



DXに向けた、デジタル技術導入講座



# ノンコードニューラル ネットワーク (AI) 技術

デジタル技術のスキルアップで仕事の幅を広げよう  
理系・文系を問わないデジタル人材の育成



四国職業能力開発大学校  
生産電子情報システム技術科  
玉井瑞又

# 本教材のねらい

近年、DXが加速する中、デジタル技術の重要性はますます高まっています。しかし、プログラミングスキルがない社員も、AIやデータ分析といった高度な技術を活用できる時代になりました。本教材では、理系・文系を問わず、どなたでもAIの基礎知識とスキルを習得することができます。

対象者

AIに興味がある方

AIを仕事に活かしたい方

理系・文系を問わず利用可能

得られるスキル

AIの基礎知識

具体的なAIの使用方法

AIツールの使い方

処理するデータは、エクセルを用いて作成しています。

後ほど、自分で用意したデータを使って処理を行うことができます。

はじめに デジタル技術について

事前情報

AIとニューラルネットワークとは

ツールの入手とインストール

サンプルを実行

サンプルの説明とデータのダウンロード

ニューラルネットワークの部品

ニューラルネットワークの作り方

2クラス、多クラス、回帰について

ニューラルネットワークでは何が行われているか

画像データの使い方

データセットの入手

# はじめに デジタル技術について

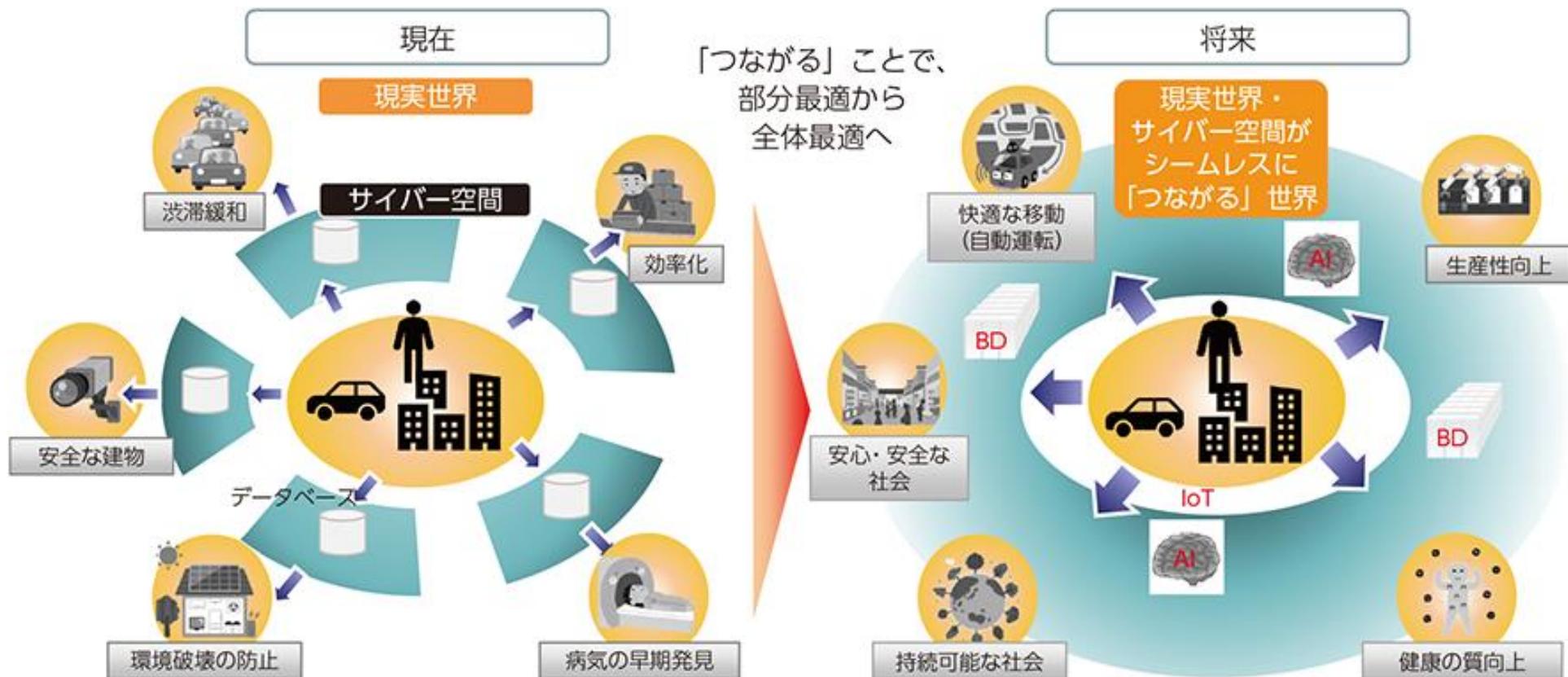
AI技術は、データの分析や予測、自動化など、さまざまなビジネスシーンで活用できるデジタル技術です。近年、AI技術の進化が著しく、ビジネスにおける活用も急速に拡大しています。例えば、製造業では、AI技術を活用して、生産ラインの検査や故障の発見など、効率化を図ることができます。AI技術は、さまざまな業界で、業務効率化や生産性の向上、新たなビジネスの創出など、さまざまなメリットをもたらす可能性を秘めています。

## デジタル・トランスフォーメーション

2004年 ウメオ大学（スウェーデン）のエリック・ストルターマン教授が提唱。

「ITの浸透が、人々の生活をあらゆる面でより**良い方向に変化**させる」

(令和3年情報通信白書より)



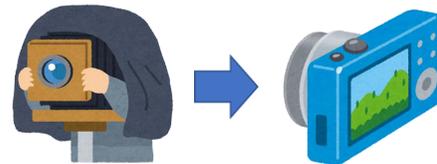
## DX導入のステップ

(中小企業庁・中小企業総合支援サイトより)



### STEP1 アナログだったものをデジタルにする

たとえば、紙だった帳簿を会計ソフトにする、お金のやりとりをバーコード決済にする、商談をオンラインにする、店舗に行く買い物をネットショップにするなど、いままでいままでもアナログだったものをデジタルにします。



アナログからデジタル



### STEP2 生産効率・業務効率が向上し、デジタルデータが蓄積される

デジタル化により業務効率が高まり、生産性が上がります。それとともにノウハウとデジタルデータが蓄積されていきます。実は、このデジタルデータが宝の山なのです。



蓄積されるデータ



### STEP3 デジタル・トランスフォーメーション（ビジネス・組織を変える）

宝の山であるデジタルデータをビジネスに活用します。たとえば、会計データを原価管理や顧客管理に活用したり、バーコード決済データを商品仕入・販促に活用したりします。これを業務プロセスやビジネスモデルに組み込めば、「DX」と言えるのではないのでしょうか。



新しい企業・ビジネス

DXに、一気にたどり着くことはできない！

デジタイゼーション



デジタルは、確立された産業の効率化などを補助するツール

## (1) Digitization (デジタイゼーション)

会社内の特定のアナログ的な工程を効率化するためにデジタルツールを導入する。既存の紙のプロセスを自動化する。物質的な情報をデジタル形式に変換する。

「デジタルライゼーション」を目標としたときの手段

デジタルライゼーション



デジタルは、産業と一体化することで、ビジネスモデル自体を変革する

## (2) Digitalization (デジタルライゼーション)

自社内だけでなく外部環境やビジネス戦略も含めたプロセス全体をデジタル化。組織のビジネスモデル全体を一新する。クライアントにサービスを提供するより良い方法を構築する。

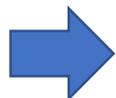
「デジタルトランスフォーメーション」を目標としたときの手段

(令和3年情報通信白書より)



Digitization

(デジタイゼーション)



Digitalization

(デジタルライゼーション)



Digital Transformation

(デジタルトランスフォーメーション：DX)



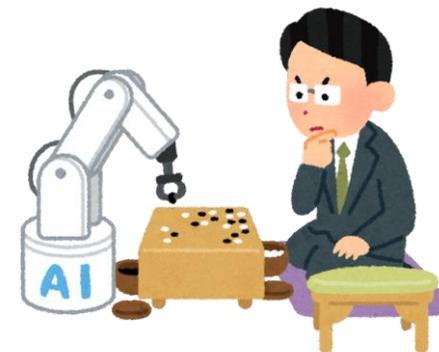
多くの資料に共通しているのは、「**DX推進人材の不足**」

DXの推進には、DX人材が必要。

DXのデジタルイゼーション、デジタルライゼーションには、**ICT人材**が必要

**ICT人材**とは、デジタル人材

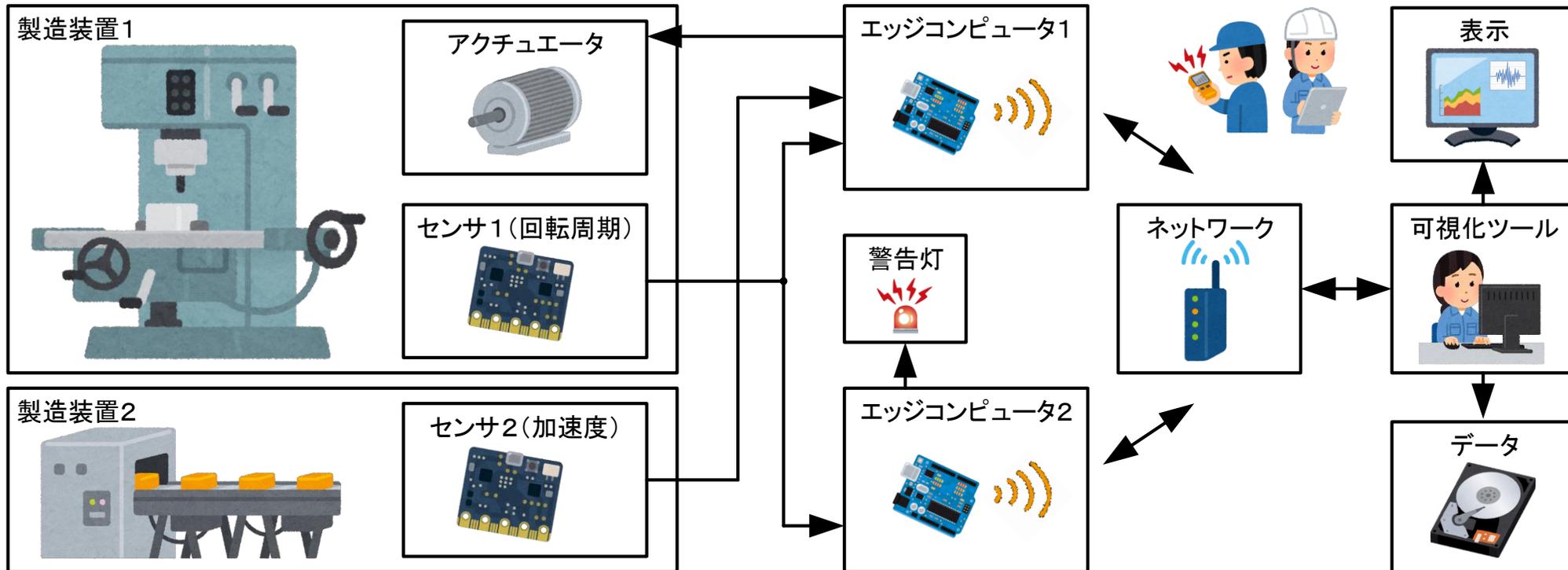
ICT → Information and Communication Technology（情報通信技術）



**DX人材** → **ICT人材** + **経営的要素**

**ICT人材** → **技術要素**としては、

**IoT**、**ビッグデータ**、**データサイエンス**、**AI**、**ロボティクス**など



工場には複数の製造装置がある。  
 複数のセンサが搭載されている。  
 Wifiにより情報収集される。

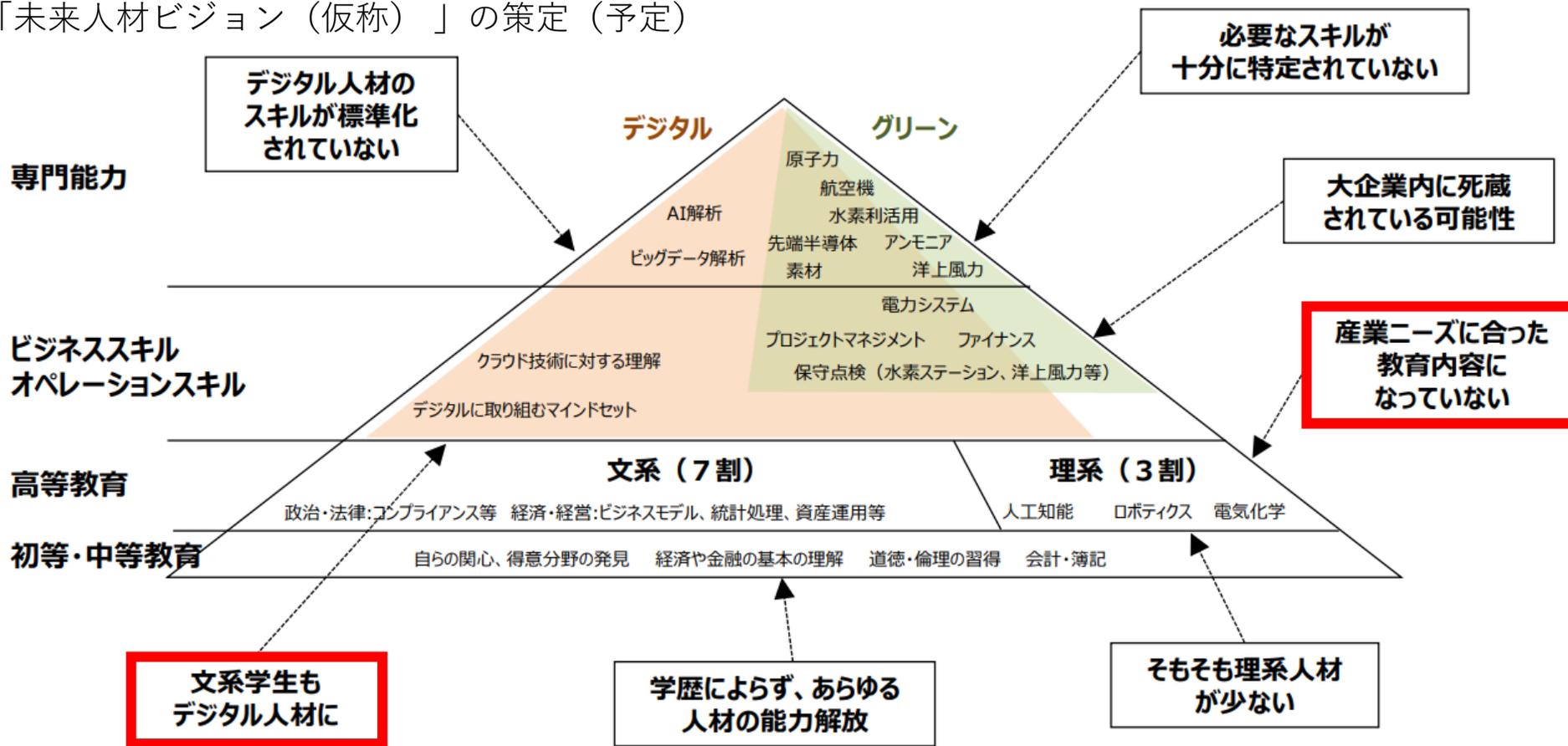
工場の情報はディスク装置に保存される。

製造装置にはアクチュエータが搭載されている。  
 複数のエッジコンピュータを使用する。

可視化ツールにより工場・製造装置の状態を監視できる。

保存されたデータをAIで分析する。

「未来人材ビジョン（仮称）」の策定（予定）



**理系・文系を問わず、ICT人材（デジタル人材）の育成が必要**  
**ICT人材育成の課題は、産業ニーズに近い課題が必要**

(第27回中央訓練協議会 資料7 経済産業省における人材育成施策より)

プレス加工メーカー（自動車関連部品）

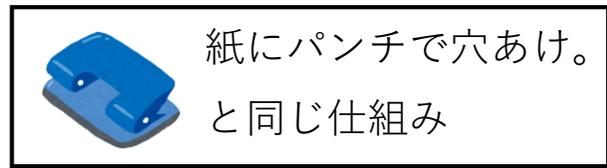
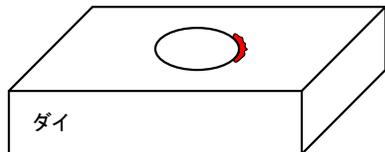
パンチとダイを使って材料を打ち抜く



パンチやダイが欠ける  
(チッピング)

→ 不良品の生産

→ 交換・調整のため生産停止



## 従業員の話を聞くと

- 金型を交換して3日目が要注意  な気がする
- 雨が降っているときが多い  気がする
- 水曜日にチッピングする  気がする
- 〇〇さんが交換すると怪しい  気がする

**いろいろな話は聞けるが原因は不明**

## 対策

- (1) 装置の振動を加速度センサで計測し  
ニューラルネットワークで分析  
→ 製造装置の状態をデジタルデータで収集  
**デジタル化、AI**
- (2) 従業員の能力開発（セミナー）  
→ 暗黙知を定量的にアウトプット  
できるように、センサや計測に関する技術  
**IoTを構成するための技術要素の学習**

## ★実は、平成10年ころの事例

当時、コンピュータやネットワークの性能が低く、十分に目的を達成できなかった。今だと、ICTを使った様々な対策が考えられる

## プレス機のメンテナンス企業



プレス機にセンサを取り付けたい。  
機械の状態を監視したい。



全国のお客さんの情報を収集したい。



それによって



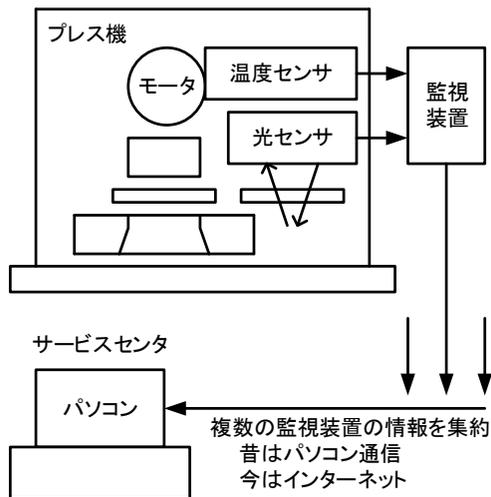
故障しそうなプレス機を事前に発見  
修理計画の提案をおこなう

当時の反応

センサが壊れるかも？ 通信回線は大丈夫？  
通信料は？ 使い人いるの？

しのはらプレスサービス株式会社 <http://www.shinohara-press.co.jp/index.html>

今は、



(日刊工業新聞2017/7/12)

常時監視することで機械の状態を→見える化

機械の故障前に対応が可能となり、生産停止期間を大幅に短縮

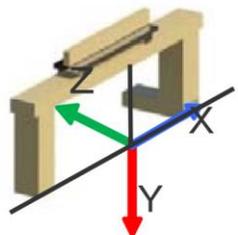
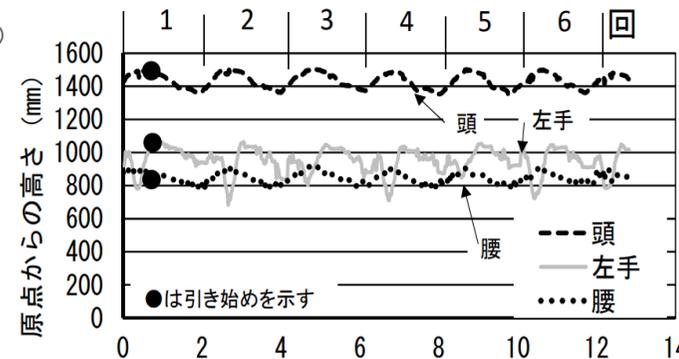
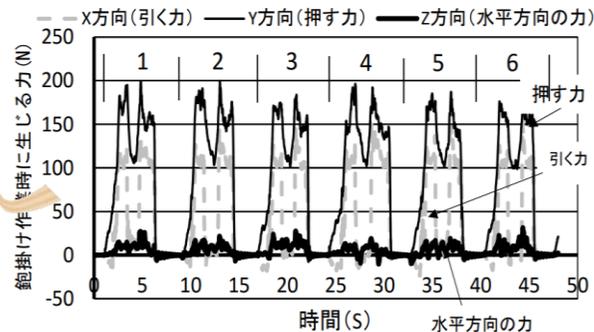
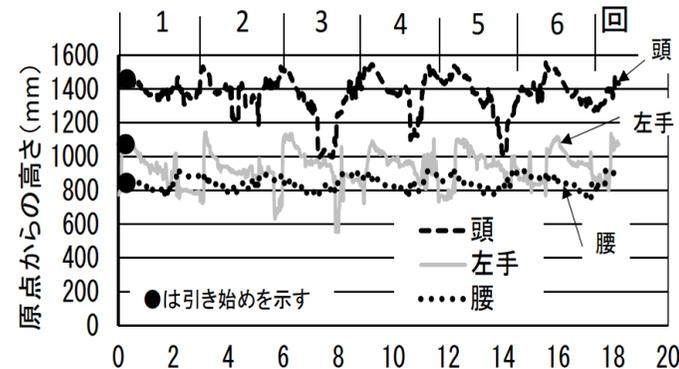
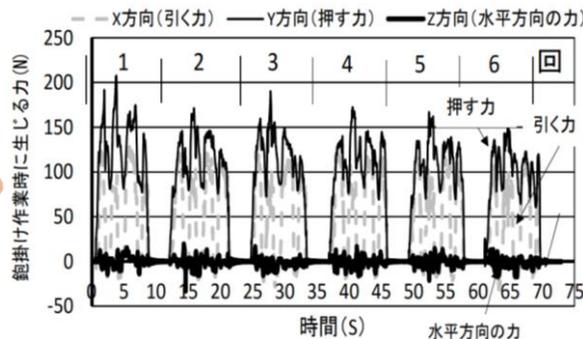
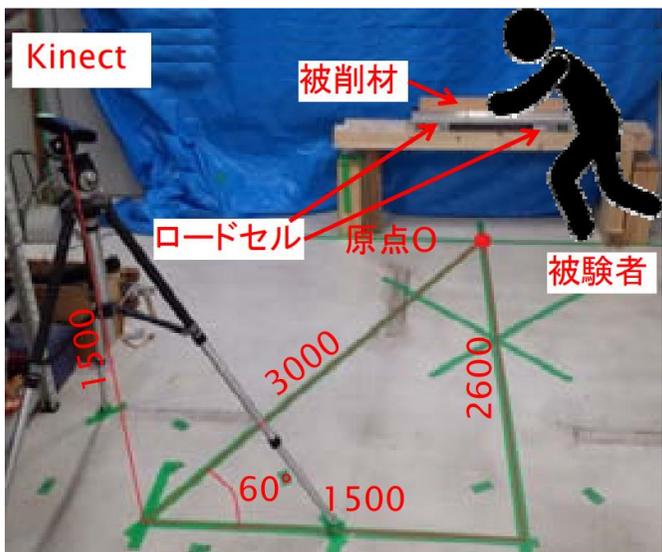
故障前に修理計画を立てることが可能

予防保全をIoTで加速

平成13年のお話し

## 大工技能の研究例

### 鉋作業の力の計測 + 姿勢の計測



X方向：引く力  
Y方向：押す力  
Z方向：水平方向の力

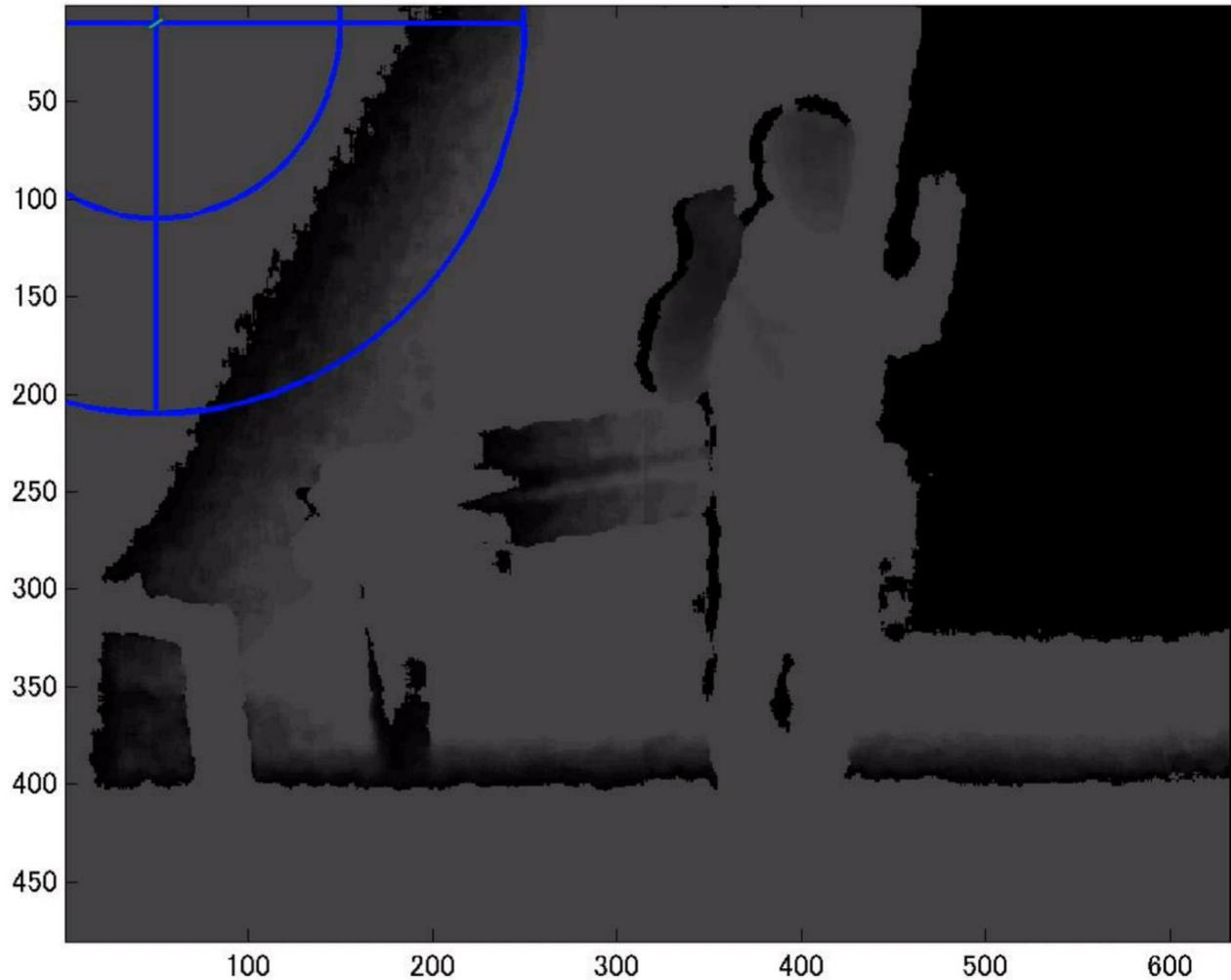
## 技能伝承

見て覚えるだけでは難しい？

的確なアドバイスが必要

日本建築学会大会学術講演会

- 2018,大工技能の動作解析と指導方法に関する研究-のこ挽き作業について-
- 2018,大工技能の動作解析に関する研究-のみと鉋刃の刃研ぎ作業について-
- 2017,大工技能の動作解析に関する研究-刃研ぎ作業について-
- 2016,大工技能の動作解析に関する研究-鉋掛け作業について-
- 2015,大工技能の動作解析に関する研究

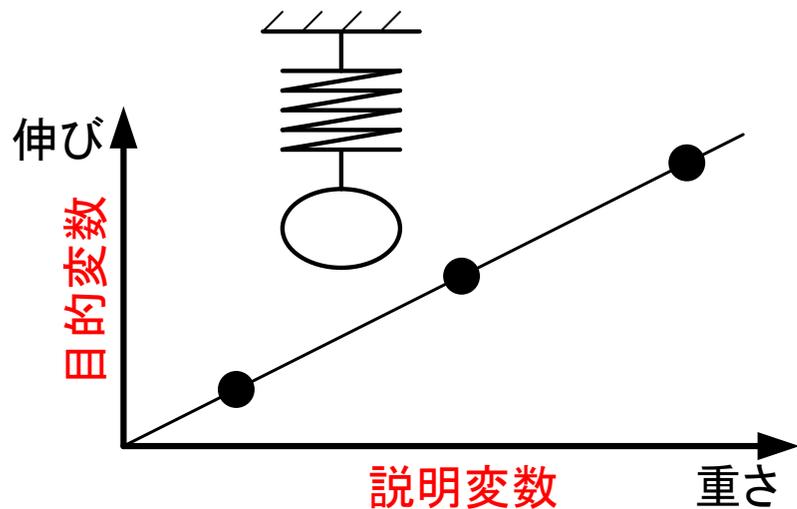


科学研究費助成事業 26350221 2014-2016 職業大  
建築大工技能の科学的手法にもとづいた指導法に関する研究

鉋がけ作業の分析  
作業風景  
奥行画像  
骨格画像  
ロードセルの値  
を同時に表示

本セミナーでは、ニューラルネットワークを作成できるツールを使用します。その中で出てくる用語や、確率統計に関する解説を行います。AIは、何かを判断するのではなく、あることの確からしさを確率で示します。例えば、犬の写真を見て、「犬 85%、猫 15%」という結果を出すような感じです。

バネに重りを付け、バネの伸びのグラフを作成



何かの原因となっている変数

説明変数→重さ

原因を受けて発生した結果となっている変数

目的変数→伸び

機械学習（教師あり学習）では、

**説明変数**→予測するために使用するデータ

**目的変数**→予測したいデータ

例 不動産物件の

駅からの距離、間取り、築年数から、  
家賃を推定したい

**説明変数**である、

駅からの距離、間取り、築年数のデータを収集

↓

それらを用いて、

↓

**目的変数**である、

家賃を推定する

確率→ある事象が起こる、確からしさの割合

下駄を投げた→表 50%、裏 50%

サイコロを投げた→1の目1/6、2の目1/6...

確率変数→ある変数の値をとる確率が存在する変数

サイコロを投げて出る目 {1, 2, 3, 4, 5, 6}

サイコロを投げて出る目は確率変数

確率変数の値 = サイコロの出る目

$$P(X) = \frac{1}{6} \quad (X = 1, 2, 3, 4, 5, 6)$$

サイコロを投げて3の目が出る事象の確率

$$P(X = 3) = \frac{1}{6}$$

条件付き確率

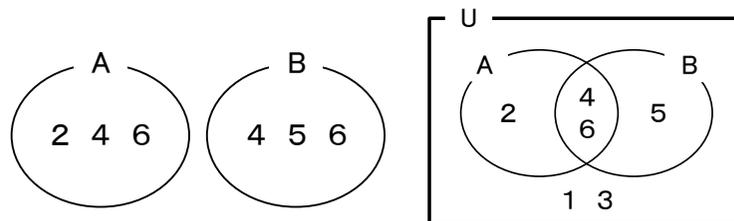
事象Aが起きたと分かったもとで事象Bが起こる確率

$$P(B|A) = \frac{P(A \cap B)}{P(A)}$$

サイコロを投げた。出目は偶数だった。このとき出目が4以上である確率は？

A : 出目が偶数 (出目が、2, 4, 6)

B : 出目が4以上 (出目が、4, 5, 6)



Aが起きるのは、 $3 / 6 = 1 / 2$

Aが起きてBが起きる確率は、 $2 / 6$

$$P(B|A) = \frac{P(A \cap B)}{P(A)} = \frac{\frac{2}{6}}{\frac{1}{2}} = \frac{2}{3}$$

イギリスの数学者・牧師であるトーマス・ベイズ (Thomas Bayes) によって示され、ある事象Aが起こる条件下で、別の事象Bの起こる確率を求めることができる

条件付き確率

事象Aが起きたと分かったもとで事象Bが起こる確率

$$P(B_i|A) = \frac{P(A \cap B_i)}{P(A)}$$

乗法定理

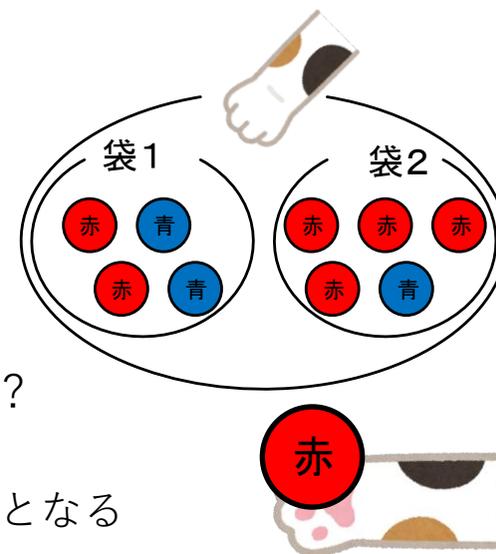
$$P(A \cap B_i) = P(B_i)P(A|B_i)$$

を代入

ベイズの定理が求まる

$$P(B_i|A) = \frac{P(B_i)P(A|B_i)}{P(A)}$$

例 ベイズの定例  
 大きな袋の中に、  
 外から分からない  
 小さな袋が2つ入っている  
 大きな袋に手を入れ、  
 ボールを1個取り出した。  
 赤のボールを取り出した。  
 袋1から取り出した確率は？



ベイズの定理の式の  $B_i \rightarrow$  袋 $_i$   $A \rightarrow$  赤 となる

$$P(\text{袋1}|\text{赤}) = \frac{P(\text{袋1})P(\text{赤}|\text{袋1})}{P(\text{赤})} = \frac{P(\text{袋1})P(\text{赤}|\text{袋1})}{P(\text{袋1})P(\text{赤}|\text{袋1}) + P(\text{袋2})P(\text{赤}|\text{袋2})}$$

$$P(\text{袋1}) = \frac{1}{2} \quad P(\text{赤}|\text{袋1}) = \frac{2}{4} = \frac{1}{2} \quad P(\text{袋2}) = \frac{1}{2} \quad P(\text{赤}|\text{袋2}) = \frac{4}{5}$$

$$P(\text{袋1}|\text{赤}) = \frac{\frac{1}{2} \cdot \frac{1}{2}}{\frac{1}{2} \cdot \frac{1}{2} + \frac{1}{2} \cdot \frac{4}{5}} = \frac{\frac{1}{4}}{\frac{1}{4} + \frac{4}{10}} = \frac{10}{10 + 16} = \frac{10}{26} = 0.385$$

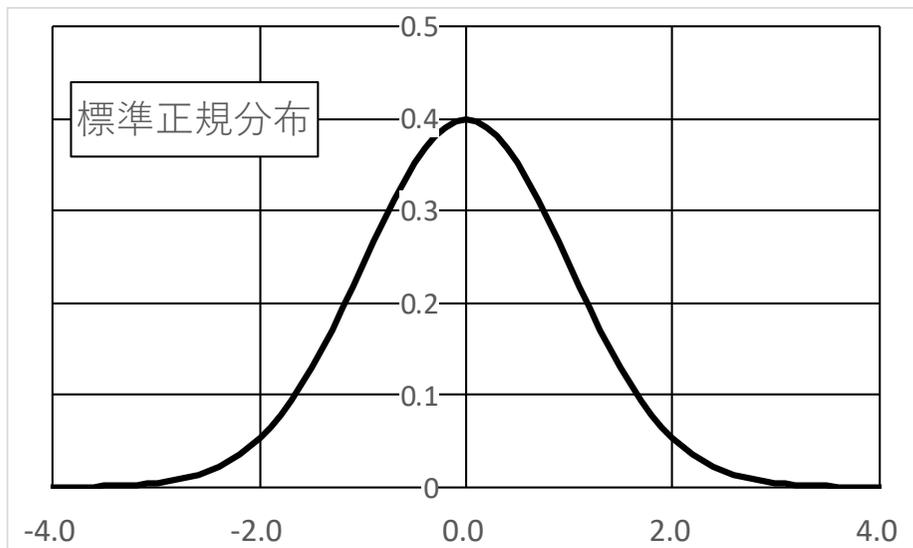
ガウス分布とも呼ばれる。

平均値 $\mu$ と分散値 $\sigma^2$ によってあらわされる。

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

標準正規分布 (平均値 $\mu = 0$ 、分散値 $\sigma^2 = 1$ )

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

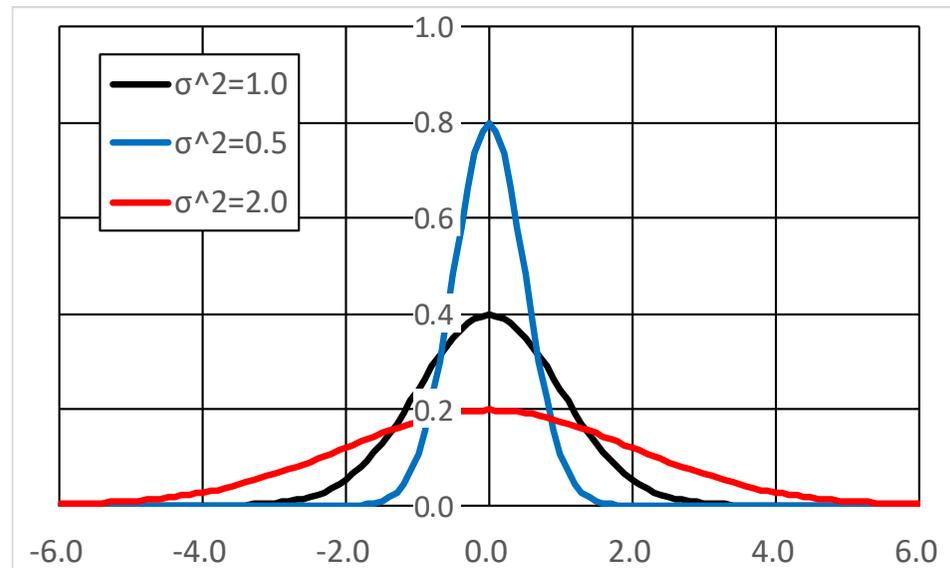


特徴

$$\int_{-\infty}^{+\infty} f(x) dx = \int_{-\infty}^{+\infty} \left[ \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{(x-\mu)^2}{2\sigma^2}\right\} \right] dx = 1.0$$

面積が 1

$\pm 1.0 \times \sigma$	$\rightarrow$	68.27%
$\pm 2.0 \times \sigma$	$\rightarrow$	95.45%
$\pm 3.0 \times \sigma$	$\rightarrow$	99.73%



# AIとニューラルネットワークとは

AIとニューラルネットワークは、イコールではありません。AIにはさまざまな技術があり、ニューラルネットワークは、その1つです。ニューラルネットワークがどのようなことをしているのかを、概要を見てみましょう。

## 初めの用語

人工知能 AI→Artificial Intelligence

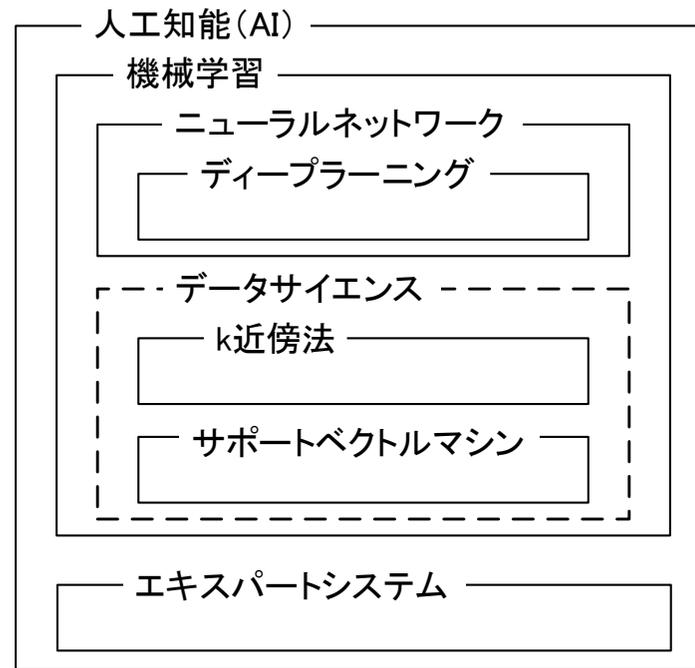
(アーティフィシャル インテリジェンス)

人間が行う「知的活動」をコンピュータプログラムとして実現すること。知的活動とは、頭（厳密には脳）で考えて実行する活動全般のこと。例えば「絵を描く」「言葉を認識する」「ゲームをする」などなど、あらゆる人間の行動がこれに当てはまる。

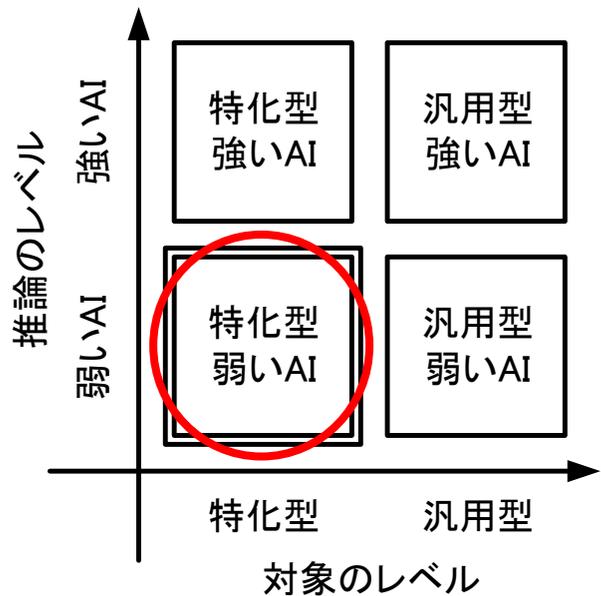
機械学習 ML→Machine Learning

AI（人工知能）を実現するための手法の一つ。人間の学習に相当する仕組みをコンピューター等で実現するものであり、一定の計算方法（アルゴリズム）に基づき、入力されたデータからコンピューターがパターンやルールを発見し、そのパターンやルールを新たなデータに当てはめることで、その新たなデータに関する識別や予測等を可能とする手法のこと。

## AIからディープラーニングなどの位置づけ



- (1)あるデータの課題
- (2)データサイエンスでデータの特徴を分析  
複雑な問題であれば
- (3)ニューラルネットワークで分析してみる



対象のレベル

現在のAIは



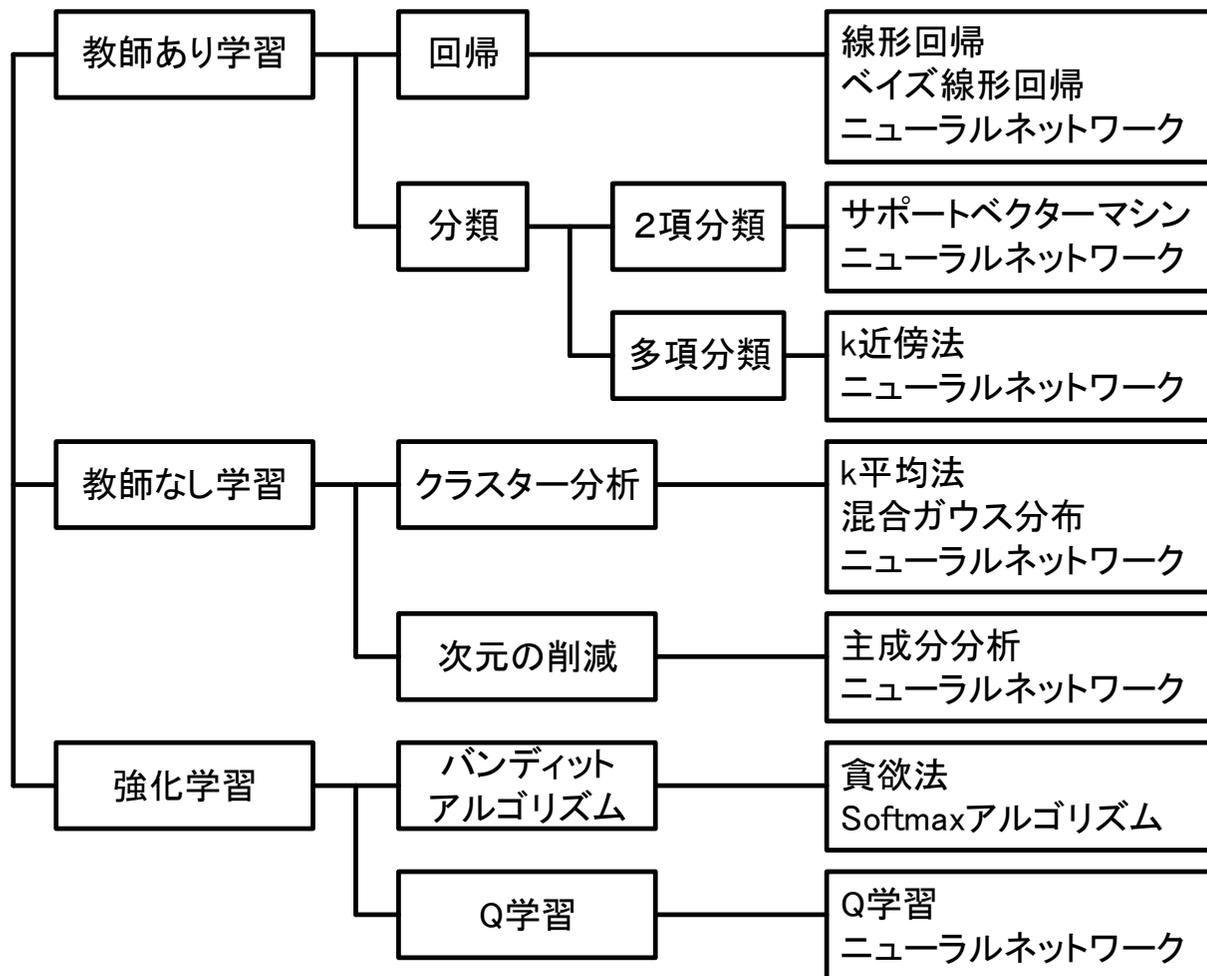
特化型：特定のタスクだけに対応

汎用型：いろいろなタスクに対応

推論のレベル

弱いAI：プログラムに従い問題解決

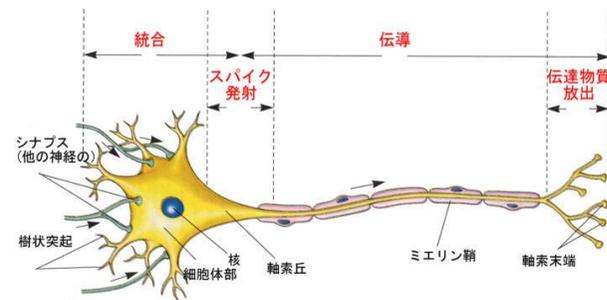
強いAI：自律的に問題解決



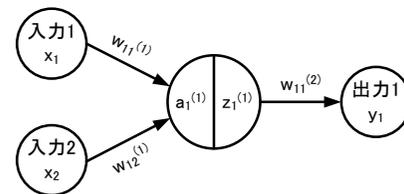
- 1943年 ウォーレン・マカロックとウォルター・ピッツが形式ニューロンを発表
- 1958年 フランク・ローゼンブラットがパーセプトロンを発表
- 1969年 マービン・ミンスキーとシーモア・パパートが著書『パーセプトロン』の中で、単純パーセプトロンは線形分離不可能なパターンを識別できない事を示した。
- 1979年 福島邦彦がネオコグニトロンを発表し文字認識に使用、後にこれが**畳み込みニューラルネットワーク**へと発展
- 1982年 ジョン・ホップフィールドによってホップフィールド・ネットワーク（再帰型ニューラルネットワーク）が提案
- 1985年 ジェフリー・ヒントンらによりボルツマンマシンが提案
- 1986年 デビッド・ラメルハートらにより**誤差逆伝播法（バックプロパゲーション）**が提案（再発見）

## ニューロンの構造

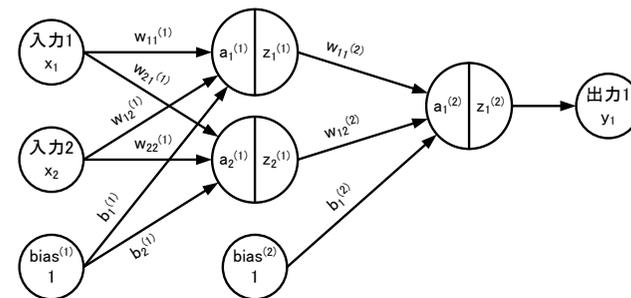
東京医科歯科大学Webより



## ニューロンのモデル

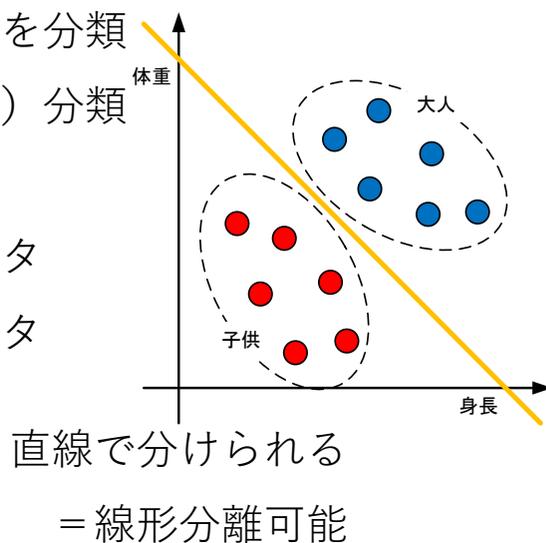


## ニューラルネットワーク



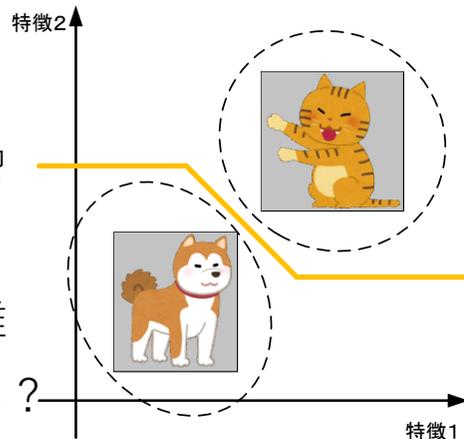
身長と体重で大人と子供を分類  
(データで) (クラスに) 分類

大人の身長と体重のデータ  
子供の身長と体重のデータ  
を学習させる。



犬と猫の画像を分類

犬と猫の情報 (破線内) を学習  
境界線が引かれる  
学習データ付近はきれいに分離  
離れたところは、どうなってる？



AIを使う

- ①学習をおこなう
- ②判断したいデータを入力してみる
- ③結果が出てくる → なぜその結果なのか？  
その理由は、わからない。

多くのパラメータに対し複雑な判別条件

→ニューラルネットワークは効果的

明確な判別条件がある

→サポートベクターマシンなどが効果的

適切な機械学習方法を用いることが必要

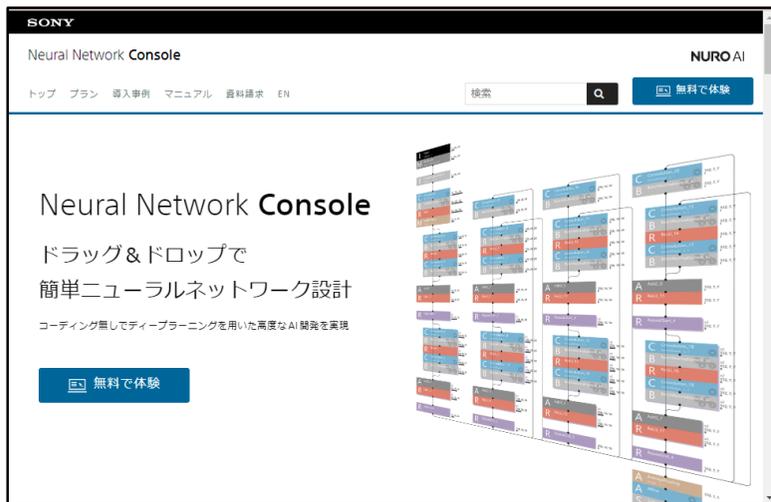
どんなアルゴリズムがあるか？

何が得意なのか？ 知っておく必要がある。

# ツールの入手とインストール

ソニーネットワークコミュニケーションズ株式会社 (Sony Network Communications Inc.) の無料のAI ツール、**Neural Network Console** の入手からインストールまでを説明します。

## Neural Network Console



Deep Learning を利用したAI開発にこれから携わる方、  
既に携わっている方、こんな悩みはありませんか？

- Pythonや数式を学ばないといけない
- ネットワーク構造やパラメータを変更するなどコーディングだと試行錯誤に時間がかかる
- 学習した何十種類ものニューラルネットワークの管理に苦労している
- ニューラルネットワークのチューニングを自動化したい
- 開発環境の構築に手間と費用がかかる

Neural Network Console で全て解決！

## 特徴



ドラッグ&ドロップによる  
簡単編集

豊富なレイヤーを駆使して、ニューラルネットワークを設計しましょう。新しいアイデアの反映もあっという間なので、試行錯誤も苦になりません。



ニューラルネットワークの  
構造自動探索

より性能が高く、軽量のニューラルネットワークを自動的に見つけてくれます。もう面倒なチューニング作業はツールに任せてしましましょう。



すぐに学習、  
すぐに結果を確認

ネットワークを設計したら、ボタン1つで Neural Network Libraries による高速な学習がスタート。学習の進捗状況や性能は、画面でリアルタイムに確認できます。



学習の履歴を  
集中管理

学習した何十種類ものニューラルネットワークを、履歴として一覧できます。設計したニューラルネットワークと性能の関係も一目瞭然です。



簡単な登録を済ませば  
すぐに開発をスタート

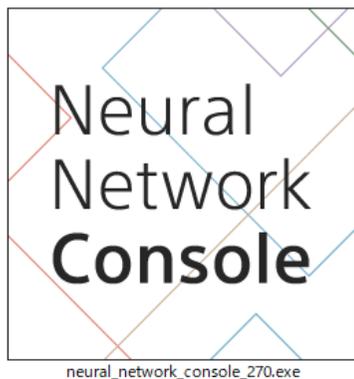
メールアドレス等簡単な登録を済ませればすぐにクラウドで開発をスタート。国内最速のスパコンABCIも利用できるので、高度な学習の処理も高速に行えます。

## 「ニューラルネットワークコンソール」で検索



Sony ネットワーク コミュニケーションズ  
Windows 版をダウンロード  
解凍する

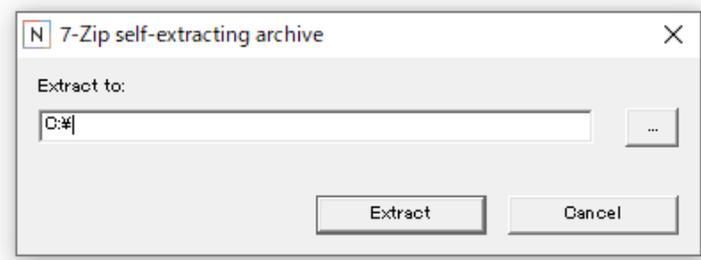
(1)Downloadフォルダにファイルができる



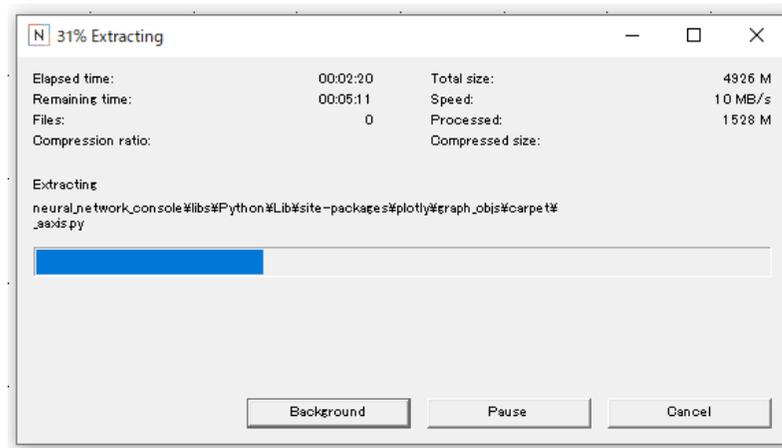
(2)解凍する。(今回は7zipを使用)

C:¥に解凍

C:¥neural\_network\_console が作られる

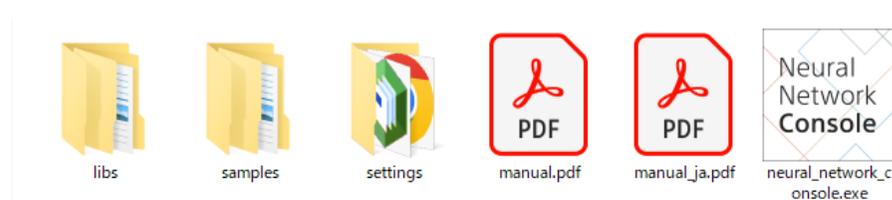


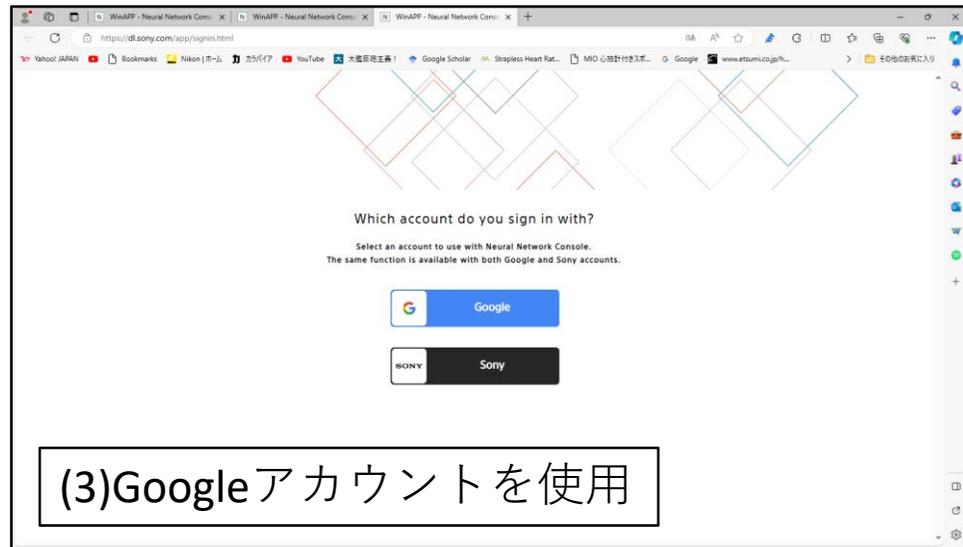
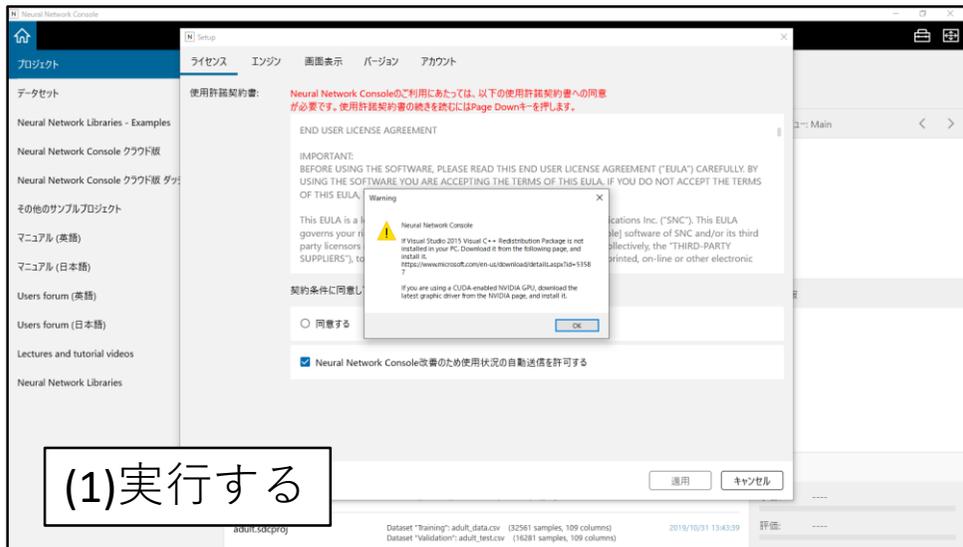
(3)解凍中



(4)解凍結果

C:¥neural\_network\_console 内のファイル







The screenshot displays the Neural Network Console interface. The main area shows a list of projects with columns for project name, dataset information, and creation time. The right sidebar contains an overview section and a statistics panel.

プロジェクト	新しいプロジェクト	プロジェクトを開く	検索	アクション
データセット	+	🔗	🔍	▼
01_logistic_regression.sdcproj				2023/12/07 7:38:08
C:\neural_network_console\samples\sample_project\tutorial\basics\				
Neural Network Libraries - Examples				
Neural Network Console クラウド版				
Neural Network Console クラウド版 ダッシュボード				
その他のサンプルプロジェクト				
02_binary_cnn.sdcproj				2019/10/31 13:43:25
Dataset "Training": small_mnist_4or9_training.csv (1500 samples, 2 columns) Dataset "Validation": small_mnist_4or9_test.csv (500 samples, 2 columns)				
C:\neural_network_console\samples\sample_project\tutorial\basics\				
マニュアル (英語)				
06_auto_encoder.sdcproj				2019/10/31 13:43:25
Dataset "Training": small_mnist_4or9_training.csv (1500 samples, 2 columns) Dataset "Validation": small_mnist_4or9_test.csv (500 samples, 2 columns)				
C:\neural_network_console\samples\sample_project\tutorial\basics\				
マニュアル (日本語)				
Users forum (英語)				
10_deep_mlp.sdcproj				2019/10/31 13:43:25
Dataset "Training": small_mnist_4or9_training.csv (1500 samples, 2 columns) Dataset "Validation": small_mnist_4or9_test.csv (500 samples, 2 columns)				
C:\neural_network_console\samples\sample_project\tutorial\basics\				
Users forum (日本語)				
Lectures and tutorial videos				
11_deconvolution.sdcproj				2019/10/31 13:43:25
Dataset "Training": small_mnist_4or9_training.csv (1500 samples, 2 columns) Dataset "Validation": small_mnist_4or9_test.csv (500 samples, 2 columns)				
C:\neural_network_console\samples\sample_project\tutorial\basics\				
Neural Network Libraries				
12_residual_learning.sdcproj				2019/10/31 13:43:25
Dataset "Training": small_mnist_4or9_training.csv (1500 samples, 2 columns) Dataset "Validation": small_mnist_4or9_test.csv (500 samples, 2 columns)				
C:\neural_network_console\samples\sample_project\tutorial\basics\				

統計情報	
Output	0
CostParameter	0
CostAdd	0
CostMultiply	0
CostMultiplyAdd	0
CostDivision	0
CostExp	0
CostIf	0

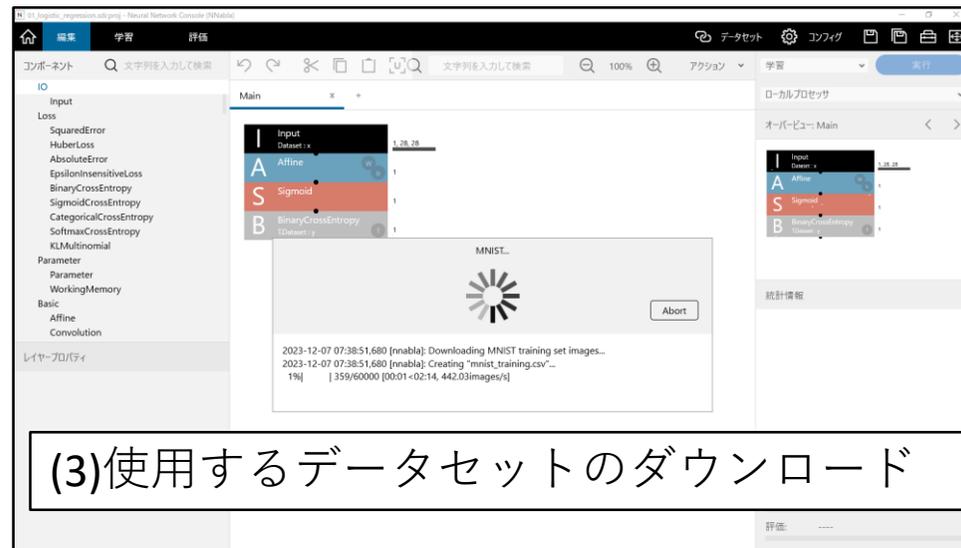
タスク	
学習:	----
評価:	----

# サンプルを実行

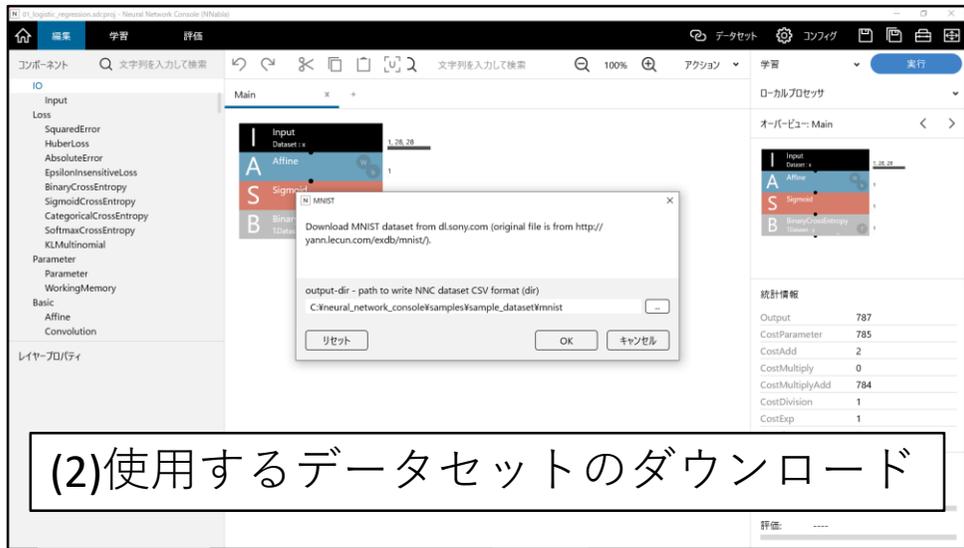
用意されているサンプルを実行し、AIツールでおこなう基本的な作業手順についてみていきます。



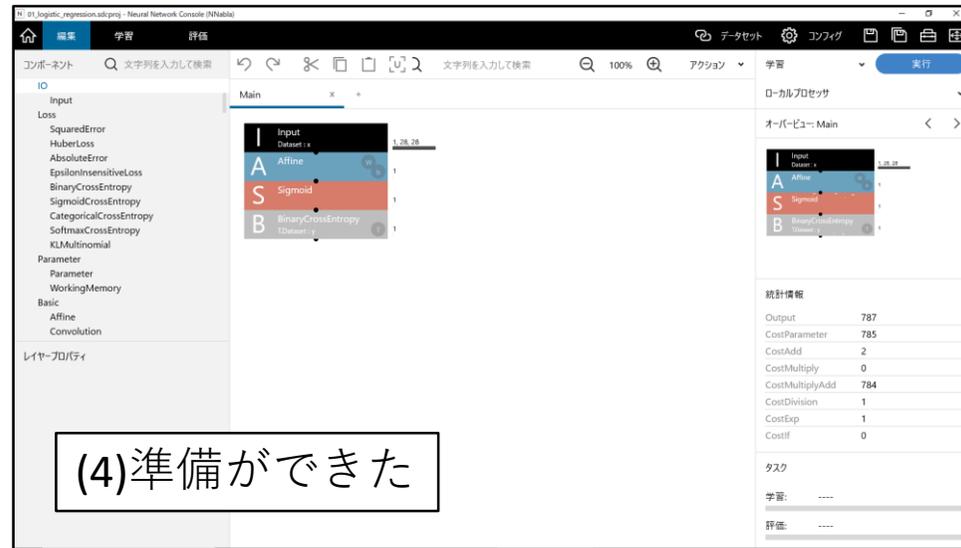
(1) サンプルの 01\_logistic regression (ロジスティック回帰)



(3) 使用するデータセットのダウンロード



(2) 使用するデータセットのダウンロード



(4) 準備ができた

実行

Run training (F3)

統計情報	
Output	787
CostParameter	785
CostAdd	2
CostMultiply	0
CostMultiplyAdd	784
CostDivision	1
CostExp	1
CostIf	0

タスク: ---  
学習: ---  
評価: ---

(5) 学習の実行

学習結果リスト

学習結果リスト	経過時間	残り時間	合計時間
20231207_074630	00:00:10	00:00:14	00:00:25 (6 / 100)

Learning Curve

タスク: 20231207\_074630  
評価: 20231207\_074630

(6) 学習中

学習結果リスト

学習結果リスト	経過時間	残り時間	合計時間
20231207_074630	00:00:25	00:00:00	00:00:25 (100 / 100)

Learning Curve

タスク: 20231207\_074630  
評価: 20231207\_074630

2023-12-07 07:46:55.157 [nnabla]: epoch 97 of 100 cost=0.062549 time=(16.0s / 16.5s)  
2023-12-07 07:46:55.374 [nnabla]: epoch 98 of 100 cost=0.053480 time=(16.2s / 16.5s)  
2023-12-07 07:46:55.558 [nnabla]: epoch 99 of 100 cost=0.064627 time=(16.4s / 16.5s)  
2023-12-07 07:46:56.968 [nnabla]: epoch 100 of 100 cost=0.060106 [train\_error=0.056343, valid\_error=0.119016] time=(16.5s / 16.5s)  
2023-12-07 07:46:56.039 [nnabla]: Training Completed.  
NNabla command line interface (Version:1.33.1, build:230206070804, Callback:NNabla SSH callback module.)

(7) 学習完了



**(11) 評価結果個別(1)**

Index	x:image	y:label;4;9	y'
1	mnist_validation_448.png 4	0	0.0459364
2	mnist_validation_445.png 4	0	0.0023943104
3	mnist_validation_997.png 9	1	0.9954808
4	mnist_validation_998.png 9	1	0.98069245

**(12) 評価結果全体(2)**

	y'=0	y'=1	Recall
y:label=0(4)	238	12	0.952
y:label=1(9)	12	238	0.952
Precision	0.952	0.952	
F-Measures	0.952	0.952	

Accuracy: 0.952  
Avg.Precision: 0.952  
Avg.Recall: 0.952  
Avg.F-Measures: 0.952

Index	x:image	y:label;4;9	y'
1	mnist_validation_444.png 4	0	0.0459364
2	mnist_validation_445.png 4	0	0.0023943104
3	mnist_validation_997.png 9	1	0.9954808

文字 4 クラス 0 0.0459364

文字 4 クラス 0 → 0.0023943104

文字 9 クラス 1 → 0.9954808

	y'=0	y'=1	Recall
y:label=0(4)	238	12	0.952
y:label=1(9)	12	238	0.952
Precision	0.952	0.952	
F-Measures	0.952	0.952	
Accuracy	0.952		
Avg.Precision	0.952		
Avg.Recall	0.952		
Avg.F-Measures	0.952		

文字 4 クラス 0、0 → 238件 1 → 12件  
文字 9 クラス 1、0 → 12件 1 → 238件

Accuracy (正解率)  
Precision (適合率、精度)  
Avg.Recall (平均再現率)  
Avg.F-Measures (平均F値)

**Precision (適合率)** 各予測結果の中での正解率

**F-Measures (F値)** 適合率と再現率の調和平均

**Accuracy (正解率)** 全てのデータに対する正答率

**Avg. Precision (適合率)** 予測結果の中での平均正解率

**Avg.Recall (平均再現率)** 正解とされるラベルの平均正答率

**Avg.F-Measures (平均F値)** 適合率と再現率の平均調和

出力結果
  混同行列:
 y - y'
 その他

	y'=0	y'=1	Recall
y:label=0(4)	238	12	0.952
y:label=1(9)	12	238	0.952
Precision	0.952	0.952	
F-Measures	0.952	0.952	
Accuracy	0.952		
Avg.Precision	0.952		
Avg.Recall	0.952		
Avg.F-Measures	0.952		

# サンプルの説明とデータのダウンロード

NeuralNetworkConsoleに用意されているサンプルには、基本的なニューラルネットワークから、コンテストなどで公開された内容のものまで、約70種類を超えたものがあります。

サンプルを動かすとニューラルネットワークでできることが良くわかりますが、学習に必要な時間が長かったり、構造がとても複雑だったり、参考になります。

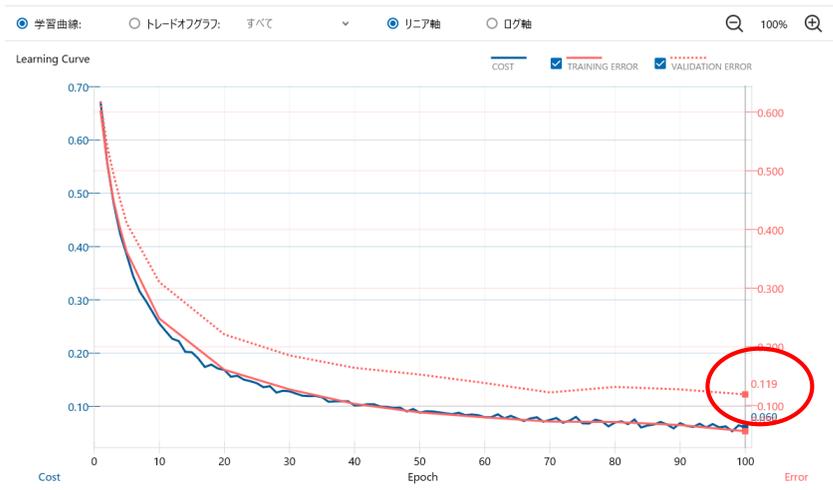
注：情報を集めてまとめた  
すべて正しいか？未確認

1	01_logistic_regression	1層のニューラルネットワークによる画像の2クラス分類
2	02_binary_cnn	4層Convolutional Neural Networkによる画像の2クラス分類
3	06_autoencoder	Auto Encoder
4	10_deep_mlp	Deep Neural Networks
5	11_deconvolution	Convolutional Auto Encoder
6	12_residual_learning	Residual NetworksによるMNISTデータセットの分類
7	adult	adultデータセットを用いたベクトル分類
8	german_credit	?
9	iris	あやめデータセットを用いたベクトル分類
10	mnist_dcgan	Deep Convolutional Generative Adversarial Networks(DCGAN)
11	mnist_dcgan_with_label	Deep Convolutional Generative Adversarial Networks(DCGAN)
12	mnist_vae	Variational Auto Encoder(VAE)
13	gmlp	?
14	mlp-mixer	?
15	repmlp	?
16	resnet-110-deepmil	画像認識ラベルのみからの学習で、Localization (画像位置検出) を可能にするネットワーク
17	resnet-110-mixup	2枚の画像を合成する画像の水増し手法を用いることで、汎化性能を向上させるテクニック
18	resnet-110	Residual Networksによる画像分類 (CIFAR100データセット)
19	resnet-110-cutout	画像の一部を塗りつぶす画像の水増し手法を用いることで、汎化性能を向上させるテクニック
20	resnet-110	Residual Networksによる画像分類 (CIFAR100データセット)
21	LeNet	4層Convolutional Neural Networkによる画像の10クラス分類(MNISTデータセット)39?
22	alexnet	2012年のImageNet Challengeで優勝した画像認識ネットワーク
23	googlenet	2014年のImageNet Challengeで優勝した画像認識ネットワーク
24	nin	Network in Networksと呼ばれる構造のCNN
25	densenet-161	浅いネットワーク~深いネットワークを同時に表現することでパラメータを削減したCNN
26	resnet-101	2015年のImageNet Challengeで優勝した画像認識ネットワーク。Residual Networksと呼ばれる構造を持つ。
27	resnet-152	2015年のImageNet Challengeで優勝した画像認識ネットワーク。Residual Networksと呼ばれる構造を持つ。
28	resnet-18	2015年のImageNet Challengeで優勝した画像認識ネットワーク。Residual Networksと呼ばれる構造を持つ。
29	resnet-34	2015年のImageNet Challengeで優勝した画像認識ネットワーク。Residual Networksと呼ばれる構造を持つ。
30	resnet-50	2015年のImageNet Challengeで優勝した画像認識ネットワーク。Residual Networksと呼ばれる構造を持つ。
31	resnext-101	Grouped Convolutionを用いたResidual Networks
32	shufflenet-0.5x	シャッフル機構によりパラメータ数と演算量を大幅に削減したCNN
33	shufflenet-2.0x	シャッフル機構によりパラメータ数と演算量を大幅に削減したCNN
34	shufflenet	シャッフル機構によりパラメータ数と演算量を大幅に削減したCNN
35	squeezenet11	ボトルネック構造によりパラメータ数を大幅に削減したCNN
36	vgg-11	2013年のImageNet Challengeで優勝した画像認識ネットワーク
37	vgg-13	2013年のImageNet Challengeで優勝した画像認識ネットワーク
38	vgg-16	2013年のImageNet Challengeで優勝した画像認識ネットワーク
39	LeNet	4層Convolutional Neural Networkによる画像の10クラス分類(Fashion MNISTデータセット)21?

40	semi_supervised_learning_VAT	VATと呼ばれるテクニックを用いた半教師あり (少ないラベル付きデータを用いた) 学習
41	sin_wave_anomaly_detection	Auto Encoderを用いてサイン波に含まれるノイズを検出
42	binary_connect_mnist_LeNet	重みを2値化することでパラメータを大幅に削減したCNN
43	binary_connect_mnist_MLP	重みを2値化することでパラメータを大幅に削減したDNN
44	binary_net_mnist_LeNet	重みとデータパスを2値化することでパラメータ、演算量を大幅に削減したCNN
45	binary_net_mnist_MLP	重みとデータパスを2値化することでパラメータ、演算量を大幅に削減したDNN
46	binary_weight_mnist_MLP	重みを2値化することでパラメータを大幅に削減したDNN
47	01_visualize_weight_of_feature	入力特徴量の重み学習により認識結果に影響を及ぼす重要な特徴量の可視化する方法
48	02_l1_regularization	L1 正則化により認識結果に影響を及ぼす重要な特徴量を可視化する方法
49	03_attention	Attentionにより、認識のために注目された入力データの箇所を明らかにする方法
50	04_interface_result_at_each_layer	各層の出力結果を可視化する方法
51	05_Mcdropout	ニューラルネットワークによる予測結果の信頼度を可視化する方法
52	cyclegan_mnist_4and9	?
53	super_resolution	低解像度の画像を入力し高解像度の画像に出力をする画像超解像のサンプルプロジェクト。
54	20newsgroups_classification	シンプルなテキストの2クラス分類
55	20newsgroups_lstm_language_model	LSTMを用いた言語モデルの学習
56	20newsgroups_transformer_language_model	Transformerを用いた言語モデルの学習
57	20newsgroups_word_embedding	COBWを用いたWord Embeddingの学習 (Word2vec)
58	synthetic_image_object_detection	円、三角、四角形、五角形を含む人工画像データを用いた物体検出
59	synthetic_image_object_detection_centernet	画像中の複数物体の大きさや位置を出力するモデル。CenterNetと呼ばれるネットワーク構造を利用。バックボーンはResNet-18を利用。
60	bidirectional_elman_net	双方向のRecurrent Neural Networks
61	elman_net	最もシンプルなRecurrent Neural Network
62	elman_net_with_attention	Attention機構を持つRecurrent Neural Network
63	gated_recurrent_unit(GRU)	Gated Recurrent Unit (GRU) と呼ばれるRecurrent Neural Networksの構造
64	long_short_term_memory(LSTM)	Long Short Term Memory (LSTM) と呼ばれるRecurrent Neural Networksの構造
65	LSTM_auto_encoder	LSTMを用いたAuto Encoder
66	stacked_GRU	GRUを2回重ねたもの
67	binary_semantic_segmentation	文字と背景を分離する2値セグメンテーション
68	DeeplabV3plus	画像のセグメンテーション。DeepLabV3+と呼ばれるネットワーク構造を利用(2クラス分類)。バックボーンはXceptionを使用。
69	FCN-VGG16	画像のセグメンテーション。FCNと呼ばれるネットワーク構造を利用(2クラス分類)。バックボーンはVGG16を使用。
70	PSPNet-ResNet50	画像のセグメンテーション。PSPNetと呼ばれるネットワーク構造を利用(2クラス分類)。バックボーンはresnet-50を使用。アップサンプリングに線形補完ではなくdeconvolution層を使用。
71	PSPNet-ResNet50_Remodeling	画像のセグメンテーション。PSPNetと呼ばれるネットワーク構造を利用(2クラス分類)。バックボーンはresnet-50を使用。アップサンプリングに線形補完ではなくdeconvolution層を使用。
72	unetlike_125px	画像のセグメンテーション。U-Netと呼ばれるネットワーク構造を利用 (20クラス分類)
73	unetlike_125px_person	画像のセグメンテーション。U-Netと呼ばれるネットワーク構造を利用 (2クラス分類)
74	noise_reduction	?
75	wav_keyboard_sound	4つのキーボードのキースイッチの種類 (メンブレン、パンタグラフ、メカニカル青軸、メカニカル赤軸) を打音から判別
76	resnet110-attention-branch-network	?
77	resnet56-tracin	?

## 01\_logistic\_regression

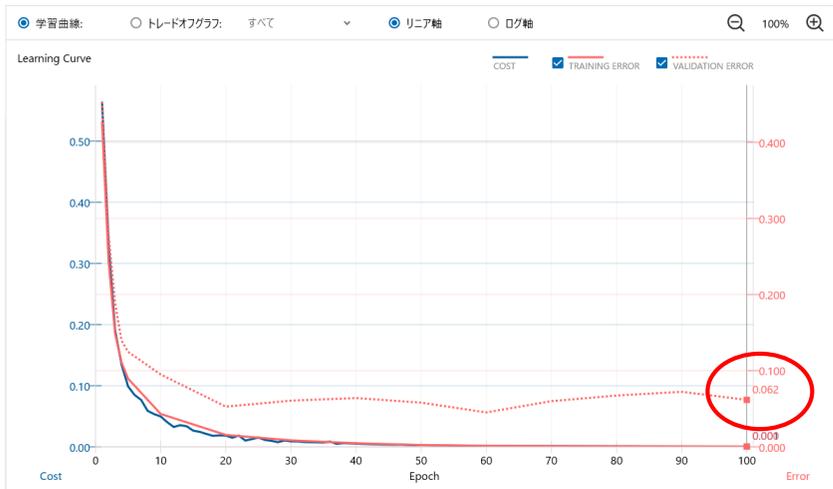
I	Input	Dataset : x	1, 28, 28
A	Affine		1
S	Sigmoid		1
B	BinaryCrossEntropy	T:Dataset : y	1



	y'=0	y'=1	Recall
y:label=0(4)	238	12	0.952
y:label=1(9)	12	238	0.952
Precision	0.952	0.952	
F-Measures	0.952	0.952	
Accuracy	0.952		
Avg.Precision	0.952		
Avg.Recall	0.952		
Avg.F-Measures	0.952		

## 02\_binary\_cnn

I	Input	Dataset : x	1, 28, 28
C	Convolution	KernelShape : 5, 5	16, 24, 24
M	MaxPooling	Shape : 2, 2	16, 12, 12
T	Tanh		16, 12, 12
C	Convolution_2	KernelShape : 5, 5	8, 8, 8
M	MaxPooling_2	Shape : 2, 2	8, 4, 4
T	Tanh_2		8, 4, 4
A	Affine		10
T	Tanh_3		10
A	Affine_2		1
S	Sigmoid		1
B	BinaryCrossEntropy	T:Dataset : y	1



	y'=0	y'=1	Recall
y:label=0(4)	246	4	0.984
y:label=1(9)	6	244	0.976
Precision	0.9761	0.9838	
F-Measures	0.9800	0.9798	
Accuracy	0.98		
Avg.Precision	0.9800		
Avg.Recall	0.98		
Avg.F-Measures	0.9799		

(1)ニューラルネットワークの深さ

01→1層

02→4層

(2)学習曲線

01→ゆっくり降下

VALIDATION ERROR=0.119

02→急速に降下

VALIDATION ERROR=0.062

(3)評価結果

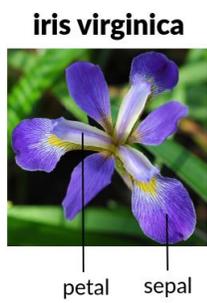
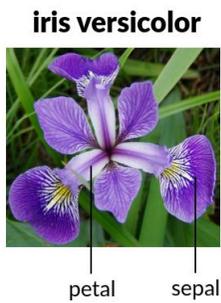
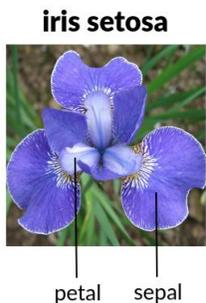
01 Accuracy=0.952

02 Accuracy= 0.98

95.2%と98.0%

機械学習でよく使用されるデータセット

	A	B	C	D	E
1	x_0:Sepal length	x_1:Sepal width	x_2:Petal length	x_3:Petal width	y:label
2	5.1	3.5	1.4	0.2	0
3	4.9	3	1.4	0.2	0
4	4.7	3.2	1.3	0.2	0
5	4.6	3.1	1.5	0.2	0
6	5	3.6	1.4	0.2	0
7	5.4	3.9	1.7	0.4	0
8	4.6	3.4	1.4	0.3	0
9	5	3.4	1.5	0.2	0
10	4.4	2.9	1.4	0.2	0



<https://morioh.com/p/eafb28ccf4e3>

A:がく片 (sepal) の長さ (cm)

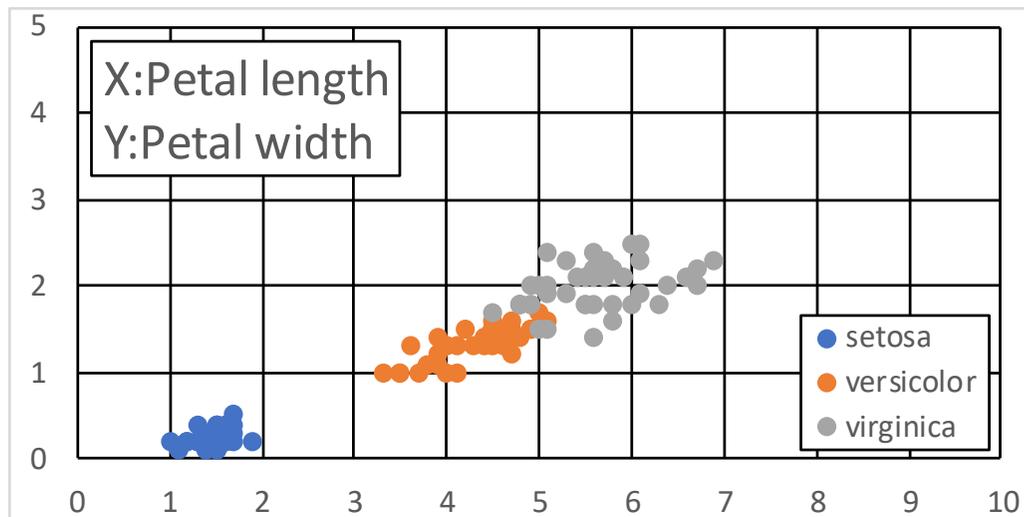
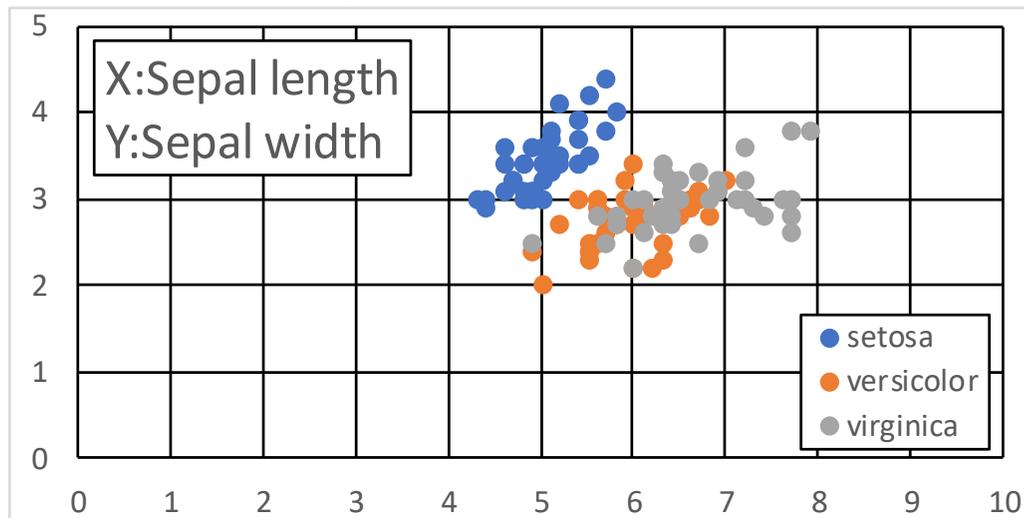
B:がく片 (sepal) の幅 (cm)

C:花びら (petal) の長さ (cm)

D:花びら (petal) の幅 (cm)

E:(0)setosa (1)versicolor (2)virginica

データセット (学習用) のグラフ



**CNN** (Convolutional Neural Network : 畳み込みニューラルネットワーク) : Convolutionとは、画像処理で使用される手法。元画像と、カーネルと呼ばれる格子状の数値データを、各要素ごとに積和計算し1つの数値に変換する処理。画像処理のフィルタリング処理。

**Auto Encoder** (自己符号化器) : ニューラルネットワークを利用した教師なし機械学習の手法の1つ。

**Deep Neural Network** : ニューラルネットワークは、入力・中間・出力の複数層で構成されている。中間層が2層以上だとDeepNeuralNetworkと呼ばれる。複雑な処理に対応可能であるとされている。

**DCGAN** (Deep Convolutional Generative Adversarial Network : 敵対的生成ネットワーク) : 生成ネットワーク (generator) と識別ネットワーク (discriminator) の2つのネットワークに畳み込みニューラルネットワークを用いる。画像生成などに使用される。

**VAT** (Virtual Adversarial Training) : 少量のラベル付きデータと、大量のラベル無しデータで学習する方法。半教師あり学習。

**LSTM** (Long Short Term Memory : 長・短期記憶) : 自然言語処理や時系列データの予測といった場所で利用される技術。

**RNN** (Recurrent Neural Network : 再帰ニューラルネットワーク) : 過去の情報を利用し現在および将来の入力に対するネットワークの性能を向上させる、ディープラーニングの構造

**GRU** (Gated Recurrent Unit : ゲート付き再帰ユニット) : LSTMのゲート機構の3ゲートのうち、出力ゲートを無くして2ゲート構成にし、計算効率と軽量化を図ったLSTMの改良版

**FCN** (Fully Convolutional Network : 完全畳み込みネットワーク) : セグメンテーション画像などの他チャンネル画像を推定する場合に、全結合層を使わず、全て畳み込み層だけで構成されているCNN

**PSPNet** (Pyramid Scene Parsing Network) : EncoderとDecoderの間にピラミッドプーリングモジュールを使用した構造。画像に含む複数物体に、物体の領域と物体名をピクセルレベルで付与する場合などに使用される

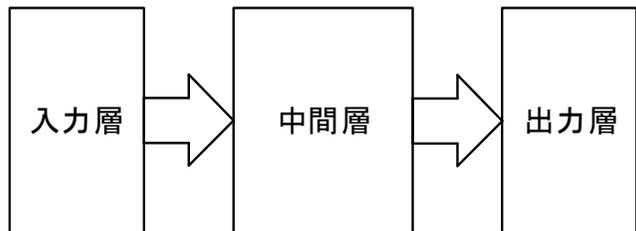
**U-Net** : 画像のセグメンテーション (物体がどこにあるか) を推定するためのネットワーク。その構造が左右対象でアルファベットの「U」に似ており「U-net」と呼ばれる。医療画像セグメンテーションで人気のある深層学習モデル。

# ニューラルネットワークの部品

ニューラルネットワークを構成する、主なパーツとその基本的な動作について説明します。

## NeuralNetworkの基本構造

入力層、中間層、出力層で構成される

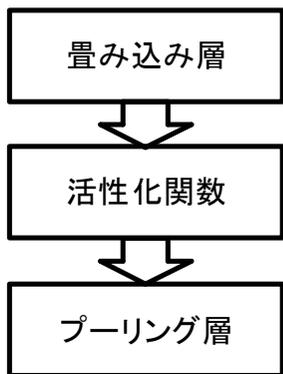
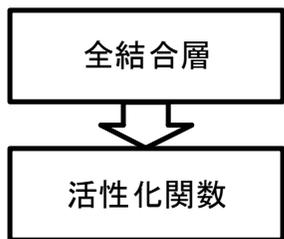


## 中間層の構成

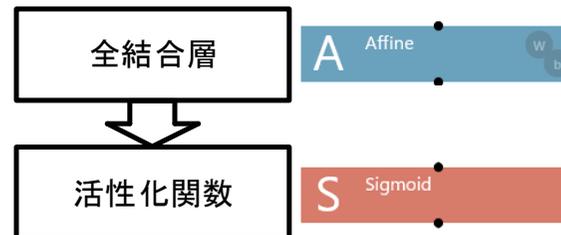
全結合要素、畳み込み要素により構成される

全結合要素の構成

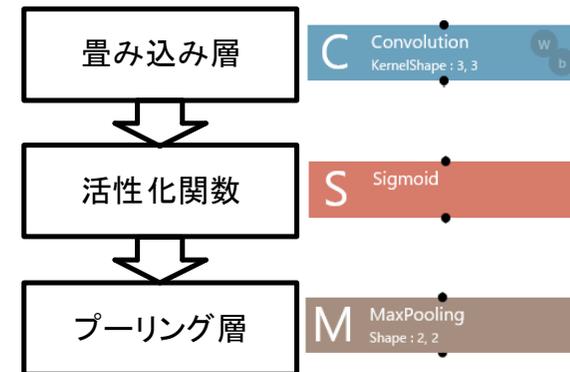
畳み込み要素の構成



## 全結合層



## 畳み込み層

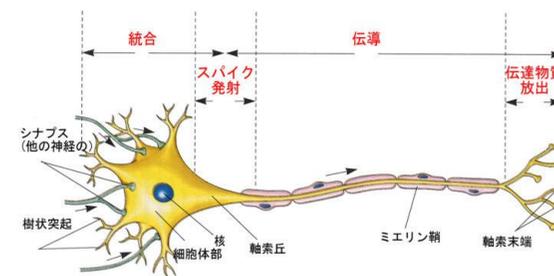
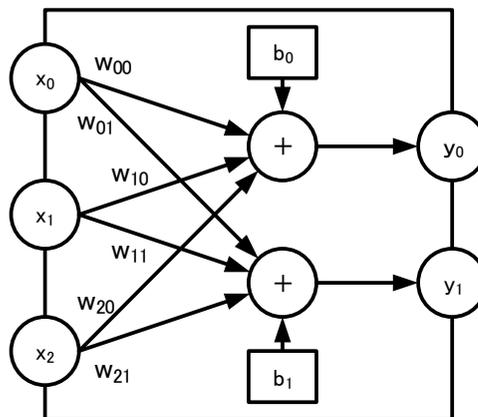


## コンポーネントの詳細



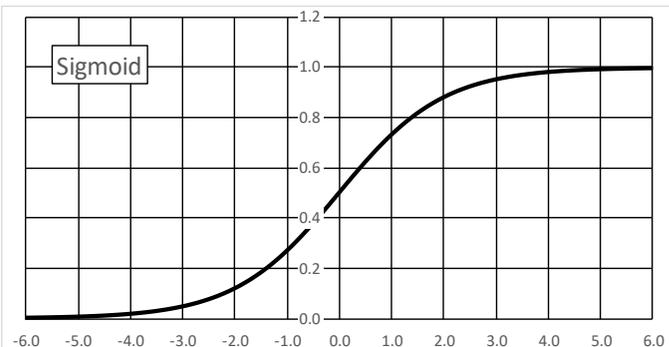
## Affine

入力に重み付けを掛け、合計する



$$\begin{bmatrix} W_{00} & W_{10} & W_{20} \\ W_{01} & W_{11} & W_{21} \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} y_0 \\ y_1 \end{bmatrix}$$

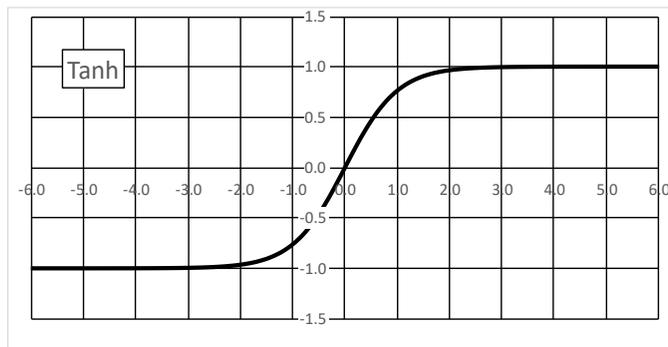
シグモイド関数



導関数の値が、**0.25**と小さい

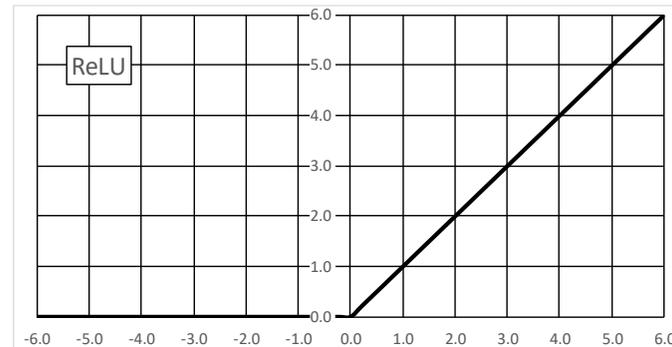
最近、中間層にあまり使われず、  
2クラス分類の出力層に使用

ハイパボリックタンジェント関数



導関数の値が、**1.0**と大きい

レルー (Rectified Linear Unit)



導関数の値が、**1.0**と大きい

計算が容易で高速

全結合層のAffineにより求められた値は、活性化関数によって計算される  
プラス側やマイナス側の大きな値が入力された場合、  
Sigmoidでは  $0 \sim 1$ 、Tanhでは  $-1 \sim 1$ 、ReLUでは  $0 \sim \text{入力値}$  の値を出力することになる。

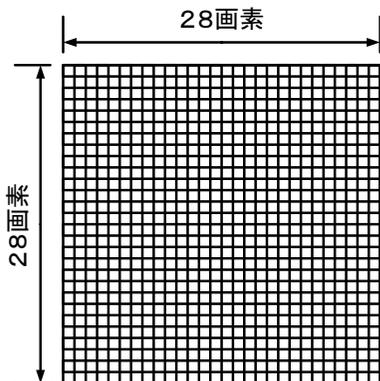


マックスプーリング

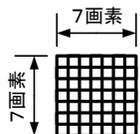
**M** MaxPooling  
Shape : 2, 2

プーリング層の動作

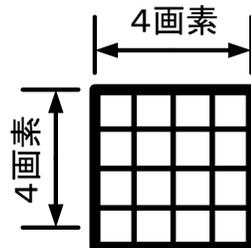
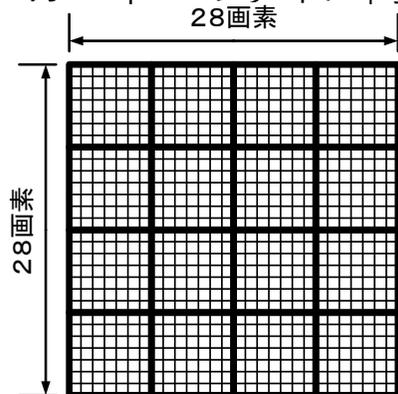
28×28画素の画像



7×7のカーネル



カーネルのサイズ内の、最大値を採用



28×28と

7×7だと

4×4の結果

情報量を減らすことができる

画像を縮小する効果がある

計算量を大幅に減らすことができる

$$28 \times 28 = 784 \text{画素}$$

$$4 \times 4 = 16 \text{画素}$$

49分の1になる

**M** MaxPooling  
Shape : 2, 2

**A** AveragePooling  
Shape : 2, 2

**S** SumPooling  
Shape : 2, 2

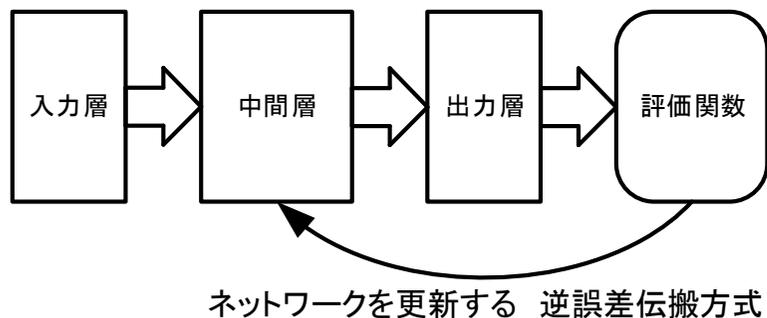
## ニューラルネットワークの学習の仕組み

入力層に、用意された教師ありの学習データを入力

中間層、出力層と計算が行われる

出力と答え（教師データ）を比較

中間層のニューラルネットワークの構造を修正



画像から、犬・猫を判別する

最近の株価から、明日の株価を予測する

評価する値が大きく違う

画像の例→正解・不正解（0 or 1）

株価の例→予測した株価がずれた金額（実数値）

## ニューラルネットワークのできる事

(1) 2クラス分類

写真を、犬と犬以外に分ける

(2) 多クラス分類

写真が、犬か、猫か、猿かに分ける

(3) 回帰

連続的な数値を予測する

この3種類が、できることの基本

Loss レイヤーとその直前の Activation レイヤーの組み合わせ

2クラス分類 → Sigmoid + BinaryCrossEntropy レイヤー

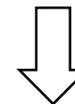
多クラス分類 → Softmax + CategoricalCrossEntropy レイヤー

連続値回帰 → Activation レイヤーなしの SquaredError レイヤー

オートエンコーダ この評価関数

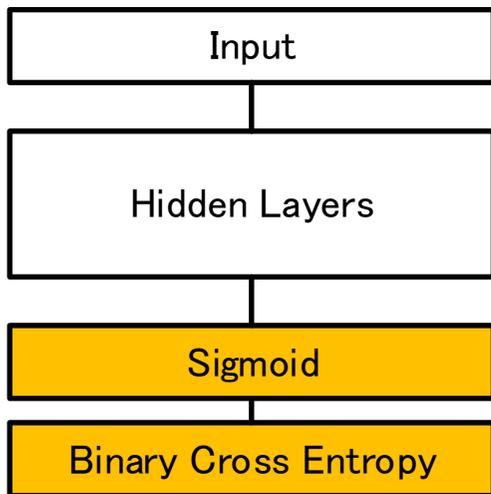
工場では、大量の良品と少しの不良品データしか入手できない

良品の特徴だけを学習させ、不良品を見つける場合に使用



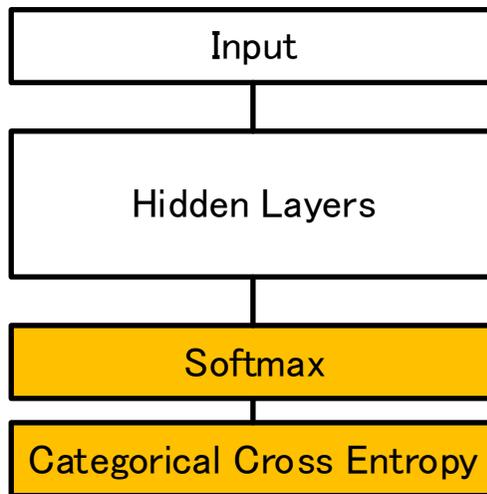
(1) 2クラス分類

入力されたもの、それ以外に分類



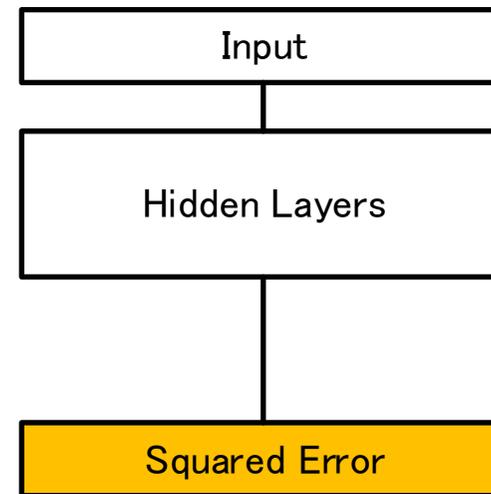
(2) 多クラス分類

入力を、複数候補に分類



(3) 回帰

入力データに基づき連続値を出力



# ニューラルネットワークの作り方

ここでは、エクセルを使って学習・評価用データを作成します。さらに、ニューラルネットワークを設定し、学習・評価を行います。

自分が使用したいデータを使って学習させたニューラルネットワークを作ることができるようになります。まずは、2クラス分類。

## NNCの使用手順

### (0) 課題の説明

### (1) 学習用データを用意する

エクセルを使用し学習用データを準備

### (2) 評価用データを用意する

エクセルを使用し学習用データを準備

### (3) 用意したデータをセットする

学習用・評価用データファイルを指定する

### (4) ニューラルネットワークを設計する

パーツを置いて、接続する

### (5) 学習に関するパラメータを設定

学習を繰り返す回数 → エポック

学習時の使用するデータ数 → バッチサイズ  
など

### (6) 学習

実際に学習を行い、学習の様子を確認する

### (7) 評価

学習が終わったモデルに、評価用データを入力

### (8) 評価結果

評価結果を確認する

# (0) 課題1の説明

- (1) 学習用データ用意
- (2) 評価用データ用意
- (3) データをセット
- (4) ニューラルネットワークを設計
- (5) 学習に関するパラメータを設定
- (6) 学習
- (7) 評価
- (8) 評価結果

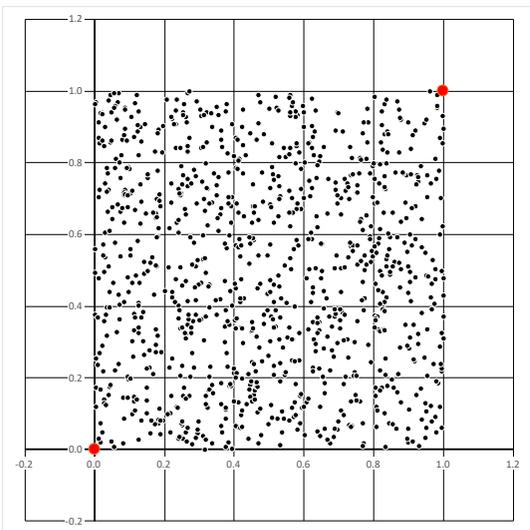
- ①点0(0,0)と点1(1,1)がある
- ②XY座標がそれぞれ、0から1までの乱数で決めた点を1000個学習用として準備
- ③XY座標がそれぞれ、0から1までの乱数で決めた点を1000個評価用として準備
- ④学習用、評価用のそれぞれの点について、点0に近ければクラス0、点1に近ければクラス1とする

エクセルを使って下記のような表を作成  
学習・評価用データを用意する。

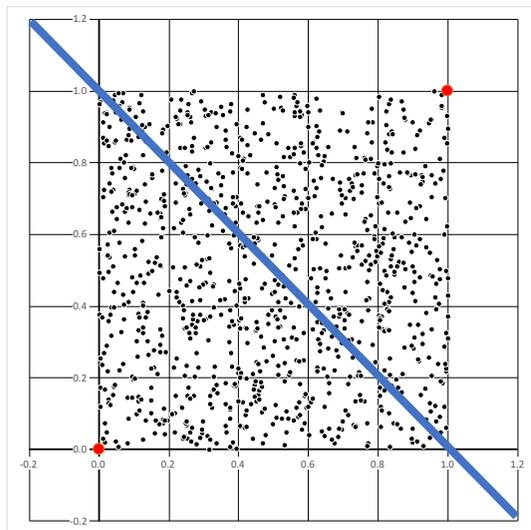
No	X0	X1	P0 (0,0)からの距離	P1 (1,1)からの距離	Y
1	0.284233	0.922186	0.964995	0.719985	1
2	0.586357	0.093303	0.593734	0.996594	0
3	0.834071	0.031644	0.834671	0.982469	0
4	0.477290	0.483275	0.679235	0.735004	0
5	0.872728	0.648696	1.087410	0.373648	1
6	0.434740	0.291877	0.523633	0.906067	0
7	0.962327	0.415521	1.048204	0.585692	1
8	0.332753	0.896369	0.956139	0.675247	1

~

996	0.995733	0.720613	1.229132	0.279420	1
997	0.249328	0.127730	0.280142	1.150810	0
998	0.358500	0.638661	0.732400	0.736267	0
999	0.947588	0.083321	0.951244	0.918176	1
1000	0.874825	0.028503	0.875289	0.979528	0



学習用、評価用



このように分類される？

# (1・2) 学習・評価用データを用意(1)

- (1) 学習用データ用意
- (2) 評価用データ用意
- (3) データをセット
- (4) ニューラルネットワークを設計
- (5) 学習に関するパラメータを設定
- (6) 学習
- (7) 評価
- (8) 評価結果

①デスクトップの作業用フォルダを使用→



②作業ホルダ内に、課題のホルダがある→  
(中に「bak」というホルダがある)

③課題フォルダ内にエクセルのファイルを作る  
ファイル名は「Data01.xlsx」→



④ B列に、1から1000番までの通し番号を設定する。

⑤ 2・3行目は表の項目、4行目は関数を設定する

⑥ C4~G4を、1003行までコピーする  
罫線を引いて見やすくしておく

	A	B	C	D	E	F	G
1							
2					P0	P1	
3		No	X0	X1	(0,0)からの距離	(1.1)からの距離	Y
4		1	0.760133	0.866265	1.152483	0.274629	1
5		2	0.808059	0.864570	1.15266	0.77802	1
6		3	0.635395	0.932248	1.190	0.846	1
7							
8		5	0.241954	0.266952	0.373153	1.040900	0
9		6	0.642558	0.037951	0.643677	1.026	0
10		7	0.493299	0.403543	0.637332	0.782628	0
1001		998	0.364263	0.220674	0.425892	1.005739	0
1002		999	0.590251	0.074066	0.594880	1.012545	0
1003		1000	0.191208	0.298360	0.354372	1.070721	0
1004							

Formula callouts for the table above:

- Cell B4: `=+RAND()`
- Cell C4: `=+RAND()`
- Cell E4: `=+SQRT((0-C4)^2+(0-D4)^2)`
- Cell F4: `=+SQRT((C4-1)^2+(D4-1)^2)`
- Cell G4: `=+IF(E4<F4,0,1)`

⑦シートをコピーする。シート見出しを変更する。

1000		997	0.257150	0.043276	0.260766	1.211259	0
1001		998	0.331358	0.712402	0.785694	0.727870	1
1002		999	0.842264	0.074416	0.845545	0.938928	0
1003		1000	0.218220	0.210589	0.325516	1.090111	0

Sheet tabs: Data01\_T, Data01\_V

⑧エクセル形式で保存する。



## ⑨シート「Data01\_T」をCSV形式で保存する

1002	999	0.842264	0.074416	0.845545	0.938928	0
1003	1000	0.248220	0.210589	0.325516	1.090111	0

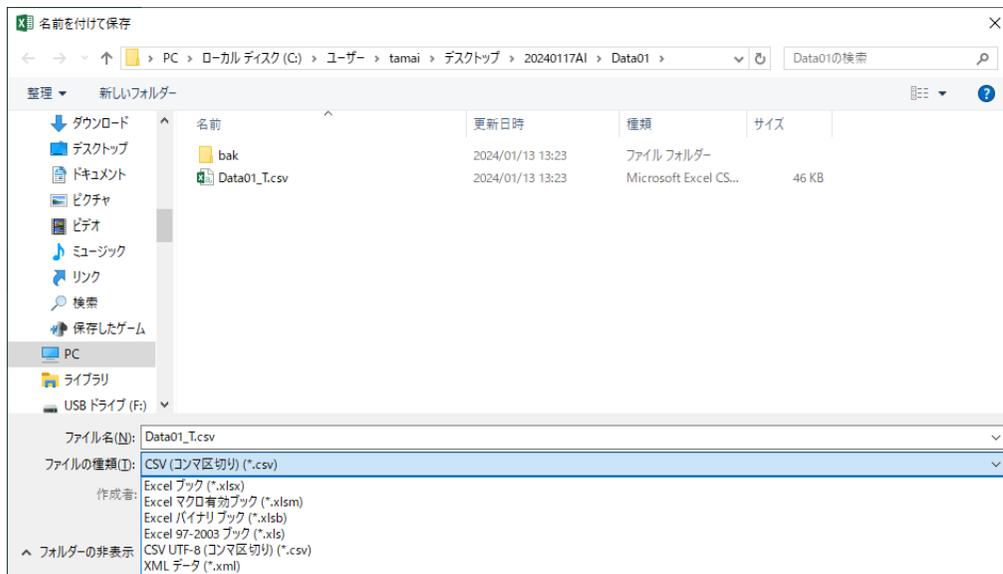
ファイル→名前を付けて保存 Data01\_T.csv

CSV (コンマ区切り) (\*.csv) で保存

## ⑩シート「Data01\_T」をCSV形式で保存する

ファイル→名前を付けて保存 Data01\_V.csv

CSV (コンマ区切り) (\*.csv) で保存



これで、3つのファイルができる



Data01.xlsx



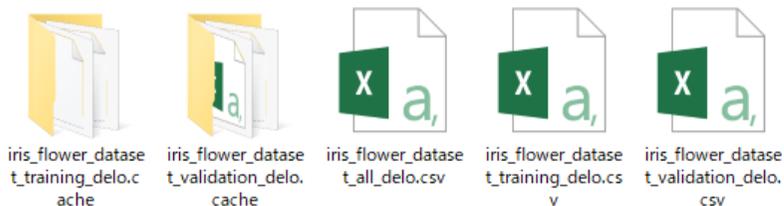
Data01\_T.csv



Data01\_V.csv

デモで入っているデータセットを参考

PC > ローカルディスク (C:) > NeuralNetworkConsole > samples > sample\_dataset > iris\_flower\_dataset >



	A	B	C	D	E
1	x__0:Sepal length	x__1:Sepal width	x__2:Petal length	x__3:Petal width	y:label
2	5.1	3.5	1.4	0.2	0
3	4.9	3.0	1.4	0.2	0
4	4.7	3.2	1.3	0.2	0
5	4.6	3.1	1.5	0.2	0
6	5	3.6	1.4	0.2	0
7	5.4	3.9	1.7	0.4	1

学習用データは、x\_\_0:ラベル x\_\_1:ラベル

教師データは、1つだと、y:ラベル

複数だと、y\_\_0:ラベル、 y\_\_1:ラベル

[注] x\_\_0:Data ← アンダーバ は2つ

## ⑪NeuralNetworkConsole用に修正する

	A	B	C	D	E	F
1				P0	P1	
2	No	X0	X1	(0,0)からの距離	(1,1)からの距離	Y
3	1	0.622857	0.125509	0.635377	0.95235	0
4	2	0.571467	0.844574	1.019745	0.455849	1
5	3	0.070067	0.192998	0.205323	1.231271	0
6	4	0.928176	0.09164	0.932689	0.911195	1
7	5	0.685365	0.071018	0.689034	0.980818	0
8	6	0.410688	0.314469	0.517258	0.904014	0

	A	B	C
1	x__0:X0	x__1:X1	y:Y
2	0.622857	0.125509	0
3	0.571467	0.844574	1
4	0.070067	0.192998	0
5	0.928176	0.09164	1
6	0.685365	0.071018	0
7	0.410688	0.314469	0

不必要な行、列を削除

1行目を変更

ファイルを保存

(csv形式)



Data01\_T.csv



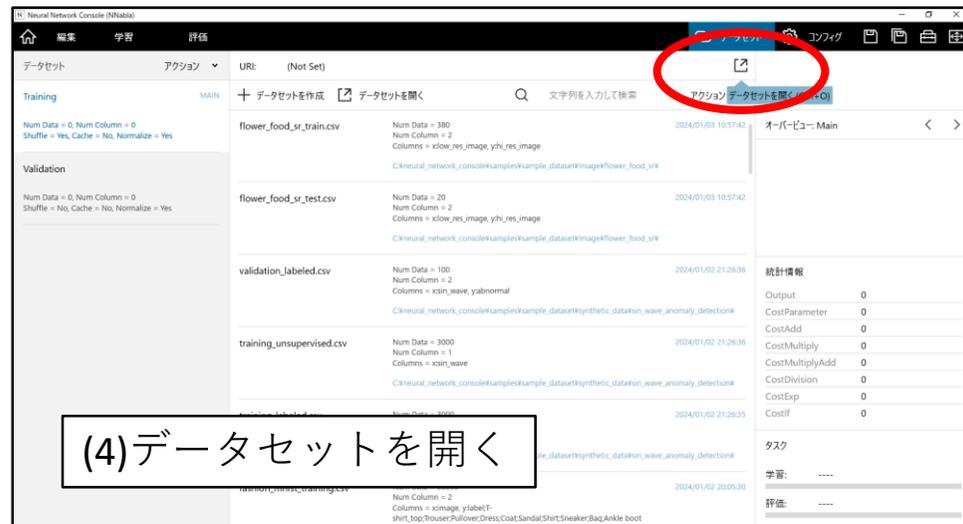
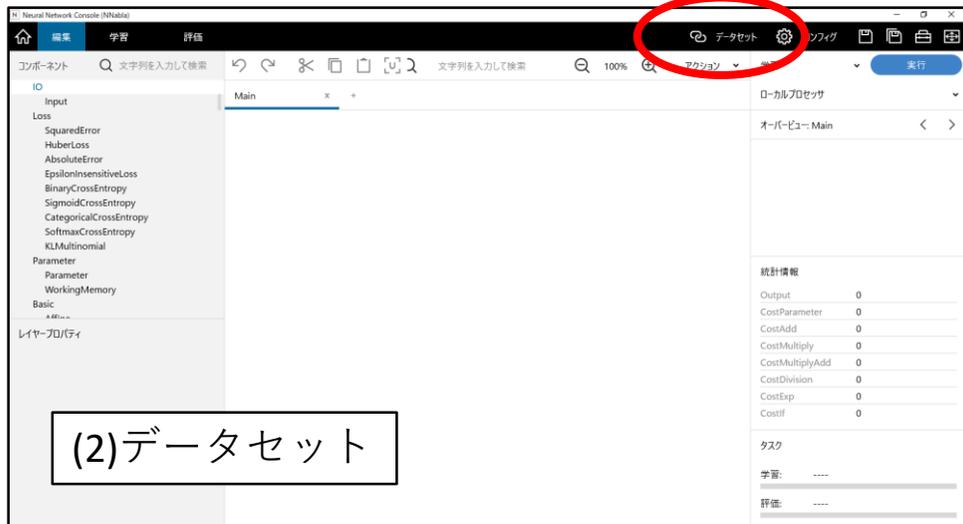
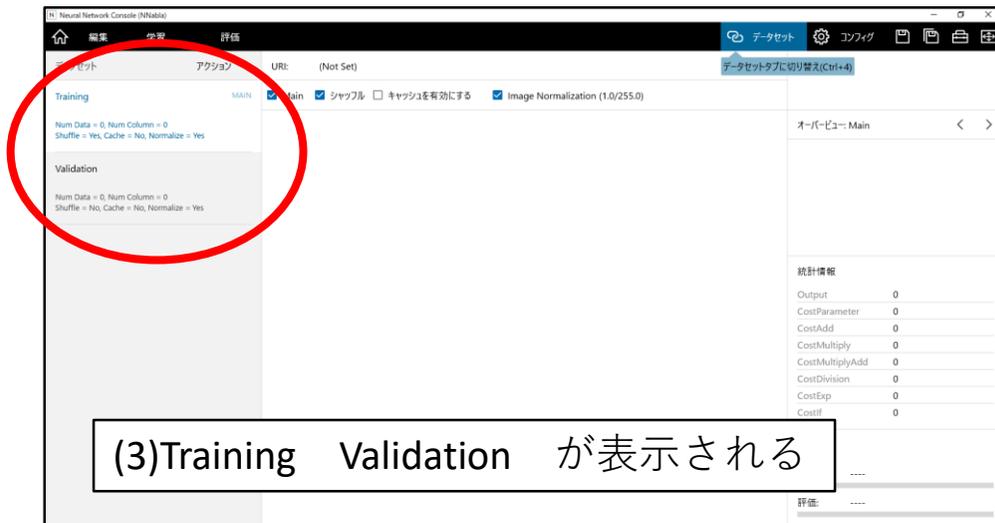
Data01\_V.csv

Data01\_T.csv 学習用

Data01\_V.csv 評価用

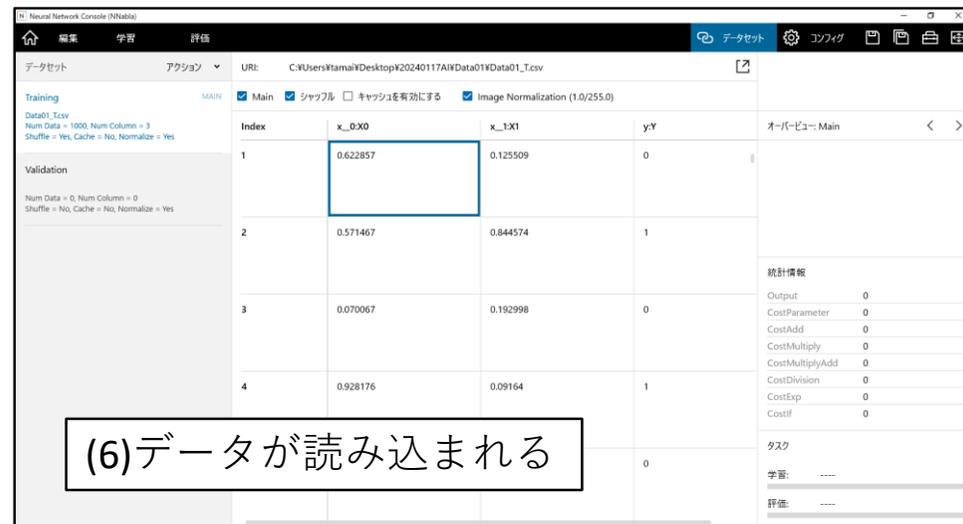
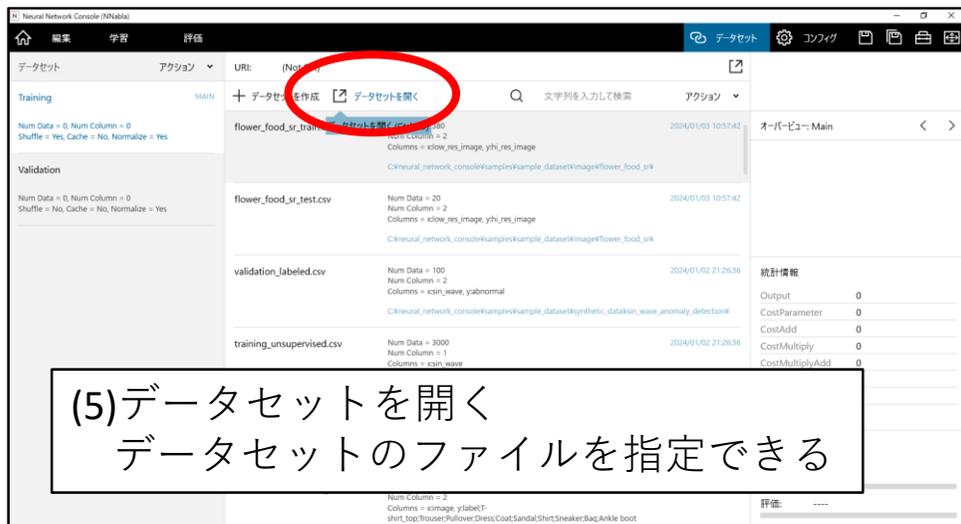
# (3) データをセットする(1)

- (1) 学習用データ用意
- (2) 評価用データ用意
- (3) データをセット
- (4) ニューラルネットワークを設計
- (5) 学習に関するパラメータを設定
- (6) 学習
- (7) 評価
- (8) 評価結果



### (3) データをセットする(2)

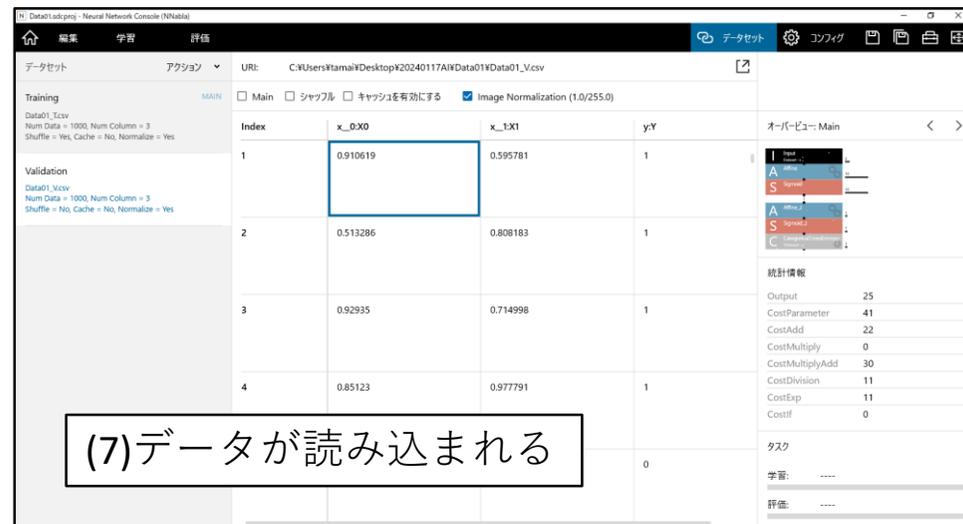
- (1) 学習用データ用意
- (2) 評価用データ用意
- (3) データをセット
- (4) ニューラルネットワークを設計
- (5) 学習に関するパラメータを設定
- (6) 学習
- (7) 評価
- (8) 評価結果



先ほど作った、  
学習用、評価用の  
データセットを選ぶ

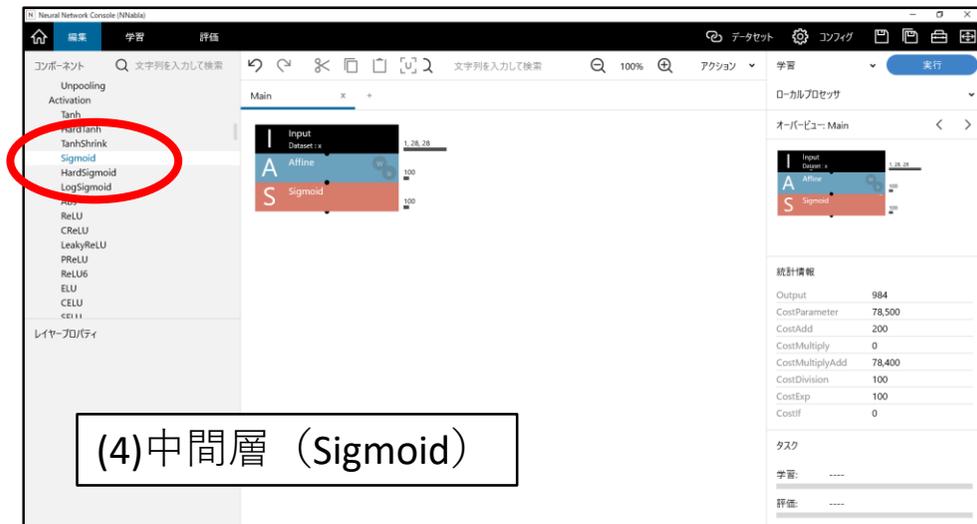
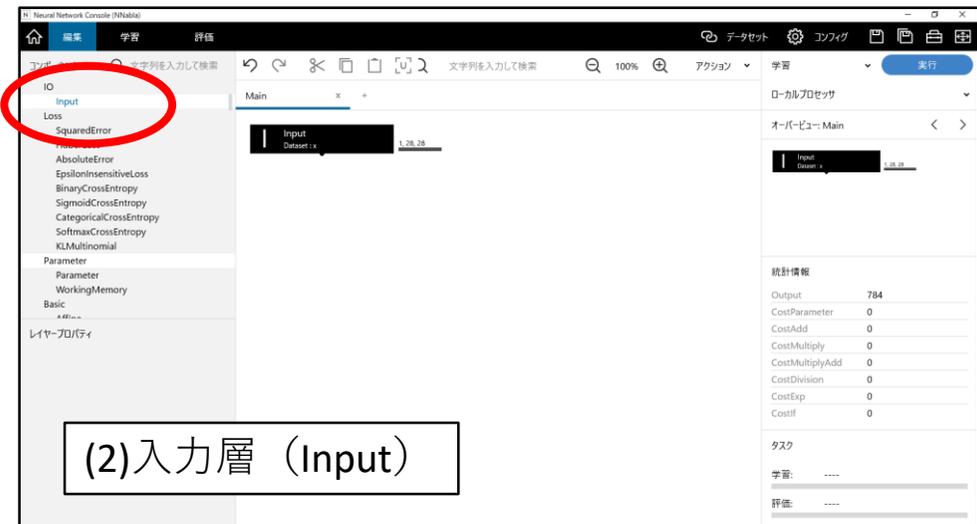
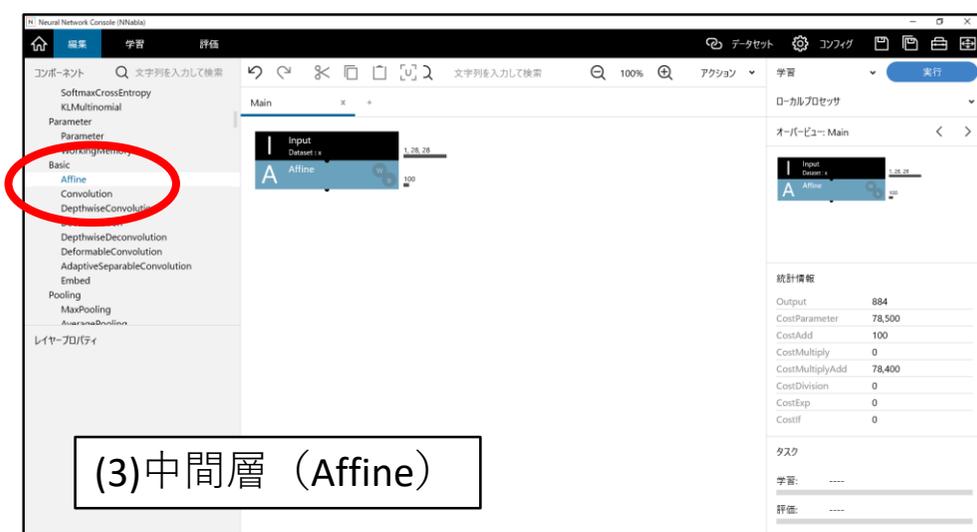
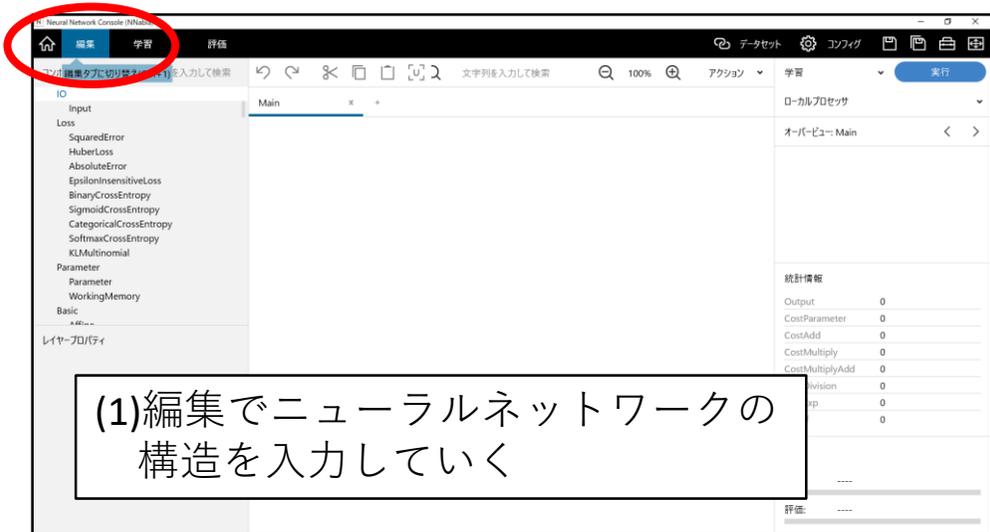
開く  
ファイルの場所を開く  
CSVファイル形式で保存...  
プラグイン ▶  
その他のツール ▶  
データセットの整合性をチェック

右クリックで  
データセットの  
整合性をチェックできる



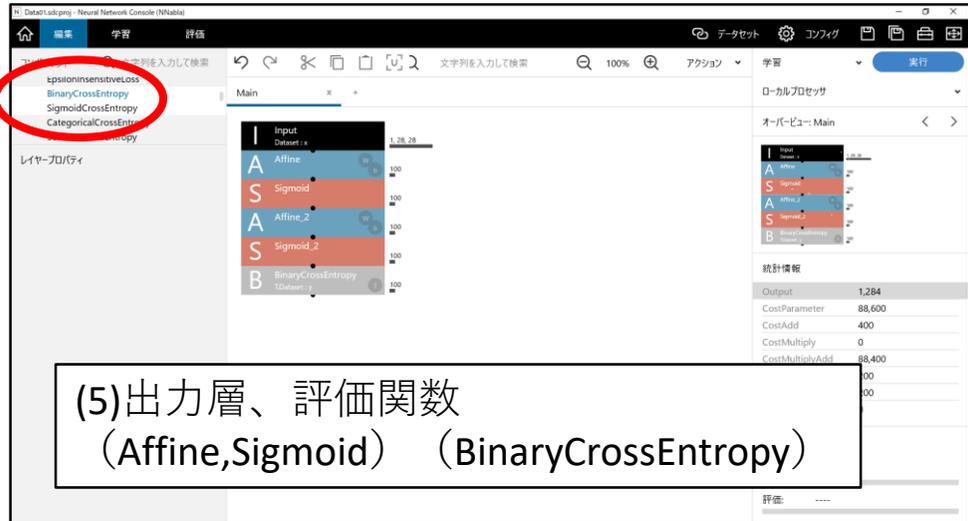
# (4) ニューラルネットワークを設計

- (1) 学習用データ用意
- (2) 評価用データ用意
- (3) データをセット
- (4) ニューラルネットワークを設計
- (5) 学習に関するパラメータを設定
- (6) 学習
- (7) 評価
- (8) 評価結果

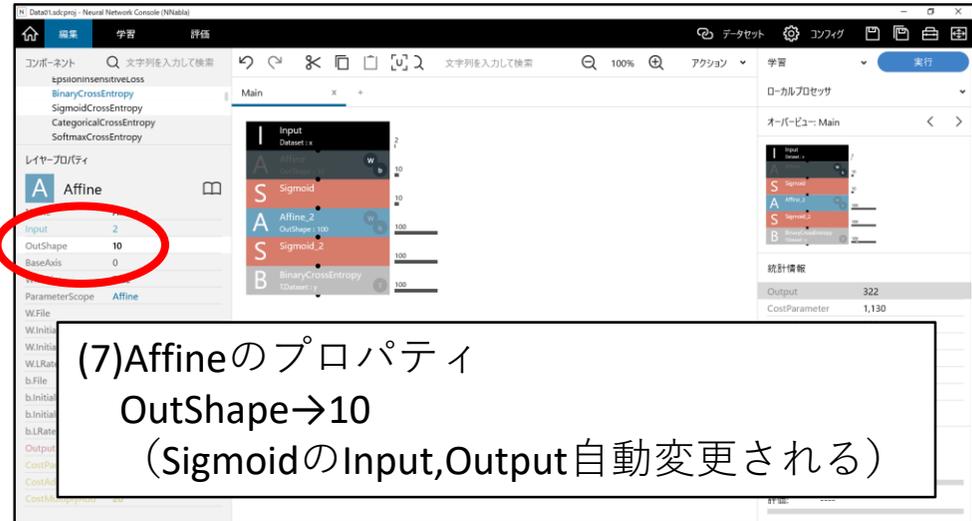


# (4) ニューラルネットワークを設計(1)

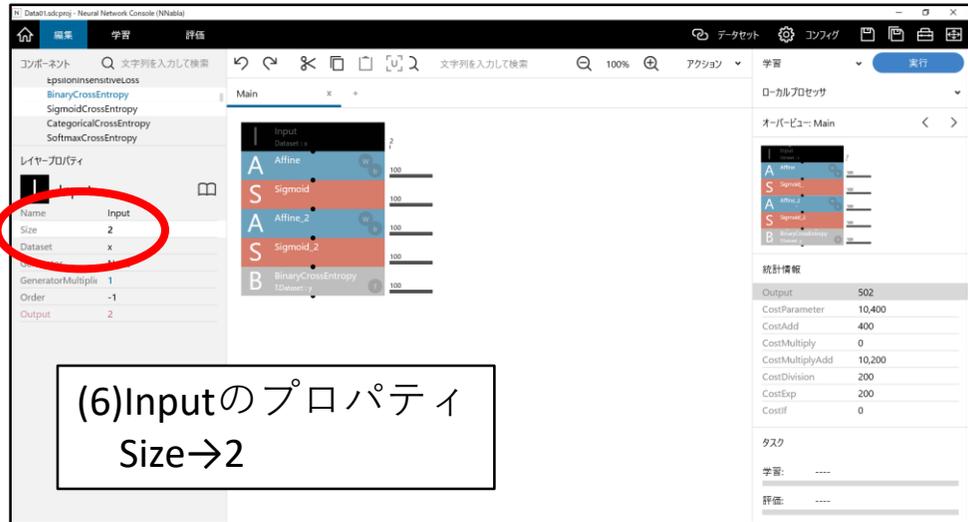
- (1) 学習用データ用意
- (2) 評価用データ用意
- (3) データをセット
- (4) ニューラルネットワークを設計
- (5) 学習に関するパラメータを設定
- (6) 学習
- (7) 評価
- (8) 評価結果



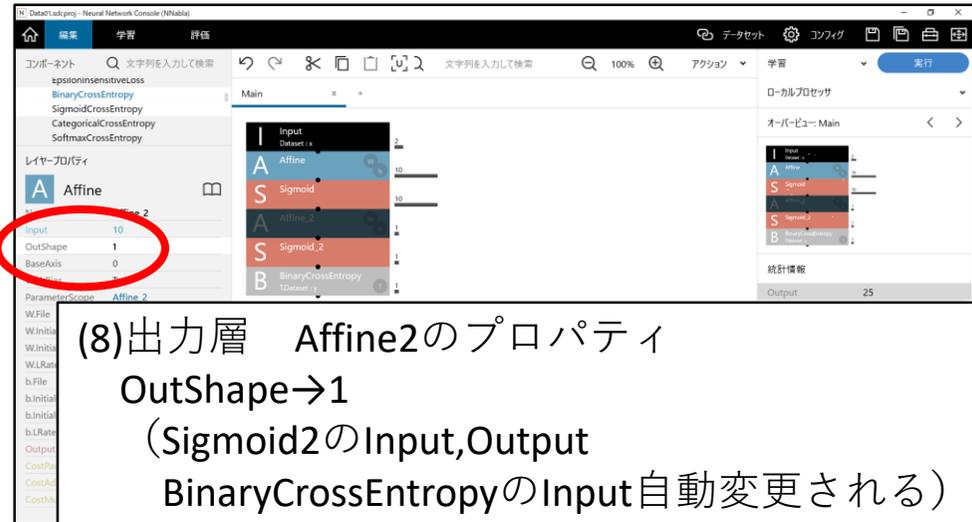
(5)出力層、評価関数  
(Affine,Sigmoid) (BinaryCrossEntropy)



(7)Affineのプロパティ  
OutShape→10  
(SigmoidのInput,Output自動変更される)



(6)Inputのプロパティ  
Size→2

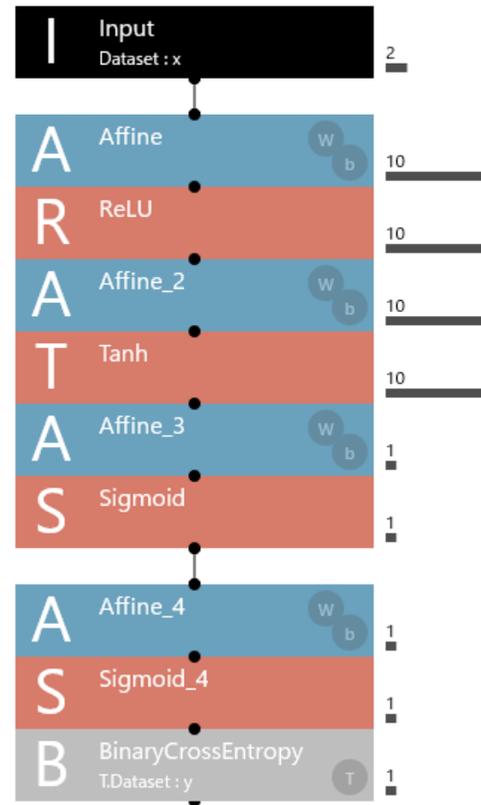
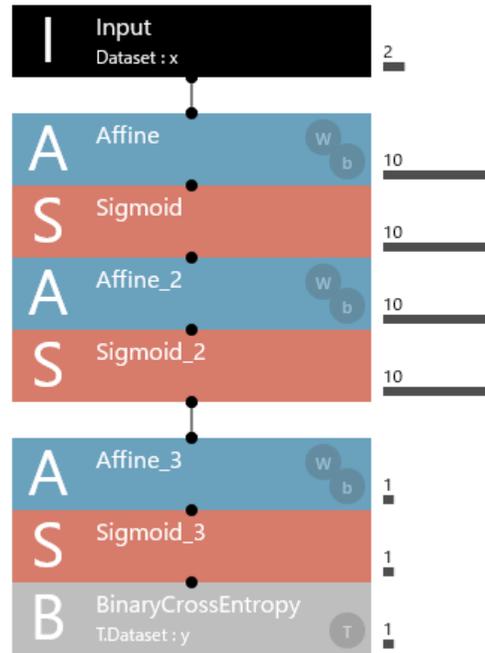
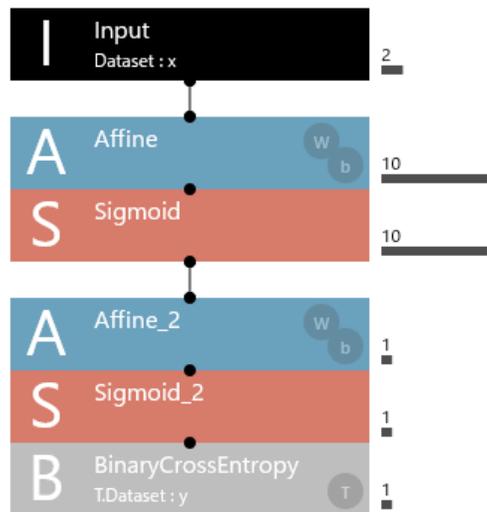


(8)出力層 Affine2のプロパティ  
OutShape→1  
(Sigmoid2のInput,Output  
BinaryCrossEntropyのInput自動変更される)

# (4) ニューラルネットワークを設計(2)

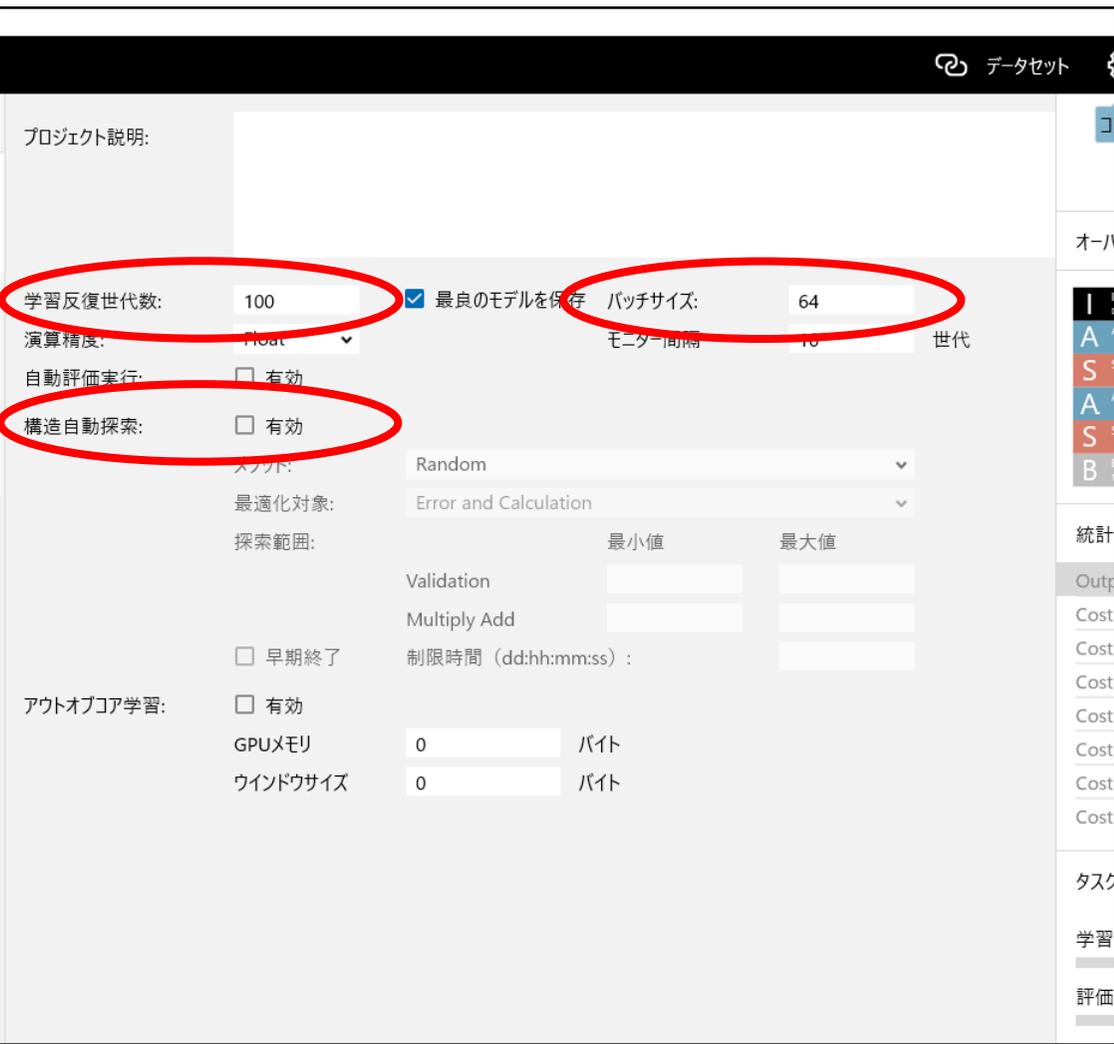
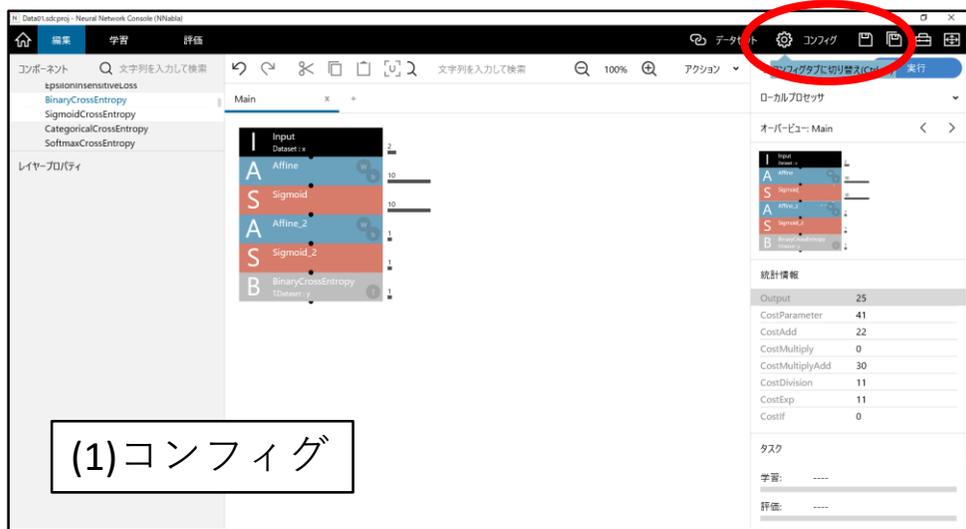
- (1) 学習用データ用意
- (2) 評価用データ用意
- (3) データをセット
- (4) ニューラルネットワークを設計
- (5) 学習に関するパラメータを設定
- (6) 学習
- (7) 評価
- (8) 評価結果

このような設計も可能です



# (5) 学習に関するパラメータ設定

- (1) 学習用データ用意
- (2) 評価用データ用意
- (3) データをセット
- (4) ニューラルネットワークを設計
- (5) 学習に関するパラメータを設定
- (6) 学習
- (7) 評価
- (8) 評価結果



学習反復世代数 (Epoch)

繰り返す学習の回数

バッチサイズ

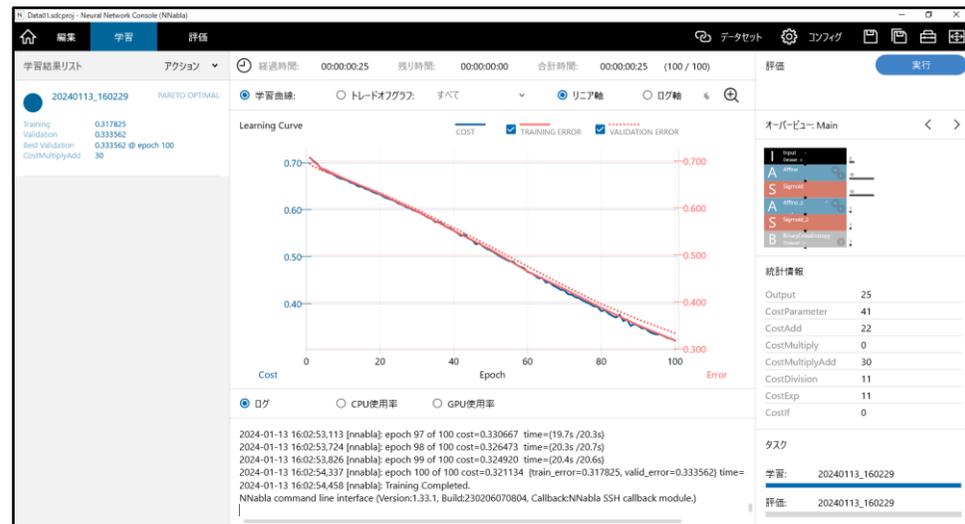
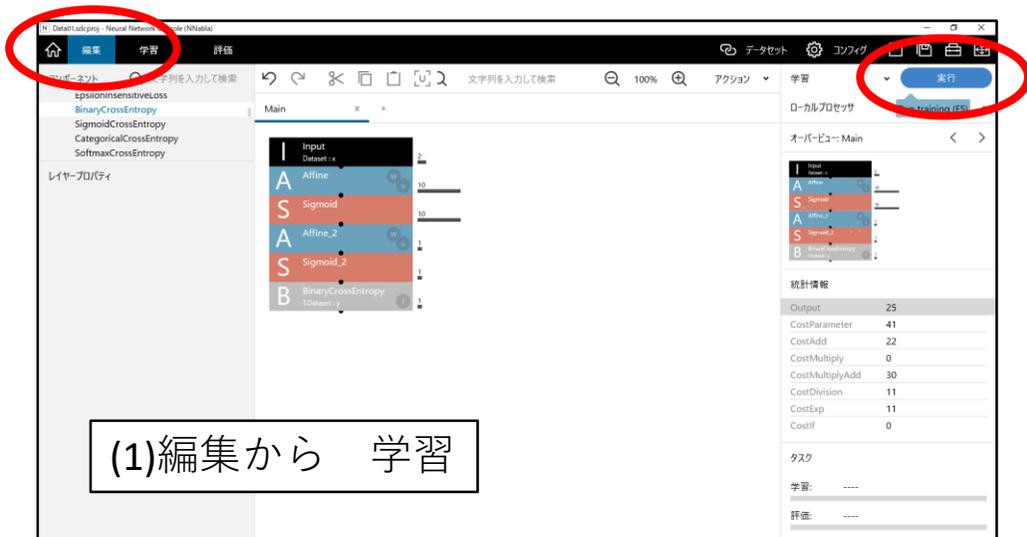
1回の学習で使用するデータ数

構造自動探索

最適な構造の最適化

# (6) 学習

- (1) 学習用データ用意
- (2) 評価用データ用意
- (3) データをセット
- (4) ニューラルネットワークを設計
- (5) 学習に関するパラメータを設定
- (6) 学習
- (7) 評価
- (8) 評価結果



コンフィグの、学習反復世代数 (Epoch) に設定した回数、学習が繰り返される。

学習中の状況は、COST、TRAINING ERROR、VALIDATION ERROR グラフで表示される。

COST → 学習中の損失関数の出力値 (修正の度合い?)

TRAINING ERROR 学習データに対する評価関数の出力値

VALIDATION ERROR 評価データに対する評価関数の出力値

損失関数：機械学習モデルが算出した予測値と実際の正解値のズレを計算するための関数

- (1) 学習用データ用意
- (2) 評価用データ用意
- (3) データをセット
- (4) ニューラルネットワークを設計
- (5) 学習に関するパラメータを設定
- (6) 学習
- (7) 評価
- (8) 評価結果

(1)学習から 評価

Index	x_0X0	x_1X1	yY
1	0.910619	0.595781	1
2	0.513286	0.808183	1
3	0.92935	0.714998	1
4	0.85123	0.977791	1

出力結果

用意した評価用データについて、検証が行われる

出力結果 評価用データ個別の結果を見ることができる

混同行列 評価結果全体を見ることができる

	y=0	y=1	Recall
yY=0	472	0	0.9833
yY=1	0	520	1
Precision	1	0.9848	
F-Measures	0.9915	0.9923	
Accuracy	0.992		
Avg.Precision	0.9924		
Avg.Recall	0.9916		
Avg.F-Measures	0.9919		

混同行列

# (※) 評価結果例

- (1) 学習用データ用意
- (2) 評価用データ用意
- (3) データをセット
- (4) ニューラルネットワークを設計
- (5) 学習に関するパラメータを設定
- (6) 学習
- (7) 評価
- (8) 評価結果

中間層を、10から100に変更

このスクリーンショットは、NNablaのニューラルネットワーク設計画面を示しています。左側の「レイヤープロパティ」では、Affine層の出力サイズが100に設定されています。中央の「Main」タブには、入力層（サイズ100）と2つの中間層（それぞれサイズ100）が配置されています。右側の「統計情報」には、出力サイズ205、コストパラメータ401、コスト追加202、コスト乗算0、コスト乗算追加300、コスト除算101、コスト展開101、コストオフ0が示されています。

Index	x_0x0	x_1x1	yY
1	0.910619	0.595781	1
2	0.513286	0.808183	1
3	0.92935	0.714998	1
4	0.85123	0.977791	1

このスクリーンショットは、学習結果リストと出力結果の表示画面です。学習結果リストには、2つのモデルのトレーニングと検証のコストが示されています。出力結果の表では、モデル1の出力が0.910619と表示されています。

このスクリーンショットは、学習曲線のグラフを表示しています。横軸はEpoch（0から100）、縦軸はCostとError（0.20から0.60）です。赤い線はCost、青い線はErrorを示しています。両方ともEpochが進むにつれて減少傾向にあります。下部には、各Epochでのコストと実行時間（例：epoch 97でコスト0.148293、時間18.3s）が記録されています。

	y=0	y=1	Recall
yY=0	477	3	0.9937
yY=1	2	518	0.9961
Precision	0.9958	0.9942	
F-Measures	0.9947	0.9951	
Accuracy	0.995		
Precision	0.991		
Avg Recall	0.9949		
Avg F-Measures	0.9949		

このスクリーンショットは、評価結果の詳細な指標を表示しています。Accuracyが0.995と非常に高い値を示しています。Precisionは0.991、Recallは0.9937、F-Measuresは0.9947と、全体的に優れた性能を示しています。

## 2クラス、多クラス、回帰について

ここでは、エクセルを使って学習・評価用データを作成します。

さらに、ニューラルネットワークを設定し、学習・評価を行います。

自分が使用したいデータを使って学習させたニューラルネットワークを作ることができるようになります。

まずは、多クラス分類

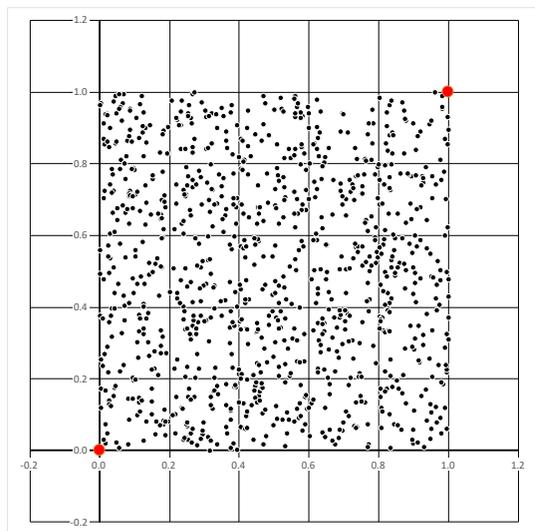
- ①点0(0,0)と点1(1,1)がある
- ②XY座標がそれぞれ、0から1までの乱数で決めた点を1000個学習用として準備
- ③XY座標がそれぞれ、0から1までの乱数で決めた点を1000個評価用として準備
- ④学習用、評価用のそれぞれの点について、点0に近ければクラス0、点1に近ければクラス1とする

エクセルを使って下記のような表を作成  
学習・評価用データを用意する。

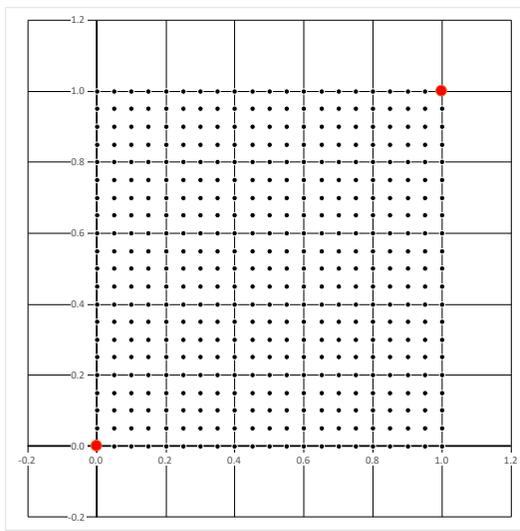
No	X0	X1	P0 (0,0)からの距離	P1 (1.1)からの距離	Y
1	0.284233	0.922186	0.964995	0.719985	1
2	0.586357	0.093303	0.593734	0.996594	0
3	0.834071	0.031644	0.834671	0.982469	0
4	0.477290	0.483275	0.679235	0.735004	0
5	0.872728	0.648696	1.087410	0.373648	1
6	0.434740	0.291877	0.523633	0.906067	0
7	0.962327	0.415521	1.048204	0.585692	1
8	0.332753	0.896369	0.956139	0.675247	1

~

996	0.995733	0.720613	1.229132	0.279420	1
997	0.249328	0.127730	0.280142	1.150810	0
998	0.358500	0.638661	0.732400	0.736267	0
999	0.947588	0.083321	0.951244	0.918176	1
1000	0.874825	0.028503	0.875289	0.979528	0



学習用

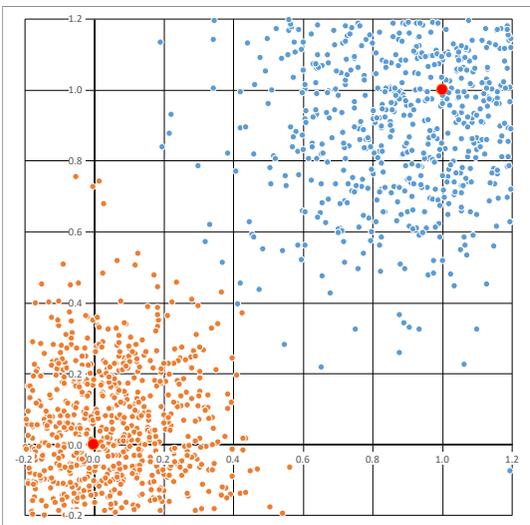


評価用

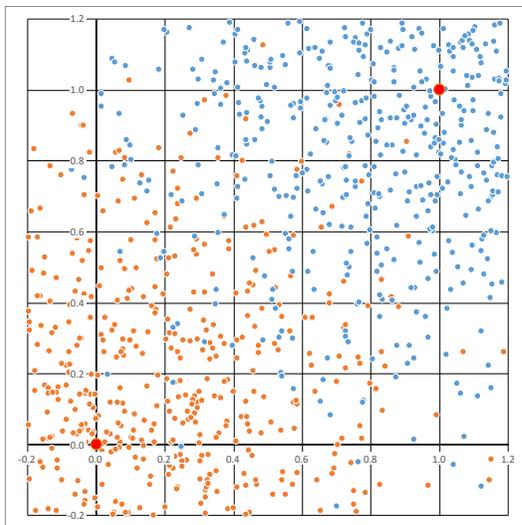
どこに境界がある？



- ①点0(0,0)と点1(1,1)がある
- ②点0が平均、 $\sigma_{x0}^2$ の分散値、 $\sigma_{y0}^2$ である正規乱数の点、同じく平均が点1、 $\sigma_{x1}^2$ 、 $\sigma_{y1}^2$ の分散値を持った点をそれぞれ1000個 (計2000個) 学習用として準備
- ③学習用と同じ条件で、2000個評価用として準備
- ④学習用、評価用のそれぞれの点について、点0に近ければクラス0、点1に近ければクラス1とする



学習用



評価用 (分散値変更など)

エクセルを使って下記のような表を作成  
学習・評価用データを用意する。

平均	0	0				平均	1	1			
分散	0.2	0.2				分散	0.2	0.2			
標準偏差	0.4472136	0.4472136				標準偏差	0.4472136	0.4472136			
			P0	P1					P0	P1	
No	Y0	Y1	(0,0)からの距離	(1,1)からの距離	X0	No	Y0	Y1	(0,0)からの距離	(1,1)からの距離	X0
1	0.531093	0.095365	0.539587	1.018940	0	1	1.363300	1.559851	2.071647	0.667398	1
2	-0.205430	-0.109708	0.232889	1.638448	0	2	0.605847	0.804411	1.007039	0.440013	1
3	0.389040	0.540626	0.666054	0.764393	0	3	1.609150	1.223367	2.021383	0.648811	1
4	-0.455994	0.399736	0.606399	1.574876	0	4	1.595713	0.371566	1.638402	0.865912	1
5	-0.645066	-0.262282	0.696349	2.073547	0	5	0.765689	1.764466	1.923440	0.799569	1
6	0.154854	-0.253490	0.297047	1.511790	0	6	1.600759	0.270159	1.623396	0.945293	1
7	0.900092	-0.240933	0.931780	1.244949	0	7	0.671687	0.710267	0.977569	0.437875	1
8	-0.153039	-0.287773	0.325936	1.728543	0	8	1.547503	0.619238	1.666800	0.666888	1
9	0.388815	0.559189	0.681080	0.753566	0	9	1.444963	0.097545	1.448251	1.006189	1
10	0.550696	-0.282750	0.619043	1.359162	0	10	1.563594	0.720955	1.721802	0.628891	1

~

996	0.271423	-0.102742	0.290218	1.321690	0	996	0.949269	0.191728	0.968438	0.809862	1
997	0.261174	-0.708744	0.755334	1.861631	0	997	1.290646	0.616595	1.430370	0.481118	1
998	0.274528	-0.613279	0.671920	1.768892	0	998	0.251041	1.175394	1.201904	0.769222	1
999	0.512509	0.253784	0.571902	0.891339	0	999	1.525768	0.972613	1.809405	0.526481	1
1000	-0.297724	0.114194	0.318872	1.571222	0	1000	1.116785	0.656162	1.295283	0.363130	1

黄色のセル値を変更すると、パラメータ変更

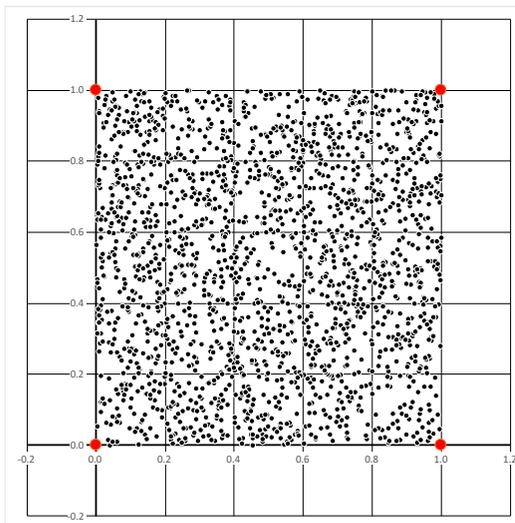
エクセルで正規乱数  
=NORMINV(RAND(), 平均,標準偏差)



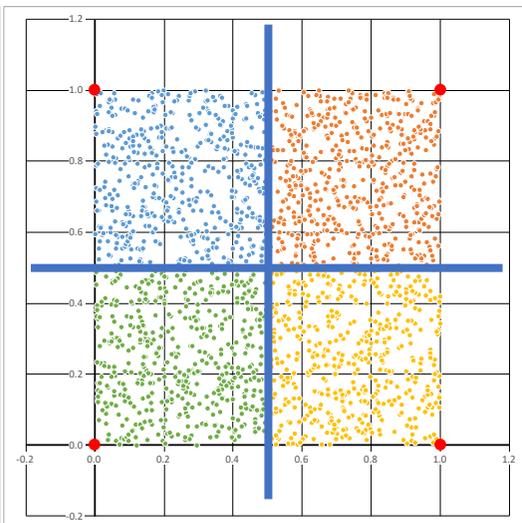
- ①点0(0,0)、点1(0,1)、点2(1,0)、点3(1,1)がある
- ②XY座標がそれぞれ、0から1までの乱数で決めた点を2000個学習用として準備
- ③XY座標がそれぞれ、0から1までの乱数で決めた点を2000個評価用として準備
- ④学習用、評価用のそれぞれの点について、点0に近ければクラス0、点1に近ければクラス1. . . とする

エクセルを使って下記のような表を作成  
学習・評価用データを用意する。

			P0	P1	P2	P3	
No	Y0	Y1	(0,0)からの距離	(0,1)からの距離	(1,0)からの距離	(1,1)からの距離	X0
1	0.455559	0.233051	0.511710	0.892045	0.592224	0.940546	0
2	0.755030	0.204982	0.782361	1.096414	0.319418	0.831903	2
3	0.169890	0.345807	0.385286	0.675893	0.899258	1.056907	0
4	0.489618	0.633737	0.800842	0.611453	0.813703	0.628202	1
5	0.707509	0.509751	0.872018	0.860763	0.587705	0.570872	3
6	0.909312	0.165143	0.924186	1.234437	0.188405	0.839769	2
7	0.901939	0.899887	1.274084	0.907478	0.905214	0.140138	3
8	0.200769	0.141807	0.245800	0.881364	0.811714	1.172717	0
9	0.725175	0.769890	1.057643	0.760808	0.817472	0.358440	3
10	0.431357	0.041970	0.433394	1.050661	0.570190	1.114081	0



学習用



評価用

どこに境界がある？

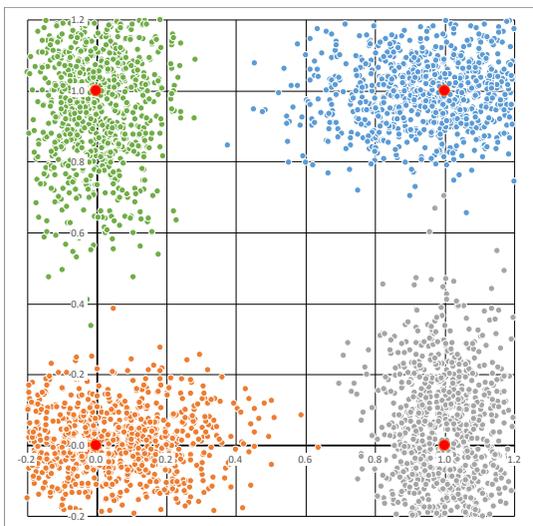
~

1996	0.306356	0.671706	0.738270	0.449034	0.965572	0.767410	1
1997	0.898943	0.959050	1.314487	0.899876	0.964360	0.109038	3
1998	0.053335	0.063816	0.083169	0.937702	0.948814	1.331396	0
1999	0.399475	0.818132	0.910450	0.438926	1.014874	0.627460	1
2000	0.051200	0.449360	0.452267	0.553015	1.049831	1.097008	0

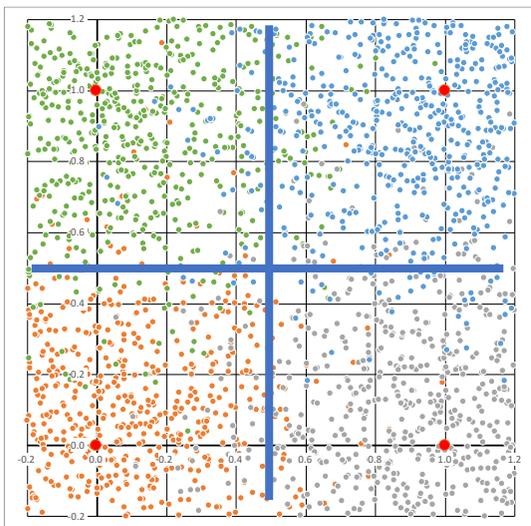


- ①点0(0,0)、点1(0,1)、点2(1,0)、点3(1,1)がある
- ②点0が平均、 $x$ 軸方向に $\sigma_{x0}^2$ の分散値、 $y$ 軸方向には $\sigma_{y0}^2$ である正規乱数の点、同じく平均が点1...、平均が点2...、平均が点3...、それぞれ1000個 (計4000個) を学習用として準備
- ③学習用と同じように、それぞれ1000個 (計4000個) を評価用として準備
- ④学習用、評価用のそれぞれの点について、点0に近ければクラス0、点1に近ければクラス1...とする

エクセルを使って下記のような表を作成  
学習・評価用データを用意する。

学習用



評価用

平均	0	0					
分散	0.04	0.01					
標準偏差	0.2	0.1					
			P0	P1	P2	P3	
No	X0	X1	(0,0)距離	(0,1)距離	(1,0)距離	(1,1)距離	Y0
1	-0.178717	-0.104540	0.207047	1.118905	1.183343	1.615358	0
2	0.316099	0.093150	0.329539	0.960362	0.690215	1.135824	0
3	-0.226513	-0.220059	0.315807	1.240908	1.246097	1.729993	0
4	0.126254	0.042956	0.133361	0.965336	0.874802	1.295904	0
5	0.242772	0.277488	0.368697	0.762208	0.806470	1.046622	0
6	0.309860	0.041578	0.312637	1.007266	0.691391	1.181044	0
7	-0.212977	-0.080247	0.227593	1.101041	1.215629	1.624268	0
8	0.153458	0.002279	0.153475	1.009454	0.846545	1.308465	0
9	0.018746	-0.105309	0.106964	1.105468	0.986889	1.478028	0
10	-0.094653	0.098273	0.136443	0.906682	1.099055	1.418230	0



Data04.xlsx

$y=ax+b$ の1次関数において、 $a=\pm 5$ 、 $b=\pm 5$ とする。

$x_0, x_1, x_2$ が $\pm 1$ のランダムな値で与えられた時

それぞれ、 $y_0, y_1, y_2$ を求める。

$(x_0, y_0)$ 、 $(x_1, y_1)$ 、 $(x_2, y_2)$ を説明変数とし、

$a, b$ の目的変数を推定する。

①  $(x_0, y_0)$ 、 $(x_1, y_1)$ 、 $(x_2, y_2)$ の3点の座標値6個と、

$a$ 、 $b$ の値を、学習用に1000個準備

②  $(x_0, y_0)$ 、 $(x_1, y_1)$ 、 $(x_2, y_2)$ の3点の座標値6個と、

$a$ 、 $b$ の値を、評価用に1000個準備

エクセルを使って下記のような表を作成

学習・評価用データを用意する。

	X0	Y0	X1	Y1	X2	Y2	←	a	b
No	x0	x1	x2	x3	x4	x5		Y0	Y1
1	0.969741	-5.956526	0.327607	-4.010612	0.866873	-5.644796		-3.030385	-3.017836
2	-0.902149	1.776191	0.959186	0.733585	0.426586	1.031915		-0.560139	1.270862
3	0.471477	-0.996831	-0.914704	1.411219	-0.459323	0.620140		-1.737183	-0.177789
4	0.929604	4.496216	-0.415188	-0.209497	-0.145111	0.735559		3.499213	1.243334
5	0.649926	-3.500658	-0.485182	-2.900207	0.253698	-3.291061		-0.528982	-3.156859
6	0.739758	-5.037449	0.306922	-3.060251	-0.220338	-0.651730		-4.567999	-1.658234
7	-0.782905	-0.424841	-0.300316	1.333942	0.224689	3.247306		3.644470	2.428435
8	-0.377123	-5.002918	-0.909198	-5.538927	0.393528	-4.226570		1.007392	-4.623008
9	0.844400	-2.807314	0.236512	-1.957997	-0.633601	-0.742309		-1.397161	-1.627551
10	0.733105	1.462175	-0.308665	1.782269	-0.726755	1.910732		-0.307260	1.687429



996	0.399837	5.828254	0.366514	5.717706	-0.470027	2.942524		3.317451	4.501815
997	-0.602688	-0.865920	-0.967022	-1.965954	-0.668542	-1.064753		3.019305	0.953780
998	0.028763	-4.335116	-0.830944	-1.144855	0.764224	-7.064321		-3.710873	-4.228382
999	-0.829553	-2.597792	0.713584	-6.117668	0.018851	-4.532990		-2.280988	-4.489992
1000	-0.274734	0.532216	-0.138108	1.020782	0.287450	2.542554		3.575943	1.514650



正弦波をサンプリング周波数1000[Hz]、32点観測した。

エクセルを使って下記のような表を作成

FFTで分析すると、500[Hz]までの周波数で

学習・評価用データを用意する。

分解能は、 $1000/32=31.25$ [Hz]となる。

$$y = a_1 \times \sin(2\pi f_1 t + \theta_1) + a_2 \times \sin(2\pi f_2 t + \theta_2)$$

振幅 $a_1, a_2 = 0.1 \sim 1.0$ 、周波数 $f_1, f_2 = 50 \sim 450$ 「Hz」

位相 $\theta_1, \theta_2 = 0 \sim 1$ [s] で与えられる

正弦波の値32個が説明変数として与えられる

周波数と位相を目的変数として推定する

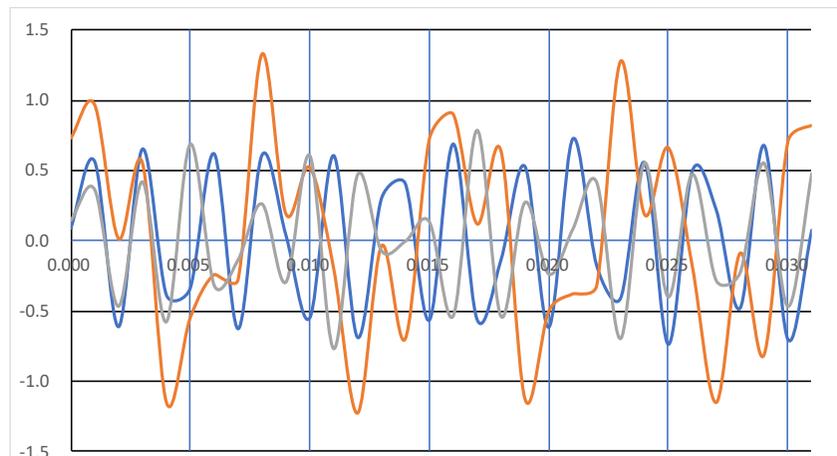
y0_:		y1_:		y2_:		y3_:		x0_:					x27_:										
No	振幅	周波数[Hz]	位相[s]	振幅	周波数[Hz]	位相[s]	No	1	2	3	4	5	時間[s]	0	0.001	0.002	0.003	0.004	28	29	30	31	32
1	0.321121	106.855388	0.968716	0.286778	234.118283	0.016161								0.2693	0.6061	0.2893	-0.225	-0.265	0.335	0.1531	0.0442	0.2945	0.3529
2	0.927878	378.244879	0.021744	0.354260	320.039926	0.145153								0.0714	0.9232	-1.229	0.6709	0.2389	0.6029	-0.553	0.1747	0.4985	-1.041
3	0.130410	188.454337	0.408413	0.582434	371.976654	0.995123								0.5404	0.02	-0.289	0.4804	-0.581	0.6629	-0.151	-0.219	0.4027	-0.613
4	0.200129	155.127029	0.207028	0.591453	372.776042	0.858494								0.4888	0.1502	-0.232	0.5933	-0.59	0.7612	-0.126	-0.232	0.3508	-0.721
996	0.331880	205.778231	0.089197	0.686541	265.123701	0.570468								0.4003	0.866	-0.324	-0.694	0.2755	0.5448	-0.389	-0.713	0.4961	0.8451
997	0.734143	139.618632	0.040899	0.765389	239.798863	0.820028								0.5896	1.1402	0.2274	-0.288	0.1174	-1.194	-1.004	0.6389	1.374	0.3319
998	0.136051	310.604815	0.145952	0.668280	444.359739	0.352087								0.2502	0.1159	-0.335	0.3909	-0.443	0.2946	-0.126	-0.21	0.5504	-0.694
999	0.682785	340.058347	0.282014	0.742098	153.760218	0.457039								0.5175	1.1857	-0.167	0.114	-0.303	1.4077	0.2375	-0.525	0.0006	-0.982
1000	0.691679	384.261427	0.411669	0.959414	409.375575	0.741482								0.9247	0.0504	-0.969	1.4676	-1.374	1.0964	-1.082	0.6081	0.1846	-0.992

①正弦波の値32個とその時の、

振幅 $a_1, a_2$ 、周波数 $f_1, f_2$ が学習用に1000個準備

②正弦波の値32個とその時の、

振幅 $a_1, a_2$ 、周波数 $f_1, f_2$ が評価用に1000個準備



(1) irisのデータセットを使って、ニューラルネットを作ってみましょう。

# ニューラルネットワークでは何が行われているか

ニューラルネットワークを使い、2クラス分類、多クラス分類、回帰の3つの機能を試してみました。ここでは簡単なネットワークを例に、学習がどのように行われているのかを調べてみたいと思います。

# (1) 学習用データを用意

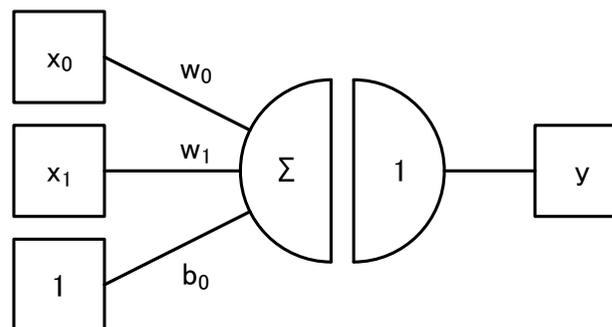
ニューラルネットワークの回帰モデルを使って、どのように学習が行われているのかを調べる。

$$y = 2 \cdot x_0 + 5 \cdot x_1 + 9$$

- (1)  $x_0, x_1$ を乱数で作る → 説明変数
- (2)  $y$ を計算する → 目的変数
- (3) 学習用に1000個のデータを用意する。
- (4) 評価用に1000個のデータを用意する。
- (5) 学習結果を出力し、学習の仕組みを調べる。

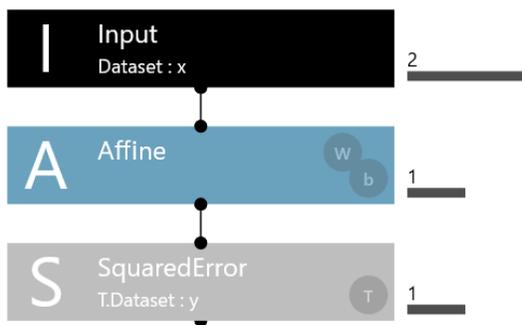
	A	B	C	D	E	F
1						
2		No	X0	X1	Y0	
3		1	81.274967	74.670179	544.900830	
4		2	72.027728	94.13590	623.73406	
5		3	8.084317	10.3989	18.38578	
6			=+100*RAND()	=+100*RAND()	=2*C4+5*D4+9	
7						
8		6	24.150158	93.200270	523.301667	

このようなネットワークを考える

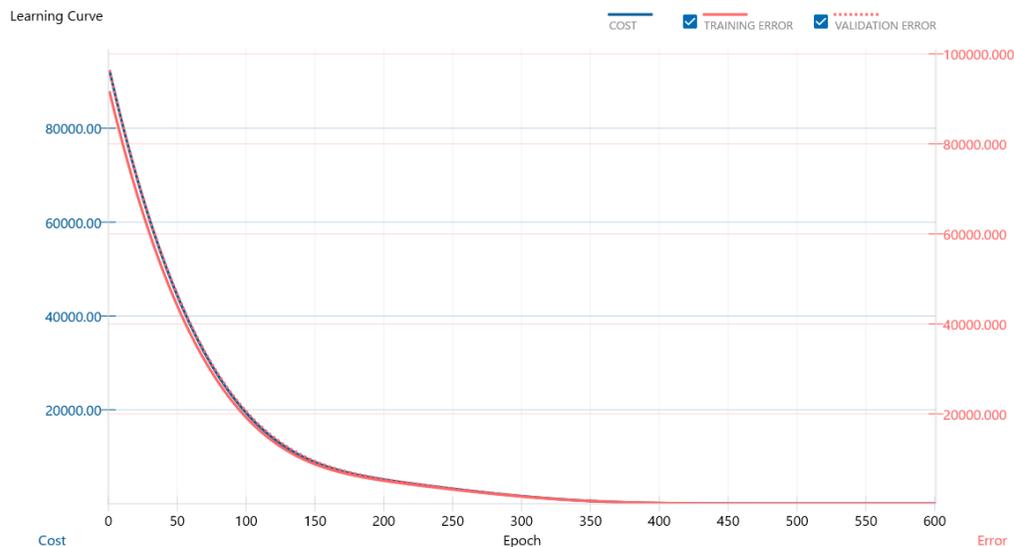


学習結果の出力は、ニューラルネットワーク内の学習後の完成したAIのパラメータを得ることになる。  
エッジコンピューティングでAI機能を使用するとき必要となる。

結果など



学習反復世代数:   最良のモデルを保存    バッチサイズ:   
演算精度:     モニター間隔:  世代

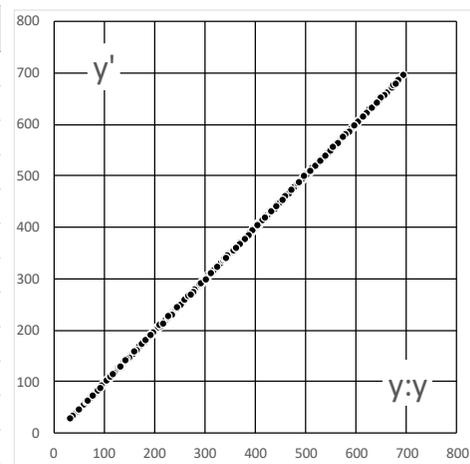


評価の出力結果をcsvファイルに保存できる

Index	x_0x0	x_1x1	y_0y	y
1	74.536703	13.535411	225.750459	224.4615
2	5.053109	37.364313	205.927782	202.9945
3	48.945444	53.34487	373.61524	373.00696
4	20.377508	10.091433	100.21218	96.72897
5	55.327475	63.774291	438.526404	438.59213

評価の結果と相関グラフ

	A	B	C	D
1	x_0:x0	x_1:x1	y_0:y	y'
2	74.536703	13.535411	225.750459	224.4615
3	5.053109	37.364313	205.927782	202.9945
4	48.945444	53.34487	373.61524	373.00696
5	20.377508	10.091433	100.21218	96.72897
6	55.327475	63.774291	438.526404	438.59213
7	67.394199	98.397235	635.774571	637.7324
8	7.377428	14.314405	95.326881	91.526276
9	29.188538	96.460257	549.678361	550.10876
10	52.289157	27.32285	250.192565	248.63235
11	6.153403	75.068214	396.647875	395.31906
12	18.060926	39.254011	241.391907	239.02968



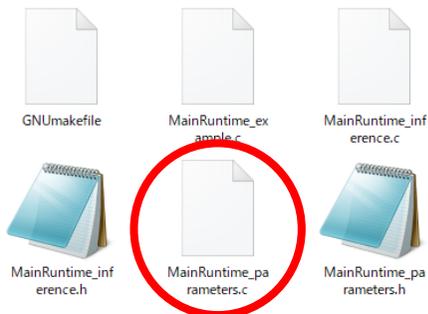
学習後の、ネットワークモデルの中を、C言語の形式で出力することができます。

The screenshot shows the NNabla Neural Network Console interface. The top navigation bar includes 'Home', 'Edit', 'Learn', and 'Evaluate'. Below it, the 'Learn Results List' shows a training session for '20240114\_204842' with 'PARETO OPTIMAL' results. A table displays training metrics: Training (3.139062), Validation (2.982089), Best Validation (2.982089 @ epoch 60), and CostMultiplyAdd (2). A context menu is open over the table, listing actions like 'Export' (NNP, NNB, ONNX, saved\_model, tflite) and 'Delete'. The table has columns for 'x\_1x1x1', 'y\_0-y', and 'y'.

	x_1x1x1	y_0-y	y
03	13.535411	225.750459	224.4615
		205.927782	202.9945
		373.61524	373.00696
08	10.091433	100.21218	96.72897
5	55.327475	63.774291	438.526404
		438.526404	438.59213

c\_source という  
フォルダが作られる

プログラムに使用する  
ファイルが出力される

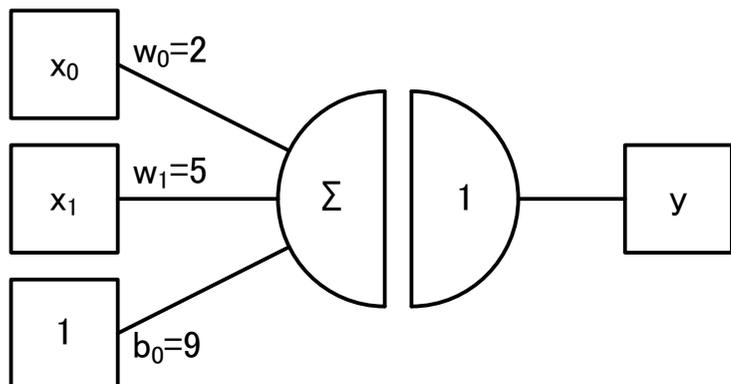


ニューラルネットワークのモデルの中の数値は  
MainRuntime\_parameters.c の中に記述されている

```
// Affine/affine/W
float MainRuntime_parameter1[] = {
    2.03787899017334,
    5.04144811630249,
};

// Affine/affine/b
float MainRuntime_parameter2[] = {
    4.326648235321045,
};
```

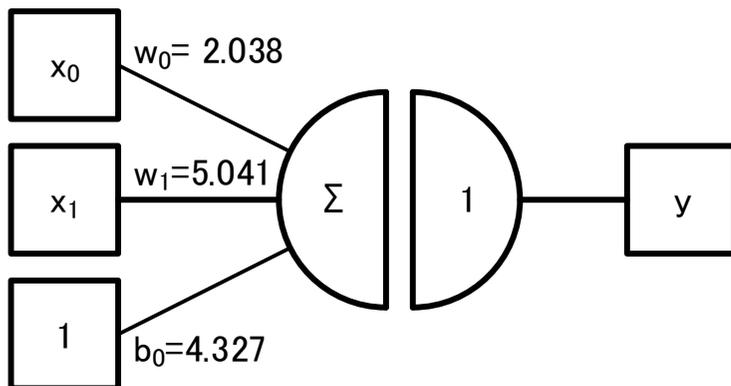
これが正解のモデル



2つの入力の重み、 $w_0, w_1$ はほぼ正しそうである。

バイアスは、正しくは9.0、モデルでは4.3であった。

学習結果では以下のモデル



# 画像データの使い方

ニューラルネットワークを使い、2クラス分類、多クラス分類、回帰の3つの機能を試してみました。ここでは簡単なネットワークを例に、学習がどのように行われているのかを調べてみたいと思います。

数値データは、エクセルなどを使用しCSVファイルを作成し、NeuralNetworkConsoleで利用できる。

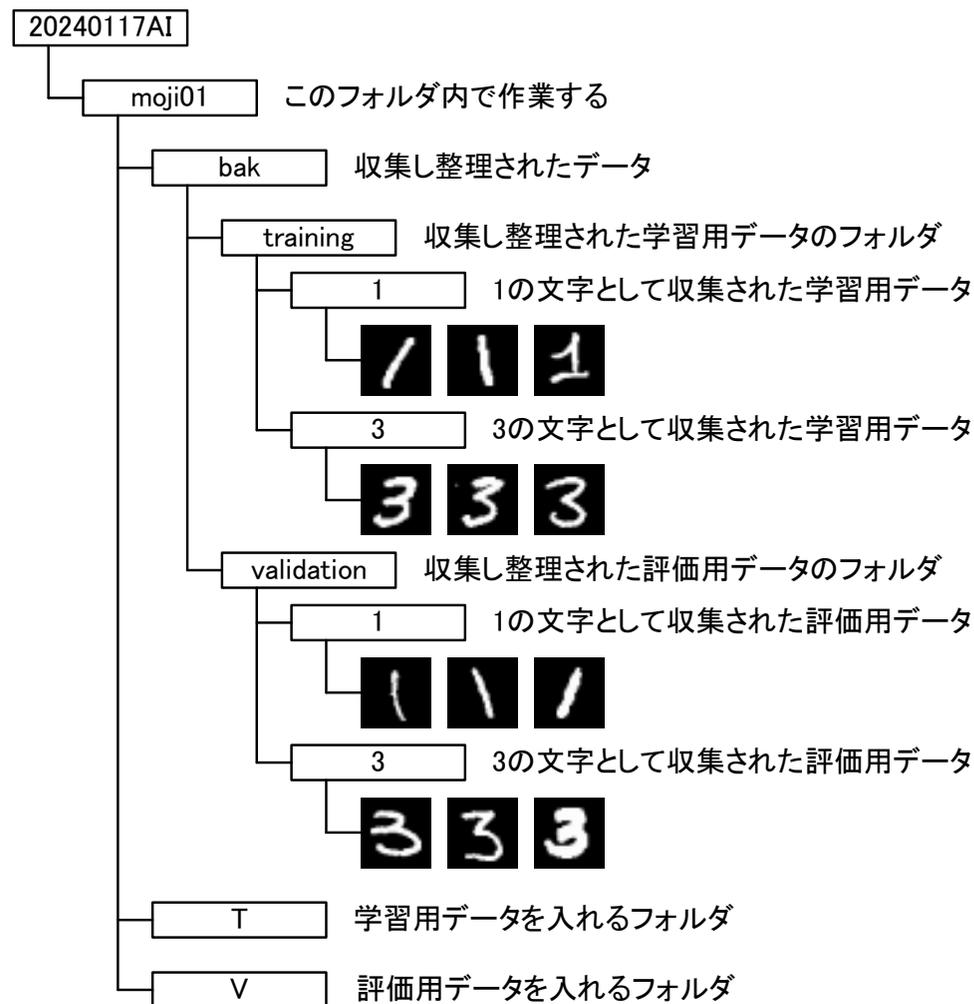
## 画像データを使用する方法

### 準備

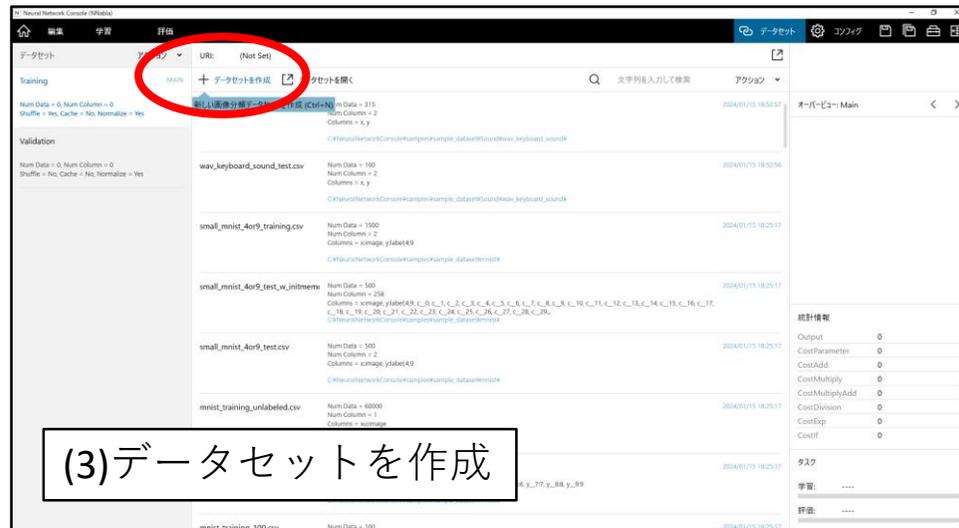
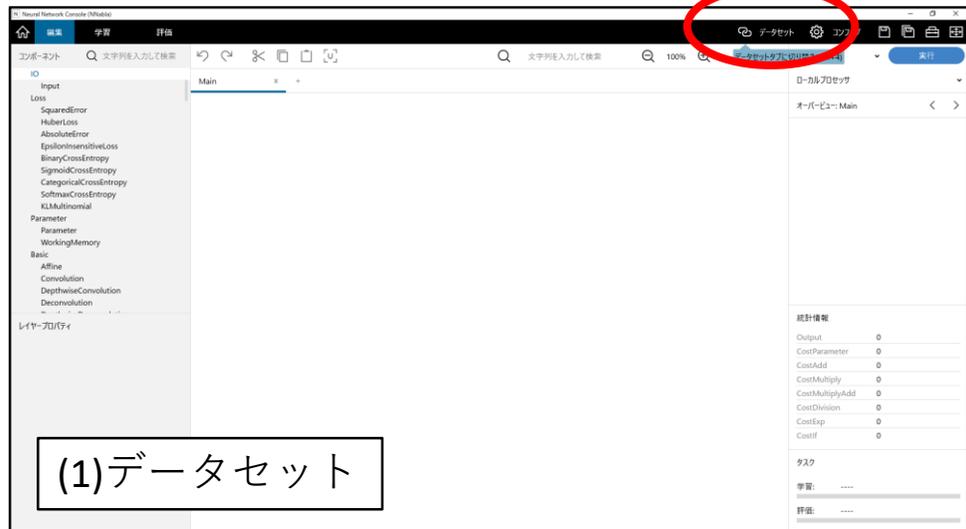
- ①学習用データのフォルダを作る。
- ②そのフォルダ内に、各クラスのフォルダを作る。
- ③各クラスのフォルダ内に画像ファイルを保存する。
- ④評価用も同様。各クラスのフォルダ名は同じ。

NeuralNetworkConsoleには、画像データを学習や評価用に整理してくれるツールがある。

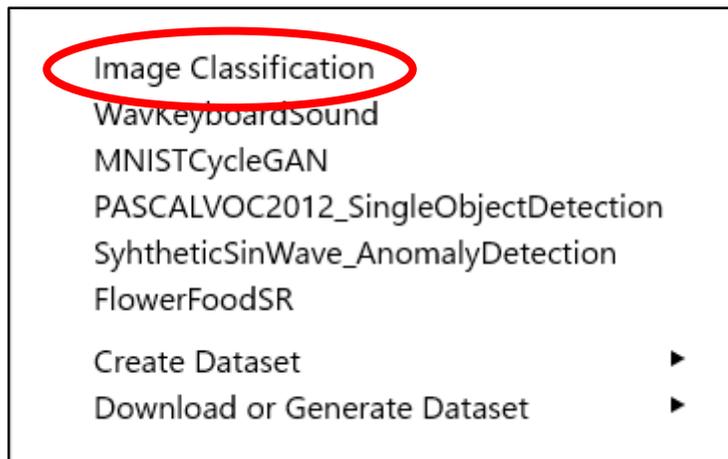
フォルダの構造（以下の構造を作っている）



## ツールの起動



## データセット作成用のツール



データセットをダウンロードするもの  
すでにあるデータを、

データセットの形式に加工してくれるもの  
などがある

画像のデータセット作成をサポートしてくれる

ツールを使用する → Image Classification  
(画像分類)



ソースディレクトリ

→収集し整理された学習用データのフォルダ

出力ディレクトリ

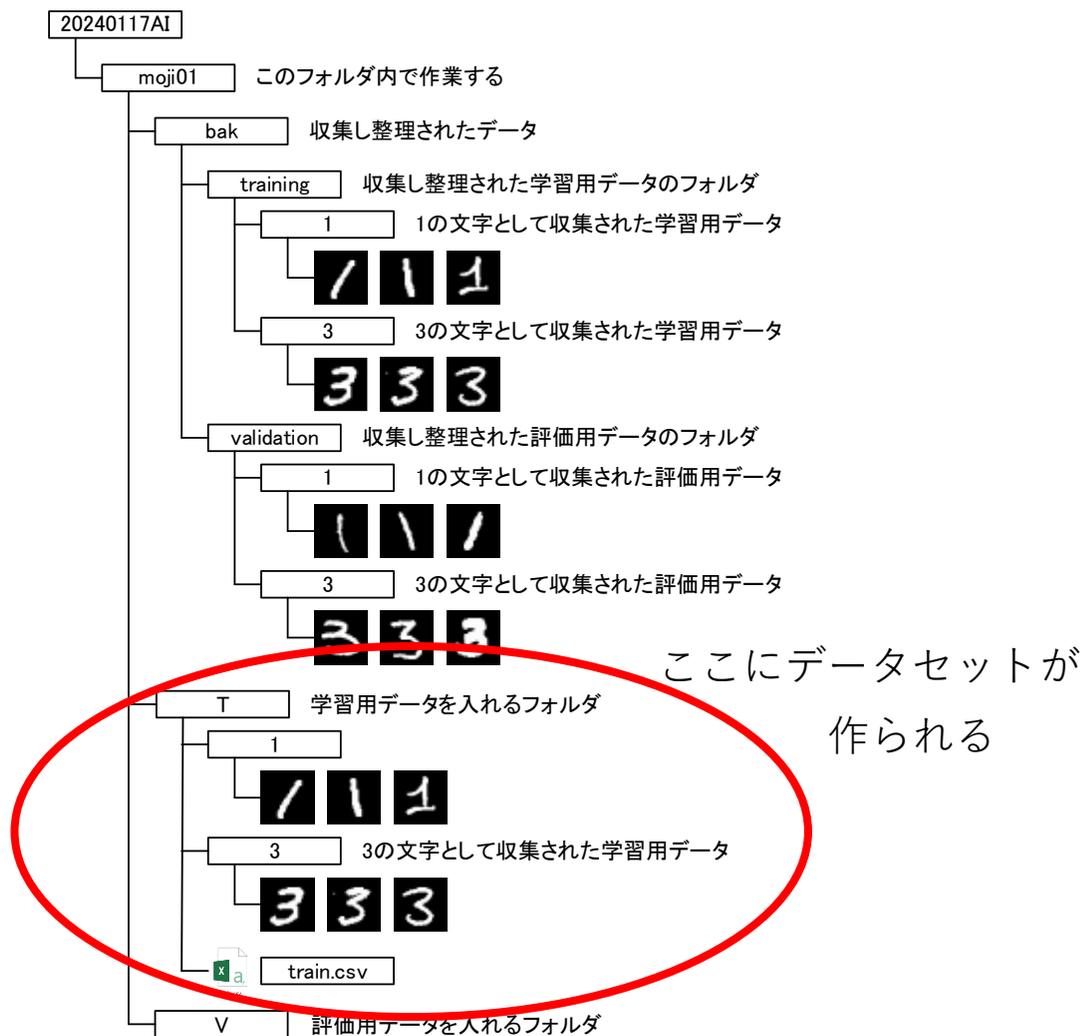
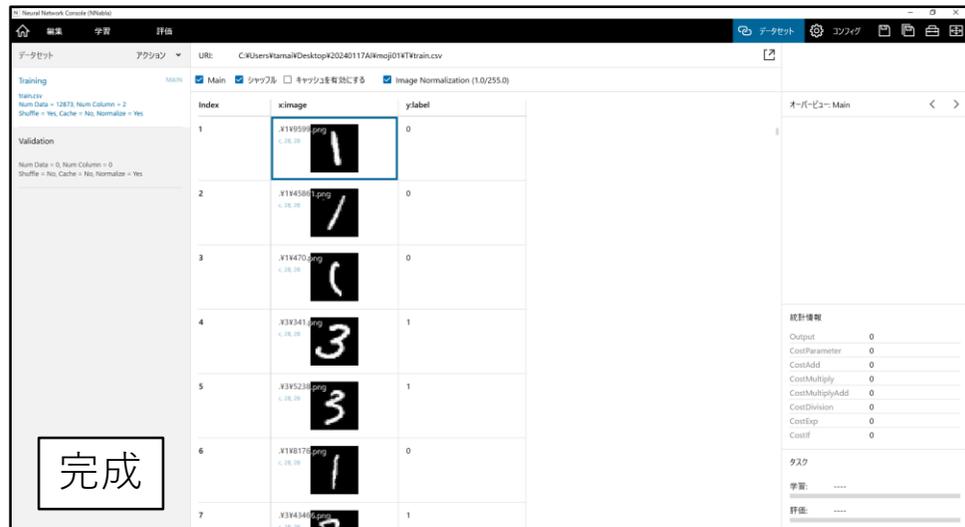
→学習用データを入れるフォルダ



適用を押すと開始される



画像サイズを変更するなどの機能がある



ここにデータセットが  
作られる

同様に、フォルダVに評価用のデータセットを作る

## (5) 学習用の画像データ

Image Classificationを使って、学習用、評価用のデータセットを作ることができました。

新しいプロジェクトの中で、2クラス分類のニューラルネットワークを設計し、1と3の文字を分類してみましょう。

# データセットの入手

ニューラルネットを試してみたり、動作の確認をするためには、自らデータを集めることも必要ですが、まずは、すでに用意してくれているデータセットを使ってみる方法もあります。

データセットを調べると、どのような形式のデータを集めればよいかがよくわかります。

高性能なニューラルネットワークには  
多くの、学習させるためのデータが必要

しかし

1万枚の画像などを個人で収集するのは難しい  
収集画像を分類する（正解を付ける）のは難しい

公的機関、大学、サイトが公開

## (2)大学が公開しているデータセット



## (1)政府統計の総合窓口（e-Stat）



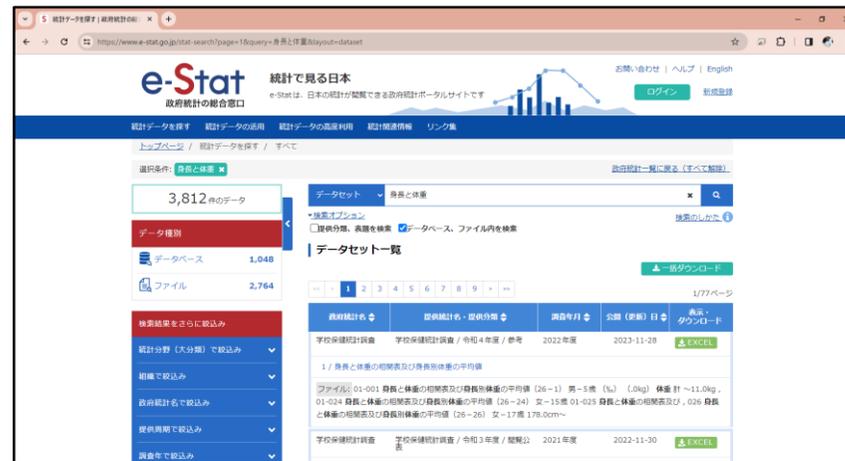
## (3)Kaggle



## (1)身長と体重のデータを入手してみる



## (3)条件のデータセット一覧



## (2)提供分類、表題を検索を外す



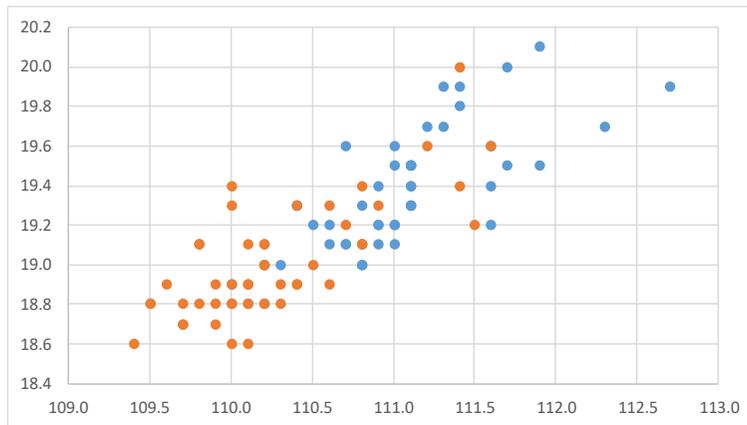
r4\_hoken\_sanko\_01.xlsx - Excel

1身長と体重の相関表及び身長別体重の平均値 (26-1)

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
1	1身長と体重の相関表及び身長別体重の平均値 (26-1)																				
2	男 5歳																				
3		体重	体重	体重	体重	体重	体重	体重	体重	体重	体重	体重	体重	体重	体重	体重	体重	体重	体重	体重	体重
4	計	~11.0kg	12.0kg	13.0kg	14.0kg	15.0kg	16.0kg	17.0kg	18.0kg	19.0kg	20.0kg	21.0kg	22.0kg	23.0kg	24.0kg	25.0kg	26.0kg	27.0kg	28.0kg	29.0kg	30.0kg
5	身長計	1000.0	0.2	-0.1	1.5	8.4	29.8	79.1	132.5	167.0	171.0	142.1	99.0	62.9	37.6	-	-	-	-	-	-
6	身長 ~93.0cm	0.2	-	0.0	-	0.1	0.0	-	-	-	-	-	-	-	-	-	-	-	-	-	-
7	身長 94.0cm	0.1	-	-	0.0	-	0.0	0.0	-	-	-	-	-	-	-	-	-	-	-	-	-
8	身長 95.0cm	0.4	-	-	0.1	0.1	0.1	-	-	0.1	-	-	-	-	-	-	-	-	-	-	-
9	身長 96.0cm	0.5	-	-	0.0	0.1	0.2	-	0.2	-	-	-	-	-	-	-	-	-	-	-	-
10	身長 97.0cm	0.8	-	-	0.1	0.3	0.2	0.1	0.0	-	-	-	-	-	-	-	-	-	-	-	-
11	身長 98.0cm	1.7	-	-	0.0	1.0	0.4	0.1	-	-	-	-	-	-	-	-	-	-	-	-	-
12	身長 99.0cm	2.9	-	-	0.2	0.7	0.8	0.6	0.3	-	0.0	-	-	-	-	-	-	-	-	-	-
13	身長 100.0cm	6.0	-	-	0.0	1.1	2.3	1.6	0.6	0.1	0.1	-	-	-	-	-	-	-	-	-	-
14	身長 101.0cm	7.4	-	-	0.1	1.2	2.3	1.6	0.6	0.1	0.1	-	-	-	-	-	-	-	-	-	-
15	身長 102.0cm	14.5	-	-	0.2	0.9	3.5	5.5	3.1	1.0	0.2	0.0	0.0	-	-	-	-	-	-	-	-
16	身長 103.0cm	20.6	-	-	0.0	0.7	4.9	7.4	4.8	2.1	0.9	0.0	-	-	-	-	-	-	-	-	-
17	身長 104.0cm	29.2	-	-	0.3	4.7	9.2	9.1	3.9	1.4	0.3	0.0	0.0	0.0	-	-	-	-	-	-	-
18	身長 105.0cm	40.1	-	-	1.1	3.5	9.6	14.0	7.7	3.3	0.7	0.2	0.1	-	-	-	-	-	-	-	-
19	身長 106.0cm	47.7	-	-	0.3	3.0	10.4	15.1	11.5	4.8	1.8	0.3	0.4	0.1	-	-	-	-	-	-	-
20	身長 107.0cm	64.9	-	-	0.0	1.6	11.8	19.0	18.7	8.1	3.7	1.3	0.4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
21	身長 108.0cm	70.9	-	-	0.1	1.4	8.8	17.5	21.1	13.7	5.6	1.9	0.6	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0
22	身長 109.0cm	79.2	-	-	0.0	0.6	5.7	16.8	22.3	19.2	9.0	3.6	1.1	0.3	0.3	0.1	0.0	0.0	0.0	0.0	0.0
23	身長 110.0cm	77.8	-	-	0.0	0.2	3.0	11.6	21.3	21.1	12.4	4.8	2.0	0.7	0.2	0.2	0.1	0.0	0.0	0.1	0.0
24	身長 111.0cm	76.6	-	-	0.0	0.1	1.3	7.9	18.9	21.2	14.9	7.8	2.6	1.0	0.5	0.4	0.1	0.2	0.2	0.1	0.2
25	身長 112.0cm	80.5	-	-	0.1	1.0	5.9	14.5	23.3	17.7	10.6	3.9	1.6	0.9	0.5	0.2	0.2	0.1	0.2	0.1	0.2

(1) 都道府県別、身長と体重のデータを入手

(3) 5歳・男子・身長と体重、女子・身長と体重

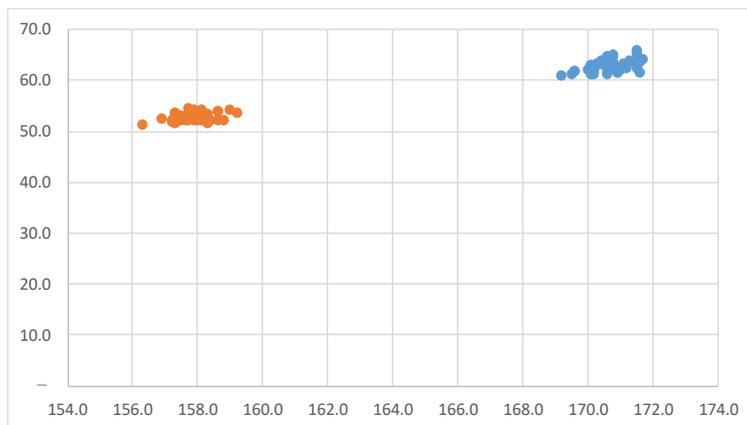


分離  
しにくそう

(2) ダウンロード

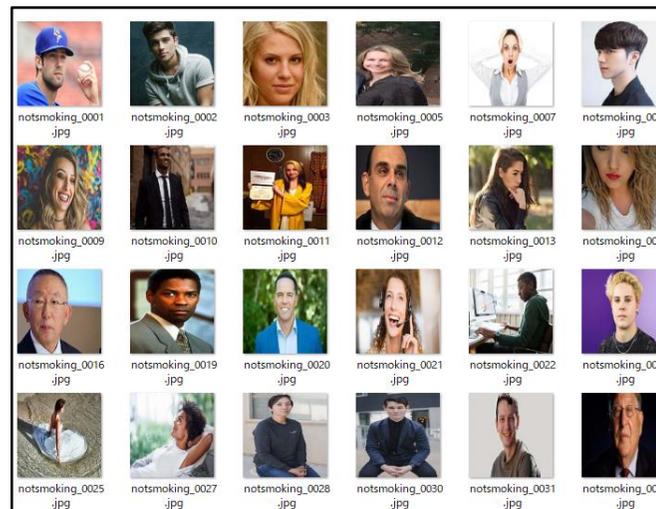
(4) 17歳・男子・身長と体重、女子・身長と体重

区分	身長 (cm)	標準偏差	平均値	標準偏差	身長 (cm)	標準偏差	平均値	標準偏差
全国	111.1	4.88	19.3	2.85	110.2	4.84	19.0	2.75
北海道	111.1	5.05	19.5	3.22	110.9	5.33	19.3	3.17
青森県	111.9	5.29	20.1	3.26	111.4	4.36	20.0	3.56
岩手県	111.7	4.90	20.0	3.73	111.2	4.60	19.6	2.87
宮城県	111.4	5.02	19.8	3.30	110.4	4.91	19.3	3.03
秋田県	111.6	4.76	19.6	3.11	111.6	5.30	19.6	3.40
山形県	111.1	4.94	19.5	2.90	110.8	4.97	19.4	2.92
福島県	110.7	4.63	19.6	3.01	110.0	4.46	19.4	2.86
茨城県	111.3	5.06	19.7	3.36	110.2	4.65	19.1	2.99
栃木県	111.0	5.19	19.6	3.20	110.6	5.35	19.3	3.06
群馬県	111.4	4.59	19.9	3.33	110.2	4.68	19.0	2.71
埼玉県	110.8	4.87	19.3	2.79	110.0	4.77	18.9	2.56
千葉県	110.9	4.77	19.4	2.77	109.9	4.90	18.9	2.71
東京都	111.6	4.75	19.4	2.78	110.5	4.89	19.0	2.80
神奈川県	110.9	4.92	19.2	2.53	109.9	4.60	18.8	2.55
新潟県	112.3	4.77	19.7	2.63	110.8	5.04	19.1	2.85



分離  
しやすそう

## (1) Kaggleから、喫煙・非喫煙者の顔写真を入手



非喫煙者



喫煙者

画像データは、解像度、縦横比率など、同じ条件のものでないと扱えない。

身長と体重のデータは、数値データ 2 個

iPhone 15の写真だと、

1179 × 2556 = 301.3524万のデータ 大きすぎ？

その他のデータセット