

当日までドラフト

ゼロから学ぶ、ラズパイAI実装ハンズオンセミナー
～セットアップから画像認識AI実装まで～

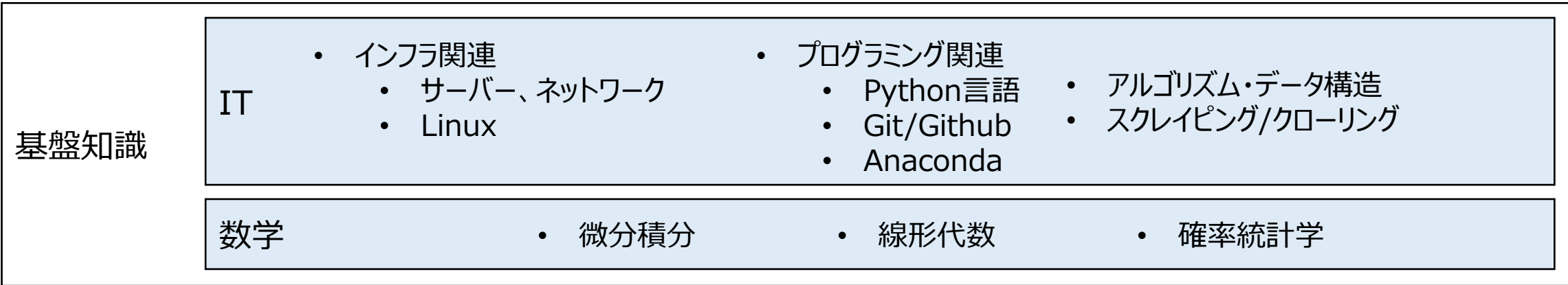
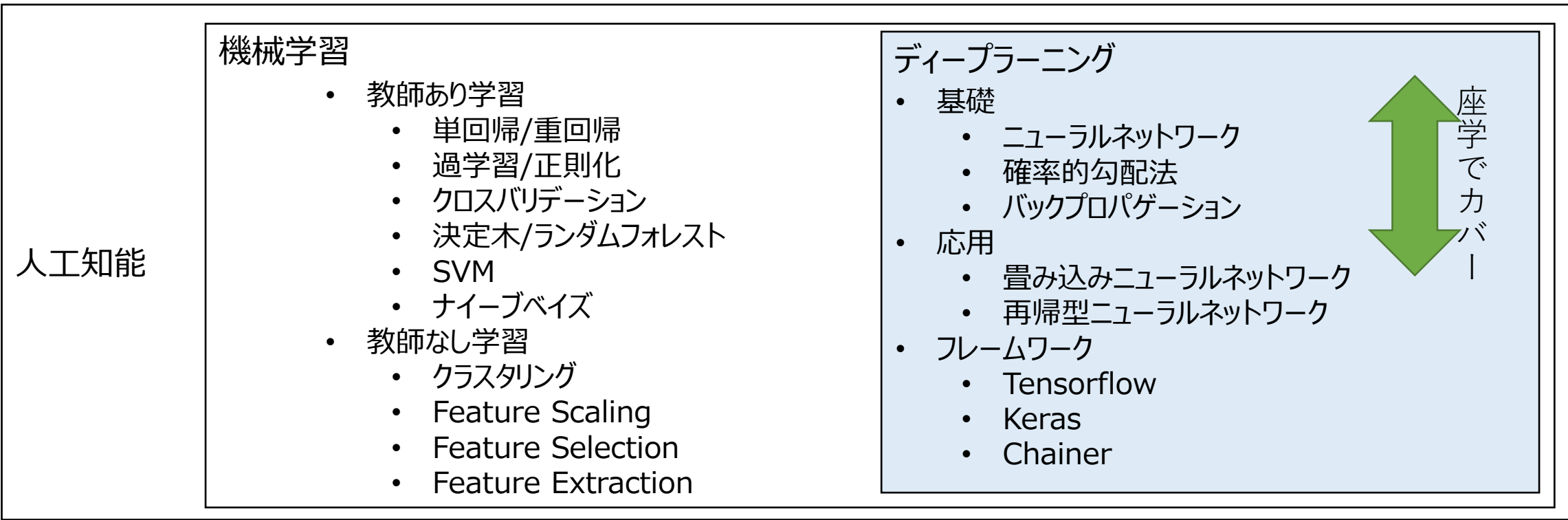
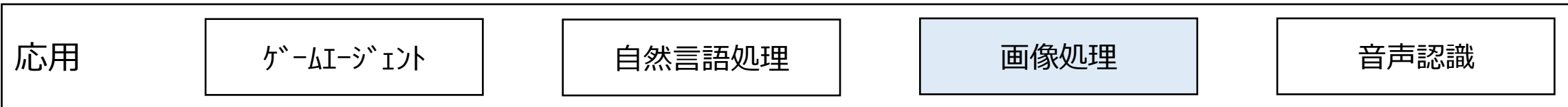
2018年2月24日

名古屋校 2011期 越智 由浩



今日の心がまえ、スタンス

- 他人の知見(ブログや記事)を参考に“真似ながら動かしてみる”を実践できるようになることを目指します
- コマンドや各種ツールなど次々に新しいことが出てきます。手順はGitHubに掲載しますので復習などにお使いください
- AIを実装して動かすまでの一通りを体感してもらうことで、どのような要素で成り立っているのか、これからこういったことを深掘りして学べばよいかを考えるきっかけになればと思います



今日の時間配分

0:00 – 2:00	<ul style="list-style-type: none">• ラズパイ基本セットアップ• カメラを使った画像配信
2:00 – 3:30	<ul style="list-style-type: none">• 座学<ul style="list-style-type: none">• ニューラルネットワーク• 手書き文字認識• 畳み込みニューラルネットワーク
3:30 – 4:30	<ul style="list-style-type: none">• 手書き文字認識システムの実装とテスト• 物体識別システムの実装とテスト
4:30 – 5:00	<ul style="list-style-type: none">• クロージング・振り返り

GitHubのガイドに沿って進んで行きましょう

https://github.com/yoshihiroo/programming-workshop/tree/master/rpi_ai_handson



今日の時間配分

0:00 – 2:00	<ul style="list-style-type: none">• ラズパイ基本セットアップ• カメラを使った画像配信
2:00 – 3:30	<ul style="list-style-type: none">• 座学<ul style="list-style-type: none">• ニューラルネットワーク• 手書き文字認識• 畳み込みニューラルネットワーク
3:30 – 4:30	<ul style="list-style-type: none">• 手書き文字認識システムの実装とテスト• 物体識別システムの実装とテスト
4:30 – 5:00	<ul style="list-style-type: none">• クロージング・振り返り

座学の部、アジェンダ

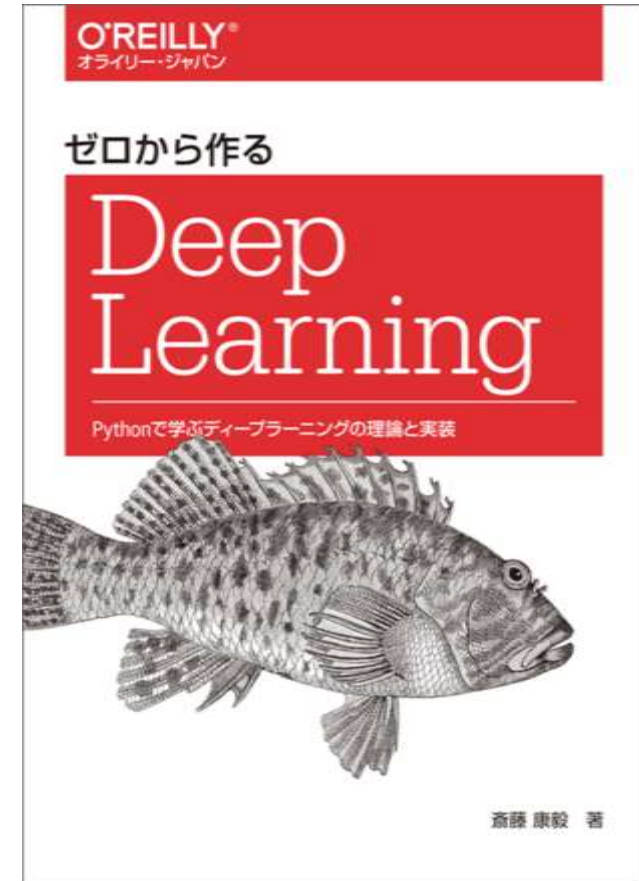
- 言葉の整理
- ニューラルネットワークの計算
- ディープラーニングによる手書き文字(数字)認識

さらに、

- 畳み込みニューラルネットワークの計算

今日の内容のベースとなる本

- 内容、ソースコード、図など、こちらの本から引用しております
- Amazonでディープラーニングで検索するとベストセラー本として簡単に見つかります
- 復習兼ねてぜひご購入をお勧めします



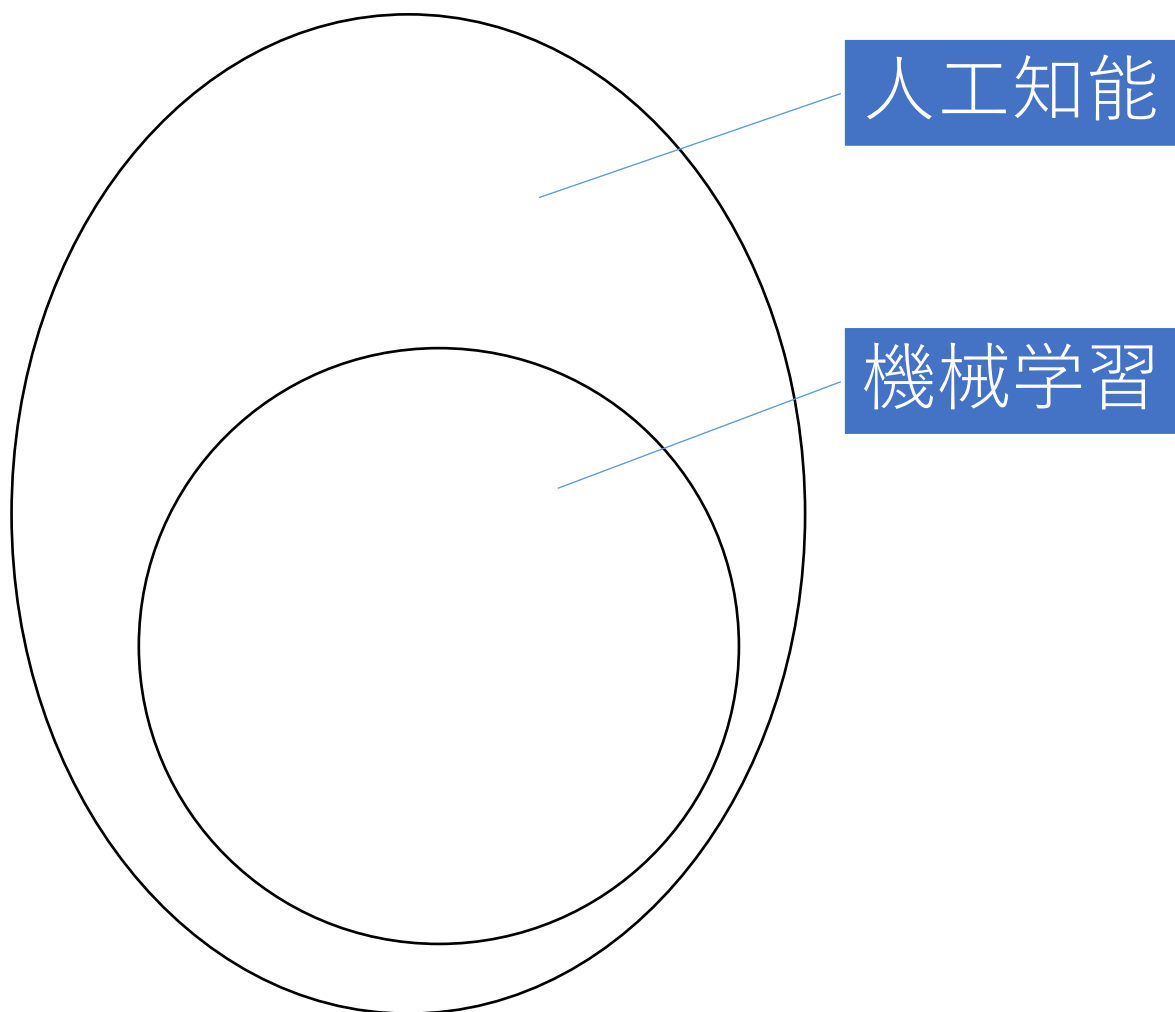
まず、言葉の整理

人工知能



「お、こいつ賢いな!」と思わせるもの、ふるまい。
またそれを探求する学問領域。

まず、言葉の整理



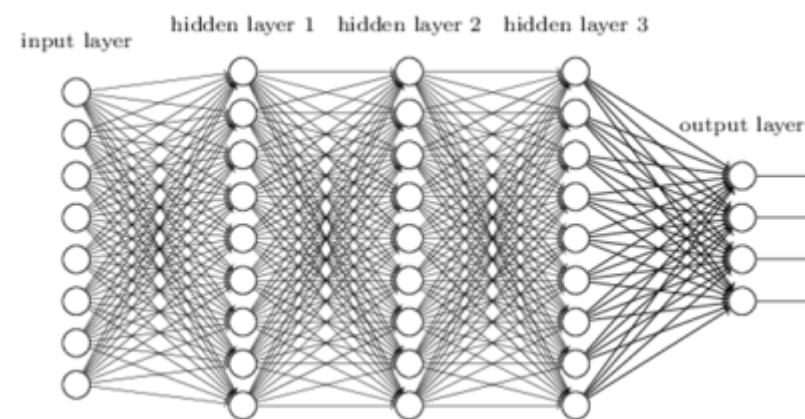
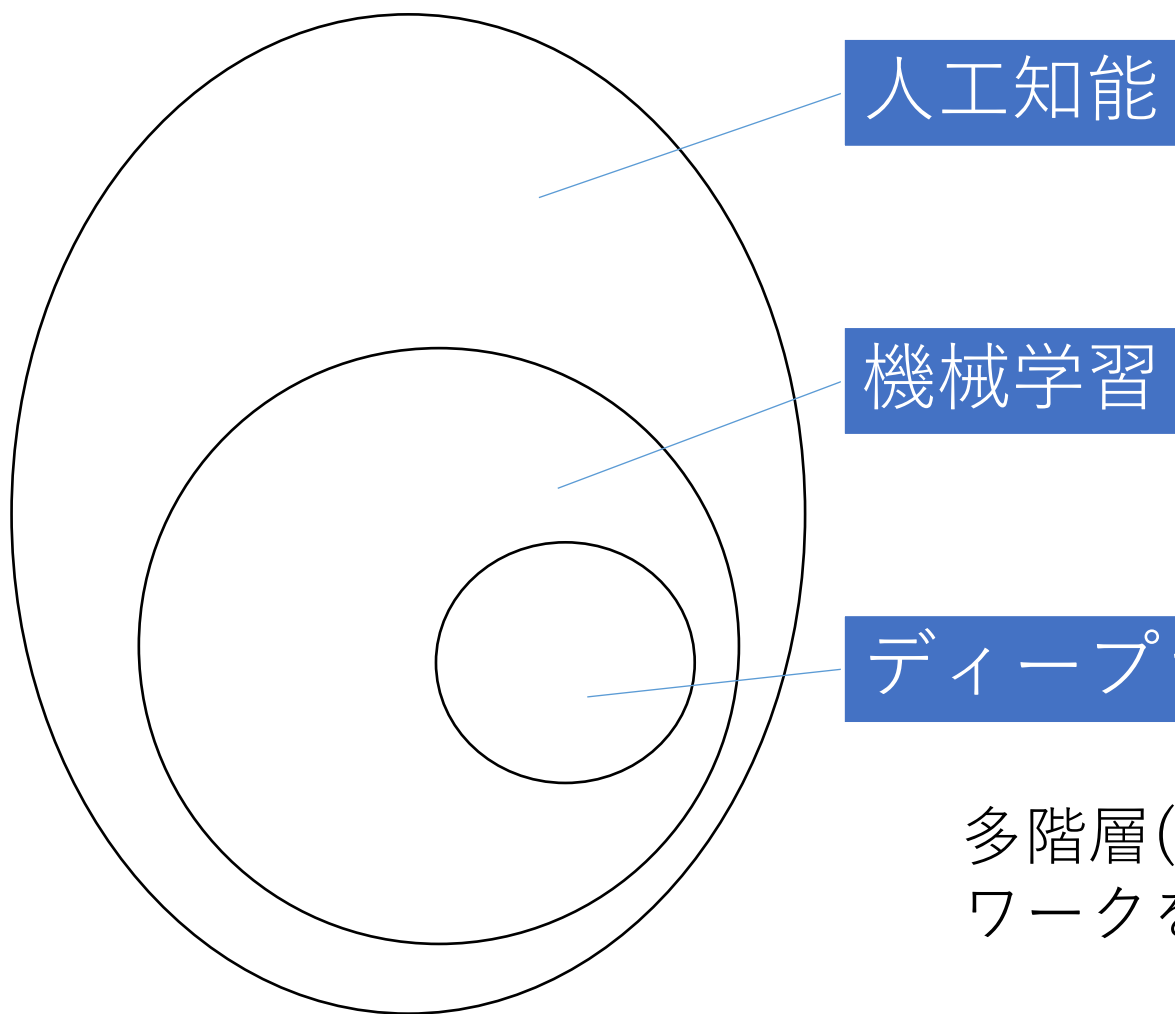
人工知能を実現する手段の一つ。
過去のデータ(知見/経験)に基づいて：

- ・ 将来を予測する
- ・ 未知のものを分類する

例)

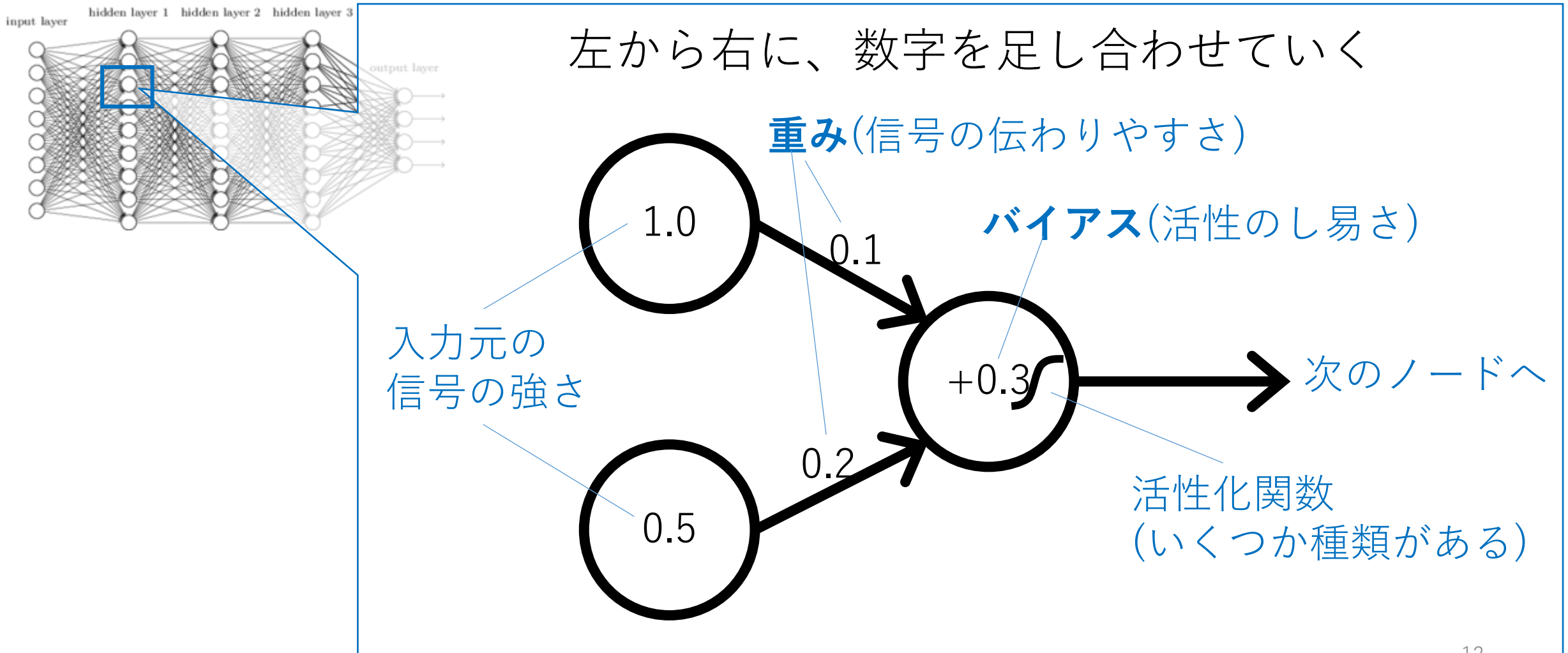
eコマースサイトのリコメンド
迷惑メールの自動振り分け

まず、言葉の整理



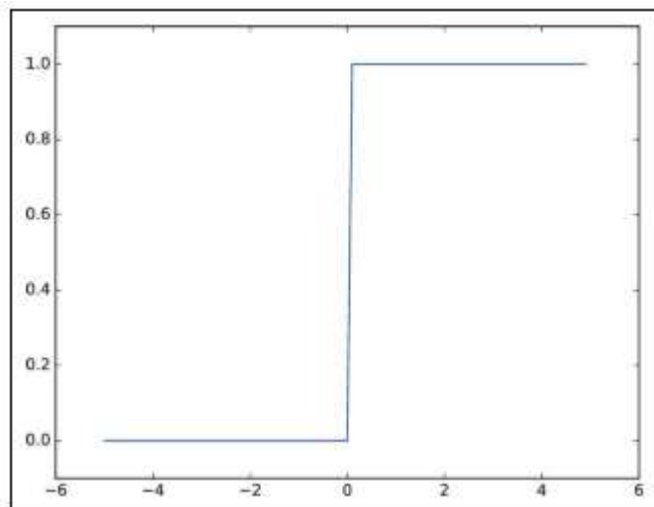
多階層(ディープな)ニューラルネットワークを用いた機械学習(ラーニング)

ニューラルネットワークの計算の考え方

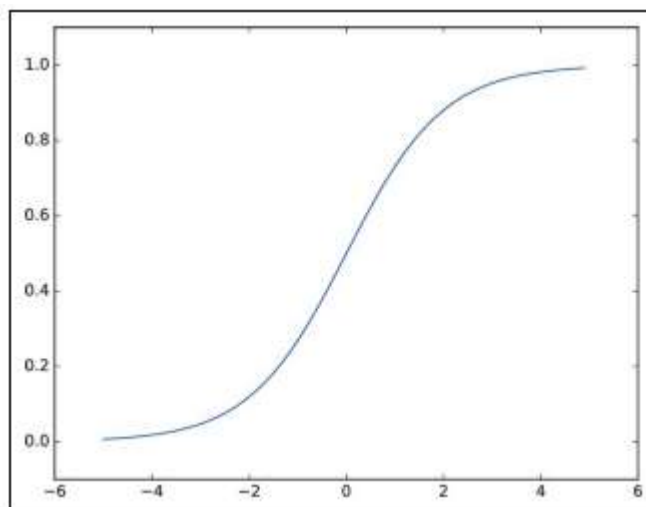


活性化関数

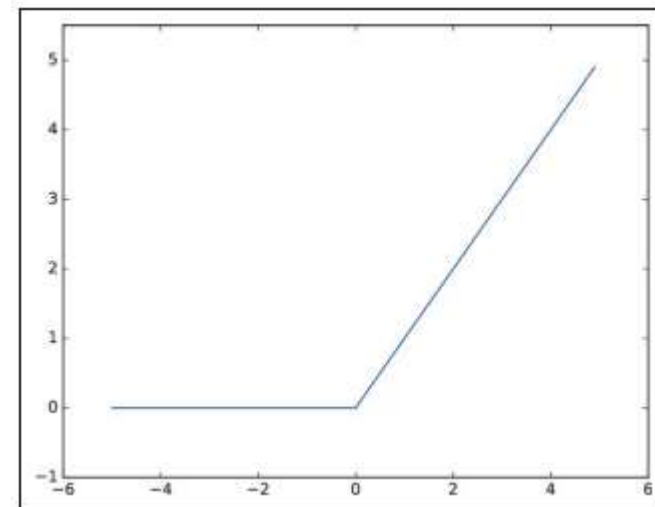
ステップ関数



シグモイド関数



ReLU関数
(Rectified Linear Unit)



出力
↑
入力の総和 + バイアス
→

$$h(x) = \begin{cases} 0 & (x \leq 0) \\ 1 & (x > 0) \end{cases}$$

$$h(x) = \frac{1}{1 + \exp(-x)}$$

$$h(x) = \begin{cases} x & (x > 0) \\ 0 & (x \leq 0) \end{cases}$$

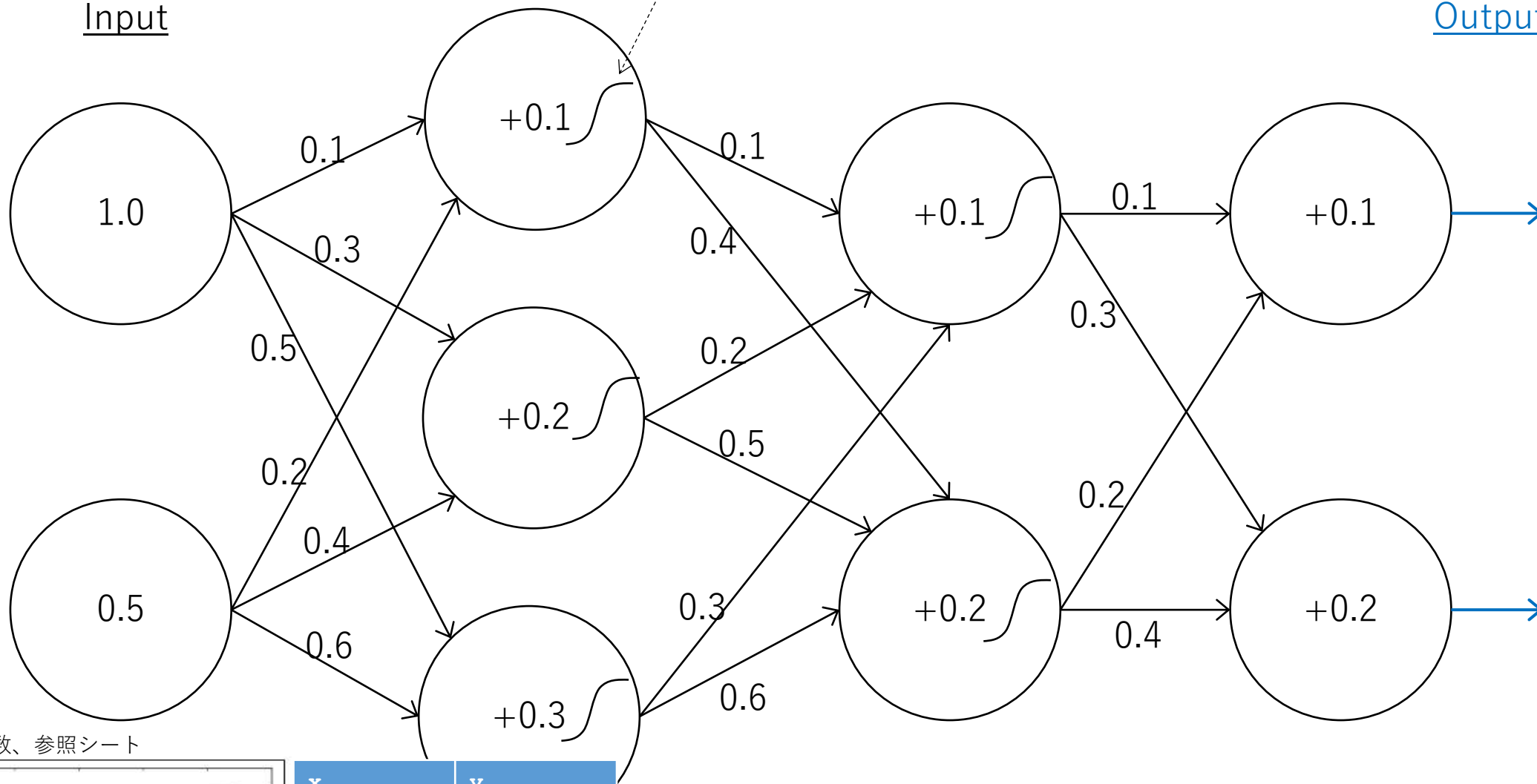
手触り感持って理解するために、
ニューラルネットワークを手計算！



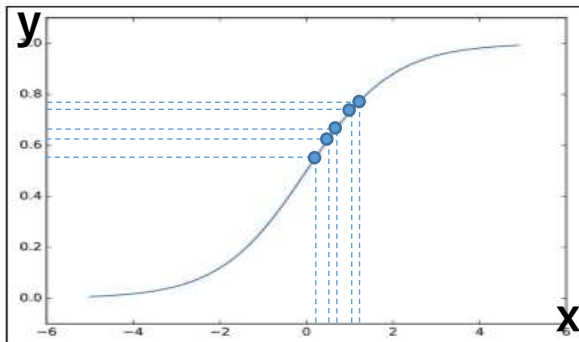
Input

Output

シグモイド関数



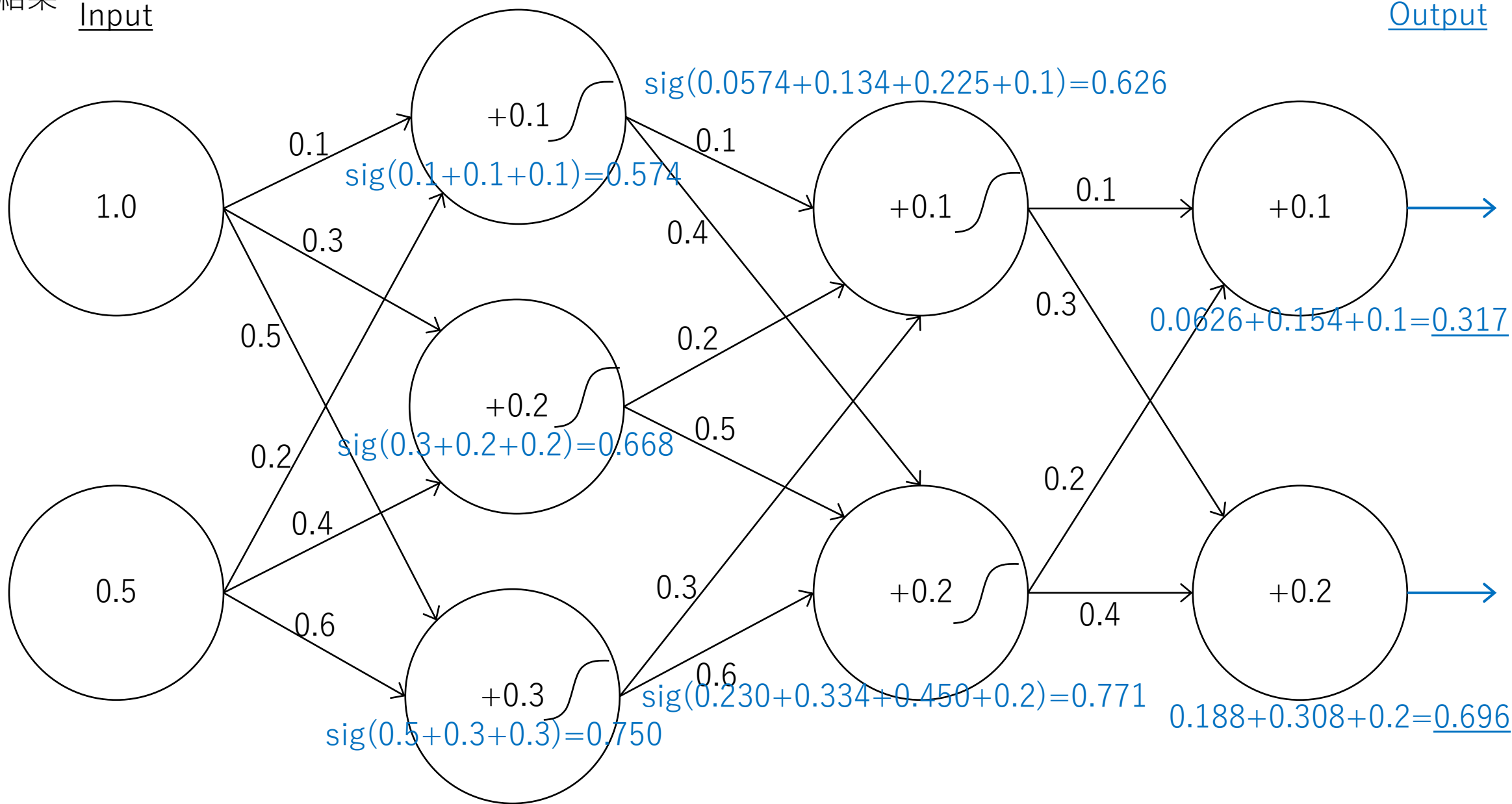
シグモイド関数、参照シート



x	y
0.3	0.574
0.516	0.626
0.7	0.668
1.1	0.750
1.214	0.771

Input

Output

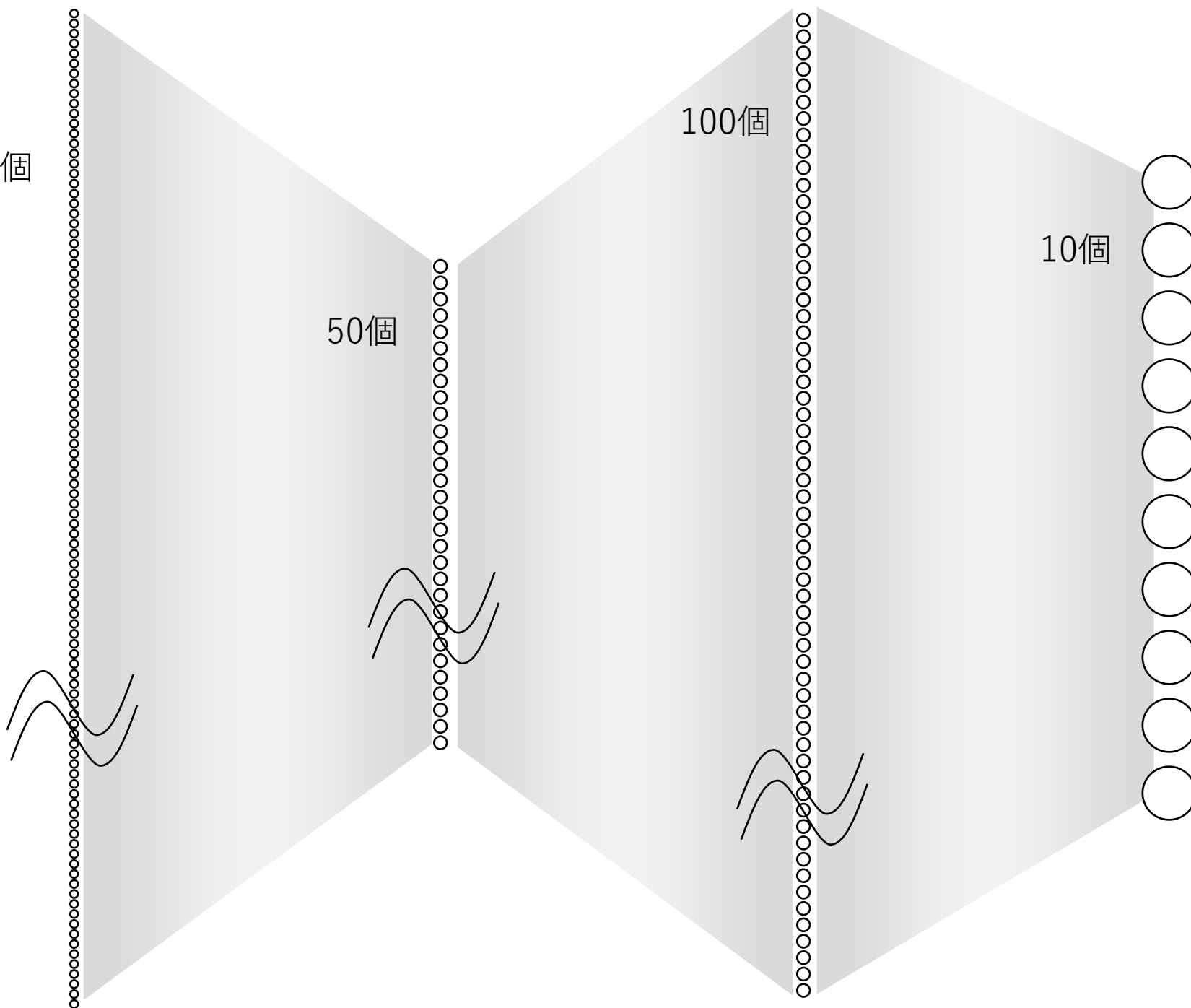


784個

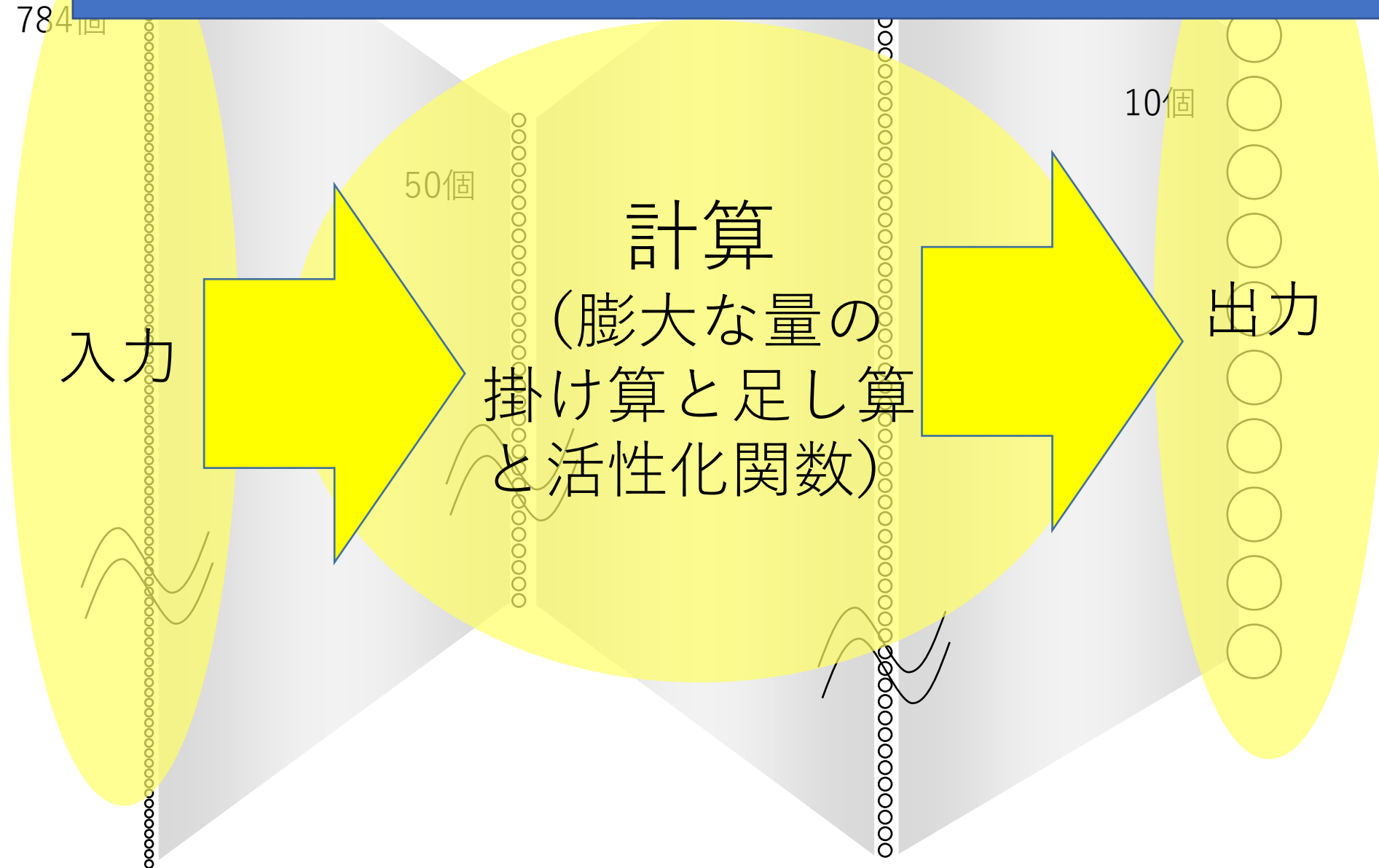
50個

100個

10個



このくらいの数のニューラルネットワークを使うと、
手書きの文字認識ができちゃいます。



手計算じゃムリ！



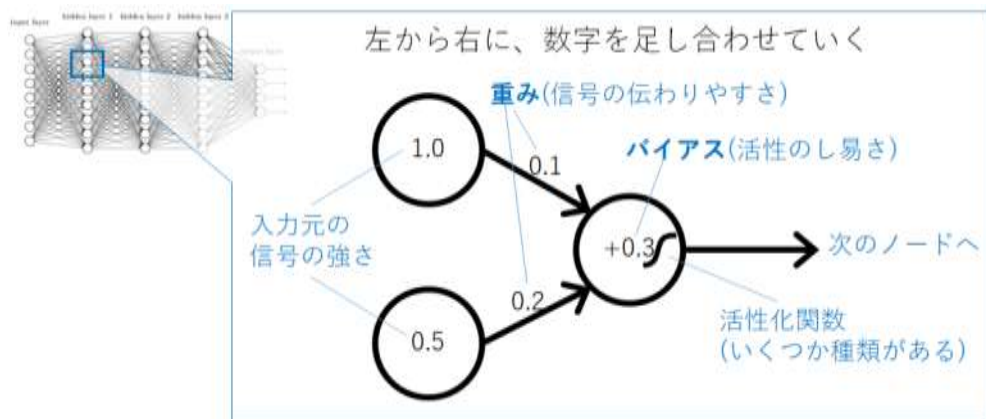
計算
(膨大な量の
掛け算と足し算
と活性化関数)

単調な計算の繰り返しは
コンピューターの得意技。
プログラムを作って計算
させちゃえばいい！

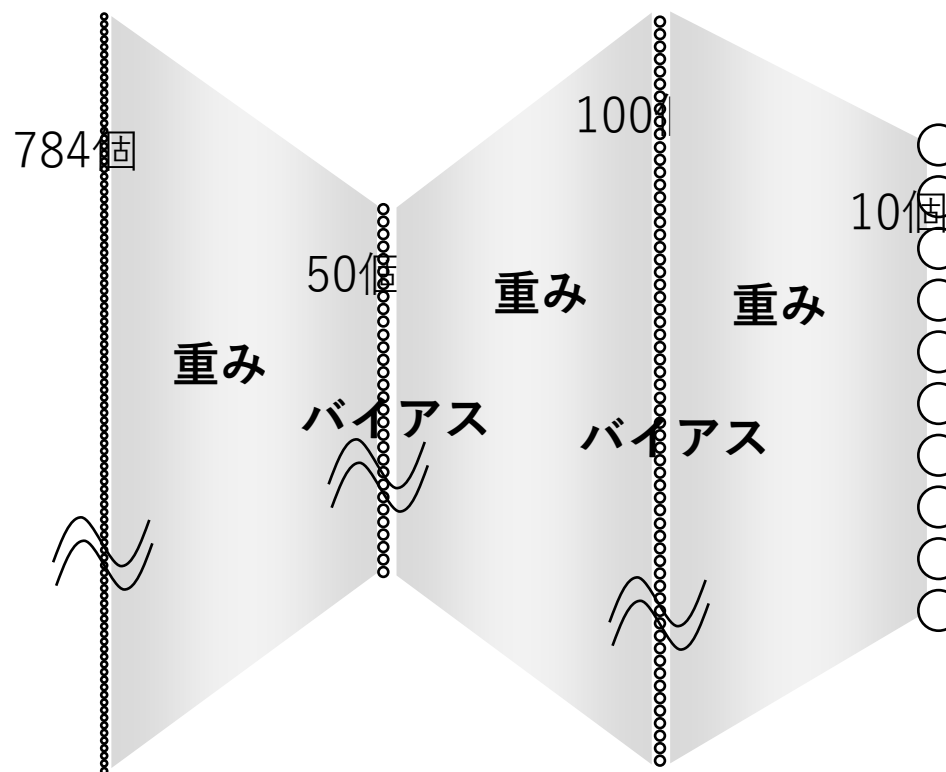


パラメータとハイパーパラメータ

先ほど見せた図



パラメータとは：右図でいう**重み**と**バイアス**



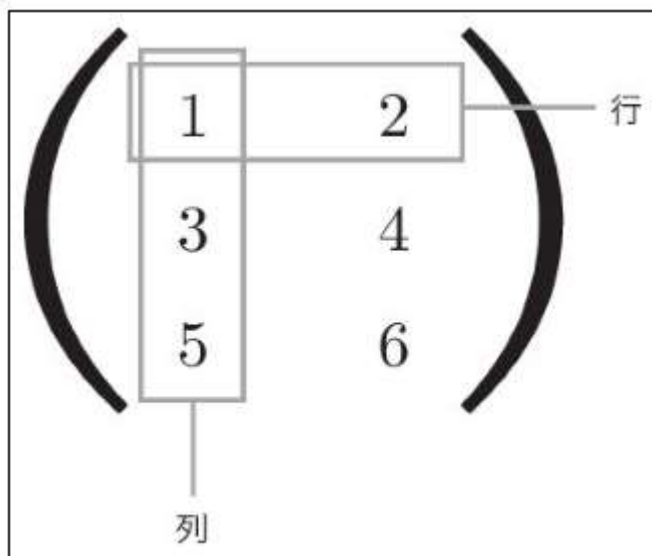
ハイパーパラメータとは： ネットワークを何層にするか、層あたりのノードの数をいくつにするかなど、ネットワークのデザインを決める



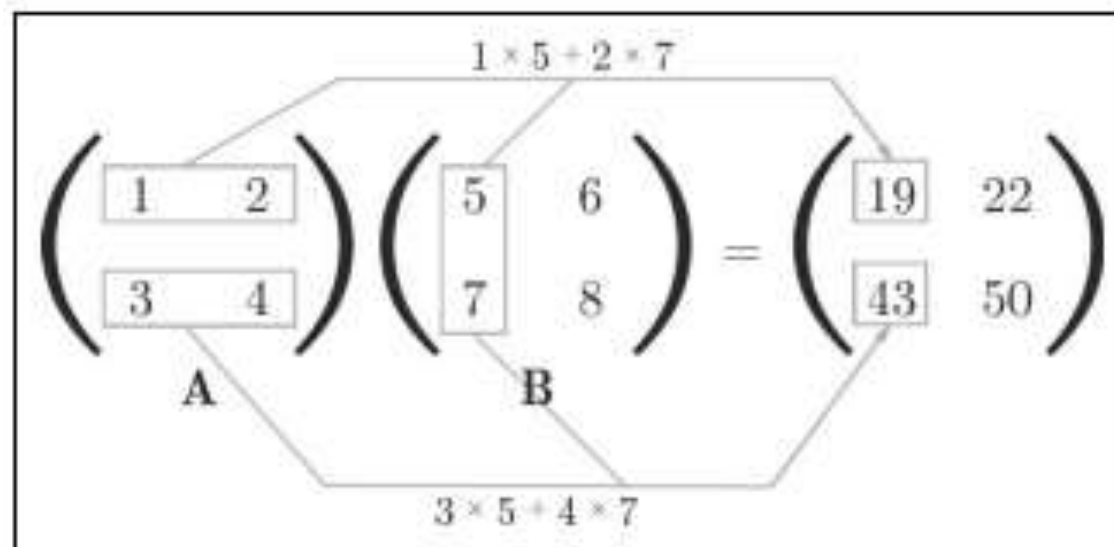
数学の便利な道具 – 行列、内積

なぜ便利かは後ほどわかる

3 × 2の行列の例



行列の内積の例



行列計算に慣れ親しむ

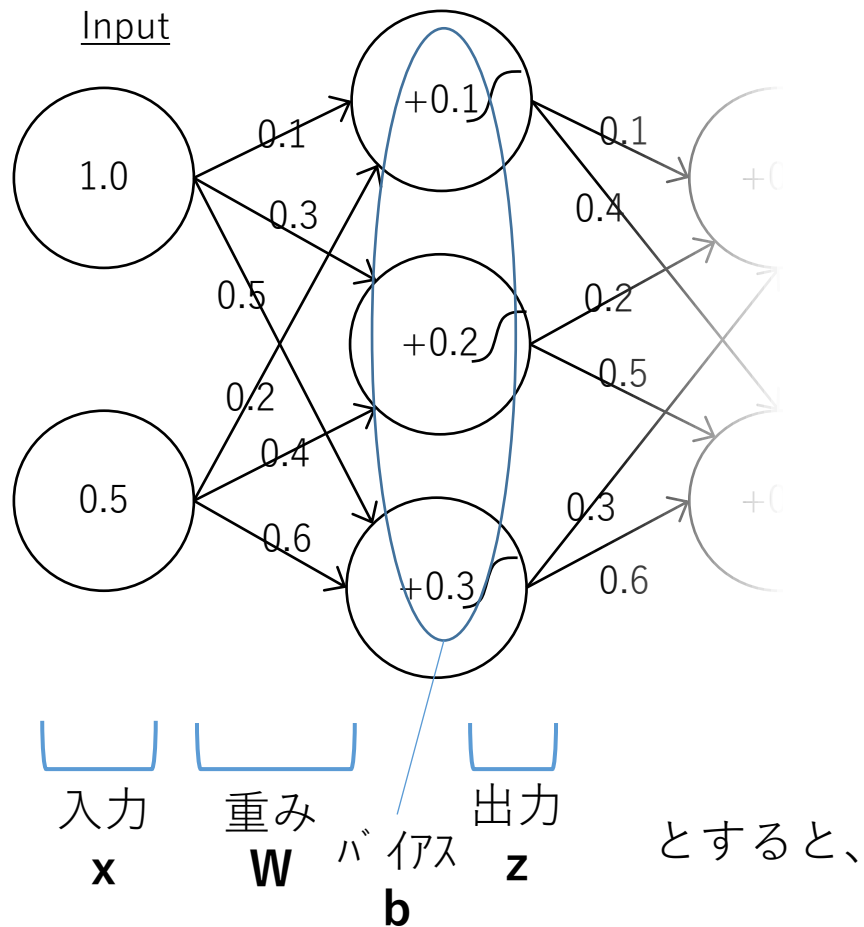
$$\begin{pmatrix} 1 & 3 \\ 5 & 7 \\ 9 & 11 \end{pmatrix} \cdot \begin{pmatrix} 11 \\ 22 \end{pmatrix} = \begin{pmatrix} 77 \\ 209 \\ 341 \end{pmatrix}$$

$$(11 \ 22) \cdot \begin{pmatrix} 1 & 3 \\ 5 & 7 \\ 9 & 11 \end{pmatrix} = \boxed{?}$$

ニューラルネットワークを行列計算で表現してみる

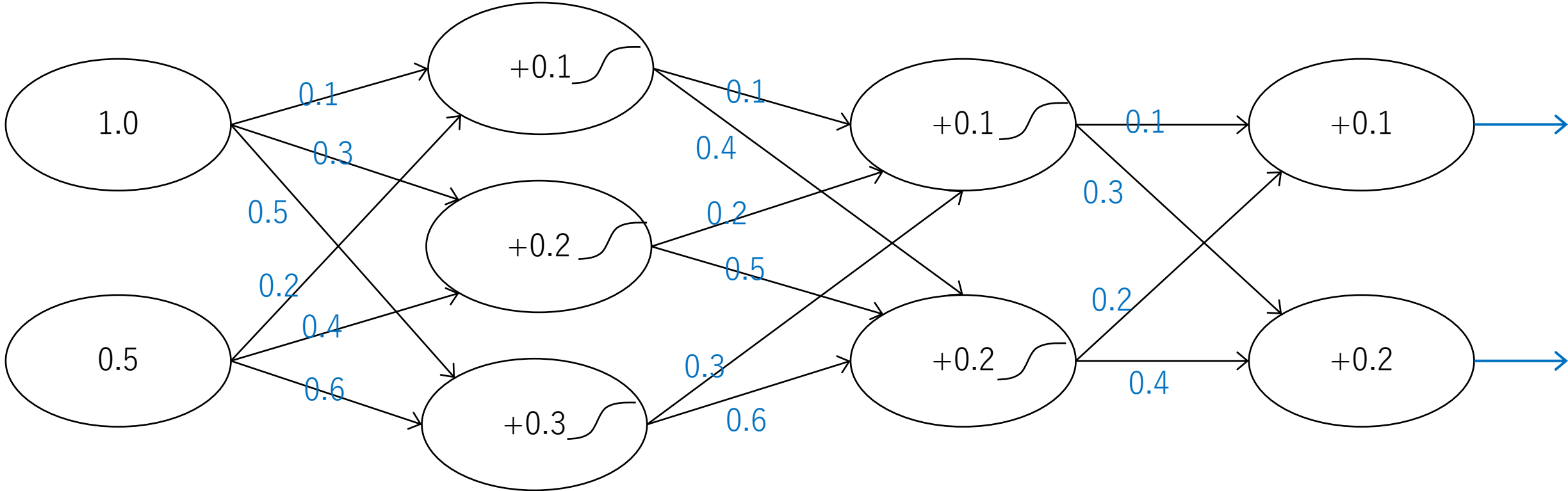
$\text{sig}()$ …シグモイド関数

先ほどの模型の前半部分



$$\begin{aligned} z &= \text{sig}(W \cdot x + b) \\ &= \text{sig}\left(\begin{pmatrix} 0.1 & 0.2 \\ 0.3 & 0.4 \\ 0.5 & 0.6 \end{pmatrix} \cdot \begin{pmatrix} 1.0 \\ 0.5 \end{pmatrix} + \begin{pmatrix} 0.1 \\ 0.2 \\ 0.3 \end{pmatrix}\right) \\ &= \text{sig}\left(\begin{pmatrix} 0.2 \\ 0.5 \\ 0.8 \end{pmatrix} + \begin{pmatrix} 0.1 \\ 0.2 \\ 0.3 \end{pmatrix}\right) \\ &= \text{sig}\left(\begin{pmatrix} 0.3 \\ 0.7 \\ 1.1 \end{pmatrix}\right) = \begin{pmatrix} 0.574 \\ 0.668 \\ 0.750 \end{pmatrix} \end{aligned}$$

先ほどのニューラルネットの例



x	W1	b1	z1	W2	b2	z2	W3	b3	z3
$\begin{pmatrix} 1.0 \\ 0.5 \end{pmatrix}$	$\begin{pmatrix} 0.1 & 0.2 \\ 0.3 & 0.4 \\ 0.5 & 0.6 \end{pmatrix}$	$\begin{pmatrix} 0.1 \\ 0.2 \\ 0.3 \end{pmatrix}$	$\begin{pmatrix} ? \\ ? \\ ? \end{pmatrix}$	$\begin{pmatrix} 0.1 & 0.2 & 0.3 \\ 0.4 & 0.5 & 0.6 \end{pmatrix}$	$\begin{pmatrix} 0.1 \\ 0.2 \end{pmatrix}$	$\begin{pmatrix} ? \\ ? \end{pmatrix}$	$\begin{pmatrix} 0.1 & 0.2 \\ 0.3 & 0.4 \end{pmatrix}$	$\begin{pmatrix} 0.1 \\ 0.2 \end{pmatrix}$	$\begin{pmatrix} ? \\ ? \end{pmatrix}$

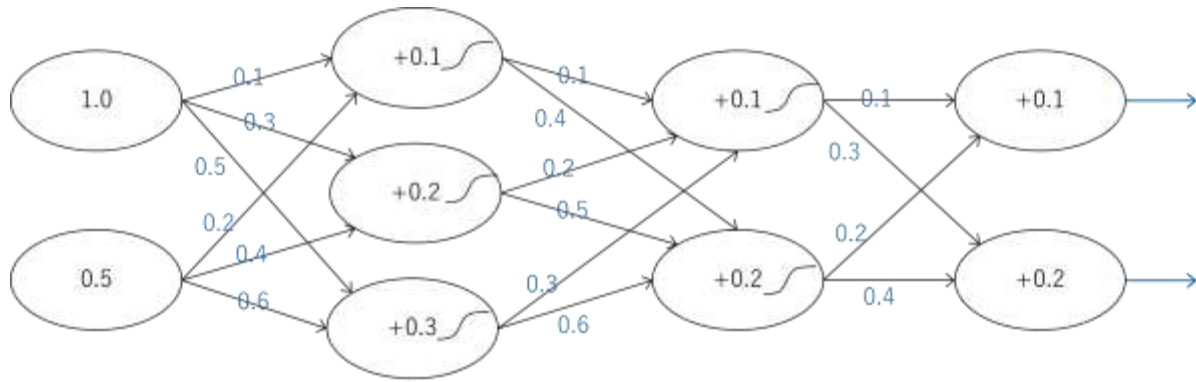
$z1 = \text{sig}(W1 \cdot x + b1)$

$z2 = \text{sig}(W2 \cdot z1 + b2)$

$z3 = W3 \cdot z2 + b3$

sig() …シグモイド関数

先ほどやったやつ



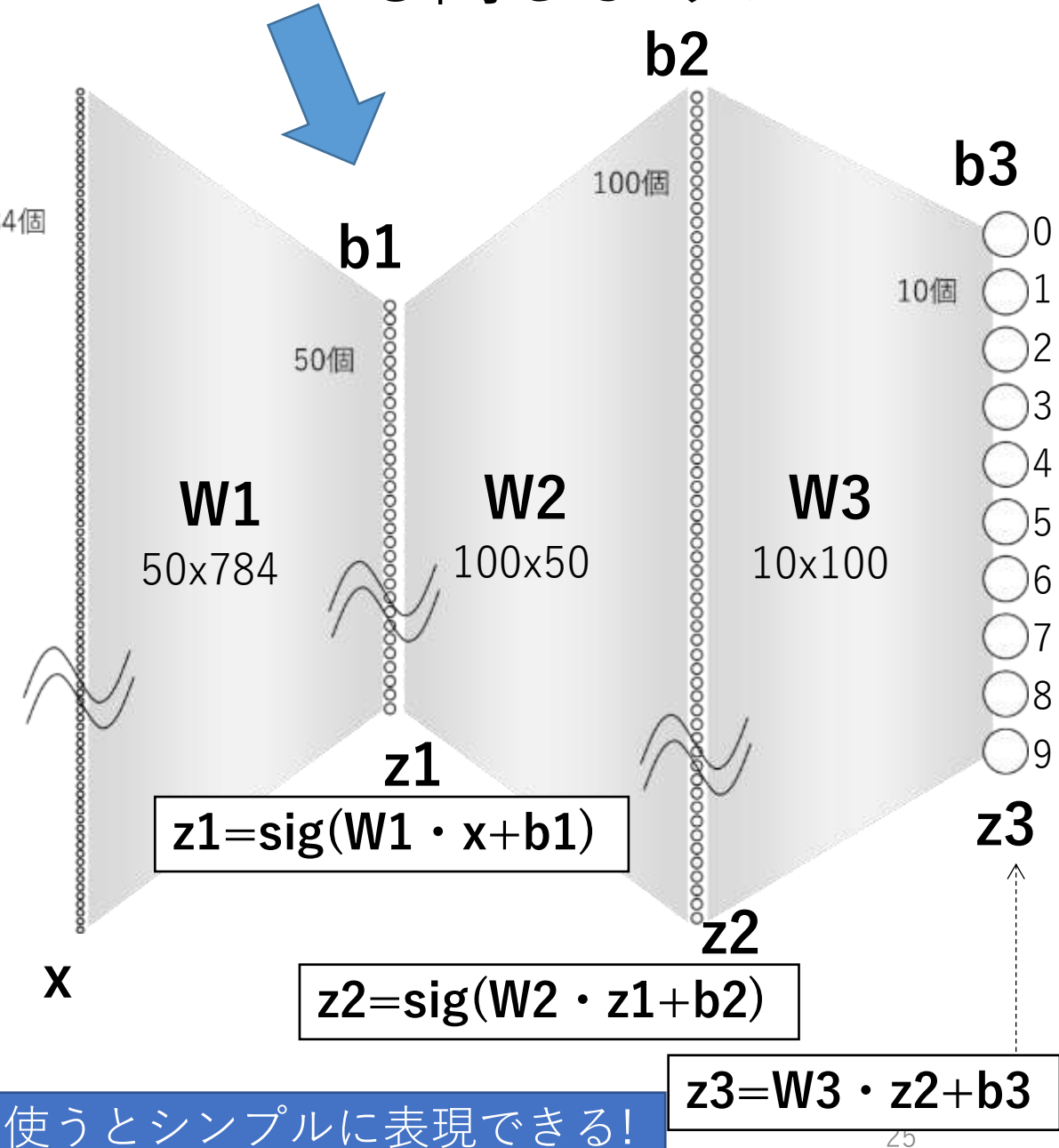
$$\begin{array}{c}
 \mathbf{x} \\
 \begin{pmatrix} 1.0 \\ 0.5 \end{pmatrix}
 \end{array}
 \quad
 \begin{array}{c}
 \mathbf{W1} \\
 \begin{pmatrix} 0.1 & 0.2 \\ 0.3 & 0.4 \\ 0.5 & 0.6 \end{pmatrix}
 \end{array}
 \quad
 \begin{array}{c}
 \mathbf{b1} \\
 \begin{pmatrix} 0.1 \\ 0.2 \\ 0.3 \end{pmatrix}
 \end{array}
 \quad
 \begin{array}{c}
 \mathbf{z1} \\
 \begin{pmatrix} ? \\ ? \\ ? \end{pmatrix}
 \end{array}
 \quad
 \begin{array}{c}
 \mathbf{W2} \\
 \begin{pmatrix} 0.1 & 0.2 & 0.3 \\ 0.4 & 0.5 & 0.6 \end{pmatrix}
 \end{array}
 \quad
 \begin{array}{c}
 \mathbf{b2} \\
 \begin{pmatrix} 0.1 \\ 0.2 \end{pmatrix}
 \end{array}
 \quad
 \begin{array}{c}
 \mathbf{z2} \\
 \begin{pmatrix} ? \\ ? \end{pmatrix}
 \end{array}
 \quad
 \begin{array}{c}
 \mathbf{W3} \\
 \begin{pmatrix} 0.1 & 0.2 \\ 0.3 & 0.4 \end{pmatrix}
 \end{array}
 \quad
 \begin{array}{c}
 \mathbf{b3} \\
 \begin{pmatrix} 0.1 \\ 0.2 \end{pmatrix}
 \end{array}
 \quad
 \begin{array}{c}
 \mathbf{z3} \\
 \begin{pmatrix} ? \\ ? \end{pmatrix}
 \end{array}$$

$$\mathbf{z1} = \text{sig}(\mathbf{W1} \cdot \mathbf{x} + \mathbf{b1})$$

$$\mathbf{z2} = \text{sig}(\mathbf{W2} \cdot \mathbf{z1} + \mathbf{b2})$$

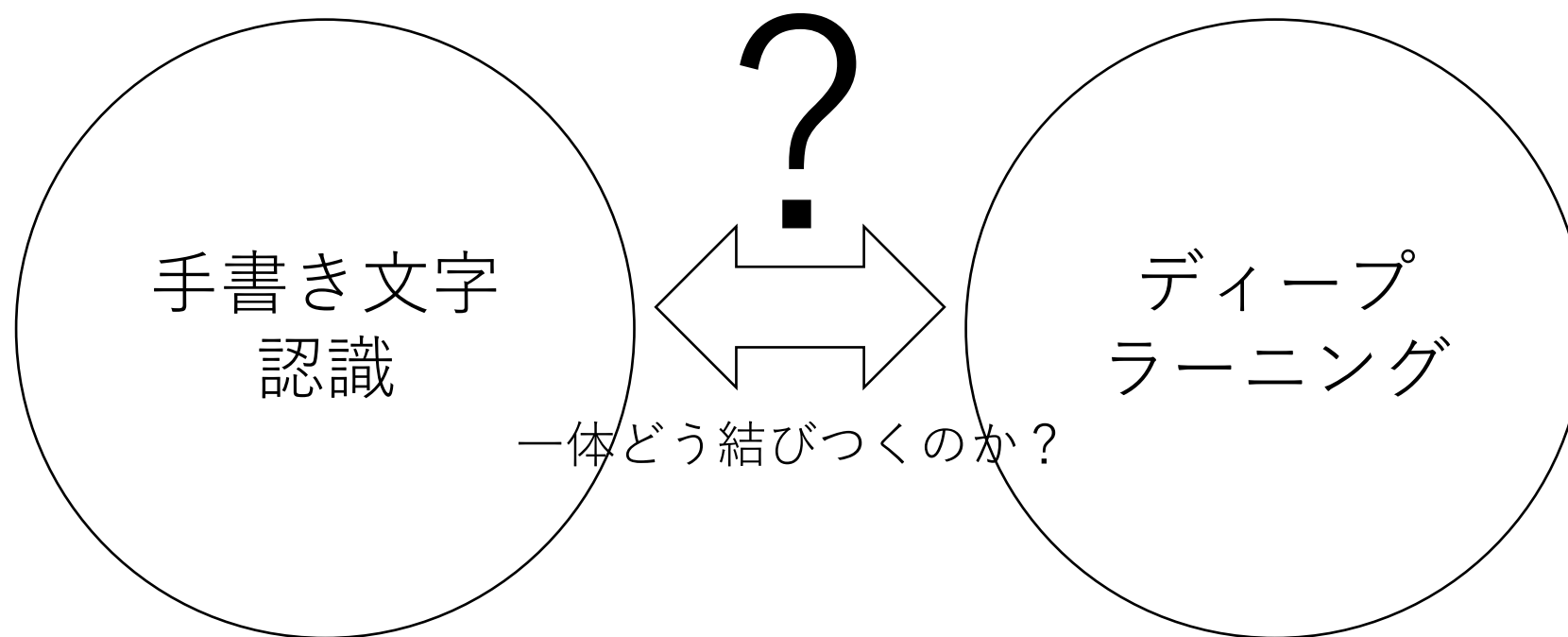
$$\mathbf{z3} = \mathbf{W3} \cdot \mathbf{z2} + \mathbf{b3}$$

こいつも同じように...



層やノードの数が増えても、行列を使うとシンプルに表現できる!

手書き文字(数字)認識をさせてみる



二つのフェーズ - 学習と推論

学習

Training

正解がわかっている既知
のデータを使って※学習
させるフェーズ

※教師あり学習

学習データ
(正解ラベル付き)

label = 5



bel = 1

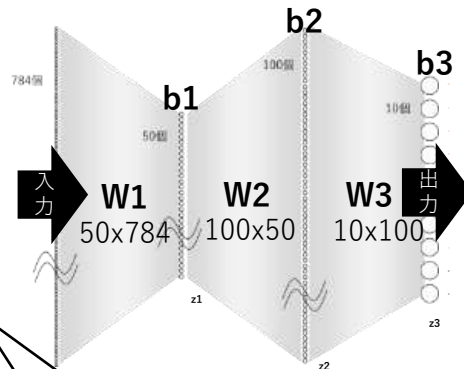


...

これは
「1」だぞ

これは
「3」だぞ

これは
「5」だぞ



推論

Inference

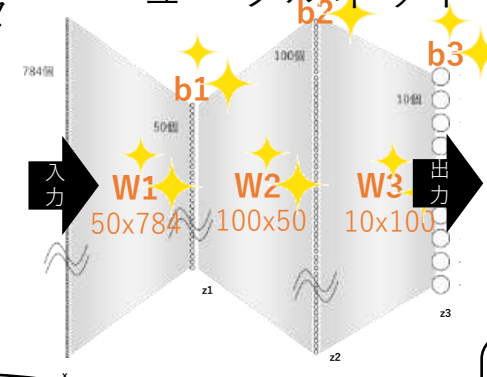
未知のデータに対して結
果を予測・分類させる
フェーズ

学習済の
ニューラルネット

未知のデータ



これは
な〜んだ？



推論の結果

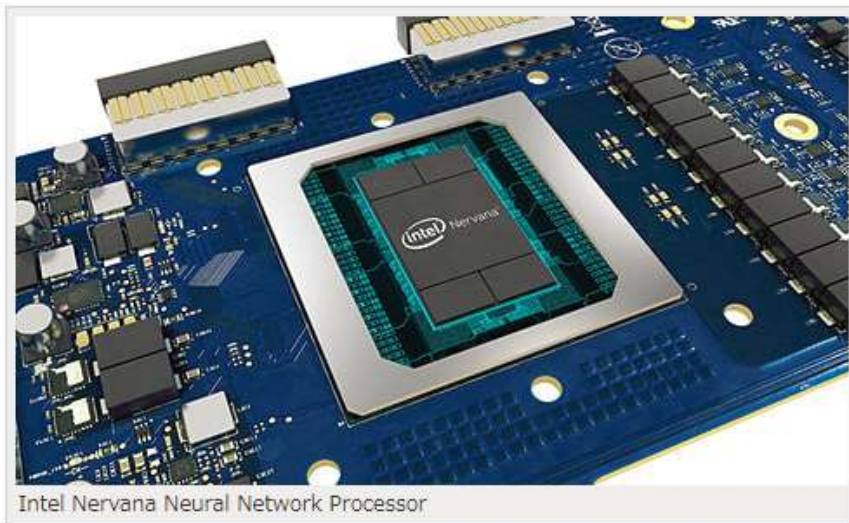
90%ノカクリツデ
3ダトオモイマス..

Intel、深層学習用プロセッサ「Nervana NNP」

～今年中に49量子ビットの量子演算チップ製造も予告

佐藤 岳大 2017年10月18日 14:41

ツイート リスト いいね! 223 シェア B! 33 Pocket 86



米Intelは17日、業界初となる人工知能プロセッサ「Intel Nervana ニューラルネットワークプロセッサ(以下Nervana NNP)」を、2017年中に提供開始することを発表した。

<https://pc.watch.impress.co.jp/docs/news/1086773.html>

Intel、世界初のスティックPC型AIアクセラレータ

～PCにつないで推論処理を高速実行、79ドル

佐藤 岳大 2017年7月21日 14:08

ツイート リスト いいね! 930 シェア B! 65 Pocket 176



米Intelは20日(米国時間)、スティック型人工知能アクセラレータ「Movidius Neural Compute Stick」を発売した。価格は79ドルで、一部のディストリビュータおよび、7月22日～25日の期間でハワイ ホノルルにて開催されている「Computer Vision and Pattern Recognition (CVPR)」カンファレンスで購入できる。

<https://pc.watch.impress.co.jp/docs/news/1071787.html>

データを用意する



MNIST手書き文字データ

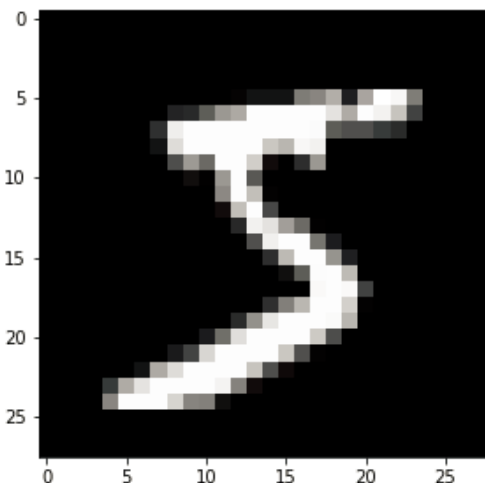
<http://yann.lecun.com/exdb/mnist/>

ディープラーニングのアルゴリズムを評価(ベンチマーク)するためによく使われる

- 60,000の学習用データ
- 10,000の検証用データ

共に正解ラベルを持つが、学習用とは別のデータを用いて推論させることで、精度を評価できる

Ground Truth : 5



28x28=784個の格子(ピクセル)ごとに0~1の255段階の値で明るさを示すことで手書き文字を数値として表現

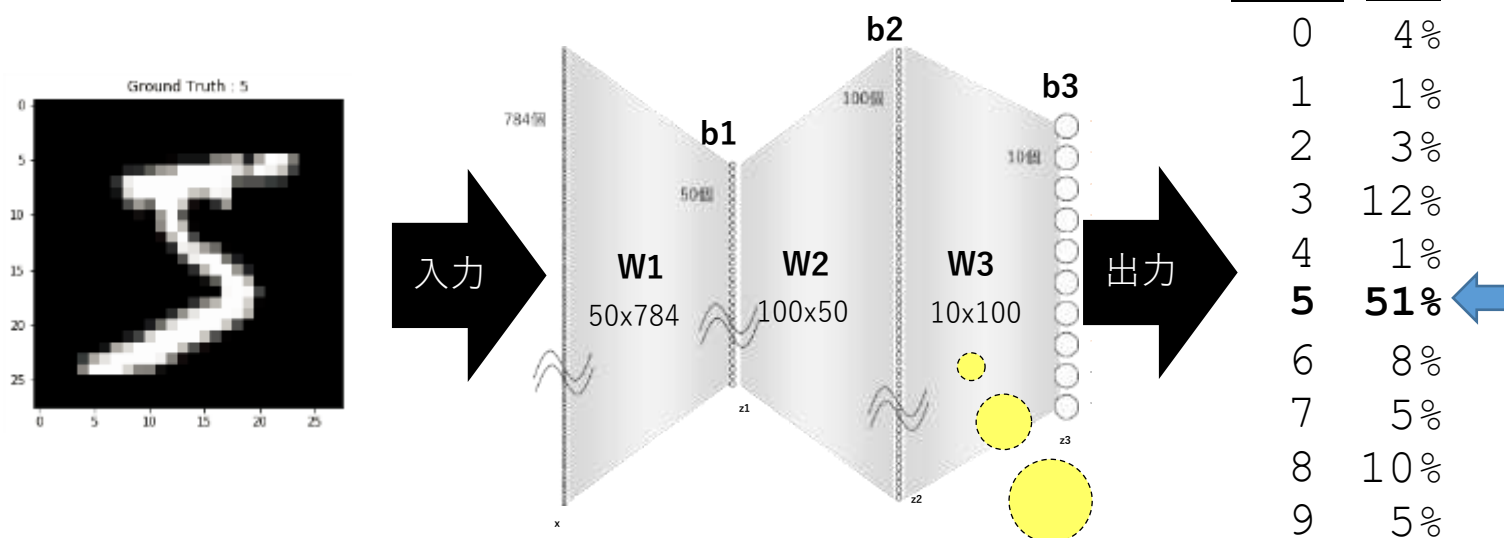
ちょい脱線

MNISTデータの他にも、10カテゴリーに分類された画像データCIFAR10や、さらに多くの分類を持つImageNetなど、多数のベンチマーク用データが公開されている

参照) Open Data for Deep Learning (<https://deeplearning4j.org/opendata>)

ディープラーニングにおける学習とは

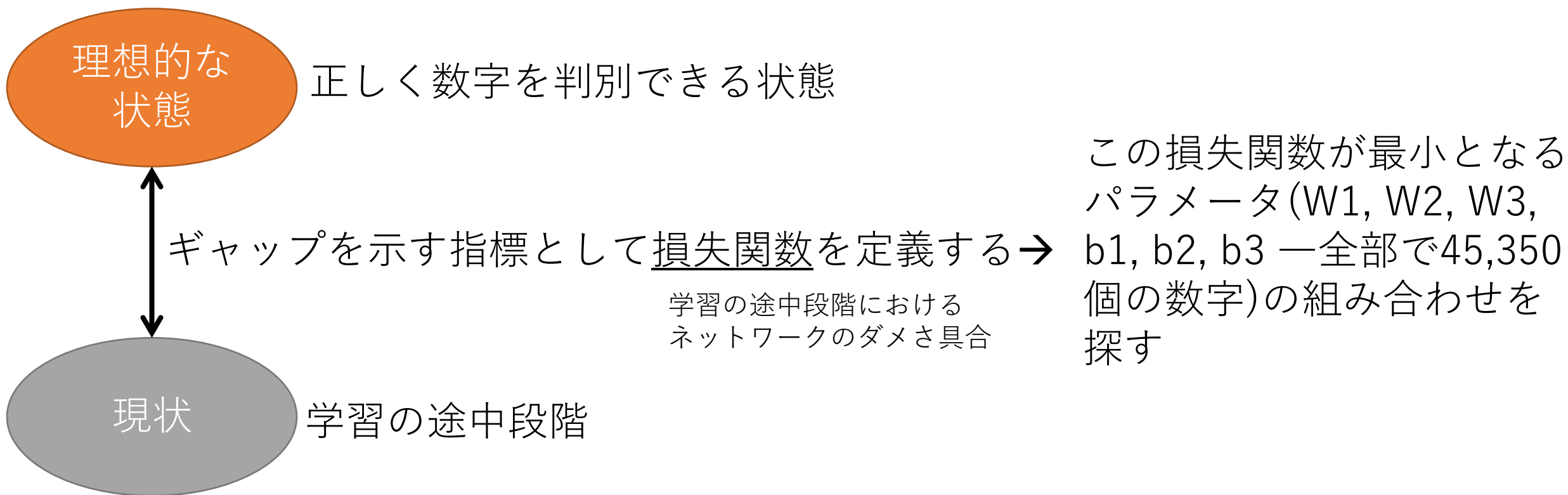
数値化された手書き文字を、ニューラルネットに食わせ・・



パラメータ ($W1, W2, W3, b1, b2, b3$ — 全部で45,350個の数字) を少しずつ変えながら、正解ラベル箇所が大きな確率を示すように、**絶妙なパラメータの組み合わせ**を探すプロセス

パラメータを調整
してるイメージ・・

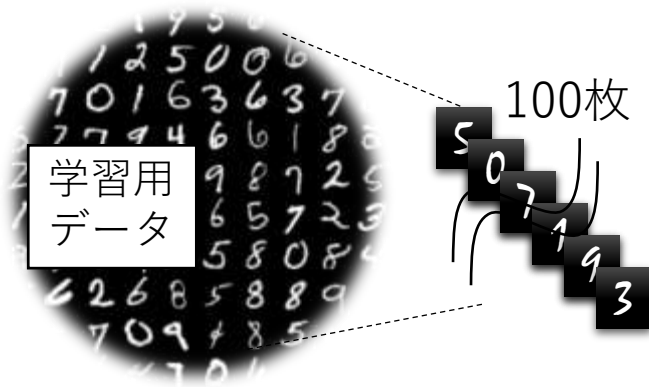
具体的にどうアプローチするか—指標の定義



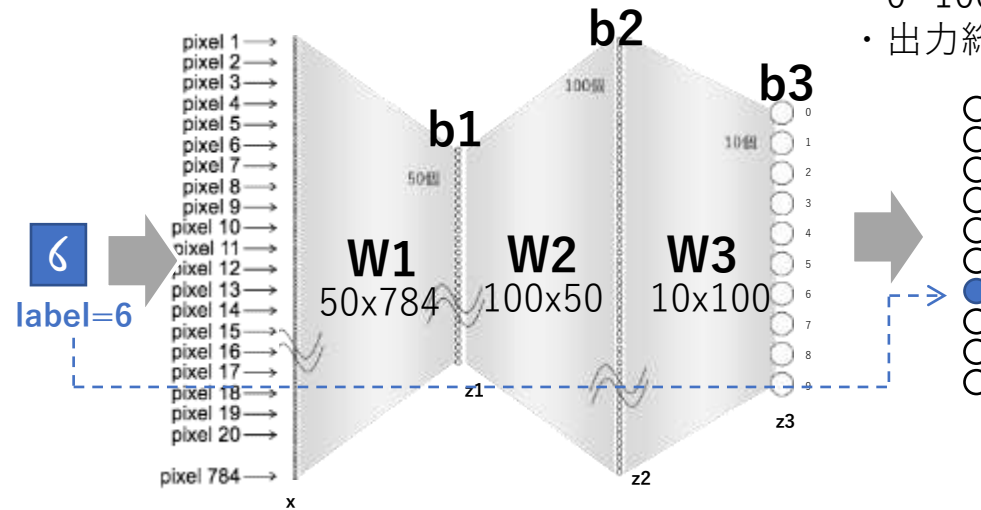


損失関数～ダメさ具合の指標

MNIST計6万枚のデータ群から、ランダムに100枚を選び出す。



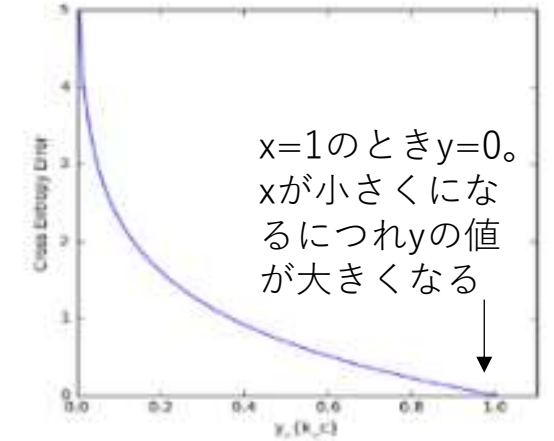
一枚ずつニューラルネットに入力して計算



出力を正規化
・ 0~100%の値
・ 出力総和=1

正解箇所の値のエンタピー
(-log)を計算

$$y = -\log(x)$$



100枚分計算して平均を求める。
これが損失関数の値となる。

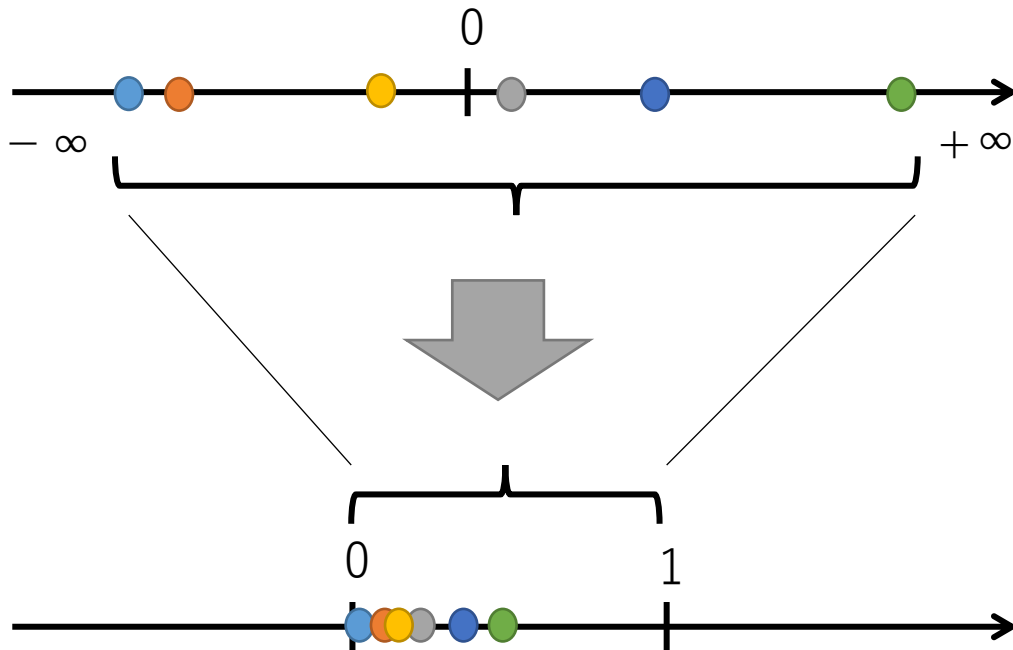
パラメータW1, W2, W3, b1, b2, b3
によって、損失関数の値が決まる

(補足) 指数関数を用いた正規化

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

正規化の様子を数直線で表現すると、
大小さまざまな数字について、それぞれの
位置関係は保ったままで、

- 0から1のあいだにギュッと押し込む
- 且つ、値の総和が1になる

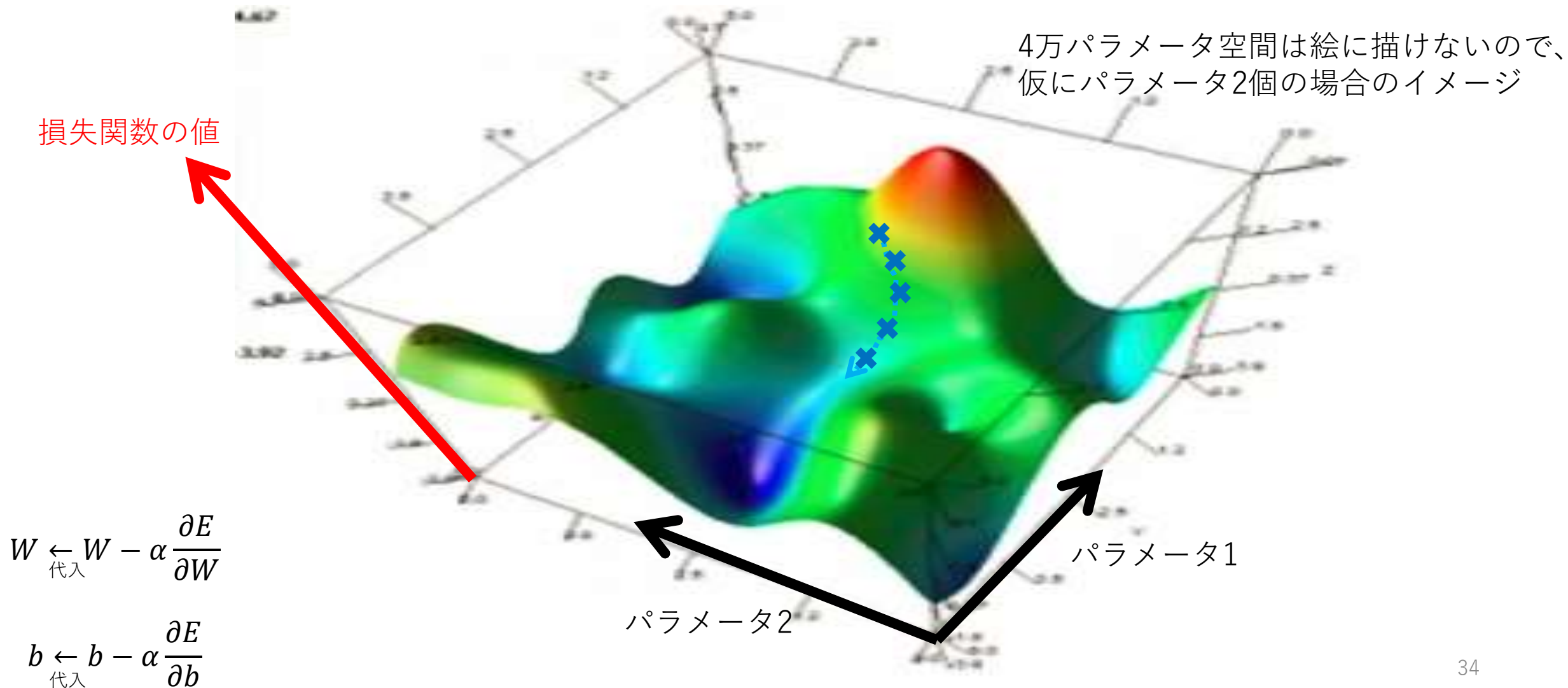


Excelシートで実際に試してみた例

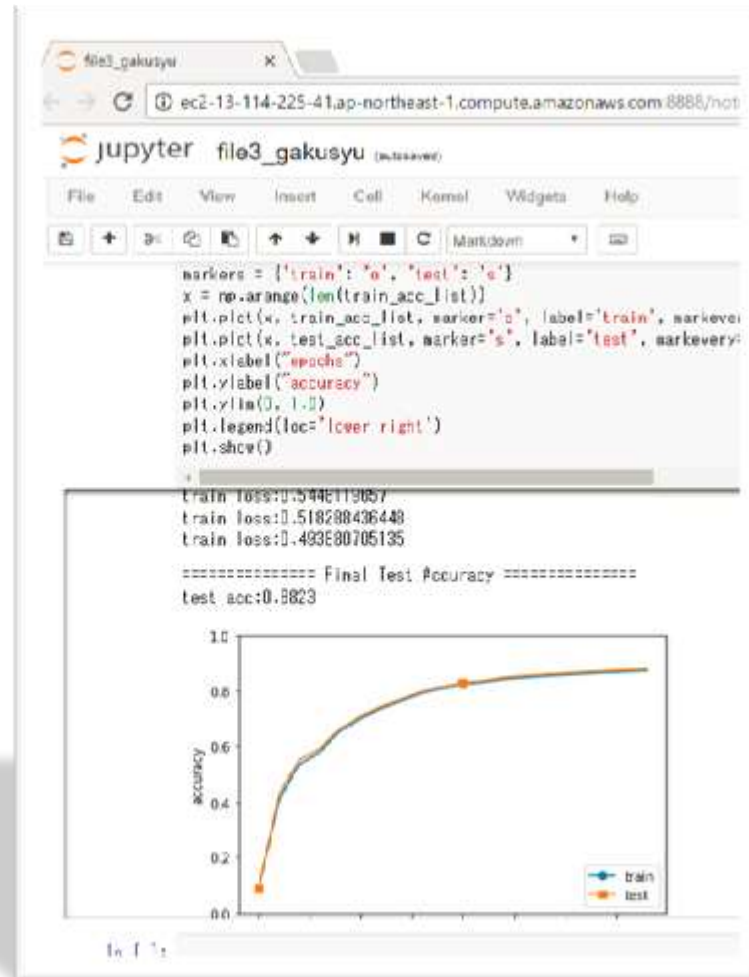
適当な数字	=EXP(左セル)		左セルを%表示
-1.2	0.301194	0.037784	4%
-0.3	0.740818	0.092934	9%
1.3	3.669297	0.460304	46%
0.1	1.105171	0.138641	14%
-1.1	0.332871	0.041758	4%
0.6	1.822119	0.22858	23%
	=SUM(列の合計)		列のSUM 確かに足したら1
	7.97147		100%

= (左セル) ÷ SUMの値

損失関数が一番小さくなるパラメータを探す = パラメータ空間(山谷)の底を探す



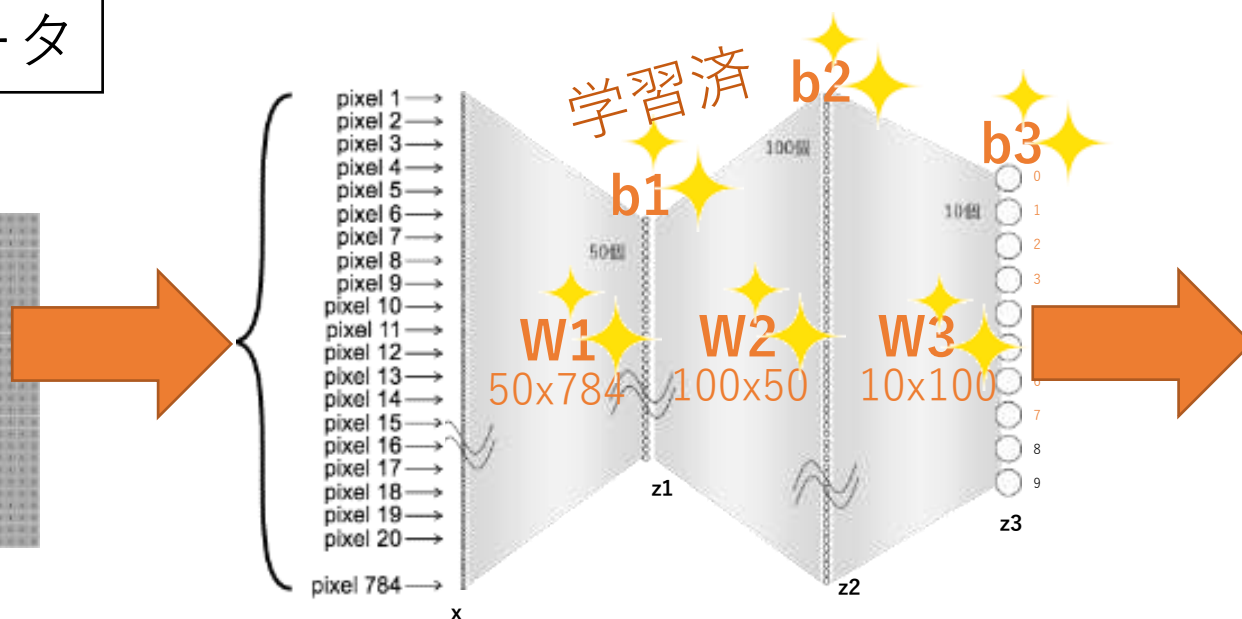
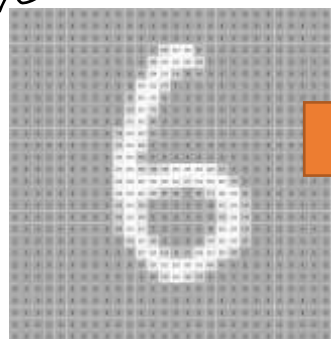
実際の学習の過程を見してみる



学習済のパラメータを使って、文字認識が正しく行われていることを確かめる

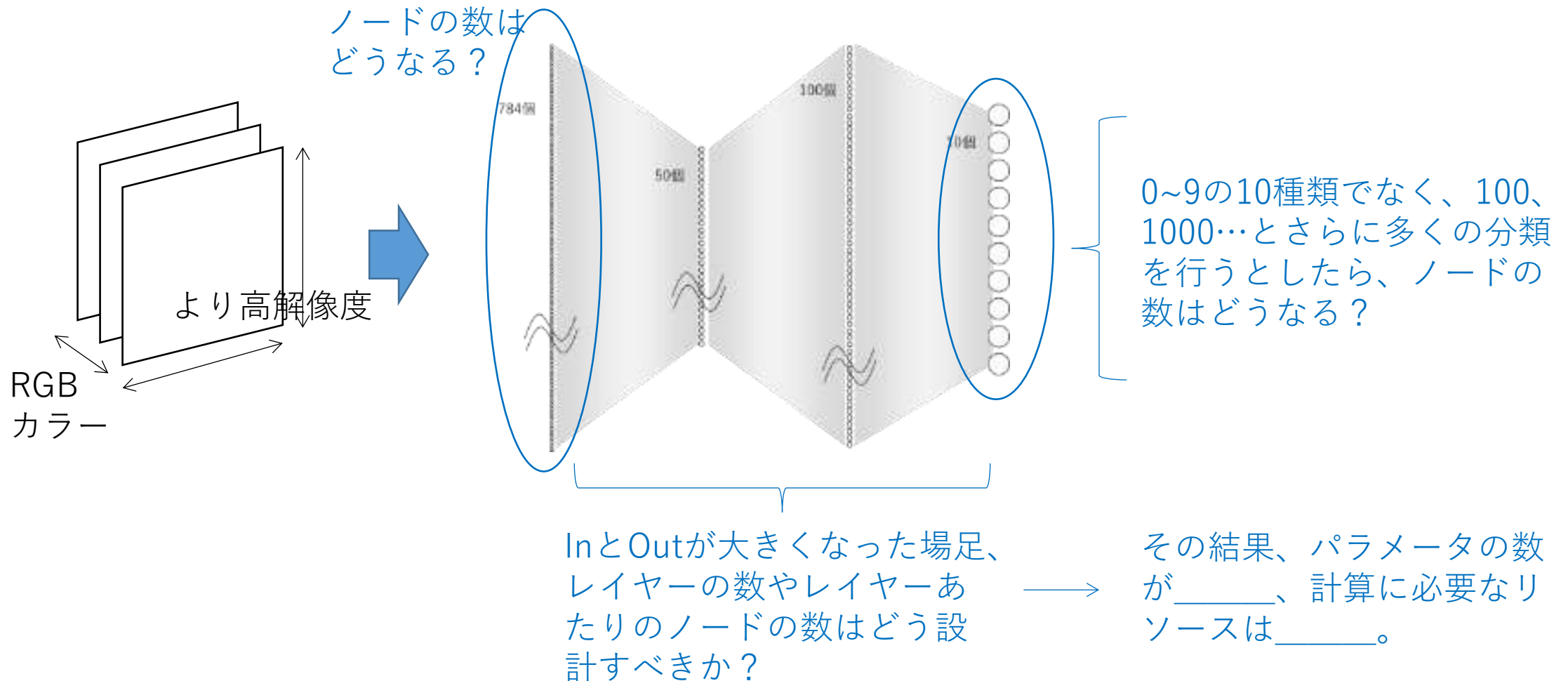
検証用のデータ

例えば「6」



「6」と認識できるか？
実際に試してみる。

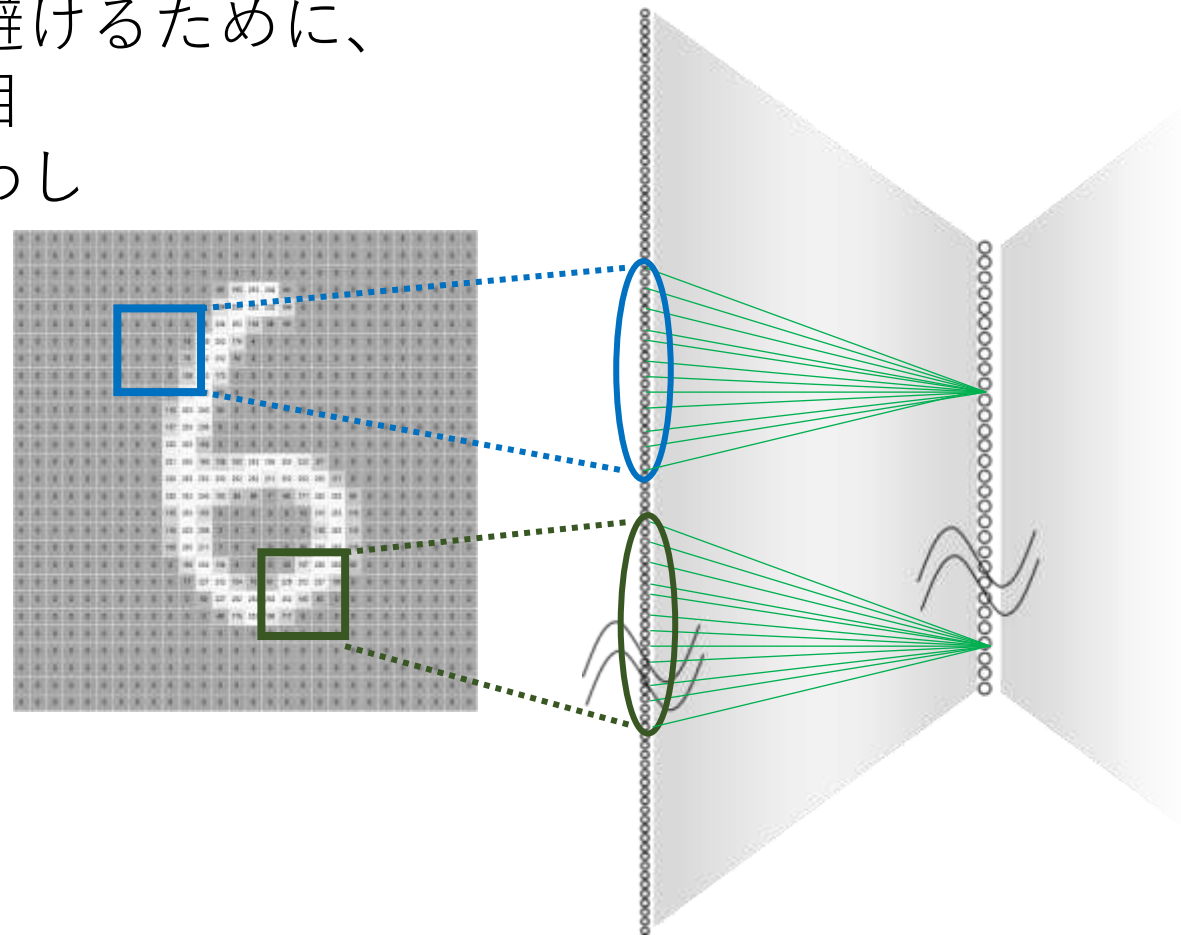
より複雑な課題への適用を考えてみる



畳み込みニューラルネットワーク (Convolutional Neural Network = CNN)

パラメータ数の爆発を避けるために、

- ・特徴量の局所性に着目
- ・パラメータの使いまわし

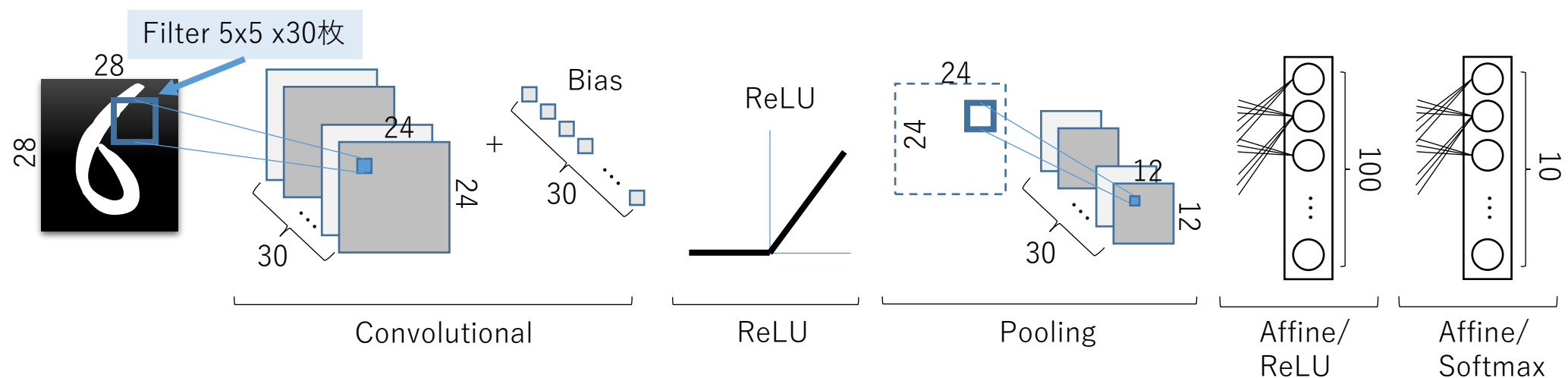


どういう計算をするか

http://brohrer.github.io/how_convolutional_neural_networks_work.html

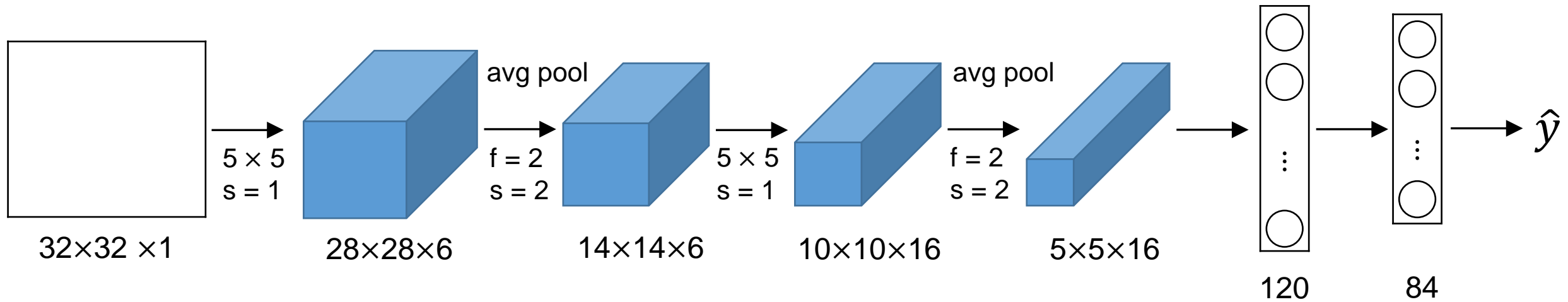
のパワーポイントを拝借して説明します。

評価に用いる畳み込みニューラルネットワークの構成

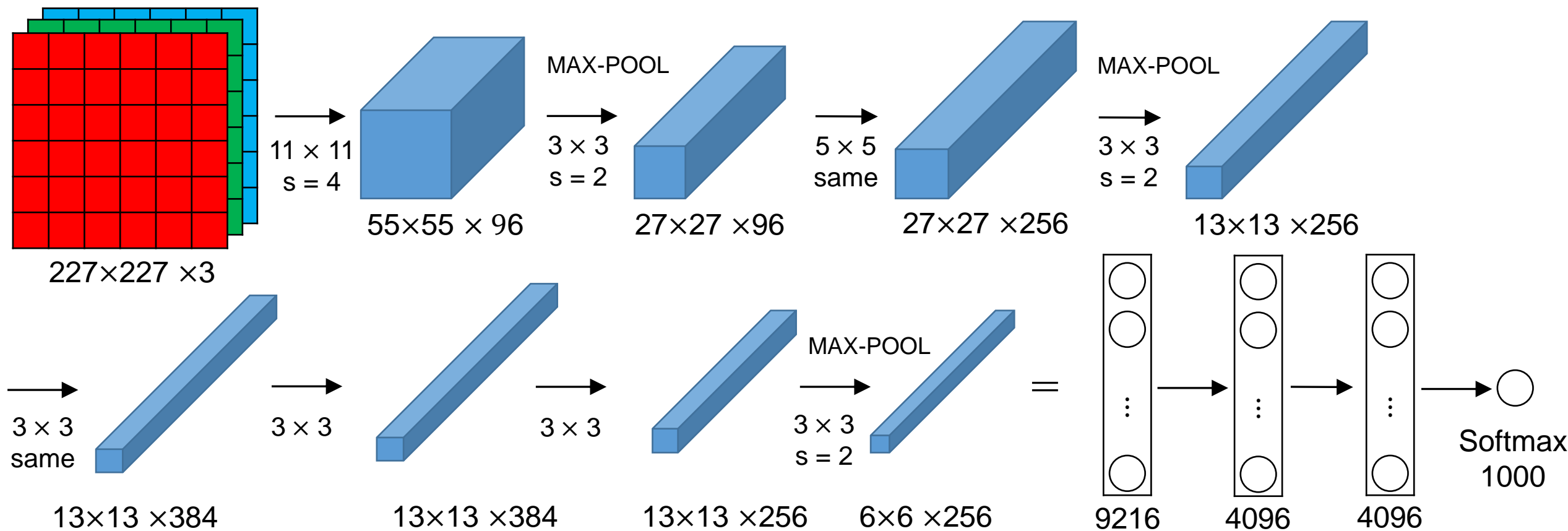


https://github.com/yoshihiroo/programming-workshop/blob/master/deep_learning_jupyter/file5_CNN.ipynb

LeNet - 5



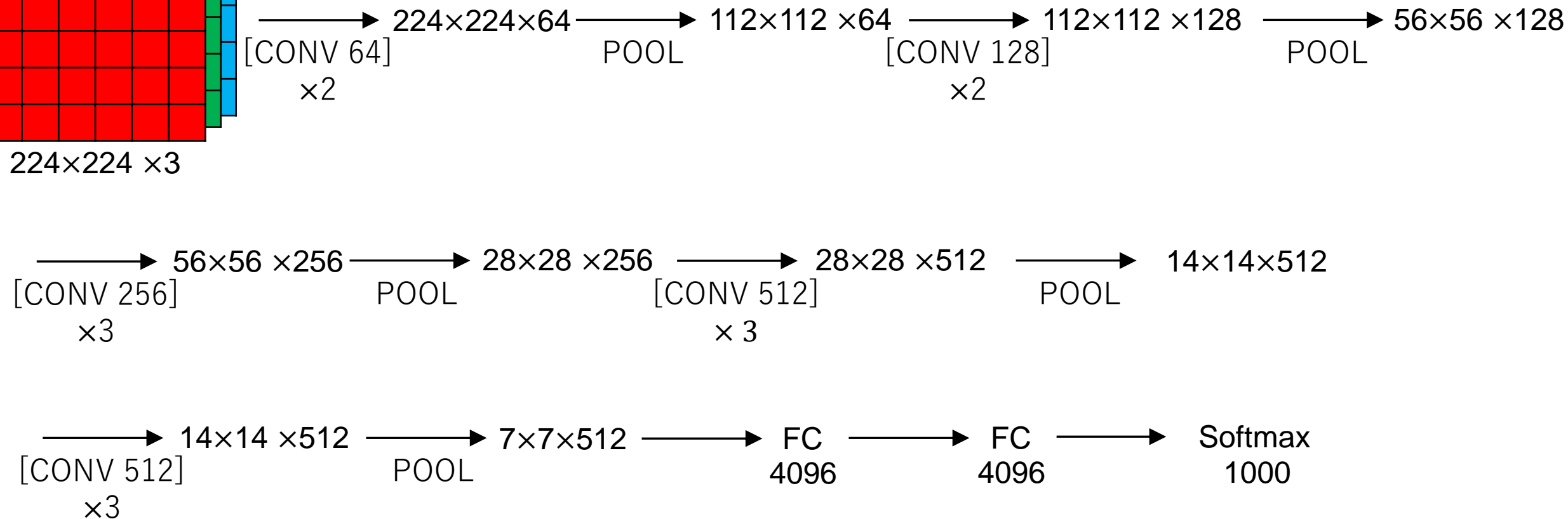
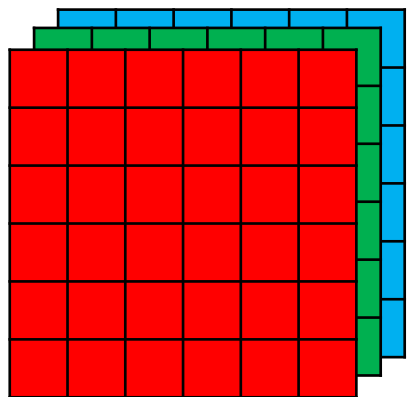
AlexNet



VGG - 16

CONV = 3×3 filter, s = 1, same

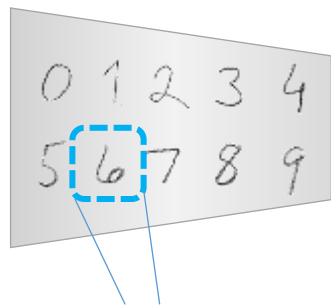
MAX-POOL = 2×2 , s = 2



今日の時間配分

0:00 – 2:00	<ul style="list-style-type: none">• ラズパイ基本セットアップ• カメラを使った画像配信
2:00 – 3:30	<ul style="list-style-type: none">• 座学<ul style="list-style-type: none">• ニューラルネットワーク• 手書き文字認識• 畳み込みニューラルネットワーク
3:30 – 4:30	<ul style="list-style-type: none">• 手書き文字認識システムの実装とテスト• 物体識別システムの実装とテスト
4:30 – 5:00	<ul style="list-style-type: none">• クロージング・振り返り

digit_recognition_NN.pyの概要

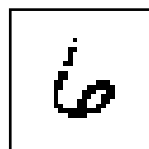


撮影

画像前処理



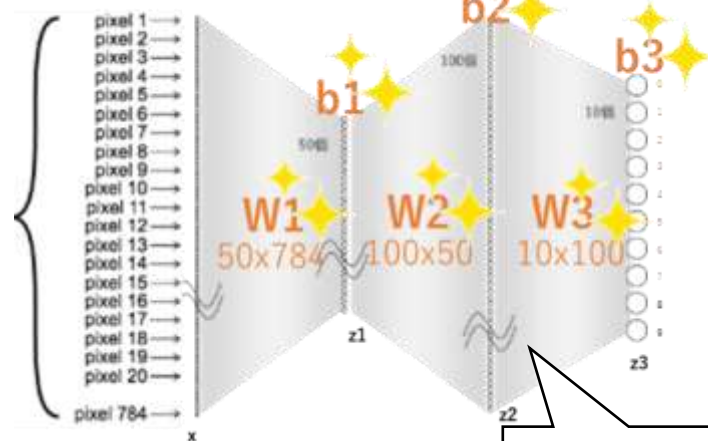
224x224
カラー



28x28
白黒(2値)



ニューラルネットワーク



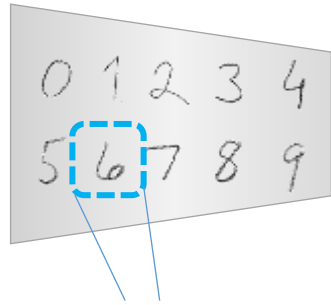
結果の出力



MNISTテスト
データで93%の
認識率

繰り返す

digit_recognition_CNN.pyの概要



撮影

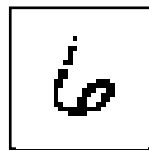
画像前処理

畳み込みニューラルネットワーク

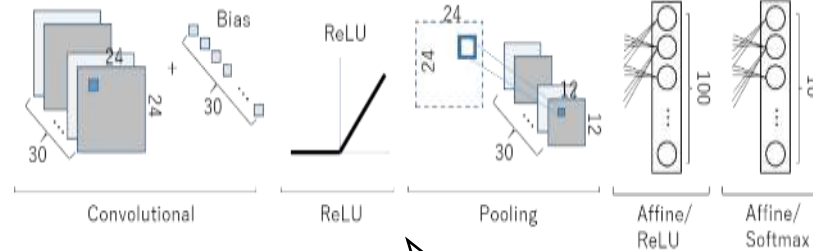
結果の出力



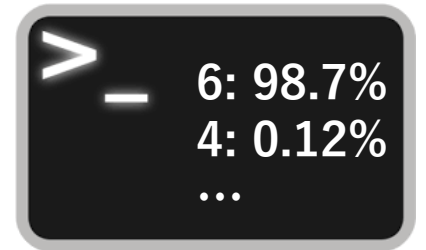
224x224
カラー



28x28
白黒(2値)

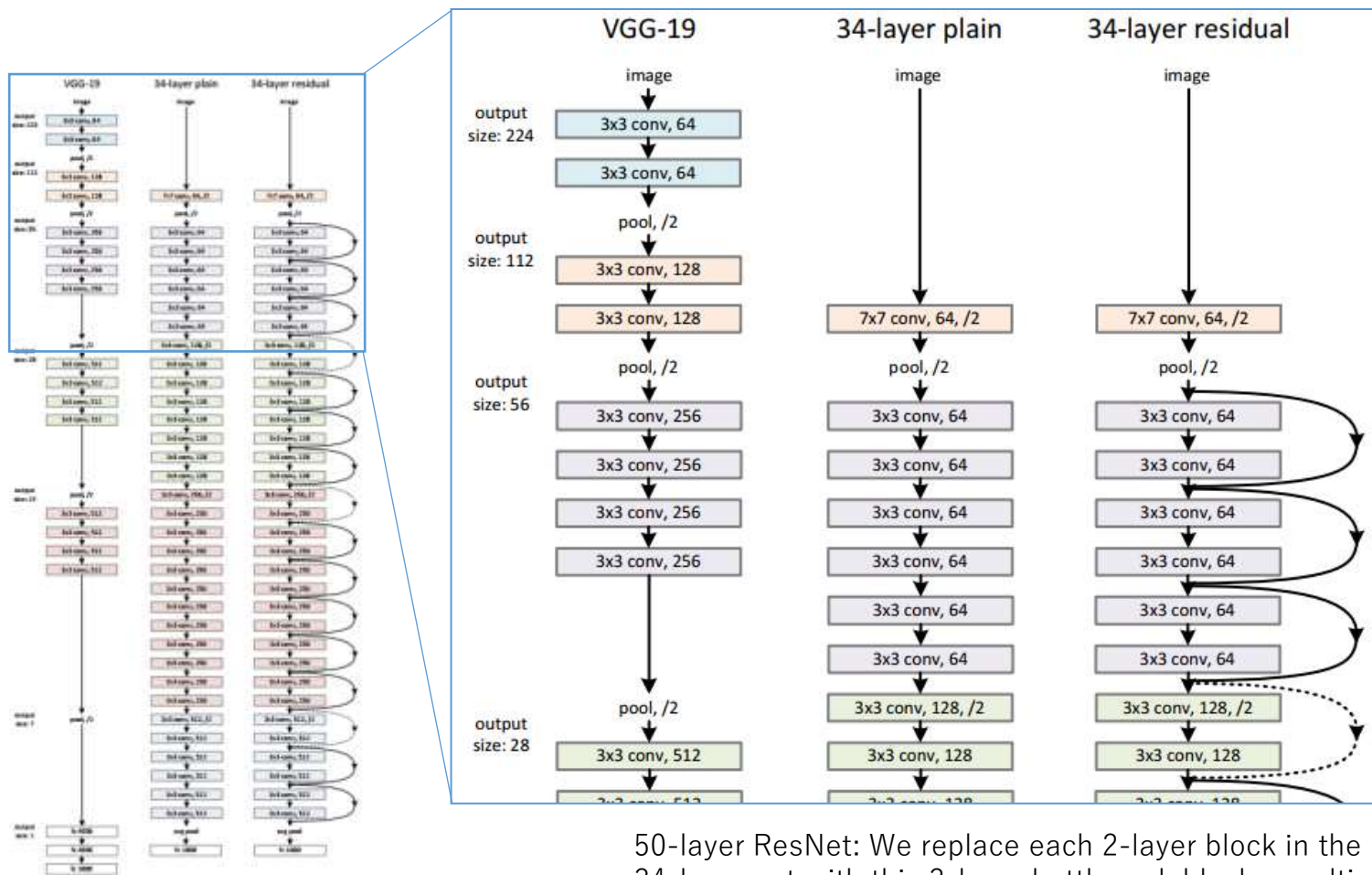


MNISTテスト
データで99%の
認識率



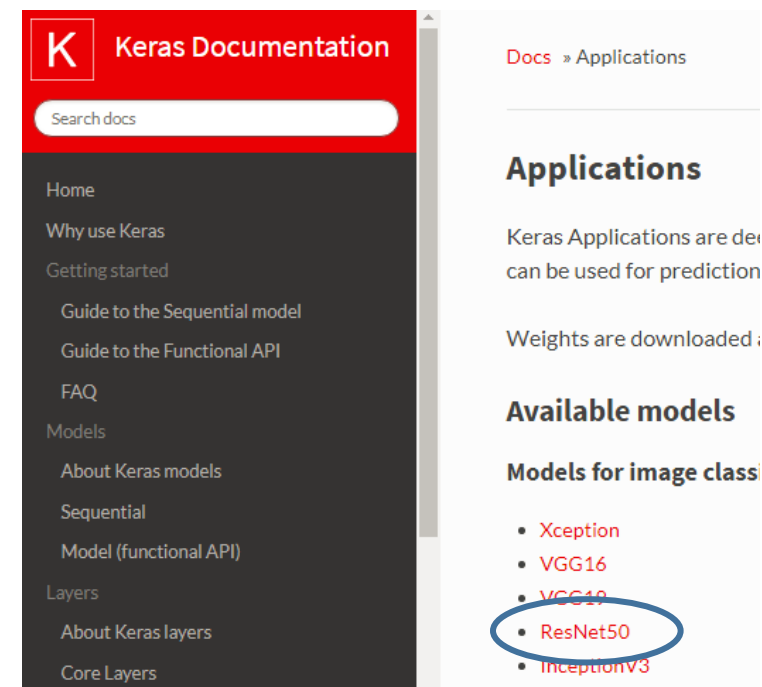
繰り返す

image_classification_resnet50.pyの概要



50-layer ResNet: We replace each 2-layer block in the 34-layer net with this 3-layer bottleneck block, resulting in a 50-layer ResNet

<https://keras.io/applications/>



上記サイトで公開されているKerasライブラリを用いたResNet50の実装コードをベースに、カメラ画像を取り込むように変更。

<https://arxiv.org/pdf/1512.03385.pdf>

image_classification_mobilenet.pyの概要



Google Research Blog

The latest news from Research at Google

MobileNets: Open-Source Models for Efficient On-Device Vision

Wednesday, June 14, 2017

Posted by Andrew G. Howard, Senior Software Engineer and Menglong Zhu, Software Engineer

(Cross-posted on the [Google Open Source Blog](#))

Deep learning has fueled tremendous progress in the field of computer vision in recent years, with neural networks repeatedly pushing the [frontier of visual recognition technology](#). While many of those technologies such as object, landmark, logo and text recognition are provided for internet-connected devices through the [Cloud Vision API](#), we believe that the ever-increasing computational power of mobile devices can enable the delivery of these technologies into the hands of our users, enabling a new breed of intelligent applications. Here are several examples of these devices and



概要

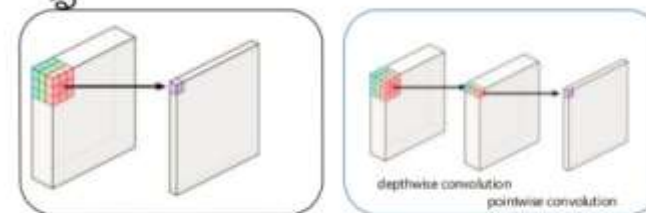
- CNNの進歩しているが、精度に比べサイズと速度の面での進歩が少ない
- サイズを小さくする研究はあったが、速度を考慮しているものは少ない
- ロボットなどの実世界でのアプリケーションでは速度が必要になる



- 効率的なネットワークアーキテクチャと2つのハイパーパラメータを提案
 - サイズを小さく、処理速度を速くする

Depthwise separable convolution

- 畳み込みを空間方向の畳込み(depthwise convolution)とチャネル方向の畳込み(pointwise convolution, 1*1 convolution)に分ける



通常の畳込み

Depthwise separable convolution

今日の時間配分

0:00 – 2:00	<ul style="list-style-type: none">• ラズパイ基本セットアップ• カメラを使った画像配信
2:00 – 3:30	<ul style="list-style-type: none">• 座学<ul style="list-style-type: none">• ニューラルネットワーク• 手書き文字認識• 畳み込みニューラルネットワーク
3:30 – 4:30	<ul style="list-style-type: none">• 手書き文字認識システムの実装とテスト• 物体識別システムの実装とテスト
4:30 – 5:00	<ul style="list-style-type: none">• クローニング・振り返り