Neural Network Console クラウド版
スターターガイド
–表データ分類編–

ソニーネットワークコミュニケーションズ株式会社

概要

本ドキュメントではNeural Network Console(NNC)を用いて、エクセルやCSVファイルなど表として整理されたデータをカテゴリ分けするための分類モデルを作成する一連の流れをまとめました。

表データ分類は、マーケティング業務においてはサービス退会予測や顧客属性分類、コールセンター業務のコール量予測、金融業では債務不履行予測など様々な利用シーンがあります。

表データは皆様の周りにもあふれていると思いますので、まずはお手持ちのデータをもとに、Deep Learningのモデル作成をお試しください。

目次

1 Deep Learningとは

2 アカウントサインイン

3 データ前処理

4 データアップロード

5 分類モデル作成

6 分類モデル利用

Deep Learningのモデルを作るとは

Deep Learningのモデルとは、分類や予測などを行うためのアルゴリズムで、ネットワークとパラメータに分解できます。よくDeep Learningは脳の神経構造に例えられますが、ネットワークとは回路図で、パラメータとはその上の抵抗値のようなものです。

モデル作成とは、目的に合わせたネットワークを構築し、準備したデータセットを用いてパラメータを最適化する作業です。データセットによるパラメータの最適化を学習と呼び、学習をしてできたモデルを学習済みモデルと呼びます。

入力データに対してカテゴリを予測するモデルを分類モデルと言い、数値を予測するモデルを回帰モデルといいます。Deep Learningの基本的なモデルはこの２種類になります。

Deep Learningの学習

目次

1 Deep Learningとは

2 アカウントサインイン

3 データ前処理

4 データアップロード

5 分類モデル作成

6 分類モデル利用

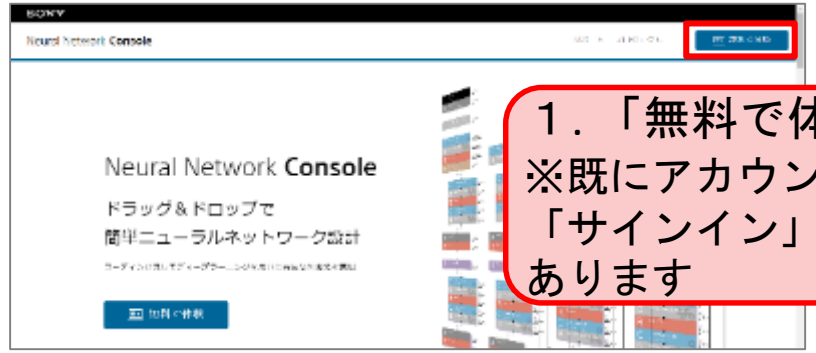
サインインページへの移動

Chromeを利用して、https://dl.sony.com/jaに移動します。

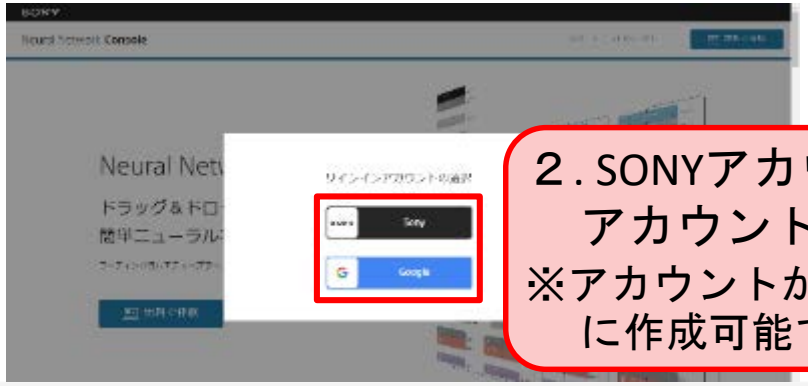
ページの右上にある「無料で体験」をクリックします。

サインインするためのアカウントをSONYアカウントまたはGoogleアカウントから選択します。どちらを選択してもこのドキュメントの内容は進めることが可能です。

サインインページへの移動方法



１．「無料で体験」をクリック※
※既にアカウント作成済みの場合は「サインイン」と表示される場合があります



２. SONYアカウントまたはGoogleアカウントのどちらかを選択
※アカウントがない場合には選択後に作成可能です

GoogleアカウントとSonyアカウントの違い

Sony アカウント	✓Sonyアカウントを利用しているその他のサービスとのアカウント連携が可能 ✓Sonyアカウントを既にお持ちの場合は、アカウント作成不要でNNCの利用が可能 ✓法人向けメニュー※の利用が可能
Google アカウント	✓Googleアカウントを利用しているその他のサービスとのアカウント連携が可能 ✓Googleアカウントを既にお持ちの場合は、アカウント作成不要でNNCの利用が可能

※法人向けメニューの利用には別途契約が必要です。
詳細はNeural Network Console法人版Basicをご確認ください。

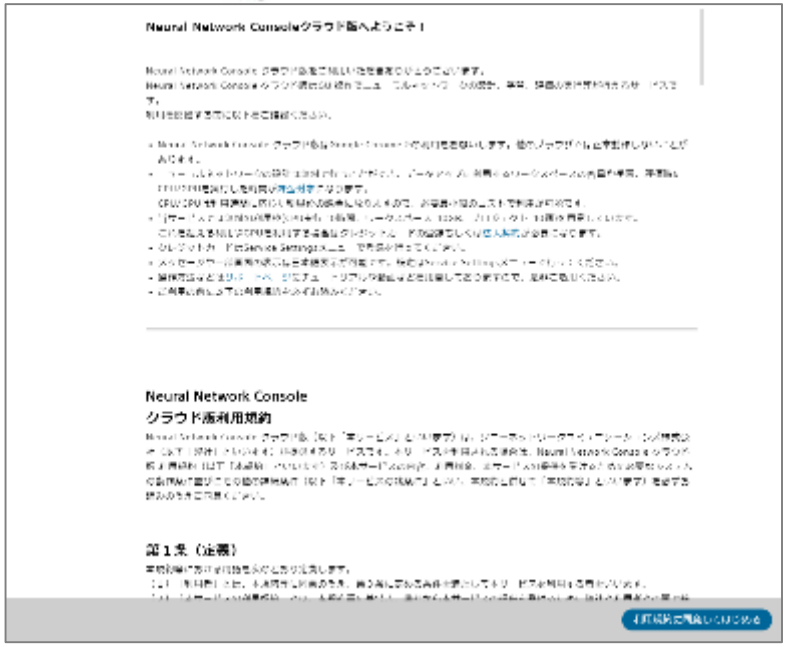
SONYアカウントでのサインイン

SONYアカウントに登録しているメールアドレス・パスワードを入力し、ログインを行います。

※アカウントをお持ちでない方は、「新しいアカウントの作成」から新規作成を行ってください。
詳細は、Appendixの[SONYアカウントの取得方法](#)に記載があります。

1．メールアドレスの入力

2．利用規約への同意



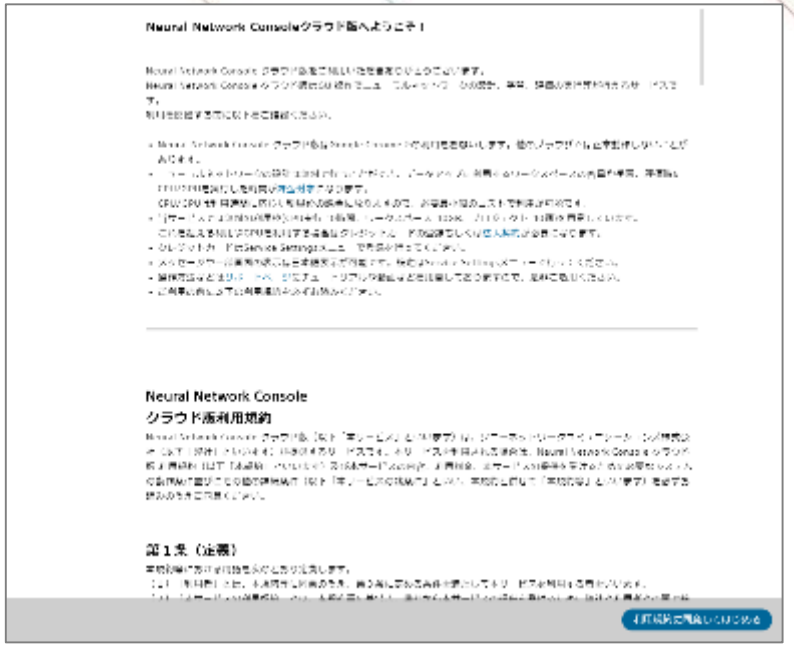
Googleアカウントでのサインイン

Googleのメールアドレス・パスワードを入力し、ログインを行います。

1．メールアドレスの入力

2．パスワードの入力

3．利用規約への同意



目次

1 Deep Learningとは

2 アカウントサインイン

3 データ前処理

4 データアップロード

5 分類モデル作成

6 分類モデル利用

データについて

お手持ちの表データを用いて分類モデルを作成する場合には、データの前処理やアップロードなどが必要ですので、次頁以降を読み進めてください。

データをお持ちでない場合には、NNCのサンプルデータを用いて本ドキュメントを進めることが可能です。本ドキュメントで利用しているNNCのサンプルデータに関してはサンプルデータの説明に説明があります。

また、UCI Machine Learning RepositoryやKaggle等で公開されているウェブ上のオープンデータを利用することで、データ準備からモデル作成を体験いただくこともできます。

データ前処理

Deep Learningの入力データは全て数値情報である必要があります。エクセルなどで管理している表データを用いてモデルを作成する際には、まずは文字列などの情報を全て数値に変換する必要があります。

ここでは表データをDeep Learningの入力データとするための一般的な前処理について解説をしていきます。

	前処理	概要
1	不要情報の削除	Deep Learningの入力・出力データとならない情報を削除します。
2	欠損値の取り扱い	Deep Learningではカラムに情報が入っていない状態では入力として取り扱えないため、事前に空白をなくす作業を行います。
3	文字列の変換	Deep Learningでは文字情報が含まれる状態では入力として取り扱えないため、事前に文字情報を数値に変換する作業を行います。
4	日時情報の変換	Deep Learningでは「:」や「-」を含んだままでは取り扱えないため、日時情報を数値に変換する作業を行います。

※このほかに数値情報は事前に正規化(入力データの値が大きい場合には学習が非効率になるために、事前に入力データを平均0、標準偏差1に変換)などの手法もありますが、ここではDeep Learningの入力データを作るうえで、必ず行わなければならない前処理について、解説をしてきます。本ドキュメントでは正則化処理と同等の処理をネットワーク内で行います。

不要情報の削除

Deep Learningの入力・出力データに該当しないカラムを削除します。
入力データには結果を予測するのに役立つ情報を用います。

例.不要データの削除(退会予測のサンプルデータ)

顧客IDと継続・退会は関係性がないのでデータを削除します。

元データ

入出力以外の不要部分は削除　入力データ　出力データ

顧客ID	入会日	性別	年齢	購入回数	購入総額	最終購入日	継続・退会
ID00001	2016/02/14	女性	36	12	17,280円	2019/11/01	継続
ID00002	2016/10/15	女性	39	8	18,000円	2019/08/23	退会
ID00003	2014/06/06		43	4	42000円	2018/01/29	退会
ID00004	2013/11/01		37	1	21600円	2019/04/06	継続
ID00005	2015/07/02	男性	46	35	20100円	2019/11/15	継続
ID00006	2016/07/10	男性	38		28800円	2019/06/19	継続
ID00007	2013/07/09	女性	28	18	42000円	2019/04/01	継続
ID00008	2012/06/09	男性	23		30000円	2018/11/07	継続
ID00009	2011/06/12		22	12	25800円	2018/12/11	退会
ID00010	2016/10/21		27	2	25200円	2019/09/23	継続

処理後のデータ

入会日	性別	年齢	購入回数	購入総額	最終購入日	継続・退会
2016/02/14	女性	36	12	17,280円	2019/11/01	継続
2016/10/15	女性	39	8	18,000円	2019/08/23	退会
2014/06/06		43	4	42000円	2018/01/29	退会
2013/11/01		37	1	21600円	2019/04/06	継続
2015/07/02	男性	46	35	20100円	2019/11/15	継続
2016/07/10	男性	38		28800円	2019/06/19	継続
2013/07/09	女性	28	18	42000円	2019/04/01	継続
2012/06/09	男性	23		30000円	2018/11/07	継続
2011/06/12		22	12	25800円	2018/12/11	退会
2016/10/21		27	2	25200円	2019/09/23	継続

欠損値の取り扱い

カラムに空欄(欠損値)がある状態ではDeep Learningの入力データとして取り扱うことはできません。
欠損値を含む行を削除するか、欠損値を他の値で補完する必要があります。

退会予測のサンプルデータに含まれる欠損値

入会日	性別	年齢	購入回数	購入総額	最終購入日	継続・退会
2016/02/14	女性	36	12	17,280円	2019/11/01	継続
2016/10/15	女性	39	8	18,000円	2019/08/23	退会
2014/06/06		43	4	42000円	2018/01/29	退会
2013/11/01		37	1	21600円	2019/04/06	継続
2015/07/02	男性	46	35	20100円	2019/11/15	継続
2016/07/10	男性	38		28800円	2019/06/19	継続
2013/07/09	女性	28	18	42000円	2019/04/01	継続
2012/06/09	男性	23		30000円	2018/11/07	継続
2011/06/12		22	12			
2016/10/21		27	2			

空欄(欠損値)のままでは取り扱えない

欠損値の取り扱い方法

欠損値を含む行の削除

- ✓欠損値を含む行を削除することで、全てのセルに情報が入ったデータを作ります
- ✓操作が簡単である反面、データ量が減少するというデメリットがあります

欠損値の補完

- ✓平均値や中央値などを用いて、空欄セルに情報を入力します
- ✓空欄セルが大量にある場合には、処理により精度が低下する懸念があります

欠損値の取り扱い: 欠損値を含む行の削除

欠損値を含む行を削除することで、欠損のないデータを作ることができます。
操作が簡易である反面、データ量が減少するため、データ量が限られる場合には不向きです。

例. 欠損値を含む行の削除

元データ

入会日	性別	年齢	購入回数	購入総額	最終購入日	継続・退会
2016/02/14	女性	36	12	17,280円	2019/11/01	継続
2016/10/15	女性	39	8	18,000円	2019/08/23	退会
2014/06/06		43	4	42,000円	2018/01/29	退会
2013/11/01		37	1	21,600円	2019/04/06	継続
2015/07/02	男性	46	35	20,100円	2019/11/15	継続
2016/07/10	男性	38		28,800円	2019/06/19	継続
2013/07/09	女性	28	18	42,000円	2019/04/01	継続
2012/06/09	男性	23		30,000円	2018/11/07	継続
2011/06/12		22	12	25,800円	2018/12/11	退会
2016/10/21		27	2	25,200円	2019/09/23	継続

処理後のデータ

入会日	性別	年齢	購入回数	購入総額	最終購入日	継続・退会
2016/02/14	女性	36	12	17,280円	2019/11/01	継続
2016/10/15	女性	39	8	18,000円	2019/08/23	退会
2015/07/02	男性	46	35	20,100円	2019/11/15	継続
2013/07/09	女性	28	18	42,000円	2019/04/01	継続

欠損値の取り扱い: 欠損値の補完

欠損部分のセルに欠損を示す新たなカテゴリや値を代入したり、欠損部分を除くデータを用いてカラムの平均値や中央値を計算し、その値を代入したりします。

また、カラムの中に欠損が多数ある場合には、そのカラムを利用しないという方法もあります。

例. 欠損値の補完

元データ

入会日	性別	年齢	購入回数	購入総額	最終購入日	継続・退会
2016/02/14	女性	36	12	17,280円	2019/11/01	継続
2016/10/15	女性	39	8	18,000円	2019/08/23	退会
2014/06/06		43	4	42,000円	2018/01/29	退会
2013/11/01		37	1	21,600円	2019/04/06	継続
2015/07/02	男性	46	35	20,100円	2019/11/15	継続
2016/07/10	男性	38		28,800円	2019/06/19	継続
2013/07/09	女性	28	18	42,000円	2019/04/01	継続
2012/06/09	男性	23		30,000円	2018/11/07	継続
2011/06/12		22	12	25,800円	2018/12/11	退会
2016/10/21		27	2	25,200円	2019/09/23	継続

処理後のデータ

入会日	性別	年齢	購入回数	購入総額	最終購入日	継続・退会
2016/02/14	女性	36	12	17,280円	2019/11/01	継続
2016/10/15	女性	39	8	18,000円	2019/08/23	退会
2014/06/06	不明	43	4	42,000円	2018/01/29	退会
2013/11/01	不明	37	1	21,600円	2019/04/06	継続
2015/07/02	男性	46	35	20,100円	2019/11/15	継続
2016/07/10	男性	38	11.5	28,800円	2019/06/19	継続
2013/07/09	女性	28	18	[illegible]	[illegible]	[illegible]
2012/06/09	男性	23	11.5	[illegible]	[illegible]	[illegible]
2011/06/12	不明	22	12	25,800円	2018/12/11	退会
2016/10/21	不明	[illegible]	[illegible]	[illegible]	[illegible]	継続

他データから計算した平均値を代入

欠損を示すカテゴリを代入

文字列の変換

Deep Learningの入力データは全て数値データにする必要があるため、文字列を取り扱うことができません。
単位やコンマを含む数値情報や、カテゴリを示すテキスト情報は事前に数値化する必要があります。

退会予測のサンプルデータに含まれる文字列

入会日	性別	年齢	購入回数	購入総額	最終購入日	継続・退会
2016/02/14	女性	36	12	17,280円	2019/11/01	継続
2016/10/15	女性	39	8	18,000円	2019/08/23	退会
2014/06/06	不明	43	4	42,000円	2018/01/29	退会
2013/11/01	不明	37	1	21,600円	2019/04/06	継続
2015/07/02	男性	46	35	20,100円	2019/11/15	継続
2016/07/10	男性	38	11.5	28,800円	2019/06/19	継続
2013/07/09	女性	28	18	42,000円	2019/04/01	継続
2012/06/09	男性	23	11.5	30,000円	2018/11/07	継続
2011/06/12	不明	22	12	25,800円	2018/12/11	退会
2016/10/21	不明	27	2	25,200円	2019/09/23	継続

※日時情報を示す文字列は日時情報の変換で説明します

文字情報の注意点

カテゴリを示すテキスト情報	✓カテゴリを示すテキスト情報は数値情報に変換をする必要があります
単位やカンマを含む数値情報	✓数値情報であっても、単位やカンマを含む場合には文字列として認識されるため、事前に削除が必要になります

文字列の変換:カテゴリを示すテキスト情報

カテゴリを示すテキスト情報の数値化には、カテゴリを数値に変換する方法と、カテゴリそのものをカラムにして該当するかどうかを0と1で表す方法(One-Hot表現)の２つの方法があります。

変換の方法については、テキストエディタやエクセルなどの置換、if関数などをお使いください。

例. カテゴリを示すテキスト情報の数値化

カテゴリを数値に変換

カテゴリマスタを準備し、それに従ってカテゴリを数値化

変換前

入会日	性別
2016/02/14	女性
2016/10/15	女性
2014/06/06	不明
2013/11/01	不明
2015/07/02	男性
2016/07/10	男性

カテゴリマスタ

変換前	変換後
男性	0
女性	1
不明	2

変換後

入会日	性別
2016/02/14	1
2016/10/15	1
2014/06/06	2
2013/11/01	2
2015/07/02	0
2016/07/10	0

※カテゴリマスタはあとで人が確認する際に利用

カテゴリをカラムに追加し数値で該当かどうかを表現

カテゴリ分のカラムを作成し、セルに該当するかどうかを0か1で記載

変換前

入会日	性別
2016/02/14	女性
2016/10/15	女性
2014/06/06	不明
2013/11/01	不明
2015/07/02	男性
2016/07/10	男性

変換後

入会日	男性	女性	不明
2016/02/14	0	1	0
2016/10/15	0	1	0
2014/06/06	0	0	1
2013/11/01	0	0	1
2015/07/02	1	0	0
2016/07/10	1	0	0

文字列の変換: 単位やカンマを含む数値情報

単位やコンマを含む数値情報は文字情報になってしまうため、単位やコンマを削除し、数値だけにする必要があります。

例. 単位やカンマの削除

元データ

入会日	性別	年齢	購入回数	購入総額	最終購入日	継続・退会
2016/02/14	1	36	12	17,280円	2019/11/01	継続
2016/10/15	1	39	8	18,000円	2019/08/23	退会
2014/06/06	2	43	4	42,000円	2018/01/29	退会
2013/11/01	2	37	1	21,600円	2019/04/06	継続
2015/07/02	0	46	35	20,100円	2019/11/15	継続
2016/07/10	0	38	11.5	28,800円	2019/06/19	継続
2013/07/09	1	28	18	42,000円	2019/04/01	継続
2012/06/09	0	23	11.5	30,000円	2018/11/07	継続
2011/06/12	2	22	12	25,800円	2018/12/11	退会
2016/10/21	2	27	2	25,200円	2019/09/23	継続

処理後のデータ

入会日	性別	年齢	購入回数	購入総額	最終購入日	継続・退会
2016/02/14	女性	1	12	**17280**	2019/11/01	継続
2016/10/15	女性	1	8	**18000**	2019/08/23	退会
2014/06/06	不明	2	4	**42000**	2018/01/29	退会
2013/11/01	不明	2	1	**21600**	2019/04/06	継続
2015/07/02	男性	0	35	**20100**	2019/11/15	継続
2016/07/10	男性	0	11.5	**28800**	2019/06/19	継続
2013/07/09	女性	1	18	**42000**	2019/04/01	継続
2012/06/09	男性	0	11.5	**30000**	2018/11/07	継続
2011/06/12	不明	2	12	**25800**	2018/12/11	退会
2016/10/21	不明	2	2	**25200**	2019/09/23	継続

日時情報の変換

日時情報も数値情報に変換する必要があります。現在までの経過日数や経過時間に変換したり、曜日をカテゴリとして作成したり、五十日のフラグを作成したりします。どのように変換を行うかはモデル化しようとしている問題にあわせて、判断してください。

退会予測のサンプルデータの例

退会予測では入会や最終利用時からどの程度時間が経っているかが重要であるため、日時情報を経過時間に変更します。
また、サービスの利用頻度から判断して、単位は”日”としています。

入会日	性別	年齢	購入回数	購入総額	最終購入日	継続・退会
2016/02/14	1	36	12	17280	2019/11/01	0
2016/10/15	1	39	8	18000	2019/08/23	1
2014/06/06	2	43	4	42000	2018/01/29	1
2013/11/01	2	37	1	21600	2019/04/06	0
2015/07/02	0	46	35	20100	2019/11/15	0
2016/07/10	0	38	11.5	28800	2019/06/19	0
2013/07/09	1	28	18	42000	2019/04/01	0
2012/06/09	0	23	11.5	30000	2018/11/07	0
2011/06/12	2	22	12	25800	2018/12/11	1
2016/10/21	2	27	2	25200	2019/09/23	0

入会後経過日数	性別	年齢	購入回数	購入総額	最終購入後経過日数	継続・退会
1386	1	36	12	17280	30	0
1142	1	39	8	18000	100	1
2004	2	43	4	42000	671	1
2221	2	37	1	21600	239	0
1613	0	46	35	20100	16	0
1239	0	38	11.5	28800	165	0
2336	1	28	18	42000	244	0
2731	0	23	11.5	30000	389	0
3094	2	22	12	25800	355	1
1136	2	27	2	25200	69	0

目次

1 Deep Learningとは

2 アカウントサインイン

3 データ前処理

4 データアップロード

5 分類モデル作成

6 分類モデル利用

データアップロードのステップ

Neural Network Consoleはクラウドサービスのため、モデルを作成するために必要なデータセットをあらかじめクラウドにアップロードする必要があります。

お手持ちのPCにデータセットを準備し、以下のステップでクラウドへのアップロードを行います。

データアップロードのステップ

1. データセットの準備
2. アップロード用CSVファイルの準備
3. データセットのアップロード

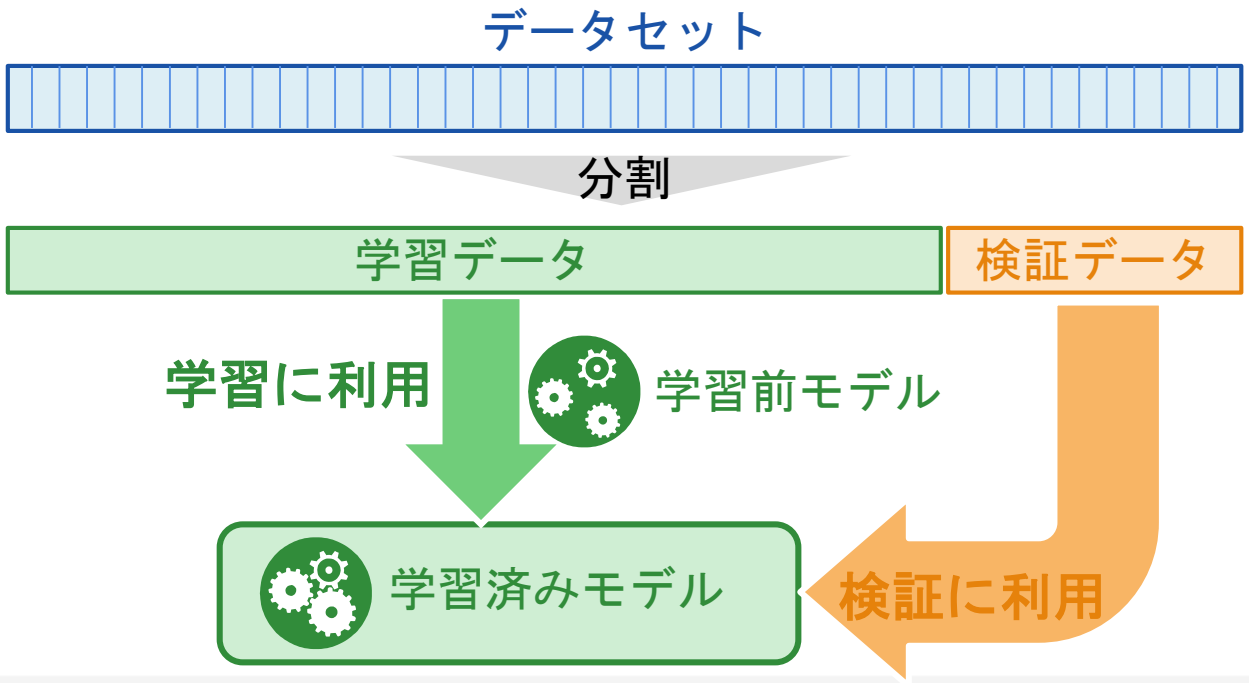
データセットの準備

モデル作成には、モデルを学習させるためのデータセット（学習データ、Training Data）と、モデルの精度を検証するためのデータセット（検証データ、Validation Data）の２つが必要になります。

作成されたモデルの精度を正しく検証するためには、学習に利用していないデータで検証データを準備する必要があるため、あらかじめ準備したデータセットを学習データと検証データに分割しておきます。

このとき、学習データと検証データの分割割合は7:3や8:2が一般的です。

データセットの件数については、Deep Learningのモデルはデータ数が多ければ多いほど、精度が高くなる傾向があります。(参考: データの重要性)



CSVファイルによるデータセット作成

データセットは入力データと出力データのセットとなり、ここでは、CSVファイルの形式で作成します。カラム名に入力データをx、出力データをyと記載することが一般的です。複数カラムの入力は”x__0”、”x__1”のようにベクトル表記をします。

データセットの作成例(退会予測のサンプルデータ)

データ前処理後のデータ

入力データ　出力データ

入会後経過日数	性別	年齢	購入回数	購入総額	最終購入後経過日数	継続・退会
1386	1	36	12	17280	30	0
1142	1	39	8	18000	100	1
2004	2	43	4	42000	671	1
2221	2	37	1	21600	239	0
1613	0	46	35	20100	16	0
1239	0	38	11.5	28800	165	0
2336	1	28	18	42000	244	0

アップロード用のデータセット

入力データをx、出力データをyにカラム名を変更
複数カラムは”__{数字}”でベクトル表現

x__0	x__1	x__2	x__3	x__4	x__5	y
1386	1	36	12	17280	30	0
1142	1	39	8	18000	100	1
2004	2	43	4	42000	671	1
2221	2	37	1	21600	239	0
1613	0	46	35	20100	16	0
1239	0	38	11.5	28800	165	0
2336	1	28	18	42000	244	0

※ 学習用と検証用で2つのデータセットを作成する必要があります

データセットのアップロード

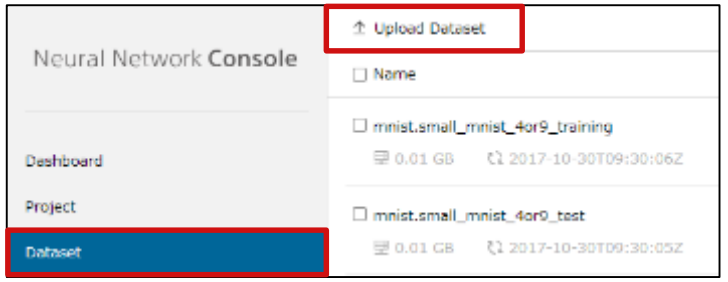
Neural Network Consoleへのデータアップロードには専用のアップローダを利用します。
アップローダ上で準備したCSVファイルを指定し、データをアップロードします。
アップロードは学習用と検証用の２回行う必要があります。

１．アップローダの取得

✓以下のリンクからアップローダをダウンロード
https://support.dl.sony.com/docs-ja/データセットアップロードツールのダウンロード/

２．アップロードキーの取得

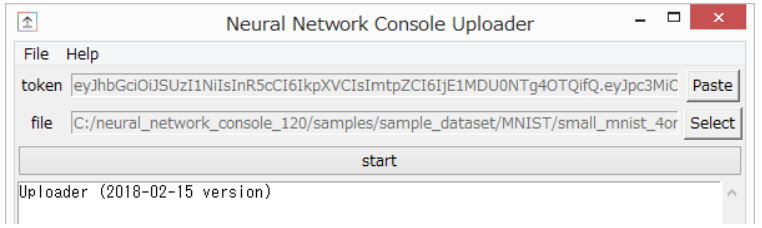
✓Neural Network Consoleにログインし、Datasetタブの中のUpload Datasetをクリック



✓ポップアップ画面に表示されるアップロードキーをコピー

３．アップローダの実行

✓１で取得したアップローダを起動
✓tokenに２で取得したアップロードキーを貼り付け
✓fileに作成したCSVファイルを指定
※ここでのファイル名がデータセット名になります
✓Startをクリックし、アップロードを実行



※アップロードには時間がかかる場合があります

アップロード先のデータセット確認

アップロード後はDatasetタブの一覧にデータセットが追加されます。

アップロード時のCSVファイルのファイル名がデータ名として一覧に表示され、選択することで中身を確認することができます。表示の際に、画像や時系列データなどはサムネイルの形で確認できます。

アップロード後のデータセットの例

アップロード時のCSVファイル

例. training_data.csv

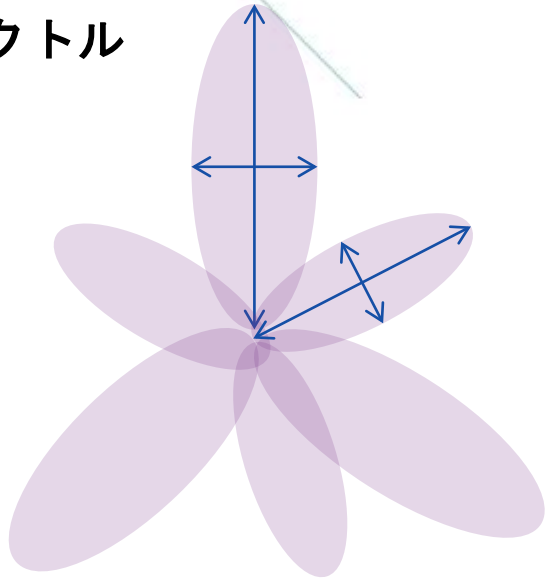
x__0	x__1	x__2	x__3	x__4	x__5	y
1386	1	36	12	17280	30	0
1142	1	39	8	18000	100	1
2004	2	43	4	42000	671	1
2221	2	37	1	21600	239	0
1613	0	46	35	20100	16	0
1239	0	38	11.5	28800	165	0
2336	1	28	18	42000	244	0

データセット一覧に追加されたデータセット

【参考】NNC上のサンプルデータ(iris_flower_dataset)の説明

3種類のあやめをがく、花びらの長さと幅の4つのパラメータから分類をします。
統計学者で植物学者でもあるRonald Fisherが発表した計測データ(Fisher Iris Flower Dataset)になります。

入力ベクトル
x



目的変数
y

0: Iris setosa
1: Iris versicolor
2: Iris virginica

データセットの例

がく、花びらの長さ、幅　　あやめの種類

	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	7	3.2	4.7	1.4	1
4	6.3	3.3	6	2.5	2
5	4.9	3	1.4	0.2	0
6	6.4	3.2	4.5	1.5	1
7	5.8	2.7	5.1	1.9	2
8	4.7	3.2	1.3	0.2	0

出典: Fisher RA. 1936. The use of multiple measurements in taxonomic problems. Annals of Eugenics 7:179-188.

目次

1 Deep Learningとは

2 アカウントサインイン

3 データ前処理

4 データアップロード

5 分類モデル作成

6 分類モデル利用

分類モデルの作成ステップ

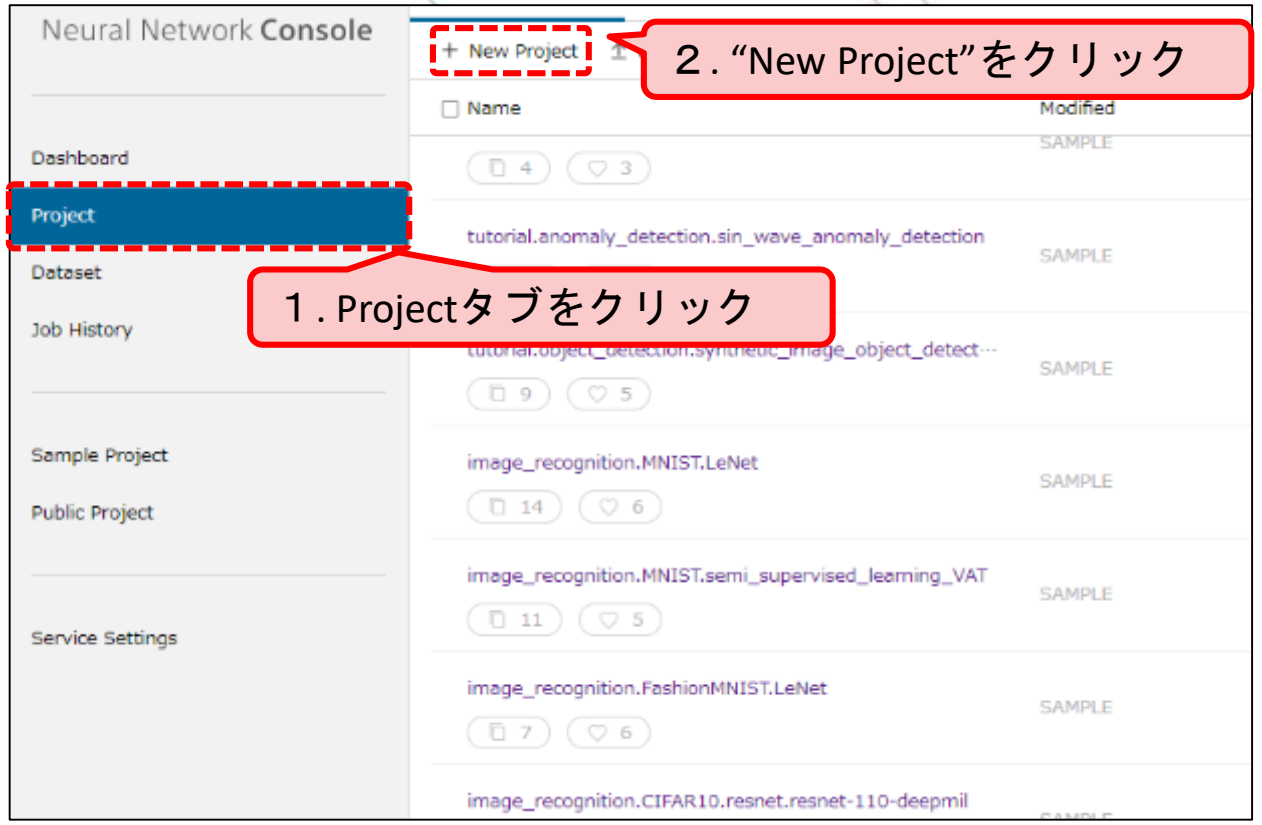
iris_flower_datasetのサンプルデータを用いて、表データの分類モデルの作成過程を説明します。

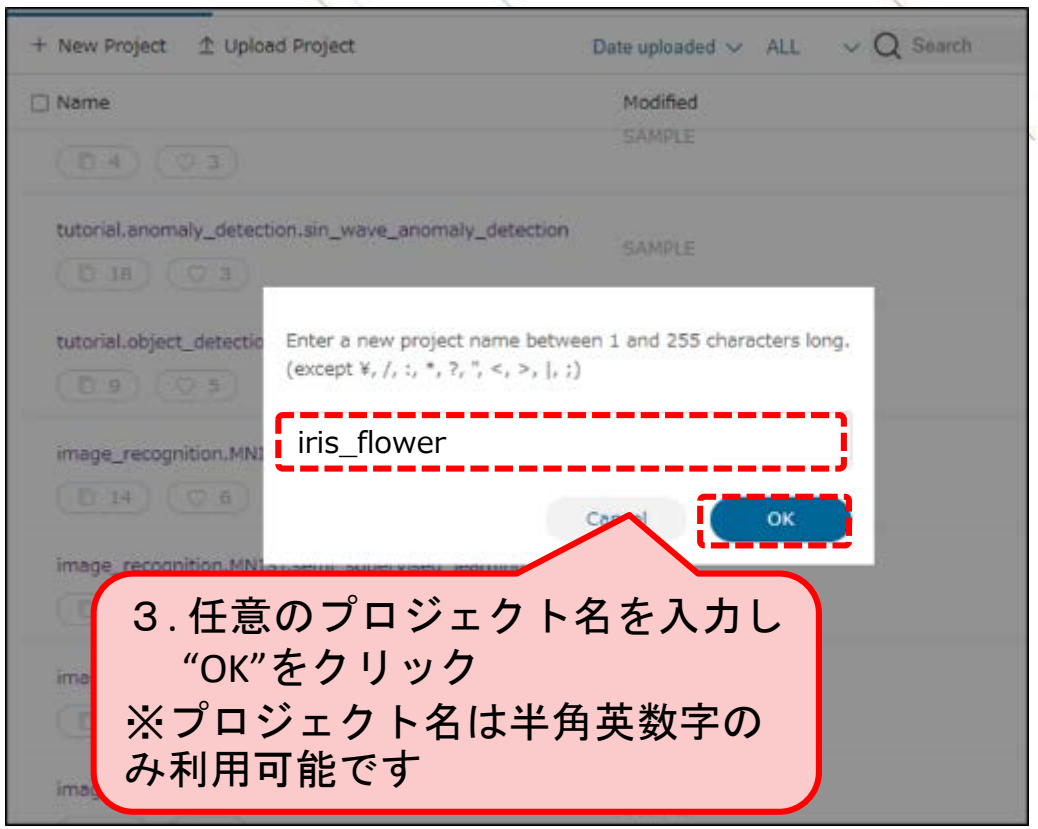
表データの分類モデルの作成ステップ

1. 新規プロジェクトの作成
2. データセットの指定
3. ネットワークの作成
4. 学習・評価

新規プロジェクトの作成

Projectタブから新規プロジェクトを作成します。





データセットの指定

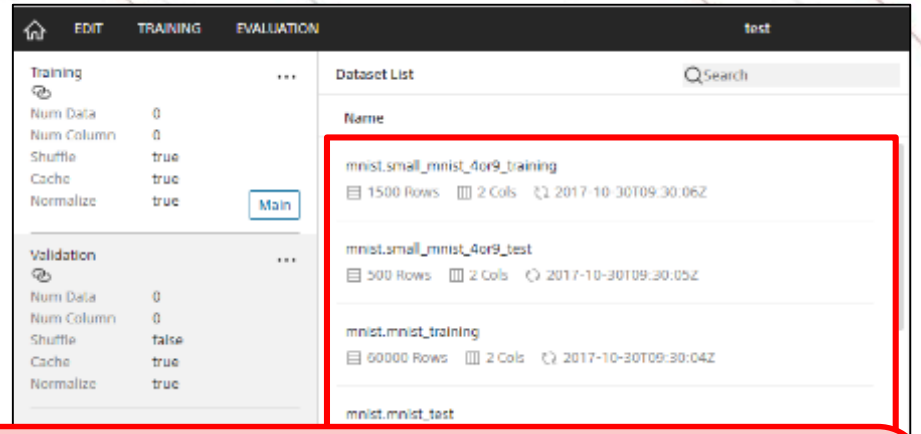
DATASETタブからTrainingとValidationのデータセットをそれぞれ紐づけます。

1. Datasetタブをクリック

2. Trainingをクリック

3. Not Setをクリック

6. Validationをクリックし、3～5と同様の手順を実施

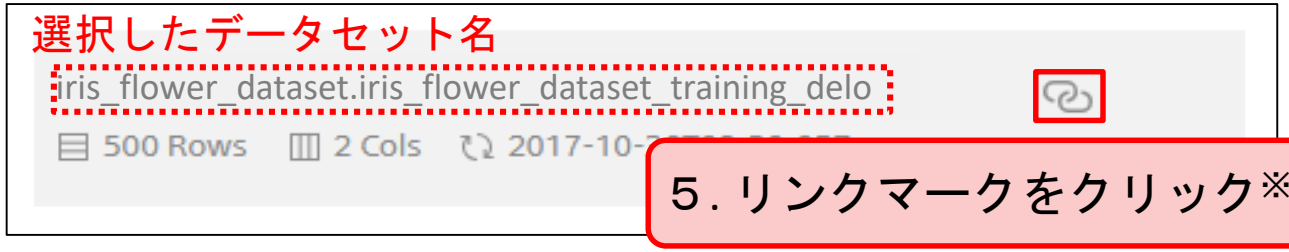


4. 一覧から学習に用いるデータセットを選択
本ドキュメントではNNCのサンプルである以下のデータセットを利用しますが、ご自身のデータセットをご利用になる場合にはデータセットのアップロード時に指定したファイル名を選択してください。

Train: iris_flower_dataset.iris_flower_dataset_training_delo

Validation: iris_flower_dataset.iris_flower_dataset_validation_delo

選択したデータセット名



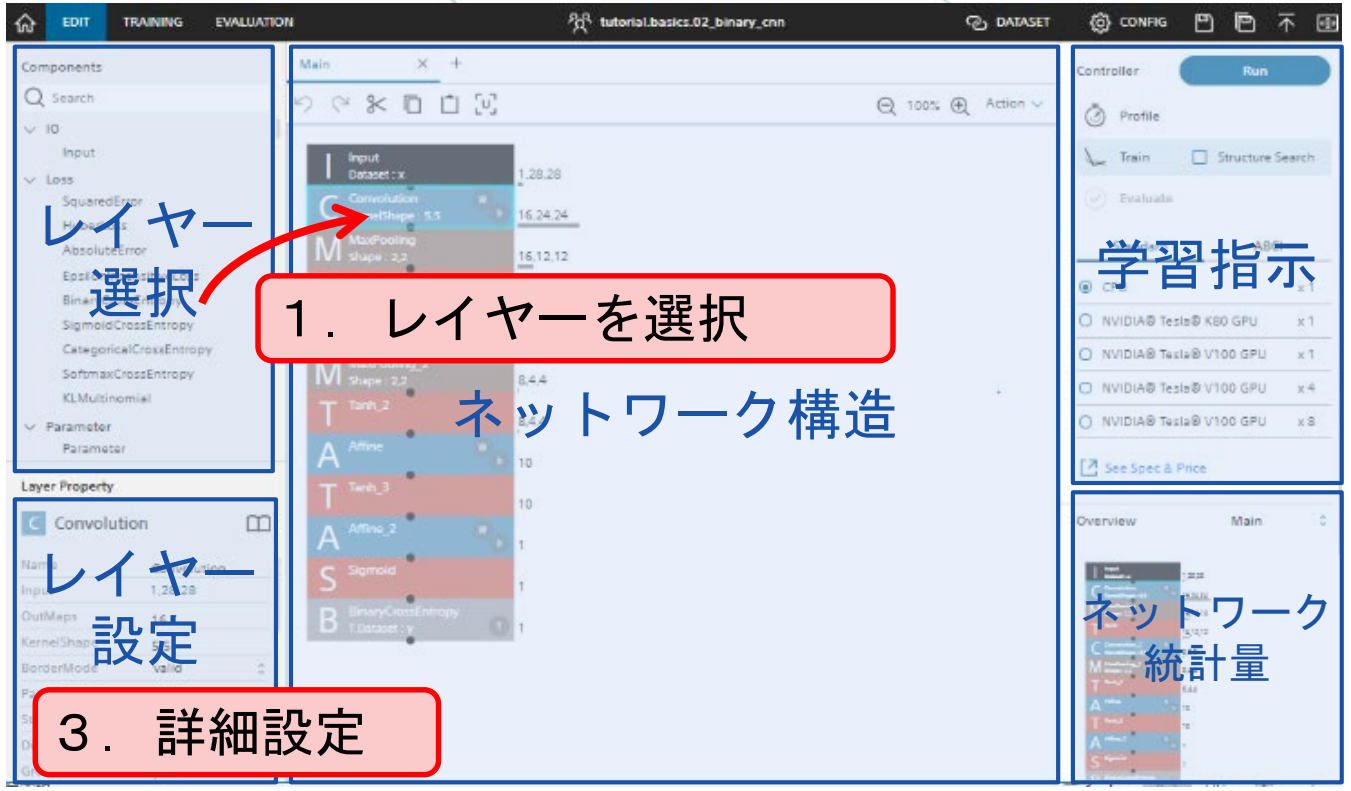
5. リンクマークをクリック※

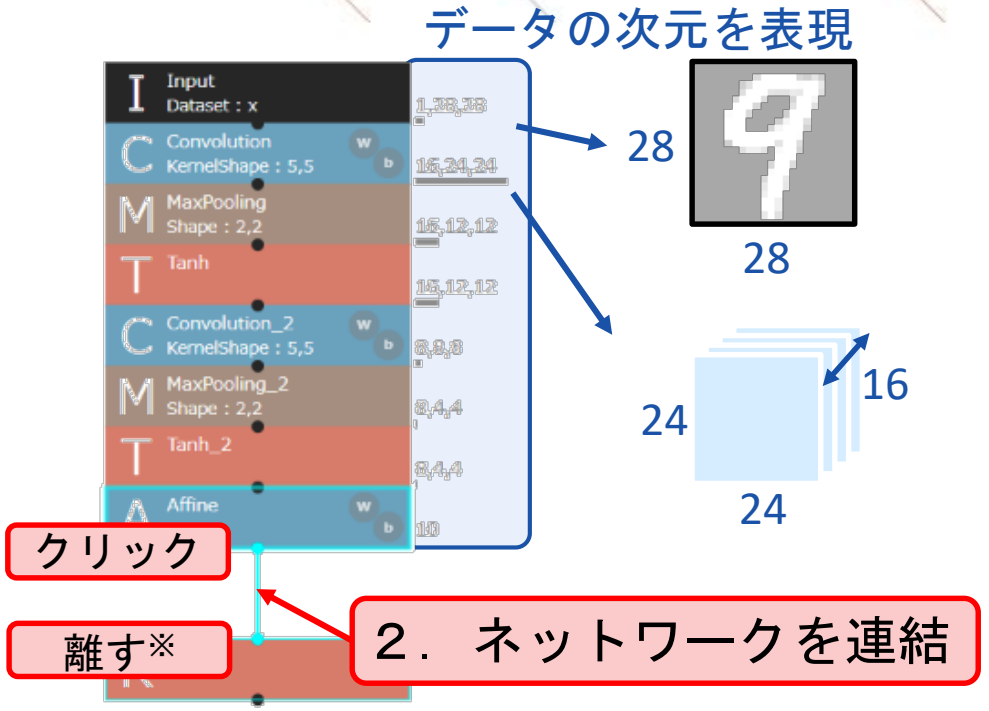
※ブラウザの拡大率によって表示されないことがあります。
表示されない場合は表示の縮小をお試しください。

ネットワーク作成の操作方法

EDITページに移動し、ネットワークを作成します。一覧から必要なレイヤーを選択し、順につなげていくことでネットワークが作成できます。レイヤーとはDeep Learningでネットワークを作成するための関数で、NNCに限らず一般的なもののため、それぞれの詳細な仕組みなどは入門書などでご確認下さい。本ドキュメントでは詳細を確認しなくともモデル作成ができる構成になっております。

表データの分類モデルに関するレイヤーの並べ方は次頁以降で説明をしていきます。





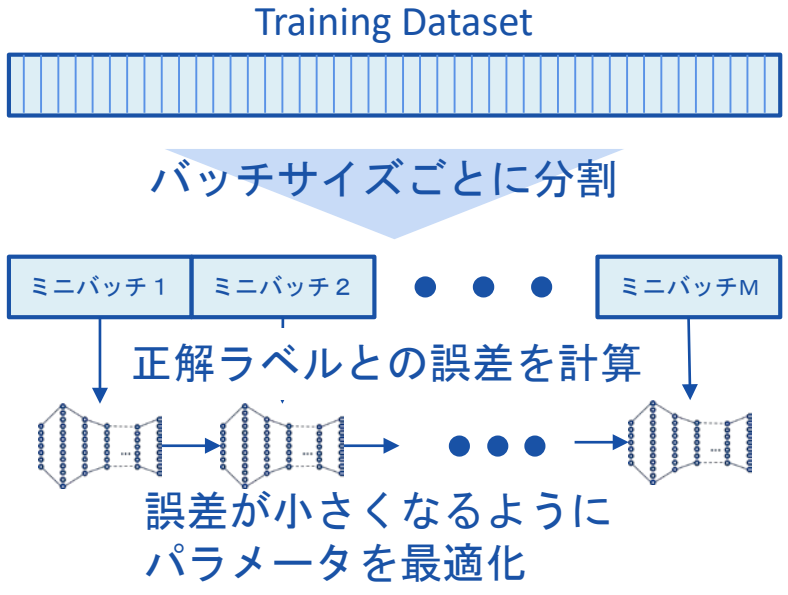
※マウスの左ボタンを離す場所はレイヤーの上端ではなく、中心にするとスムーズに連結できます。

バッチサイズの変更

学習はTraining Datasetを分割して行いますが、分割する単位をバッチサイズと呼び、CONFIGタブから変更することができます。

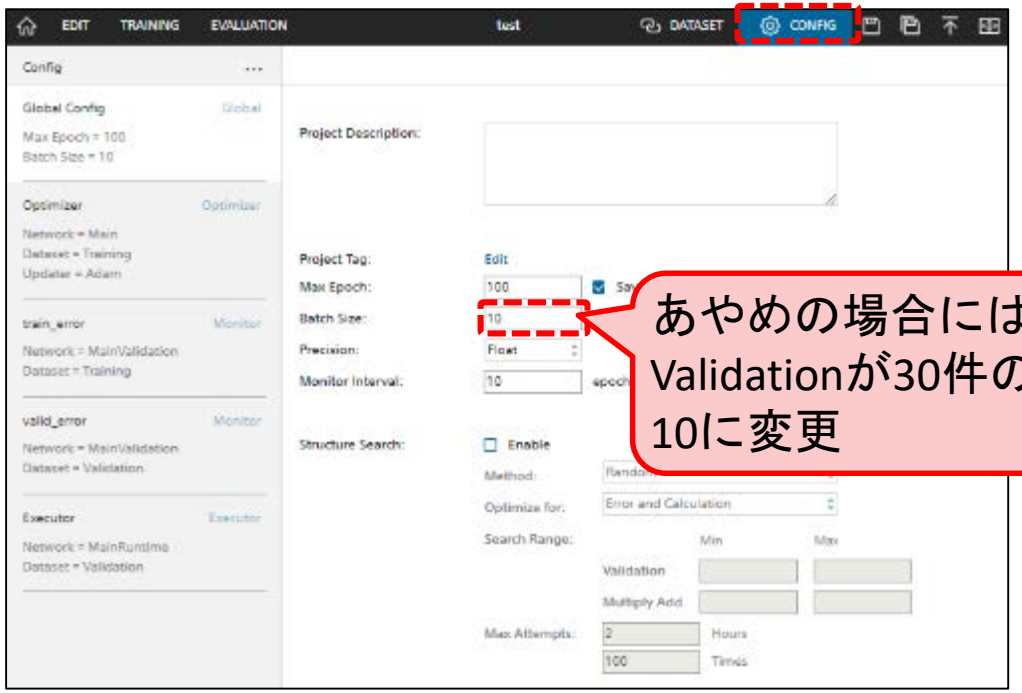
バッチサイズがデータ件数よりも大きすぎるとエラーが発生しますし、極端に大きな数値はメモリ不足でのエラーの原因にもなります。一方で小さすぎるとパラメータ更新が頻発し、学習が非効率になります。

学習の仕組み



※各世代ごとにランダムにミニバッチを取り直すことが一般的

CONFIGタブでのバッチサイズの設定



入出力のレイヤーを配置

まずは入出力のサイズや学習のための誤差などをレイヤーを並べて指定します。

分類をするモデルの場合には以下のセットが基本となります。
数値を予測する回帰モデルを作成する場合にはこのセットが異なります。(参考: 回帰モデルの入出力レイヤー)

レイヤーの役割

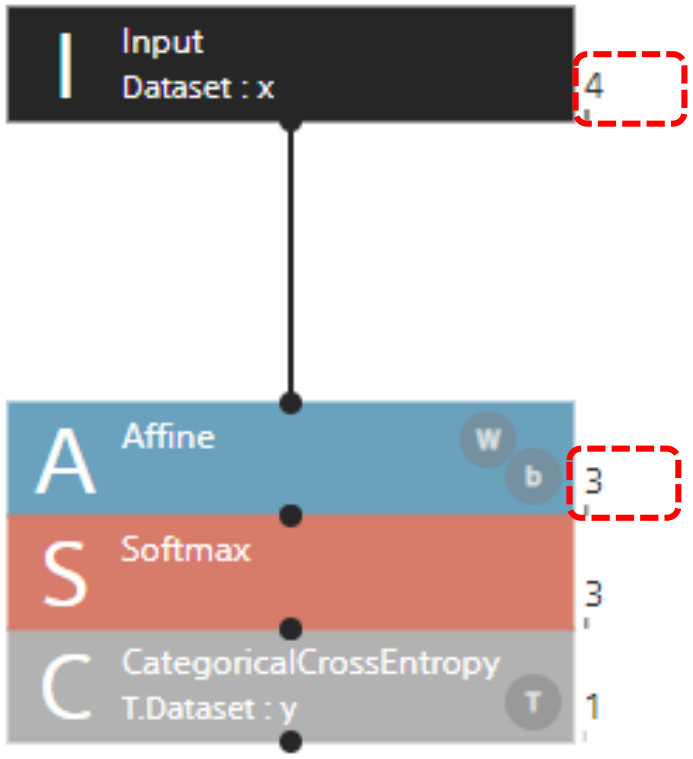
入力データのサイズを設定
当サンプルではアヤメの「がくの幅」、「がくの長さ」、「花びらの幅」、「花びらの長さ」の4つ

出力の分類数を設定
当サンプルでは3種類のアヤメに分類

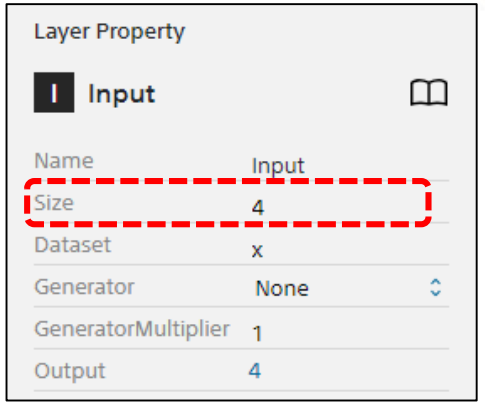
確率値での出力を設定

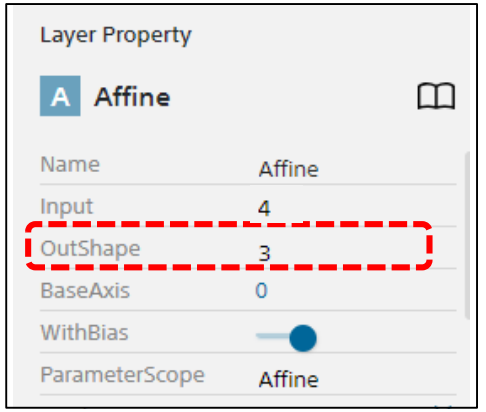
誤差の計算方法を設定

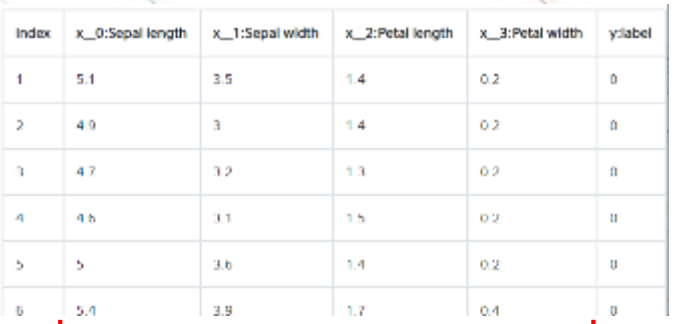
レイヤー配置



修正箇所





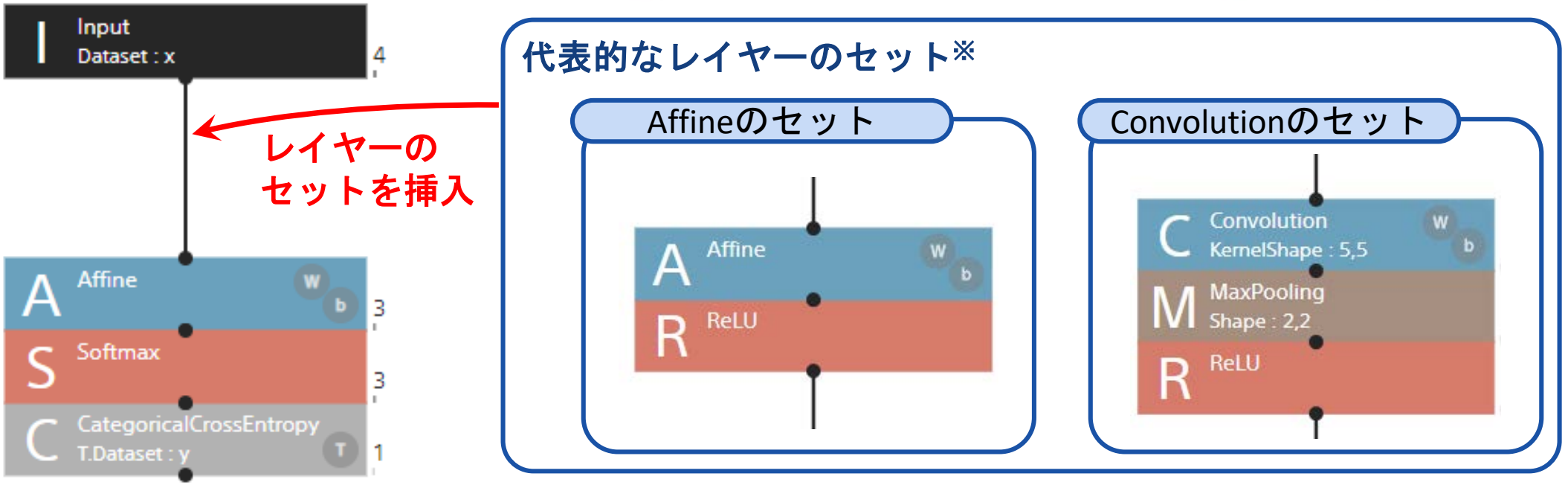


入力データのサイズはDatasetタブで入力カラム数で確認

ネットワーク作成の考え方

前頁で作成したネットワークは入出力だけの最小の状態です。これから入力と出力の間にレイヤーを入れることでネットワークを作成していきます。

一般的にはレイヤーを組み合わせたセットを最小単位として考え、このセットを組み合わせていきます。
代表的なセットは下で示したAffineのセット、Convolutionのセットになります。
Convolutionのセットは入力データが画像の場合によく使われます。



※ 一般にAffineのセットだけを組み合わせたモデルをDeep Neural Network(DNN)、Convolutionのセットを用いたモデルをConvolutional Neural Network(CNN)と呼びます。

Affineのセットの挿入

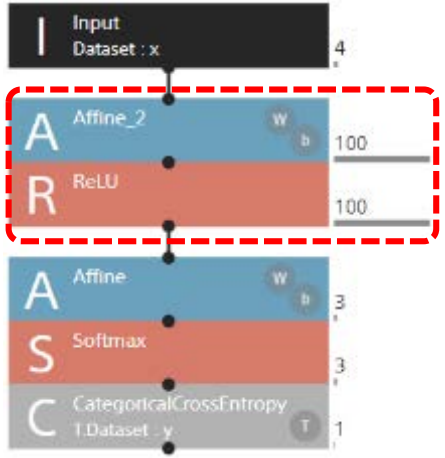
まずはAffineのセットを一つ挿入し、次頁以降の学習・評価を実施します。

次にネットワークを作り替え、学習・評価を繰り返し、より高精度のモデルを作成していきます。ネットワークの作り替えは、Affineのセットの数を変更する方法とAffineのサイズを変更する方法の2種類があります。セットの数やサイズは大きくすればモデルが複雑になるため精度向上が期待されますが、一方で、複雑すぎて上手く学習させることができず精度が低下する場合もあります。

モデル作成（1回目）

Affineのセットを追加

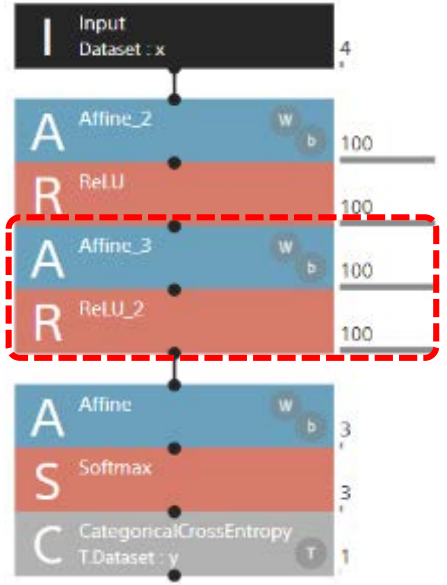
入出力の間にAffineとReLUの層を挿入



モデル作成（2回目以降）

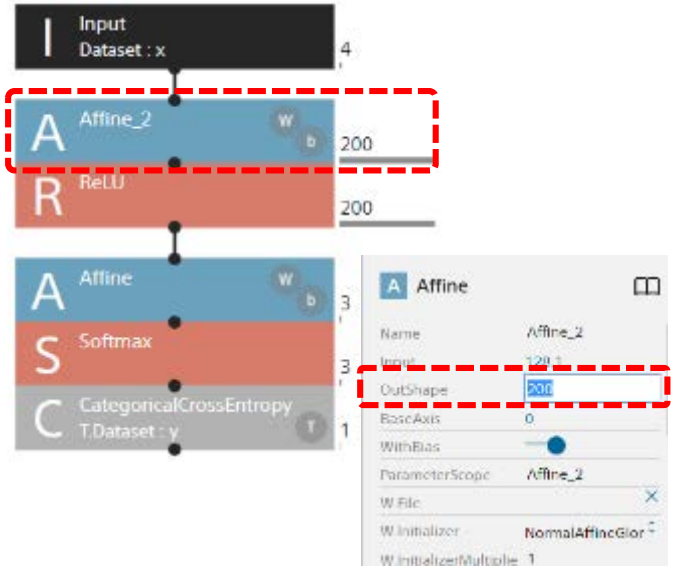
Affineのサイズの変更

入出力の間にAffineとReLUの層を挿入



Affineのセットを追加

AffineのパラメータのOutShapeを変更



…

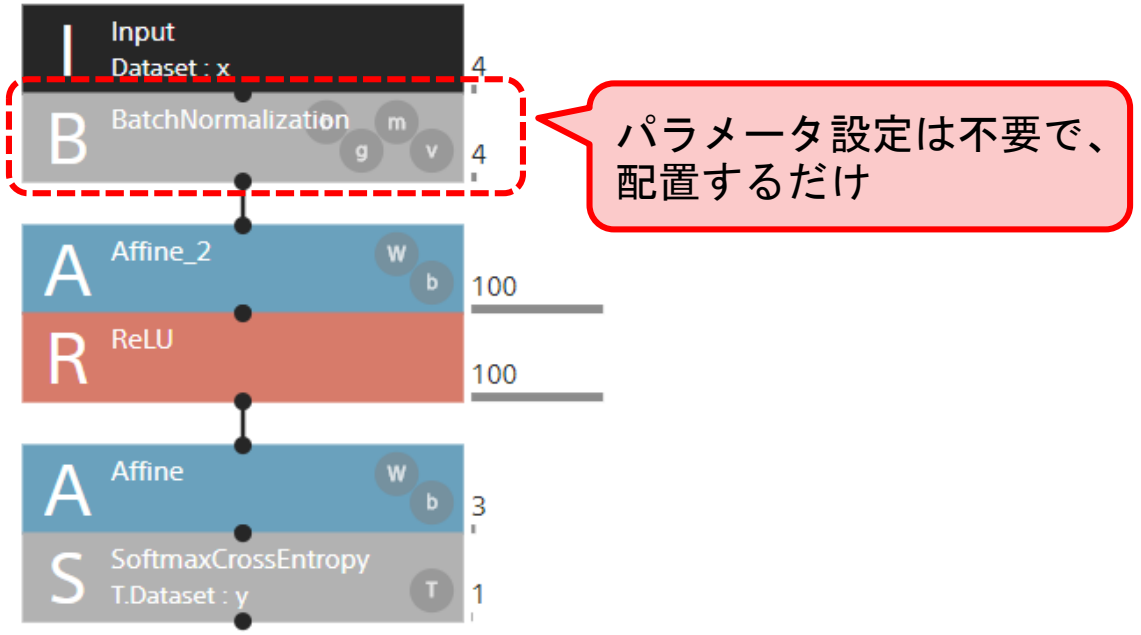
(参考) BatchNormalizationの利用

BatchNormalizationとは前の層のデータを平均を0、標準偏差を1に正規化するレイヤーです。

Deep Learningではネットワーク中の値が大きくなると学習効率が悪くなるため、ネットワークの途中にBatchNormalizationを用いることで学習プロセスを全体的に安定化させ、学習速度を高めることができます。

入力データの値が大きい場合にはInputレイヤーの直下に配置することで精度が向上する可能性があります。

Inputレイヤーの直下にBatchNormalizationを利用する例



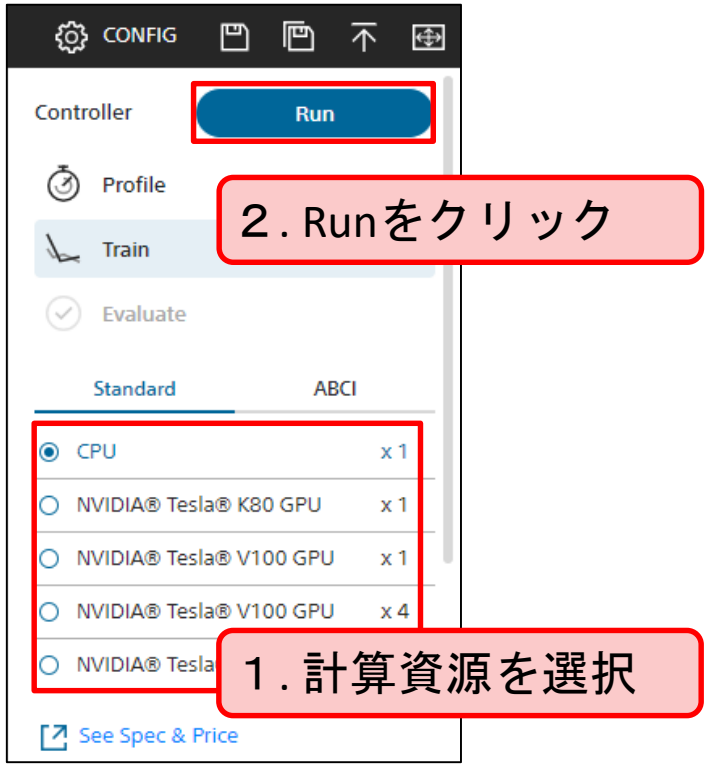
学習の実行

EDITページのRunボタンをクリックすることで学習が実行されます。

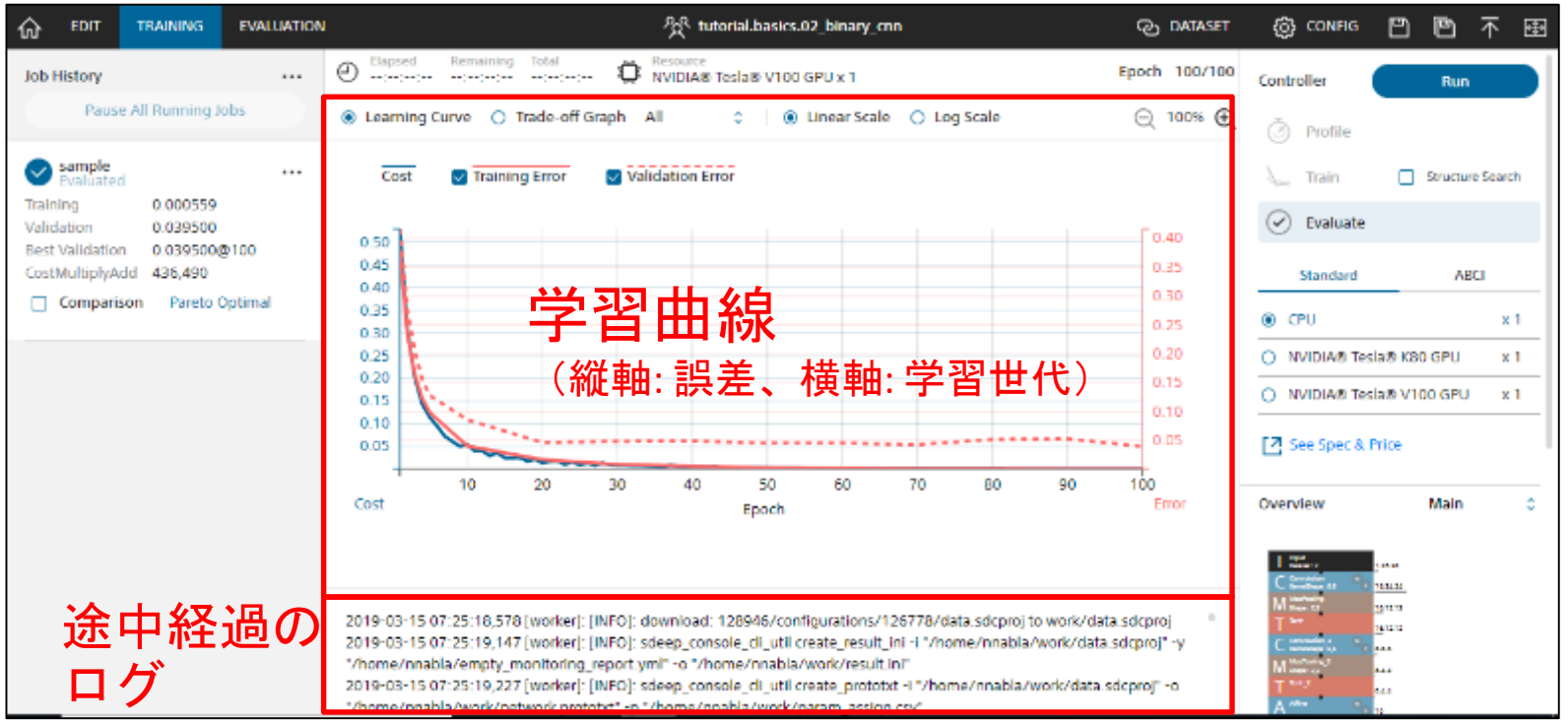
GPUを選択すると、高速に学習を行うことができます。（参考: 学習環境と処理時間）

GPU等有料のメニューを利用する場合は事前にクレジットカード登録もしくは法人契約が必要になります。
(法人契約：https://dl.sony.com/ja/business/)

学習実行の方法



TRAININGページの概要



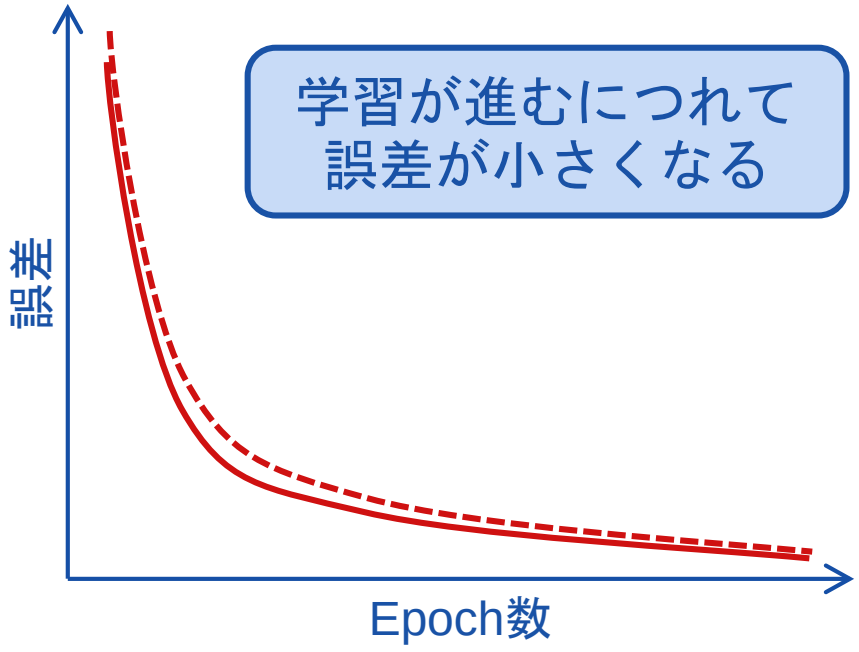
学習曲線の読み取り方

学習結果の良し悪しは、まずは学習曲線から判断をします。

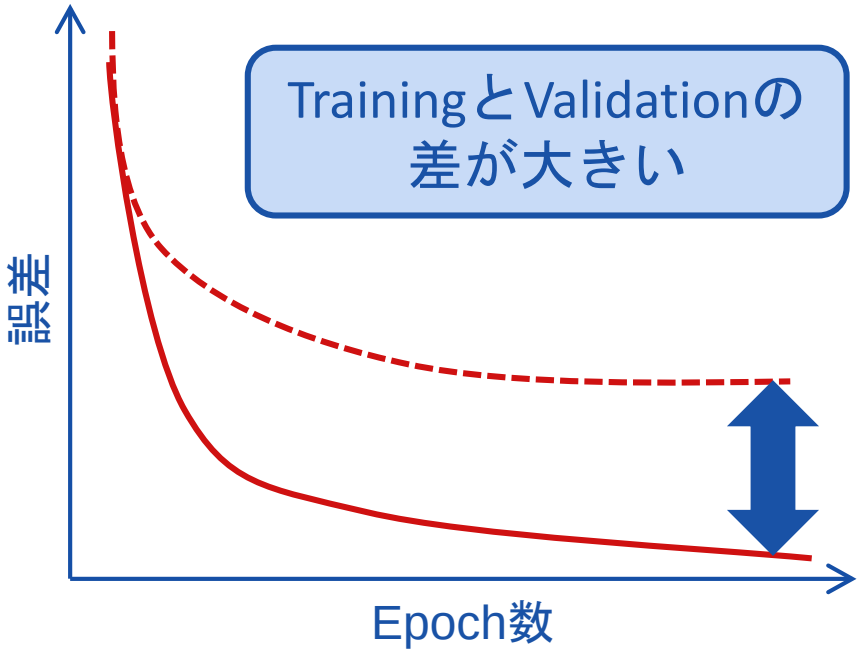
TrainingとValidationの差が大きい場合(過学習)は、モデルがTraining Dataに特化し過ぎた状態(教科書を丸暗記した場合に応用問題が解けないのと似た状態)です。

未知のデータの予測精度が低いため、データを増やしたり、レイヤーの数を減らすなどの改善が必要です。

良いモデルの学習曲線



悪いモデルの学習曲線（過学習）



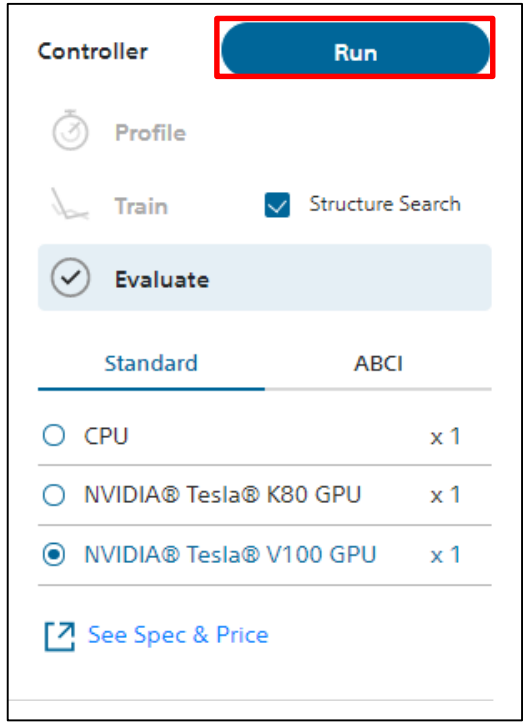
Training Data
(学習に使用したデータ)

Validation Data
(学習に使用しないデータ)

評価の実行

TRAININGページのRunをクリックするとEVALUTIONページに遷移し、詳細な判定結果を確認できます。
各データに対するモデルの判定結果や統計的な精度や指数、混同行列などを確認できます。

評価実行の方法



表示可能なグラフの概要

評価グラフ	内容	問題
Output Result	各データの１つ１つの判定結果	分類/回帰
Confusion Matrix	データセット全体の統計的な指標と混同行列(分類ラベルごとに結果を集計した表)	分類
Classification Result	各データの判定確率上位３カテゴリの確率	分類
Classification Matrix	カテゴリごとのモデルの判定傾向	分類
Likelihood Graph	判定確率と正答率の傾向	分類

評価の見方：Output Result

検証用データの右側にモデルの予測結果が表示されます。
各検証用データに対して、作成したモデルがどのように判断したか確認できます。

Output Resultページのスナップショットとその見方

Output Result　Confusion Matrix　y - y'　　« ‹ 1 / 3 › »

クリックして選択　Classification Matrix　y - y': Recall　Likelihood Graph　y - y'

ページを変更し全ての結果を確認可能

Index	x__0:Sepal length	x__1:Sepal width	x__2:Petal length	x__3:Petal width	y:label	y'__0	y'__1	y'__2
1	5	3.5	1.3	0.3	0	0.9998672	0.00013279427	1.2827646e-13
2	4.5	[illegible]	[illegible]	[illegible]	0	[illegible]	[illegible]	[illegible]16e-10
3	4.4	3.2	1.3	0.2	0	0.9997135	0.00028647474	1.9616004e-12
4	5	3.5	1.6	0.6	0	0.9992925	0.0007074302	4.8992225e-12
5	5.1	3.8	1.9	0.4	0	0.99963176	0.00036823808	8.635526e-13
6	4.8	3	1.4	0.3	0	0.99918336	0.0008166093	6.631654e-12
7	5.1	3.8	1.6	0.2	0	0.9999064	9.353141e-05	4.6298235e-

検証用データ

モデルの判定結果

追記カラム名について

- 出力をアポストロフィー付きで表記
  例: 学習時にx → yであれば、y'を出力
- 分類問題の場合にはさらにindexを用い、各クラスの予想確率を出力
  例: 2値分類の場合、y'__0とy'__1を出力

ラベル0の確率: 0.9992925
ラベル1の確率: 0.0007074302
ラベル2の確率: 4.8992225e-12
→ラベル0と予測できている
※e-12は指数関数表示で10^{-12}であり、非常に小さいことを示す

評価の見方：Confusion Matrix

検証用データに対する統計的な評価指標と混同行列を表示します。
混同行列を用いて、全体の正答数や間違いやすいラベルの傾向などを確認できます。

Confusion Matrixページのスナップショットとその見方

Output Result | Confusion Matrix y - y' | Classification Result y - y' | Classification Matrix y - y': Recall | Likelihood Graph y - y'

クリックして選択

モデル全体の評価指標

Accuracy	0.8333
Avg.Precision	0.8889
Avg.Recall	0.8333
Avg.F-Measures	0.8222

Accuracy: 全データの内、正答した数の割合
Precision:予測を正と判断した中で、答えも正のものである割合
Recall:答えが正の中で、予測が正とされた割合
F-Measure: PrecisionとRecallの調和平均

y'(モデルによる判定結果)

混同行列

	Recall	y'__0	y'__1	y'__2
		1	1	0.6667
F-Measures		1	0.6667	0.8
y:label=0	1	10	0	0
y:label=1	0.5	0	5	5
y:label=2	1	0	0	10

y (あらかじめ設定した正解結果)

正解がラベル"1"のものを"2" と予想し、間違えた数が5つ

目次

1 Deep Learningとは

2 アカウントサインイン

3 データ前処理

4 データアップロード

5 分類モデル作成

6 分類モデル利用

分類モデル利用のステップ

Neural Network Consoleから分類モデルをダウンロードすることで、お客様の環境で自由にモデル利用ができます。

モデルを実行するためには、Neural Network Libraries(NNL)が必要になります。

NNLを用いることでコマンドラインやpythonなど様々な方法で分類モデルが実行可能となります。

分類モデル利用のステップ

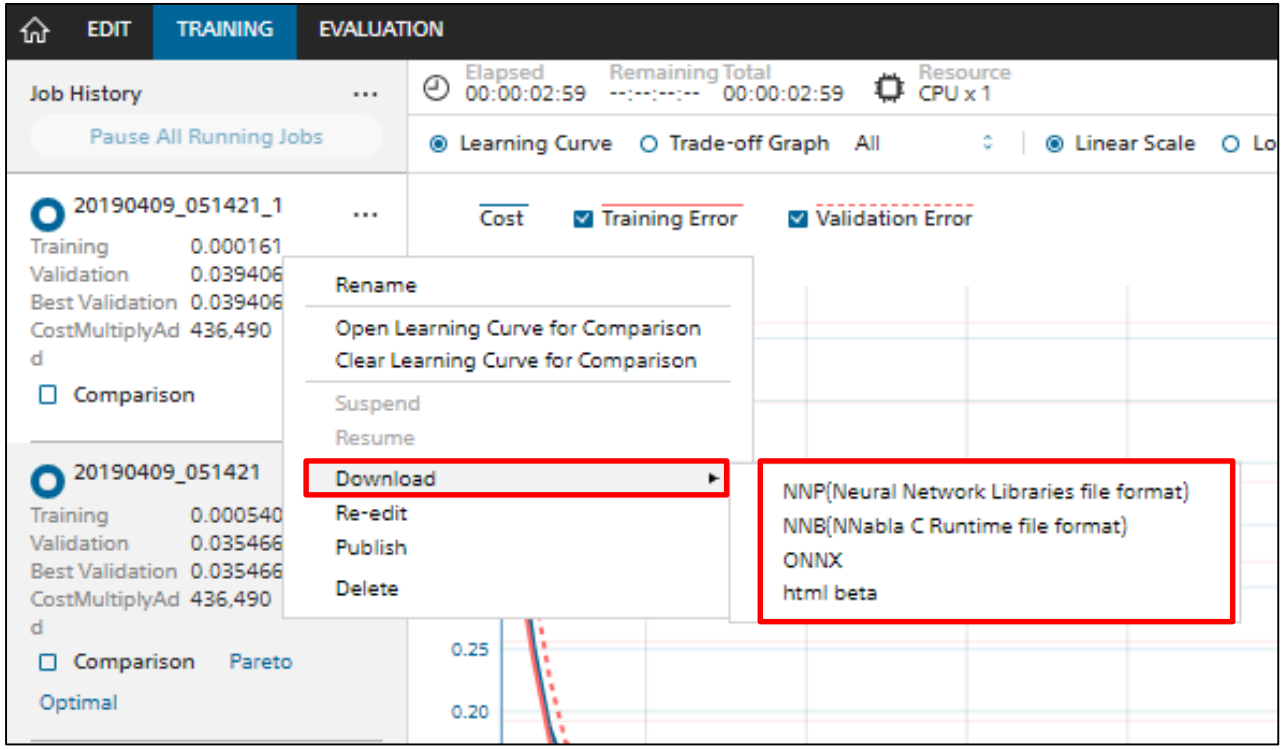
1. 分類モデルのダウンロード
2. Neural Network Librariesの設定
3. 分類モデルの実行

分類モデルのダウンロード

学習が完了し最適なモデルを作成した後は、「Job History」の中から該当のモデルを右クリックし、選択肢の中の「Download」をクリックすることでモデルをダウンロードできます。

NNP、NNB、ONNXは作成したネットワークと学習済みパラメータの値が含まれたファイルで、実行方法に応じて使い分けをします (詳細はモデルの実行方法) 。また、html betaは学習結果などの内容をhtml形式で出力したものです。

作成したモデルの権利は作成者に帰属し、自由にDeep Learningのモデルを利用することができます。



Neural Network Librariesの設定

任意のPCにNeural Network Librariesをインストールします。

Neural Network Librariesのインストールについては、以下のドキュメントをご参照ください。
http://nnabla.readthedocs.io/en/latest/python/installation.html

モデルの実行方法

Neural Network Librariesを用いてモデルを実行する方法は、使用する言語に応じて様々な方法があります。
また、ONNXを利用することで、他のDeep Learningのフレームワークを利用することも可能です。
次頁以降では、コマンドラインまたはPythonで実行する方法を解説いたします。

	実行方法	GPU利用	特徴	ダウンロードファイル	参考URL
1	コマンドライン	可能	最も簡単に利用可能	NNPファイル	https://support.dl.sony.com/docs-ja/チュートリアル：neural-network-consoleによる学習済みニューラ/
2	Python	可能	比較的容易に利用可能	NNPファイル	https://support.dl.sony.com/docs-ja/チュートリアル：neural-network-consoleによる学習済みニューラ/
3	C++	可能	推論環境にPythonのインストールが不要	NNPファイル	https://github.com/sony/nnabla/tree/master/examples/cpp/mnist_runtime
4	C	不可	非常にコンパクトであり、組み込み利用向き	NNBファイル	https://github.com/sony/nnabla-c-runtime
5	他Deep Learningフレームワーク	環境依存	環境依存	ONNXファイル	https://nnabla.readthedocs.io/en/latest/python/file_format_converter/file_format_converter.html
6	TensorFlow	環境依存	環境依存	TensorFlow frozen graphファイル	TensorFlowのウェブページをご覧ください

次頁に解説あり

コマンドラインでの推論実行

Neural Network Librariesのインストールされたpython環境で、コマンドラインから以下を実行します。

```
nnabla_cli forward ¥
  -c [ダウンロードしたネットワークモデルファイル(*.nnp)] ¥
  -d [推論をするデータセットを取りまとめたCSVファイル] ¥
  -o [推論結果の出力ディレクトリ]
```

※ Neural Network ConsoleのEVALUATIONタブでの推論実行時に同様のコマンドを使用しているため、
ログの出力ウインドウに同様のものが出力されています。

```
2017-10-24 05:54:28,942 [worker]: [INFO]: nnabla_cli forward -c
/home/nnabla/results/results_current_100.nnp -d ccbf15a0-bcb6-4ba6-b10e-
27fc877c4348/1002/index.csv -o /home/nnabla/results
```

Pythonでの実行方法

ダウンロードしたネットワークファイルをPythonで読み込んで利用します。

```
# NNablaのインポート
import nnabla as nn
from nnabla.utils import nnp_graph
# nnpファイルの読み込み、ネットワークモデルの取り出し
nnpFile = nnp_graph.NnpLoader('./result.nnp’)
networkModel = nnpFile.get_network(‘MainRuntime ', batch_size=1)
# 入出力レイヤーの名前を取得
inputName = list(networkModel.inputs.keys())[0]
outputName = list(networkModel.outputs.keys())[0]
# 入出力レイヤーの数値変数を取得
x = networkModel.inputs[inputName]
y = networkModel.outputs[outputName]
# 推論の実行
x.d = np.array(data)
y.forward(clear_buffer=True)
# 推論結果の表示
print(y.d[0])
```

Appendix

SONYアカウントの取得

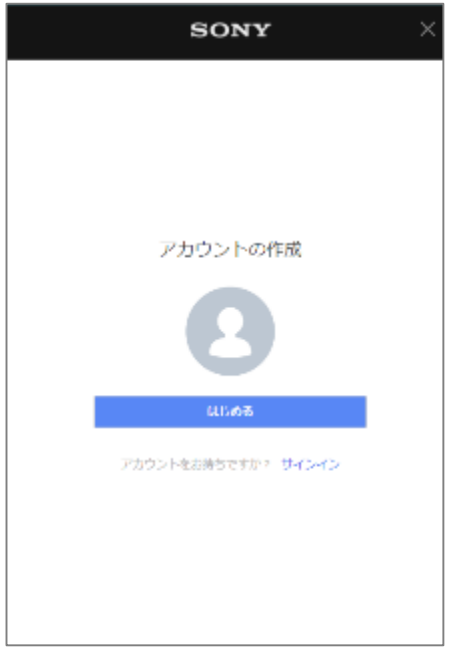
アカウント作成ページに移動し、メールアドレスやパスワードなどを設定します。

1．作成ページへの移動１

✓「新しいアカウントの作成」を押下

2．作成ページへの移動２

✓「はじめる」を押下



3．メールアドレス等の入力

✓登録するメールアドレスとパスワードを入力

SONYアカウントの取得

生年月日などを入力し、利用規約などの確認を行います。

4．生年月日の入力

✓国/地域、言語、生年月日を入力

5．利用規約への同意

✓メール配信の有無を選択
✓利用規約・アカウントポリシーの確認

6．セキュリティ認証

✓「私はロボットではありません」を押下
※画像選択が表示された場合には指示に従う

SONYアカウントの取得

確認メールを受信し、アカウントの有効化を行います。

7．確認メールの送付

✓登録したメールアドレス宛に確認メールが送付される

8．確認メールの確認

✓確認メールを開き、「確認する」を押下

学習環境と処理時間

一般的にDeep LearningはGPUを用いることにより、CPUと比べ高速に学習処理を行うことが可能です。

学習実行環境と処理時間・ご利用料金

	学習実行環境	学習処理時間	1時間当たりのご利用料金	ご利用料金目安
1	CPU	1,209,600秒 (336時間)	85円	約28,560円
2	NVIDIA® TESLA® K80 GPU	14,976秒 (4.16時間)	210円	約874円
3	NVIDIA® TESLA® V100 GPU	3,960秒 (1.1時間)	560円	約616円

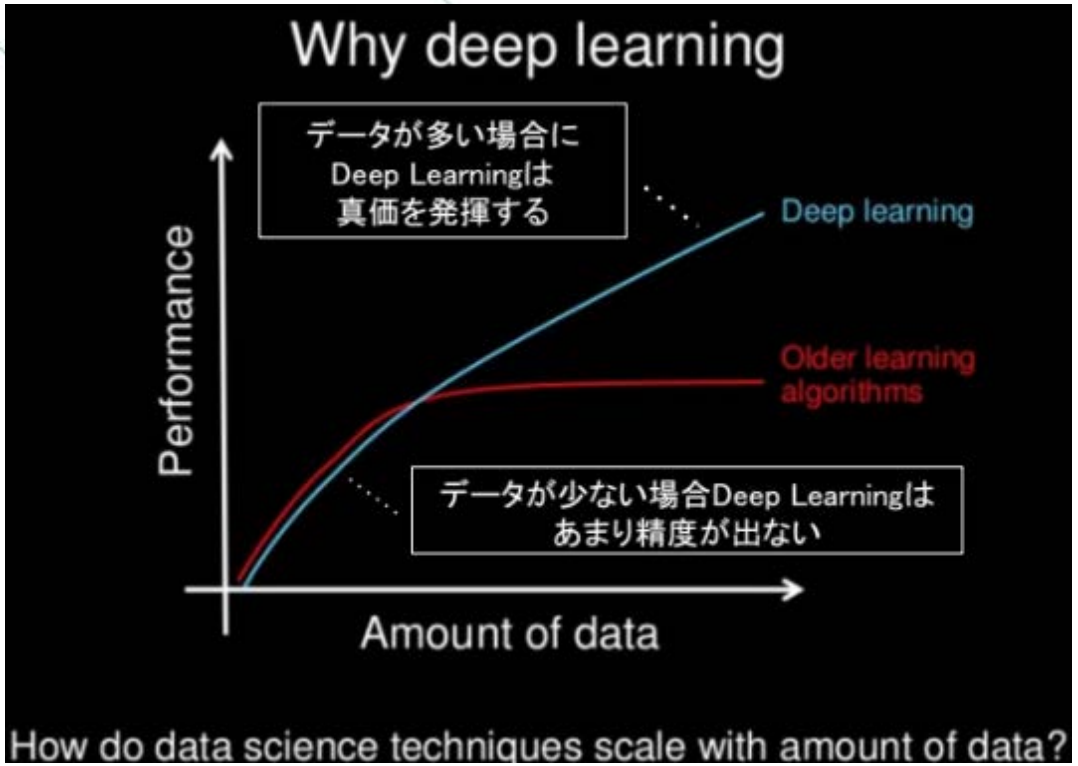
【検証環境】

- ✓ データセット：CIFAR 10
- ✓ ネットワーク：ResNet-101
- ✓ epoch：300

データ量の重要性

Deep Learningで高い精度を得るにはデータ量が重要になります。Deep Learningではデータを増やせば増やすだけ精度が向上する傾向にあります。

一方でデータ量が少ない場合には、Deep Learning以前の従来型の機械学習に比べても精度が劣ることもあります。



出典: https://www.slideshare.net/ExtractConf/andrew-ng-chief-scientist-at-baidu

回帰モデルの入出力レイヤー

Deep Learningではカテゴリを予測する分類モデルと数値を予測する回帰モデルでは最後のネットワーク部分だけが異なります。中間のネットワークの作り方については違いはありません。

数値を予測する回帰モデルの最小モデル

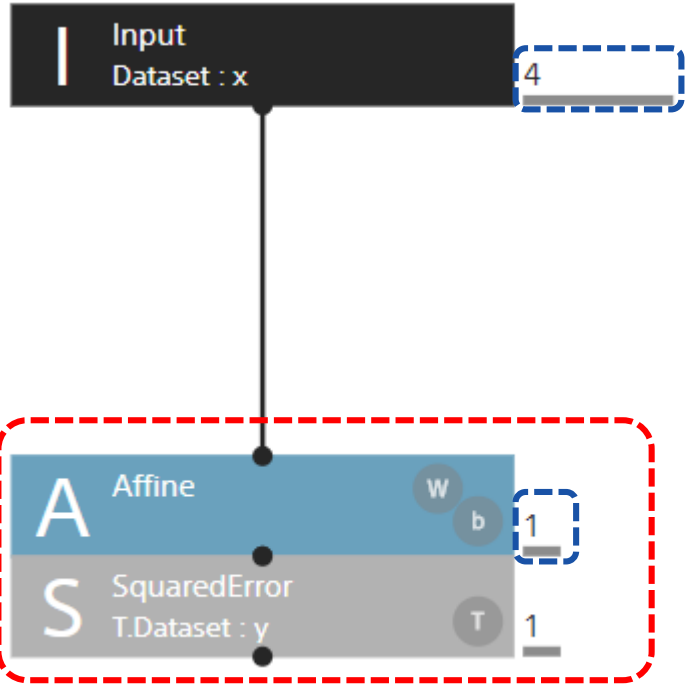
レイヤーの役割

入力データのサイズを設定

予想する数値の個数を設定

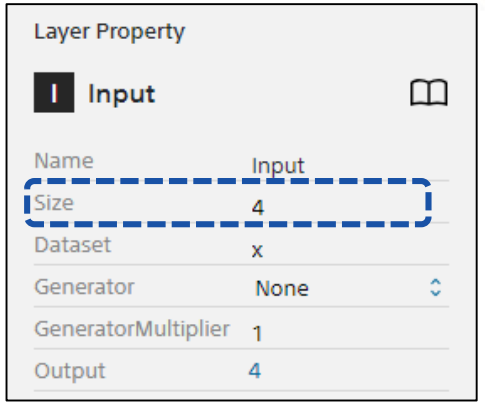
二乗誤差で予測誤差を計算

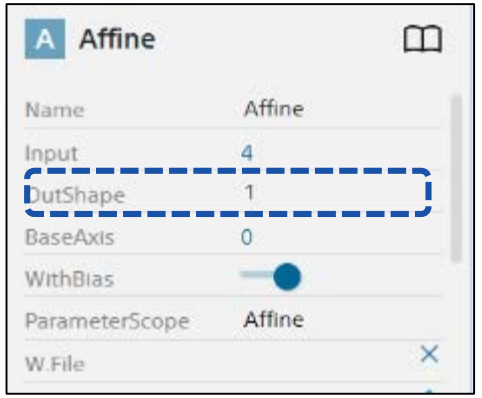
レイヤー配置

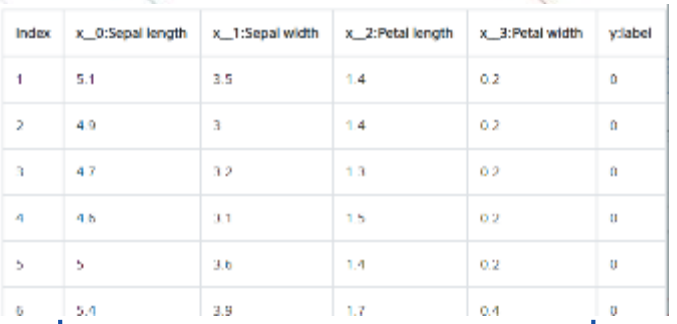


レイヤーの最後の部分だけが分類モデルと異なる

修正箇所







入力データのサイズはDatasetタブで入力カラム数で確認