機械学習講習会 第五回

-「ニューラルネットワークの学習と評価」

traP アルゴリズム班 Kaggle部

2023/6/28

第五回: ニューラルネットワークの学習と評価

✅ 「ニューラルネットワーク」を導入して、

任意の関数を表現できるようになった！

振り返りタイム

- 我々の学習手法は、$f(x) = ax + b$というモデルの構造自体に直接依存しているわけではなかった
- そして、$f(x) = ax + b$というモデルの構造では直線しか表現することができないので、違う形を考えることにした
- 「基になる」簡単な関数の、合成と和を考えることでかなり複雑な関数も表現できることがわかった
- 「基になる」関数の選び方を考える上で、この関数自体もパラメータによって変化させるモデルとして、ニューラルネットワークを導入した
- ニューラルネットワークは任意の関数を表現できることがわかった

今日やること

ニューラルネットワークは非常に多くのパラメータをもつ
(全結合層は、$W \in \mathbb{R}^{n \times m}$と$b \in \mathbb{R}^m$のパラメータを持つ)

⇨ 学習は、かなりむずかしいタスク

ニューラルネットワーク研究の歴史を遡ってみると...

- 1950年代: パーセプトロン
- 1986年ごろ?: 多層パーセプトロン
  → ニューラルネットで全部表現できる！すごい！！
  → 数学的な研究も進み始める

今日やること

1990年 ~ 2000年代ごろ

- ニューラルネットワークを大きくしていくと全然学習がうまくいかない
  (= まともなパラメータを獲得してくれない)

⇨ ニューラルネットワークはオワコン！

< :oisu-:

ジェフリー・ヒントン(Geoffrey Everest Hinton)

- オートエンコーダによる初期化など、学習を安定化させる方法を提案
  ⇨ ニューラルネットワークの学習ってがんばればうまくいく？
  ⇨ 2010年代に入って、ニューラルネットワークが再び注目される

おしながき

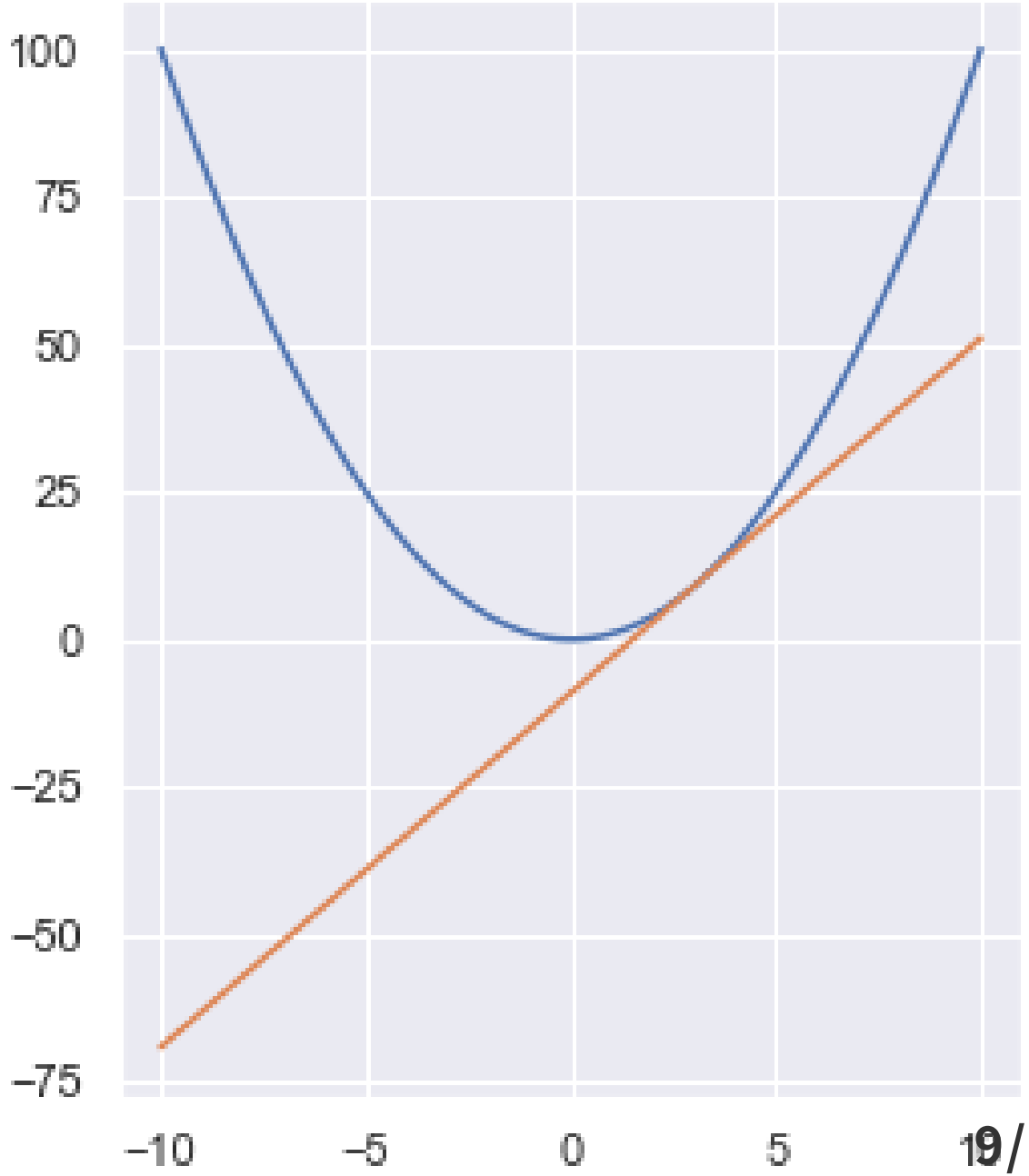
- 勾配降下法をニューラルネットワークに適用するための工夫
- ニューラルネットワークを評価する

微分係数

- 微分係数 $f'(x)$ は、
  xにおける接線の傾き

  ⬇

  $-f'(x)$**方向に関数を少し動かすと、関数の値はすこし小さくなる**



勾配降下法

関数 $f(x)$ と、初期値 x_0 が与えられたとき、
次の式で x を更新する

$$x_{n+1} = x_n - \eta f'(x_n)$$

(η は**学習率**と呼ばれる定数)

今日やること1.「ニューラルネットワーク向け」の学習

勾配降下法... $x_{n+1} = x_n - \eta f'(x_n)$

をニューラルネットワークに適用するための色々な技法

- 初期化 (x_0を決める)

⇩

- 計算 ($x_{n+1} = x_n - \eta f'(x_n)$を計算する)

のそれぞれをカスタマイズします

ニューラルネット向けに書き直すとパラメータをならべたベクトル$\boldsymbol{\theta}$と、$\boldsymbol{\theta}$の関数である損失関数$L(\boldsymbol{\theta})$に対して$\boldsymbol{\theta}_{n+1} = \boldsymbol{\theta}_n - \eta \nabla L(\boldsymbol{\theta}_n)$と定義される式に沿って更新するという感じになります。

今日やること1. 初期値

勾配降下法... $x_{n+1} = x_n - \eta f'(x_n)$

x_0はわれわれが決めなければいけなかった！
⇨ どう決めるのがいいのか？

1. Xavierの初期値

2. Heの初期値

Xavierの初期値

2010年、Xavierらが提案

$$W_{i,j} \sim U\left[-\sqrt{\frac{1}{n}}, \sqrt{\frac{1}{n}}\right]$$

(nは全結合層の入力の次元)

おきもち... 活性化関数の微分が「いい感じ」に働く分布になるよう初期値を置く

“Understanding the difficulty of training deep feedforward neural networks" [Glorot and Bengio, 2010]
https://proceedings.mlr.press/v9/glorot10a/glorot10a.pdf

Heの初期値

2015年、Kaiming Heらが提案

$$W_{i,j} \sim \mathcal{N}\left(0,\ \sqrt{\frac{2}{n}}\right)$$

https://arxiv.org/abs/1502.01852v1

初期値の決め方

- 初期値の決め方は知見が蓄積されている
- 活性化関数などの性質によってこういう初期化がいい、みたいな知見もある
  - 例) 活性化関数に$\sin$を使うSIRENと呼ばれるモデルでは $U[-\sqrt{6/n}, \sqrt{6/n}]$がいいなど
- たいていの場合はフレームワークのデフォルトの初期化方法を使えばいいので、特にいじる場面は多くない(最近は特に必要性が薄れている... 正規化層)
- 特殊なネットワークを作るときには、初期化方法をいじるといいかもしれない

ミニバッチ学習

☑ 初期化 (x_0を決める)

⇩

☐ 計算 ($x_{n+1} = x_n - \eta f'(x_n)$を計算する)

のそれぞれをカスタマイズします

ミニバッチ学習

損失関数は

$$L(\boldsymbol{\theta}) = \frac{1}{N}\sum_{i=1}^{N}\left(y_i - f(\boldsymbol{x}_i;\boldsymbol{\theta})\right)^2$$

⇨ これを計算するには、N回の計算が必要

⇨ データ数が少ない場合はいいが、数GB, TB, ...となっていく

⇨ メモリに乗り切らず実用的な速度で計算することが難しくなる！

ミニバッチ学習

✅ データをランダムにいくつか選んで、そのデータだけを使って損失関数を計算する

例) データ数$N = 10000$から$s = 100$個のデータをランダムに選ぶ

⇨ メモリに乗り切って計算できる！

この選んだ小さいデータの集合のことを**ミニバッチ**と呼び、
そのサイズを **バッチサイズ(batch size)** と呼ぶ

そしてこれを使った勾配降下法を「確率的勾配降下法」という

確率的勾配降下法を単一データのみの損失を計算して行う手法と呼ぶ派閥もいるみたいですが、ミニバッチ学習に対しても確率的勾配降下法と呼ぶ人が多いと思います。

- 計算 ($x_{n+1} = x_n - \eta f'(x_n)$を計算する)

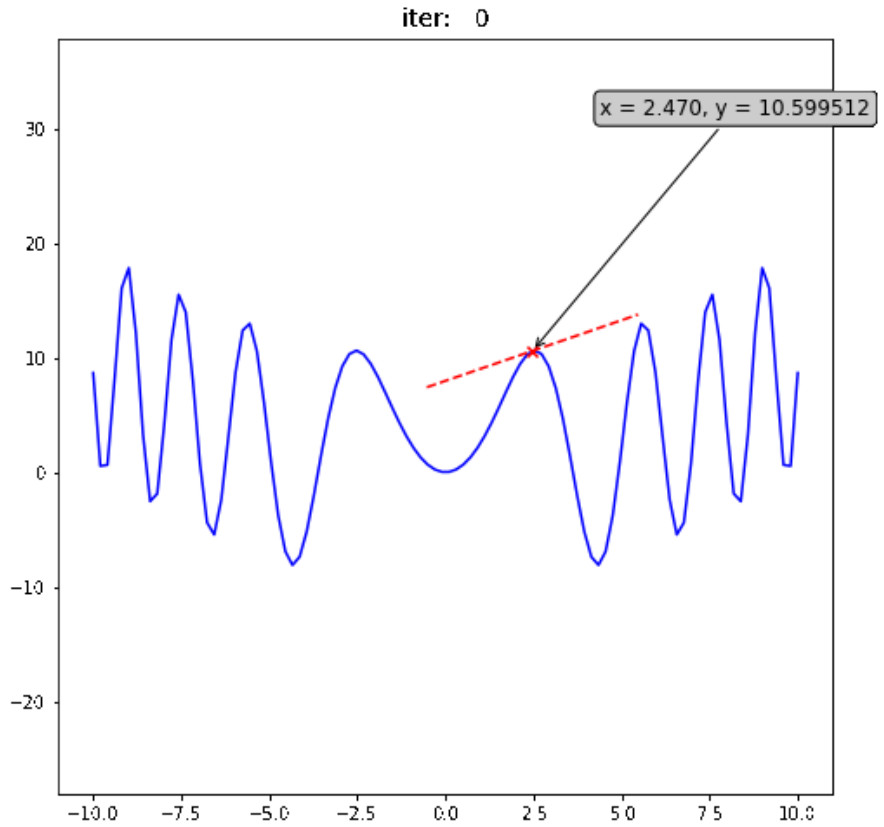
の、$L(\boldsymbol{\theta_n})$の計算はできるようになった！

Next

⇨ 更新式自体もいじる

オプティマイザ

局所最適解 ... 付近では最小値だが、全体の最小値ではない
大域最適解 ... 全体で最小値



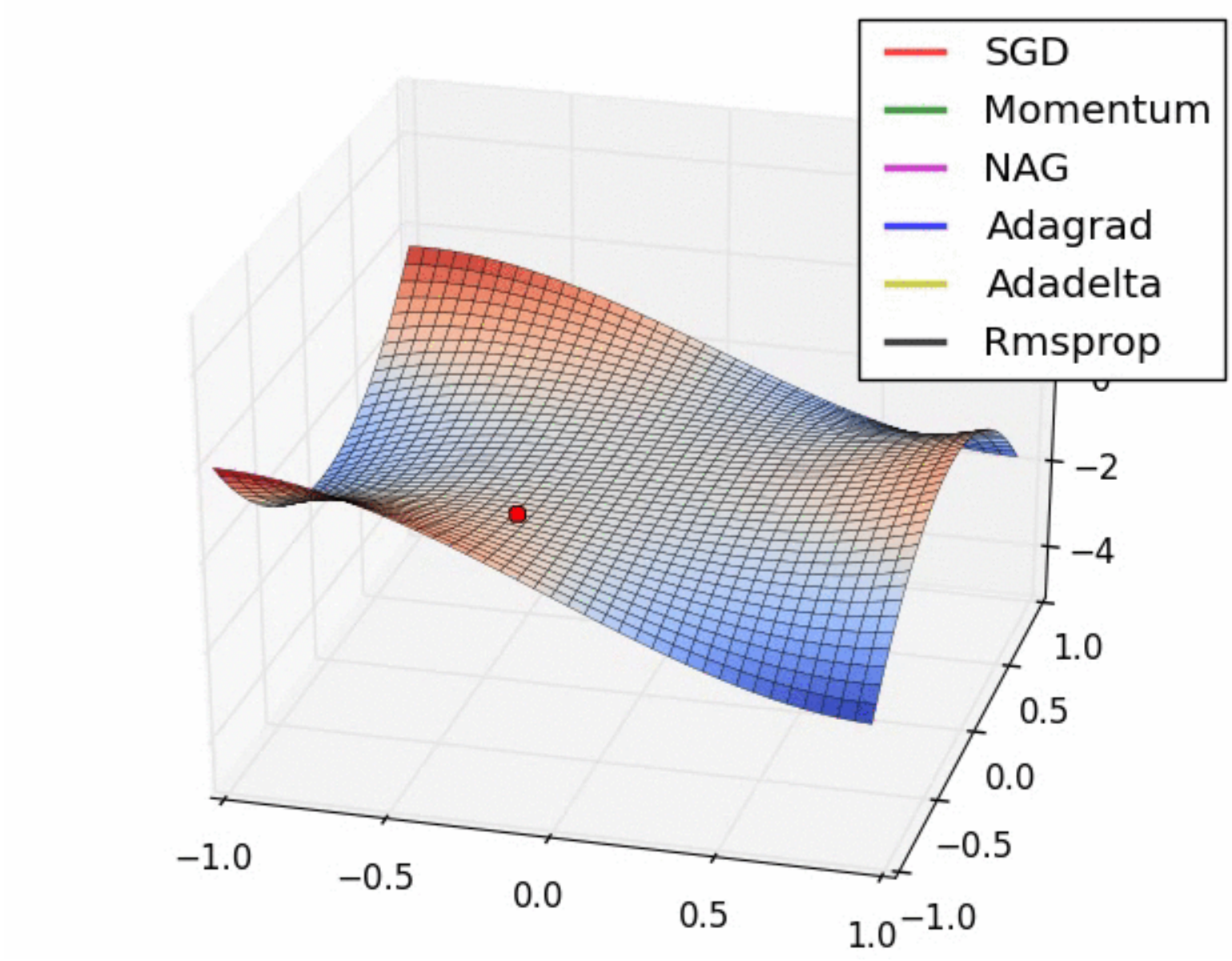
オプティマイザ

局所最適解にハマりにくい工夫された更新式をもつ色々なオプティマイザが提案されている

$Adadelta, Adagrad, Adam, AdamW, SparseAdam, Adamax, ASGD, LI$

他にも学習率(η)を変化させる手法など、いろいろと工夫されている

⇨ 基本的に、Adam(とその派生系)とSGDが使われることが多い
(使い方は第6回でやります！)


SGD
Momentum
NAG
Adagrad
Adadelta
Rmsprop
−2
−4
1.0
0.5
0.0
−0.5
−1.0
−1.0
−0.5
0.0
0.5
1.0

学習

- [x] 初期化 (x_0を決める)

⇩

- [x] 計算 ($x_{n+1} = x_n - \eta f'(x_n)$を計算する)

✅ 「学習」部分は完了

学習した後のことを考えよう

- 例) アイスの予測ができるモデルが完成した！！！

⇨ こいつの「良さ」をどう定義するべきか

今までは、損失関数の値が小さいほど良いと考えていた(学習時)

⇩

学習した後のことを考えよう

例) アイスの値段予測をするモデルを作った！

学習の際に使ったデータは、
(20℃, 300円), (25℃, 350円), (30℃, 400円), (35℃, 450円), (40℃, 500円)

⇨ さぁこれを使ってアイスの値段を予測するぞ！
⇨ 来るデータは....

(22℃, 24℃, 25℃, ...)

※ 重要: これらの正解データは(少なくとも推論をする段階では)

存在しない

学習した後のことを考えよう


< なんか来月の予想平均気温30度って気象庁が言ってたな。
来月の売り上げが予想できたらどのくらい牛乳仕入れたらいいかわかって嬉しいな。

⇨ われわれの全体としての本当の目的はこれらの答えが存在していないデータに対して精度良く推論すること

学習した後のことを考えよう

ところが、まだ答えが存在していないデータを使って学習を行うことはできない！
(なぜなら、答えがないので誤差が定義できない)

⇨ (20℃, 300円), (25℃, 350円), ... ,(40℃, 500円)　に対して
妥当な推論が行えるようになったのなら、21℃や26℃に対しても「ある程度」妥当な推論が行えるようになっているはず

このように知らないデータに対しても推定が行えることを**汎化**といいます。かっこいいですね。この汎化をどのように実現するかと言うのが機械学習の主要な課題です。

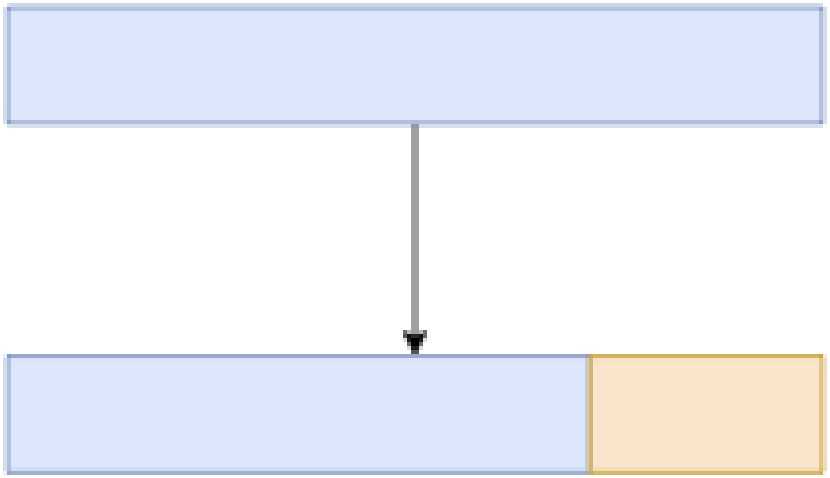
学習した後のことを考えよう

学習後に関心があること: 未知のデータへの予測性能

⇨ これを検証しておこう

バリデーション

- 学習データを分割して、一部を学習に使い、残りを検証に使う



未知のデータに対する推論制度を検証する

学習データ

(20℃, 300円), (25℃, 350円), (30℃, 400円), (35℃, 450円), (40℃, 500円)

⇩

- 学習データ
  (20℃, 300円), (25℃, 350円), (30℃, 400円)
- 検証用データ
  (35℃, 450円), (40℃, 500円)

未知のデータに対する推論制度を検証する

- 学習データ
  (20℃, 300円), (25℃, 350円), (30℃, 400円)

のみで学習をおこなう

⇩

(35℃, 450円), (40℃, 500円)に対して推論を行い、誤差を評価

400円、500円と推論したとすると、
「検証用データに対する」平均二乗誤差は

$$\frac{1}{2}\left((400-450)^2+(500-500)^2\right)=1250$$

未知のデータに対する推論制度を検証する

- 学習データ
  (20℃, 300円), (25℃, 350円), (30℃, 400円)

のみで学習し、検証用データは学習に関与させない

何をしている？

⇨ 擬似的に未知のデータを作成して、

「未知のデータに対する推論の精度」を擬似的に評価している

損失関数と評価指標

これらは、学習とは全く独立した作業
⇨ これの計算結果に基づいてモデルが影響を受けることはない

⇩

損失関数が満たす必要があった

- 微分可能
- 微分がほとんど0にならない

などの条件は必要ない！

この検証用データに対して定義される評価の指標を
「評価指標」 という。

\ 重要 / \ 重要 / \ 重要 /

「損失関数」と評価指標は別の概念

\ 重要 / \ 重要 / \ 重要 /

例) アイスの予測では損失関数として平均二乗誤差を使ったが、
別に評価指標が平均百乗乗誤差でもok.

損失関数と評価指標

これらは、学習とは全く独立した作業
⇨ これの計算結果に基づいてモデルが影響を受けることはない

⇩

逆にいえば評価指標は直接最適化の対象にはならない！

⇨ うまく設計された損失関数を使って、
評価指標の値を最適化できるように頑張る

未知のデータに対する推論制度を検証する

つまり...

損失関数の値は、あくまで「訓練データに対してこれくらいの誤差になるよ」という値

⇨ われわれが興味があるのは、知らないデータに対してどれくらいうまく予測できるか

これの検証のために、擬似的に学習に使わない未知のデータを作り、未知のデータに対する予測の評価をする

バリデーションの手法や切り方についてはいろいろあり、話すとかなり長くなりますのでここでは割愛します。メジャーなものだとCross Validationや時系列を意識したValidationなどがあります。

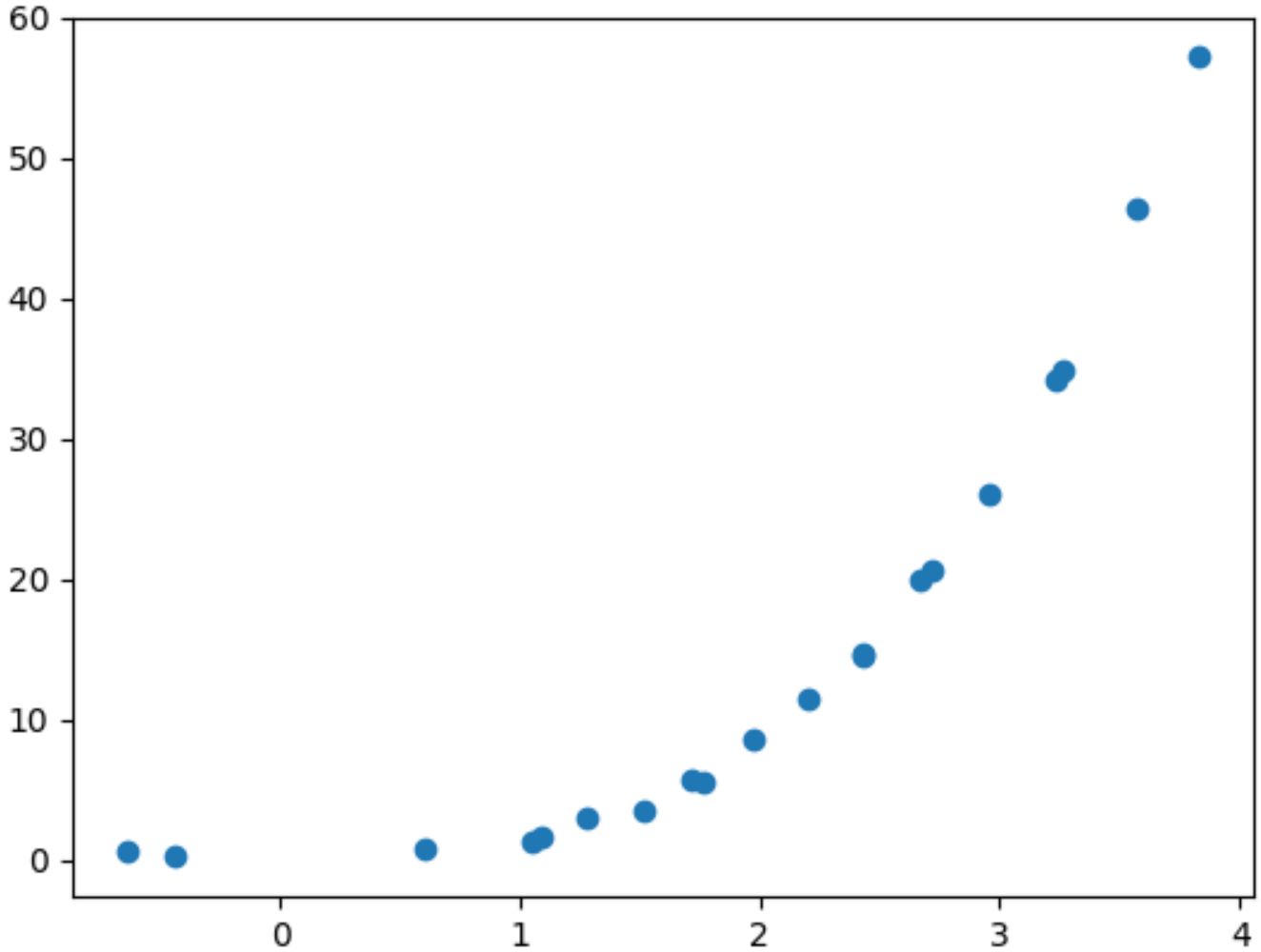
未知のデータに対する推論制度を検証する

ニューラルネットワークの万能近似性から、(矛盾のないデータからうまく学習さえできれば)　学習時の損失を0にすることができる

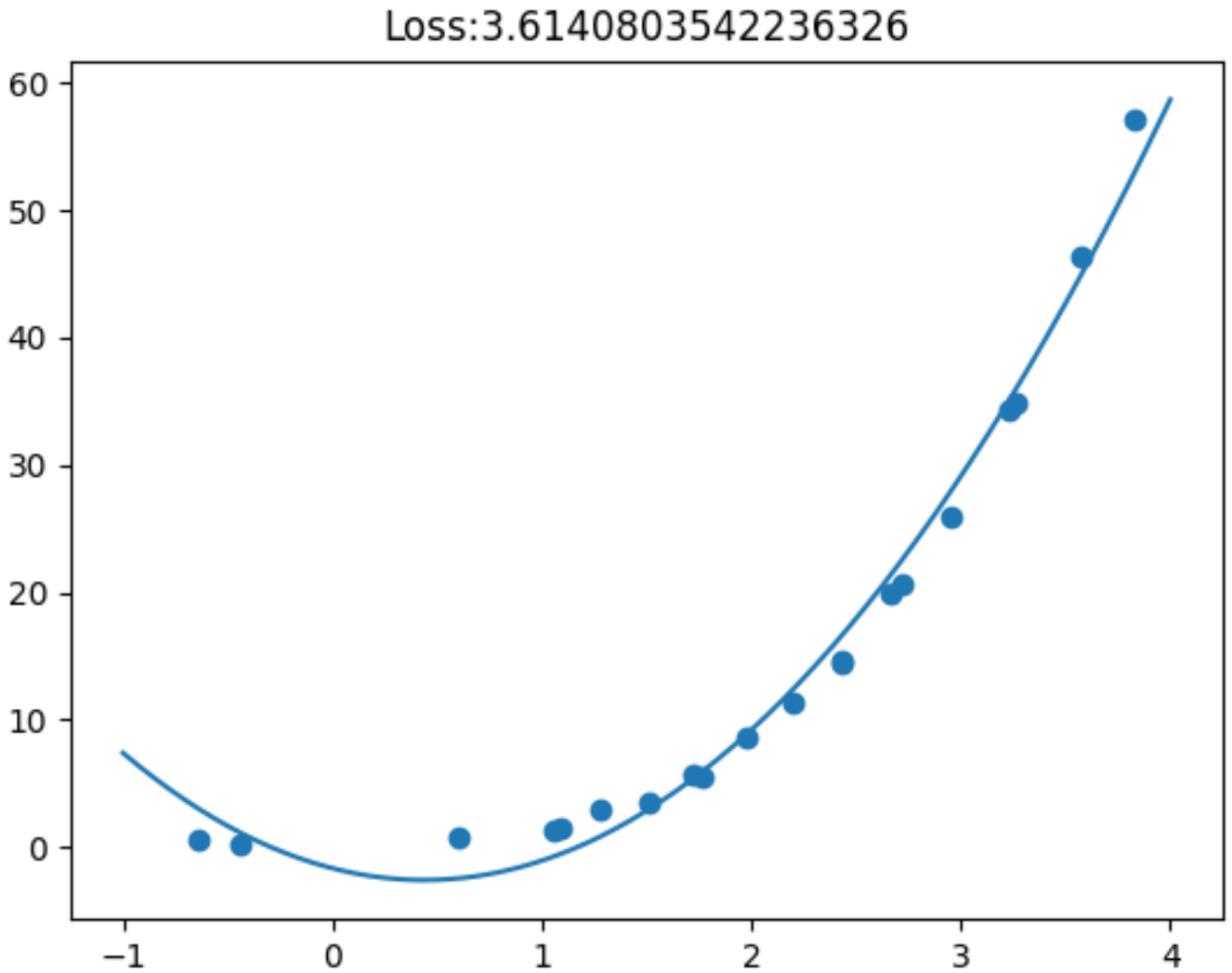
⇨ 一方で、検証用データに対しては、損失が大きくなる場合がある

過学習

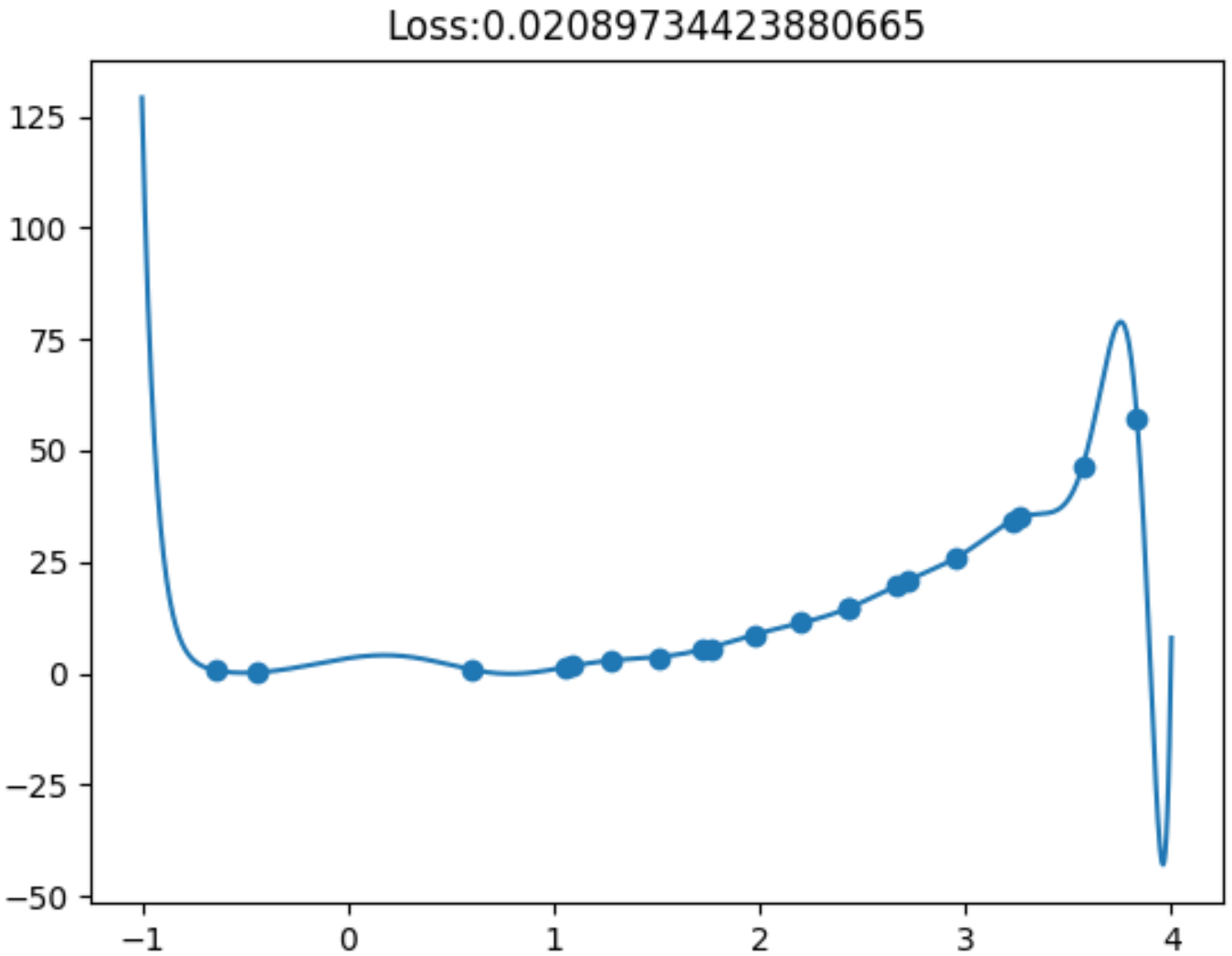
過学習



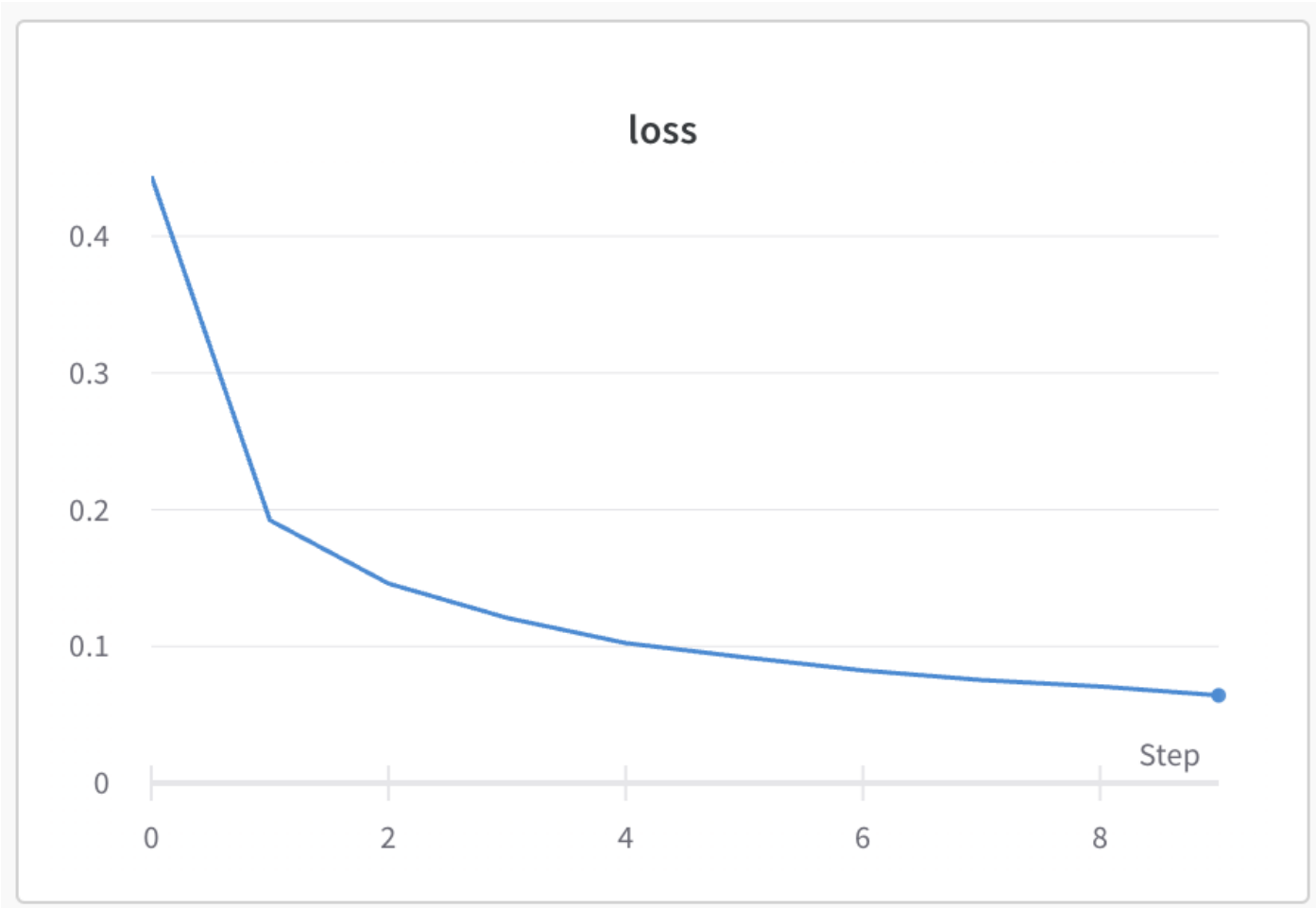
過学習



過学習



バリデーションと学習曲線(learning curve)



横軸... 学習ステップ
縦軸... 評価(学習時 or バリデーション)

バリデーションの重要性について

＜「AI作りました！ちなみにどのくらいの精度で動作するかはわからないです笑」

⇨ きちんとバリデーションを行い、未知のデータに対する予測性能を評価することが大切

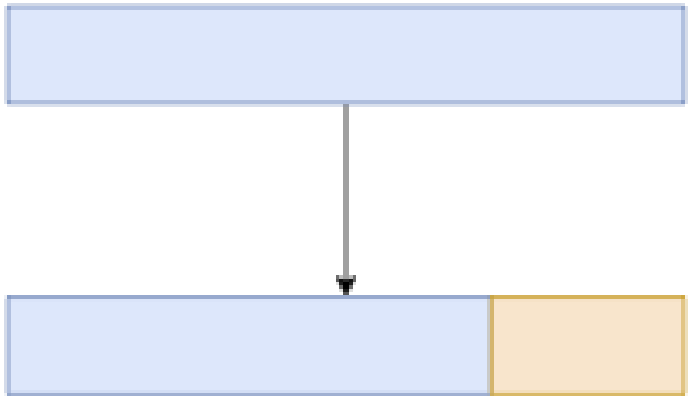
⇨ 逆に、適切にバリデーションを行なっていないが故の嘘に気をつけよう！！

~ Learning Curve 閲覧 TIME ~

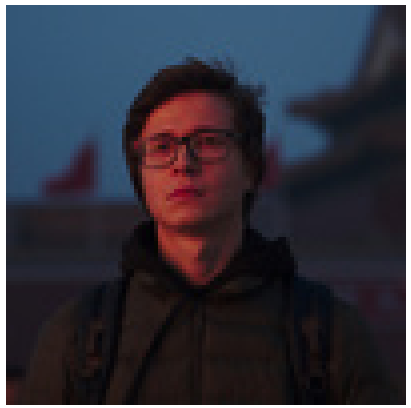
バリデーションの重要性について

Kaggleをはじめとするデータ分析コンペは、
「未知の情報」を予測するモデルの精度を競う

⇨ 試行錯誤している手法の、「未知の情報を予測する能力」(=**汎化性能**)
をきちんと評価することが大切！！！！！！！！



バリデーションの重要性について



< "A good CV is half of success."

bestfitting

今日のまとめ

- ニューラルネットワークは、学習において培われてきたいろいろな工夫があった
- バリデーションを行うことで、未知のデータに対しての予測性能「**汎化性能**」を評価することができる
- バリデーションデータに対して行う評価は、学習とは独立した作業なので、微分可能であったり微分の性質が良い必要はなく、いろいろな評価指標を用いることができる
- 訓練データのみに過剰に適合した状態のことを「過学習」といい、学習曲線に目を光らせるととでこれに気をつける必要があった