# 深層学習 Part 1

東京海洋大学

竹縄知之

# 目次

# 1　ディープラーニングの概要

　本講義ではディープラーニングで用いられている技術を詳しく説明していきますが，一つ一つがなかなか重いテーマなので，今何を学んでいるのか見失ってしまう懸念があります．そこで，まずはディープラーニングについて概観することにより，全体像をつかんでおきましょう．本講義の前半はこの節の内容をより詳細にみていくことになります．説明なく新しい用語が出てくるかもしれませんが，後ほど改めて説明するのでとりあえず脇へおいて進んでください．

## 1.1　一言でいうと

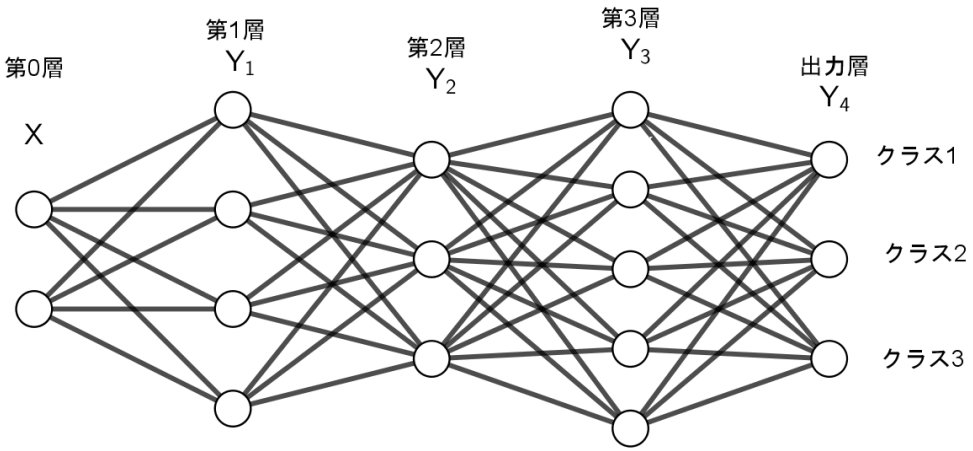
- 沢山の層を持つ複雑なニューラルネットワークによる機械学習
- 画像の解析（映っているものの内容を分析したり，画像から物体を抽出する）や音声や文章の解析（内容を分析したり，翻訳したり）について，近年，人と同程度か上回るレベルで高速に行えるようになり，急速に普及している


Mask R-CNN

## 1.2　ニューラルネットワーク学習 3 つの基本原理

1. 線形変換および活性化関数を繰り返し適用することによる，入力データから予測値の計算（必要な知識：線形代数）
2. 予測値と正解の間の誤差についての，ネットワークの持つパラメータに関する微分（必要な知識：確率・統計・情報理論・多変数の微分）
3. 確率的勾配降下法によるパラメータの更新　（必要な知識：確率・最適化）

入力 ⇒

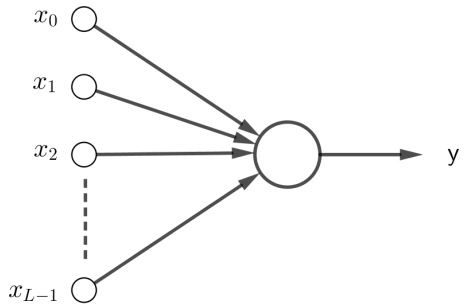



⇒ 出力

# 1．線形変換および活性化関数による予測値の計算

## パーセプトロン

パーセプトロンは複数の信号を入力として受け取り，一つの信号を出力するアルゴリズムです．




上図の左側に縦に並んでいる頂点（ニューロン）から，矢印をたどって右側のニューロンに信号が送られ，一つの信号にまとめられて出力されます．

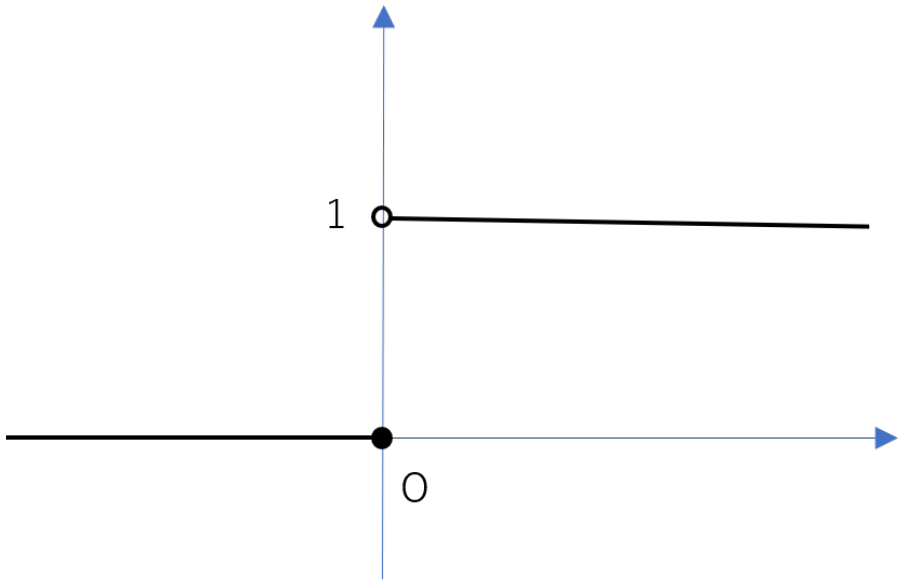
　パーセプトロンは「重み」 $w_0, w_1, \ldots, w_{L-1}$ および 「バイアス」$b$ をパラメータとして持ち，左側のニューロンからの入力 $x_0, x_1, \ldots, x_{L-1}$ に対して，右側のニューロンから

$$a = x_0 w_0 + x_1 w_1 + \cdots + x_{L-1} w_{L-1} + b$$
$$y = h(a) = \begin{cases} 1 & (a > 0) \\ 0 & (a \leq 0) \end{cases}$$

を出力します．$h(x)$ は活性化関数と呼ばれる関数でこの場合はステップ関数です．

## 微分と学習の可能性

パーセプトロンはニューロンとも呼ばれ，ニューラルネットワークは基本的にはパーセプトロンを組み合わせて作ります．しかし，ステップ関数は微分が原点を除いて０になってしまい学習に向かないので，ニューラルネットワークでは活性化関数を微分できるものに変更します．




ステップ関数は $x = 0$ で微分不可能，$x \neq 0$ で微分は $0$　（どちらも学習には困る）

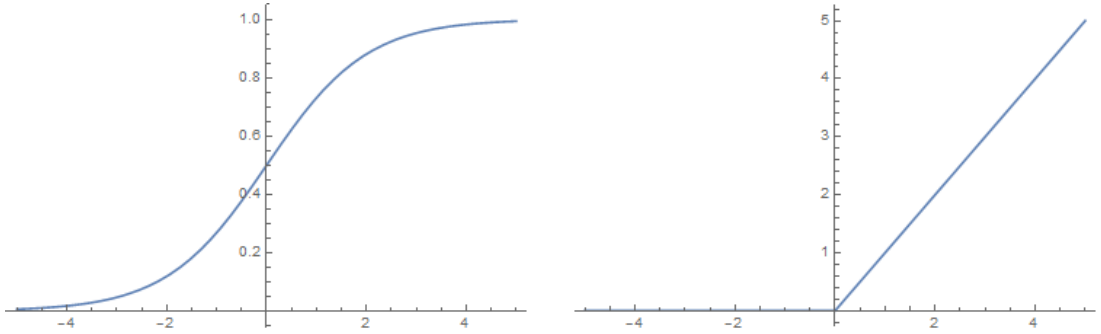
## ニューラルネットワーク

ニューラルネットワークは複数の平行に並んだニューロン（グラフの頂点）からなる層を複数積み上げたネットワークです．第 $i$ 層の $m$ 番目のニューロンにおける演算は，パーセプトロンと同じく，線形変換してから非線形変換を施したもの：

$$\begin{cases} a_m^{(i)} &= x_0^{(i-1)} w_{0m}^{(i)} + x_1^{(i-1)} w_{1m}^{(i)} + \cdots + x_{L-1}^{(i-1)} w_{L-1,m}^{(i)} + b_m^{(i)} \\ x_m^{(i)} &= f\left(a_m^{(i)}\right) \end{cases}$$

です．ここで，$(x_0^{(i-1)}, \cdots, x_{L-1}^{(i-1)})$ は前の層の出力値，$w_{lm}^{(i)}$ および $b_m^{(i)}$ はモデルの重みパラメータと呼ばれる値．また，$f(a)$ は活性化関数と呼ばれる非線形関数であり，以下の 2 つのように微分が 0 でないものを用います．

シグモイド関数 $f(x) = \dfrac{1}{1+e^{-x}}$（左図）　ReLU（Rectified Linear Unit）関数 $f(x) = \max\{0, x\}$（右図）



万能近似定理

「自然数 $d$，$e$ に対して $\mathbb{R}^d$ から $\mathbb{R}^e$ への任意のボレル可測な関数は入力層 $d$ 次元，出力層 $e$ 次元で適当な活性化関数を持つ隠れ層を 1 層のみ持つ順伝播型ニューラルネットワークによって任意の精度で近似できる.」という定理があります [Hornik et al., 1989; Cybenko, 1989; Leshno et al., 1993].

ここで，関数がボレル可測であるという条件は，例えば，$\mathbb{R}^d$ から $\mathbb{R}^e$ への区分的に連続な関数ならば満たされるので，すべての現実的な関数は近似できるといって構いません.

万能近似定理により，（大きな隠れ層を持つ）ニューラルネットワークにより，どんな関数でもで近似できることが保証されていますが．それが学習によって達成できるかどうかは別問題です.

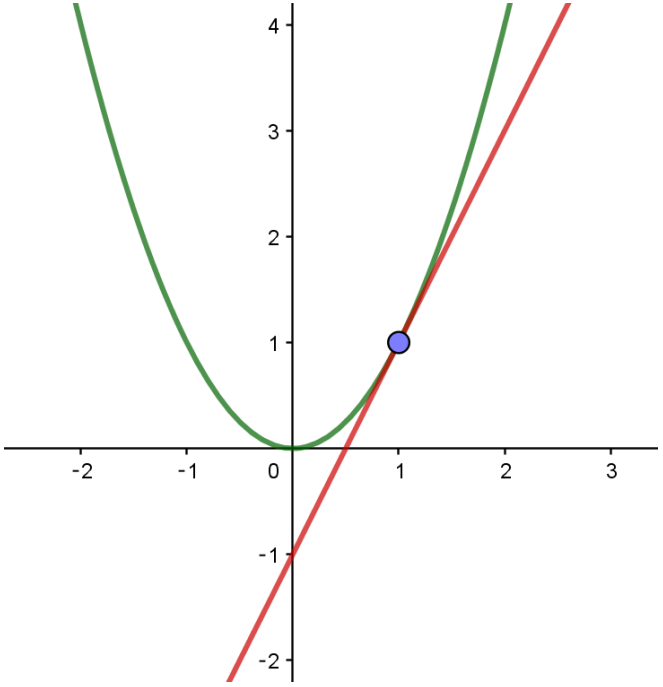
実際，ノーフリーランチ定理[*1]という定理があり [Wolpert, D.H., Macready, W.G., 1995]，全ての問題に対して平均的に汎化性能が他のモデルよりも優れた学習法は存在しないことが示されています.

[*1] ちなみに，ノーフリーランチ定理というのは変わった名称ですが，これは R．A．ハインラインの小説「月は無慈悲な夜の女王」に登場する「無料のランチなんてあるわけない」という，「無料のランチなんてあっても他で代償を払わされている」という意味の格言に由来するそうです.

## 2. 誤差のパラメータによる微分

### 微分

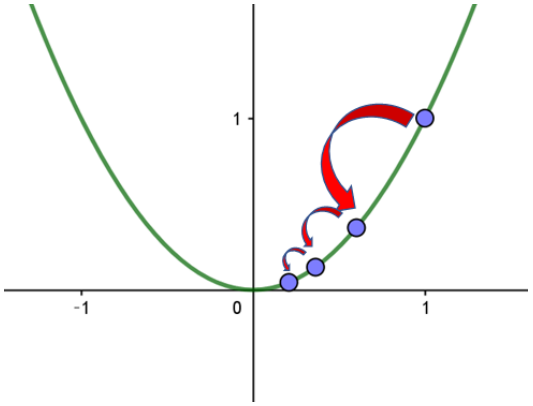
関数の微分とは，関数の入力を少し変化させたときに，出力がその何倍変化するかを表す量です．ニューラルネットワークの場合，重みパラメータを入力，予測値と正解との差から決まる誤差を出力としたときの微分を計算する必要があります．重みパラメータは沢山あるので多変数の微分．層も沢山あるので合成関数の微分になっています．ただし，ディープラーニング用のライブラリ（Tnsorflow, Chainer, Pytorch など）では自動的に計算されます．



## 3. 確率的勾配降下法によるパラメータの更新

### まずは普通の勾配降下法：関数 $f(x)$ の極小値を求める

1. $x$ の値を適当にとる
2. $f'(x)$ を計算（微分はグラフの傾きを表していることに注意）
3. 実定数 $\lambda > 0$ に対して，$x$ を $x \leftarrow x - \lambda f'(x)$ と更新して 2 に戻る




$f(x) = x^2$，$\lambda = 0.2$，$x = 1$ のとき，$x - \lambda f'(x) = 1 - 0.2 \times 2 = 0.6$

このように，関数の値が小さくなるようにグラフの坂を下っていきます．

$\lambda$ が小さい正の数のとき，$f(x + \Delta x) \fallingdotseq f(x) + f'(x)\Delta x$ に $\Delta x = -\lambda f'(x)$ を代入すると，

$f\big(x - \lambda f'(x)\big) \fallingdotseq f(x) - \lambda\big(f'(x)\big)^2 \leq f(x)$ となります．

## 確率的勾配降下法

**確率的勾配降下法（Stochastic Gradient Decent）のアルゴリズム：**

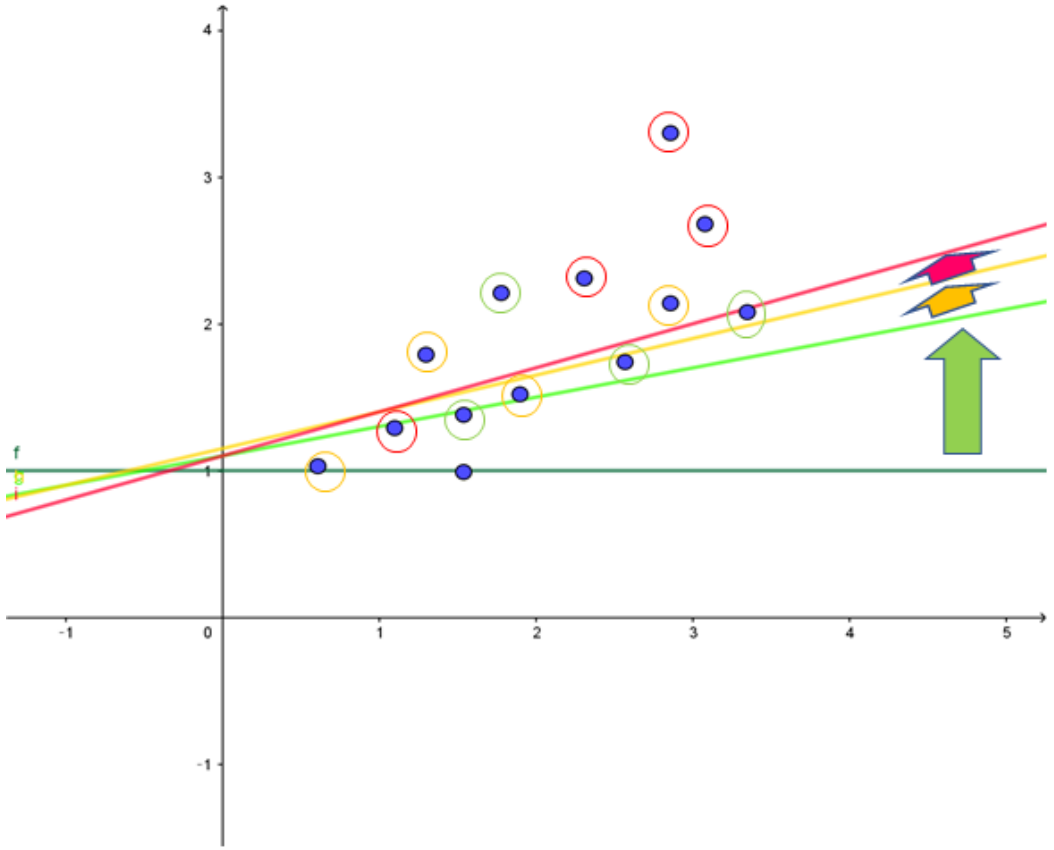
正解付き入力データの集合があるとする．

1. パラメータ $W$ の値を適当にとる
2. 正解付きの入力データの集合から，いくつかのサンプル（ミニバッチ）をランダムに選ぶ
3. 入力データそれぞれに対して予測値を計算する
4. ミニバッチにおける正解との誤差の平均 $E$ のパラメータ $W$ に関する微分 $\dfrac{\partial E}{\partial W}$ を計算
5. 実定数 $\lambda > 0$ に対して，パラメー $W$ を

$$W \leftarrow W - \lambda \frac{\partial E}{\partial W}$$

と更新して 2 に戻る

このようにいくつかのデータをランダムに選んで行う学習法を一般にミニバッチ学習といいます．それに対して全てのデータを一度に用いて行う学習をバッチ学習といい，一度に一つのデータをランダムに選んで行う学習をオンライン学習といいます．




線形回帰問題に対して SGD を用いた場合

いくつかののデータ（図では色のついた丸で囲まれたデータ）を選択して回帰直線を更新して行きます．

## 1.3　深層化へ向けて

### ニューラルネットワークの特徴と課題

- 多くのパラメータによる豊富な表現力を有します．
- 学習の目的は訓練データに対して誤差を小さくすることそのものではなく，未知の入力データに対して正しく予想することです．
- 上の二つの事柄は一般には排反な事象であり，豊富な表現力を持つ予測機を訓練しすぎると，未知のデータに対する予測能力（汎化（はんか）性能）はかえって落ちることが多いです．この状態を過学習といいます．
- SGD は収束は遅いが，過学習を防ぐには有利．入力データをランダムに変形するなどの手法（データ増大）も有効．
- 層が増える（5，6 層以上）と学習がうまく進まなくなり，その対策として，最適化法 SGD の改良，ドロップアウト，バッチ正規化，重みパラメータの初期値の取り方の工夫などがありますが，ネットワークの構造自体を変えた方が効果が大きいです．（ただし，これらはあくまでも実験による結果であって，理論的な説明はほとんどないようです．）

# 畳み込みニューラルネットワーク

- 畳み込みニューラルネットワークは次ページで紹介する畳み込み演算を用いたニューラルネットワークです．
- 通常のニューラルネットワークでは，ある層のニューロンの出力値を計算するのに，前の層の全てのニューロンの出力値を用います．畳み込み演算では，前の層のあるニューロンの「近く」のニューロンの値のみを用いることにより，「近さ」という情報を取り出しやすくします．
- これはディープラーニングにおいて最も基本的な手法であり，ディープラーニングで用いられるネットワークは多くが畳み込みニューラルネットワークおよびその拡張です．

## 畳み込み演算

例えば，ある層への入力 $X$ およびフィルターと呼ばれる重みパラメータ $F$ が

$$X = \begin{array}{|c|c|c|c|}\hline 1 & 2 & 3 & 4 \\ \hline 5 & 6 & 7 & 8 \\ \hline 9 & 10 & 11 & 12 \\ \hline 13 & 14 & 15 & 16 \\ \hline \end{array} \quad F = \begin{array}{|c|c|c|}\hline 1 & 1 & 1 \\ \hline 2 & 0 & 1 \\ \hline -1 & -1 & 0 \\ \hline \end{array}$$

(グレーの部分は以下の $y_{00}$ を計算するときに使うエリア) であるとき，畳み込み演算の出力は

$$Y = \begin{array}{|c|c|}\hline y_{00} & y_{01} \\ \hline y_{10} & y_{11} \\ \hline \end{array} = \begin{array}{|c|c|}\hline 4 & 8 \\ \hline 20 & 24 \\ \hline \end{array}$$

となります．ここで，$y_{00}$ はフィルターと $X$ から左上隅のそれと同じ形の部分を取り出したもの（をそれぞれベクトルとみなしたときの）の内積

$$\begin{aligned} y_{00} =& (1,2,3,5,6,7,9,10,11) \cdot (1,1,1,2,0,1,-1,-1,0) \\ =& 1+2+3+10+7-9-10 = 4 \end{aligned}$$

であり，$y_{01}$ は $X$ からとる部分を右に $1$ ずらしたもの，$y_{10}$ は下に $1$ ずらしたもの，$y_{11}$ は右にも下にも $1$ ずらしたものとなっています．

## 1.4　デモンストレーション動画

ここで　Mask R-CNN というディープラーニングのモデルを用いて作成した動画を見てみましょう．

MaskRCNN 動画.html（index.html から該当するリンクをたどって下さい．）

以上深層学習の概要を大雑把に説明しました．

続いて，Python のライブラリを用いて実際にニューラルネットおよびディープラーニングのモデルを作成し，学習を行ってみます．やったことのある方も多いかもしれませんが，意外と簡単に（少なくとも短いコードで）できてしまうということが分かるかと思います．

本講義では，しばらくはこれらのライブラリの中身がどうなっているのかということについて学んでいきます．

## 1.5 Scikit-learn および Keras を用いた演習

使用するノートブック：

- `1-1_Keras`（と `XGBoost`）で回帰分析
- `1-2_Keras` で手書き数字の分類問題

Scikit-learn は深層学習以外の機械学習用の Python ライブラリであり，Keras は主として Tensorflow 等の深層学習ライブラリをより使いやすくするための高次ライブラリです．

次の点を意識して，Scikit-learn および Keras を用いた演習を行って下さい．

- 回帰問題と分類問題の違い
- 学習用データとテストデータの違い
- どのようなデータをどのように読み込んでいるか
- ボストン住宅価格の予測において，線形回帰，ニューラルネットワーク，決定木ではどの方法の精度が良いか
- 手書き文字認識（MNIST）の認識において，k-近傍法と，畳み込みニューラルネットワークではどちらの方が精度が良いか．学習時間はどちらがかかるか

# 2　機械学習とは

　ディープラーニングはニューラルネットワークの一種でした．そしてニューラルネットワークは（統計的）機械学習の一種です．ここで「(統計的)」という用語はデータを母集団からのサンプリング（標本）と見ていることから来ています．

　本節では機械学習とはそもそもなんであるかということと，関連する用語について整理します．

## 2.1　学習アルゴリズムとは

T. M. Mitchell [Machine Learning, 1997] による定義：

> コンピュータプログラムが，ある種のタスク T（task）と性能指標 P（performance measure）に関して経験 E（experience）から学習するとは，P による評価の下でタスク T の性能を経験 E によって改善することである．

### タスク T

分類，回帰（実数値の予測），転写（文字認識や音声認識のようにあまり構造を持たないデータからテキストデータのような離散データを抽出する），機械翻訳，構造出力（データの特徴を表す木構造を求めたり，画像から物体の場所を特定したりする），異常検知（クレジットカードの不正使用検知など），合成とサンプリング（音声合成や画像生成のように新たなサンプルを作ること），欠損値補完，ノイズ除去，確率密度推定，などがあります．また上記それぞれについて入力に欠損値のある場合もタスクとして挙げられます．

## 性能指標 P

機械学習アルゴリズムの能力を評価するには，その性能を測る定量的な尺度を設計しなければなりません．例えば分類問題ではモデルの精度（accuracy）や交差エントロピー誤差などが，回帰問題では 2 乗誤差がよく用いられるように，性能指標 P はタスク T ごとに適切なものを選ぶ必要があります．また，機械学習では学習において経験していない未知のデータに対してタスクが達成されているかどうかに興味があるので，性能の評価は学習の際には使用していないデータ（テストデータと呼ぶ）に対して行う必要があります．テストデータに対して，学習に使用するデータを訓練データ（training data）または学習データと呼びます．

## 経験 E

機械学習アルゴリズムは大きく教師あり学習と教師なし学習に分けられます．教師あり学習をよく行うタスクとしては分類問題や回帰問題，機械翻訳などがあげられます．教師なし学習をよく行うタスクとしては分類問題や，確率密度推定，ノイズ除去などが挙げられます．

## モデルの能力・過剰適合・過少適合

一般に機械学習においては，(1) 訓練データを用いて学習し，(2) 訓練データに対する誤差 (訓練誤差，training error) を算出し，(3) テストデータに対する誤差（汎化誤差／テスト誤差，generalization/test error) の算出を行います．

普通の最適化法の目的は訓練誤差を小さくすることですが，機械学習の目的は汎化誤差を小さくすることです．これは未知のデータに対する予測誤差を小さくしたいということです．汎化誤差を小さくするには訓練誤差も小さくしなくてはいけないので，結局は両方を小さくすることが機械学習の目的といっても良いでしょう．

学習の結果，訓練誤差があまり小さくなっていない状態を過少適合（underfitting)，訓練誤差は小さくなったがテスト誤差は小さくなっていない状態を過剰適合（overfitting）といいます．

モデルが識別できるデータの種類の数を能力あるいは容量（capacity）といいます．モデルの容量が小さいと訓練をしてもあまり効果がなく過少適合となりますが，容量が大き過ぎると未知のデータに対してはかえって誤差が増えて過剰適合を起こしやすくなります．

## ハイパーパラメータと評価データ

ほとんどの機械学習アルゴリズムはそのアルゴリズムの振る舞いをコントロールするためのパラメータを持っています．これらのパラメータは学習そのものでは変わりません．このようなパラメータをハイパーパラメータと呼びます．

機械学習モデルがその容量を決めるパラメータを持つとき，それらは学習で最適化してはならず，必ずハイパーパラメータとしなければなりません．なぜなら，これらのパラメータを学習してしまうと，訓練誤差を小さくするために容量を増やすように学習してしまい，却って汎化性能が落ちるからです．

よいハイパーパラメータを選ぶには，（訓練誤差ではなく）汎化誤差が小さくなるように選ぶ必要がありますが，そのためにテストデータとは別に評価データ（validation data set）を用意する必要があります．ハイパーパラメータの調整にテストデータを用いてしまうと，もはやそのテストデータはランダムに選ばれたデータとは言えなくなってしまうからです．極端な例ですが，たくさんのハイパーパラメータがある場合，それらをテストデータで調整するとは，結局テストデータでハイパーパラメータを含むモデルを訓練していることになります．

## 2.2　回帰直線を確率的勾配降下法で計算する

学習アルゴリズムの簡単な例として，確率的勾配降下法を用いて回帰直線を計算してみましょう．

データセット $(x, y) = (x_0, y_0), (x_1, y_1), \ldots, (x_{N-1}, y_{N-1})$ に対して回帰直線 $y = ax + b$ を SGD で求めるアルゴリズムは以下のようになります．

1. $a$，$b$ の値を適当にとる
2. データセットから $K$ 個のサンプルをランダムに選び，改めて $(x_0, t_0), (x_1, t_1), \ldots, (x_{K-1}, t_{K-1})$ とおく
3. $y_k = ax_k + b$ で予測値 $y_k$ を計算

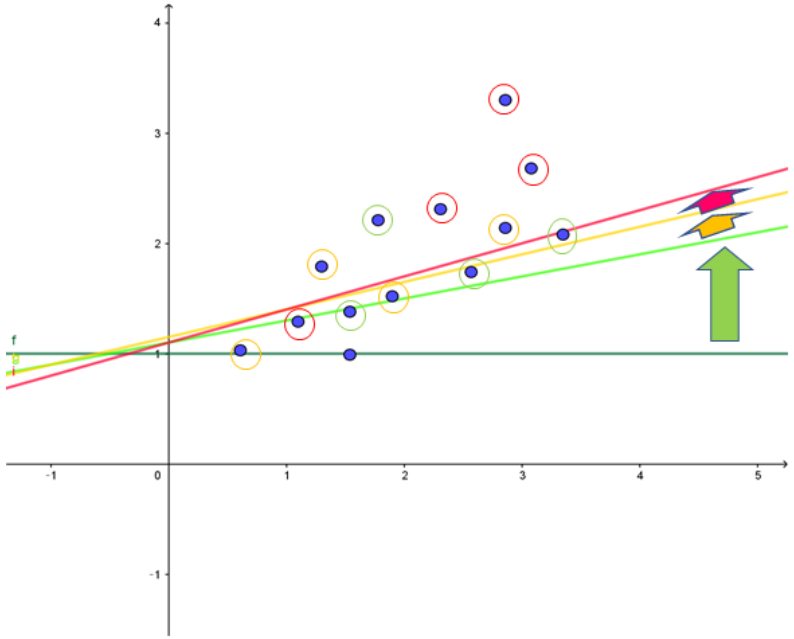
4. **正解との 2 乗誤差** $E = \sum_{k=0}^{K-1} (y_k - t_k)^2$ **のパラメータ** $a$**,** $b$ **に関する微分は**

$$\frac{\partial E}{\partial a} = \frac{\partial}{\partial a} \sum_{k=0}^{K-1} (y_k - t_k)^2 = \sum_{k=0}^{K-1} 2(y_k - t_k) \frac{\partial y_k}{\partial a} = 2 \sum_{k=0}^{K-1} (y_k - t_k) x_k$$

$$\frac{\partial E}{\partial b} = \frac{\partial}{\partial b} \sum_{k=0}^{K-1} (y_k - t_k)^2 = \sum_{k=0}^{K-1} 2(y_k - t_k) \frac{\partial y_k}{\partial b} = 2 \sum_{k=0}^{K-1} (y_k - t_k)$$

5. **実定数** $\lambda > 0$ **に対して，パラメー** $a$**,** $b$ **を以下のように更新して 2 に戻る**

$$a \leftarrow a - \lambda \frac{\partial E}{\partial a}, \quad b \leftarrow b - \lambda \frac{\partial E}{\partial b}$$




アルゴリズムの進行とともに直線が変化して行きます．

注：回帰直線の計算に SGD を用いたのはあくまで例としてであり，回帰直線は通常は，$\dfrac{\partial E}{\partial a} = 0$，$\dfrac{\partial E}{\partial b} = 0$ という条件から求めます．

## 2.3　NumPy による確率的勾配降下法の演習

使用するノートブック：1-3_確率的勾配降下法およびその解答

NumPy は Python 用の数値計算ライブラリで，配列（ベクトルや行列）の計算などが手軽に実行できます．

次の点を意識して，NumPy を用いて線形回帰問題に対する勾配降下法および確率的勾配降下法についての演習を行ってください．

- 勾配降下法と確率的勾配降下法の違い
- ミニバッチ学習
- Numpy における配列の扱い方
- Numpy における乱数（与えられた確率分布に従う標本）の扱い方
- そもそも線形回帰問題に対して確率的勾配降下法を用いるのは適切か

# 3　情報理論

　ニューラルネットワークを含む統計的機械学習は確率論や情報理論に基づいて設計されます．例えば学習するパラメータの初期値は乱数でとるので確率分布の知識が必要になりますし，誤差関数の設計（つまりどのような量を減らせば正しく学習が行えるか）には情報理論の理解が欠かせません．

　本節では，

- 確率論の初歩
- 統計学の初歩
- 情報量
- 情報エントロピー
- 交差エントロピー

について解説します．

## 3.1　確率論と統計学の初歩

**確率分布（離散的な場合）**

根元事象（全体で全事象をなし，排反でそれ以上分割できない事象）が $A_1, A_2, \cdots, A_K$ という $K$ 個からなるとき，各 $A_k$ の起こりやすさを表す値 $P(A_k)$ を確率といいます．ただし，全事象の確率は $1$ とします．

一方，（確率とは別に）各 $A_k$ に実数値 $X = x_k$ が対応するとき，$X$ を確率変数といいます．実数 $x$ に対し，$X = x$ となる根元事象は複数あるかもしれませんが，それらの確率の和を $P(X = x)$ と書き，確率変数 $X$ の確率分布といいます．

例：$A$，$B$，$C$，$D$ の $4$ 枚のカードのうち一枚を適当に選ぶとき，$A$ と $B$ を $1$ 点，$C$ を $2$ 点，$D$ を $5$ 点とし，点数を確率変数 $X$ とするとき，$X$ の確率分布は以下のようになります．

$$P(X = 1) = P(A) + P(B) = \frac{1}{2}$$
$$P(X = 2) = P(C) = \frac{1}{4}$$
$$P(X = 5) = P(D) = \frac{1}{4}$$

**確率分布（連続的な場合）**

**実数に値をとる変数 $X$ に対し，**

$$P(a \leq X \leq b) = \int_a^b p(x)\, dx$$

**（ただし，$P(-\infty \leq X \leq \infty) = 1$）を連続確率分布といい，$p(x)$ を確率密度関数といいます．**

**$X$ に $Y = f(X)$ と関数を施したものも確率変数です．**

**例：区間 $[-10, 10]$ 上の一様確率分布の確率密度関数および確率は**

$$p(x) = \frac{1}{20}, \quad P(a \leq X \leq b) = \frac{b-a}{20}$$

**（ただし，$-10 \leq x \leq 10$，$-10 \leq a \leq b \leq 10$）で与えられますが，$Y = X^2$ に対しては，**

$$\begin{aligned} P_Y(1 \leq Y \leq 25) &= P(-5 \leq X \leq -1) + P(1 \leq X \leq 5) \\ &= \frac{8}{20} = \frac{2}{5} \end{aligned}$$

**となります．**

## 期待値と分散

**確率変数 $X$ に対し，期待値 $\mu = E[X]$ および分散 $V = V[X]$ が以下のように定まります．**

**離散確率分布の場合：$X$ の取りうる値を $x_1, x_2, \cdots$ とするとき，**

$$\begin{aligned}\mu = E[X] &= \sum_i x_i P(x_i) \\ V = V[X] &= E[(X-\mu)^2] = \sum_i (x_i - \mu)^2 P(x_i) \\ &= E[X^2] - \mu^2\end{aligned}$$

**連続確率分布の場合：**

$$\begin{aligned}\mu = E[X] &= \int x p(x) dx \\ V = V[f(X)] &= E[(X-\mu)^2] = \int (x-\mu)^2 p(x)\, dx \\ &= E[X^2] - \mu^2\end{aligned}$$

**また，$\sigma = \sqrt{V}$ を標準偏差といいます．標準偏差は $0$ 以上の実数です．**

## 期待値・分散の基本性質

2 つの確率変数 $X$，$Y$ および実数 $a$，$b$ に対して，

$$E[X+Y] = E[X] + E[Y]$$
$$E[aX+b] = aE[X] + b$$
$$V[aX+b] = a^2 V[X]$$

が成り立ちます．

## 応用例：

期待値 $\mu$，分散 $v = \sigma^2$ の確率変数 $X$ を，期待値 $0$，分散 $1$ になるように正規化するには，

$$Y = \frac{X-\mu}{\sqrt{v}} = \frac{X-\mu}{\sigma}$$

と変換すれば良い．（これはよく使うので覚えておきましょう．）

問．上の応用例の式を示してください．

## ベルヌーイ分布

確率 $p$ で $X = 1$ を，確率 $q = 1 - p$ で $X = 0$ となる離散確率分布をベルヌーイ分布といいます．

$$\text{ベルヌーイ分布：}\ P(X = 1) = p, \qquad P(X = 0) = q = 1 - p$$

ベルヌーイ分布の期待値は

$$E[X] = 1 \cdot P(X = 1) + 0 \cdot P(X = 0) = p$$

であり，分散は

$$\begin{aligned} V[X] &= (1 - p)^2 \cdot P(X = 1) + (0 - p)^2 \cdot P(X = 0) \\ &= (1 - p)^2 p + p^2 (1 - p) \\ &= pq \end{aligned}$$

となります．

- 機械学習においてベルヌーイ分布は 2 値分類問題において現れます．

## カテゴリカル分布

$A_1, A_2, \cdots, A_K$ を根元事象とするとき，$A_k$ が起こる確率が $p_k$ であるとしましょう（ただし，$p_1 + \cdots + p_K = 1$）．これに対し，$k$ 個の確率変数からなるベクトル

$$X = (X_1, X_2, \cdots, X_k)$$

を，$A_k$ が起こるとき $X_k = 1$，起こらないとき $X_k = 0$ と定めると，

$$P(X = (0, \cdots, 0, 1, 0, \cdots, 0)) = p_k \qquad (\text{左辺の } X \text{ は } k \text{ 番目の成分のみ } 1)$$

となり，一つの成分だけに注目すると $P(X_k = 1) = p_k$，$P(X_k = 0) = 1 - p_k$ となります．これをカテゴリカル分布（またはマルチヌーイ分布）といいます．

従って，カテゴリカル分布は $K$ 個のパラメータ

$$(p_1, p_2, \cdots, p_K) \quad \text{ただし，} p_1 + \cdots + p_K = 1$$

で表され，$p_k$ は事象 $A_k$ が起こる確率となります．なお，カテゴリカル分布において $X_k$ の値がどうなるかは $X_l$ の値がどうなるかに影響を与えるので，$X_k$ と $X_l$ は独立ではありません．

- 機械学習における分類問題では $K$ 種類のラベルを上のようなベクトルで表したものをワンホット表現と呼び，モデルの予測はカテゴリカル分布を推定することによって行います．

## 正規分布（ガウス分布）

確率密度関数が

$$p(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

で与えられるとき，連続確率分布 $X$ を正規分布（またはガウス分布）といい，

$$\text{期待値：}\ E[X] = \mu \qquad \text{分散：}\ V[X] = \sigma^2$$

となります．

正規分布をよく，$N(\mu, \sigma^2)$ と表します．

特に $N(0,1)$，つまり，$\mu = 0$，$\sigma^2 = 1$ のとき，標準正規分布といいます．

- 正規分布には相似性があり，$X$ が標準正規分布 $N(0,1)$ に従うとき，$\sigma X + \mu$ の分布は $N(\mu, \sigma^2)$ となります．（これはニューラルネットワークの重みパラメータの初期値などで使います．）

## 3.2　情報理論の基礎

情報量

ある試行において事象 $A$ が起こったときの情報量を，その事象の起こる確率 $P(A)$ を用いて

$$I(A) = -\log P(A)$$

と定めます．対数関数の底は通常 $2$ か自然対数の底 $e$ とします．$0 \leq P(A) \leq 1$ なので情報量は $0$ 以上（または無限大）です．

例えば，事象 $A$ が確率 $2^{-k}$ で起こるとき，底を $2$ とするときの事象 $A$ の情報量は

$$-\log_2 P(A) = -\log_2 2^{-k} = k$$

となります．確率の逆数を $2$ 進数で表すと，

$$2^k \ (10 \text{ 進数}) \ = 10^k \ (2 \text{ 進数})$$

となるので，情報量は確率の逆数を $2$ 進数で表したときの桁数 $-1$ となります．このように情報量は事象の起こりにくさを表すのに必要な記号列の長さを表す量です．

## 情報エントロピー（平均情報量）

カテゴリカル分布と同様に，全事象が排反な $K$ 個の事象 $A_1, A_2, \cdots, A_K$ からなるとき，

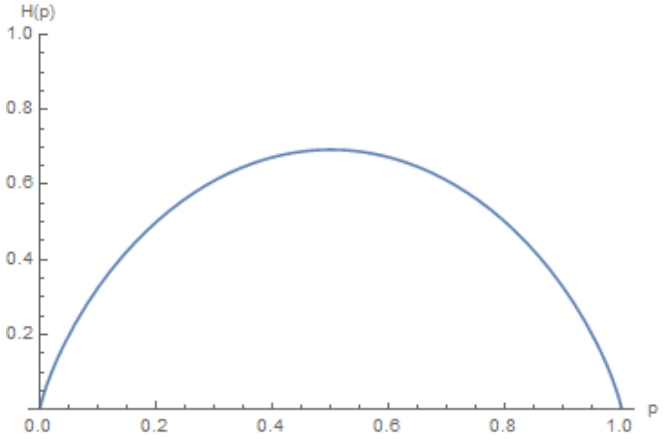
$$H(P) = -\sum_{k=1}^{K} P(A_k) \log P(A_k)$$

を情報エントロピーといいます．ただし，$P(A_k) = 0$ のときは $P(A_k) \log P(A_k) = 0$ とみなします．情報エントロピーは $A_k$ の情報量 $I(A_k)$ を確率変数と見たときの期待値といえます．

例えば，ベルヌーイ分布の情報エントロピーは

$$\begin{aligned} H(P) &= -P(X=1) \log P(X=1) - P(X=0) \log P(X=0) \\ &= -p \log p - (1-p) \log (1-p) \end{aligned}$$

となります．




ベルヌーイ分布の情報エントロピー

横軸 $p$，縦軸 $H(P)$

情報量は確率が小さい事象ほど大きいのですが，その期待値である情報エントロピーは，確率が均等なとき（つまり予測が難しいとき）の方が大きくなります．

## 交差エントロピー

全事象が排反な $K$ 個の事象 $A_1, A_2, \cdots, A_K$ からなり，2 つの確率分布 $P(A_k)$ と $Q(A_k)$ （$k = 1, 2, \cdots, K$）が与えられているとします．このとき，

$$H(P, Q) = -\sum_{k=1}^{K} P(A_k) \log Q(A_k)$$

を（$P$ に対する $Q$ の）交差エントロピーといいます．

交差エントロピーは確率分布 $P$ に関する確率分布 $Q$ についての情報量の期待値ということができます．交差エントロピーは $P$，$Q$ に対して対称ではないことに注意しましょう．

一般に，$P$ を固定して $Q$ に関して交差エントロピー誤差を最小化すると，$Q = P$ のとき最小値として $H(P, Q) = H(P)$ をとることが示せます．

- 機械学習では交差エントロピーは $K$ 種類への分類問題における誤差関数として非常に身近なものです．このときは，$P$ を正解値のワンホット表現から定まるカテゴリカル分布，$Q$ を予測のカテゴリカル分布とします．

## 交差エントロピーの意味

確率分布 $P$ を固定して確率分布 $Q$ を変化させるとき，交差エントロピー $H(P,Q)$ は $P=Q$ のとき最小となります．つまり，交差エントロピーは確率分布 $P$ と $Q$ がどれくらい異なるかという尺度です．

例えば $P(X=1)=p$，$Q(X=1)=q$ という 2 つのベルヌーイ分布 $P$，$Q$ に対する交差エントロピーは

$$\begin{aligned} H(P,Q) &= -P(X=1)\log Q(X=1) - P(X=0)\log Q(X=0) \\ &= -p\log q - (1-p)\log(1-q) \end{aligned}$$

となります．簡単のため $p=\dfrac{1}{2}$ とすると，

$$H(P,Q) = -\frac{1}{2}\log q(1-q) = -\frac{1}{2}\log\left(-\left(q-\frac{1}{2}\right)^2+\frac{1}{4}\right)$$

となります．よって $q=\dfrac{1}{2}$ のとき，交差エントロピーは最小値 $\dfrac{1}{2}\log 4$ をとり，$q$ がそこから離れるとともに増大します．