

9. 深層学習

- 本章の説明手順

1. 深層学習の定義と学習の枠組み

2. 多階層学習における工夫

- 事前学習、活性化関数の工夫、最適化器の工夫

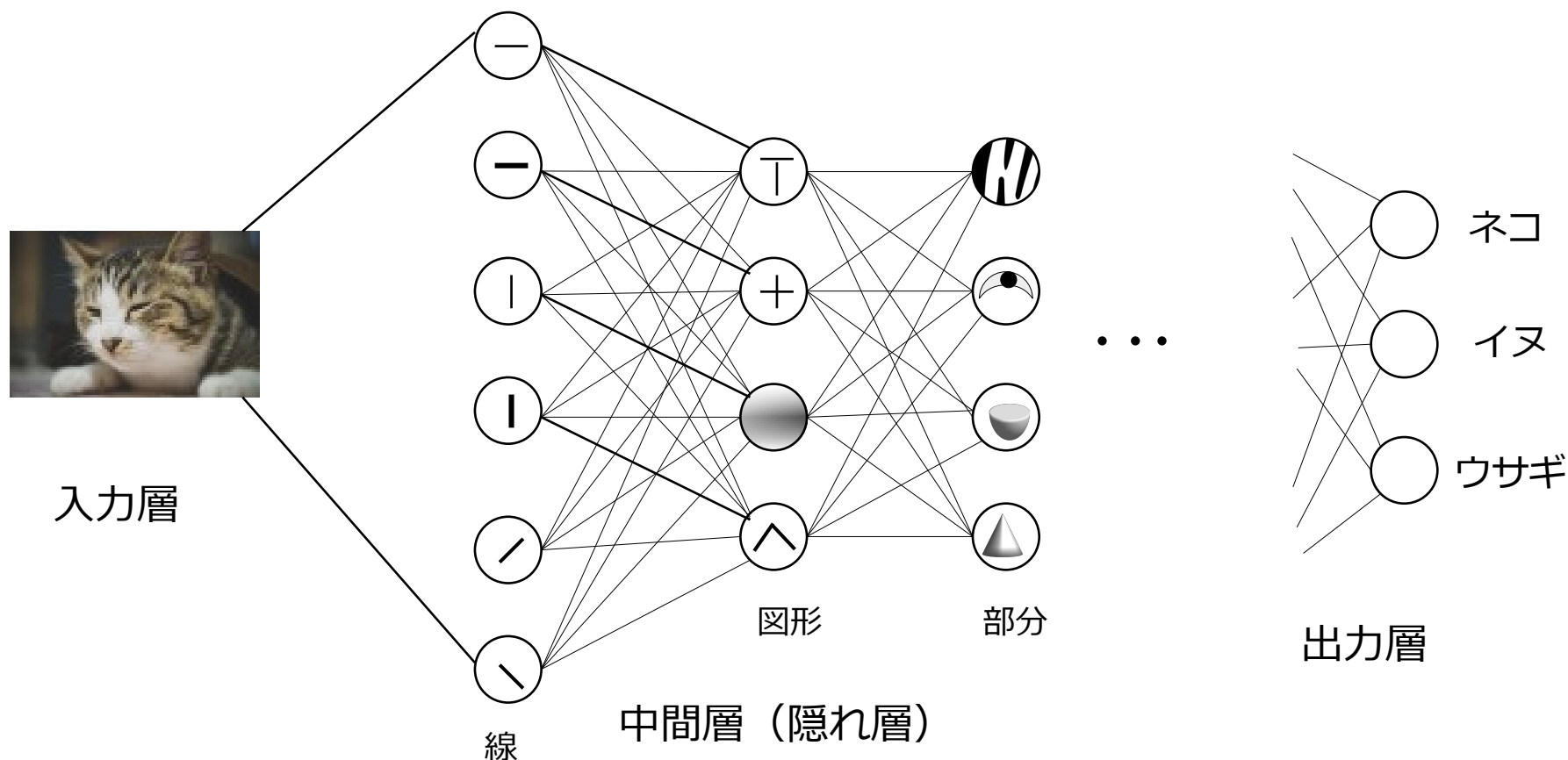
- 過学習の回避：ドロップアウト

3. 画像認識に適したネットワーク構造

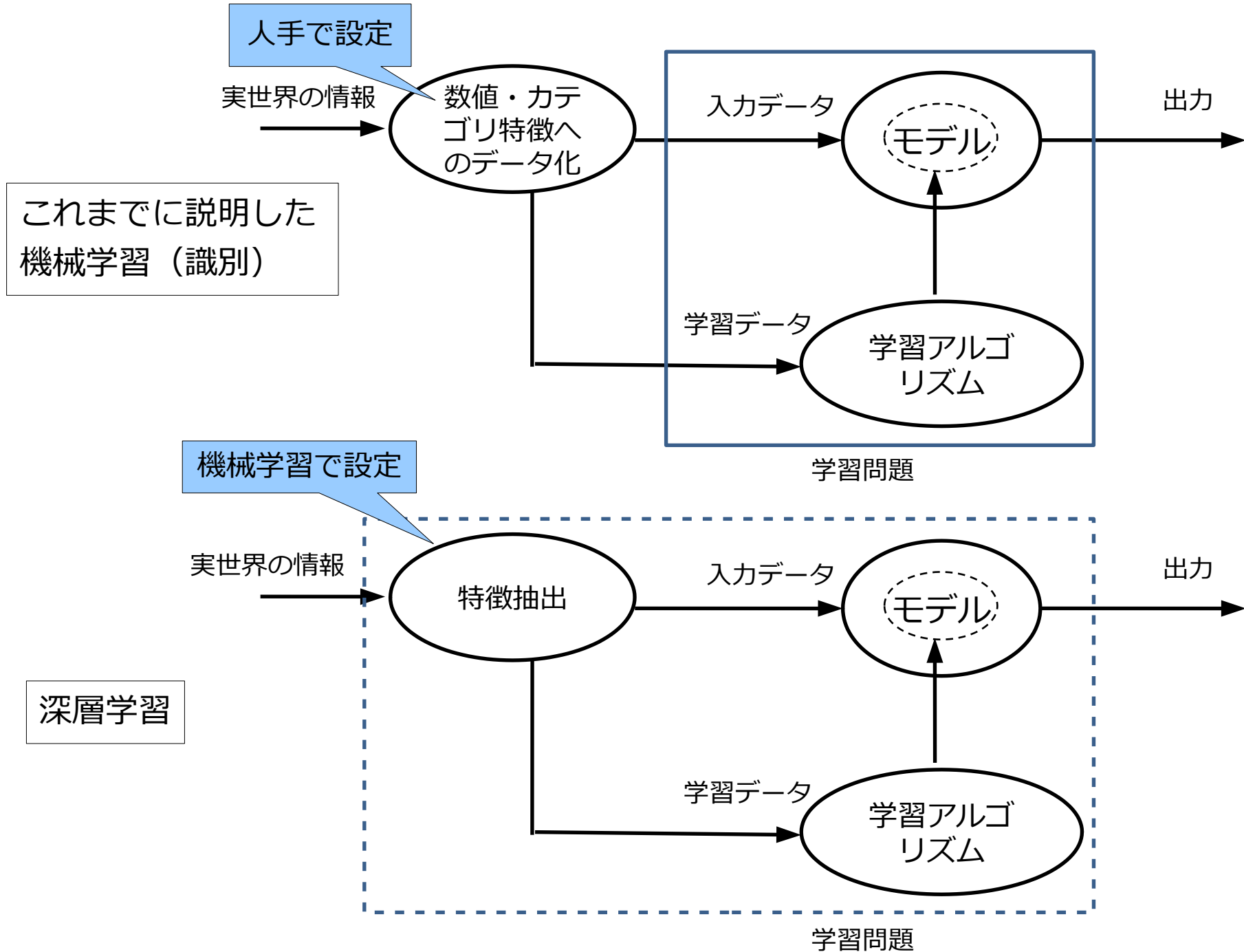
4. 自然言語処理に適したネットワーク構造

9.1 深層学習とは

- 深層学習 (deep learning) の定義
 - 多階層ニューラルネットワークの学習
 - 表現学習：処理を多層化することで特徴抽出も学習

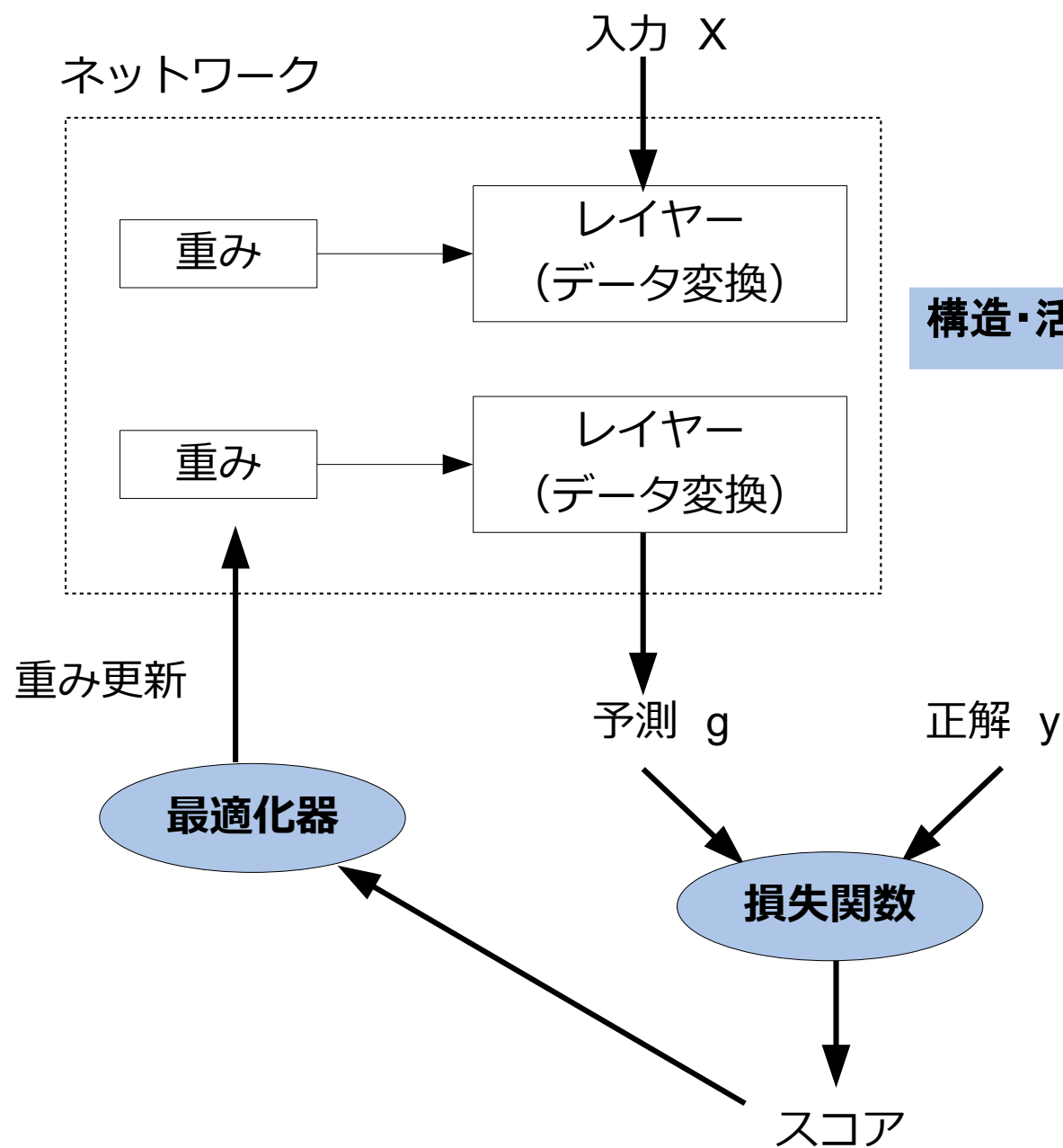


9.1 深層学習とは



9.2 深層学習の枠組み

復習



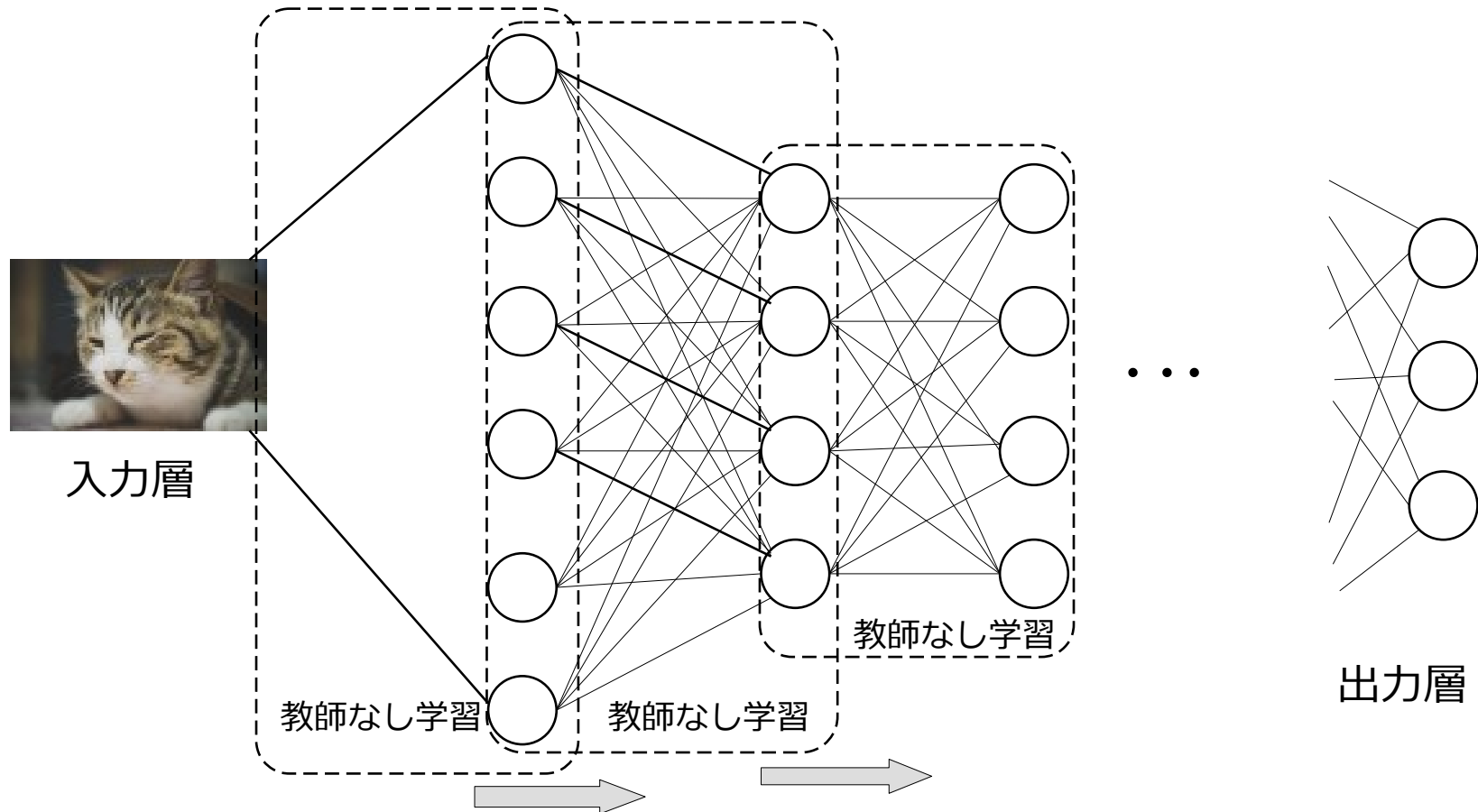
9.3 多階層ニューラルネットワーク

9.3.1 多階層ニューラルネットワークの学習

- 多階層学習の問題点
 - 勾配消失問題 第 8 章で説明済み
 - 多数のパラメータ（＝過学習になりやすい）
- 解決法
 - 事前学習（現在はあまり使われていない）
 - ただし、オートエンコーダの概念は有用
 - 活性化関数、最適化器の工夫 第 8 章で説明済み
 - 活性化関数：ReLU, tanh, LeakyReLU, ELU
 - 最適化器：Adam, AdaGrad, RMSProp
 - ドロップアウト
 - 正則化と同様の機能を持つ、学習上の工夫

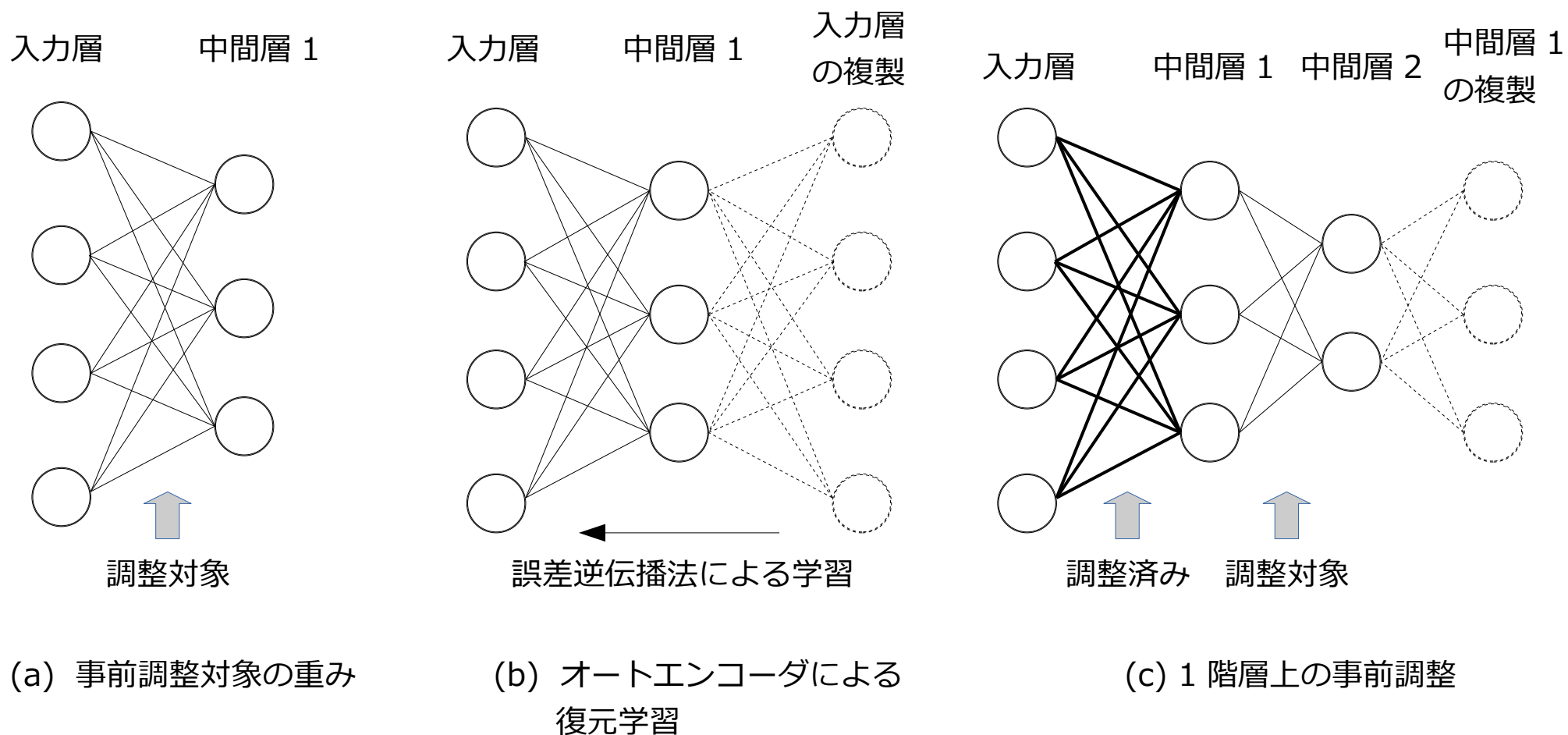
9.3.1 多階層ニューラルネットワークの学習

- 事前学習法のアイデア
 - 重みの初期値を、入力層から順に教師なしで学習 (pre-training)
 - その後、ネットワーク全体を教師あり学習 (fine-tuning)



9.3.2 オートエンコーダ

- アイデア：自己写像の学習で情報圧縮を実現
 - より低次元の本質的な情報が得られていることになる



9.3.2 オートエンコーダ

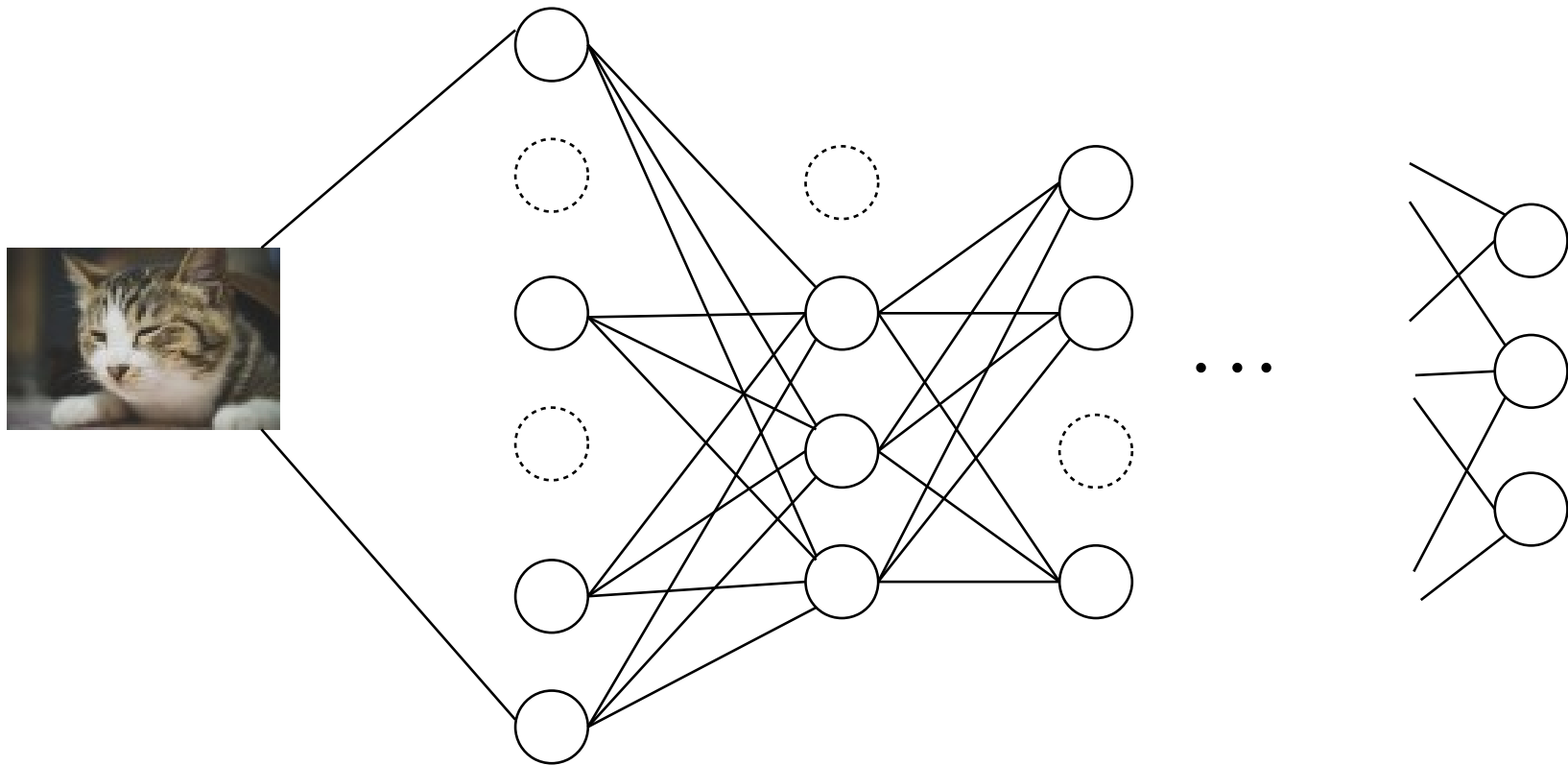
- 活性化関数や最適化器の工夫によって多階層学習が可能になったので、現在は事前学習自体はあまり用いられていない
- しかし、正解なしデータを使って教師あり学習問題を自動的に設定し、その中間表現を得ること自体は、現在でも有用なアイディア

例 1: Word2Vec による単語のベクトル化

例 2: Denoising Auto-encoder による画像変換

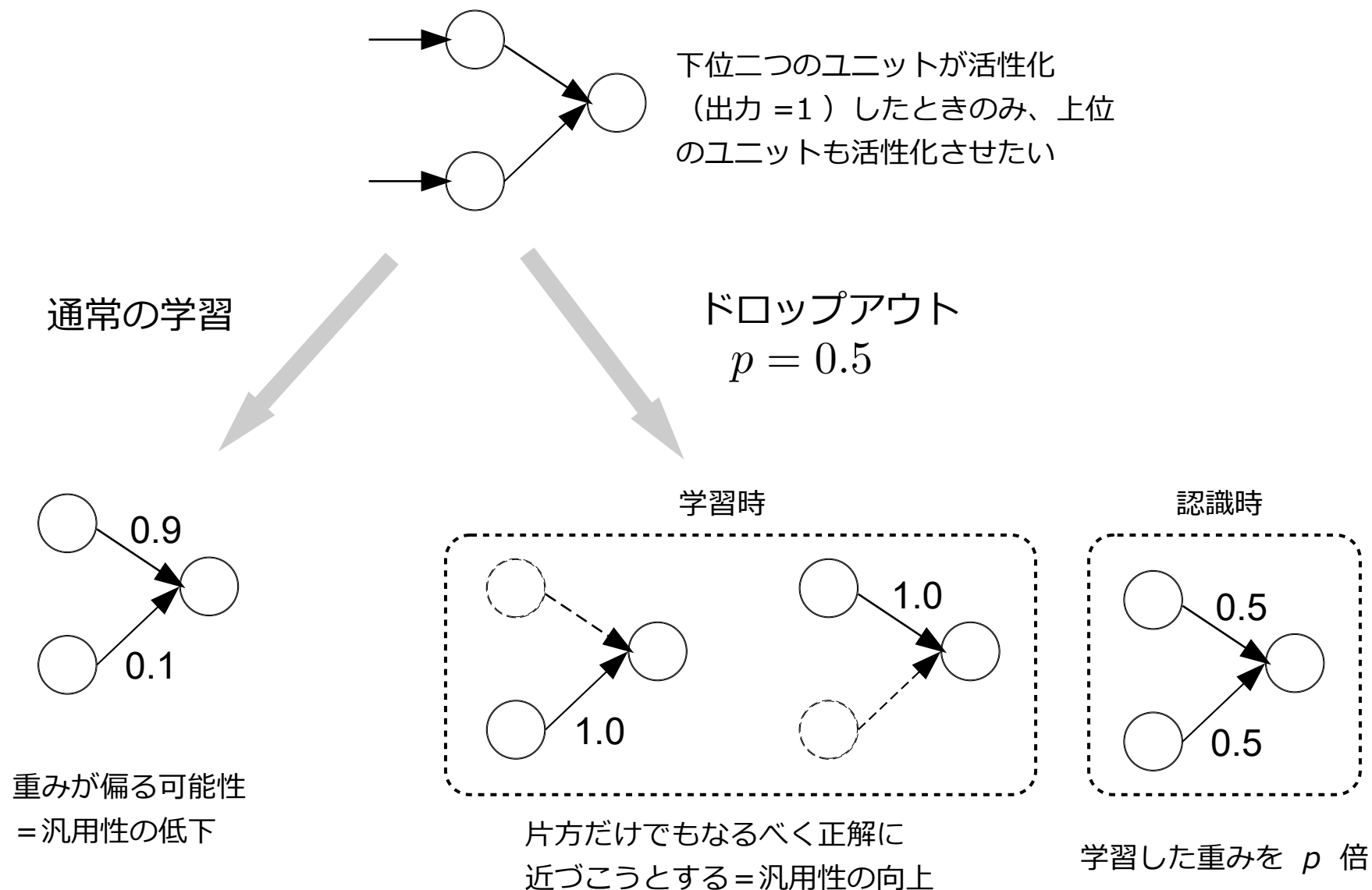
9.3.3 多階層学習における工夫

- 過学習の回避：ドロップアウト
 - ランダムに一定割合のユニットを消して学習
 - 認識時には学習後の重みに消去割合を掛ける



9.3.3 多階層学習における工夫

- ドロップアウトにおける正則化

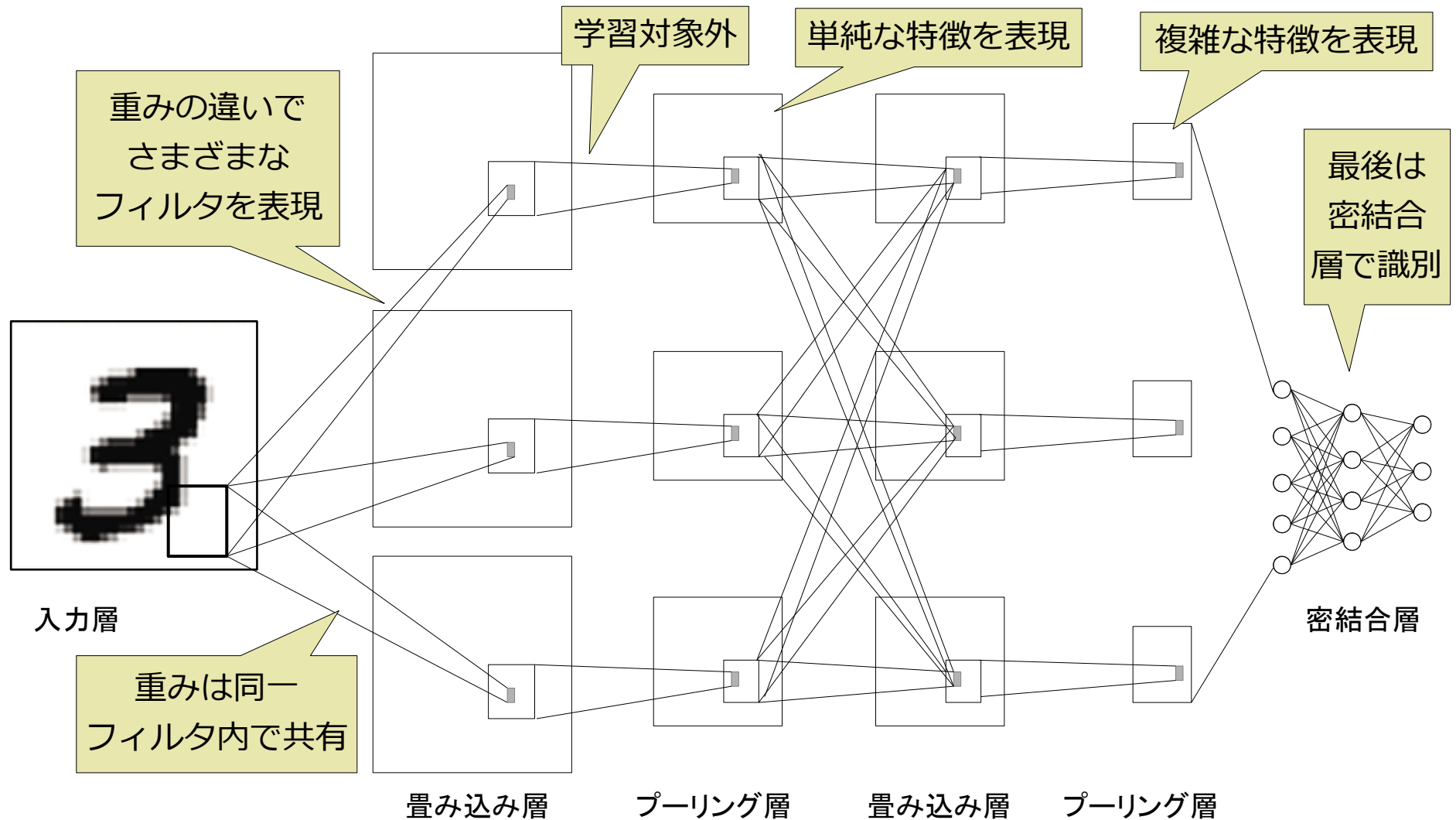


9.4 畳み込みネットワーク

- 畳み込みネットワーク (CNN) とは
 - 画像処理に適したネットワーク
 - 畳み込み層とプーリング層を交互に重ねて特徴抽出
 - 畳み込み層はフィルタを使って画像のパターンを見つける
 - プーリング層は位置の変動を吸収するダウンサンプリングの役割を果たす
 - 交互に重ねることで、複雑な特徴を表現可能
 - 最後は数段の密結合層（ReLU+Softmax）

9.4 畳み込みネットワーク

・畳み込みネットワークの構造

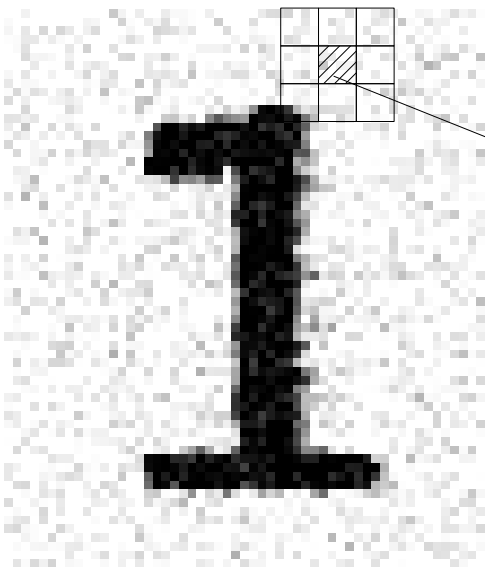


9.4 畳み込みネットワーク

- 畳み込み層の処理
 - 画像フィルタの適用と同じ

特定の画像入力に反応する
脳の視覚野領域の処理に対応

画素をずらして走査



この画素の値を

$$\sum_{p=0}^2 \sum_{q=0}^2 x_{i+p,j+q} w_{pq}$$

と置き換える

$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$
$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$
$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$

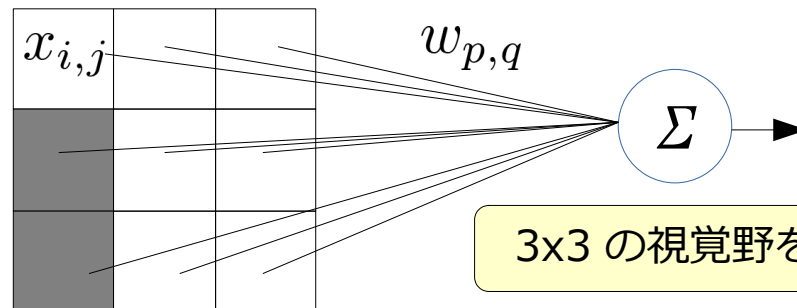
平均値フィルタ

ノイズを軽減する

-1	0	1
-1	0	1
-1	0	1

(縦) エッジフィルタ

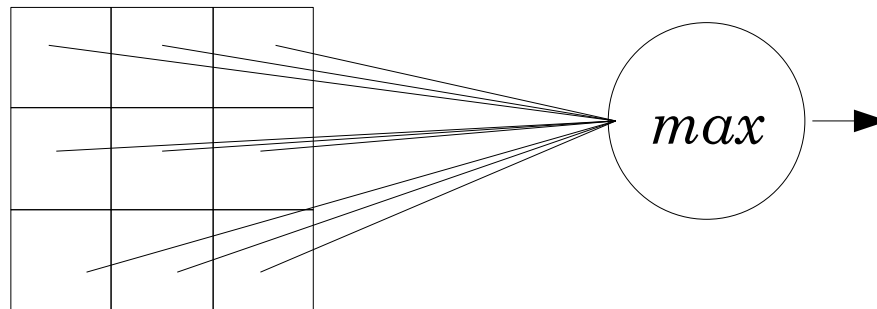
縦線を検出する



3x3 の視覚野を持つニューロン

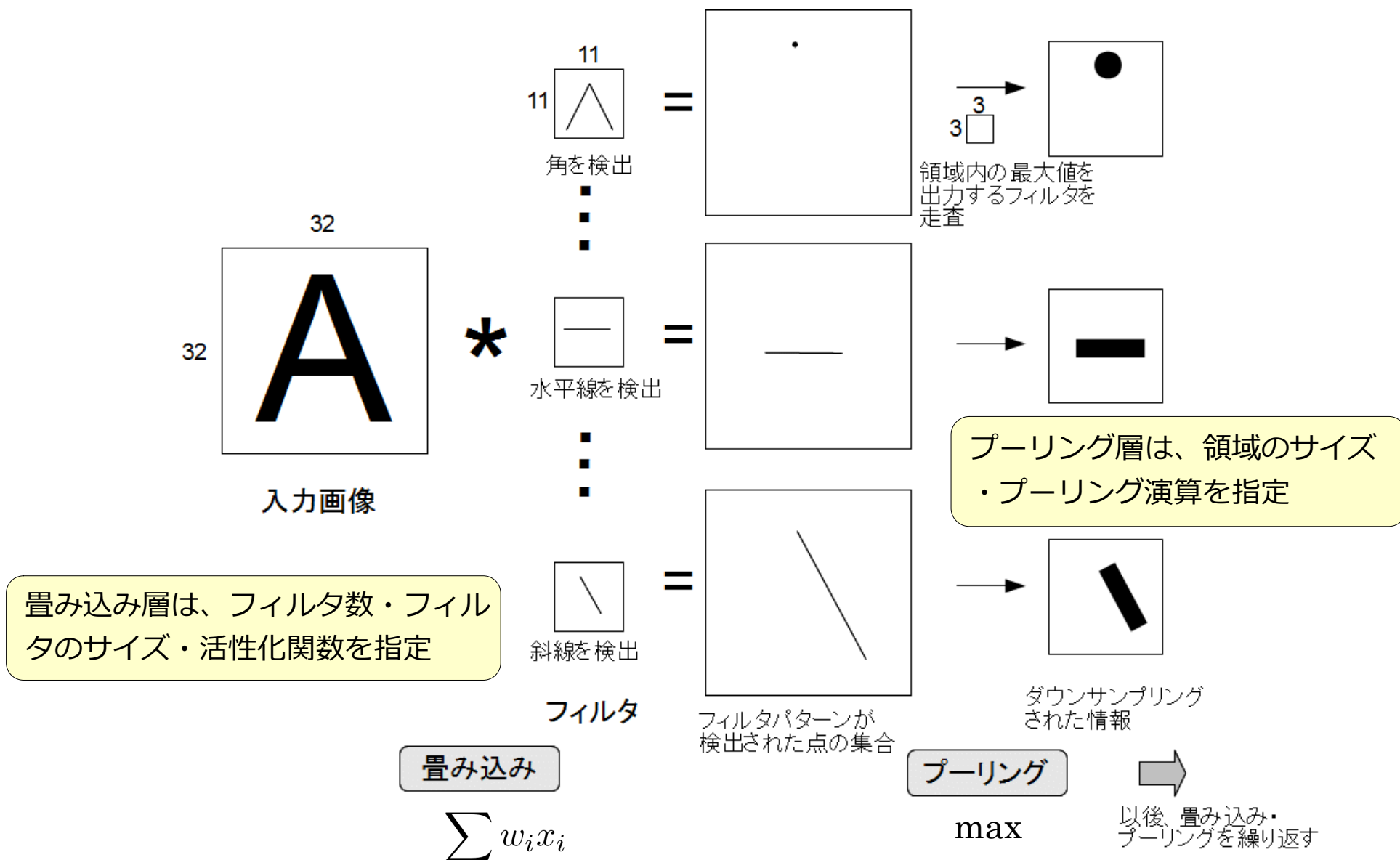
9.4 畳み込みネットワーク

- プーリング層の処理
 - 一定範囲の最大値あるいは平均値を計算
 - 畳み込み層によって発見した特徴の位置をぼかす役割



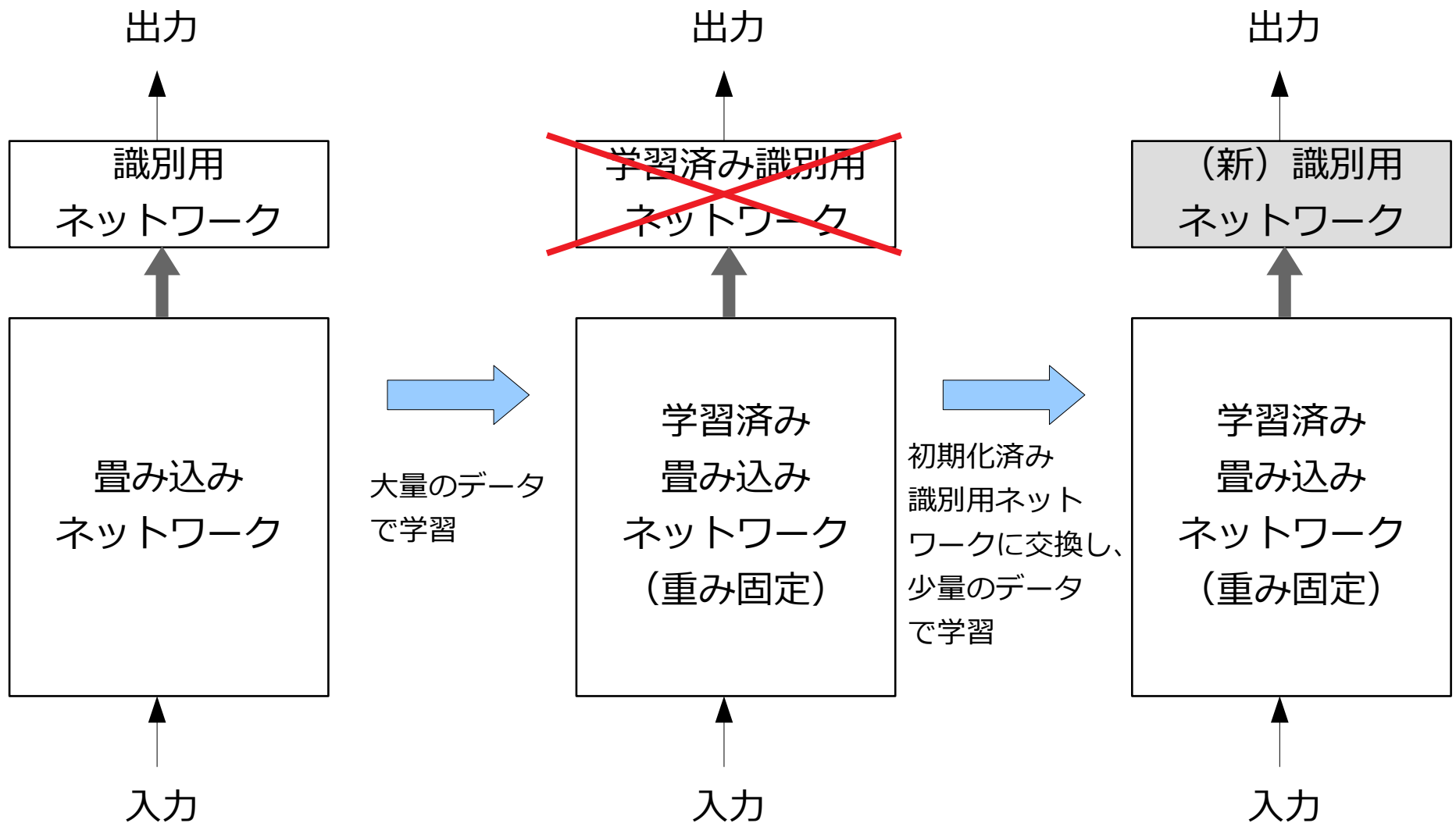
9.4 畳み込みネットワーク

・畳み込みネットワークの演算



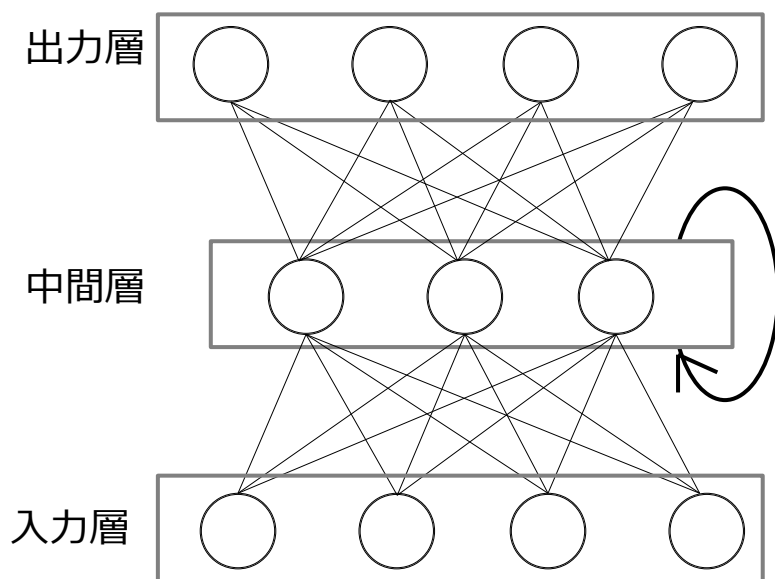
転移学習

- 大量のデータで学習済みの CNN があるとき
- 対象タスクのデータが少量でも転移学習が可能

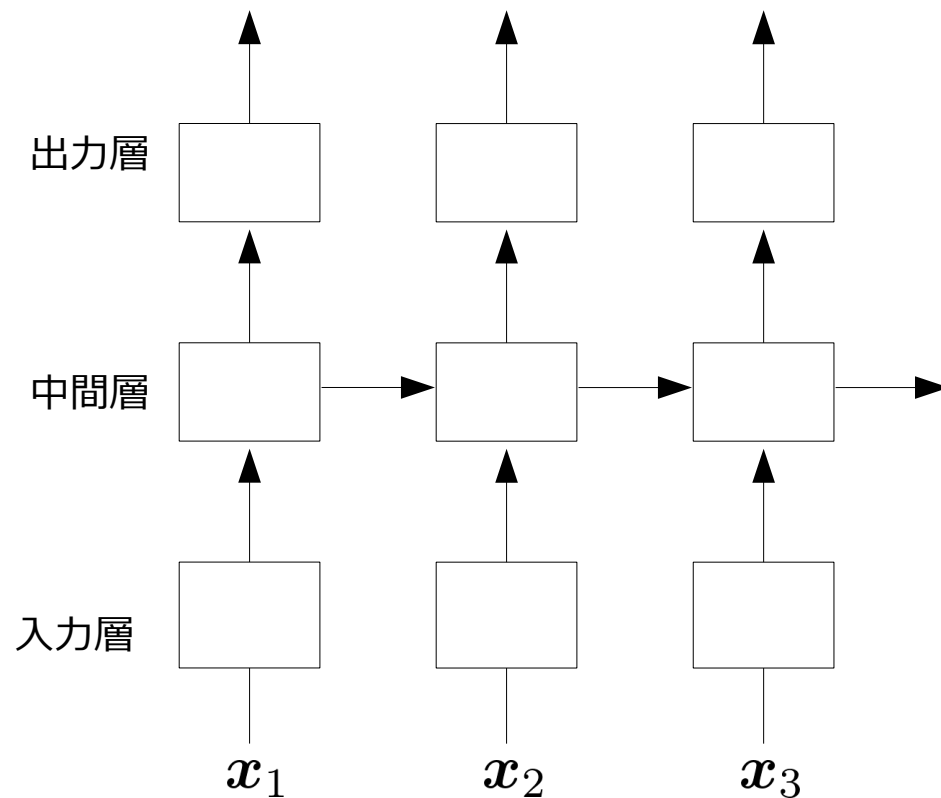


9.5 リカレントニューラルネットワーク

- リカレントニューラルネットワークとは
 - 時系列信号の認識や自然言語処理に適する
 - 一つ前の中間層の出力を、次の入力と結合



(a) リカレントニューラルネットワーク



(b) 帰還路を時間方向に展開

9.5 リカレントニューラルネットワーク

- リカレントネットワークの学習

- 通常 of 誤差逆伝播法の更新式

$$w'_{ji} \leftarrow w_{ji} + \eta \delta_j x_{ji}$$

に対して、時間を遡った更新が必要

- 時刻 t において、 k 個過去に遡った更新式

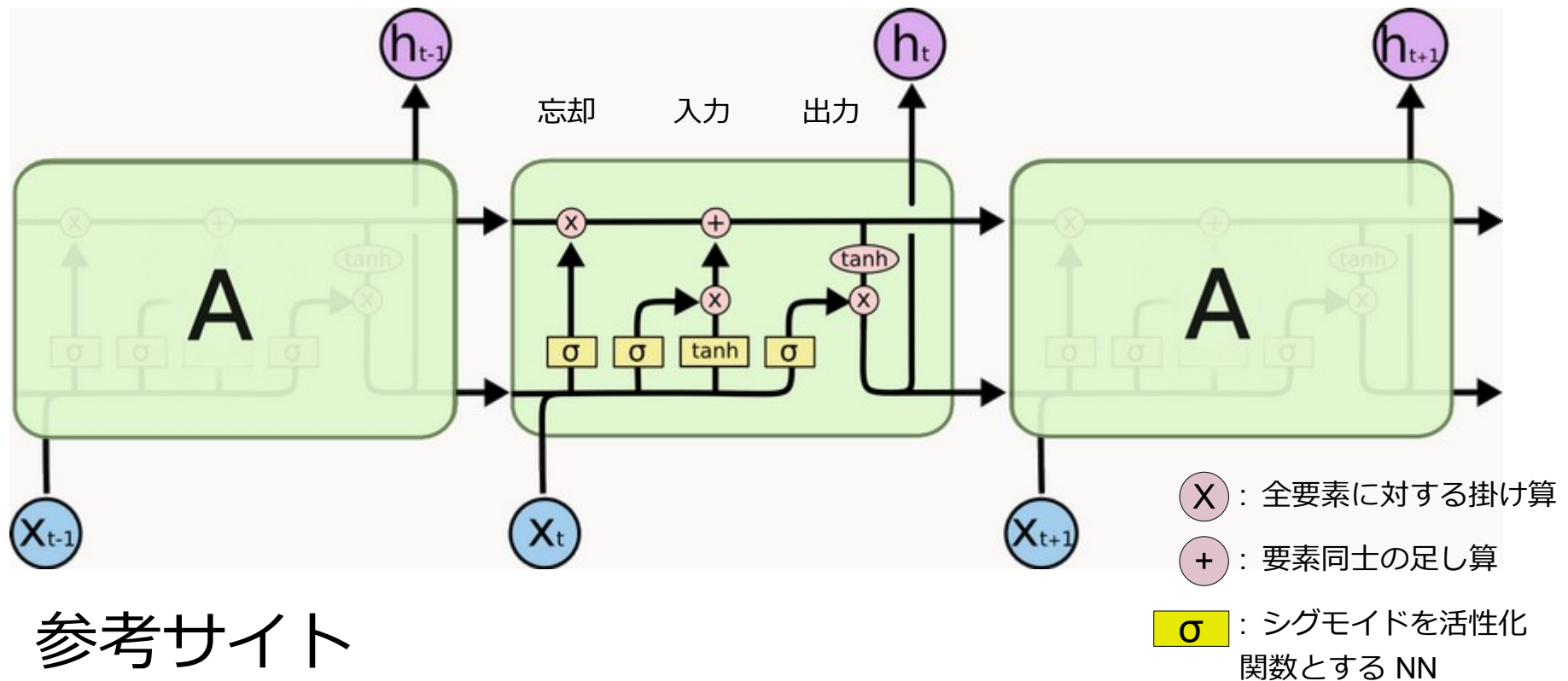
$$w_{ji}(t) \leftarrow w_{ji}(t-1) + \sum_{z=0}^k \eta \delta_j(t-z) x_{ji}(t-z-1)$$

- 問題点：識別問題において、系列前方の情報ほど出力に与える影響が小さくなる

- 例：英語の否定表現

9.5 リカレントニューラルネットワーク

- LSTM (long short-term memory)
 - いくつかのゲートからなる内部構造をもつユニット
 - ゲート：選択的に情報を通すメカニズム



- 参考サイト

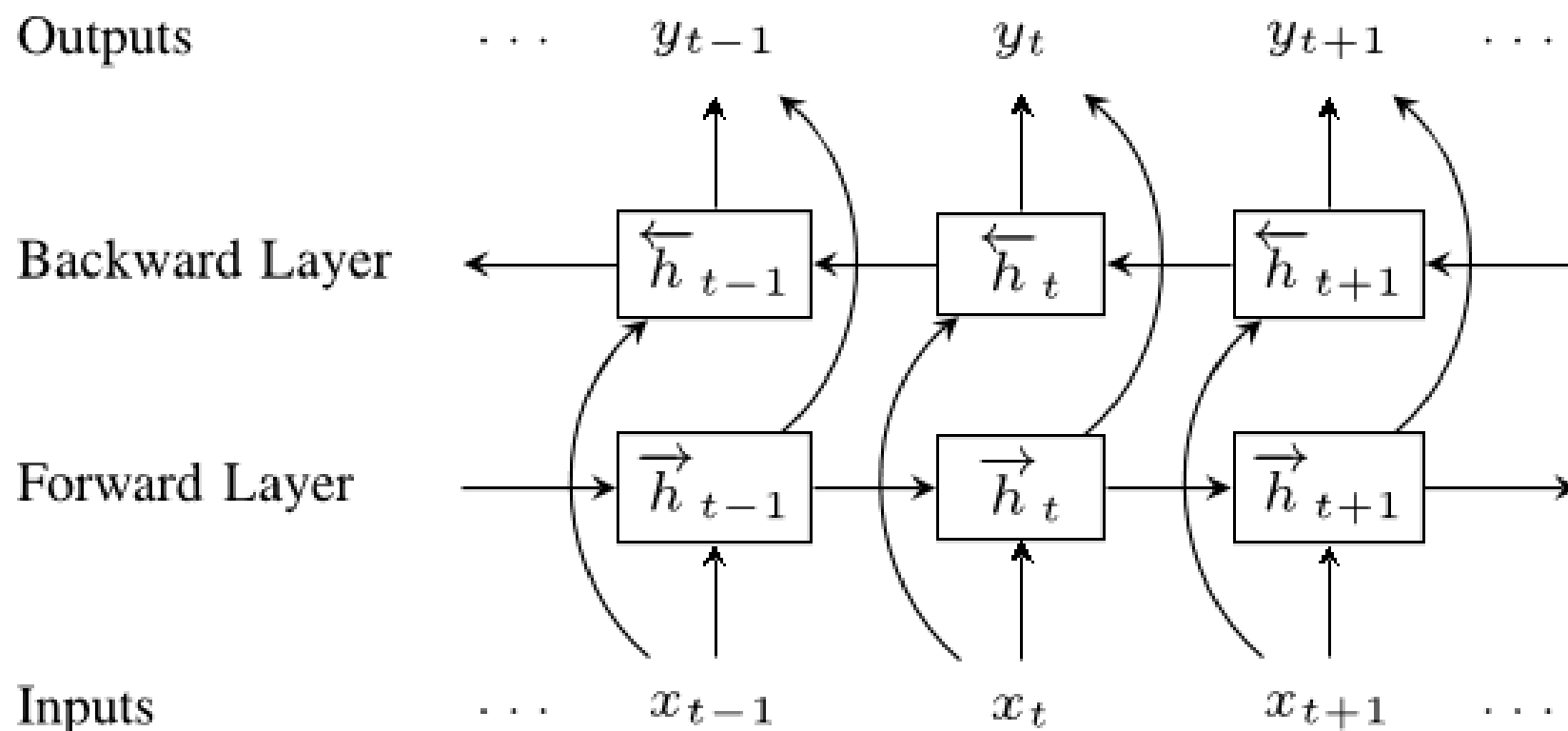
<http://colah.github.io/posts/2015-08-Understanding-LSTMs/>

9.5 リカレントニューラルネットワーク

- LSTM のゲート
 - 忘却ゲート：セルの内容を捨てるかどうか
 - 例) 言語モデルにおいて、新たな主語が現れた場合、古い主語の性別は捨てる
 - 入力ゲート：セルの内容のどの部分を更新するか
 - 例) 古い主語の性別を新たな主語の性別で置き換える
 - 出力ゲート：セルの内容のどの部分を出力するか
 - 例) 主語に続く動詞の形を決めるために、主語の単複を出力

9.5 リカレントニューラルネットワーク

- 双方向 RNN
 - 過去だけでなく、未来の情報も用いて出力を計算

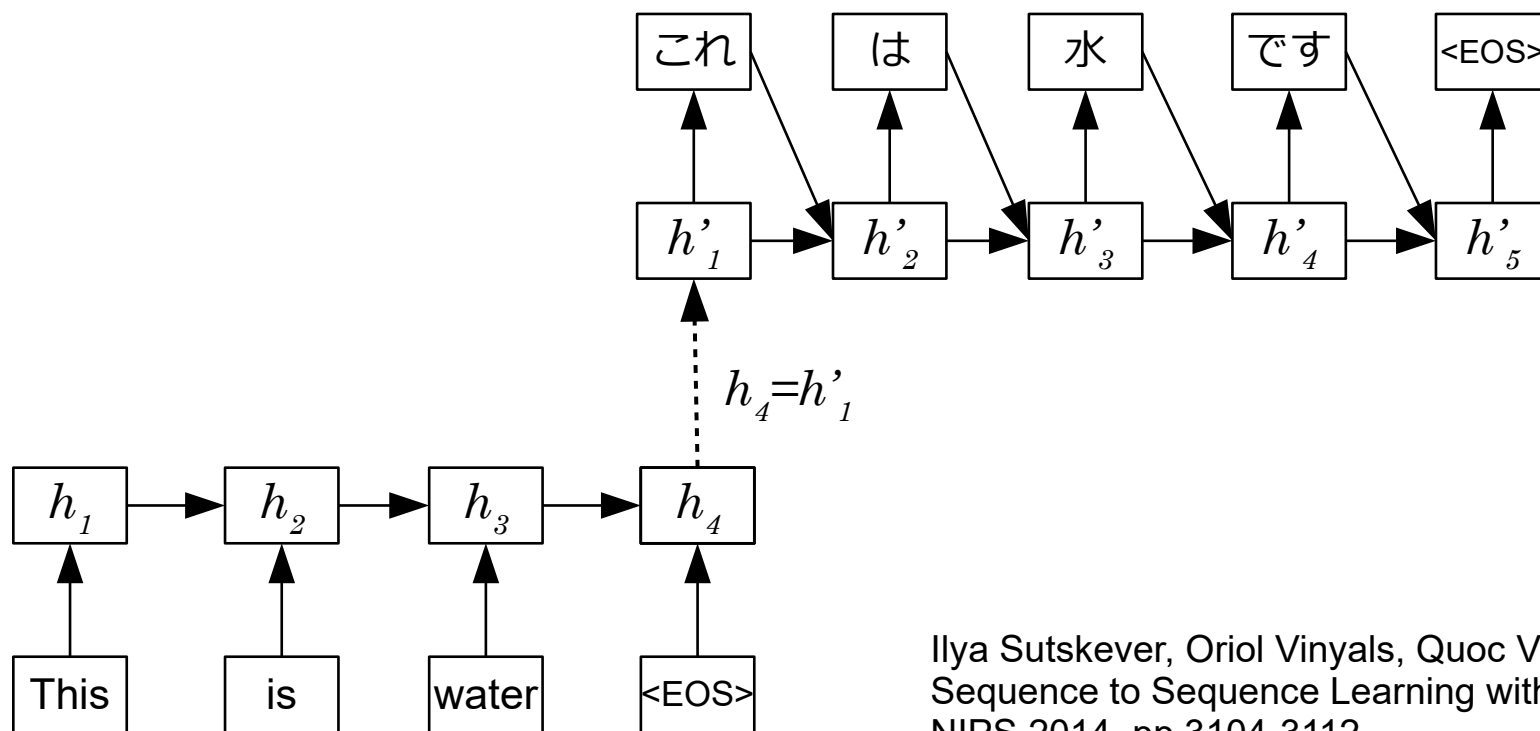


He, L., Qian, Y., Soong, F.K., Wang, P., & Zhao, H. (2015). A Unified Tagging Solution: Bidirectional LSTM Recurrent Neural Network with Word Embedding. CoRR, abs/1511.00215.

9.5 リカレントニューラルネットワーク

- Encoder-Decoder

- 入力の内容をひとつの表現にまとめて、そこから出力を生成

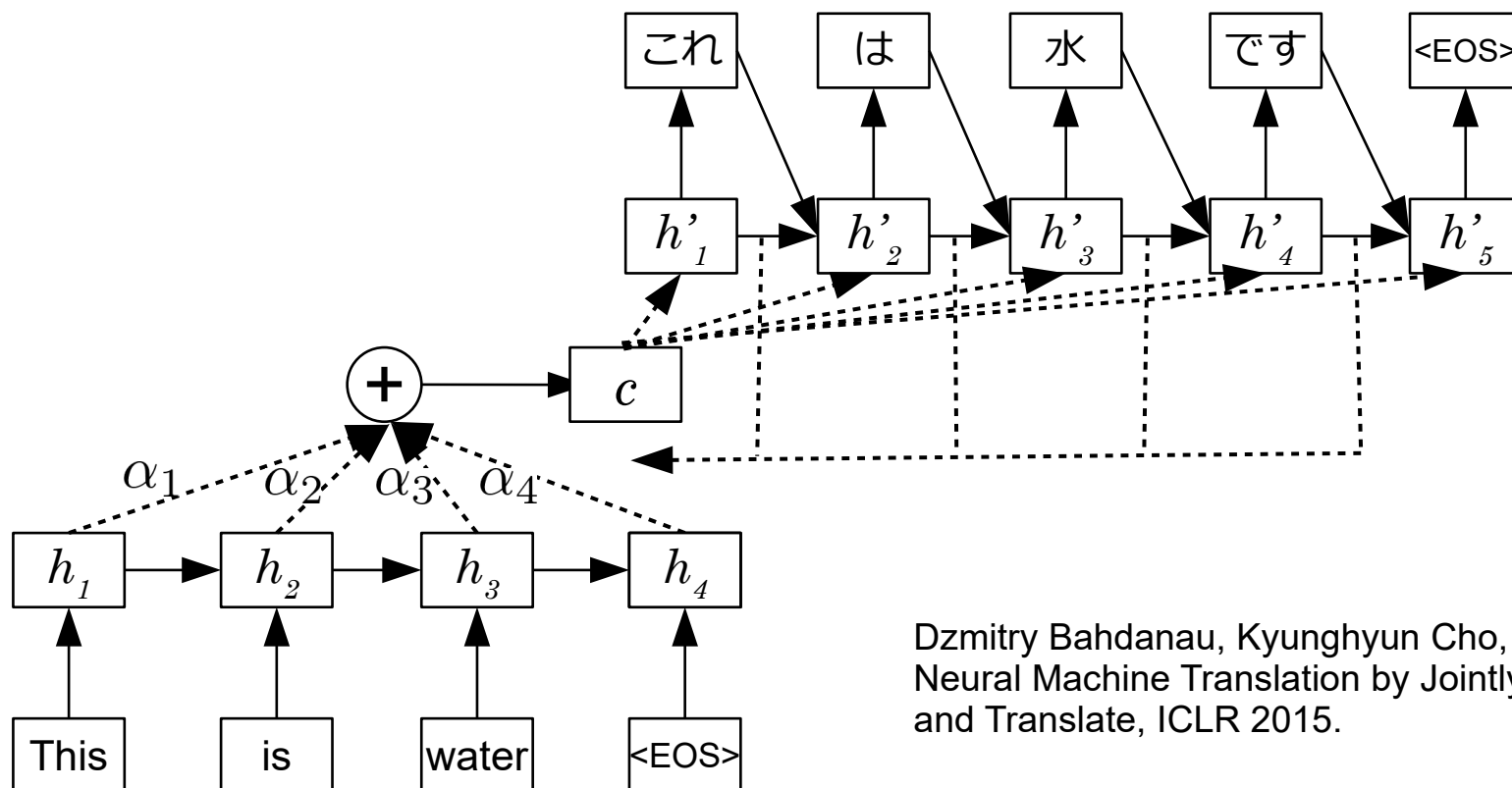


Ilya Sutskever, Oriol Vinyals, Quoc V. Le :
Sequence to Sequence Learning with Neural Networks,
NIPS 2014, pp.3104-3112.

9.5 リカレントニューラルネットワーク

- Encoder-Decoder+Attention

- アテンション： Encoder 内のすべての中間層出力の重み付き和（入力のどの部分を見るかという情報）

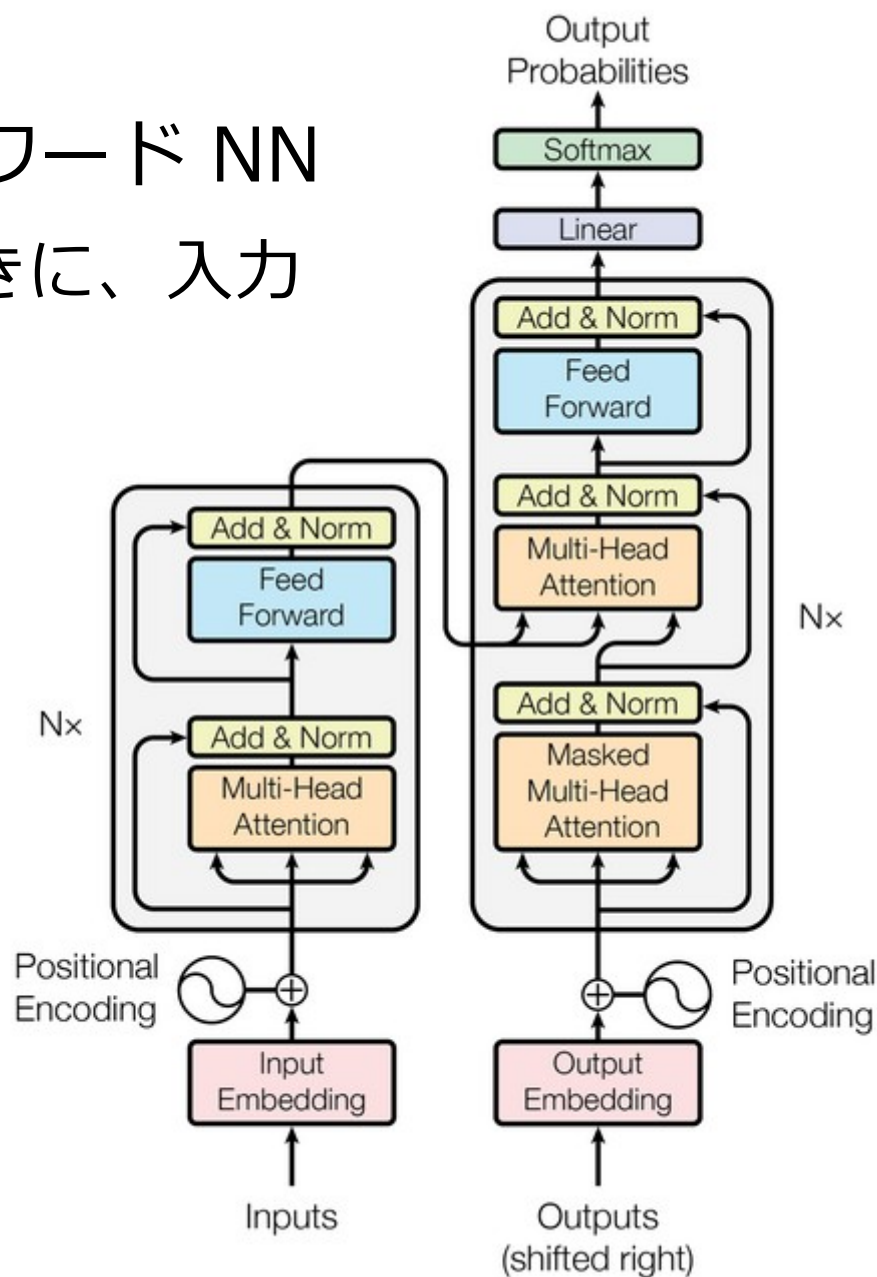
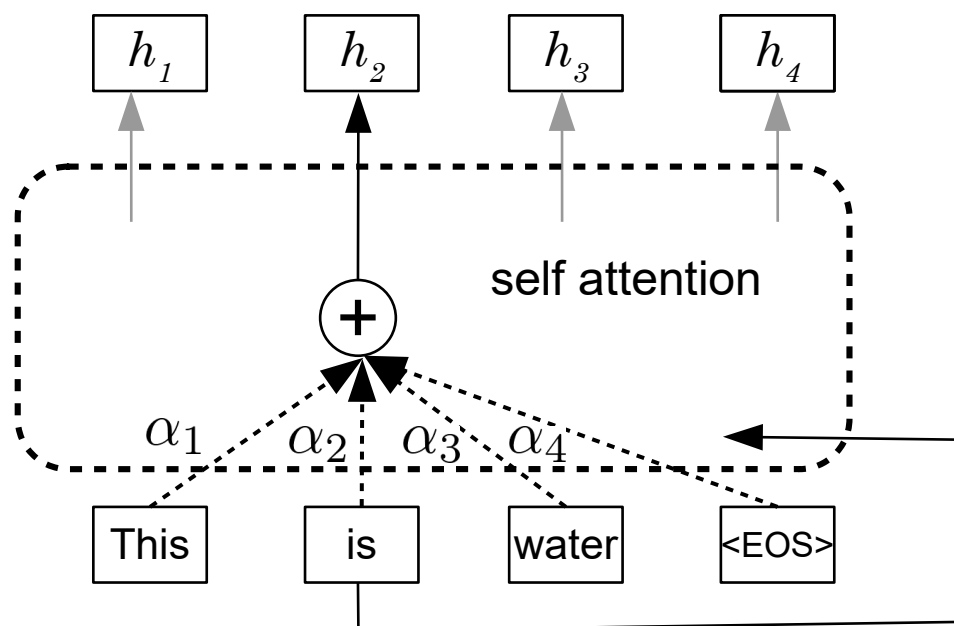


Dzmitry Bahdanau, Kyunghyun Cho, Yoshua Bengio :
Neural Machine Translation by Jointly Learning to Align
and Translate, ICLR 2015.

9.5 リカレントニューラルネットワーク

Transformer

- Attention+ フィードフォワード NN
 - 自分の中間表現を作るときに、入力のどの部分を見るか

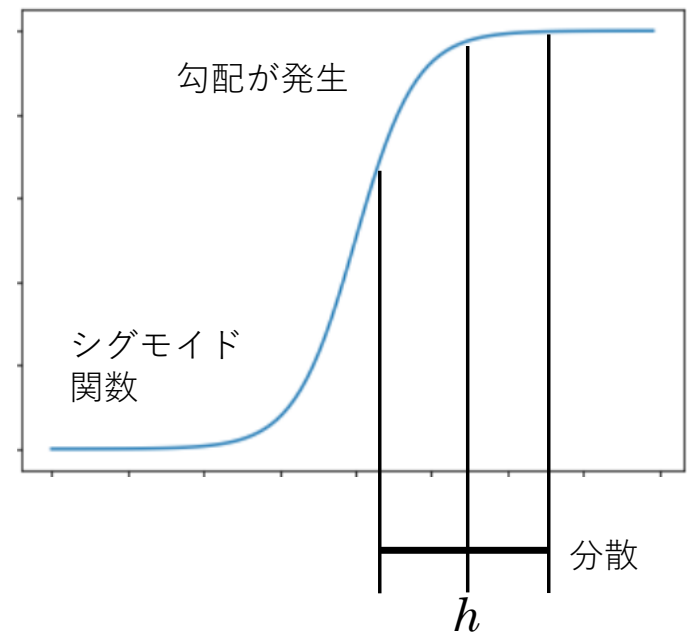


補足

9.3.3 多階層学習における工夫

- ドロップアウトが過学習を回避する理由
 - ネットワークの自由度を下げることで、正則化と同様の効果が生じる
 - ユニットへの入力 h の分散が大きくなるので、学習時に勾配が生じやすくなる

同じネットワークを同じデータで何度も学習することを避けていることがポイント



9.4 畳み込みネットワーク

- バッチ標準化の必要性
 - 入力データが標準化されていることは前提
 - 多階層のネットワークで演算を行うと、それぞれの階層の出力が適切な範囲に収まっているとは限らない（たとえば正の大きな値ばかりかもしれない）
- バッチ単位の標準化演算
 - 平均値を引いて標準偏差で割る
 - 1層のネットワークで実現可能