

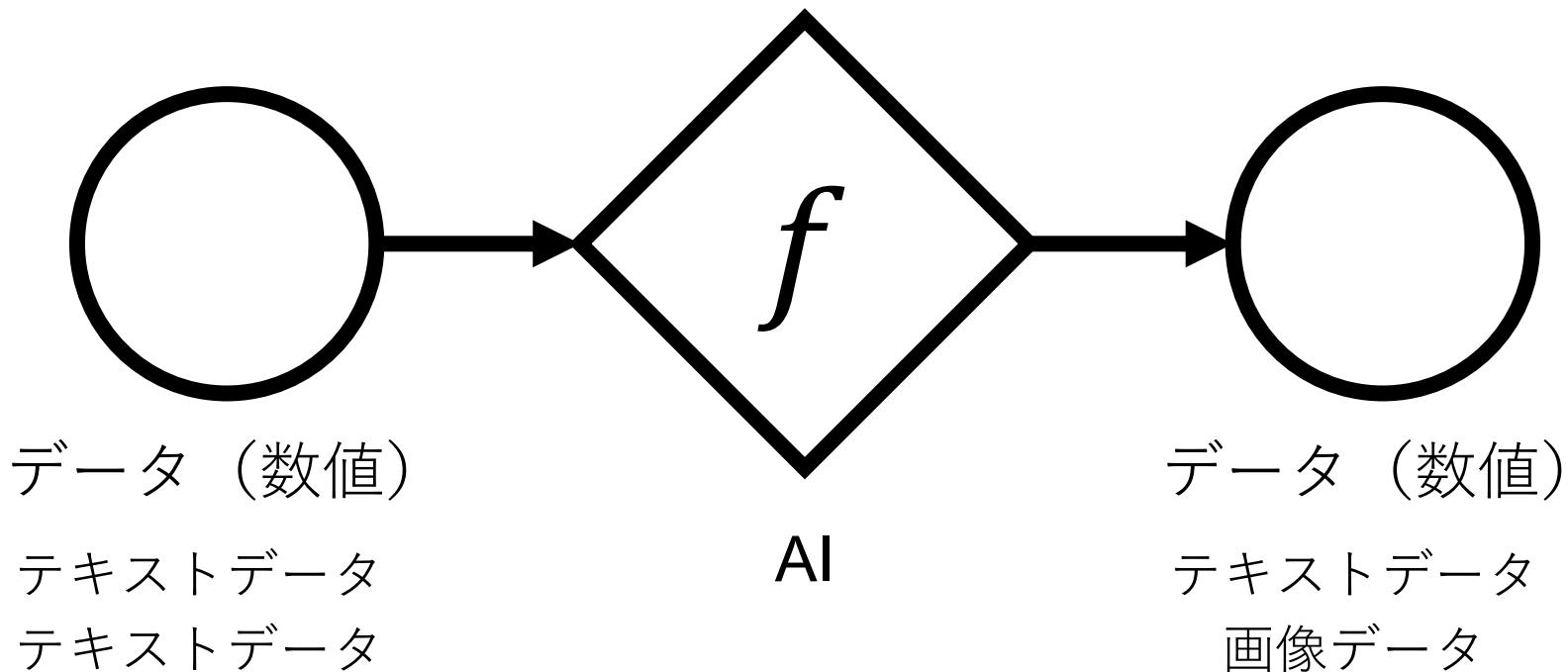
深層學習

先進医用画像解析処理研究室

AIとは

AIの概念

AIは、データを異なるデータへ変化する関数



機械学習

コンピュータが「自分で学ぶ」仕組み

1. データを集める

「リンゴ」「バナナ」などの果物の写真と名前をコンピュータに見せる。

2. ルールを学ぶ

リンゴとバナナの違いを見つけ出します。たとえば「赤いものはリンゴ」「細長いものはバナナ」というルールを作る。

3. 新しいデータを予測する

新しい果物の写真を見せたとき、「赤い画像だ」⇒「これはリンゴ！」と判断します。

深層学習

機械学習の中の特別な方法

1. 特徴を自分で学ぶ

機械学習では人間が特徴（例：猫の耳が尖っているなど）を教えますが、深層学習はコンピュータが自分で特徴を見つけ出します。

2. ニューラルネットワークを利用

深層学習は、人間の脳を真似た「ニューラルネットワーク」という仕組みを使います。層（レイヤー）がたくさんある「深い」構造なので「深層」と呼ばれます。

主なタスク

画像分類

- 動物の分類

物体検出・セグメンテーション

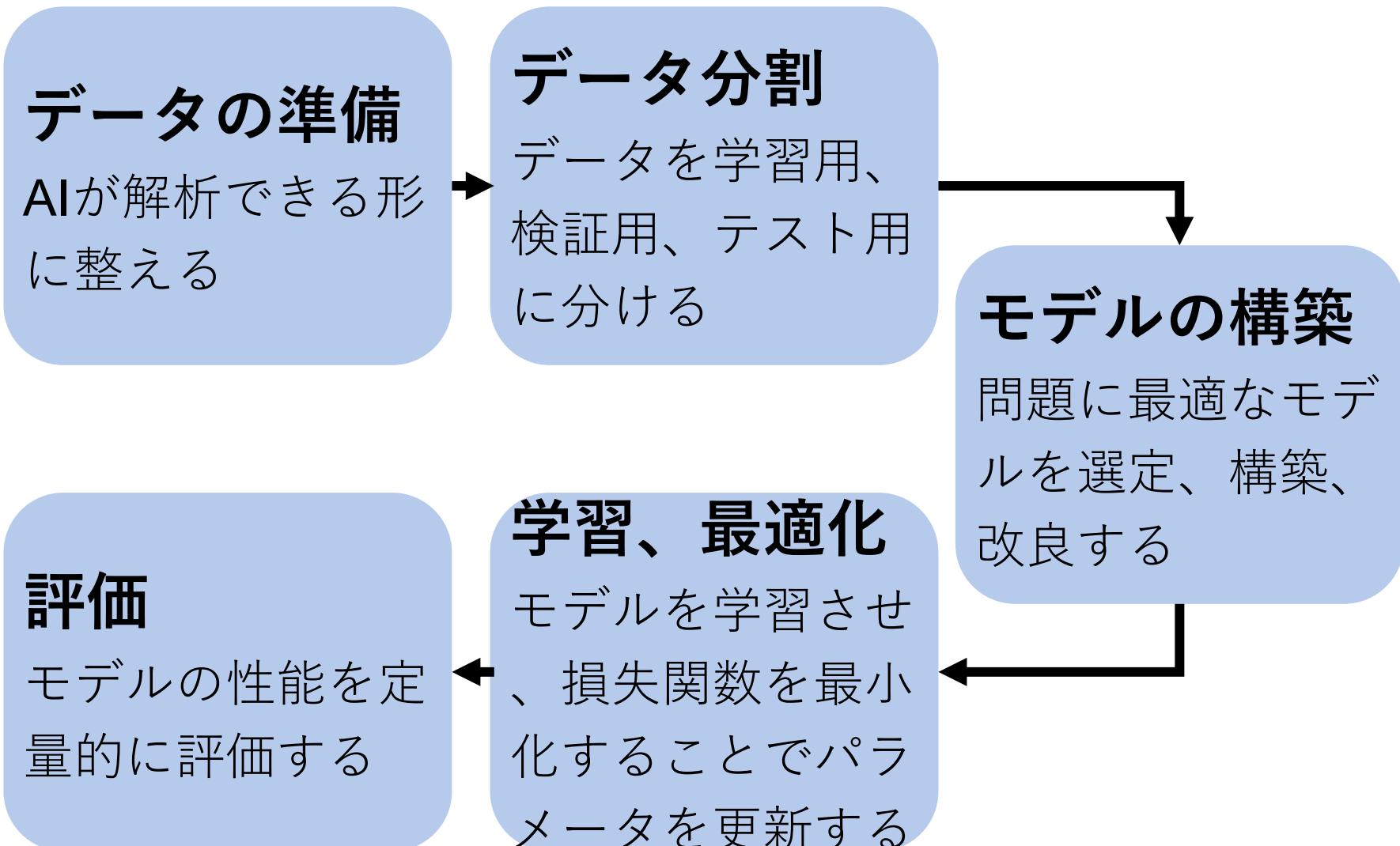
- 自動運転での歩行者検出
- 空港でのスキャン物の特定
- CT・MRI画像から特定の臓器や病変を抽出

生成

- GPT
- 画像生成
- 音楽生成

解析の流れ

基本ワークフロー



プログラムとの比較

データの準備

- ・ データセット～カスタムデータセットクラスの定義

データ分割

- ・ データセットの作成、データローダーの作成

モデルの構築

- ・ モデルの作成、モデルの学習

学習、最適化

- ・ 損失関数とオプティマイザの定義～モデルの保存

評価

- ・ 検証データの確認～

用語の確認

用語の確認

- エポック (epoch)
- バッチサイズ (batch size)
- オプティマイザ (optimizer)

エポック (epoch)

エポック (epoch)

- **定義:** 1エポックとは、 モデルにtrainデータを1周する学習させること
- **詳細な説明:**
 - **学習の流れ:** (trainデータ数が1000の場合)
 - 1エポックの間に全てのデータをモデルに渡し、 その情報をもとにパラメータを更新
 - この学習を何度も繰り返して、 モデルの性能を改善
 - **イメージ:**
 - 1エポックを「1冊の本を読み終える」ようなイメージ
 - 何度も本を読み返すことで、 内容が頭に入ってくる

例:

- **学習回数:** 例えば、 エポック数を10回に設定すると、 1000サンプルのデータが10回学習されることになります
- **注意点:** エポック数が多すぎると過学習（オーバーフィッティング）する可能性があり、 少なすぎると学習が不十分になることがあります

バッチサイズ (batch size)

バッチサイズ (batch size)

- **定義:** バッチサイズとは、一度にモデルに渡すデータ数
- **詳細な説明:**
 - バッチサイズはメモリの制約や学習速度に影響します。大きすぎるとメモリを大量に消費し、小さすぎると学習が遅くなることがあります
 - 一度に大きなデータを渡すことで、学習が安定しますが、あまり大きすぎると計算時間が長くなるので、適切なバランスを取ることが重要です

例:

- **バッチサイズの設定:**
 - 小さいバッチサイズ（例: 16）は学習が不安定になることがあります、メモリに優しい
 - 大きいバッチサイズ（例: 128）は計算が安定しますが、メモリを多く消費

オプティマイザ (optimizer)

オプティマイザ (optimizer)

定義: オプティマイザは、モデルの重み（パラメータ）を更新して損失関数を最小化するためのアルゴリズムです。

詳細な説明:

目的: モデルのパラメータを最適化することで、訓練データに対する損失（誤差）を最小化します。

学習率 (learning rate) : オプティマイザがどれだけモデルのパラメータを更新するかを決定するパラメータです。高すぎると発散し、低すぎると収束が遅くなります。

主なオプティマイザ:

SGD (確率的勾配降下法) : 最も基本的なオプティマイザ。計算が安価でシンプルですが、学習率の調整が難しいことがあります。

計算: 勾配 (derivative) に沿ってパラメータを更新します。

Adam (Adaptive Moment Estimation) : より洗練されたオプティマイザで、学習率の調整が自動で行われるため、一般的に広く使用されます。学習が速く安定しています。

例:

SGD: モデルの重みが勾配の方向に少しずつ調整されます。最初は学習が遅く、後半は速く収束します。

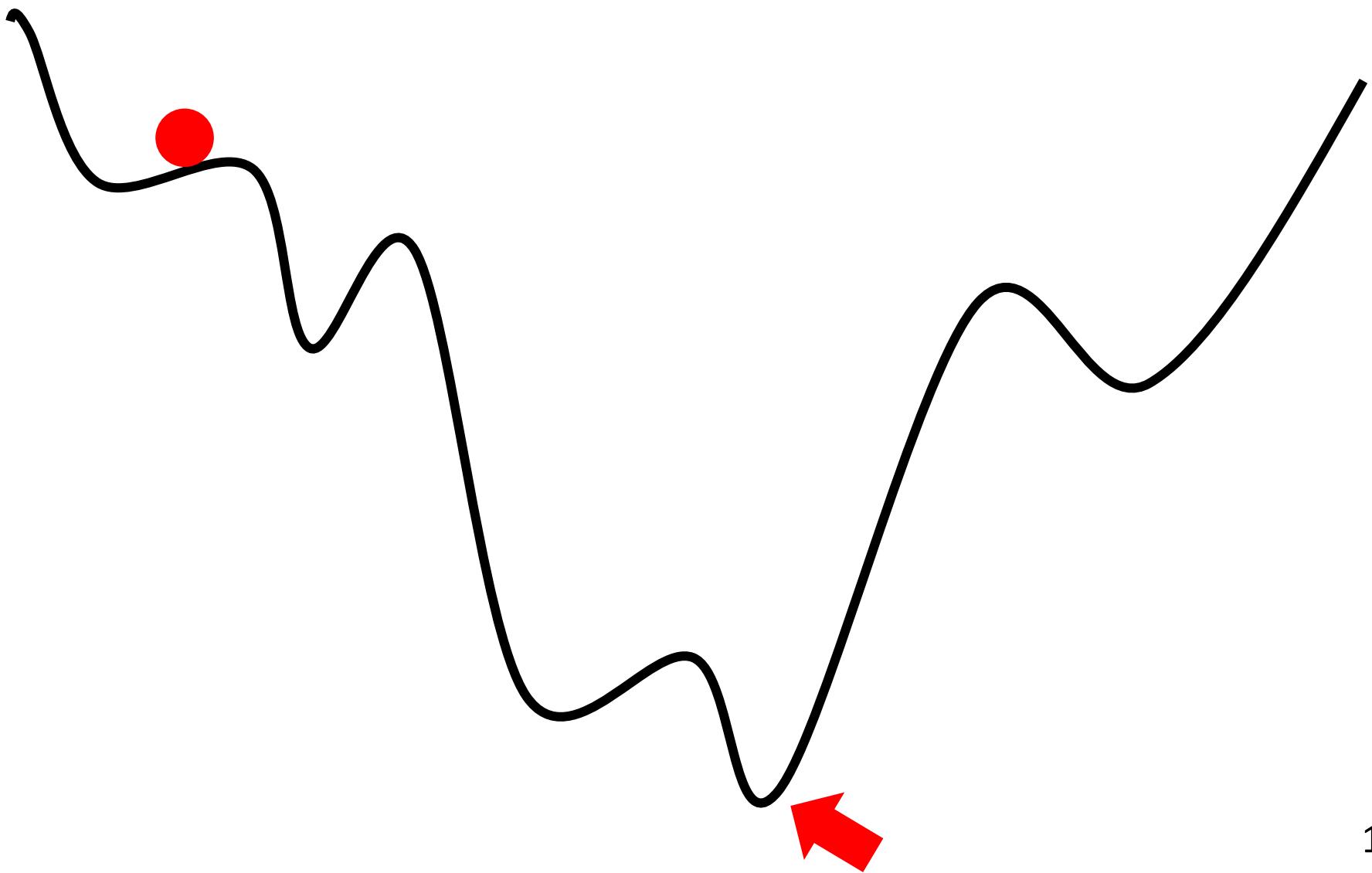
Adam: 勾配の大きさと過去の勾配情報を用いて、学習率を調整するので、学習が早く安定します。

損失関数

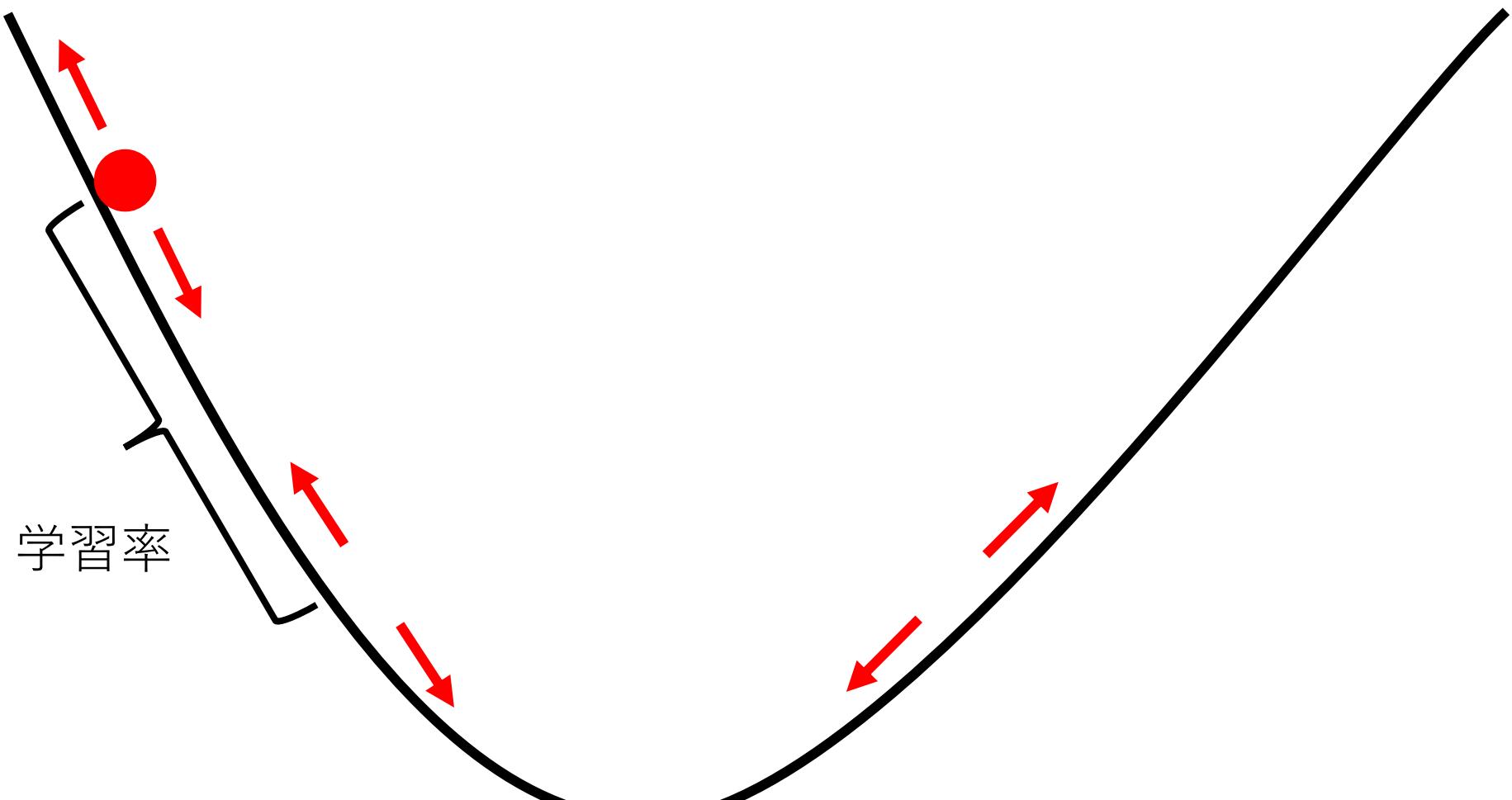
損失関数（Loss）とは

- ・モデルの「間違え具合」を数字で表すもの
- ・値が小さいほど良い (= 正解に近い予測をしている)
- ・学習の目的：**損失関数の値を最小にすること**
- ・学習中に毎ステップ計算され、オプティマイザがこの値を見て重みを更新する

損失関数（Loss）とは



損失関数とオプティマイザ



損失関数とオプティマイザ

登山に例えると

損失関数：山

オプティマイザ：ガイドさん

学習率：登山のスピード

プレ卒での改良点

モデル

- ・ どんなモデルを用いるのか

前処理

- ・ どのようなデータをモデルに入力するのか

損失関数・オプティマイザの調整

- ・ 損失関数やオプティマイザの種類や、学習率の調整

閾値の決定

- ・ 0.5でいいのか