計算を行うための
オープンソースソフトウェアライブラリ」

計算を高速で行うことができる

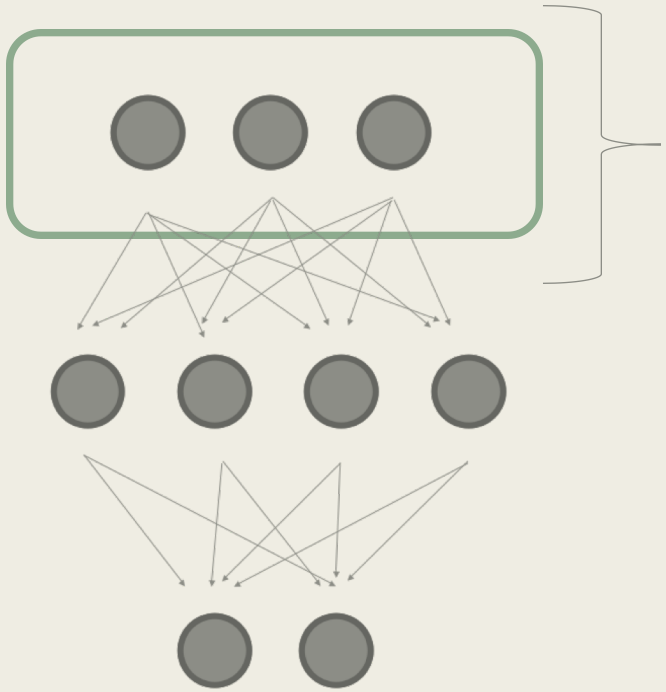
データフローグラフとは

頂点は計算処理を表し，枝で処理の流れを表すグラフ

TensorFlow TensorFlow
https://www.tensorflow.org/?hl=ja
2019年1月17日参照

2-4モデルの構築

モデルの構築



2500次元

入力層

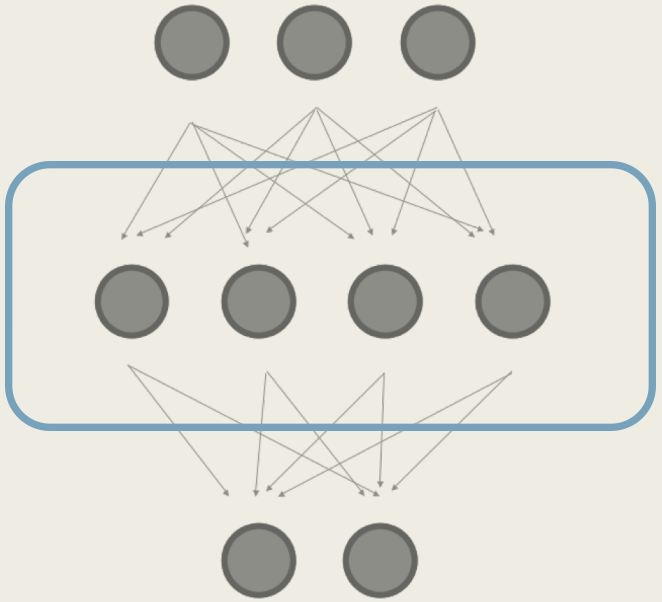
（50×50）のRGB画像（2次元）の数値を入力

入力層は2500次元

Keras の“Sequential”で構築

2-4モデルの構築

モデルの構築



中間層
CNNを適用

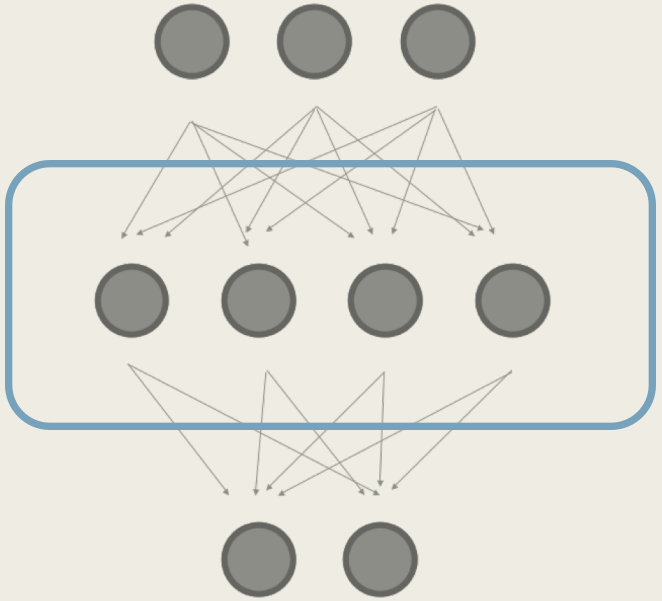
- 畳み込み層
- プーリング層
- 活性化関数

Keras の
“Conv2D”
を使用

3層ずつ重ねたモデルを作成

2-4モデルの構築

モデルの構築

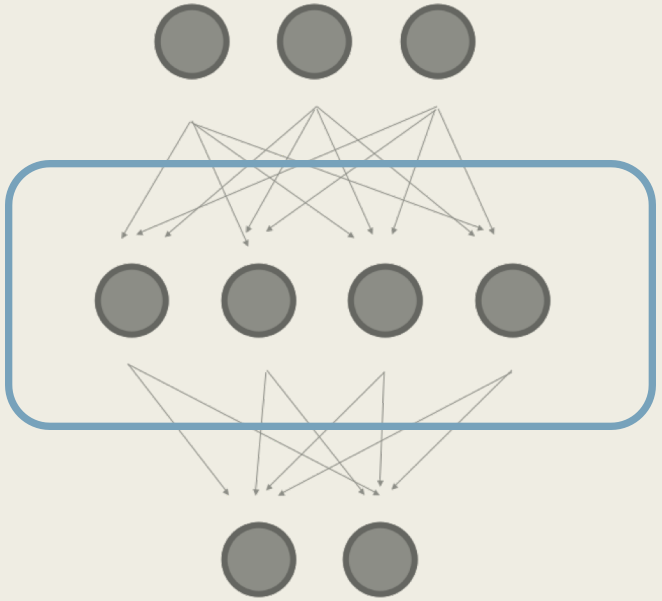


Keras の
“Conv2D”
を使用

中間層
畳み込み層
・フィルタは3×3
・0でパディングを行うよう設定
・ストライドは１

2-4モデルの構築

モデルの構築

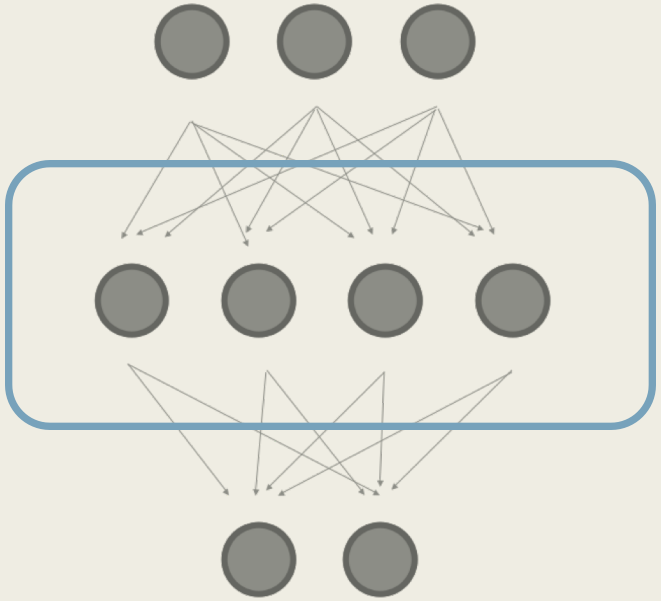


Keras の
“MaxPooling2D”
を使用

中間層
・プーリング層
“MaxPooling”を2×2で設定

2-4モデルの構築

モデルの構築

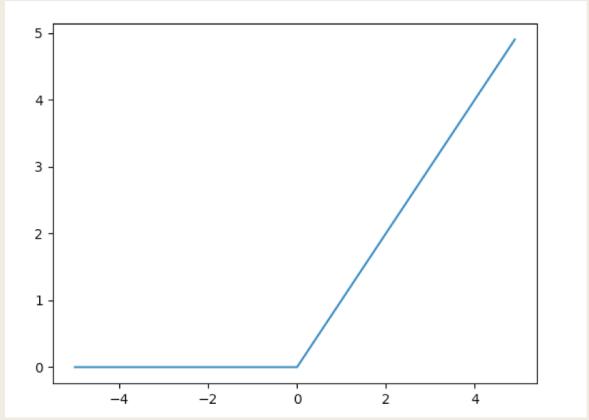


Keras の
“Activation”
を使用

中間層

活性化関数には “ReLU”(正規化線形関数)を使用

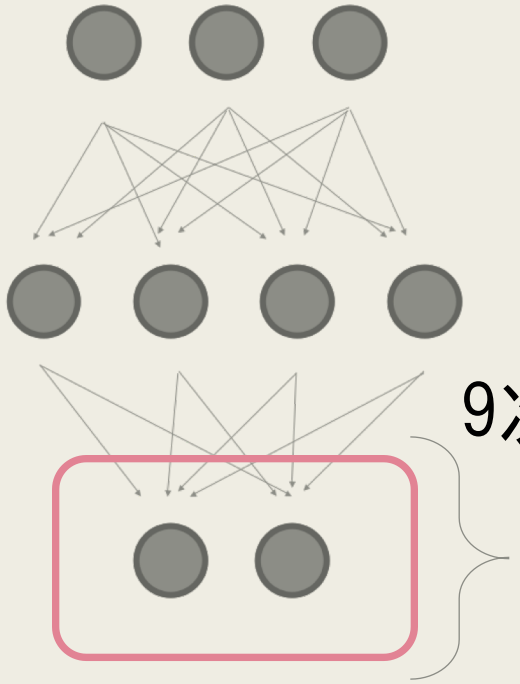
$$x = x\ (x > 0)$$
$$x = 0\ (x \leq 0)$$



2-4モデルの構築

Keras の
“Dense”
“Activation”
を使用

モデルの構築

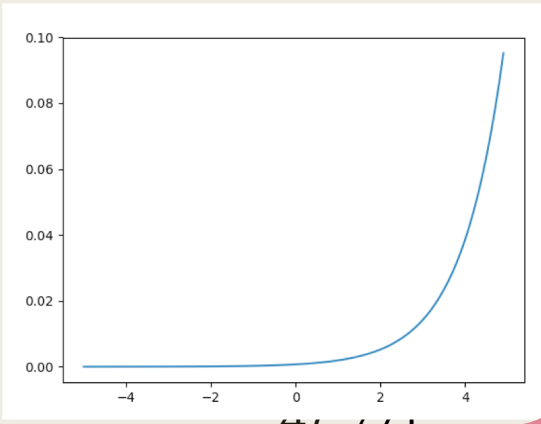


9次元

出力層
　中間層から伝わってきた値に
　「SoftMax関数」を適用
　9クラスに分類するので9次元

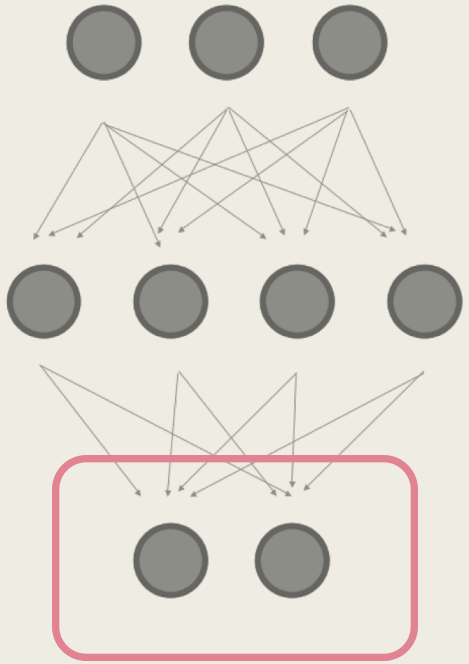
出力値の合計が1になるように変換
出力を確率とみなすことができる

$$y_k = \frac{exp(a_k)}{\sum_{i=1}^{n} exp(a_i)}$$



2-4モデルの構築

モデルの構築

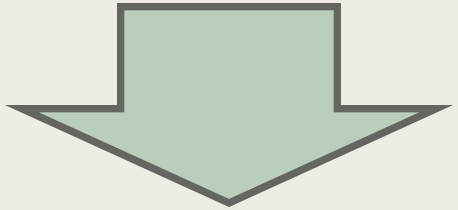


出力層

「SoftMax関数」において求めた
尤度が最大となる
クラスを分類クラスとする

2-4モデルの構築

何回繰り返して学習させるか



ホールドアウト法を用いて評価する

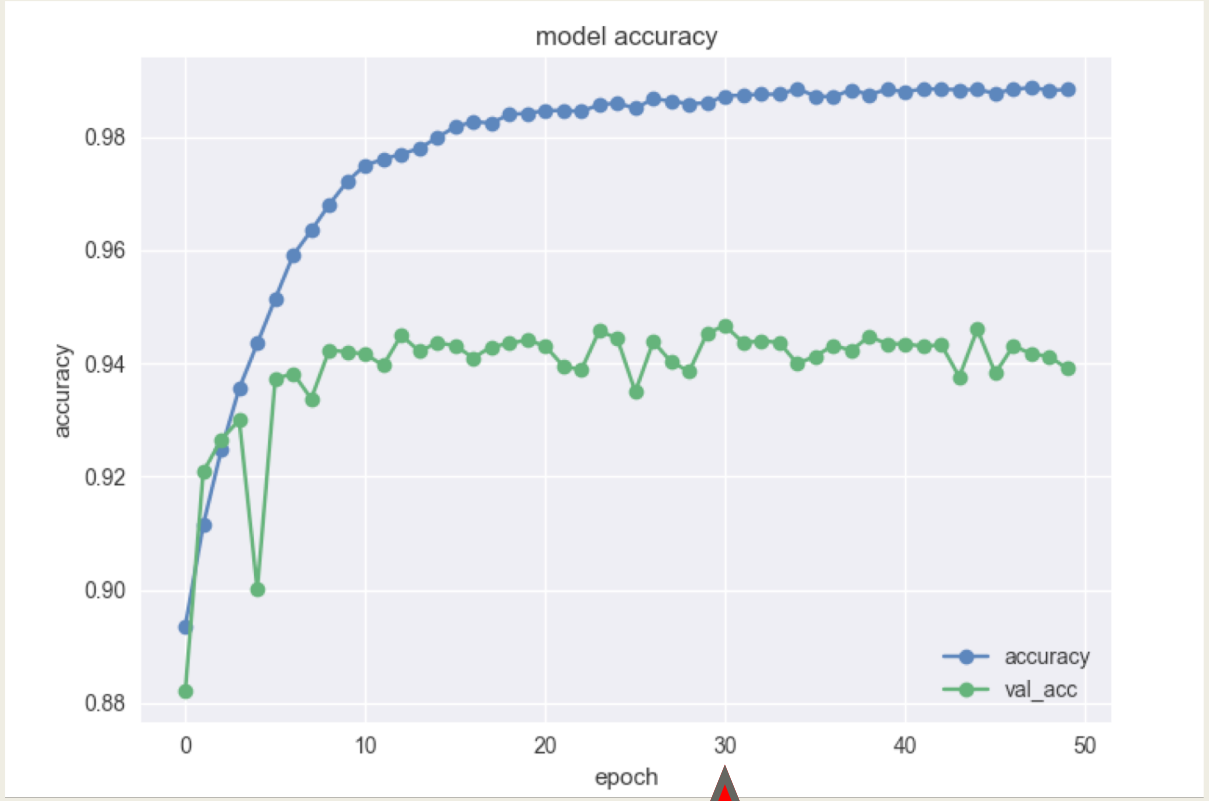
2-4モデルの構築

ホールドアウト法

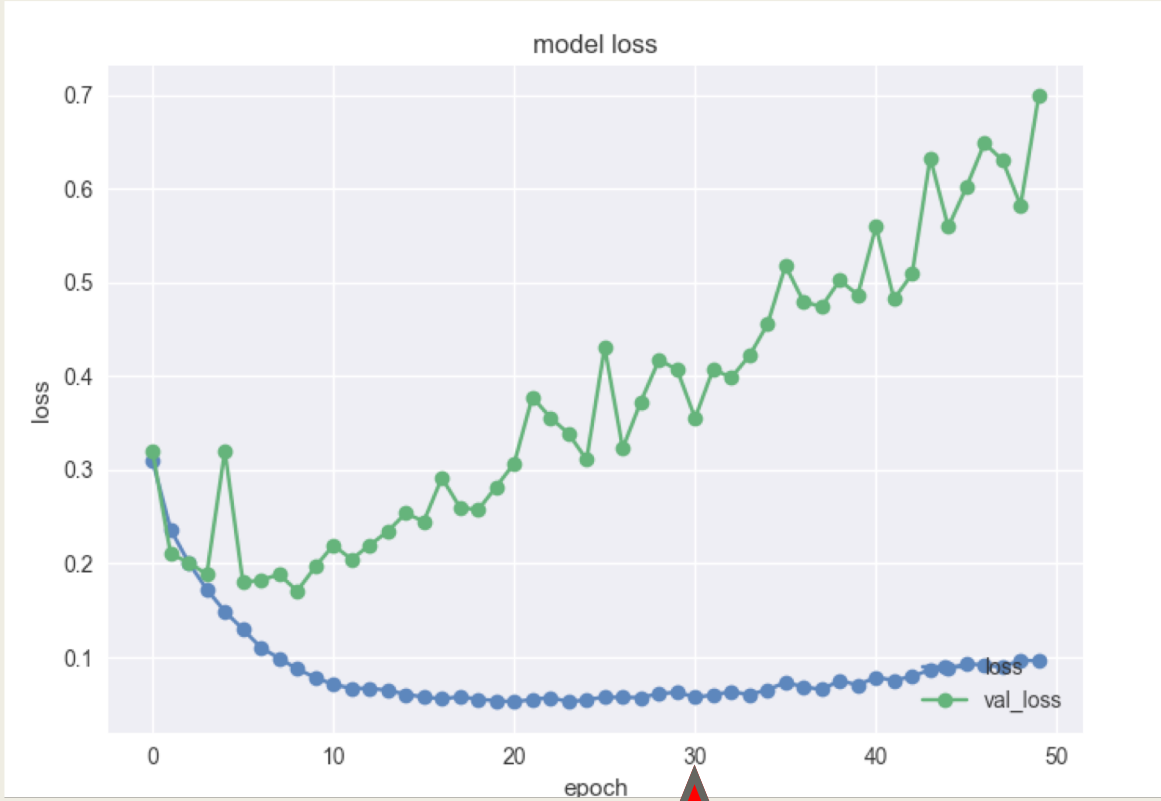
学習データとテストデータの2分割にする

2-4モデルの構築

学習回数に対する正答率



学習回数に対する誤差関数の出力値



学習回数を30回とする

2-4モデルの構築

生成した学習モデルの保存

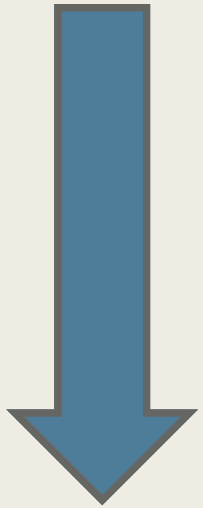
生成した学習モデルを hdf5 形式で保存

hdf5 形式

階層化されたデータ群を取り扱うファイル形式

3-1学習モデル生成手順(再掲)

データの収集
データの整形
モデルの構築
学習モデルの評価



3-5学習モデルの評価

+ 入力：image3.jpeg
| グループ名：Felis
ネコ科です！今すぐ離れて！！！

猫を『買う』以外の選択肢　意外と知られていない？
https://grapee.jp/329450
2019年1月28日参照

3-5学習モデルの評価

みんなの犬図鑑
https://www.min-inuzukan.com/
2019年1月28日参照

+ 入力：image1.jpeg
| グループ名：other
ネコ科ではありません。触ってよし！！

3-5学習モデルの評価

+ 入力：image6.jpeg
| グループ名：other
ネコ科ではありません。触ってよし！！

猫が喜ぶおもちゃはこれ！ 試してわかった傑作猫用グッズ20選【猫を愛するすべての人へ】
https://kakakumag.com/houseware/?id=9840
2019年1月28日参照

3–5学習モデルの評価

交差検証法

使用したライブラリ： sckit-learn

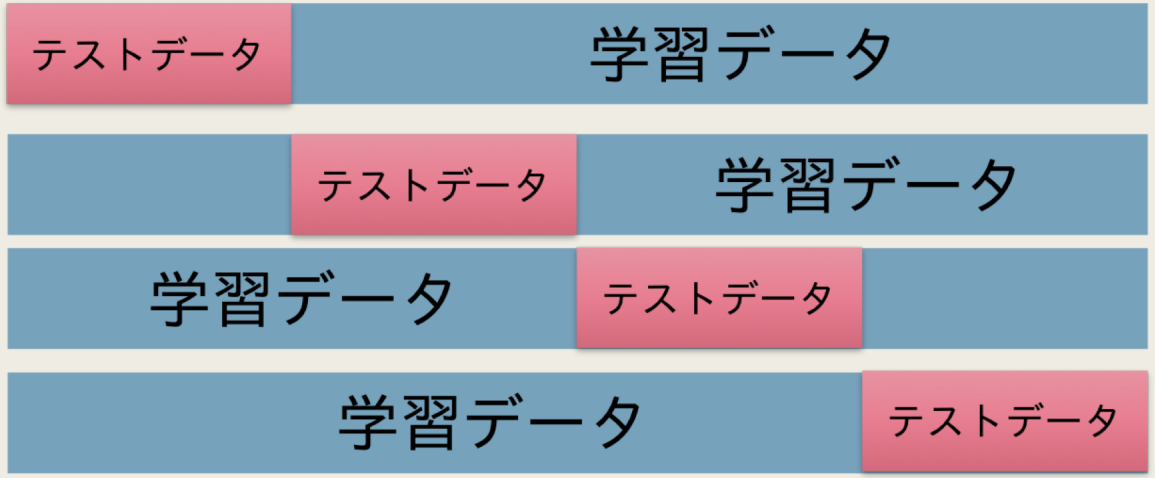
sckit-learnとは

“Simple and efficient tools for data mining and data analysis”
（シンプルで効率的にデータマイニングとデータ分析を行うツール）

Scikit-learn scikit-learn
https://scikit-learn.org/stable/index.html#
2019年1月29日参照

3–5学習モデルの評価

交差検証法



K 個のデータに分割して K 回評価する
その後平均値を取る

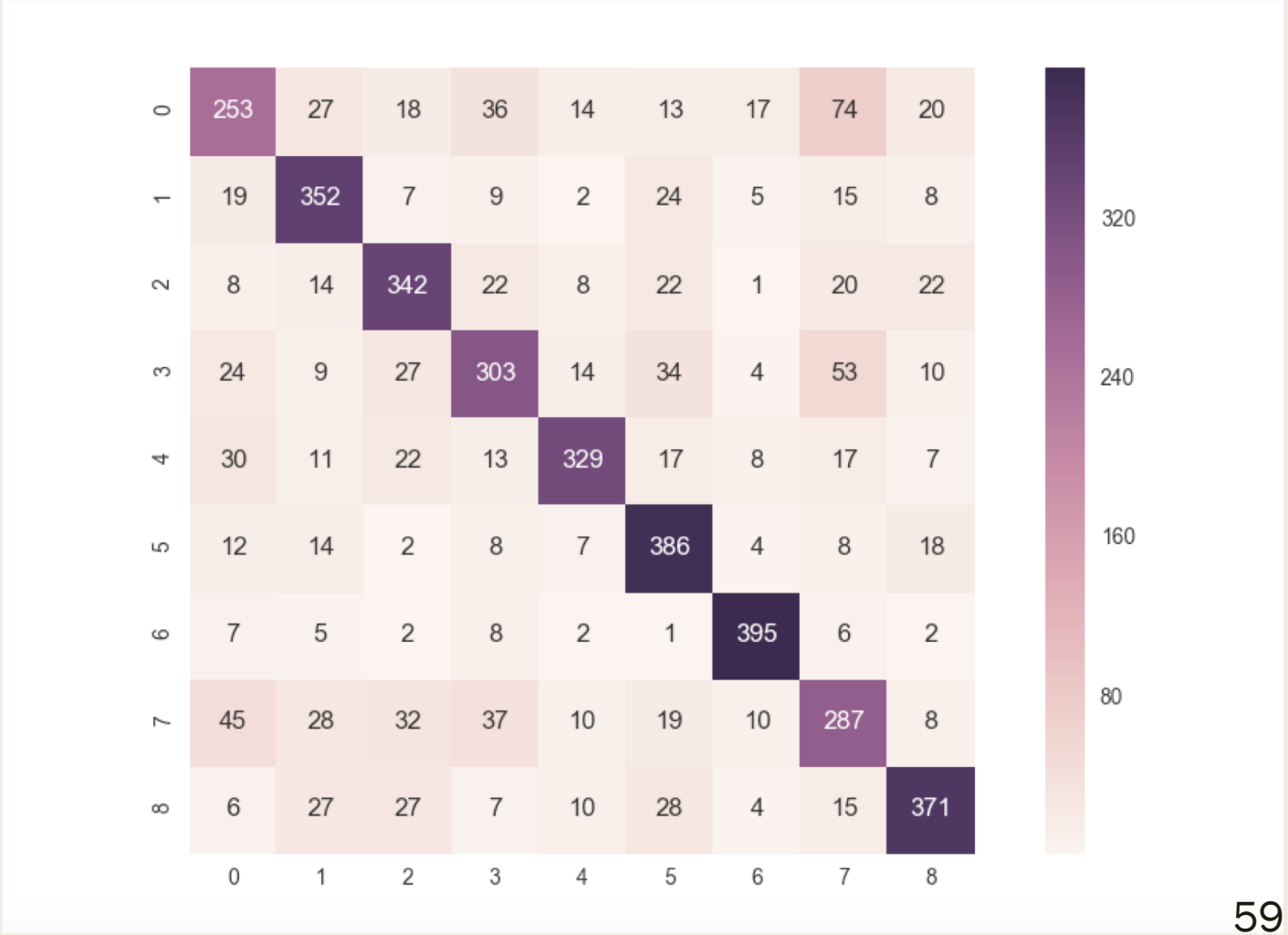
本演習では4個のデータに分割

3-5学習モデルの評価

混同行列

予想されたクラス

実際のクラス



3-5学習モデルの評価

正解率 (Accuracy)

　正や負と予測したデータのうち，実際にそうであるものの割合

適合率 (Precision)

　実際に正であるもののうち，正であると予測されたものの割合

適合率(Recall)

　正と予測したデータのうち，実際に正であるものの割合

F-値 (F-measure)

　適合率と適合率の調和平均

3-5学習モデルの評価

番号	属	正解率	適合率	再現率	F-値
1	Panthera lineage	0.947	0.828	0.782	0.804
2	Caracal lineage	0.932	0.759	0.733	0.746
3	Ocelot lineage	0.907	0.679	0.634	0.656
4	Lynx lineage	0.943	0.795	0.773	0.784
5	Bay cat lineage	0.908	0.544	0.696	0.611
6	Puma lineage	0.931	0.844	0.703	0.767
7	Leopard cat lineage	0.968	0.967	0.815	0.885
8	Domestic cat lineage	0.900	0.559	0.667	0.608
9	other	0.946	0.886	0.709	0.788

４.学習モデルを用いた
アプリケーション化

4–1アプリケーション化手順

・演習環境

機器	MacBook Air 11-inch
Xcode	10.0
デバイス	iPhone X
OS	iOS 12.0.1

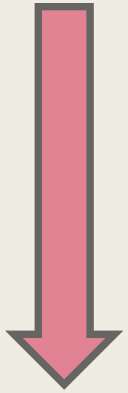
・開発言語

Swift	4. 2

4-1アプリケーション化手順

・アプリケーション化手順

生成した学習モデルを変換
iOSアプリケーションの作成
~~アプリケーションの評価~~



4-2アプリケーション化内容・方法

Core ML を用いて学習モデルをiOSで扱うことのできる

mlmodel 形式に変換

Core MLとは

「Siri，カメラ，QuickTypeなど，
様々なApple製品で用いられている機械学習フレームワーク」

「アプリで使用するための機械学習モデルを作成する」

Apple Inc.　機械学習
https://developer.apple.com/jp/machine-learning/
Apple Inc.　Create ML
https://developer.apple.com/documentation/createml
2019年1月11日参照

4-2アプリケーション化内容・方法

Core ML を用いて学習モデルをiOSで扱うことのできる

mlmodel 形式に変換

mlmodel 形式

Keras などにて生成した学習モデルを
CoreML フレームワークで使用するフォーマットに変換したもの

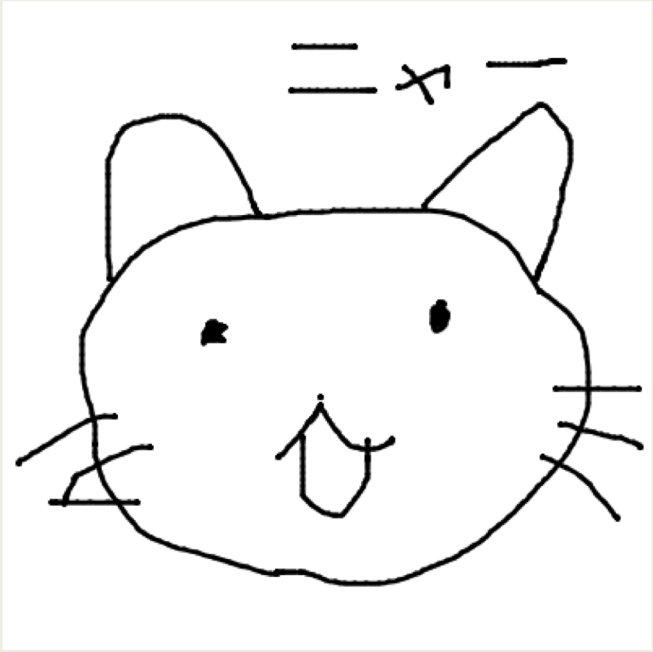
4-2アプリケーション化内容・方法

Xcode,Swift を用いてスマートフォンで撮影した写真がネコ科かどうかを判定するアプリケーションを作成する

4-2アプリケーション化内容・方法

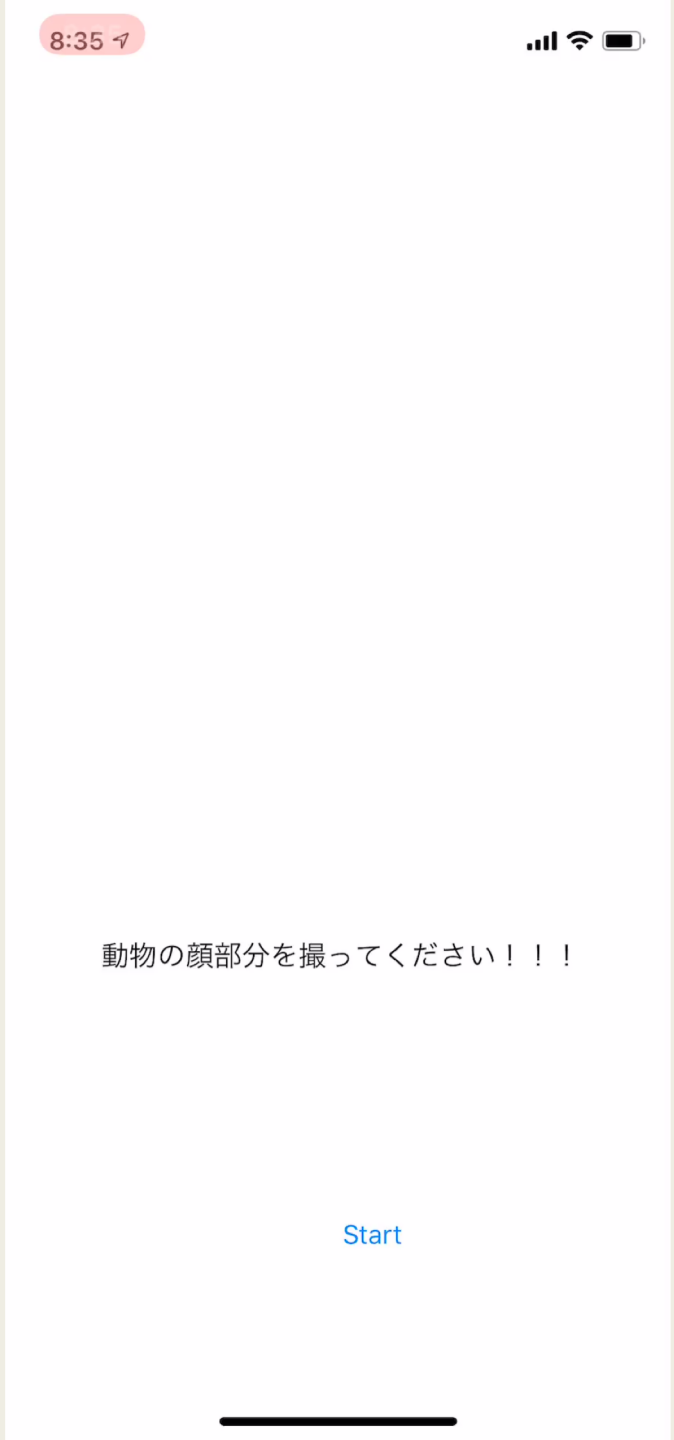
アイコン

ペイントを用いて１０秒で作成



4-2アプリケーション化内容・方法

デモンストレーション動画


8:35
動物の顔部分を撮ってください！！！
Start

５.今後の課題

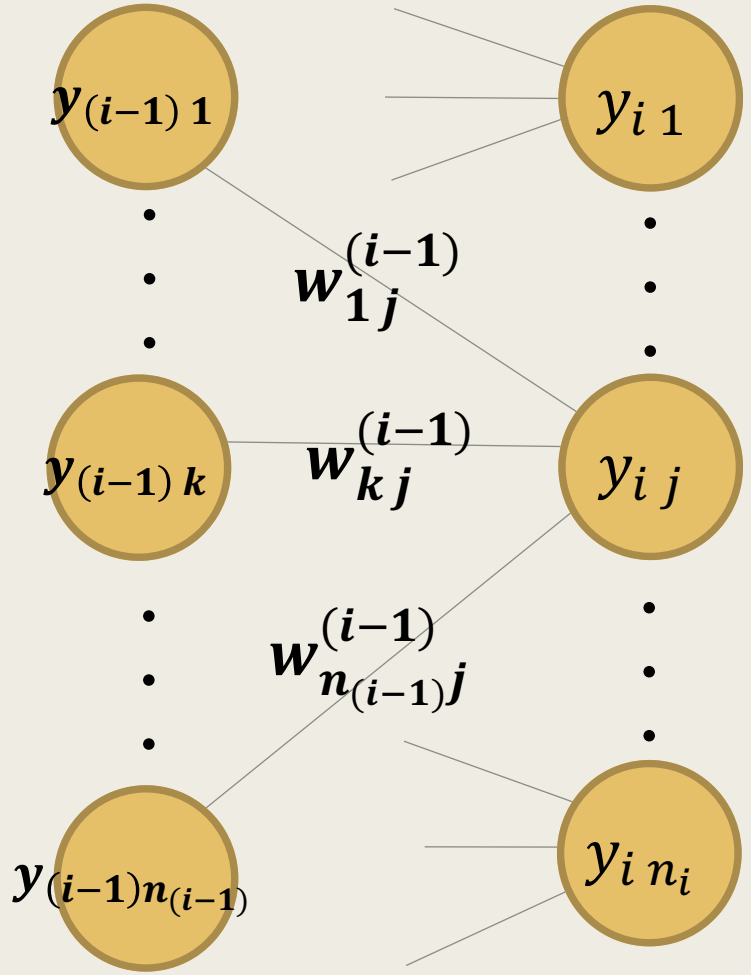
- 生成した学習データをもとに
  自動で画像を整形するものを作成
- 他の手法で学習モデルを生成する
- 学習モデルの強化
- アプリケーションでの評価

参考スライド

2-2 深層学習

ニューラルネットワーク

ラベル付きのデータのデータを
入力とし、出力値を算出



$$y_{ij} = \sum_{k=1}^{n_{(i-1)}} y_{(i-1)\,k} * w_{kj}^{(i-1)}$$

y_{ij}： i 層の j 番目のノードの値

$w_{kj}^{(i-1)}$ ：$y_{(i-1)\,k}$ と y_{ij} 間の重み

2-2 深層学習

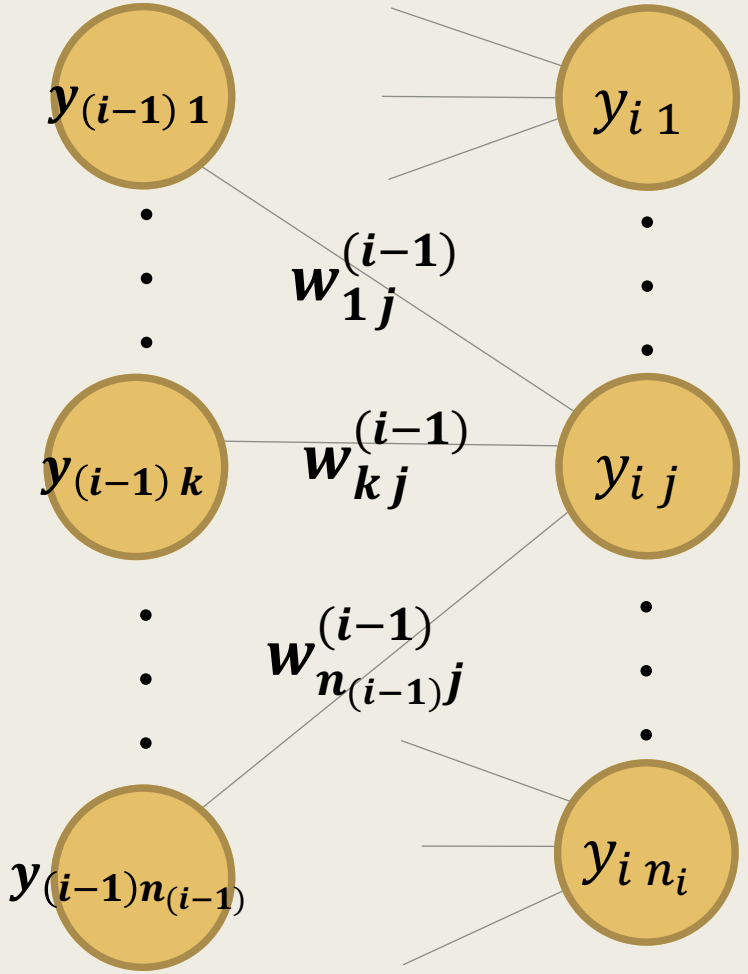
ニューラルネットワーク

ラベル付きのデータのデータを
入力とし、出力値を算出

y_m を非線形関数を用いて変換

$y_m = f(y_m)$

$f(y_m)$ ： 活性化関数



2-2 深層学習

ニューラルネットワーク

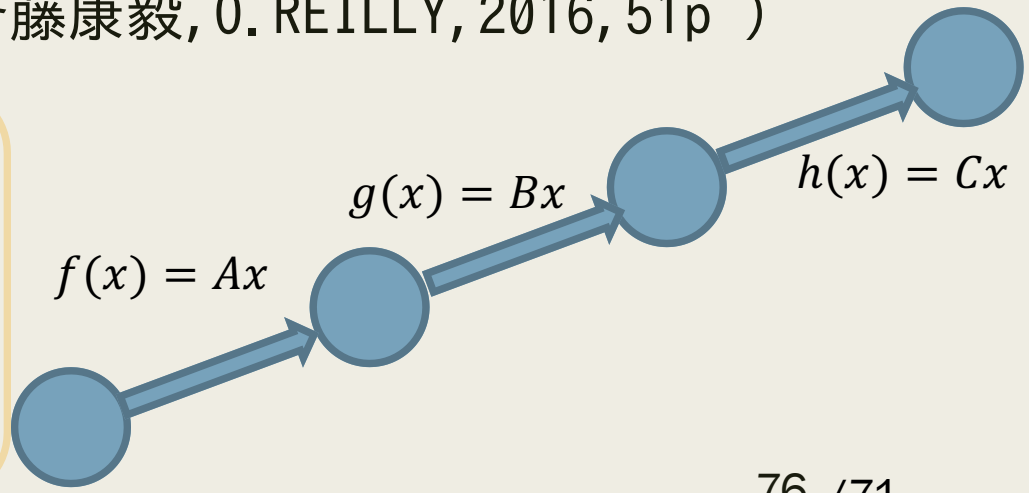
ニューラルネットワークでは活性化関数に
非線形関数を用いる必要がある
線形関数を用いると層を深くする意味がなくなる

（ゼロから作るDeep Learning,斎藤康毅,O.REILLY,2016,51p ）

$$y(x) = f(g(h(\boldsymbol{x})))$$

これは $y(x) = A(B(C\boldsymbol{x}))$
$= ABC(\boldsymbol{x})$

つまり ABC 一層で表現できる



2-2 深層学習

ニューラルネットワーク

ラベルと出力値の誤差を計算

入力データのラベルと出力層の差異を
損失関数で判定する

・二乗誤差

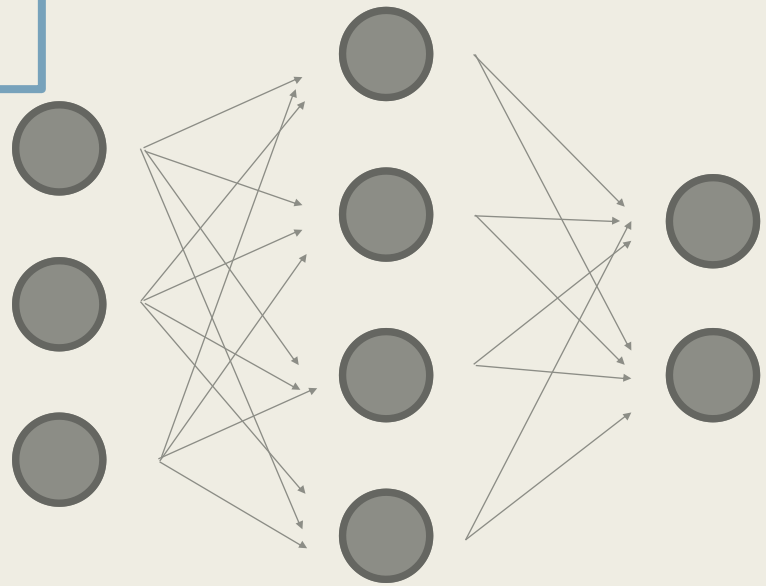
$$E = (y_p - y_t)^2$$

y_p ： 出力値
y_t ： ラベル(0,1)

2–2 深層学習

ニューラルネットワーク

ラベルと出力値の誤差が
小さくなるように重みを修正する

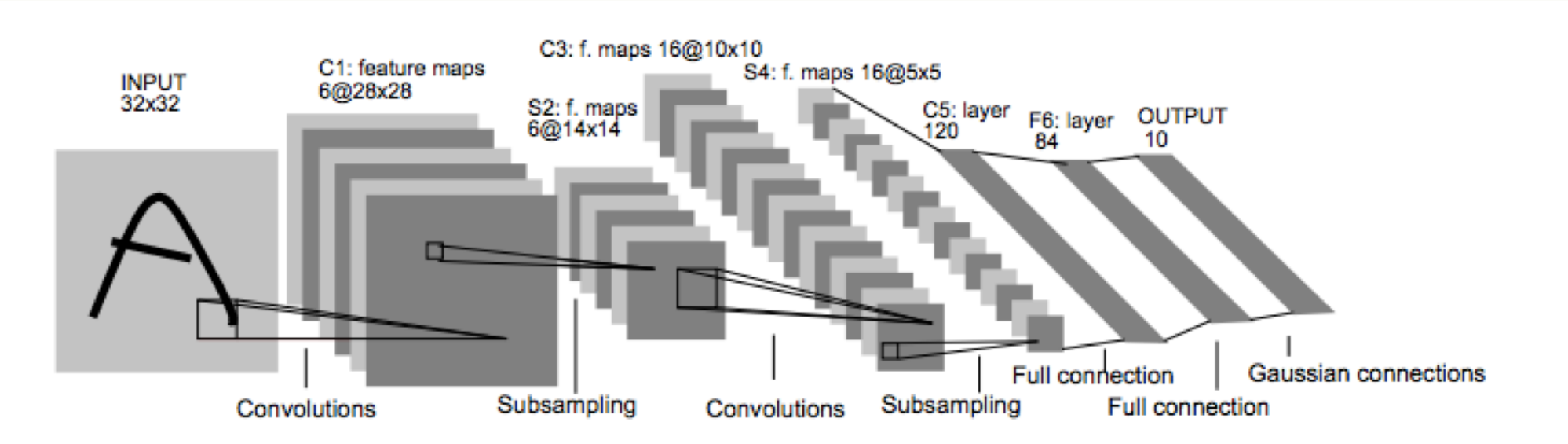


・確率的勾配降下法

$$w = w - \eta \frac{\partial E}{\partial w}$$

η :学習率
E :損失関数
w :重み

2-3 CNN(Convolutional Neural Network)



http://yann.lecun.com/exdb/publis/pdf/lecun-98.pdf
Yann, LeCun , L éon Botton, Yoshua Bengio, and Patrick Haffner
2018年12月15日参照

2-3 CNN(Convolutional Neural Network)

パディング

畳み込み層の前に入力データの周囲に固定のデータを埋めること

パディングを使用すれば空間サイズを一定のまま次層へデータを渡せる

2-3 CNN(Convolutional Neural Network)

・畳み込み層

計算

1. 入力に対してフィルタとの内積を計算する
2. 内積の値を格納し特徴マップを作成する
3. 特徴マップを次層の入力にする