

Deep Learning Learning 2019

IPLAB UBIQUITOUS(SAGA)チーム M2
我妻正太郎

本勉強会の目標

2

DeepLearningについて
ざっくり知る

本日の予定

3

- Deep Learningについての説明(本日)
 - ▣ Deep Learningって??
 - ▣ ニューラルネットワーク
 - ▣ 学習の進み方 など

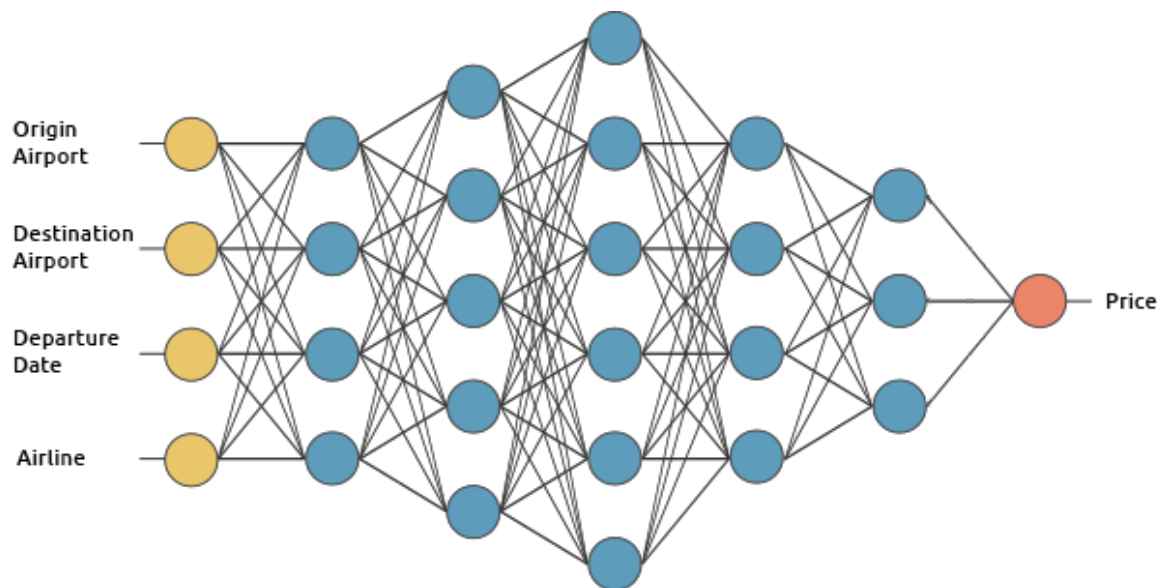
- Deep Learningのテスト(後日)
 - ▣ 小規模なニューラルネットワークで実験

Deep Learningってなんぞや

4

多くの層で構成された ニューラルネットワーク

を用いた機械学習手法



Deep Learningの特徴

5

□ 良いところ

- ▣ 他の機械学習手法に比べ、**精度が高い**
- ▣ **データの質に左右されにくい**
 - 生データのまま学習可能

□ 悪いところ

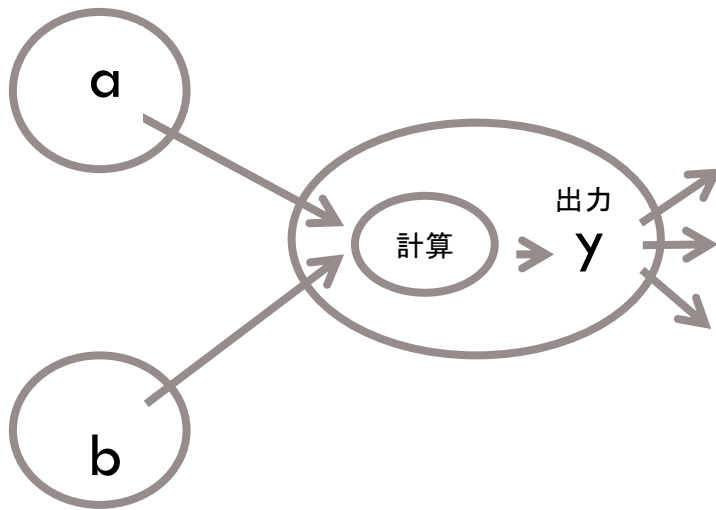
- ▣ **大量の学習用データが必要**
- ▣ **膨大な計算量, 時間がかかる**
- ▣ **学習の処理の可視化が難しい**

ニューラルネットワーク？ (1/2)

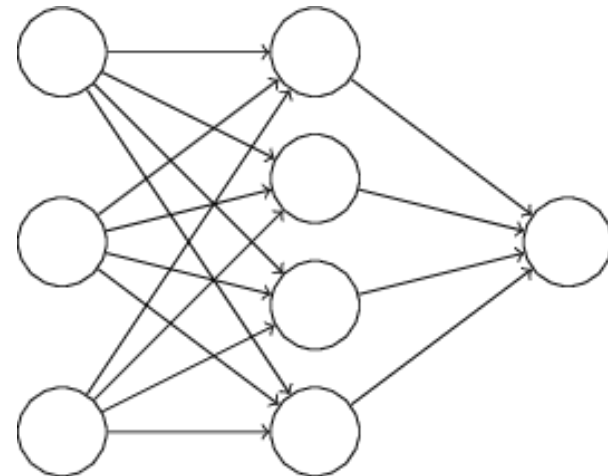
6

□ 多層パーセプトロン

- パーセプトロン...複数の信号を入力, 一つ出力



パーセプトロン



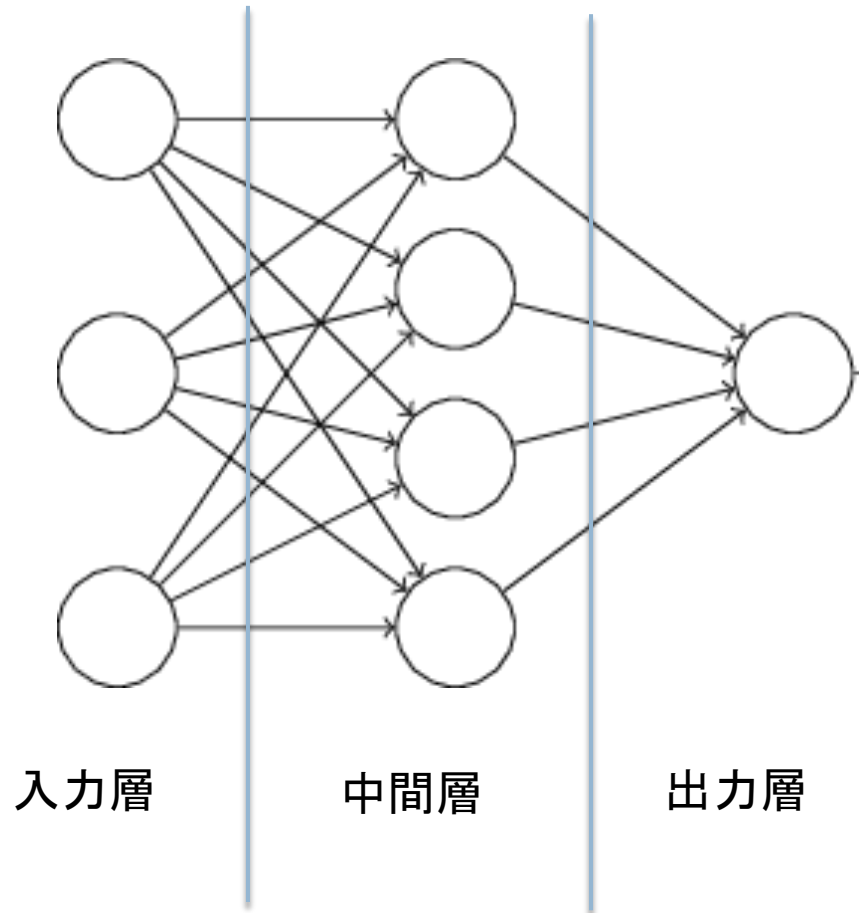
ニューラルネットワーク

ニューラルネットワーク？ (2/2)

7

□ 3種類の層

- 入力層... データを入力
- 中間層... データを推測
 - 複数存在する
 - 重みとバイアスを持つ
- 出力層... 結果表示



中間層での値の伝播

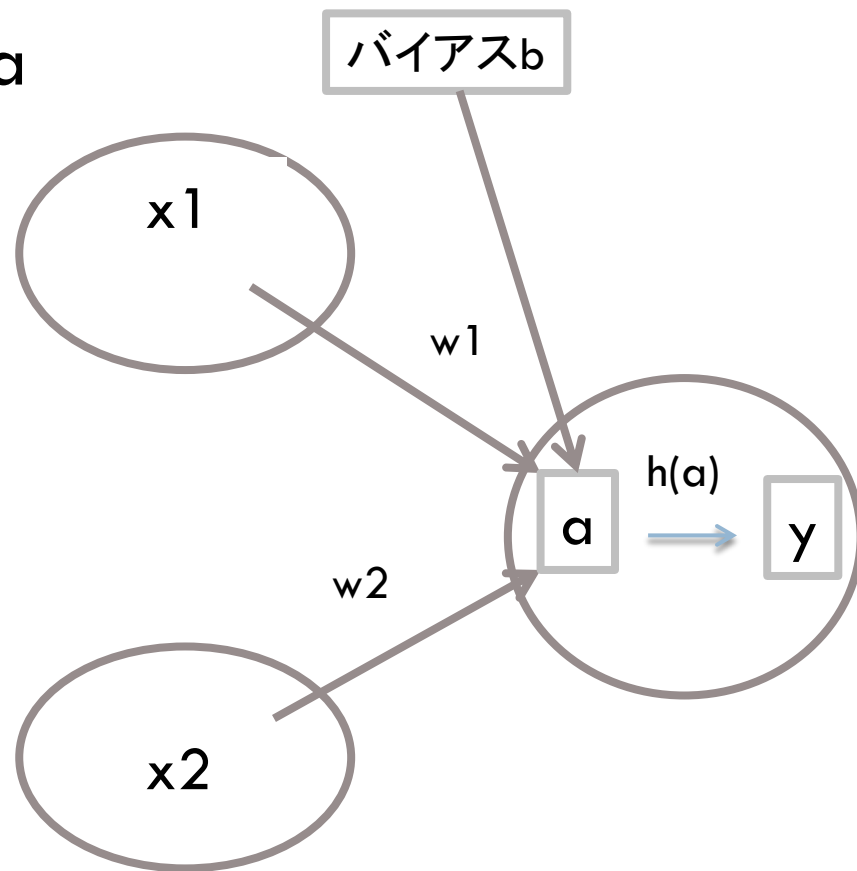
8

- 重み付き入力信号の総和 a を計算

$$a = x_1 w_1 + x_2 w_2 + b$$

- a を活性化関数 $h(a)$ にかけ, 出力

$$y = h(a)$$



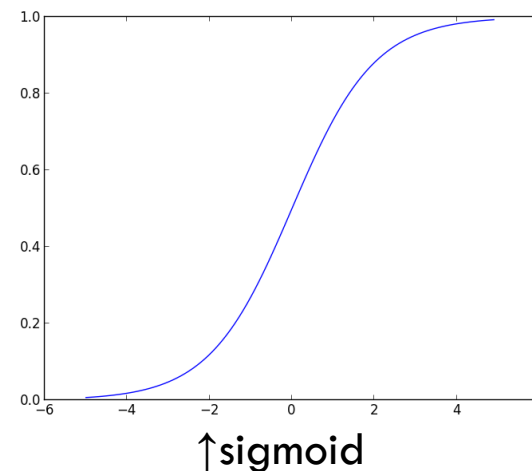
活性化関数

9

□ 入力信号を出力信号へと変換する関数

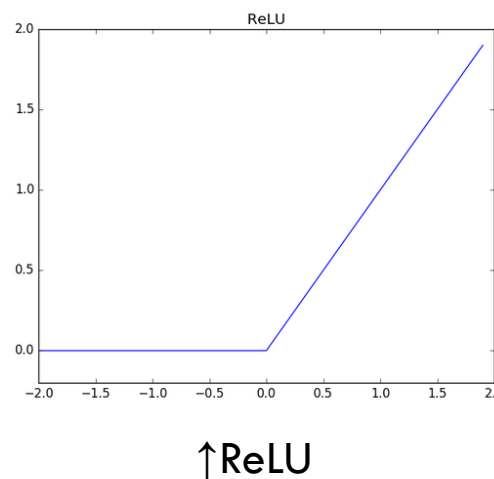
□ Sigmoid関数

- $h(x) = 1 / (1 + e^{-x})$
- 微分の計算が容易



□ ReLU関数

- $h(x) = x(x > 0), 0 (x \leq 0)$
- 2011年, Xavierらによって提唱



□ 一般的に, ReLU関数が用いられる

出力層での処理

10

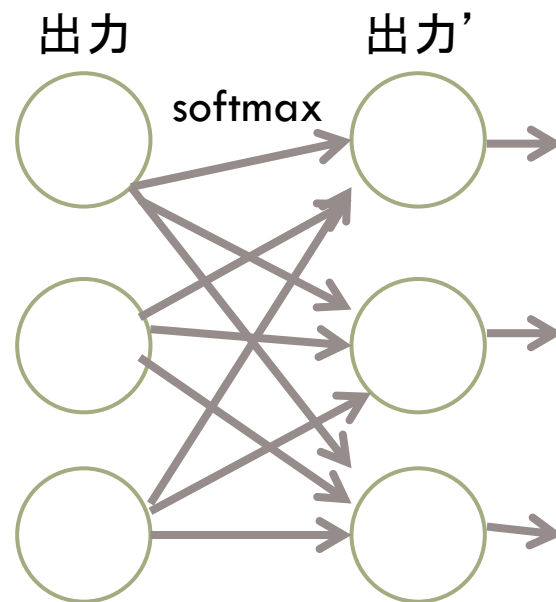
- 中間層を抜けたデータの中から、一番大きい出力データを分類結果とする

- Softmax関数

- 出力層がn個あるとすると、k番目の出力 y_k は

$$y_k = \frac{e^a}{\sum_{i=1}^n e^{a_i}}$$

- 0~1の要素へ出力を変換
 - 信号を確率として捉えられる



ニューラルネットワークの学習

11

- 訓練データから、最適な重み、バイアスを獲得する
 - 損失関数
 - ニューラルネットワークの性能の悪さを判定する
 - モデルによって様々なものが存在

ニューラルネットワークの学習

損失関数の値を小さく抑える
重みパラメータを探す

損失関数の例

12

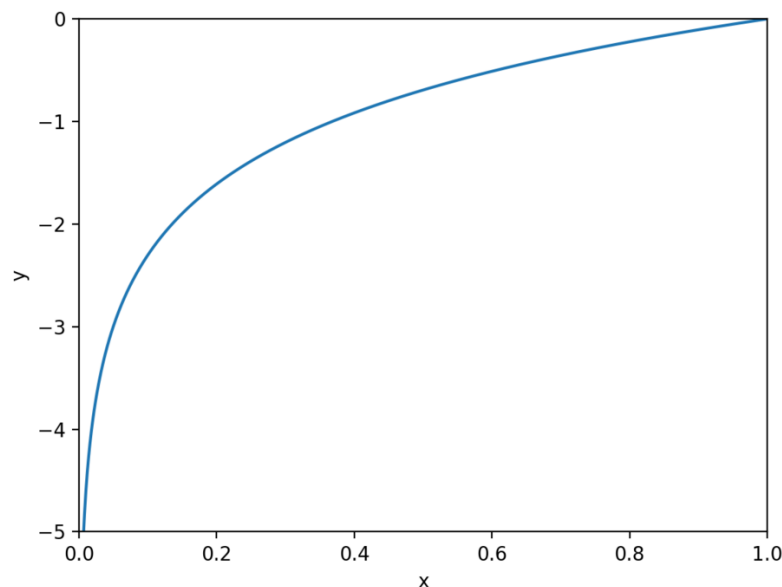
□ 交差エントロピー誤差

□ y_k = ニューラルネットワーク出力

□ t_k = 正解ラベル

■ 正解が1, 他が0のone-hot-vectorを用いる場合がある

$$E = - \sum_k t_k \log y_k$$



重みパラメータの更新

13

□ 勾配法

- ▣ 勾配...数式の全ての変数の偏微分のベクトル
- ▣ 損失関数を重みパラメータで微分, 勾配を求める
 - 勾配の方向(損失関数の値を減らす方向)にパラメータを更新する
 - 更新したら, また勾配を求めて...を繰り返す

勾配法の式

14

- 関数 f が x_1, x_2 の2変数の式であった場合,

$$x_0 = x_0 - \eta \frac{\delta f}{\delta x_0}$$

$$x_1 = x_1 - \eta \frac{\delta f}{\delta x_1}$$

- η ...学習率
 - ▣ どれだけ値を更新するか決定する
 - ▣ 人が決めるハイパーパラメータ

勾配法の式(例)

15

- 重み W , 損失関数 L とすると以下のようなになる

$$W = \begin{pmatrix} w_{11} & w_{21} & w_{31} \\ w_{12} & w_{22} & w_{32} \end{pmatrix}$$
$$\frac{\delta L}{\delta W} = \begin{pmatrix} \frac{\delta L}{\delta w_{11}} & \frac{\delta L}{\delta w_{21}} & \frac{\delta L}{\delta w_{31}} \\ \frac{\delta L}{\delta w_{12}} & \frac{\delta L}{\delta w_{22}} & \frac{\delta L}{\delta w_{32}} \end{pmatrix}$$

- 例えば, $w_{11} = 0.2$ とすると, w_{11} を h 増やすと, 損失関数は $0.2h$ 増える
 - w_{11} をマイナス方向に更新したほうが良い

ここまでのまとめ

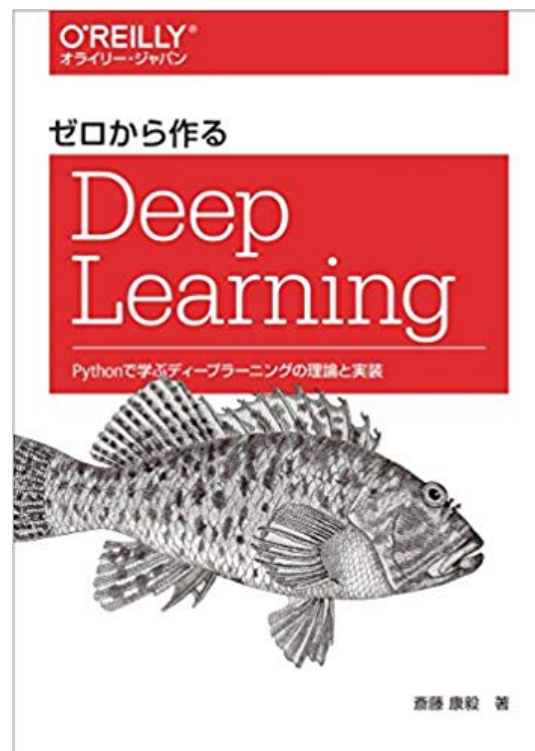
16

- Deep Learning ... 大きなニューラルネットワーク
- ニューラルネットワーク
 - ▣ データを入力
 - ▣ 重みによって、データが変化しつつ中間層を伝播
 - ▣ 出力, 一番大きな値の出力要素を分類結果とする
- 学習は, 損失関数の値を小さくするような重みを探すことで行われる
 - ▣ 各層での出力に関する微分(勾配)の値から算出

出典

17

- ゼロから作るDeepLearning
 - ニューラルネットワーク使うなら読んどくと幸せ
 - (以下内容)
 - 最低限のpythonの勉強
 - ニューラルネットワークについて
 - CNNについて
 - 学習を効率化させる小細工紹介
 - CNNの事例紹介 (2016年現在)

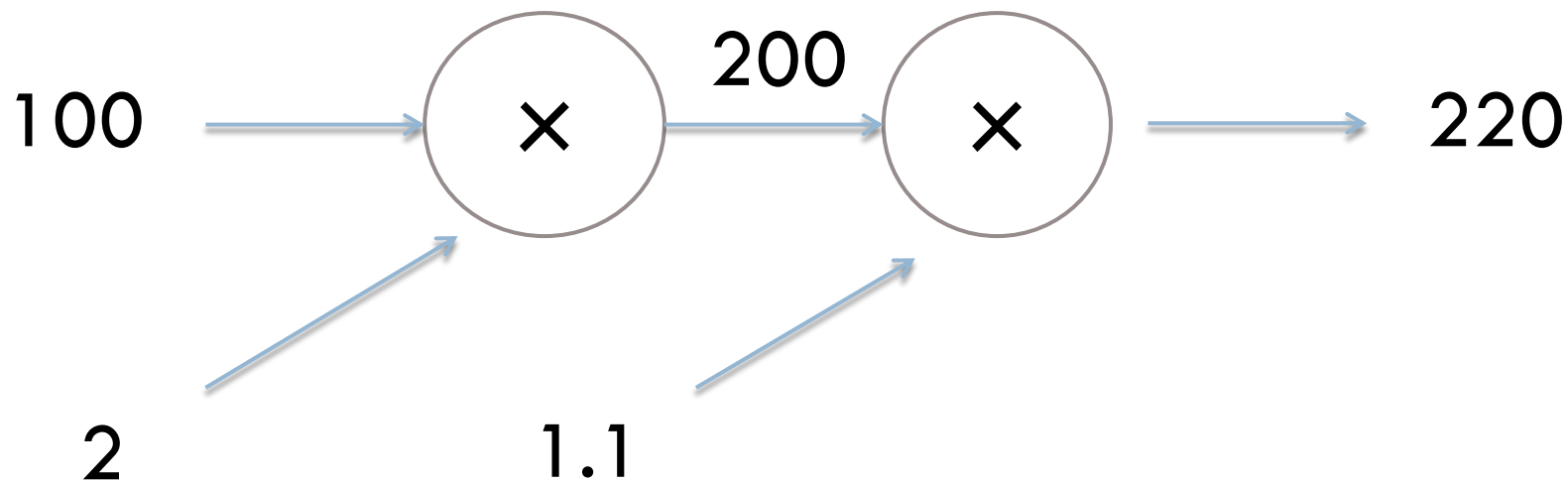


- 以下付録
- よくわかってない事項とも言う

誤差逆伝播法 (1/5)

19

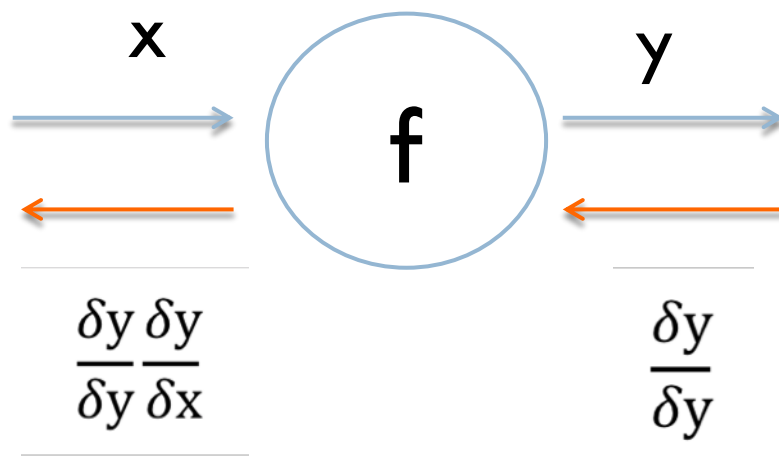
- 高速に各層の微分を行う
 - 計算グラフ
 - 以下例... $(100 \times 2) \times 1.1$



誤差逆伝播法 (2/5)

20

- 計算グラフの逆伝播
 - 順方向の伝播とは逆向き
 - 局所的な微分を計算し, 信号に乗算していく
 - 以下 $y=f(x)$ の例



誤差逆伝播法 (3/5)

21

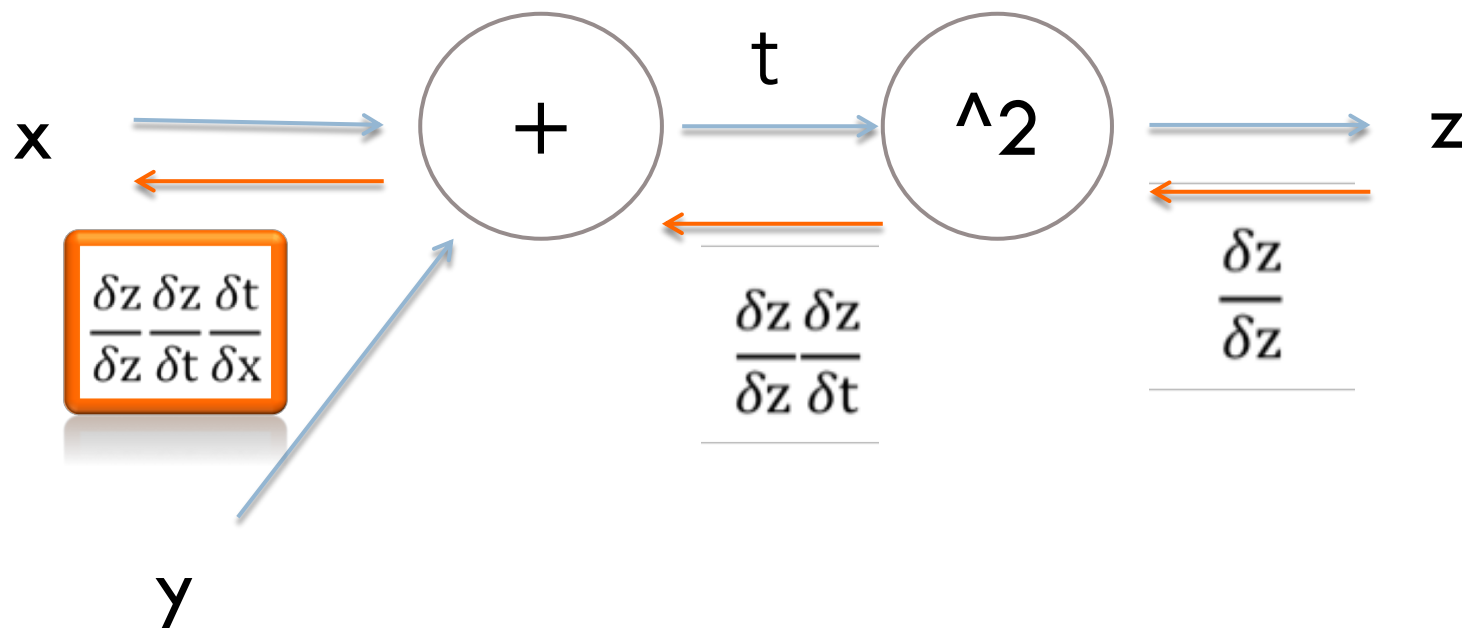
- 連鎖律に基づいている
 - ある関数が合成関数で表される場合, その合成関数の微分は合成関数を構成するそれぞれの関数の微分の積で表すことができる
 - 例... $z = t^2$, $t = x + y$ とすると, x に関する z の微分は

$$\frac{\delta z}{\delta x} = \frac{\delta z}{\delta t} \frac{\delta t}{\delta x}$$

誤差逆伝播法 (4/5)

22

- $z = t^2$, $t = x + y$, x に関する z の微分を求める
 - 計算グラフを使ってみる
 - 逆伝播により, 効率的に微分が求められる



誤差逆伝播法 (5/5)

23

□ 実際の利用

□ $Y = XW + B$

■ Y...出力 X...入力 W...重み B...バイアス

