現場で使えるディープラーニング基礎講座

DAY4

SkillUP AI

2019 SkillUP AI All rights reserved

前回の復習

- 前回は何を学びましたか？

2019 SkillUP AI All rights reserved

ディープラーニングの様々なモデル

2019 SkillUP AI All rights reserved

ディープラーニングの様々なモデル

- ディープラーニングには様々なモデルがあるが、特に覚えておくべきは以下の5つ！

確定的　　確率的

階層型	自己符号化器型	ボルツマンマシン型
全結合型ニューラルネットワーク fully connected neural network	自己符号化器 autoencoder	ボルツマンマシン Boltzmann machine
畳み込みニューラルネットワーク convolutional neural network	雑音除去自己符号化器 denoising autoencoder	制約ボルツマンマシン restricted Boltzmann machine
再帰型ニューラルネットワーク recurrent neural network	変分自己符号化器 variational autoencoder	深層信念ネットワーク deep belief network
	スパース自己符号化器 sparse autoencoder	深層ボルツマンマシン deep Boltzmann machine

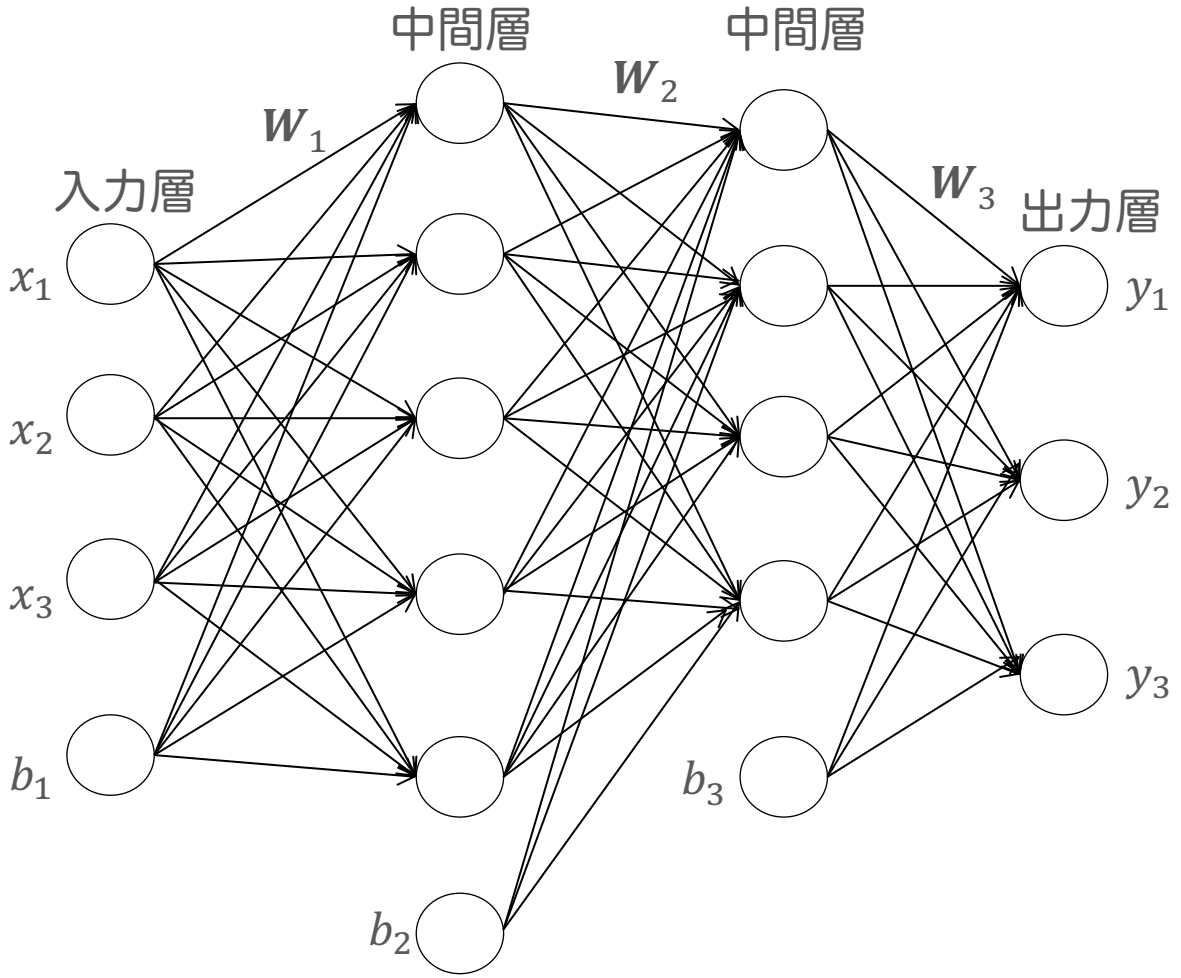
参考：『深層学習』(神嶌など、近代科学社)

2019 SkillUP AI All rights reserved

全結合型ニューラルネットワークの計算グラフ

- いわゆる普通のニューラルネットワーク

全結合型ニューラル
ネットワークの例

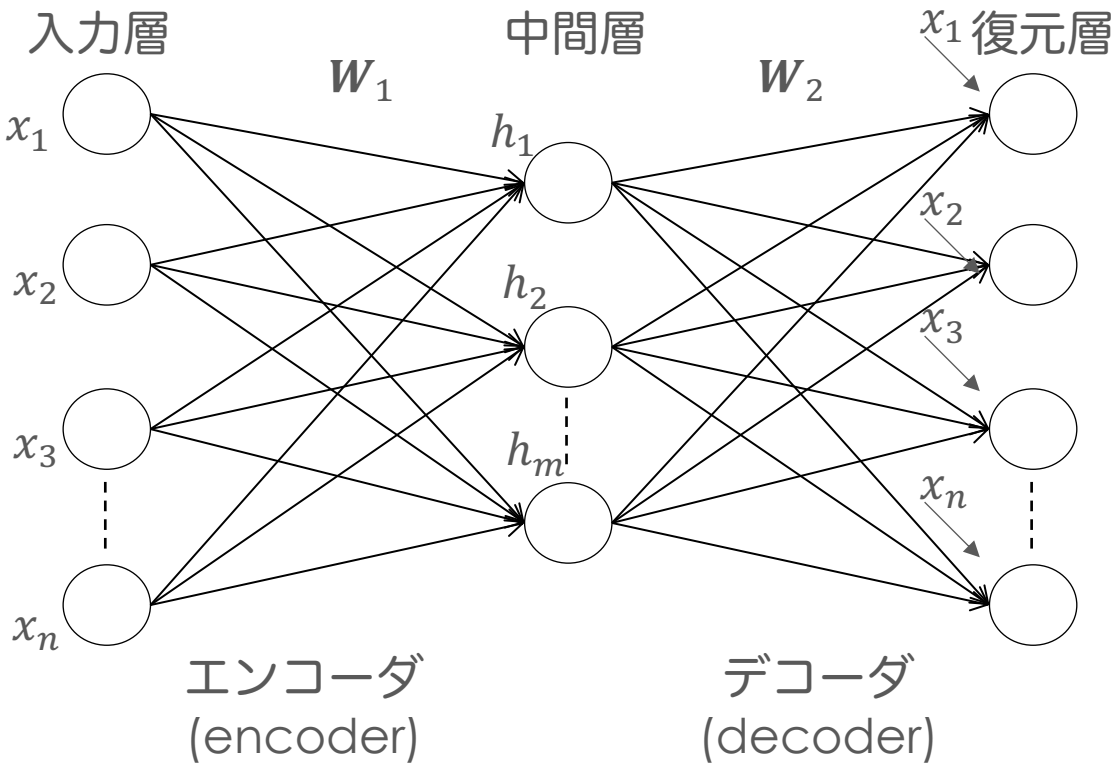


2019 SkillUP AI All rights reserved

自己符号化器型の計算グラフ

- 自己符号化器型のニューラルネットワークでは、入力層と復元層に同じデータを与えて学習を行う。
- 学習後、 データ$\boldsymbol{x}$を入力したときの中間層の出力$\boldsymbol{h}$が特徴量になる

自己符号化器の例



2019 SkillUP AI All rights reserved

自己符号化器とは

- 自己符号化器(autoencoder)とは、エンコーダとデコーダを組み合わせたネットワークを組み、ノード間の重みを学習する方法。
- 特徴抽出器(次元圧縮)、ノイズ除去、復元誤差を利用した異常検知などに利用される。
- スパース自己符号化器
  - 自己符号化器に正則化項を加えて学習させる方法。
  - 効率的にユニット数を少なくできる。
- 雑音除去自己符号化器
  - ランダムなノイズを付加したデータを入力層に入れる方法。
  - 復元層から出てくるデータがノイズを付加する前のデータに近くなるように学習させる。ノイズ除去の能力が備わることを期待できる。
- 変分自己符号化器
  - 生成モデルの1つ。
  - データが生成するメカニズムを仮定する。

2019 SkillUP AI All rights reserved

再帰型ニューラルネットワークとは

- 再帰型ニューラルネットワーク(recurrent neural network, RNN)とは、時間方向に状態を引き継ぎながら計算を進めることができるニューラルネットワーク。
- 自然言語などのように単語が順番に並んでいるデータを自然に扱うことができるため、自然言語処理や音声認識に利用される。
- 自然言語のように入力長が毎回異なるデータを扱う場合は、RNNが向いている。

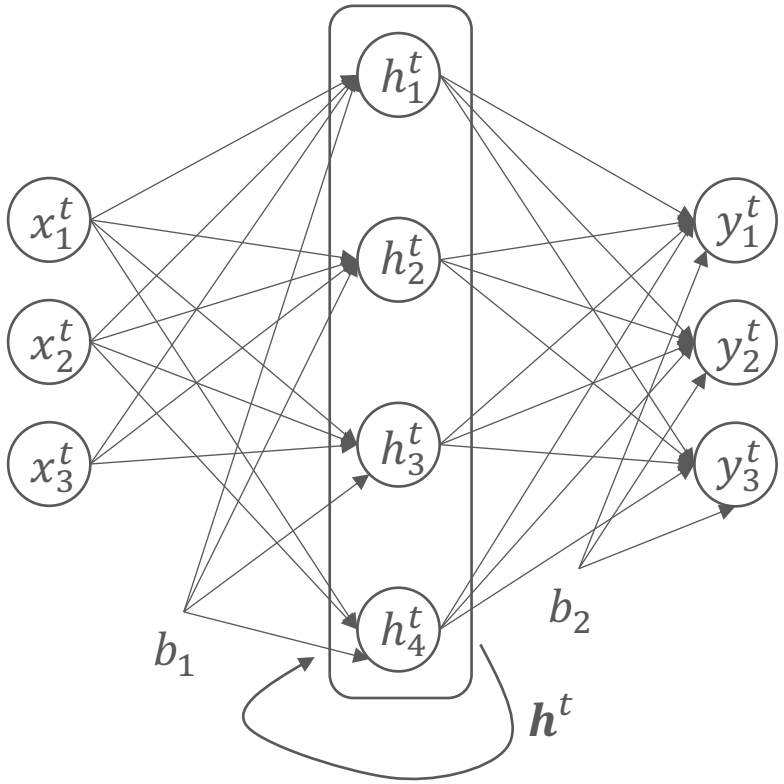
2019 SkillUP AI All rights reserved

再帰型ニューラルネットワークの種類

- シンプルなRNN
  - 単純な全結合層を用いて状態を更新していく。
  - 長い系列を学習しても、過去の情報がほとんど反映されないという欠点がある。
- LSTM (Long Short-Term Memory)
  - シンプルなRNNの課題を解決するために提案されたRNN。
  - 入力ゲート(input gate)、出力ゲート(output gate)、忘却ゲート(forget gate)、記憶セル(memory cell)と呼ばれる仕組みが導入され、必要な情報を長く記憶できるようになった。
  - “Long Short-Term Memory”とは、”短期記憶を長い時間保持できること”を意味する。
- GRU(Gated Recurrent Unit)
  - LSTMをシンプルにしたRNNであり、LSTMよりもパラメータの数が少ない。
  - リセットゲート(reset gate)と更新ゲート(update gate)のみで構成される。

2019 SkillUP AI All rights reserved

シンプルなRNNの計算グラフ



中間層の出力が次の時刻
の中間層への入力になる

2019 SkillUP AI All rights reserved

RNNに関する注意事項

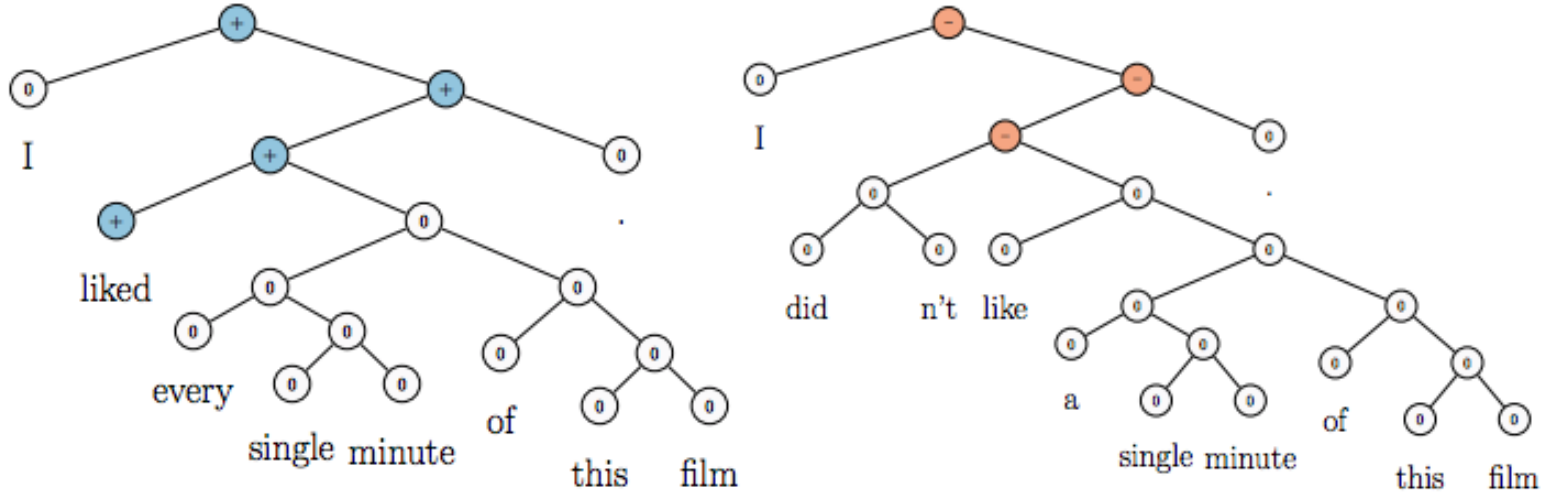
- 本講座では、recurrent neural networkのことを再帰型ニューラルネットワークと呼んでいる。
- 『深層学習, Goodfellow』では、 recurrent neural networkは回帰結合型ニューラルネットワークと訳されている。また、この書籍では、再帰型ニューラルネットワークという言葉が、 recursive neural networkというモデルの訳として用いられている。
- E資格を受験される方は要注意。

2019 SkillUP AI All rights reserved

参考：recursive neural network

- 木構造の計算グラフをもつネットワーク。
- 最適な木構造を決定する方法が確立されておらず、応用例は少ない。

recursive neural networkを用いた感情分析モデルの例



Richard Socher , Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank
https://nlp.stanford.edu/~socherr/EMNLP2013_RNTN.pdf

2019 SkillUP AI All rights reserved

Any Questions?

2019 SkillUP AI All rights reserved

畳み込みニューラルネットワーク

2019 SkillUP AI All rights reserved

目次

2019 SkillUP AI All rights reserved

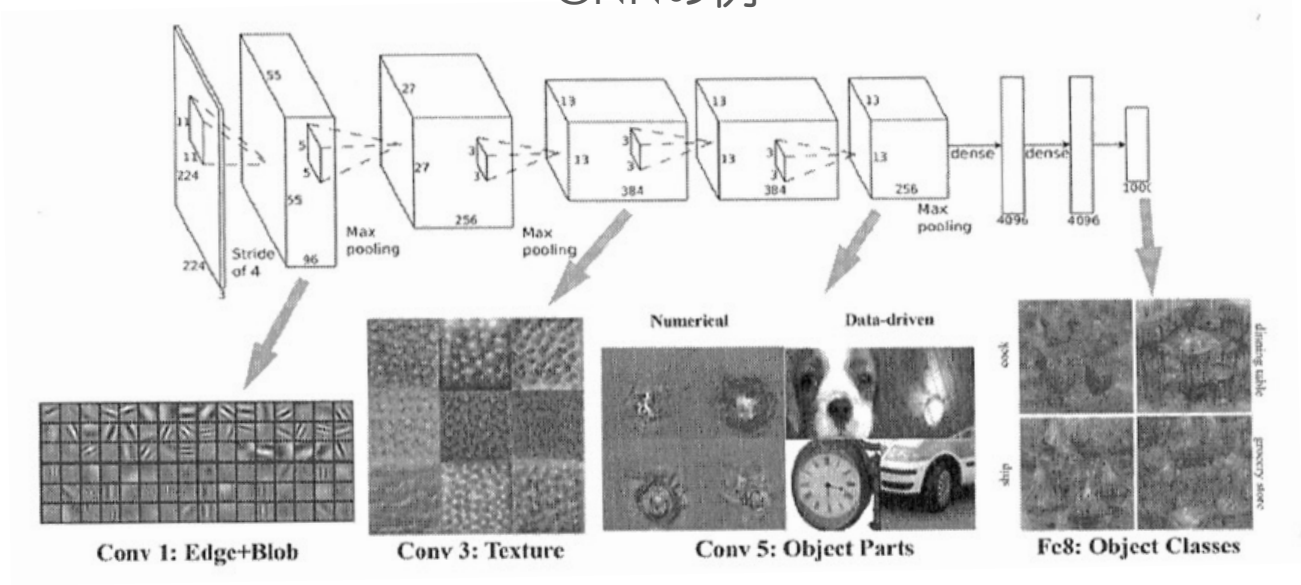
CNN概要

2019 SkillUP AI All rights reserved

CNN (Convolutional Neural Network) とは

- CNNとは、畳み込み層とプーリング層を用いたニューラルネットワークのこと。
- 主に画像認識に使われる。
- 畳み込み層とプーリング層は、脳の視覚野に関する神経科学の知見をヒントにしている。
  参考：『深層学習, 岡谷, p.79』

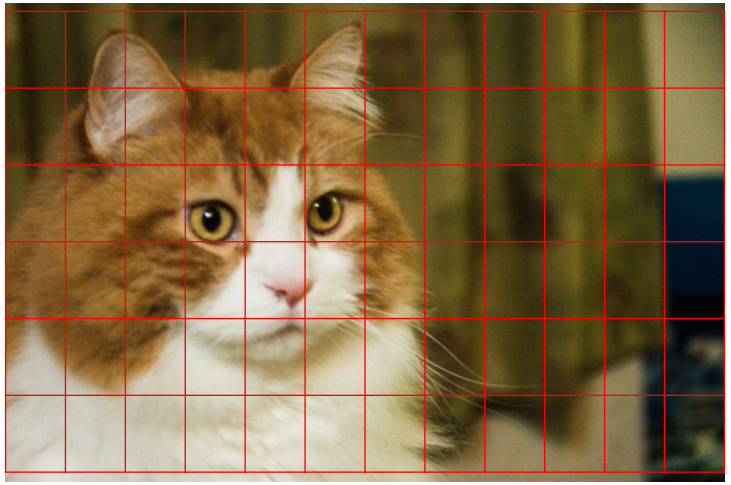
CNNの例

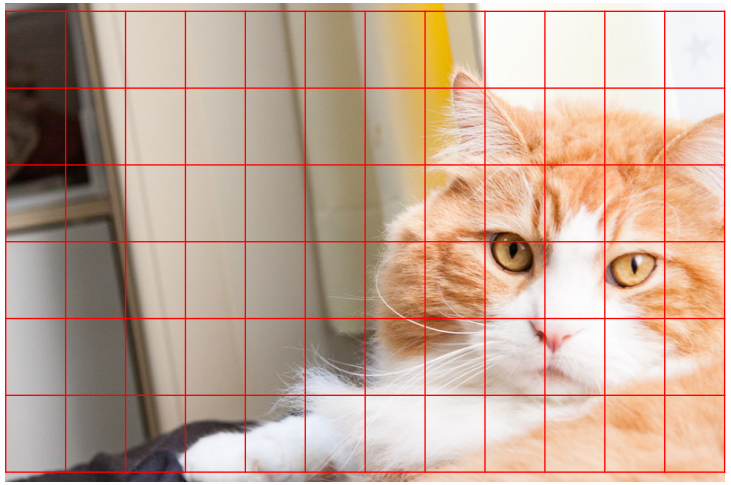


2019 SkillUP AI All rights reserved

CNNのコンセプト

- 以下の2枚はどちらも猫の写真。
- でも、顔の位置や向きが異なる。
- どんな猫の写真が入力されてもきちんと猫であると識別できるモデルをつくりたいが、通常のNNでは特徴のズレを考慮できない。
- 特定の写真に依存しない普遍的な特徴を抽出するにはどうすればいいか？
- その難問に応えるべく考案されたのが畳み込み層とプーリング層。





2019 SkillUP AI All rights reserved

従来の識別手法とCNNの違い

人の考えたアルゴリズム → 答え

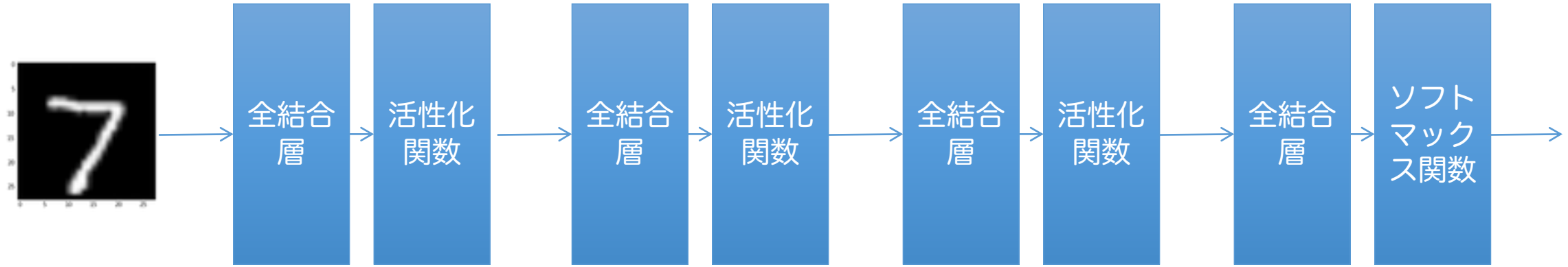
人の考えた特徴量
(SIFT、HOGなど) → 機械学習
(SVM、KNNなど) → 答え

ニューラルネットワーク
(ディープラーニングなど) → 答え

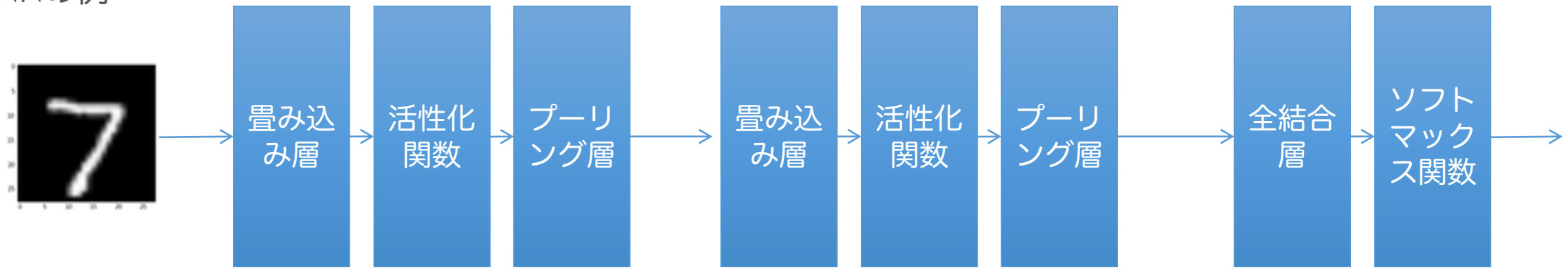
2019 SkillUP AI All rights reserved

全結合型NNとCNNの違い

全結合型NNの例



CNNの例



2019 SkillUP AI All rights reserved

Any Questions?

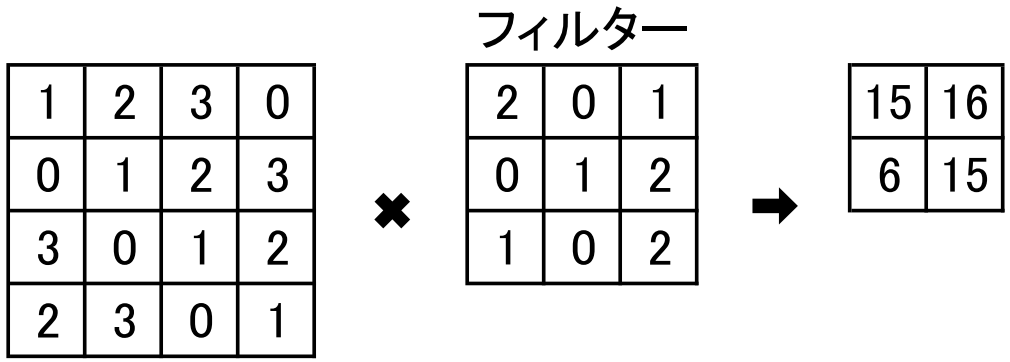
2019 SkillUP AI All rights reserved

畳み込み層

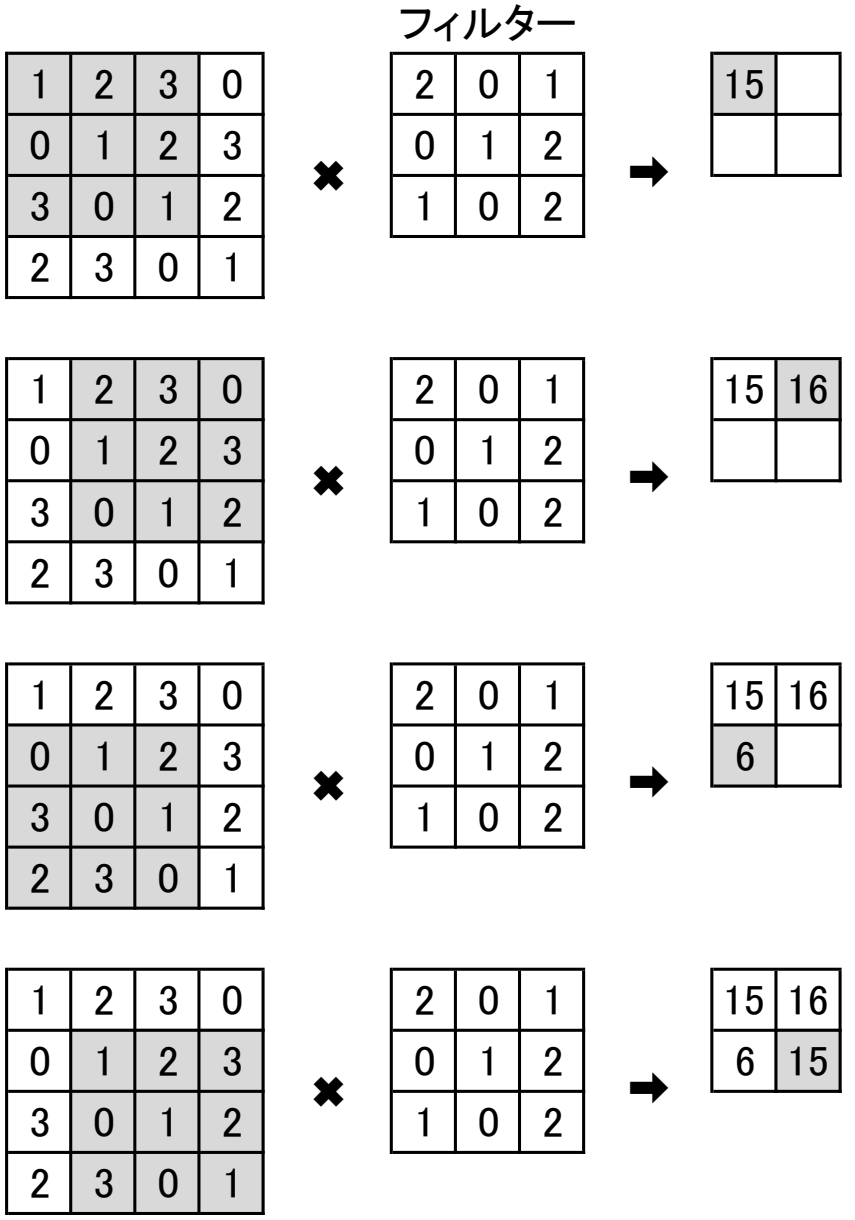
2019 SkillUP AI All rights reserved

畳み込み演算

- 畳み込み演算は、画像処理におけるフィルター演算に相当する。
- フィルターという言葉は、文献によってはカーネルと表現されることもある。
- フィルターの値は、NNの重みとして扱うことができ、学習によって最適な値が求められる。



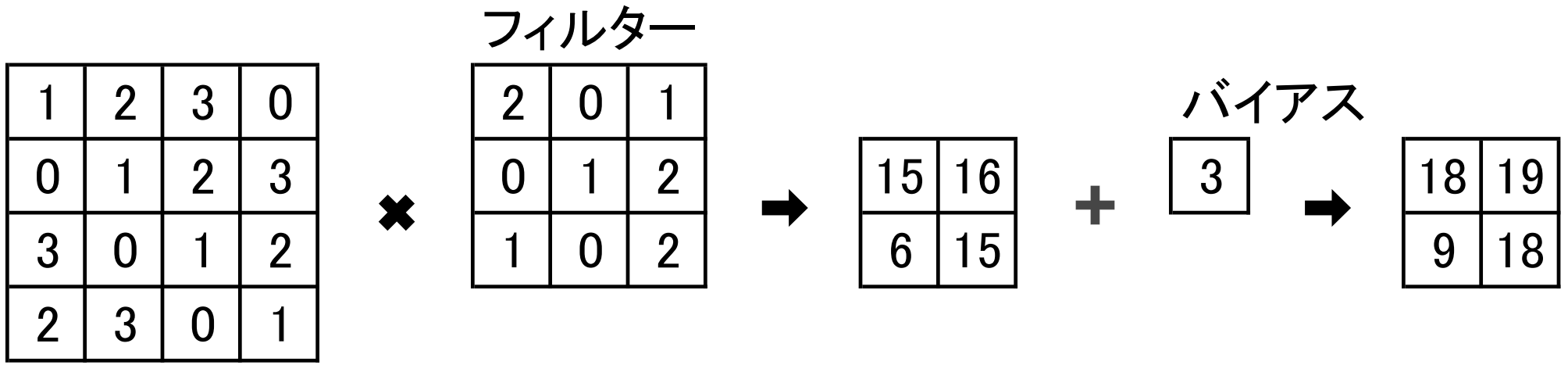
✖ : 畳み込み演算



2019 SkillUP AI All rights reserved

畳み込み演算のバイアス

- 畳み込み演算時のバイアスは、フィルター内で共通の1つのパラメータとして扱われることが多い。

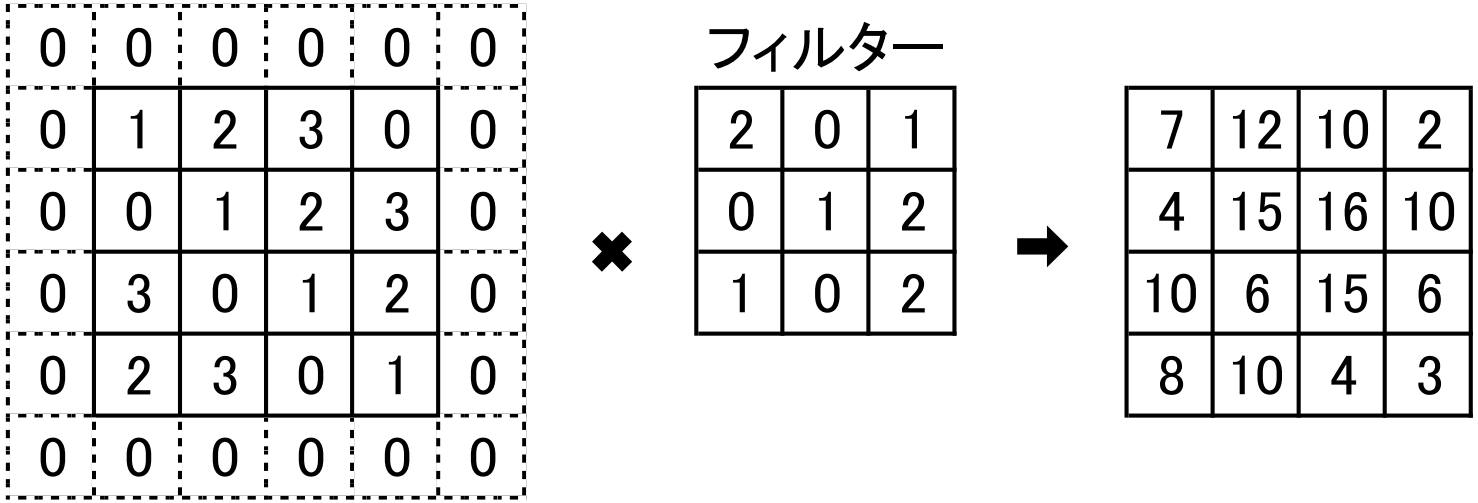


✖ : 畳み込み演算

2019 SkillUP AI All rights reserved

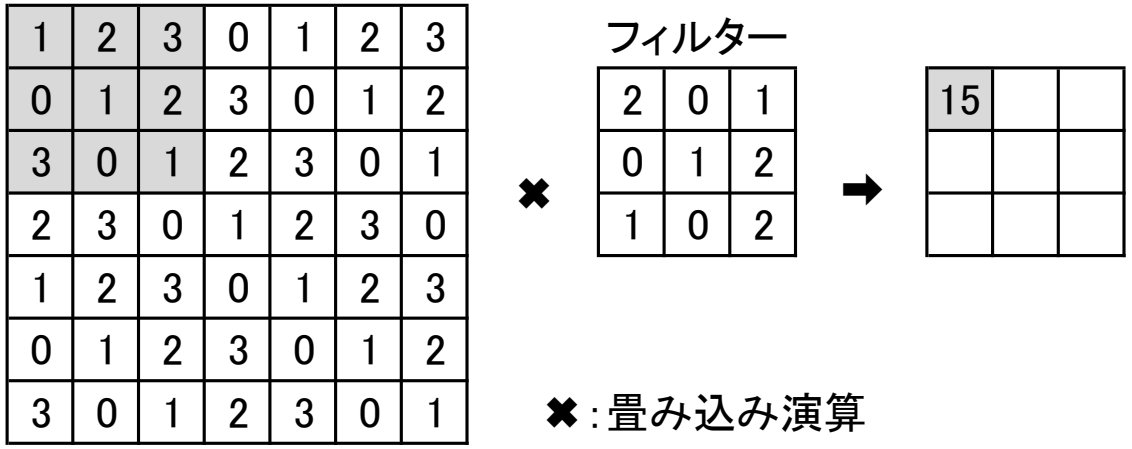
畳み込み演算のパディング処理

- 畳み込み演算を行う前に、画像の周辺に要素を追加することをパディング処理という。
  - 周辺は0で埋められることが多い(ゼロパディング)。
- パディング処理を行うことによって、畳み込み演算後のサイズを元のサイズと同じにすることが可能になる。
  - また、パディング処理により、画像の周辺部の特徴を捉えやすくなる。

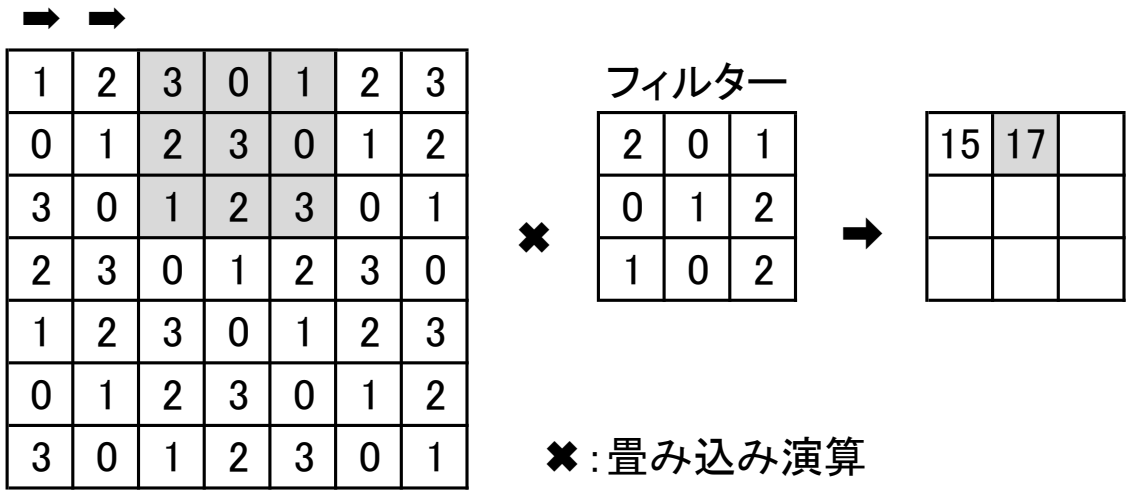


2019 SkillUP AI All rights reserved

畳み込み演算のストライド



ストライド=2



2019 SkillUP AI All rights reserved

[演習] 畳み込み演算における出力サイズの計算

- 畳み込み演算における出力サイズは以下の式によって計算できます。
- 手計算にて、前頁の出力サイズを計算してみましょう。

$$OH = \frac{H + 2P - FH}{S} + 1$$

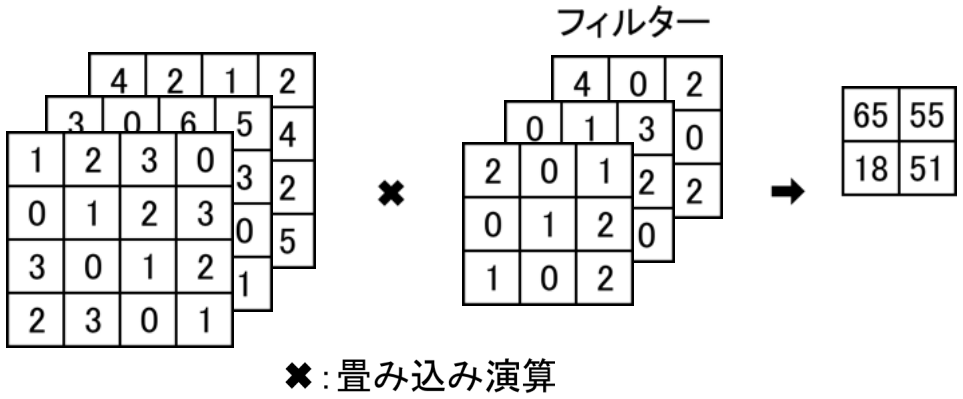
$$OW = \frac{W + 2P - FW}{S} + 1$$

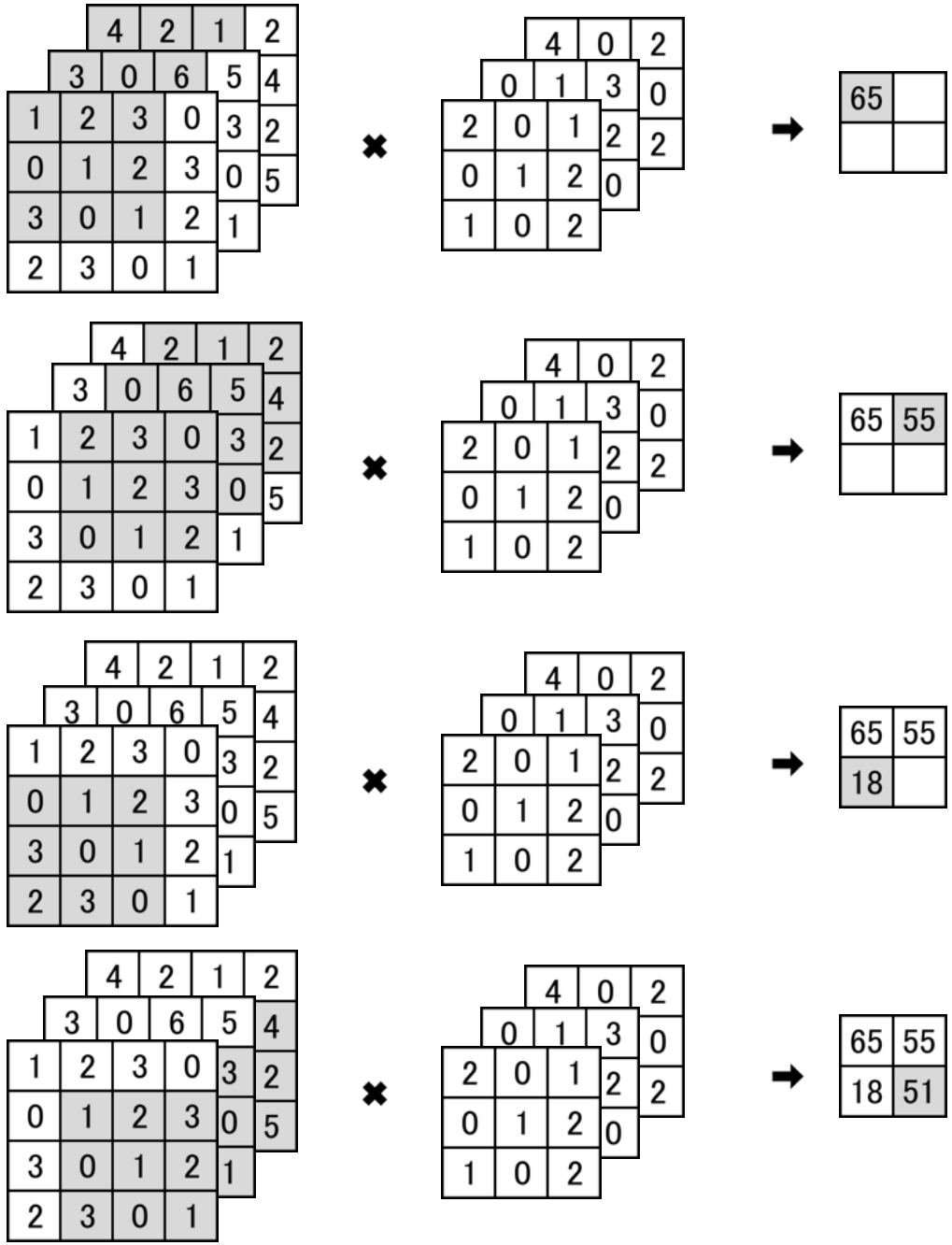
OH : 出力の高さ
OW : 出力の幅
H : 入力の高さ
W : 入力の幅
P : パディングサイズ
FH : フィルターの高さ
FW : フィルターの幅
S : ストライドサイズ

2019 SkillUP AI All rights reserved

3次元データ(3チャンネル)の畳み込み

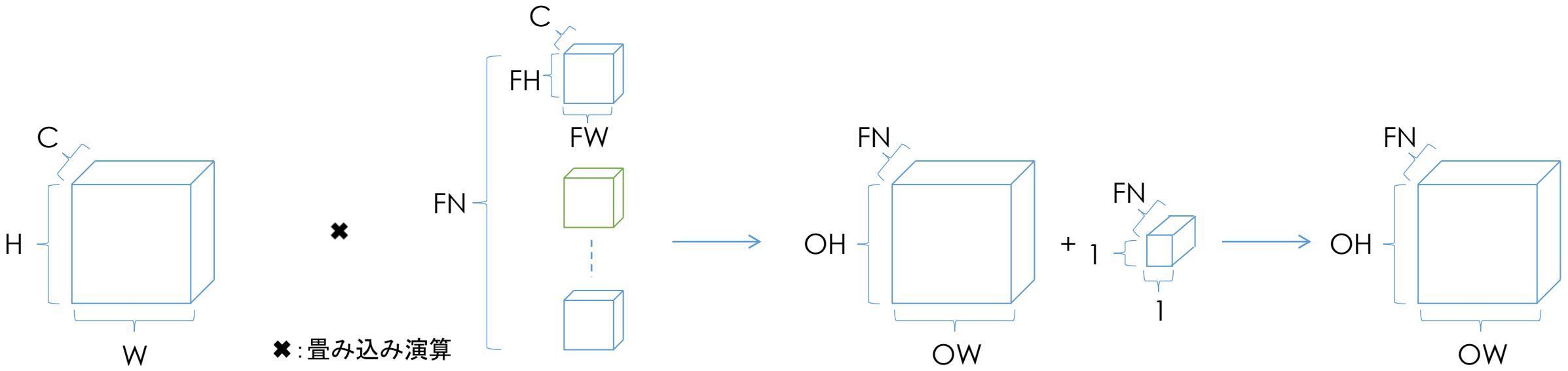
- 複数のチャンネルがある場合、チャンネル毎に入力データとフィルターの畳み込みを行い、それらの結果を加算して1つの出力を得る。
- もしRGB毎の出力が有効である場合、フィルターの重みがそのように学習される。





2019 SkillUP AI All rights reserved

畳み込み演算をブロックで考える



入力データ

(バッチ処理時は
これがN個ある)

フィルター

(バッチ処理時は
これがN個ある)

バイアス

出力データ

(バッチ処理時は
これがN個ある)

2019 SkillUP AI All rights reserved

[演習] 畳み込み演算の仕組みを確認

- 4_1_convolution.ipynb
  - 畳み込み演算の仕組みを確認しましょう。

2019 SkillUP AI All rights reserved

Any Questions?

2019 SkillUP AI All rights reserved

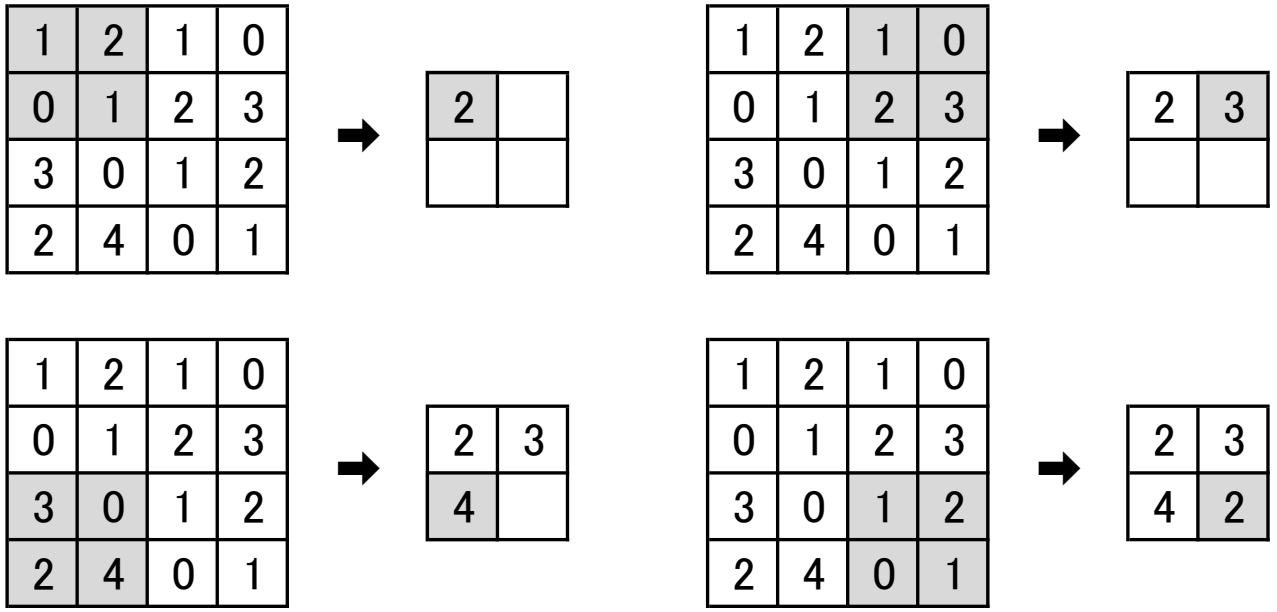
プーリング層

2019 SkillUP AI All rights reserved

プーリング演算

- ある範囲の平均や最大値をとることによって、要素を集約する処理のことをプーリング演算という。
- マックスプーリングや平均プーリングなどがある。
- 畳み込み演算と異なり、学習する重みはない。

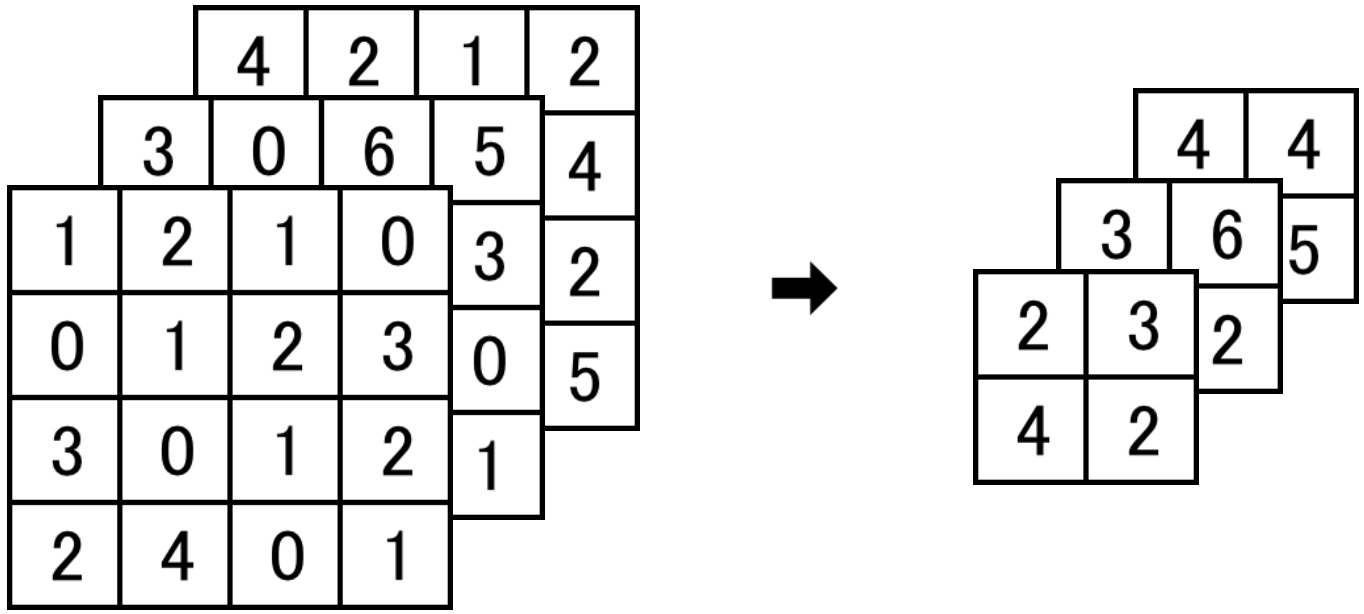
マックスプーリングの例



2019 SkillUP AI All rights reserved

複数チャンネルデータのプーリング演算

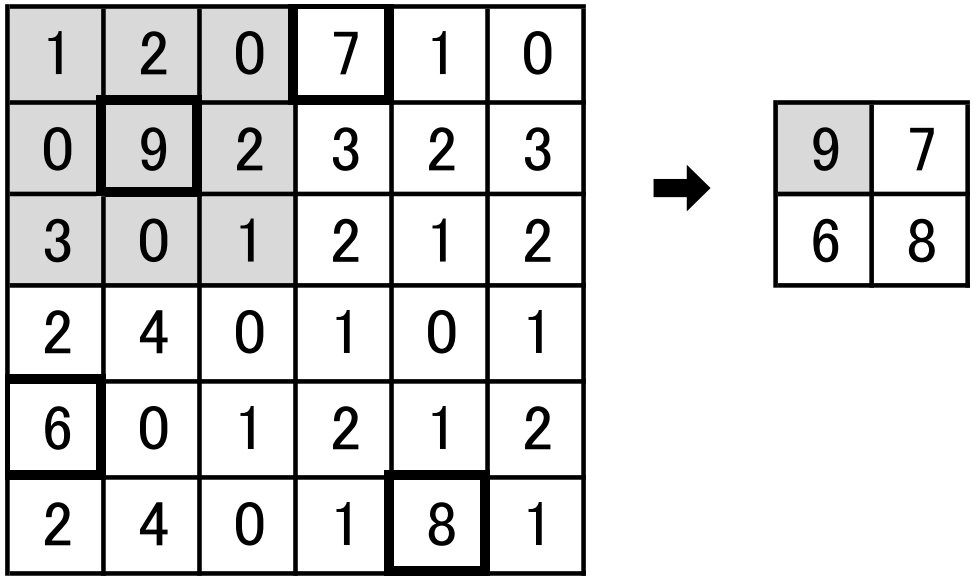
- 畳み込み演算と異なり、プーリング演算を行なってもチャンネル数は変わらない。

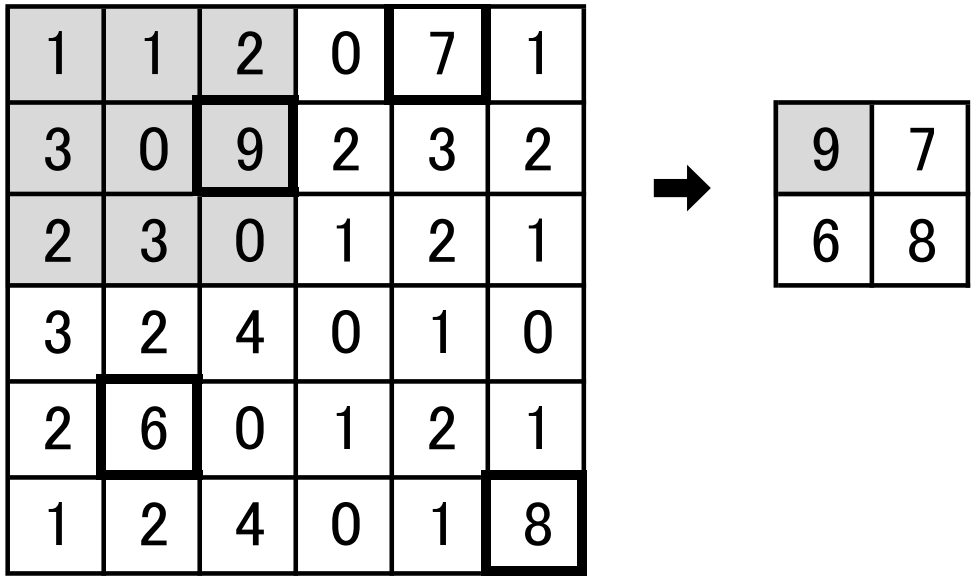


2019 SkillUP AI All rights reserved

プーリング演算の意義

- プーリング演算によって画像のズレを吸収できる。
- 例えば、以下のようなマックスプーリングでは、画像がずれていてもプーリング後の出力が同じになる。





2019 SkillUP AI All rights reserved

[演習] マックスプーリング演算の仕組みを確認

- 4_2_maxpooling.ipynb
  - マックスプーリング演算の仕組みを確認しましょう。

2019 SkillUP AI All rights reserved

Any Questions?

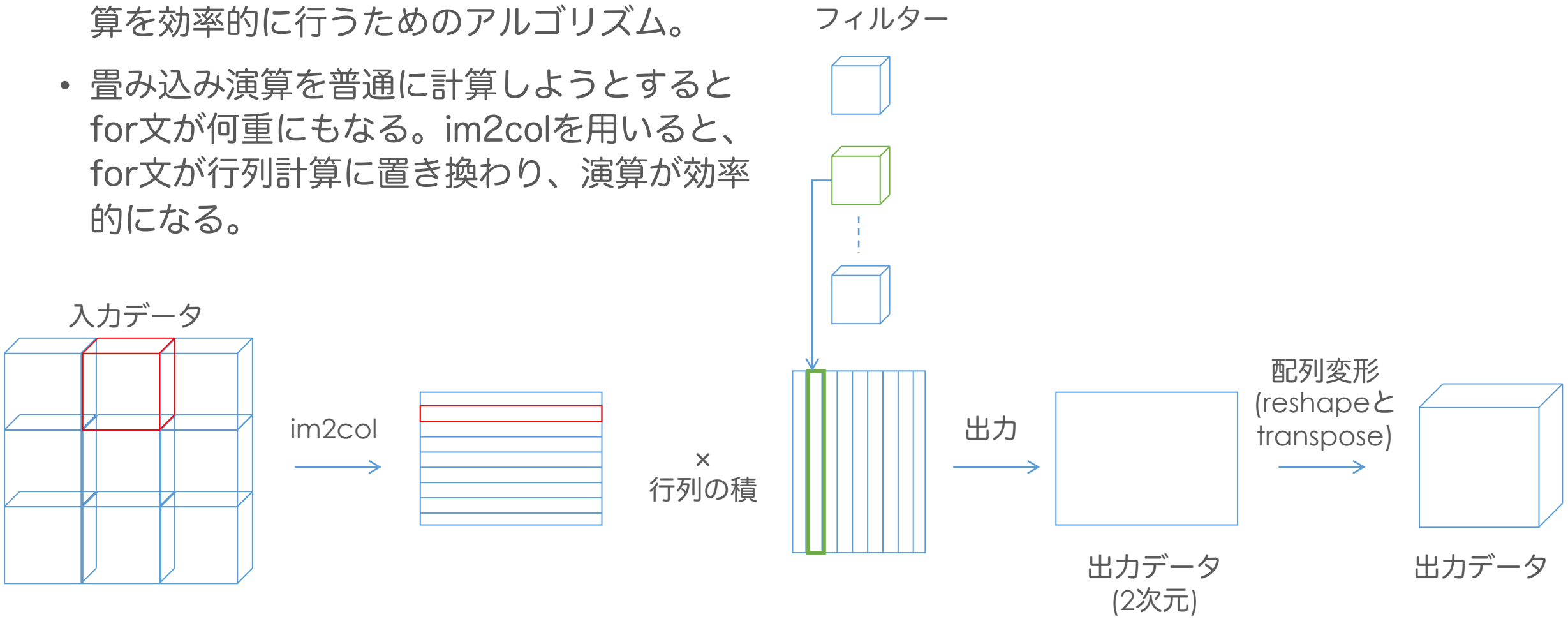
2019 SkillUP AI All rights reserved

im2col

2019 SkillUP AI All rights reserved

im2col

- im2col(image to column)は、畳み込み演算を効率的に行うためのアルゴリズム。
- 畳み込み演算を普通に計算しようとするとfor文が何重にもなる。im2colを用いると、for文が行列計算に置き換わり、演算が効率的になる。

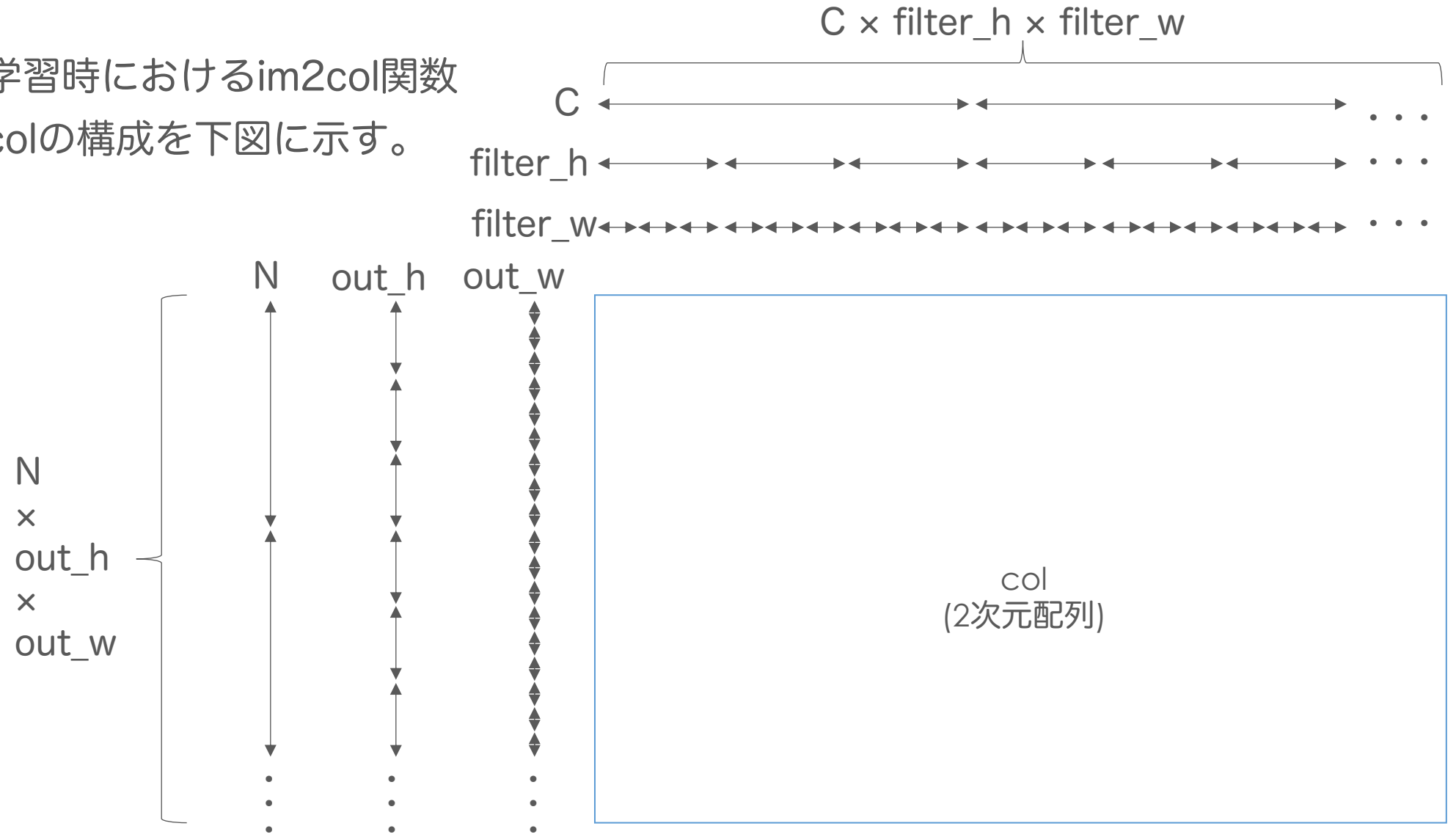


2019 SkillUP AI All rights reserved

im2col関数が返す配列

```
def im2col(input_data, filter_h, filter_w, stride=1, pad=0, constant_values=0):
 中略
 return col
```

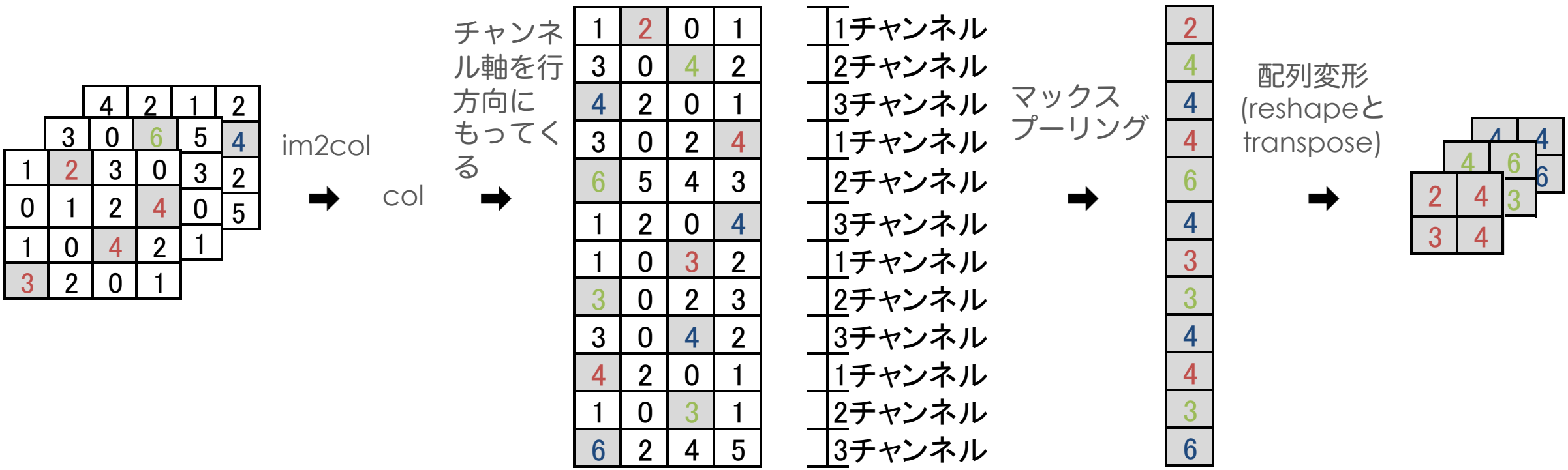
- ミニバッチ学習時におけるim2col関数が返す配列colの構成を下図に示す。



2019 SkillUP AI All rights reserved

プーリング演算時のim2col

- im2colは、プーリング演算にも用いることができる。



2019 SkillUP AI All rights reserved

[演習] im2colとcol2imの実装

- 4_3_padding_reshape_tanspose.ipynb
  - im2colの実装に必要になる関数の挙動を確認しましょう。
- 4_4_im2col_col2im_trainee.ipynb
  - im2colとcol2imを実装しましょう。

2019 SkillUP AI All rights reserved

[演習] 畳み込み層クラスの実装

- 4_5_convolution_layer_trainee.ipynb
  - 畳み込み層のクラスを実装しましょう。

2019 SkillUP AI All rights reserved

[演習] マックスプーリング層クラスの実装

- 4_6_maxpooling_layer_trainee.ipynb
  - マックスプーリング層のクラスを実装しましょう。

2019 SkillUP AI All rights reserved

[演習] シンプルなCNNの実装

- 4_7_simpleconvnet_trainee.ipynb
  - シンプルなCNNを実装し、MNIST問題を解いてみましょう。

2019 SkillUP AI All rights reserved

Any Questions?

2019 SkillUP AI All rights reserved

その他の話題

2019 SkillUP AI All rights reserved

目次

1. データ拡張
2. 構造出力
3. CNNで扱うデータの種類

2019 SkillUP AI All rights reserved

データ拡張

2019 SkillUP AI All rights reserved

データ拡張

- 学習用データを加工複製することをデータ拡張(augmentation)と呼ぶ。
  - 生成されたデータを学習に用いることで、汎化性能向上を狙う。
  - 特に、CNNの学習でよく行われる。
- 拡張方法の例
  - 回転、反転、伸縮、ノイズを加える、部分的に隠す、など
- データ拡張に使えるライブラリ
  - Augmentor
    - https://github.com/mdbloice/Augmentor
  - ImageDataGenerator
    - https://keras.io/preprocessing/image/
  - Imgaug
    - https://github.com/aleju/imgaug
  - Albumentations
    - https://github.com/albumentations-team/albumentations

2019 SkillUP AI All rights reserved

[演習] データ拡張の試行

- 4_8_Augmentor.ipynb
  - Augmentorを使ってデータ拡張を試行してみましょう。
- 4_9_ImageDataGenerator_sample.ipynb
  - ImageDataGeneratorを使って、データ拡張を試行してみましょう。

2019 SkillUP AI All rights reserved

構造出力

2019 SkillUP AI All rights reserved

構造出力

- 畳み込みニューラルネットワークでは、クラスラベルや実数値を出力するだけでなく、高次元な構造を持つオブジェクトを出力することもできる。
  - 『Ian Goodfellow,深層学習, 9.6節』では、このことを構造出力と呼んでいる。
- 例えば、入力画像の各ピクセルに対応するラベルを予測するセマンティックセグメンテーションというタスクがある。
  - 詳しくは、DAY5の「セマンティックセグメンテーションタスクとCNN」の節を参照されたい。

2019 SkillUP AI All rights reserved

CNNで扱うデータの種類

2019 SkillUP AI All rights reserved

CNNで扱うデータの種類

- CNNで扱えるデータは画像だけではない。
- 扱えるデータの例を以下に示す。
- 参考：『Ian Goodfellow, 深層学習, 9.7節』

	単一チャンネル	複数チャンネル
1次元	時系列データ (時間の1次元で、何らかの値が1チャンネル。音声データなど。畳み込みを行う軸は時間軸)	スケルトンアニメーションデータ (時間の1次元で、ある関節のある軸が1チャンネルになりそれが複数ある)
2次元	グレースケール画像データ (縦と横の2次元で、グレースケールの1チャネル) フーリエ変換で事前処理された時系列データ (周波数軸と時間軸の2次元で、スペクトルが1チャンネル)	カラー画像データ (縦と横の2次元で、RGBの3チャネル)
3次元	体積データ (縦と横と奥行きの3次元で、何らかの値が1チャンネル。CTスキャンのデータなど)	カラー動画データ (縦と横と時間の3次元で、RGBの3チャネル)

2019 SkillUP AI All rights reserved

Any Questions?

2019 SkillUP AI All rights reserved

[グループワーク] カタカナ15文字のデータ拡張

- カタカナ15文字の学習用データを用いて様々なデータ拡張を行ってみましょう。
- データ拡張ライブラリであるAugmentorとImageDataGeneratorの使い勝手を比較してみましょう。
- カタカナ15文字を分類するモデルを学習させる際は、どのようなデータ拡張が向いているか、考えてみましょう。
- 上記を予習段階で各自取り組みましょう。
- 予習段階の取り組み結果をグループで共有します。(45分)
- 最後に、グループごとに発表していただきます。 (15分)

2019 SkillUP AI All rights reserved

講座の時間が余ったら

- 今回の復習をします。
- 次回の予習をします。

2019 SkillUP AI All rights reserved

Appendix

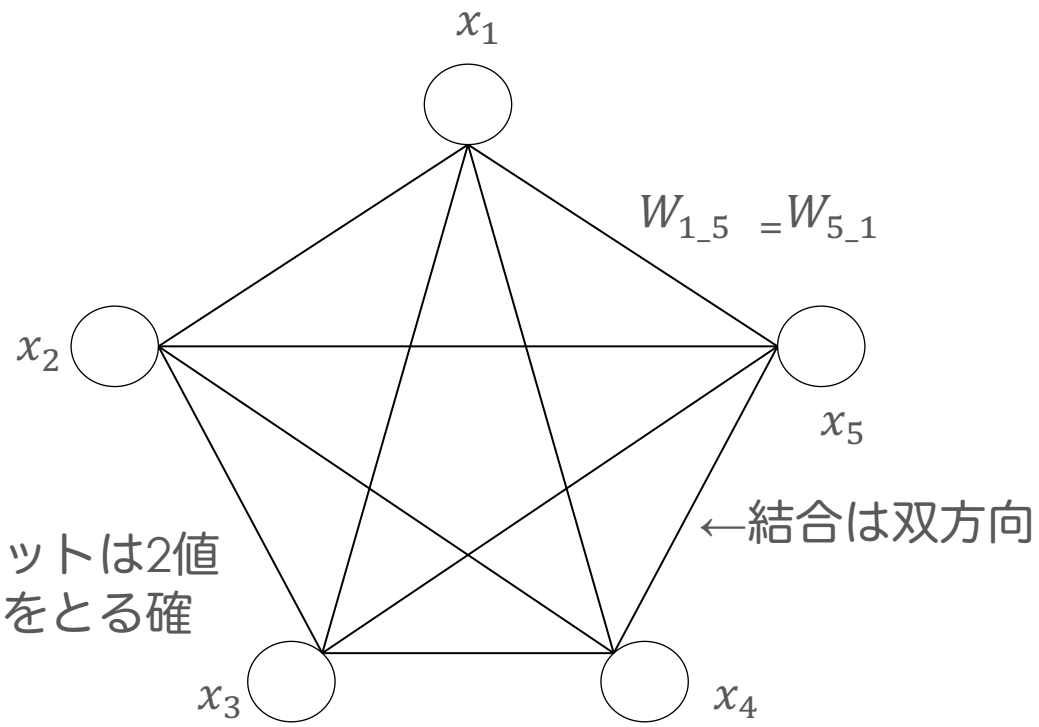
2019 SkillUP AI All rights reserved

ボルツマンマシンから深層ボルツマンマシンまで

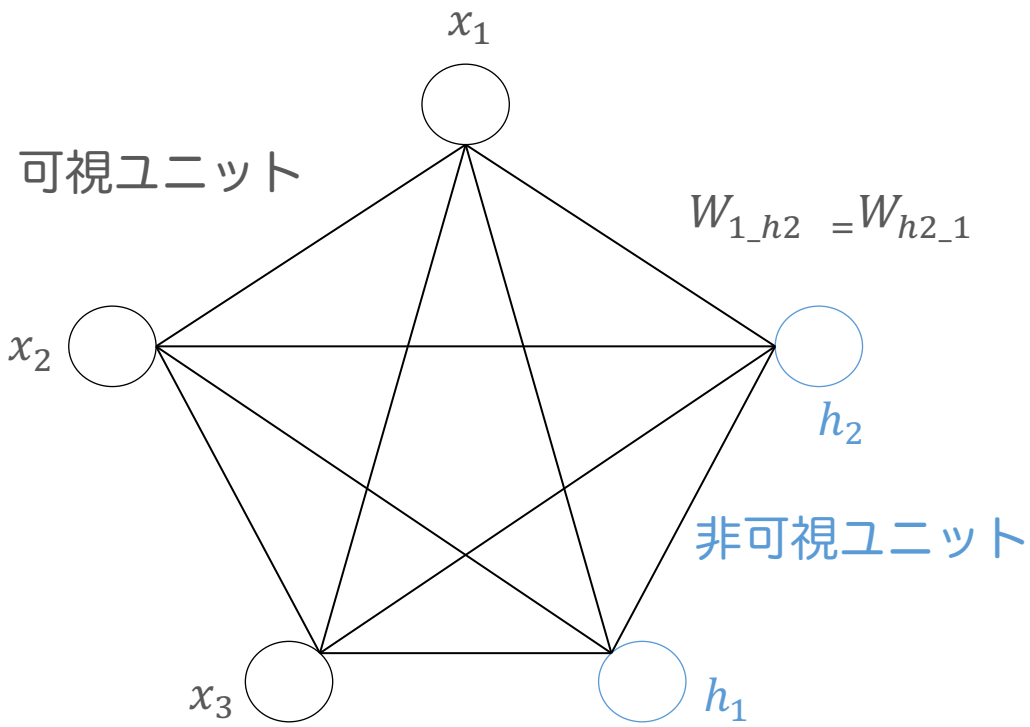
2019 SkillUP AI All rights reserved

ボルツマンマシン型の計算グラフ

ボルツマンマシンの例
(全てが可視ユニット)



隠れ層ユニットを持つ
ボルツマンマシンの例



2019 SkillUP AI All rights reserved

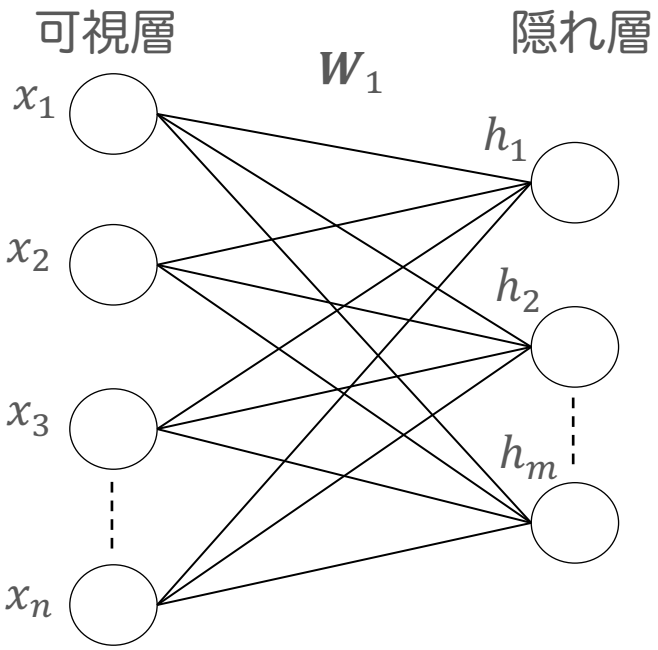
ボルツマンマシンとは

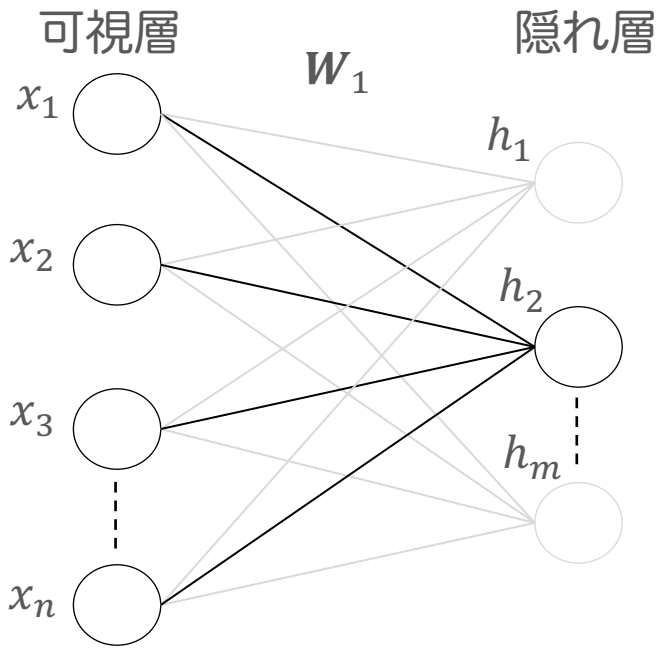
- ボルツマンマシン(Boltzmann machine)とは、ユニット間結合が双方向性を持つニューラルネットワークのこと。
- 1983年頃に提案された方法。
- 脳の連想記憶メカニズムに着想を得ている。
- ネットワークの挙動を確率的に記述することが特徴。
- ユニット間の結合重みは対称。例、$W_{1\_5} = W_{5\_1}$
- 各ユニットの出力は確率的に決まる。
- マルコフ確率場の1種。
- 生成モデルとして利用される。
- ボルツマンマシンを詳しく解説している書籍はこちら。
    - 『学習とニューラルネットワーク』(熊沢, 森北出版)

2019 SkillUP AI All rights reserved

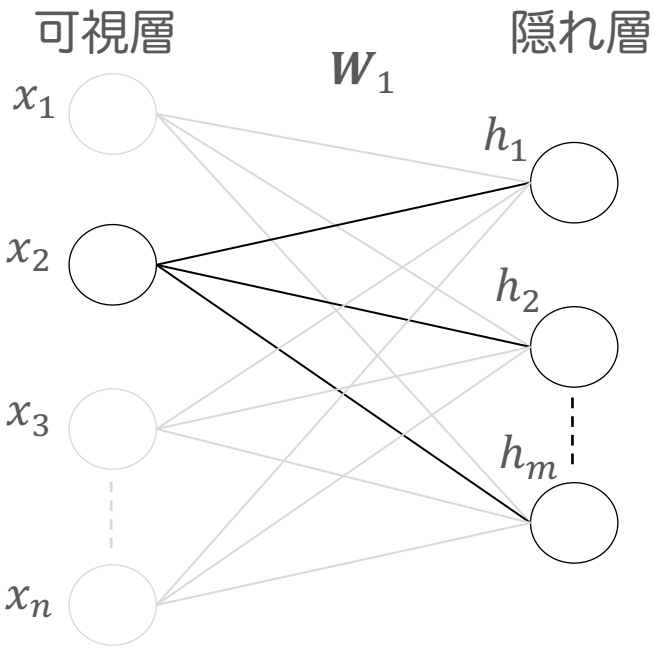
制約ボルツマンマシンの計算グラフ

制約ボルツマンマシンの例





可視層x_1~x_nが決まると、確率的に隠れ層h_2が決まる



隠れ層h_1~h_mが決まると、確率的に可視層x_2が決まる

2019 SkillUP AI All rights reserved

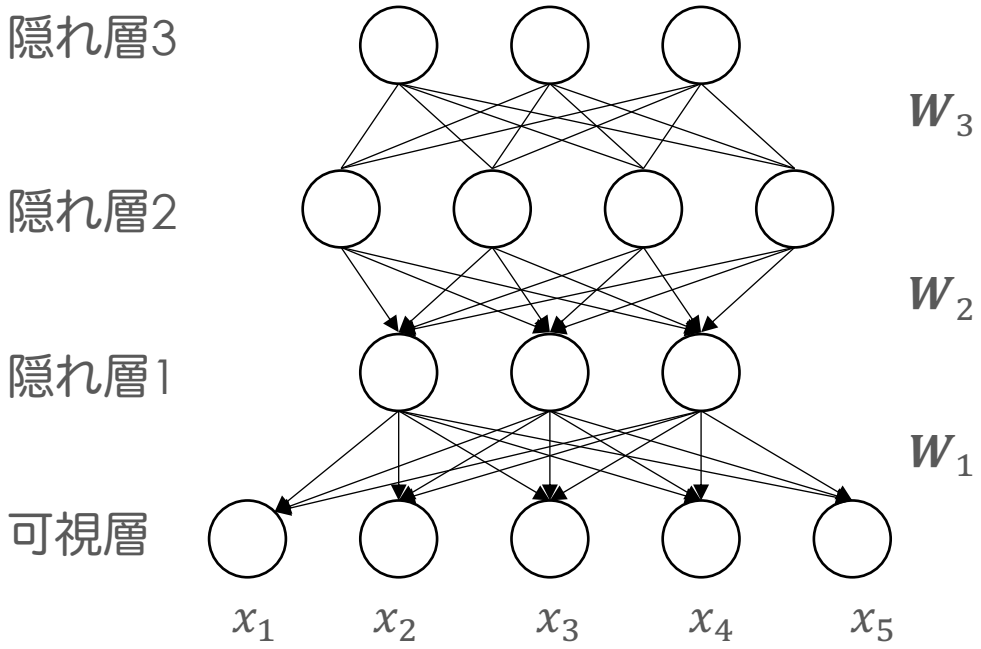
制約ボルツマンマシンとは

- 制約ボルツマンマシン(restricted Boltzmann machine)とは、隠れ変数を含むボルツマンマシンの一種。
- 1986年頃に提案された方法。
- 可視変数ユニットどうし、隠れ変数ユニットどうしの結合は持たない。
- 全ての結合は双方向。
- 自己符号化器と似た働きをもつ。
- 生成モデルとして利用されたり、自己符号化器のように隠れ層h_mを入力データの圧縮データとして扱うこともできる。
- 畳み込みニューラルネットワークの事前学習に用いられることもある。
- 深層信念ネットワークとは、制約ボルツマンマシンを多層化したもの。

2019 SkillUP AI All rights reserved

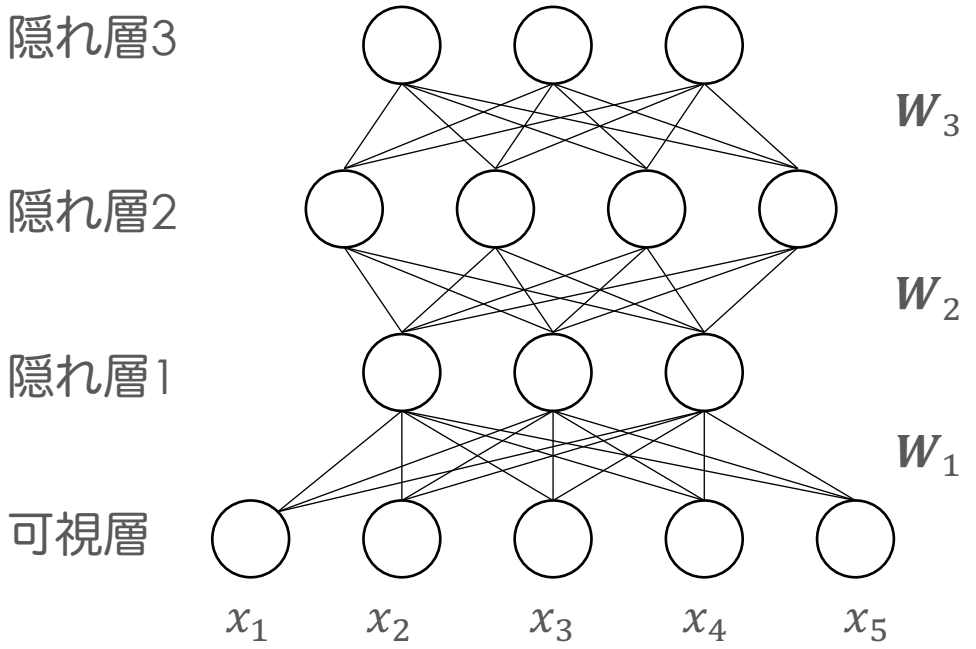
深層信念ネットワークと深層ボルツマンマシンの計算グラフ

深層信念ネットワークの計算グラフ例



最上位層のみ無向エッジ(双方向)で、
それ以外は有向エッジ

深層ボルツマンマシンの計算グラフ例



全ての結合が無向エッジ(双方向)

2019 SkillUP AI All rights reserved

深層信念ネットワークとは

- 深層信念ネットワーク(deep belief network, DBN)とは、隠れ変数を含むボルツマンマシンの一種。
- 2006年頃に提案された方法。
- 近年の深層学習ブームの契機となったモデル。
- 最上位層のみ無向エッジ(双方向)で、それ以外は有向エッジ。

2019 SkillUP AI All rights reserved

深層ボルツマンマシンとは

- 深層ボルツマンマシン(deep Boltzmann machine, DBM)とは、制約ボルツマンマシンを多層化したもの。
- 2009年頃に提案された方法。
- 全ての結合が無向エッジ(双方向) 。

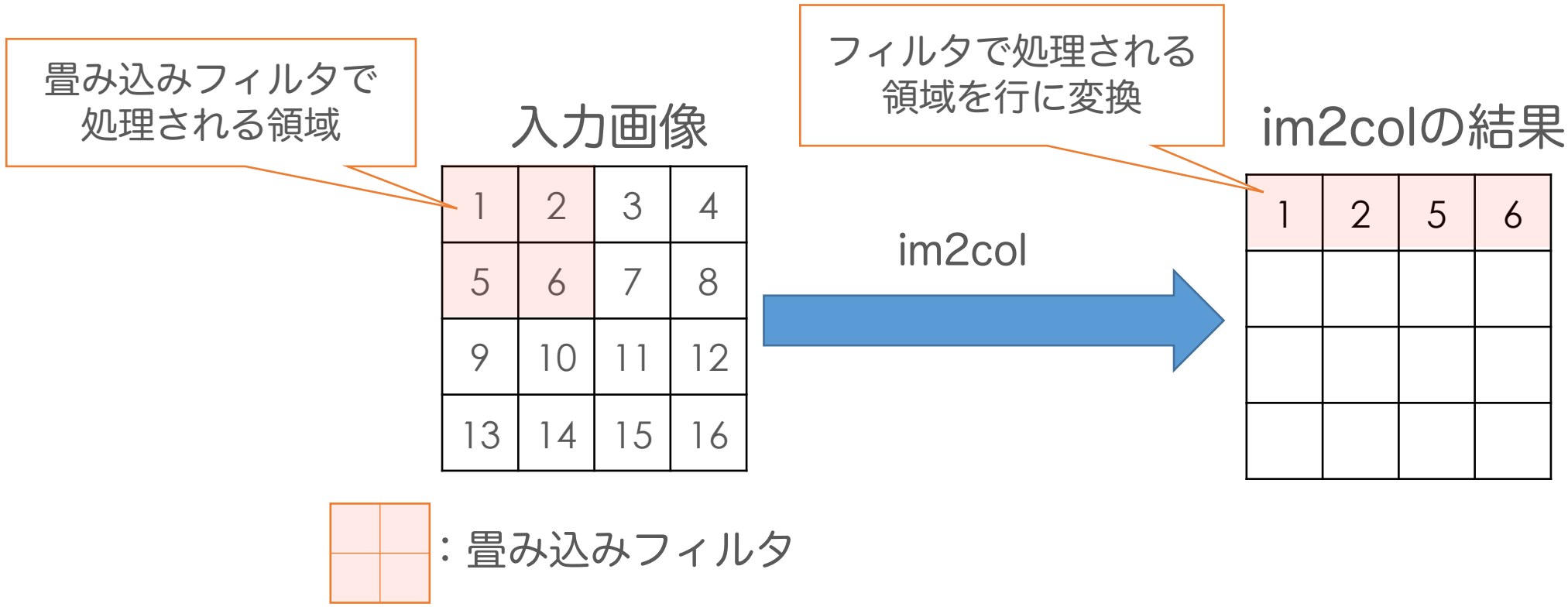
2019 SkillUP AI All rights reserved

im2colおよびcol2imに関する補足資料

2019 SkillUP AI All rights reserved

im2col

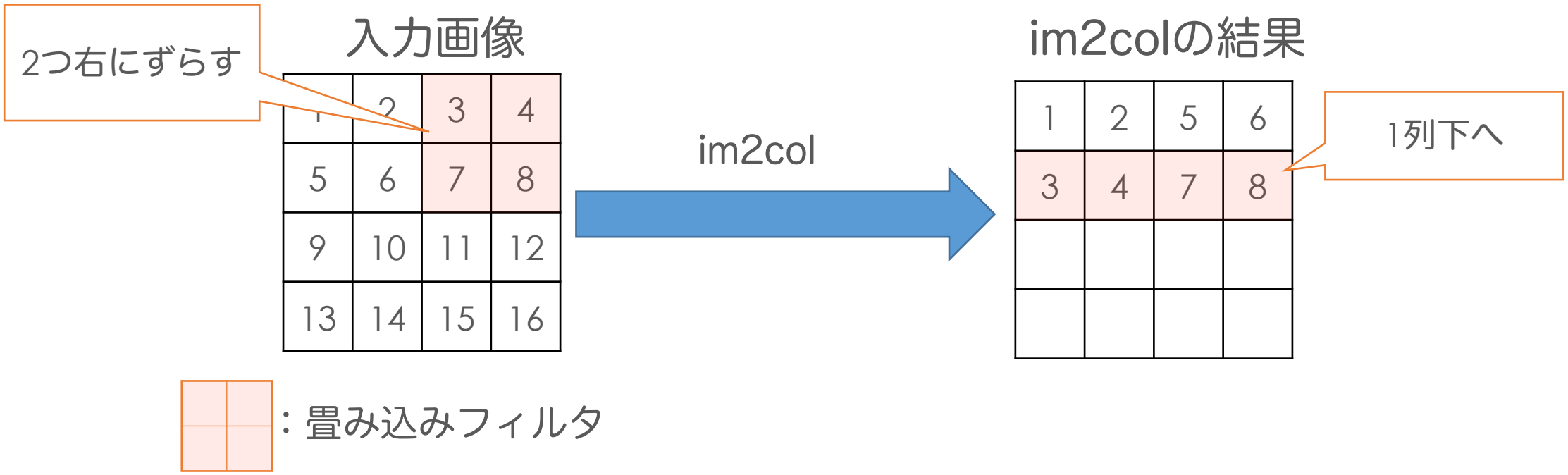
- image to columnsの略称で、畳み込み演算を一度の行列積で計算するために行う画像の変換方法のこと
- 画像サイズ4x4、フィルタサイズ2x2、ストライド2での例：



2019 SkillUP AI All rights reserved

im2col

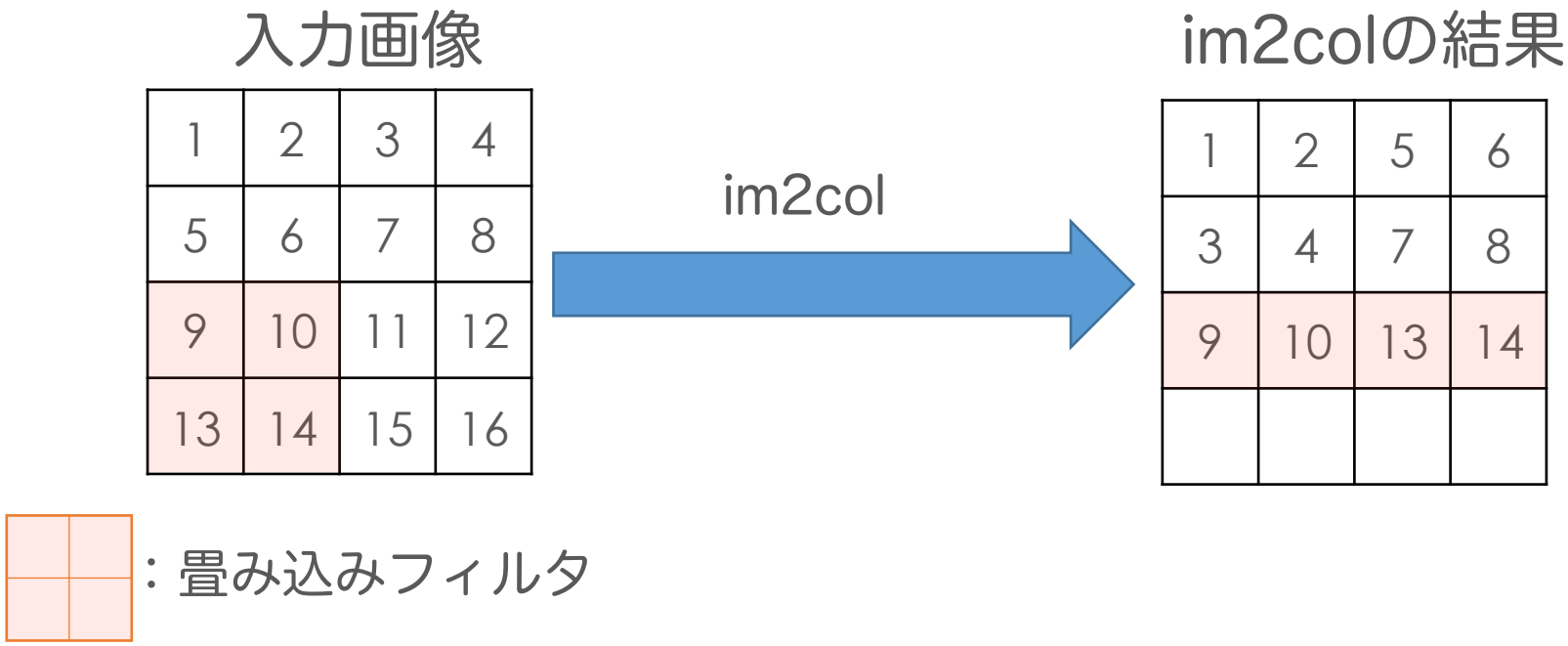
- Image to columnsの略称で、畳み込み演算を一度の行列積で計算するために行う画像の変換方法のこと
- 画像サイズ4x4、フィルタサイズ2x2、ストライド2での例：



2019 SkillUP AI All rights reserved

im2col

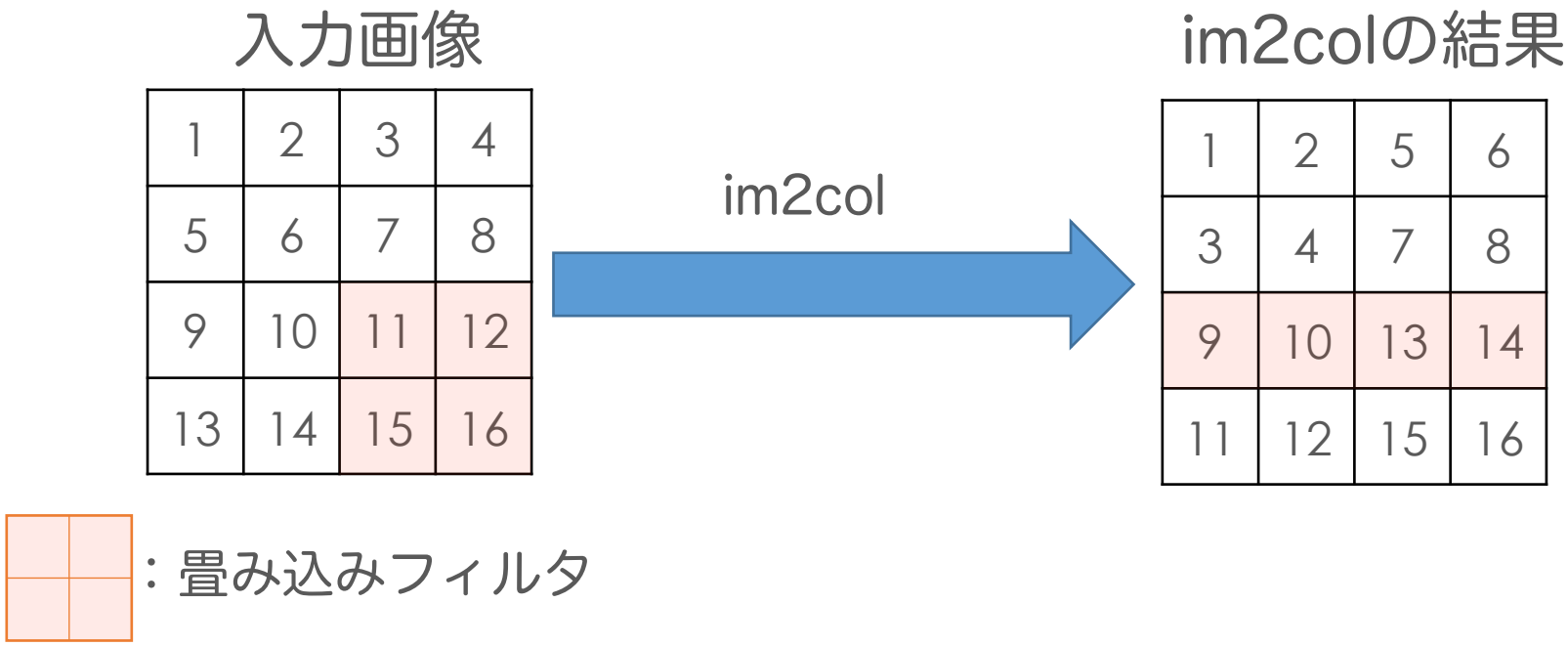
- Image to columnsの略称で、畳み込み演算を一度の行列積で計算するために行う画像の変換方法のこと
- 画像サイズ4x4、フィルタサイズ2x2、ストライド2での例：



2019 SkillUP AI All rights reserved

im2col

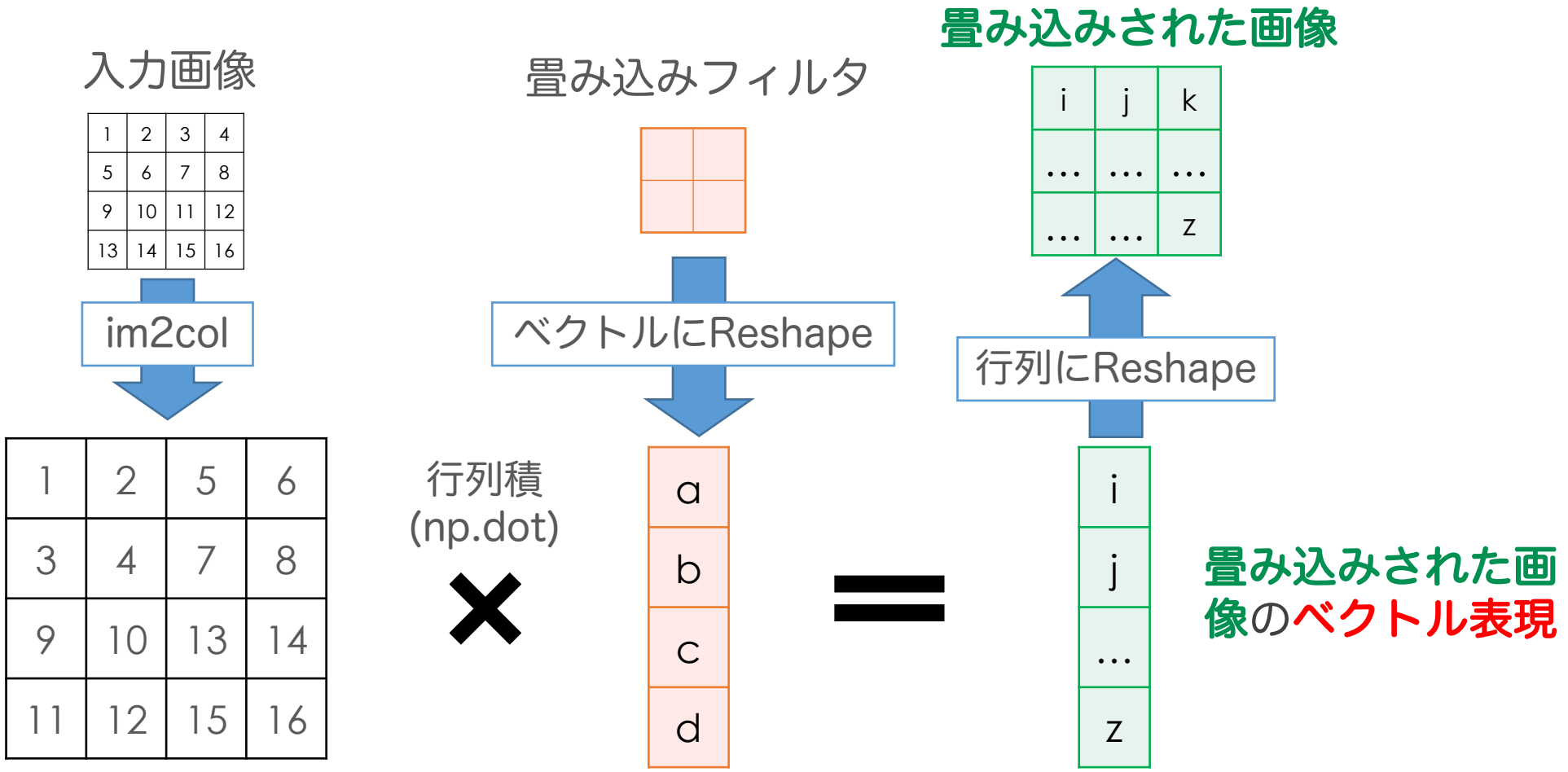
- Image to columnsの略称で、畳み込み演算を一度の行列積で計算するために行う画像の変換方法のこと
- 画像サイズ4x4、フィルタサイズ2x2、ストライド2での例：



2019 SkillUP AI All rights reserved

im2colと畳み込み演算

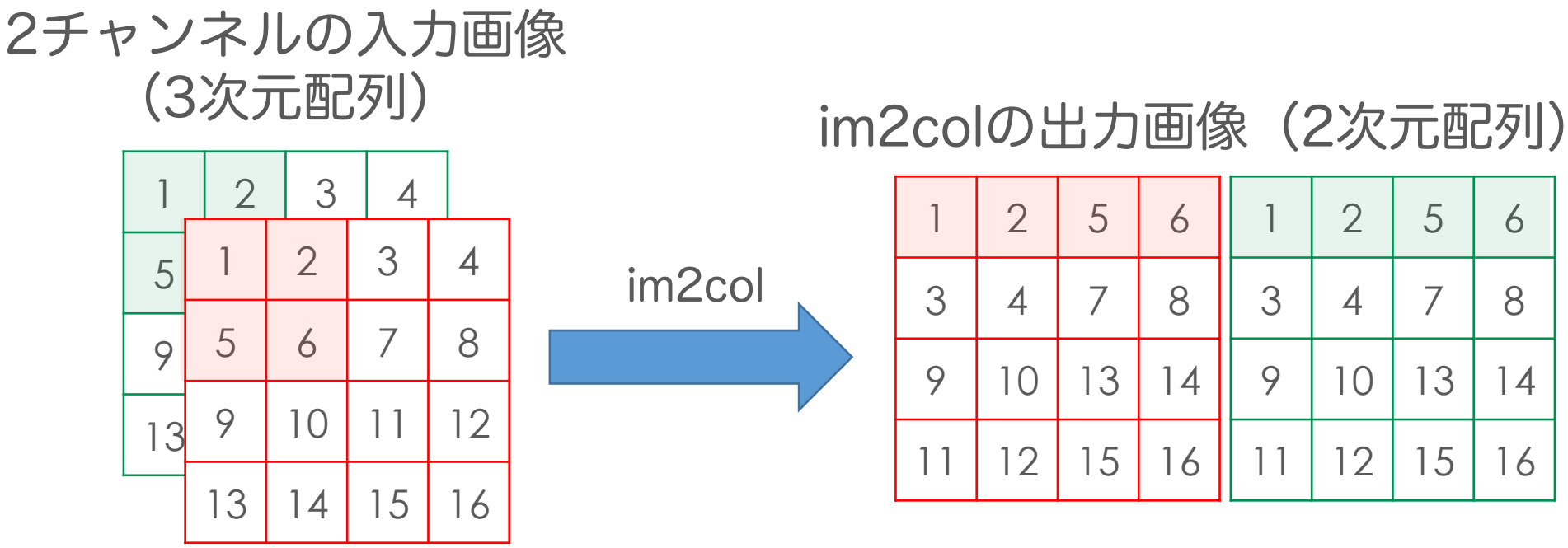
- im2colによって得られた画像とベクトルに変換した畳み込みフィルタの積を取ると畳み込みの結果が得られる



2019 SkillUP AI All rights reserved

im2colの多チャンネル拡張

- Im2colを多チャンネルに拡張するときは、チャンネル数に応じて列数を増やせばよい

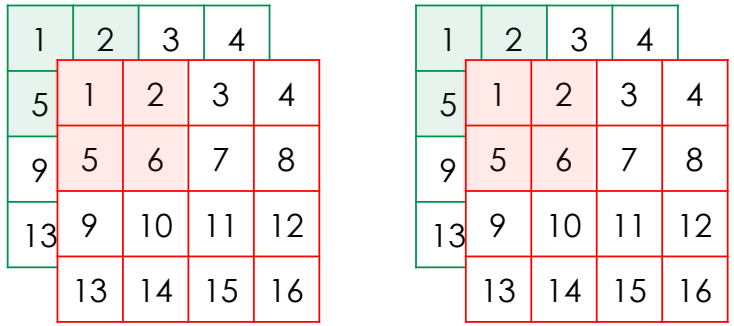


2019 SkillUP AI All rights reserved

im2colのバッチ処理への拡張

- さらにバッチ処理に拡張する場合は、バッチサイズに応じて行数を増やす

2チャンネルの入力画像2枚
（4次元配列）



im2col

im2colの出力画像（2次元配列）

1	2	5	6
3	4	7	8
9	10	13	14
11	12	15	16

1	2	5	6
3	4	7	8
9	10	13	14
11	12	15	16

1枚目

1	2	5	6
3	4	7	8
9	10	13	14
11	12	15	16

1	2	5	6
3	4	7	8
9	10	13	14
11	12	15	16

2枚目

2019 SkillUP AI All rights reserved

im2colの出力画像サイズ

- im2colの出力画像サイズは以下の式で求めることができる：

im2colの出力画像サイズ公式

$$\text{出力画像の高さ：} H_{\text{im2col}} = N \times H_{\text{out}} \times W_{\text{out}}$$

$$\text{出力画像の幅：} W_{\text{im2col}} = C \times H_{\text{filter}} \times W_{\text{filter}}$$

Cは入力画像のチャンネル数、 $H_{\text{filter}}, W_{\text{filter}}$はそれぞれフィルタの高さと幅、
Nはバッチサイズ、$H_{\text{out}}, W_{\text{out}}$はそれぞれ畳み込み後の出力画像の高さと幅を意味する

2019 SkillUP AI All rights reserved

畳み込み・プーリング後の出力画像サイズ

- 畳み込みあるいはプーリング後の出力画像サイズの高さH_{out}と幅W_{out}は以下の式で求めることができる：

畳み込み後の出力画像サイズ公式

$$\text{出力画像の高さ：} H_{\text{out}} = \text{floor}\left(\frac{H+2p-H_{\text{filter}}}{s}\right)+1$$

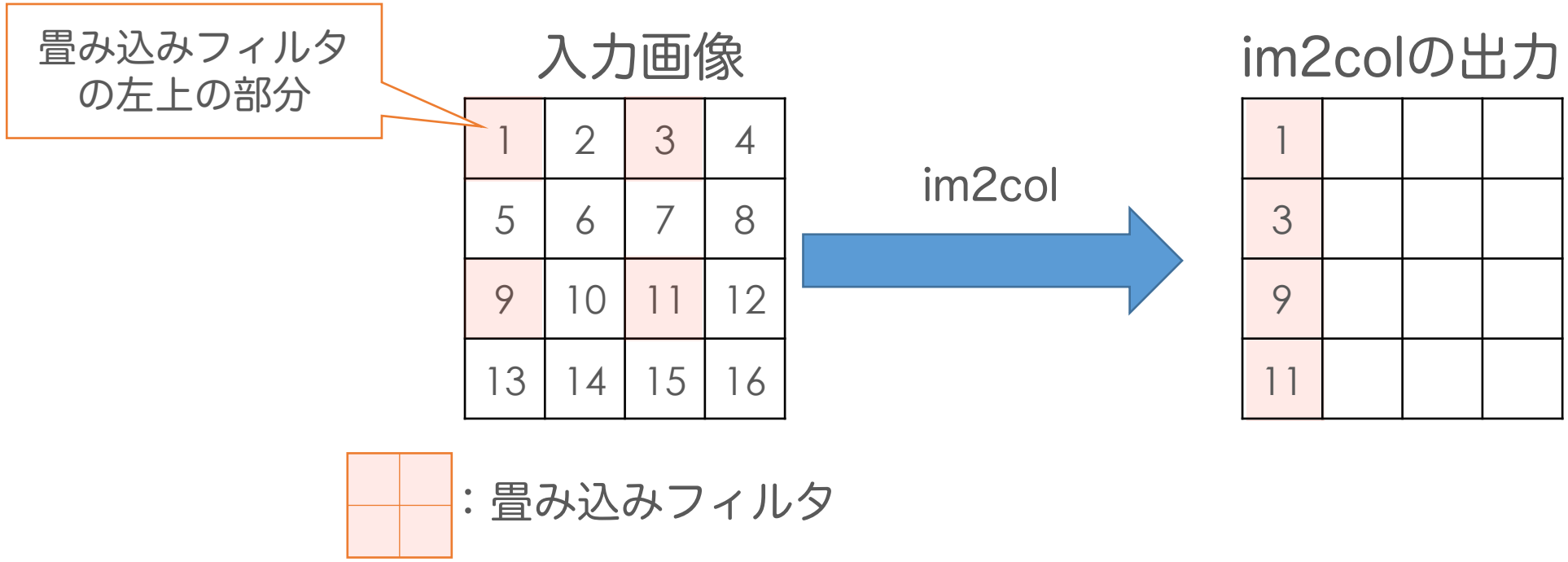
$$\text{出力画像の幅：} W_{\text{out}} = \text{floor}\left(\frac{W+2p-W_{\text{filter}}}{s}\right)+1$$

H, Wはそれぞれ入力画像の高さと幅、
$H_{\text{filter}}, W_{\text{filter}}$はそれぞれフィルタの高さと幅、$p$はパディングサイズ、
sはストライド、関数floorは小数点以下を切り捨てる関数を意味する

2019 SkillUP AI All rights reserved

スライスを用いたim2colの効率的な実装

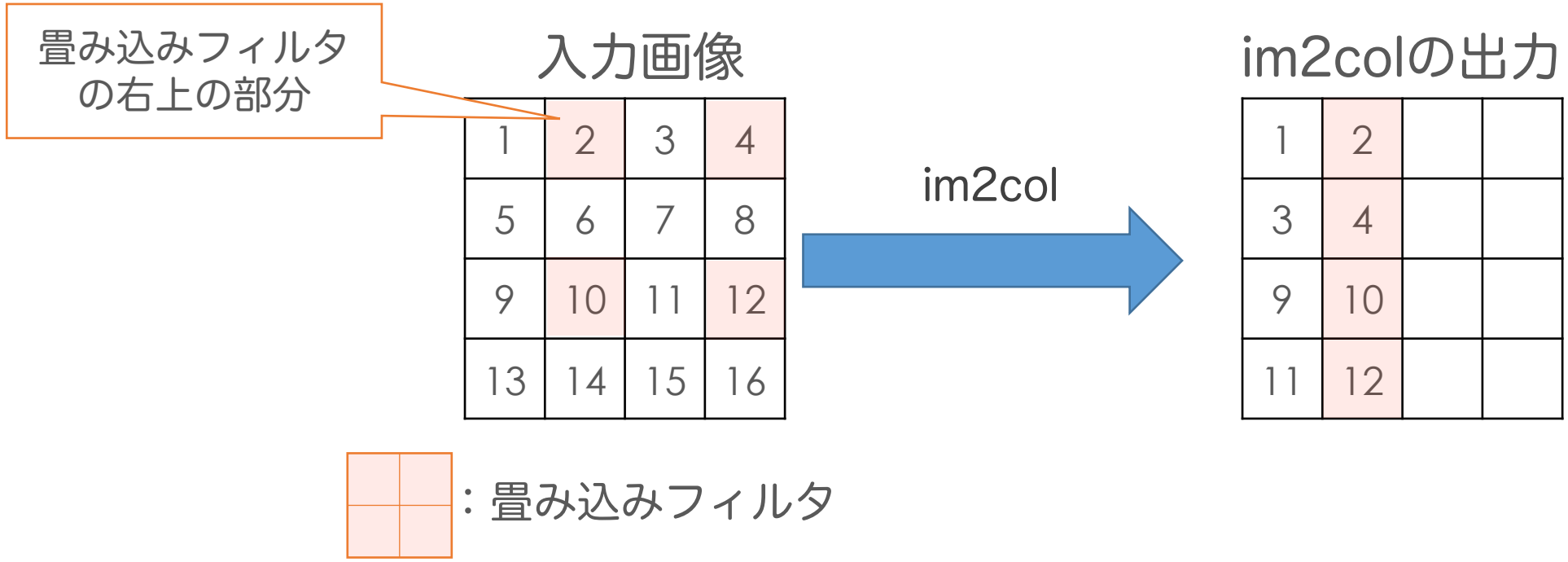
- そのまま実装すると入力画像が大きくなると処理が増えて非効率
- Notebookではスライスを用いることで、im2colを効率的に実装している
- 画像サイズ4x4、フィルタサイズ2x2、ストライド2での例：



2019 SkillUP AI All rights reserved

スライスを用いたim2colの効率的な実装

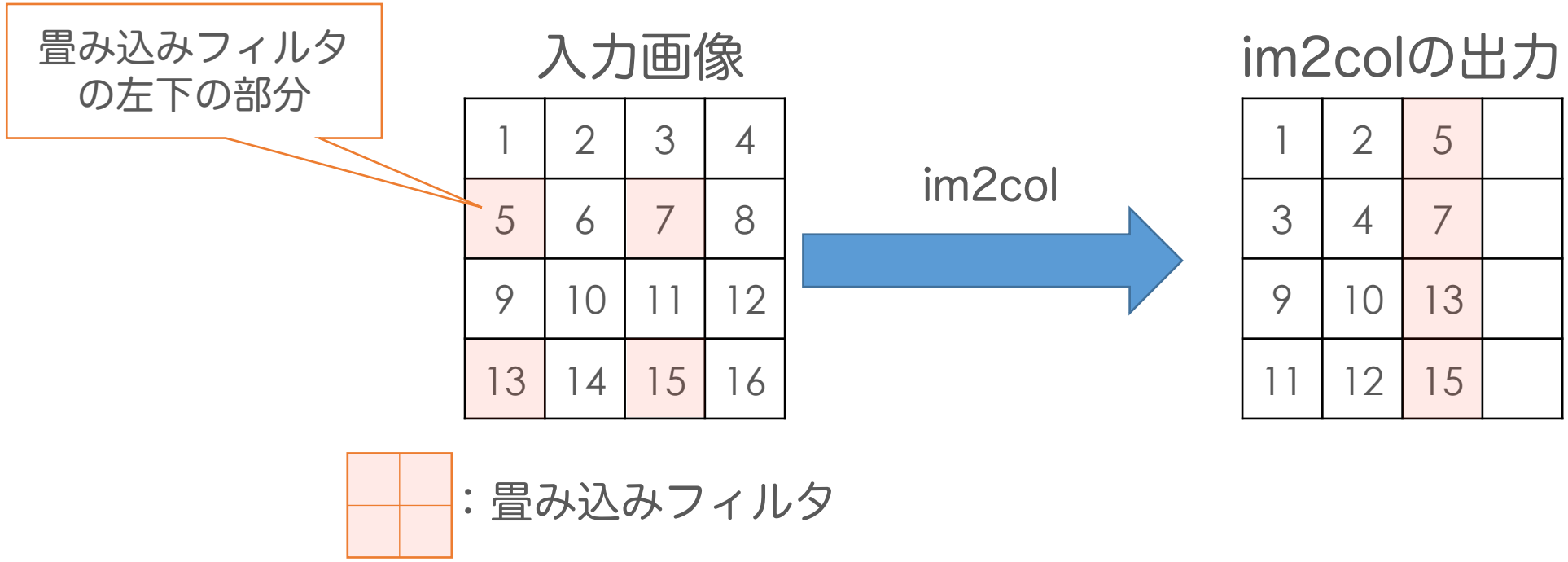
- そのまま実装すると入力画像が大きくなると処理が増えて非効率
- Notebookではスライスを用いることで、im2colを効率的に実装している
- 画像サイズ4x4、フィルタサイズ2x2、ストライド2での例：



2019 SkillUP AI All rights reserved

スライスを用いたim2colの効率的な実装

- そのまま実装すると入力画像が大きくなると処理が増えて非効率
- Notebookではスライスを用いることで、im2colを効率的に実装している
- 画像サイズ4x4、フィルタサイズ2x2、ストライド2での例：



2019 SkillUP AI All rights reserved

スライスを用いたim2colの効率的な実装

- そのまま実装すると入力画像が大きくなると処理が増えて非効率
- Notebookではスライスを用いることで、im2colを効率的に実装している
- 画像サイズ4x4、フィルタサイズ2x2、ストライド2での例：

畳み込みフィルタの右下の部分

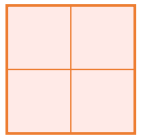
入力画像

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

im2col

im2colの出力

1	2	5	6
3	4	7	8
9	10	13	14
11	12	15	16

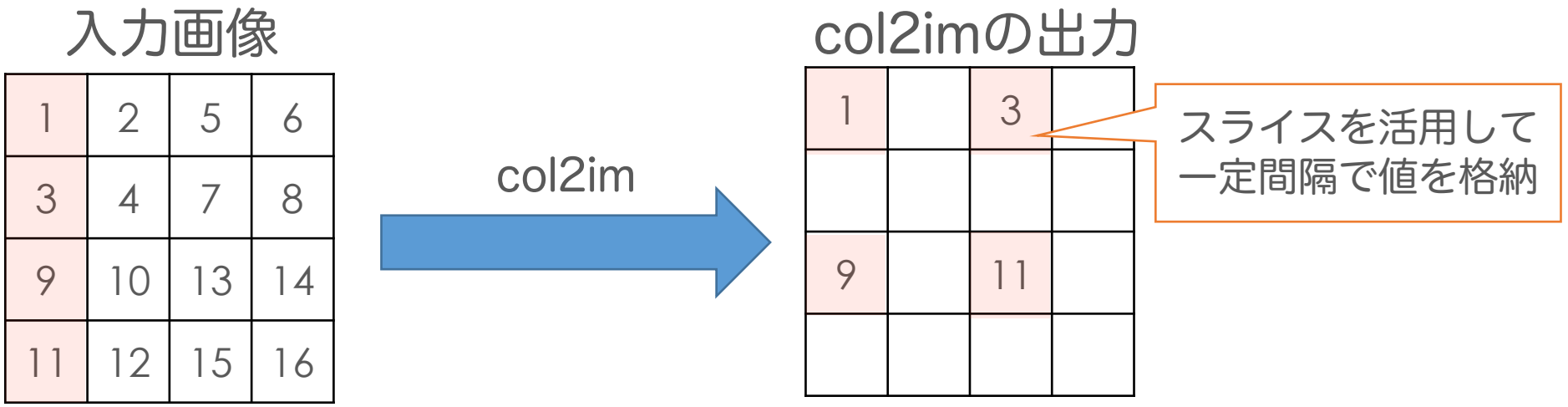

：畳み込みフィルタ

スライスを用いれば画像のサイズに依らずにフィルタの要素数回の処理でim2colが実装可能

2019 SkillUP AI All rights reserved

col2im

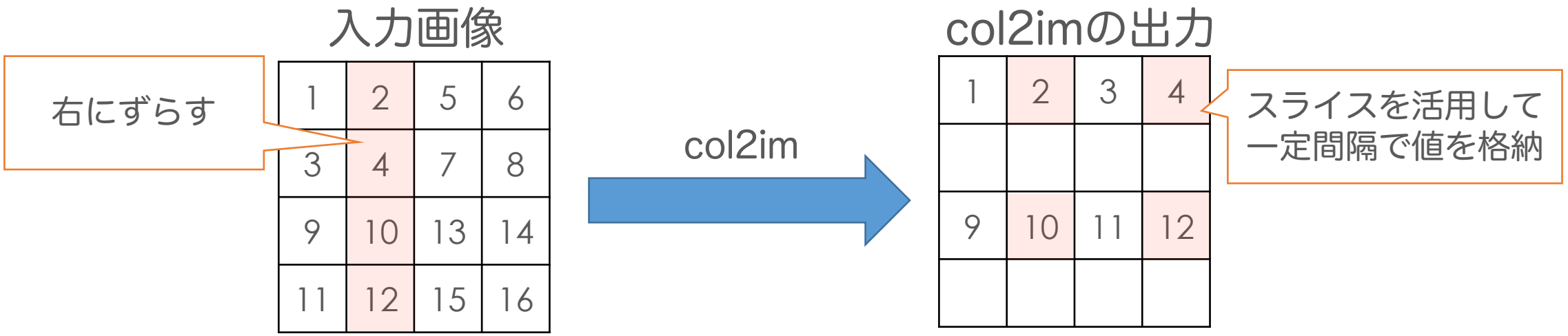
- Columns to imageの略称で、im2colの逆操作として定義
- im2colが順伝播計算、col2imが逆伝播計算のときにそれぞれ使用される
- im2colと同じくスライスによって効率的に実装可能



2019 SkillUP AI All rights reserved

col2im

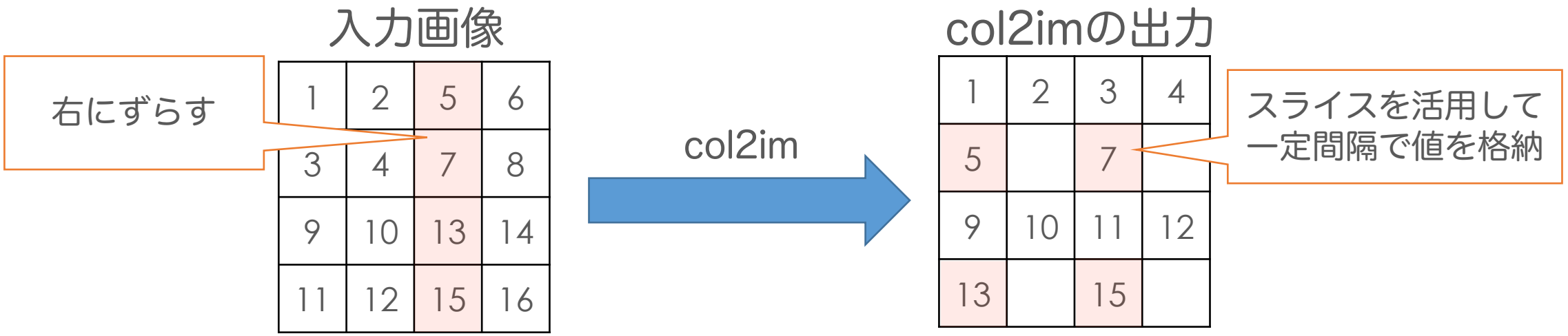
- Columns to imageの略称で、im2colの逆操作として定義
- Im2colが順伝播、col2imが逆伝播のときにそれぞれ使用される
- Im2colと同じくスライスによって効率的に実装可能



2019 SkillUP AI All rights reserved

col2im

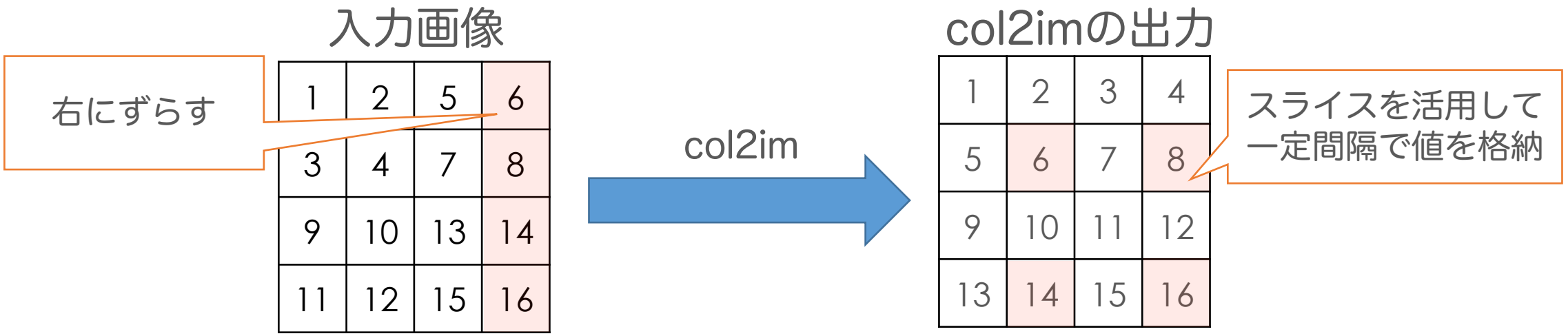
- Columns to imageの略称で、Im2colの逆操作として定義
- Im2colが順伝播、col2imが逆伝播のときにそれぞれ使用される
- Im2colと同じくスライスによって効率的に実装可能



2019 SkillUP AI All rights reserved

col2im

- Columns to imageの略称で、im2colの逆操作として定義
- im2colが順伝播、col2imが逆伝播のときにそれぞれ使用される
- im2colと同じくスライスによって効率的に実装可能



2019 SkillUP AI All rights reserved

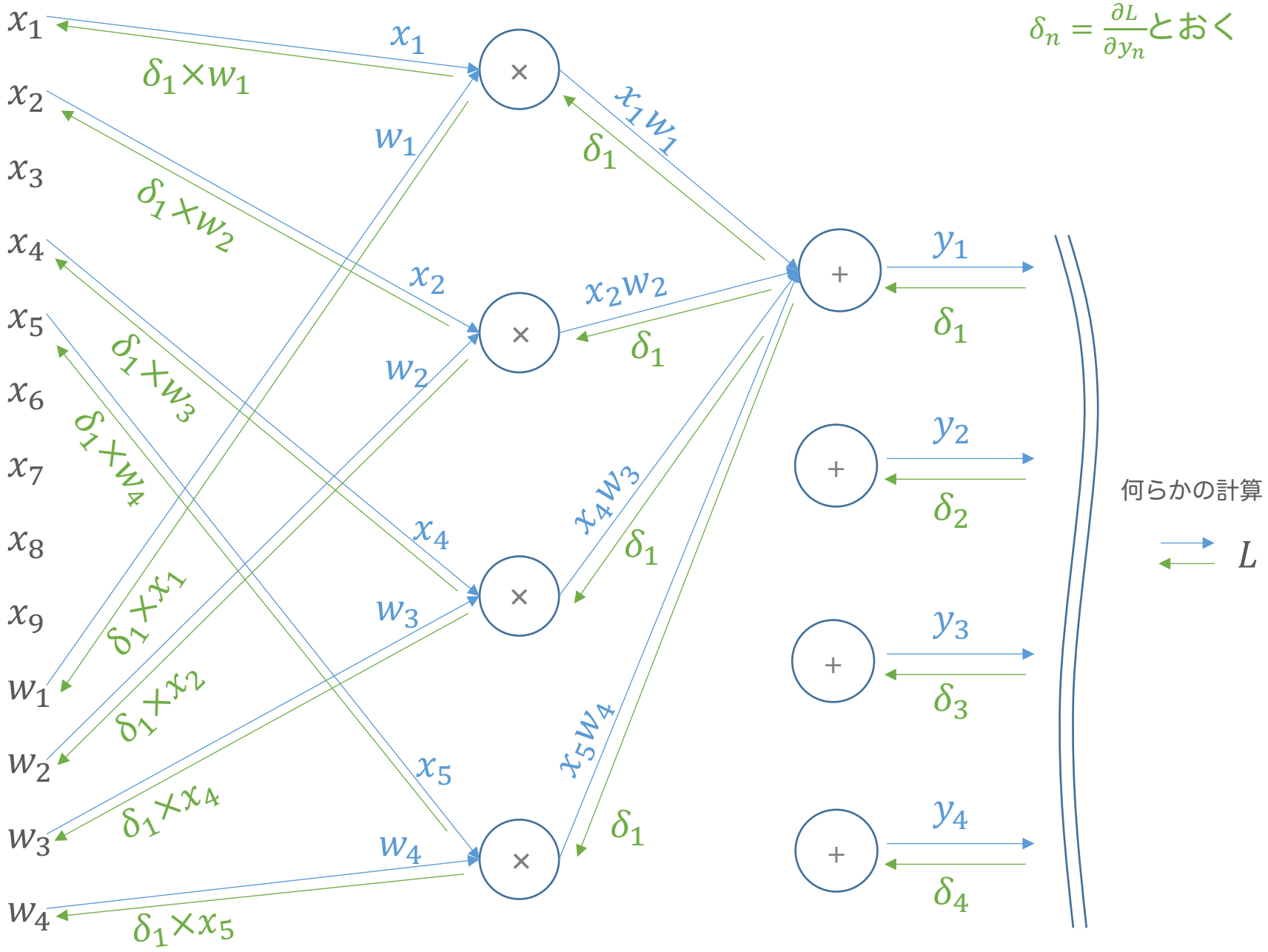
畳み込みレイヤおよびマックスプーリングレイヤに関する補足資料

2019 SkillUP AI All rights reserved

畳み込みレイヤ

- 畳み込みレイヤの計算グラフ例を右記に示す。
- バッチ版の考え方は、アフィンレイヤが参考になる。

<条件>
入力データの高さ=3
入力データの幅=3
フィルタの高さ=2
フィルタの幅=2
ストライド=1
パッド=0
バイアスは省略

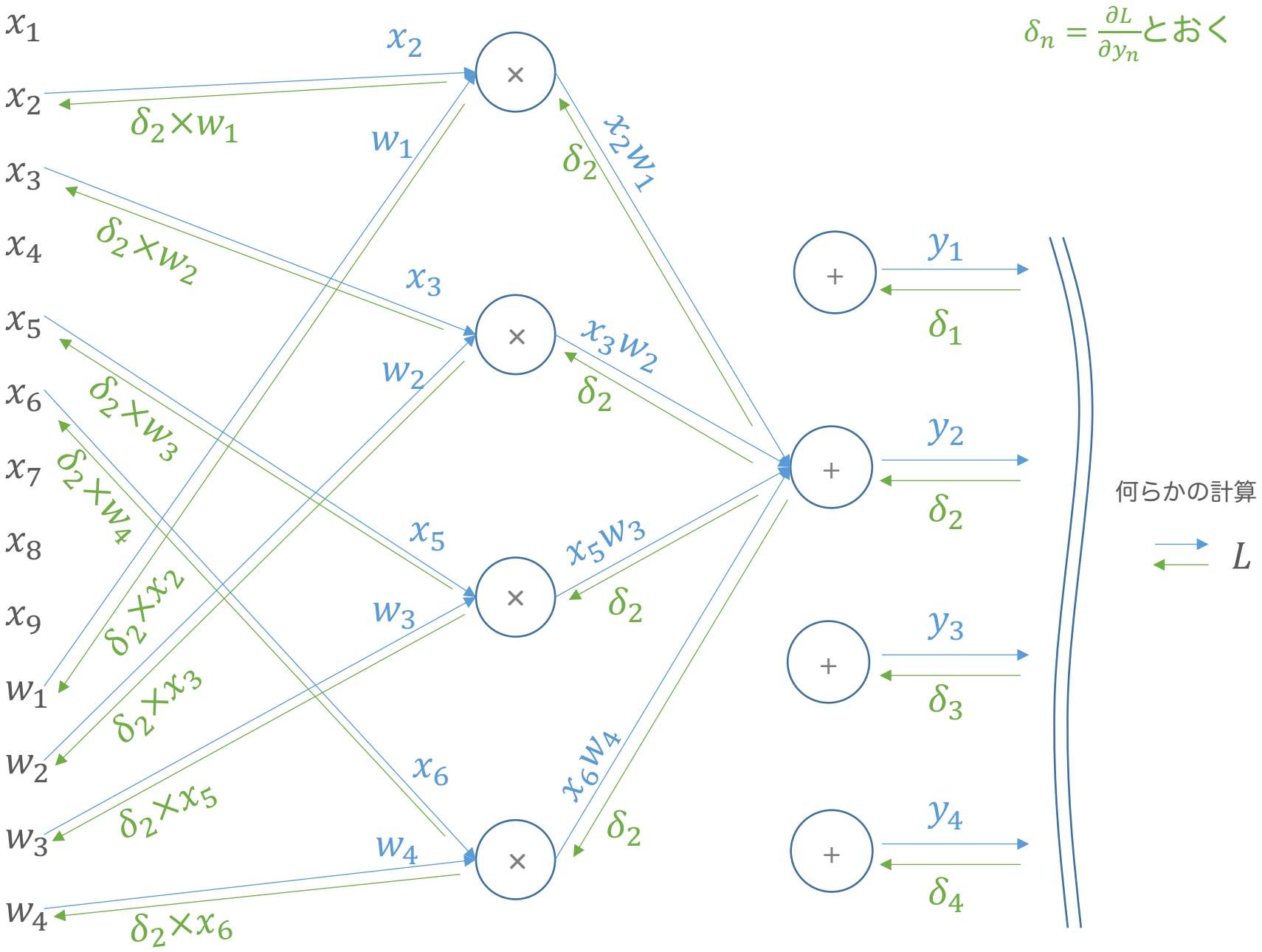


2019 SkillUP AI All rights reserved

畳み込みレイヤ

- 畳み込みレイヤの計算グラフ例を右記に示す。
- バッチ版の考え方は、アフィンレイヤが参考になる。

<条件>
入力データの高さ=3
入力データの幅=3
フィルタの高さ=2
フィルタの幅=2
ストライド=1
パッド=0
バイアスは省略

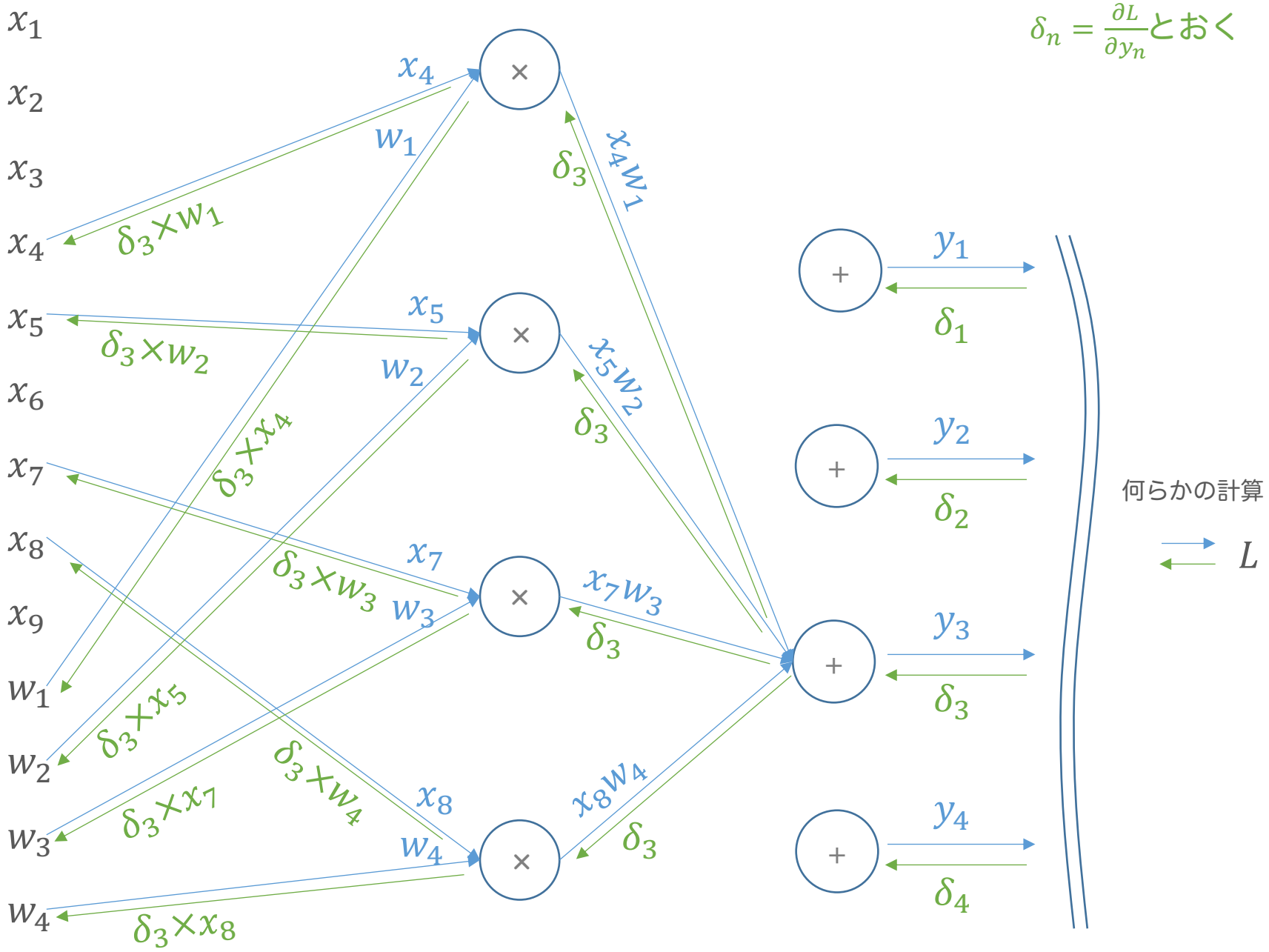


2019 SkillUP AI All rights reserved

畳み込みレイヤ

- 畳み込みレイヤの計算グラフ例を右記に示す。
- バッチ版の考え方は、アフィンレイヤが参考になる。

<条件>
入力データの高さ=3
入力データの幅=3
フィルタの高さ=2
フィルタの幅=2
ストライド=1
パッド=0
バイアスは省略

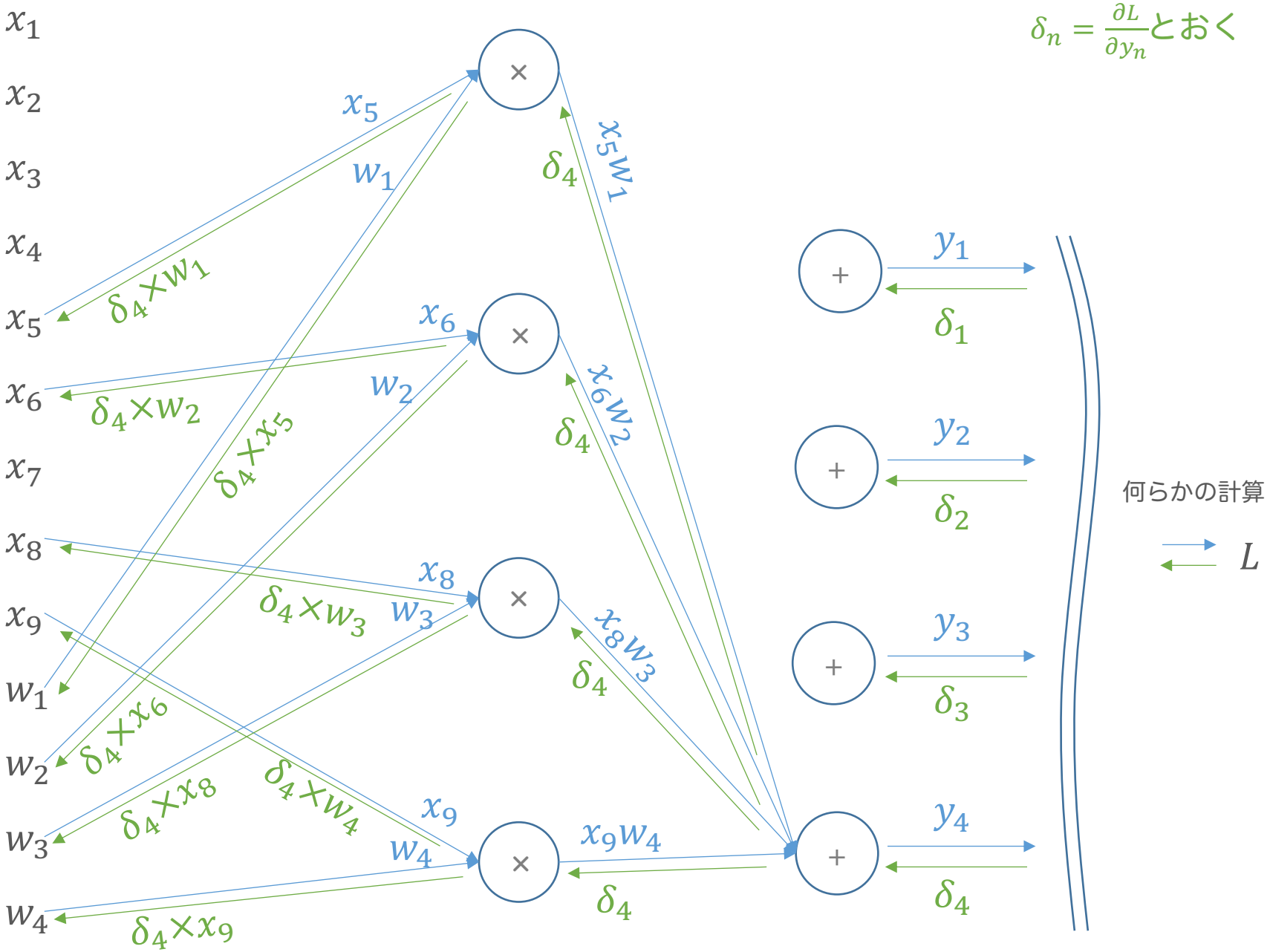


2019 SkillUP AI All rights reserved

畳み込みレイヤ

- 畳み込みレイヤの計算グラフ例を右記に示す。
- バッチ版の考え方は、アフィンレイヤが参考になる。

<条件>
入力データの高さ=3
入力データの幅=3
フィルタの高さ=2
フィルタの幅=2
ストライド=1
パッド=0
バイアスは省略



2019 SkillUP AI All rights reserved

畳み込みレイヤ

- 畳み込みレイヤの計算グラフ例を右記に示す。
- バッチ版の考え方は、アフィンレイヤが参考になる。

<条件>
入力データの高さ=3
入力データの幅=3
フィルタの高さ=2
フィルタの幅=2
ストライド=1
パッド=0
バイアスは省略

$$x_1 \xrightarrow{x_1} \xleftarrow{\partial L/\partial x_1 = \delta_1 \times w_1}$$
$$x_2 \xrightarrow{x_2} \xleftarrow{\partial L/\partial x_2 = \delta_1 \times w_2 + \delta_2 \times w_1}$$
$$x_3 \xrightarrow{x_3} \xleftarrow{\partial L/\partial x_3 = \delta_2 \times w_2}$$
$$x_4 \xrightarrow{x_4} \xleftarrow{\partial L/\partial x_4 = \delta_1 \times w_3 + \delta_3 \times w_1}$$
$$x_5 \xrightarrow{x_5} \xleftarrow{\partial L/\partial x_5 = \delta_1 \times w_4 + \delta_2 \times w_3 + \delta_3 \times w_2 + \delta_4 \times w_1}$$
$$x_6 \xrightarrow{x_6} \xleftarrow{\partial L/\partial x_6 = \delta_2 \times w_4 + \delta_4 \times w_2}$$
$$x_7 \xrightarrow{x_7} \xleftarrow{\partial L/\partial x_7 = \delta_3 \times w_3}$$
$$x_8 \xrightarrow{x_8} \xleftarrow{\partial L/\partial x_8 = \delta_3 \times w_4 + \delta_4 \times w_3}$$
$$x_9 \xrightarrow{x_9} \xleftarrow{\partial L/\partial x_9 = \delta_4 \times w_4}$$
$$w_1 \xrightarrow{w_1} \xleftarrow{\partial L/\partial w_1 = \delta_1 \times x_1 + \delta_2 \times x_2 + \delta_3 \times x_4 + \delta_4 \times x_5}$$
$$w_2 \xrightarrow{w_2} \xleftarrow{\partial L/\partial w_2 = \delta_1 \times x_2 + \delta_2 \times x_3 + \delta_3 \times x_5 + \delta_4 \times x_6}$$
$$w_3 \xrightarrow{w_3} \xleftarrow{\partial L/\partial w_3 = \delta_1 \times x_4 + \delta_2 \times x_5 + \delta_3 \times x_7 + \delta_4 \times x_8}$$
$$w_4 \xrightarrow{w_4} \xleftarrow{\partial L/\partial w_4 = \delta_1 \times x_5 + \delta_2 \times x_6 + \delta_3 \times x_8 + \delta_4 \times x_9}$$

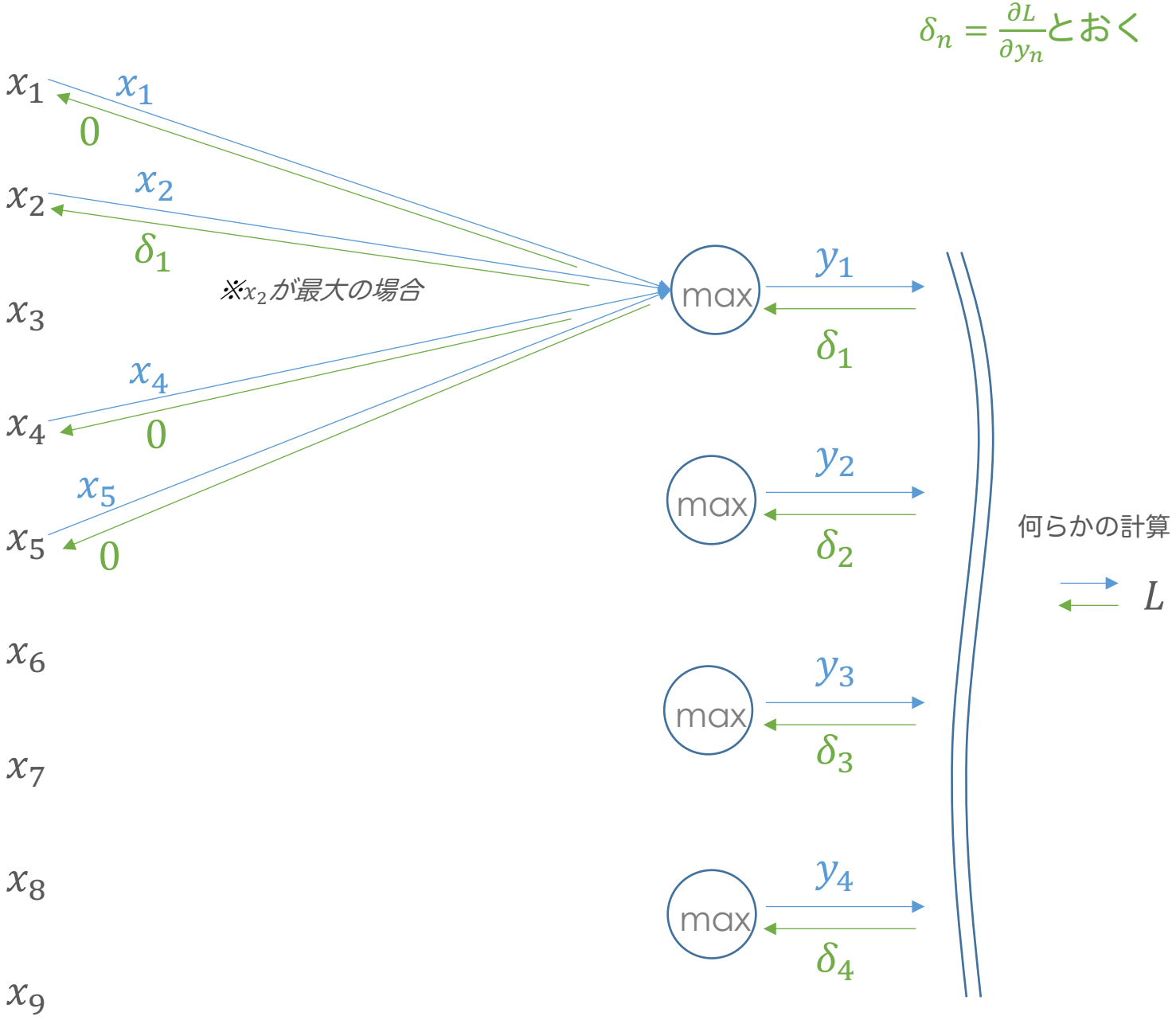
x_n毎、w_n毎に勾配を足し合わせる

2019 SkillUP AI All rights reserved

マックスプーリングレイヤ

- マックスプーリングレイヤの計算グラフ例を右記に示す。
- バッチ版の考え方は、アフィンレイヤとドロップアウトレイヤが参考になる。

<条件>
入力データの高さ=3
入力データの幅=3
プーリング窓の高さ=2
プーリング窓の幅=2
ストライド=1
パッド=0

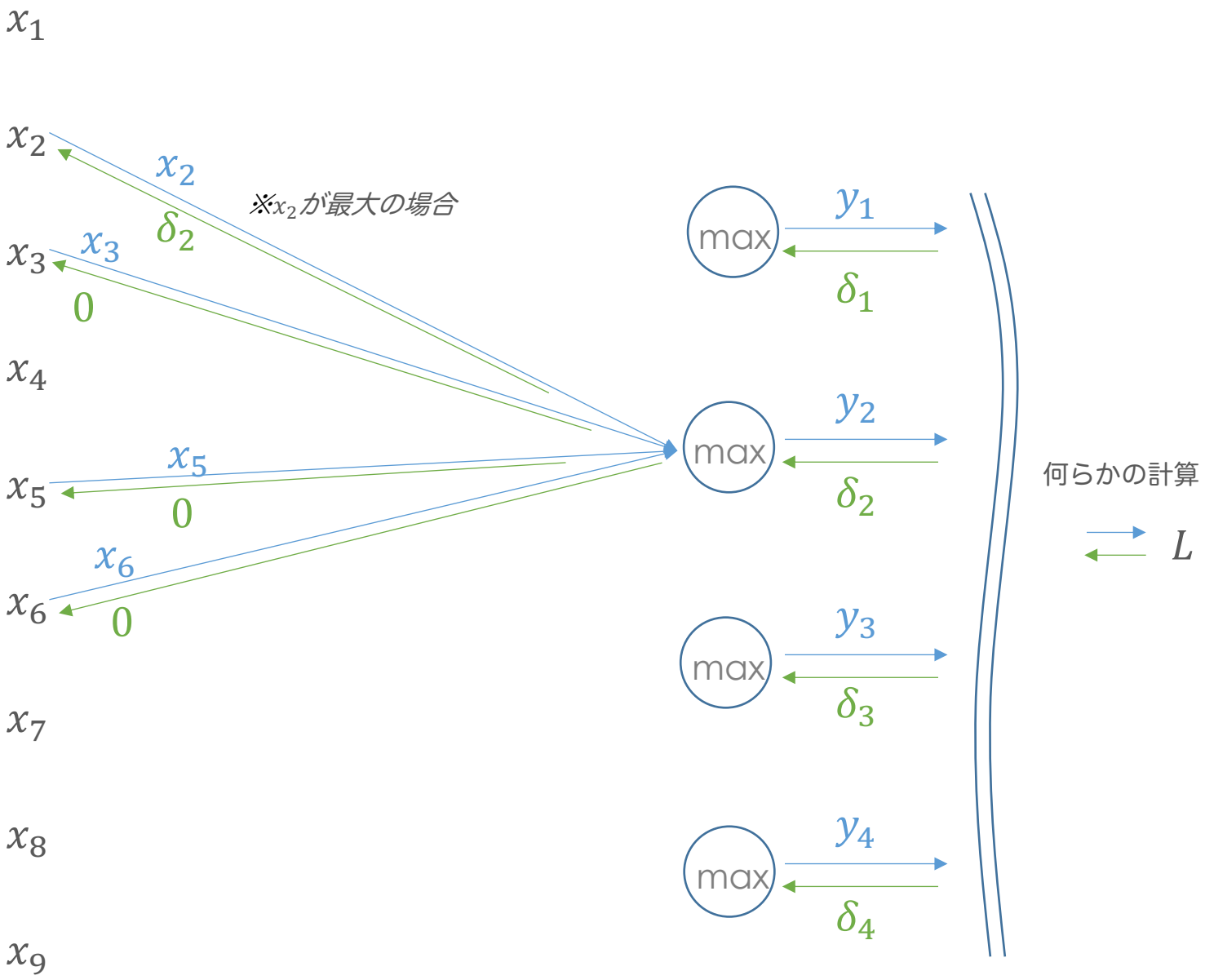


2019 SkillUP AI All rights reserved

マックスプーリングレイヤ

- マックスプーリングレイヤの計算グラフ例を右記に示す。
- バッチ版の考え方は、アフィンレイヤとドロップアウトレイヤが参考になる。

<条件>
入力データの高さ=3
入力データの幅=3
プーリング窓の高さ=2
プーリング窓の幅=2
ストライド=1
パッド=0

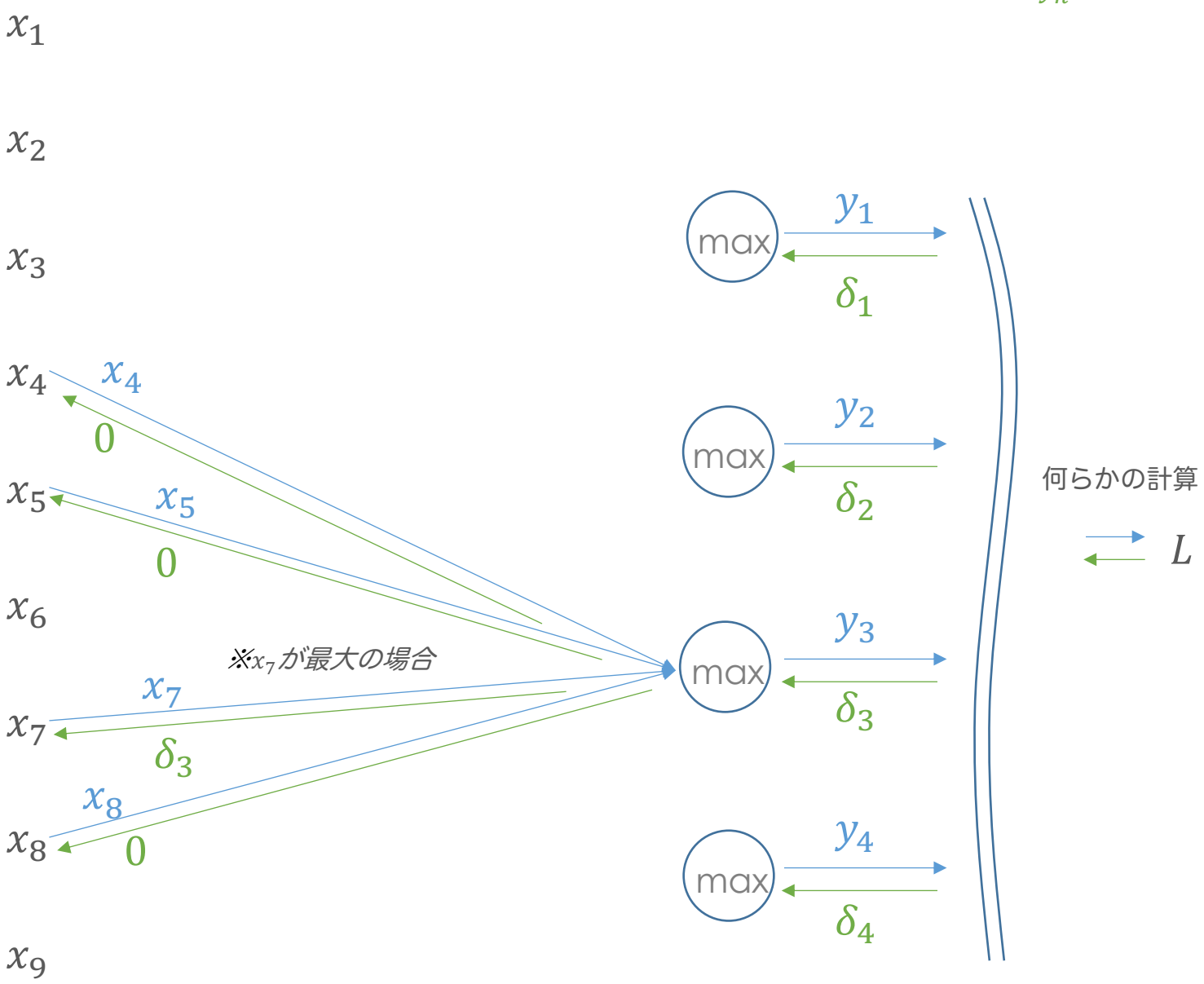


2019 SkillUP AI All rights reserved

マックスプーリングレイヤ

- マックスプーリングレイヤの計算グラフ例を右記に示す。
- バッチ版の考え方は、アフィンレイヤとドロップアウトレイヤが参考になる。

<条件>
入力データの高さ=3
入力データの幅=3
プーリング窓の高さ=2
プーリング窓の幅=2
ストライド=1
パッド=0

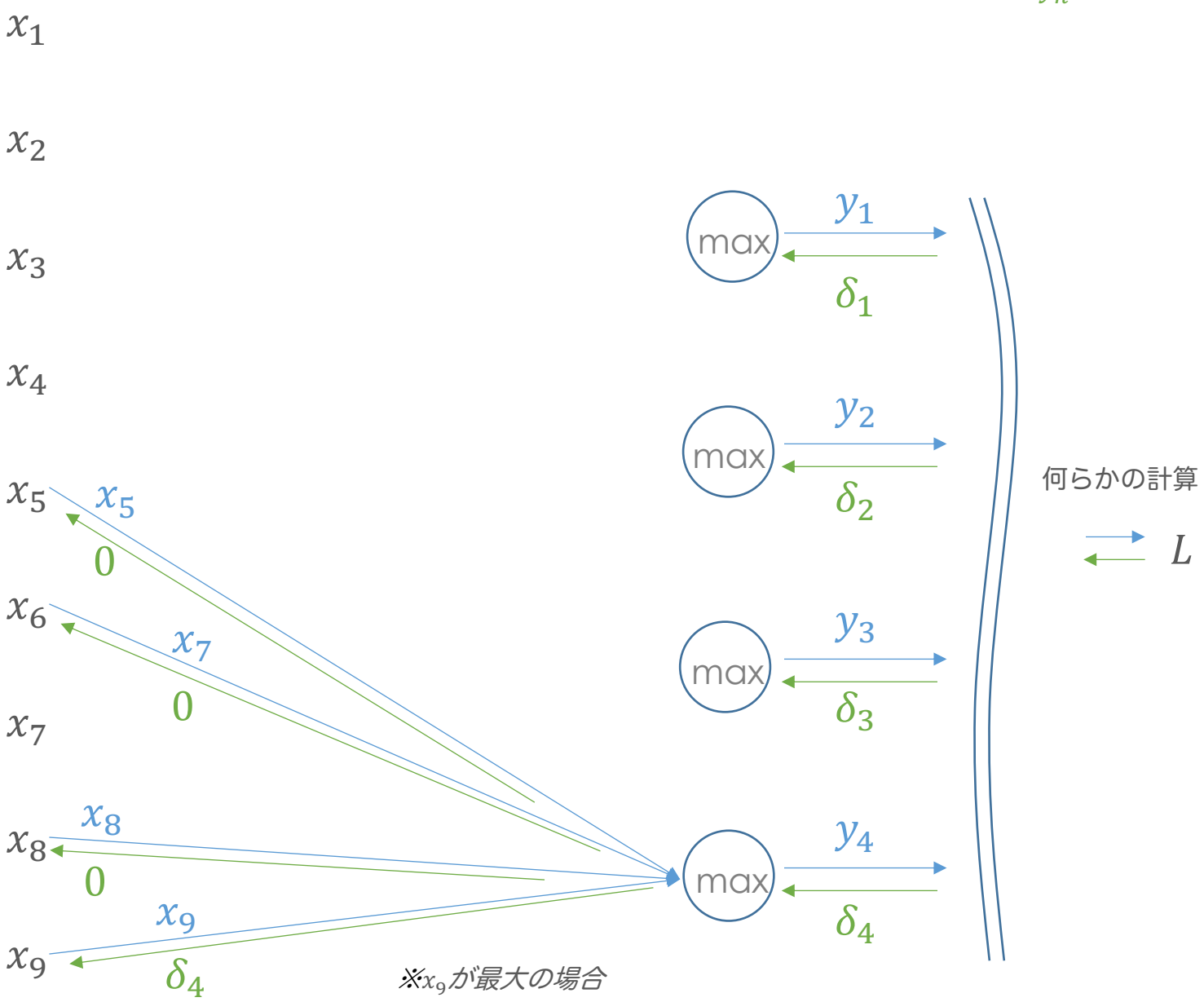


2019 SkillUP AI All rights reserved

マックスプーリングレイヤ

$\delta_n = \frac{\partial L}{\partial y_n}$とおく

- マックスプーリングレイヤの計算グラフ例を右記に示す。
- バッチ版の考え方は、アフィンレイヤとドロップアウトレイヤが参考になる。



<条件>
入力データの高さ=3
入力データの幅=3
プーリング窓の高さ=2
プーリング窓の幅=2
ストライド=1
パッド=0

2019 SkillUP AI All rights reserved

マックスプーリングレイヤ

- マックスプーリングレイヤの計算グラフ例を右記に示す。
- バッチ版の考え方は、アフィンレイヤとドロップアウトレイヤが参考になる。

<条件>
入力データの高さ=3
入力データの幅=3
プーリング窓の高さ=2
プーリング窓の幅=2
ストライド=1
パッド=0

$x_1 \xrightarrow{x_1} \xleftarrow{\partial L/\partial x_1 = 0}$

$x_2 \xrightarrow{x_2} \xleftarrow{\partial L/\partial x_2 = \delta_1 + \delta_2}$

$x_3 \xrightarrow{x_3} \xleftarrow{\partial L/\partial x_3 = 0}$

$x_4 \xrightarrow{x_4} \xleftarrow{\partial L/\partial x_4 = 0 + 0}$

$x_5 \xrightarrow{x_5} \xleftarrow{\partial L/\partial x_5 = 0 + 0 + 0 + 0}$

$x_6 \xrightarrow{x_6} \xleftarrow{\partial L/\partial x_6 = 0 + 0}$

$x_7 \xrightarrow{x_7} \xleftarrow{\partial L/\partial x_7 = \delta_3}$

$x_8 \xrightarrow{x_8} \xleftarrow{\partial L/\partial x_8 = 0 + 0}$

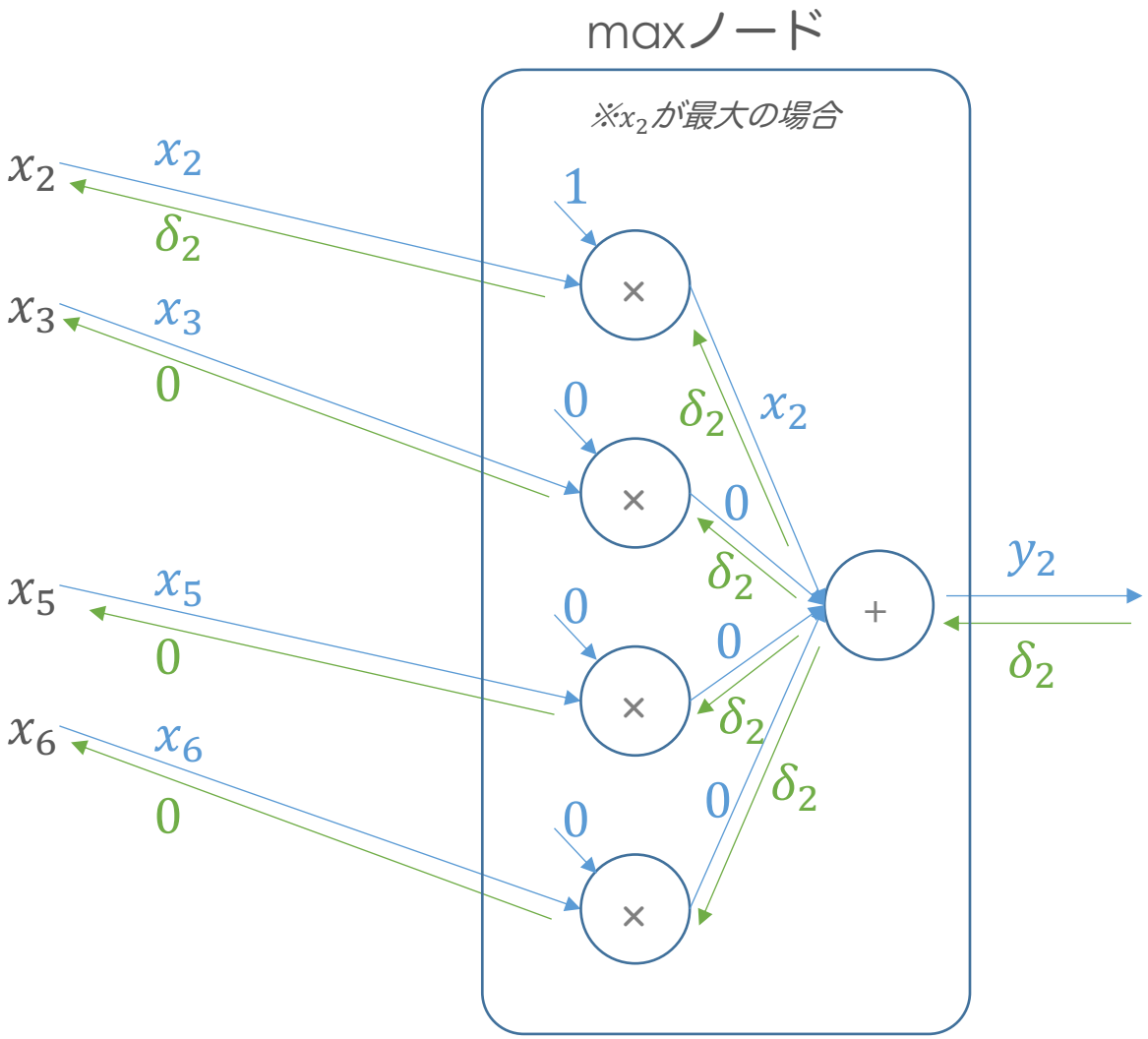
$x_9 \xrightarrow{x_9} \xleftarrow{\partial L/\partial x_9 = \delta_4}$

x_n毎に勾配を足し合わせる

2019 SkillUP AI All rights reserved

マックスプーリングレイヤ

- maxノードでの計算を詳細に書くと右記のようになる。



2019 SkillUP AI All rights reserved

Any Questions?

2019 SkillUP AI All rights reserved