

教師あり学習（感情分析の場合）

$$\text{学習データ } D = \{(x_1, y_1), \dots, (x_N, y_N)\}$$

・ x は 文.

・ $y \in \{1, 2, 3, 4, 5\}$

“れくらいユーザ”が

商品に対して肯定的か？

みたいな数値（定義はゆるい）

一般的に生文だと計算機であつかいにくい
ので 前処理をおこなう。

文の分割、形態素解析など
(mecab)

学習するもの：関数 $f: x: \text{文} \mapsto \text{感情スコア} \in [1, 5] \subset \mathbb{R}$

例として 線形モデル を考えてみる。

深層学習ではない。

感情分析(数値ポイント)	感情分析(数値ポイント)	感情分析(数値ポイント)
CMで見た新しいチークがすごく良さそう。特にベージュ系の色が綺麗だったから絶対買おうと思う。	ポジティブ	+2
先週買った口紅、チラシでみた印象より色がマイナチで、全然合わなかった。	ネガティブ	-2
ロクシタンのハンドクリーム、値段は少し高いけどコスパが良いからおすすめ！	ポジティブ	+1
ハーブ系のハンドクリームは、匂いにクセがあって好きじゃない。	ネガティブ	-1
初めてBBクリームを試してみたけど、悪くないね。	ニュートラル	0

$$f_w(x) = w^T \phi(x).$$

「 w は x -タに依存して3

ここで

• $w \in \mathbb{R}^d$ 重みベクトル (パラメータ)

• $\phi(x) \in \mathbb{R}^d$ 素性ベクトル (特徴ベクトル)

$$= \begin{pmatrix} \phi_1(x) \\ \vdots \\ \phi_d(x) \end{pmatrix} \quad \text{feature vector.}$$

例:

$$\phi_1(x) = \mathbb{1}(x \text{ に「悪」が含まれる}) \quad \phi_2(x) = \mathbb{1}(x \text{ に「好き」が含まれる})$$

$$\phi_{103}(x) = \mathbb{1}(x \text{ に「悪く」+「ない」が含まれる})$$

↑
条件

$\left(\begin{array}{l} \mathbb{1}(\text{条件}) = \text{条件が成り立つは } 1 \\ \text{そうでなければ } 0 \end{array} \right)$

$$f \begin{pmatrix} \text{BB 71-6} \\ \text{試してみた} \\ \text{1+2「悪くない」} \end{pmatrix} = \begin{pmatrix} -3 \\ 5.8 \\ 4 \\ 0.2 \end{pmatrix}^T \begin{pmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 1 \end{pmatrix} \leftarrow \begin{array}{l} \phi_1(x) \\ \phi_2(x) \\ \vdots \\ \phi_{103}(x) \end{array} = 2.8$$

$x \quad w \quad \phi(x)$

学習：重みベクトルの中の値を調節して、
できるだけ多くの文に正しいスコア
を当てられるようにする。

どうするか：
①どれくらい間違えているか（損失；loss）を評価する。
②損失が小さくなるように重みを調整する。

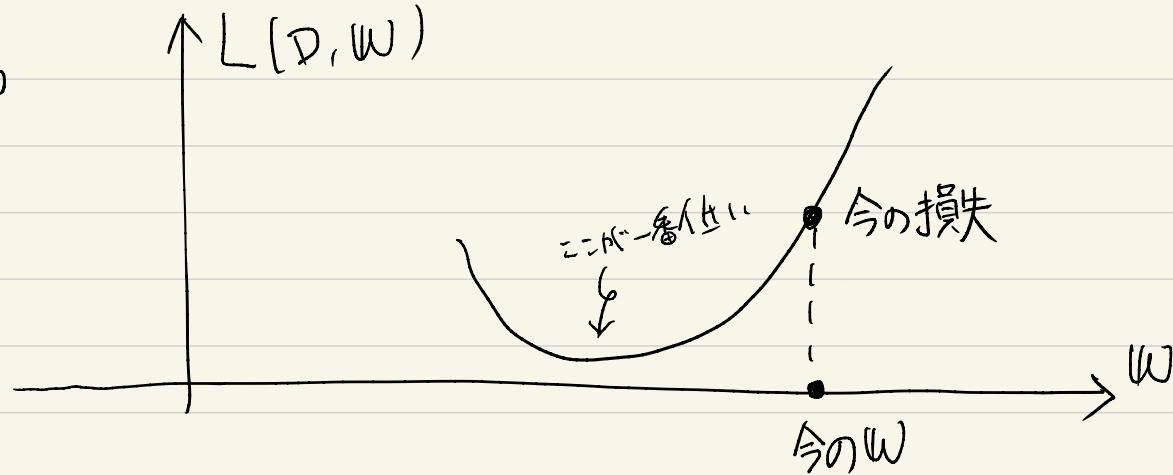
① 損失の計算：（例えば）二乗誤差関数

$$L(D, w) = \sum_{(x, y) \in D} (y - f_w(x))^2$$

正解のスコア
 $\{1, 2, 3, 4, 5\}$

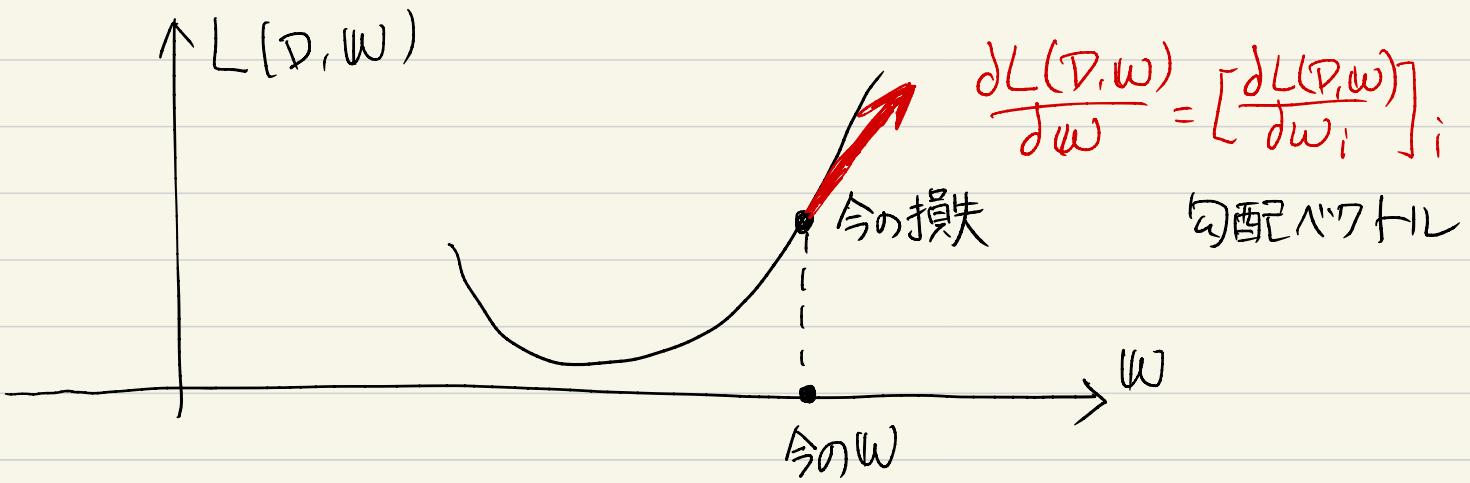
線形モデルの予測
結果

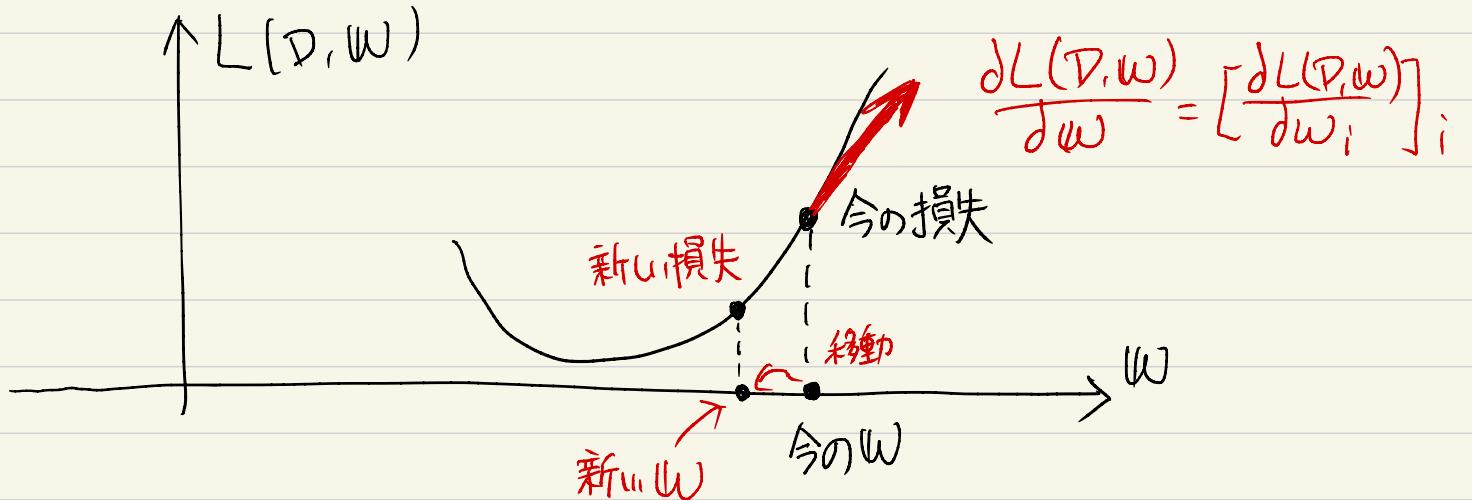
損失と
パラメータの
関係：



損失を下げるには： w をもう少し左に移動させれば良い。
図を描かず”に移動する方向はどうやって調べられるか?
微分して勾配を調べる。

$$\frac{\partial L(D, w)}{\partial w_i} = \sum_{(x, y) \in D} \left\{ 2(y - w^T \phi(x)) \times (-\phi_i(x)) \right\}$$





$$\text{新しい } w = \text{今の } w - \alpha \frac{\partial L(D, w)}{\partial w}$$

↑
学習率 (1より小さく設定する)

と重みを調整

これで“損失(間違え具合)が減る！”(一般的に)
勾配降下法(Gradient Descent)という。

ここまでまとめ.

- ① データを用意 $D = \{(x_1, y_1), \dots, (x_N, y_N)\}$
- ② 関数を用意 (線形モデルなど) $f_w(x) = w^T \phi(x)$
- ③ 学習する.

1) 損失の計算 $L(D, w)$

2) 勾配の計算 $\partial L(D, w) / \partial w$

3) 勾配降下 (w を更新)

深層学習は？

線形モデルのいいところ

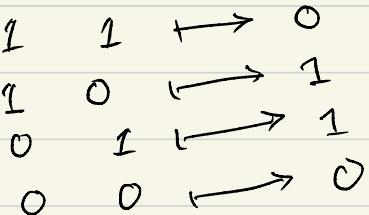
- 学習が安定
- 解釈性が高いなどいいところもあるやい

悪いところとして

- 性能がいくい！
- 素性ベクトルの設計に依存

有名な例として、

線形モデルでは XOR を表現できぬ！



深層学習モデル

$$f(x) = w_n \left(\dots \left(w_2 \sigma \left(w_1 \phi(x) + b_1 \right) + b_2 \right) \dots \right)$$

のように、雑に言ってしまえば複雑になっただけ。
学習等はまったく同じで、微分して勾配降下。

関数が“ -7° ”

複雑なので、人手で微分するのは
大変

\Rightarrow Chainerの仕事。
(自動微分)

確率はどこで使うか

今回は二乗誤差を使つたが、

問題によつては他の損失を使う。そこで確率の言葉をつかう。

例：文章分類では KL ダイバージェンス が 損失

