

機械学習（10回目）

創域理工学部 情報計算科学科

桂田 浩一

1

10/19/2023

2

前回の復習

- 制限を加えたボルツマンマシン
 - 隠れユニットのあるボルツマンマシン
 - 制限ボルツマンマシン

10/19/2023

3

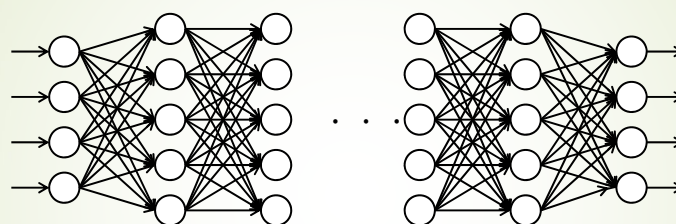
本日の内容

- 深層学習（1）
 - 層を多くした時の問題点
 - 問題の解決方法（3種類）

10/19/2023

4

層を多くしたときの問題点



誤差逆伝搬法

$$\delta_{p,j}^{(l)} = f'(\text{net}_{p,j}^{(l)}) \sum_{k=1}^{n^{(l+1)}} \delta_{p,k}^{(l+1)} w_{k,j}^{(l)}$$

$|f'(\text{net}_{p,j}^{(l)})| < 1$ だと $\delta_{p,j}^{(l)}$ がどんどん小さくなり（勾配消失問題）,
 $|f'(\text{net}_{p,j}^{(l)})| > 1$ だと $\delta_{p,j}^{(l)}$ がどんどん大きくなる（勾配爆発問題）

10/19/2023

5

勾配消失・爆発問題の解決法

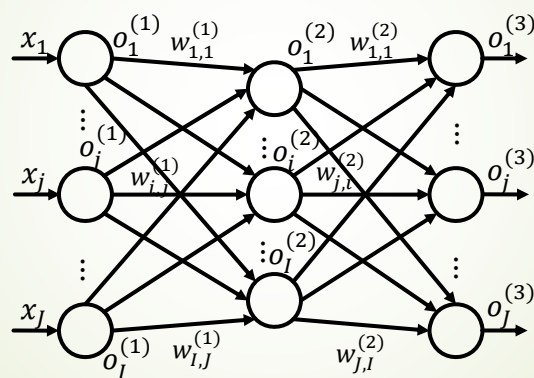
1. 誤差が伝わっていくのをやめる
 - Stacked AutoEncoder
 - RBMを用いたDNN (DBM, DBN)
2. 活性化関数を工夫して誤差が小さくなるのを防ぐ
 - ReLUの利用など
3. ネットワーク構造を工夫して誤差が小さくなるのを防ぐ
 - スキップコネクションなど

10/19/2023

6

AutoEncoder (復習)

- 入力 = 出力となる恒等変換を学習するニューラルネット

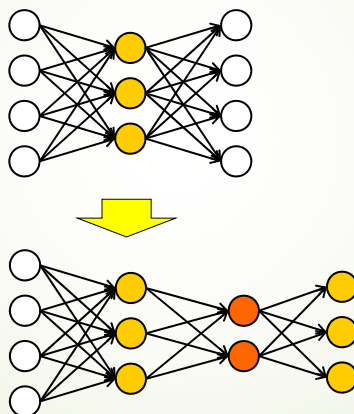


10/19/2023

7

Stacked AutoEncoder

- AutoEncoderの中間層を入出力として層を増やす

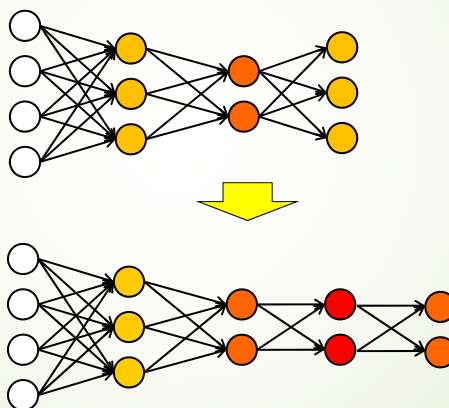


10/19/2023

8

Stacked AutoEncoder

- AutoEncoderの中間層を入出力として層を増やす



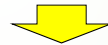
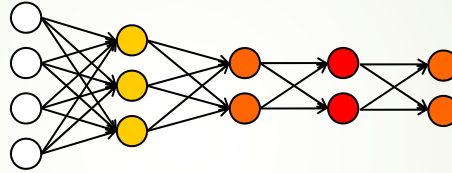
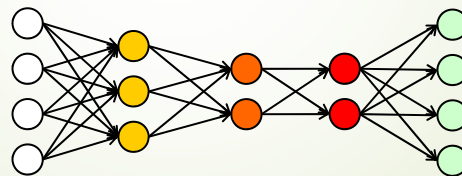
10/19/2023

9

事前学習とファインチューニング

- 層ごとに学習を行った後に全体の学習を行う方法

事前学習

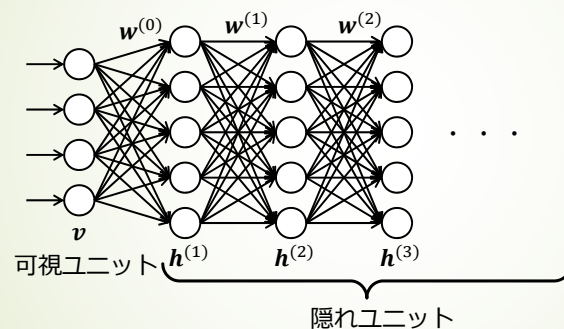
ファイン
チューニング

10/19/2023

10

Deep Boltzmann Machine

- 隠れユニットが層を形成しているボルツマンマシン



エネルギー関数

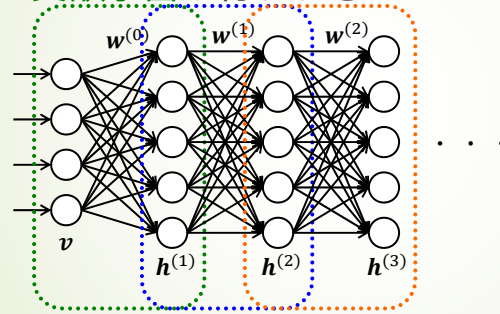
$$\Phi(\mathbf{v}, \mathbf{h}, \boldsymbol{\theta}) = -\sum_{(i,j)} w_{j,i}^{(0)} v_i h_j^{(1)} - \sum_{r=2}^R \sum_{(i,j)} w_{j,i}^{(r-1)} h_i^{(r-1)} h_j^{(r)}$$

10/19/2023

11

Deep Boltzmann Machine

- 現実的な学習時間にするため層ごとにRBMを用いた貪欲学習が行われる



RBMの学習 RBMの学習 RBMの学習

$h^{(1)}$ を可視層とみなす

最後に出力層を加えてファインチューニングを行う

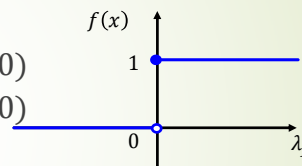
10/19/2023

12

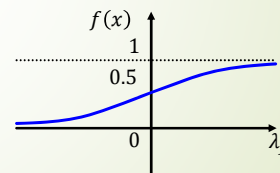
活性化関数を工夫する

- 活性化関数の種類 ($\lambda_j = \sum_{i=1}^l x_i w_{j,i} - \theta_j$)

- ステップ関数: $f(\lambda_j) = \begin{cases} 1 & (\lambda_j \geq 0) \\ 0 & (\lambda_j < 0) \end{cases}$
 $f'(\lambda_j) = 0$



- シグモイド関数: $f(\lambda_j) = \frac{1}{1+e^{-\lambda_j}}$
 $f'(\lambda_j) = \frac{e^{-\lambda_j}}{(1+e^{-\lambda_j})^2}$



10/19/2023

13

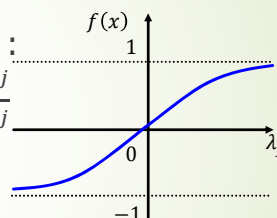
活性化関数を工夫する

■ 活性化関数の種類 ($\lambda_j = \sum_{i=1}^l x_i w_{j,i} - \theta_j$)

■ ハイパボリックタンジェント関数 :

$$f(\lambda_j) = \tanh(\lambda_j) = \frac{e^{\lambda_j} - e^{-\lambda_j}}{e^{\lambda_j} + e^{-\lambda_j}}$$

$$f'(\lambda_j) = \frac{4}{(e^{\lambda_j} + e^{-\lambda_j})^2}$$



■ ソフトマックス関数 :

$$f(\lambda_j) = \frac{e^{\lambda_j}}{\sum_{m=1}^M e^{\lambda_m}}$$

$$f'(\lambda_j) = \frac{e^{\lambda_j}}{\sum_{m=1}^M e^{\lambda_m}} - \left(\frac{e^{\lambda_j}}{\sum_{m=1}^M e^{\lambda_m}} \right)^2$$

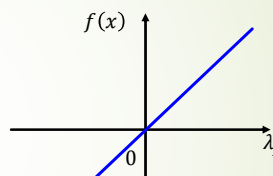
10/19/2023

14

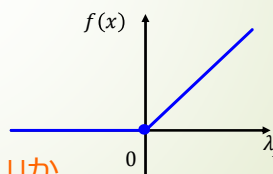
活性化関数を工夫する

■ 活性化関数の種類 ($\lambda_j = \sum_{i=1}^l x_i w_{j,i} - \theta_j$)

■ 恒等関数 : $f(\lambda_j) = \lambda_j$
 $f'(\lambda_j) = 1$



■ ReLU : $f(\lambda_j) = \max(0, \lambda_j)$
 $f'(\lambda_j) = 0$ または 1



最近のDeep Learningは大抵ReLUか、
その亜種です

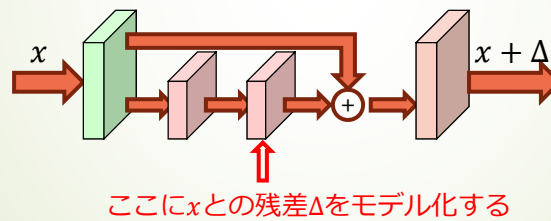
10/19/2023

15

ネットワーク構造を工夫する

- スキップコネクション
 - 低層から高層にバイパスを通す

ResNet (Residual Network)



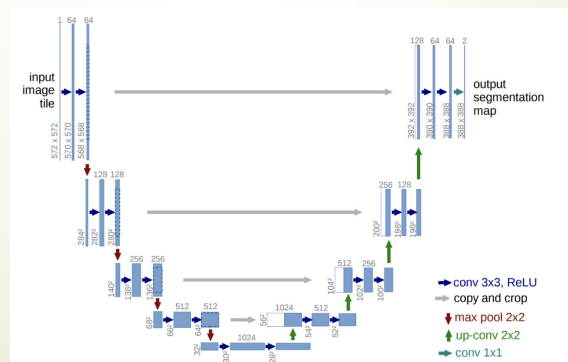
10/19/2023

16


ネットワーク構造を工夫する

- スキップコネクション
 - 低層から高層にバイパスを通す

U-Net (セマンティックセグメンテーション等で用いられる)



10/19/2023



出題予定の演習課題

- スキップ接続の誤差逆伝搬