Neural Network Console クラウド版
スターターガイド
-画像分類編-

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

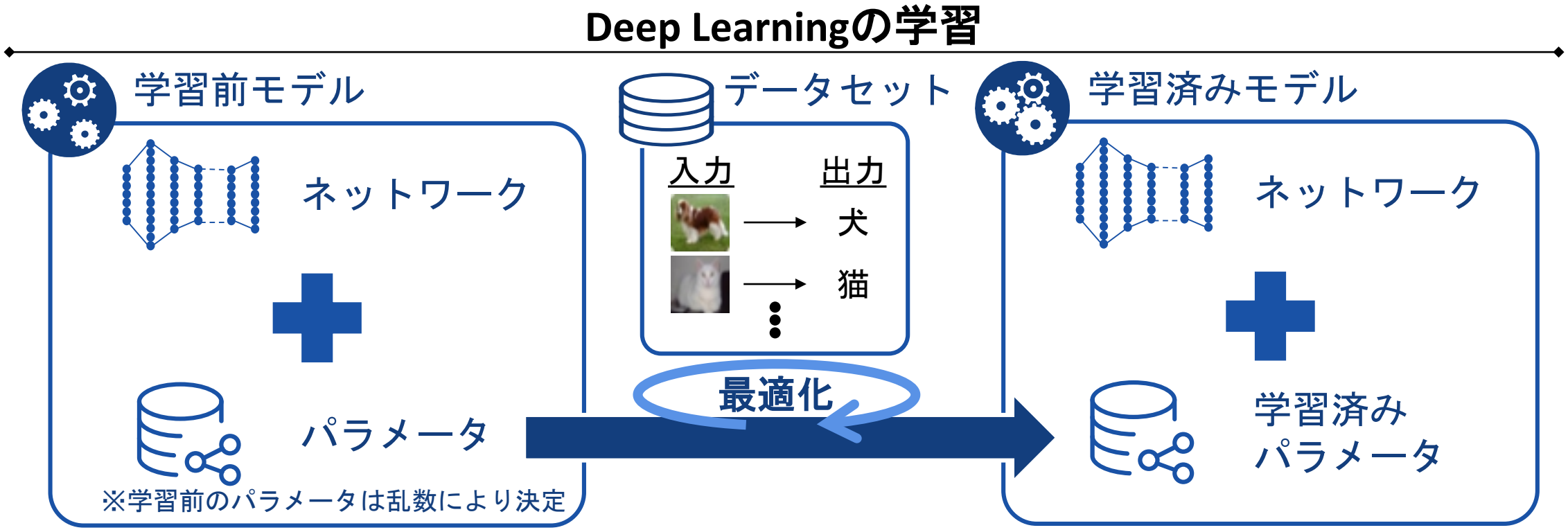
本ドキュメントではご自身で用意した画像データを使って、Neural Network Consoleで画像分類モデルを作成する一連の流れをまとめました。
「画像分類モデル作成」のステップでは過去に画像認識コンテストで優勝した実績のある「ResNet」というモデルを用いる流れになっていますので、Deep Learningモデルの設計ノウハウが無い方でも開発に取り組み易い流れになっています。

目次

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。

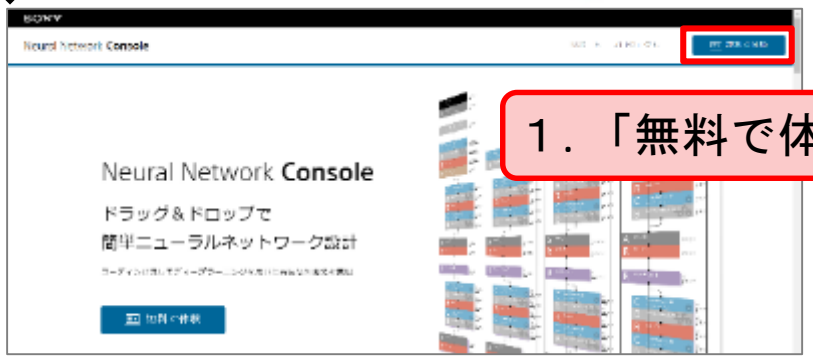


目次

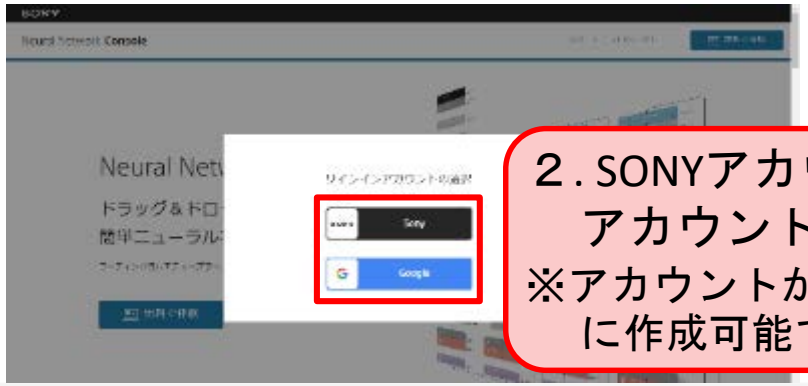
サインインページへの移動

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



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。

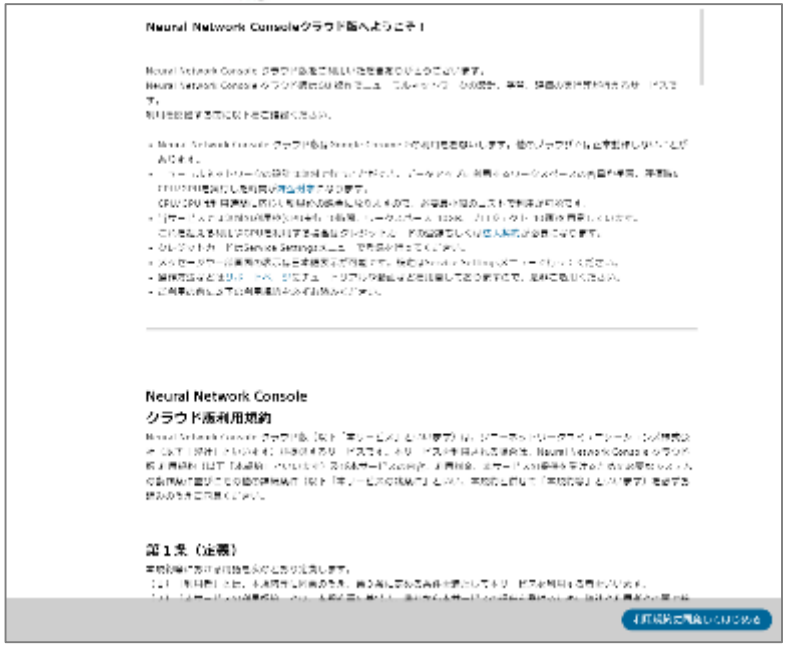
SONYアカウントでのサインイン

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



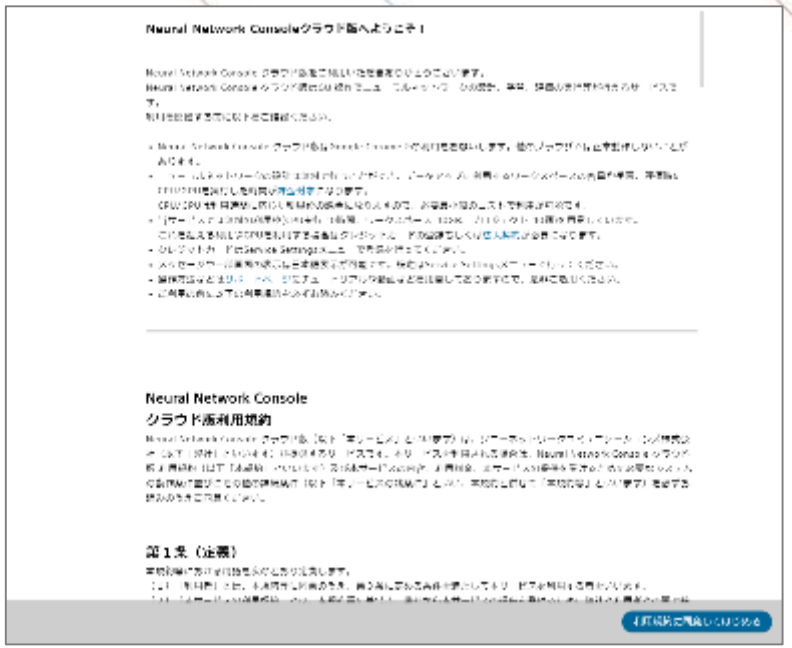
Googleアカウントでのサインイン

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



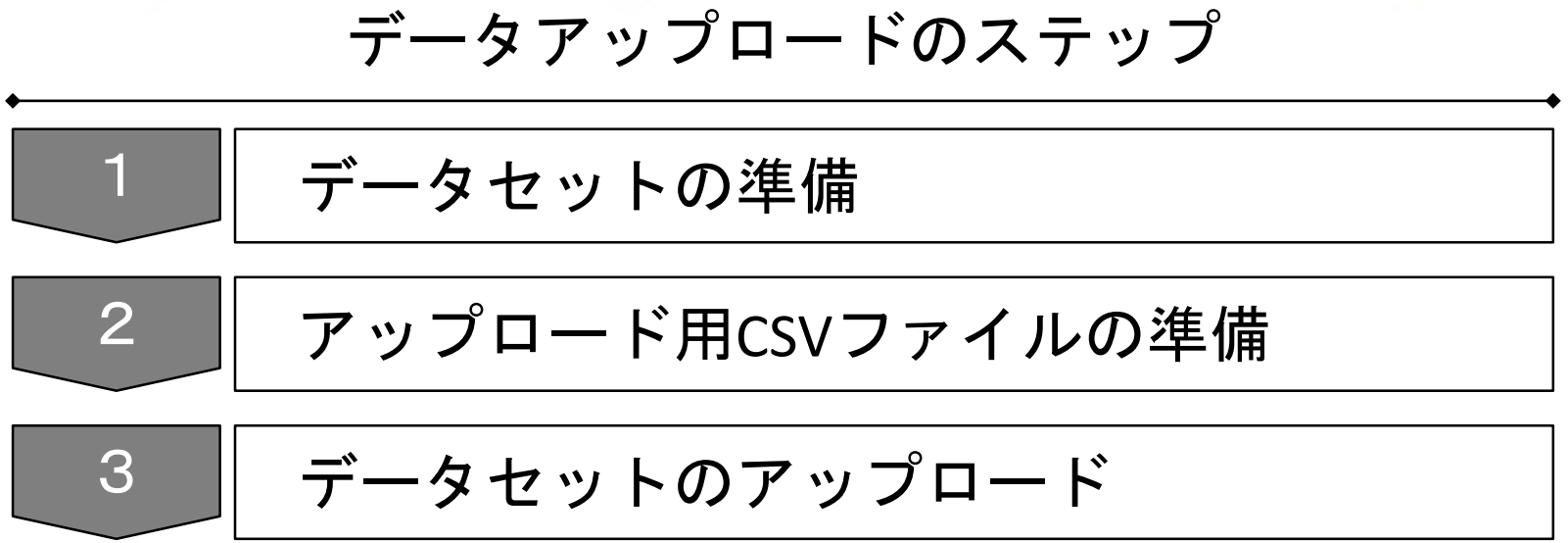
目次

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。

データセットとは入力データと出力データのセットで、画像分類の場合には、入力の画像データと出力の判別ラベルのセットになります。

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。



※モデルを作成するためのデータセットをお持ちでない場合には、Neural Network Console上のサンプルデータまたはウェブ上のオープンデータを利用することも可能です。
Appendixにウェブ上に公開されている代表的なオープンデータをまとめております。

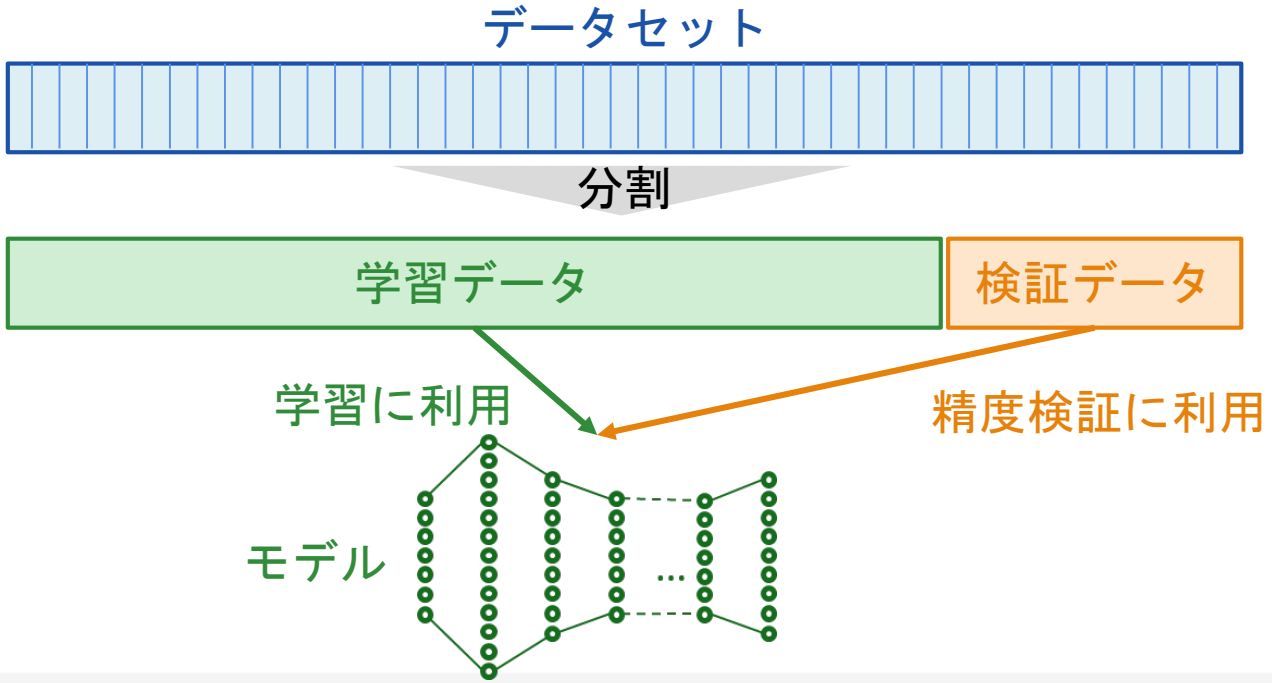
データセットの準備

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)

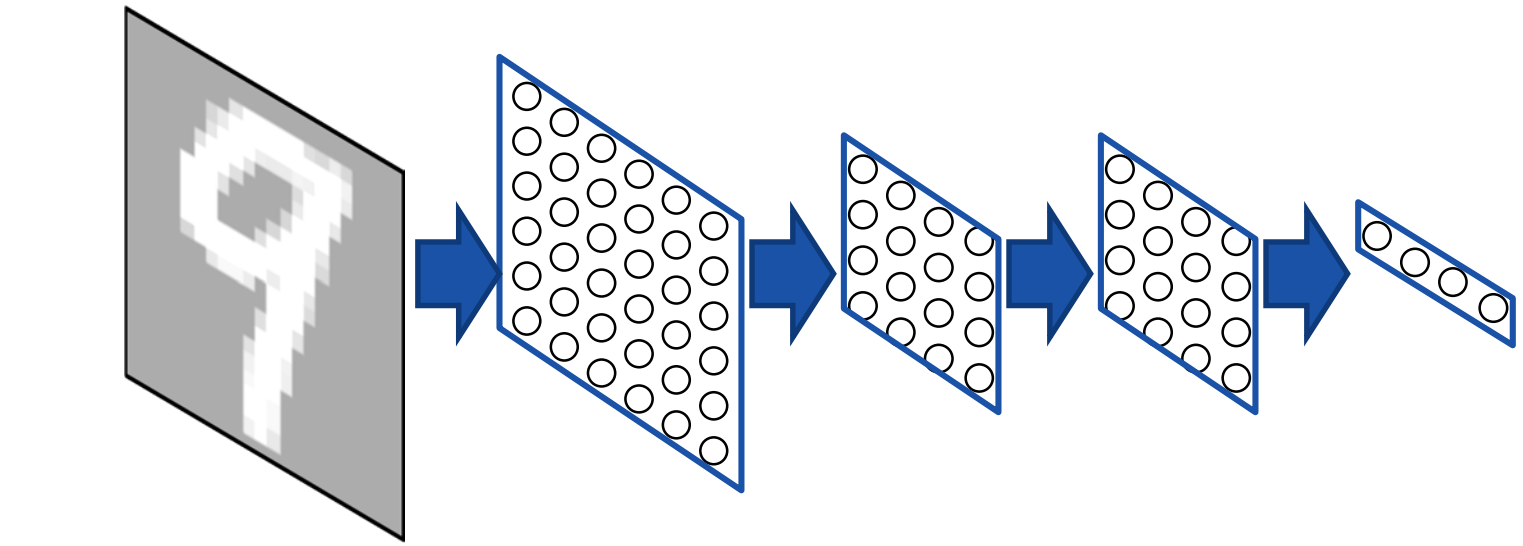


画像サイズ、カラー/モノクロの統一

データセットに画像を利用する場合は、画像サイズとカラー/モノクロを統一する必要があります。
各画像データのサイズが異なっていると、モデルへの入力データが不一致となりエラーとなります。

また、画像サイズは、人が見て認識できる程度に圧縮することをお勧めいたします。
これは入力された画像が数値データ(配列)としてそのままモデルの最初のレイヤーになるため、画像サイズが大きいとモデル自体が大きくなってしまい、学習時間が長くなる、処理速度が遅くなる、などのデメリットをもたらします。

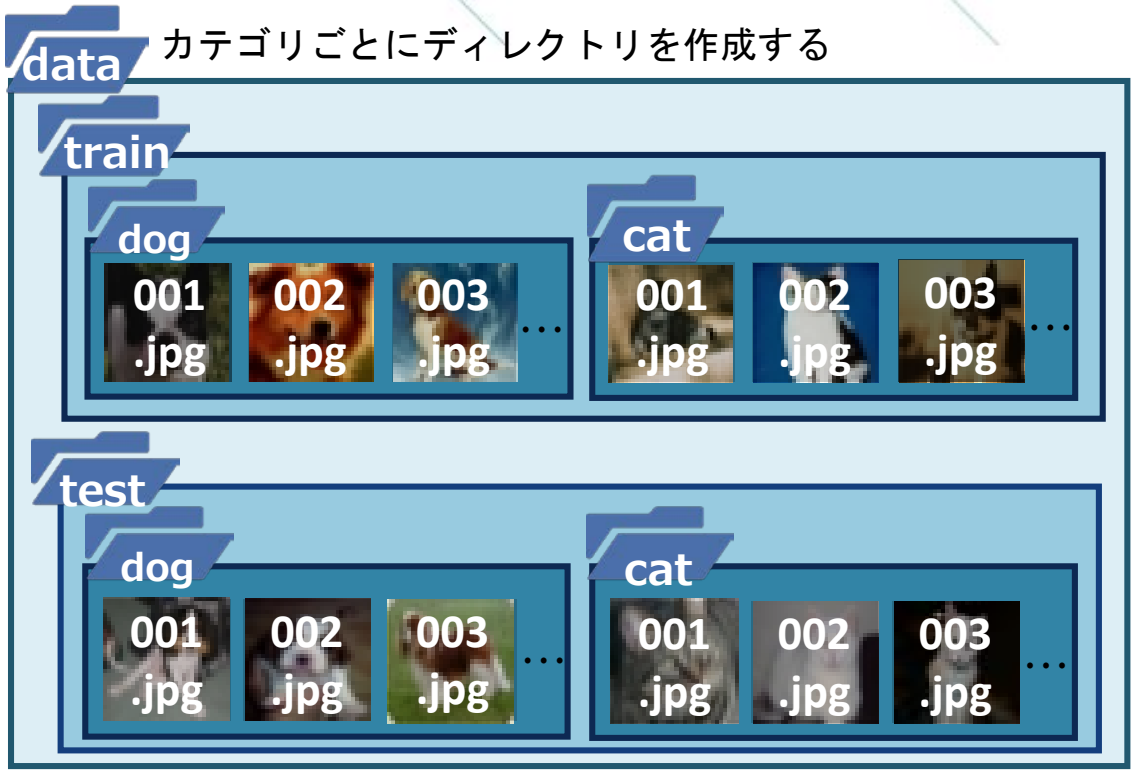
画像を入力としたDeep Learningのイメージ



CSVファイルによるデータセット作成

データセットはCSVファイルの形で入力と出力を取りまとめておきます。CSVファイルの中には、画像データは相対パスを記載し、ラベルデータは事前に数値化し記載します。
CSVファイルは学習用と検証用の２つを準備する必要があります。

画像分類のCSVファイル作成 (犬と猫の2分類のモデル作成時の例)



カラム名のxが入力で、yが出力

trainData.csv

x	y
data/train/dog/001.jpg	0
data/train/dog/002.jpg	0
data/train/dog/003.jpg	0
⋮	
data/train/cat/001.jpg	1
data/train/cat/002.jpg	1
data/train/cat/003.jpg	1
⋮	

testData.csv

x	y
data/test/dog/001.jpg	0
data/test/dog/002.jpg	0
data/test/dog/003.jpg	0
⋮	
data/test/cat/001.jpg	1
data/test/cat/002.jpg	1
data/test/cat/003.jpg	1
⋮	

画像は相対パスを記載　　ラベルは事前に数値化

この例ではdogを0、catを1と定義し、データを準備

データセットのアップロード

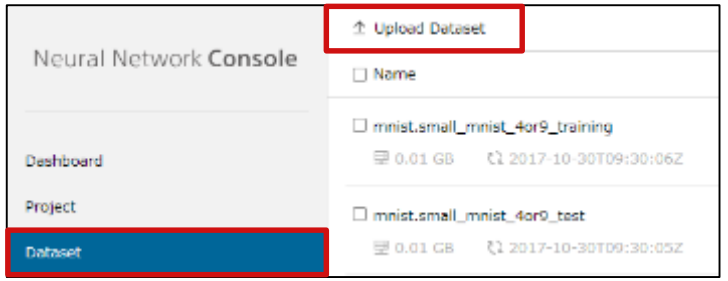
Neural Network Consoleへのデータアップロードには専用のアップローダを利用します。
アップローダ上で準備したCSVファイルを指定し、データをアップロードします。

１．アップローダの取得

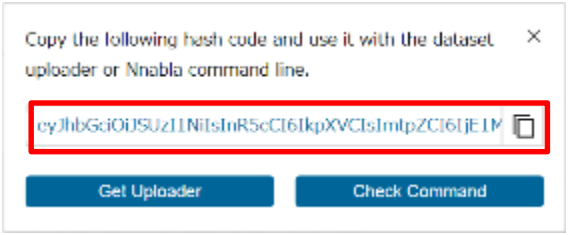
✓以下のリンクからアップローダをダウンロード
https://support.dl.sony.com/docs-ja/データセットアップロードツールのダウンロード/

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

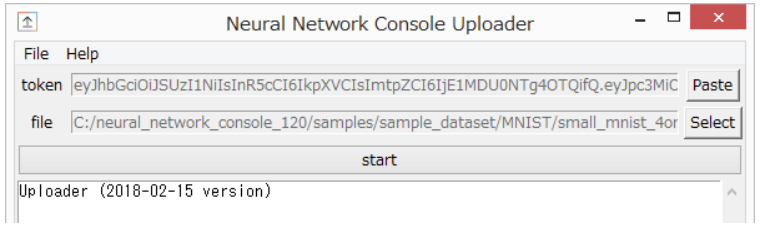


✓ポップアップ画面に表示されるアップロードキーをコピー



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
✓Startをクリックし、アップロードを実行



※画像データはCSVに含まれるパスを参照し自動的にアップロードされます。

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。

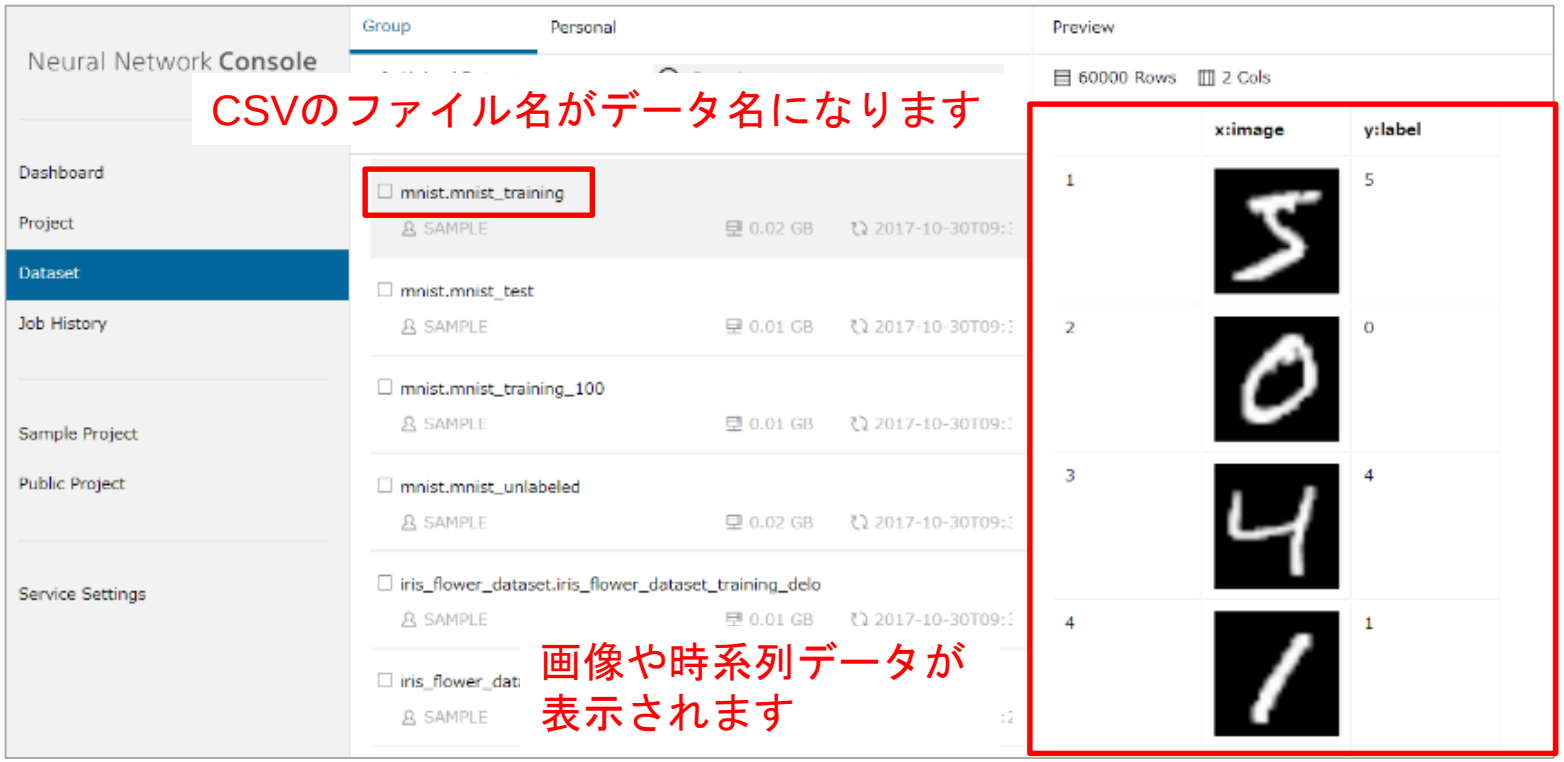
アップロード後のデータセットの例

アップロード時のCSVファイル

mnist.mnist_training.csv

x:image	y:label
data/train/5/0001.jpg	5
data/train/0/0001.jpg	0
data/train/4/0001.jpg	4
data/train/1/0001.jpg	1
⋮	⋮

データセット一覧に追加されたデータセット

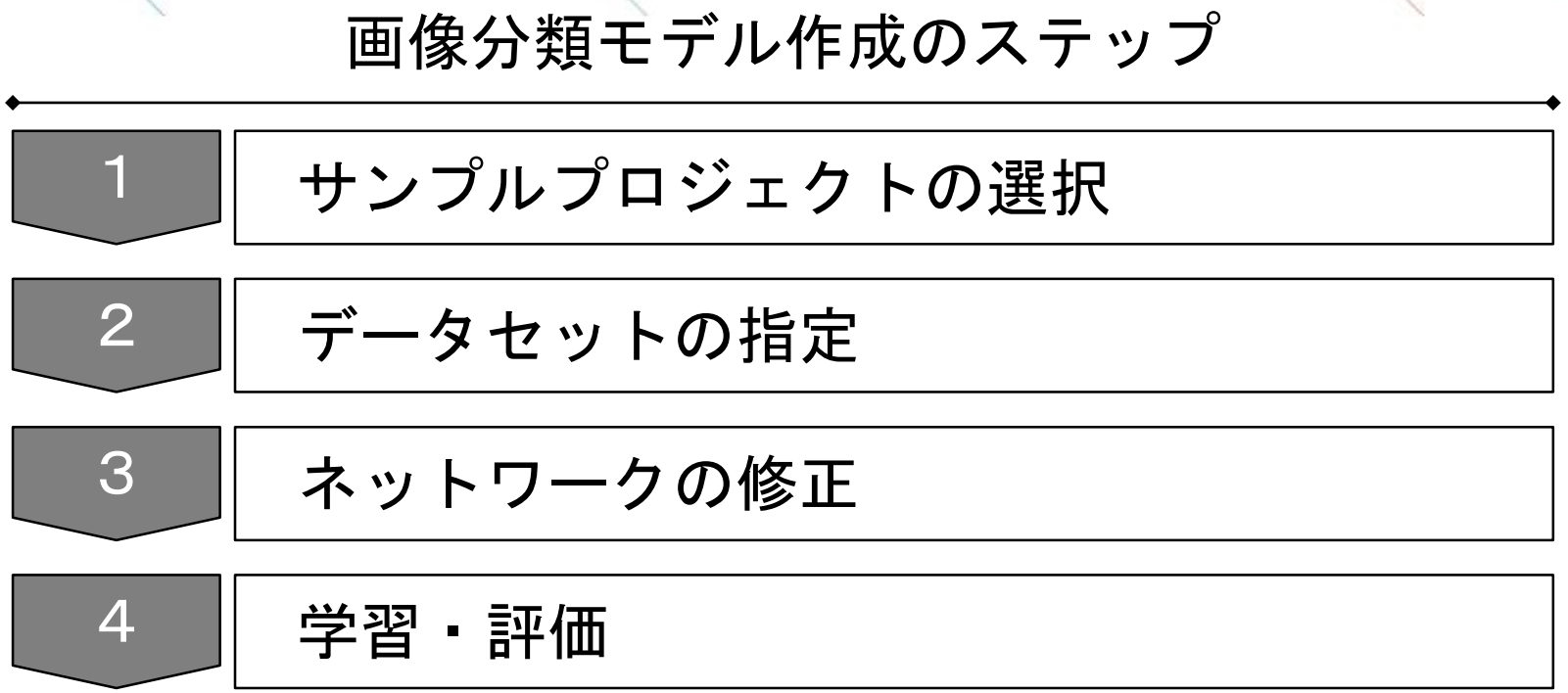


目次

画像分類モデル作成のステップ

Neural Network Consoleでは新規にモデルを作成することも可能ですが、

このドキュメントでは、過去に画像認識コンテストで優勝した実績のある「ResNet」というモデルをベースにすることで、比較的容易にモデル作成を可能にします。

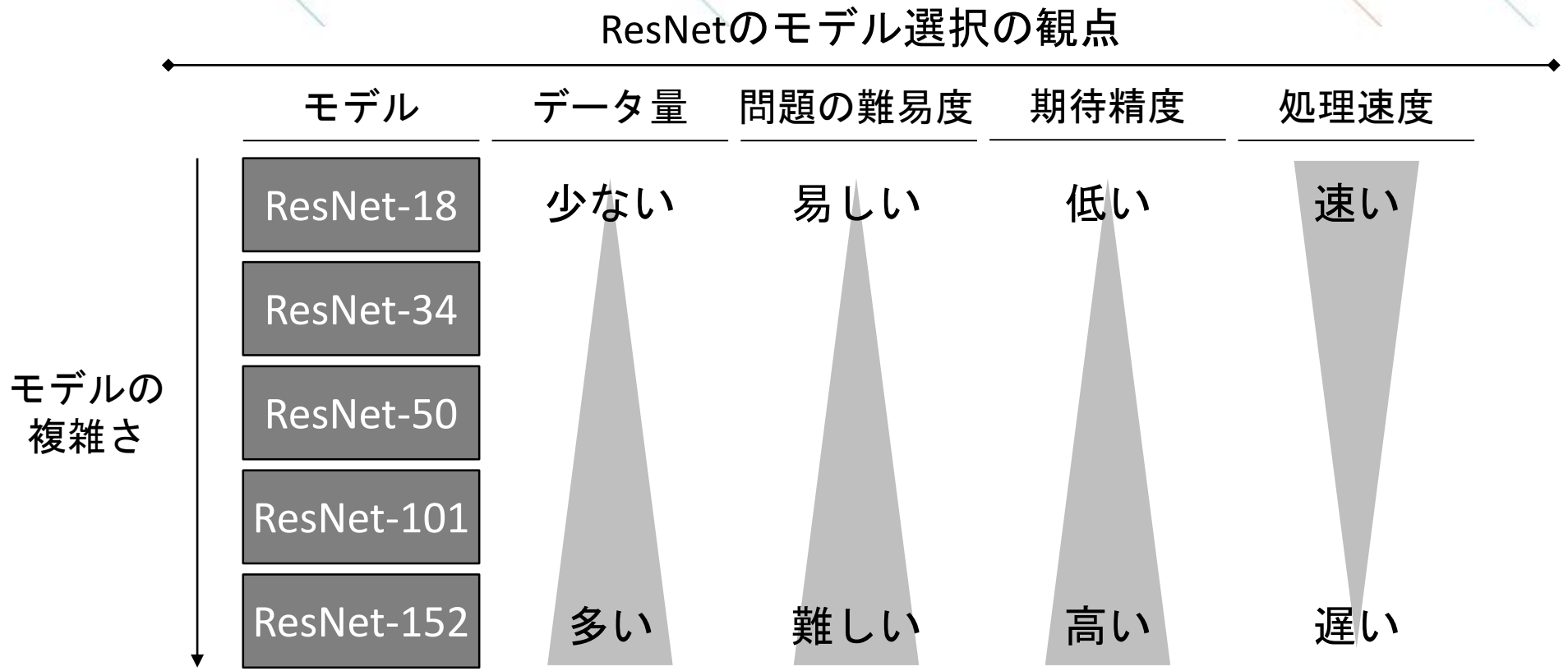


ネットワークモデルの選択

ResNetはモデルの複雑さに応じて5種類のサンプルがあり、以下に示す観点を参考にモデルを選択します。

モデルが複雑であればより高い判別精度を期待できますが、処理速度が遅くなるなどのデメリットもあります。

条件が明確でない場合は、期待精度と処理速度のバランスを考慮し、まずはResNet-50をお勧めします。

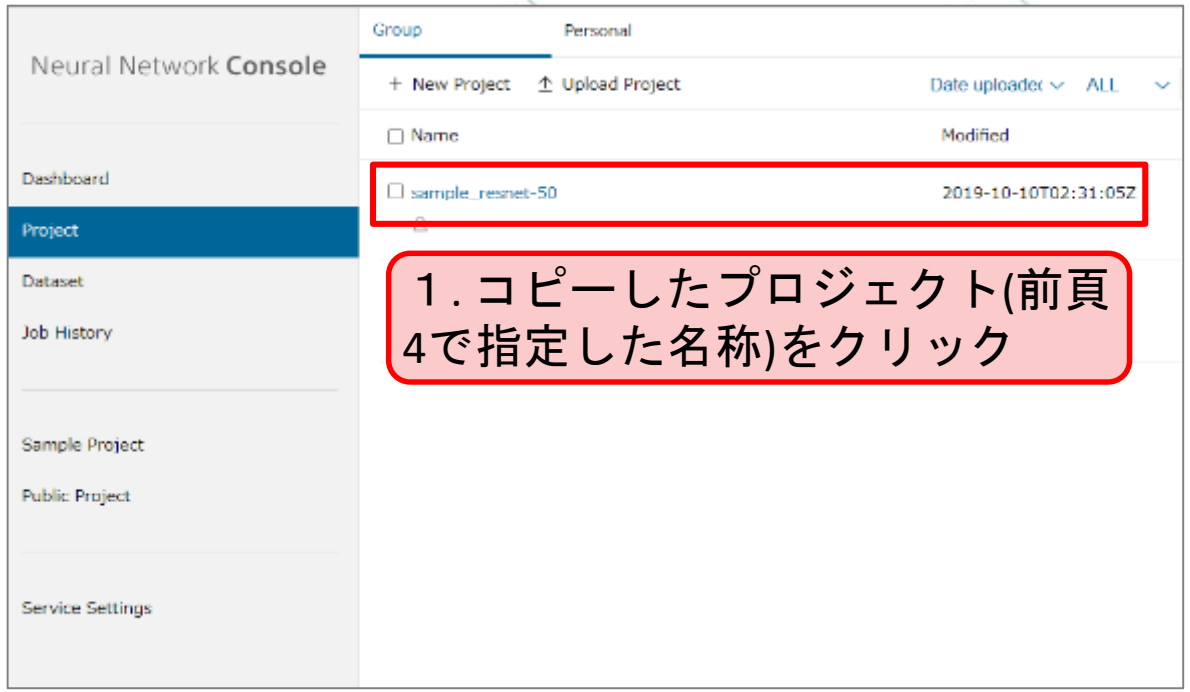


サンプルプロジェクトのコピー

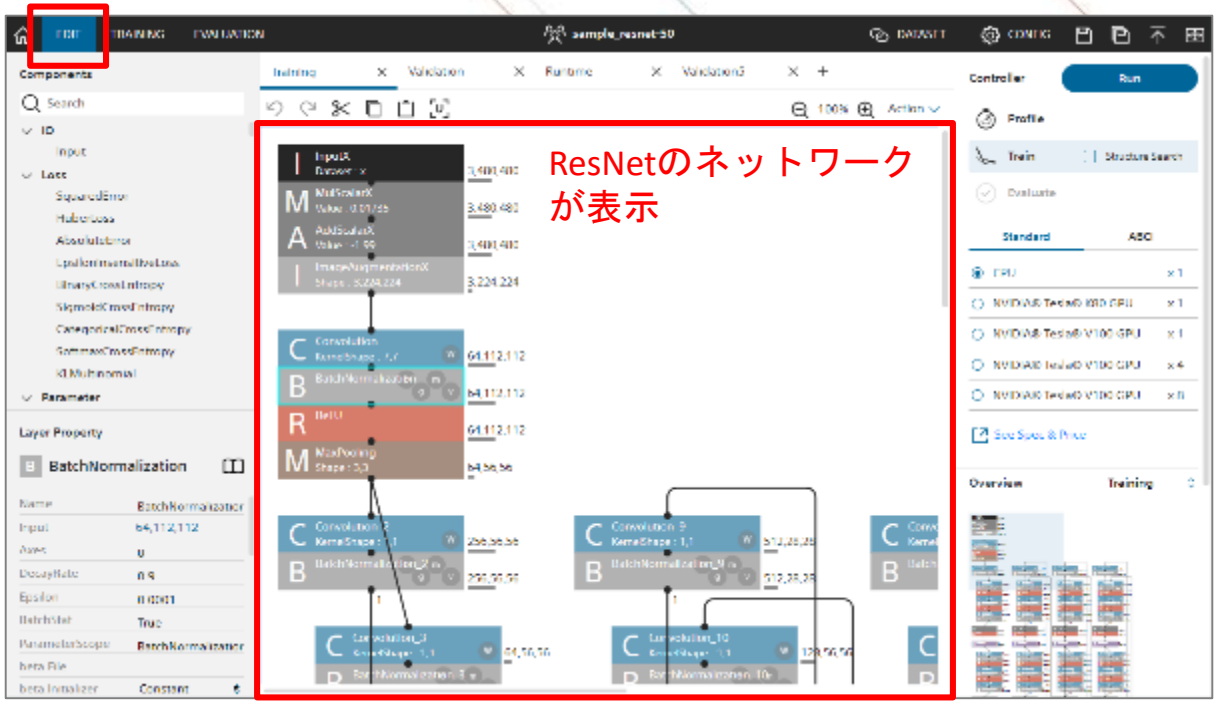
利用するResNetのモデルを決定したのちに、そのサンプルプロジェクトをコピーします。

プロジェクトの起動

コピーしたプロジェクトをクリックし、プロジェクトを起動します。
以下のようなResNetのネットワークが表示されていることを確認します。



Editタブ: ネットワークを作成するページ



データセットの指定

プロジェクトに準備した事前にアップロードしたデータセットを紐づけます。

1. Datasetタブをクリック

2. Trainingをクリック

3. Not Setをクリック

4. 一覧から学習に用いるデータセットを選択

6. Validationをクリックし、3～5と同様の手順を実施

5. リンクマークをクリック※

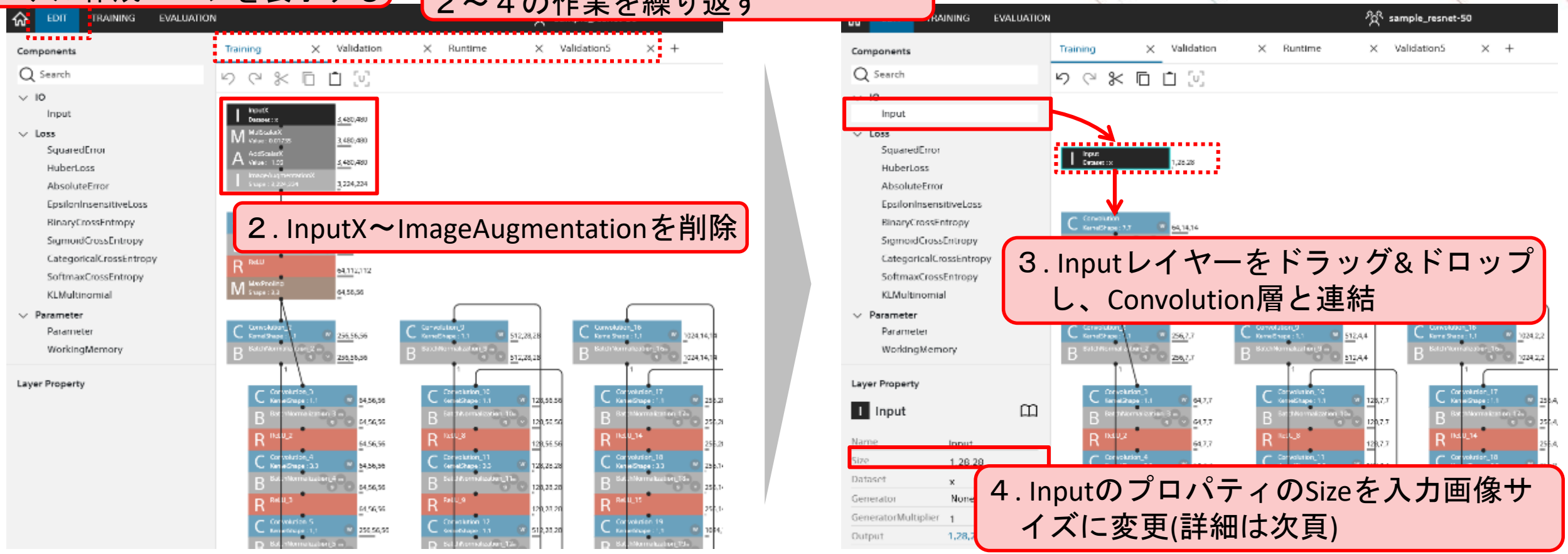
※ブラウザの拡大率によって表示されないことがあります。
表示されない場合は表示の縮小をお試しください。

入力層(画像サイズ)の修正

入力画像の加工等を行うレイヤー(Deep Learningでネットワークを作成するための関数)は設定が複雑なため、本ドキュメントではこれらを除いたネットワークを作成します。

1. Editタブをクリックし、モデル作成ページを表示する

5. ４つのネットワークタブ全てで２～４の作業を繰り返す



2. InputX～ImageAugmentationを削除

3. Inputレイヤーをドラッグ&ドロップし、Convolution層と連結

4. Inputのプロパティの Sizeを入力画像サイズに変更(詳細は次頁)

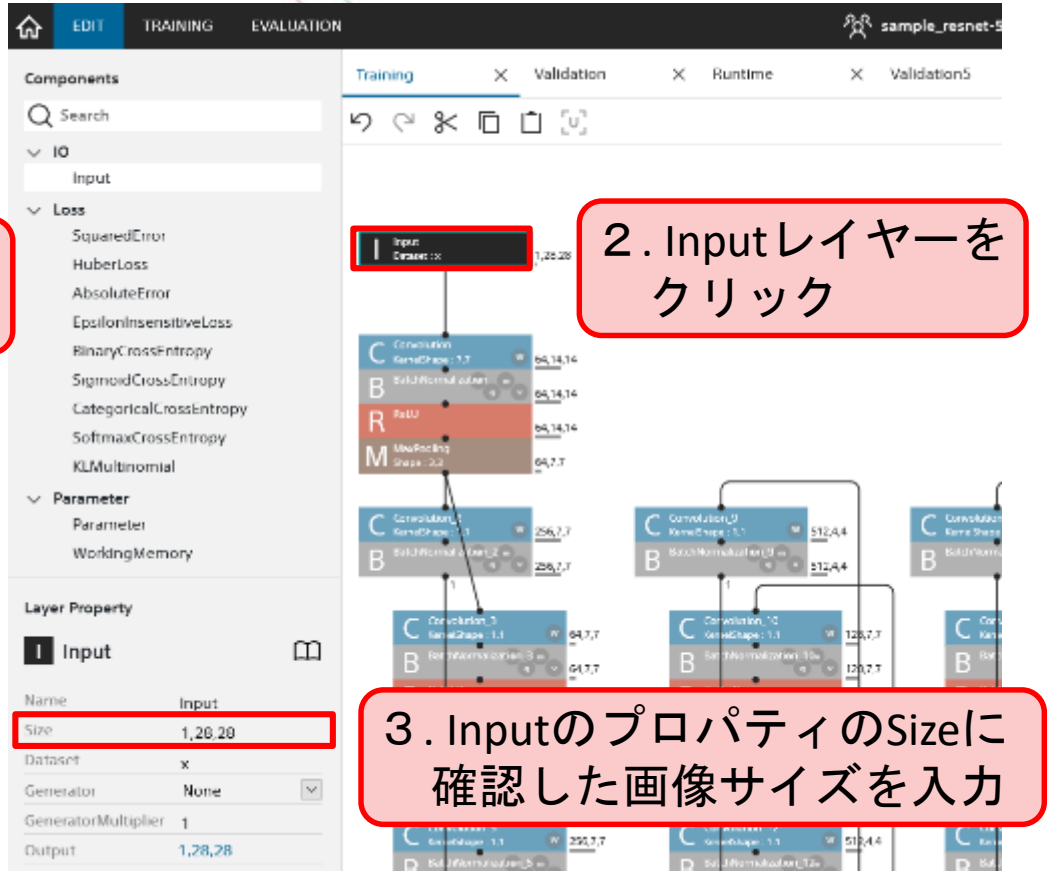
※大きなネットワークの表示をするので、初回のネットワーク表示の際やタブを切り替えた際に、表示までに少し時間がかかります。

入力画像サイズの設定

Datasetタブに移動し、画像データのサイズを確認し、Inputレイヤーのサイズとして入力します。

指定したデータサイズと入力画像のサイズが合わない場合にはエラーとなるため、学習データの中に画像サイズが異なるものが含まれるとその時点で学習がストップしてしまいます。

1. Datasetタブに移動し、画像データのサイズを確認



2. Inputレイヤーをクリック

3. Inputのプロパティのサイズに確認した画像サイズを入力

画像データのサイズについて

入力サイズの3つの数字はカラー(3)orモノクロ(1)、縦サイズ、横サイズを順に示しています
カラー画像の場合にはRed, Green, Blueの3枚のレイヤーがあると考えるため、最初の数字が3となります

例. (3, 256, 128)

256ピクセル

128ピクセル

出力層(分類数)の修正

ネットワークの最後の部分のAffineレイヤーで画像分類の分類数を設定します。

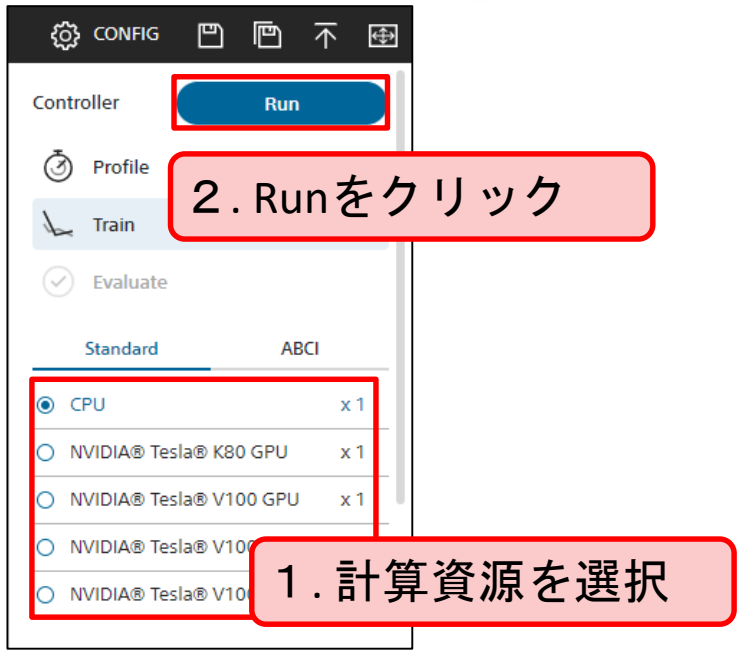
学習の実行

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。

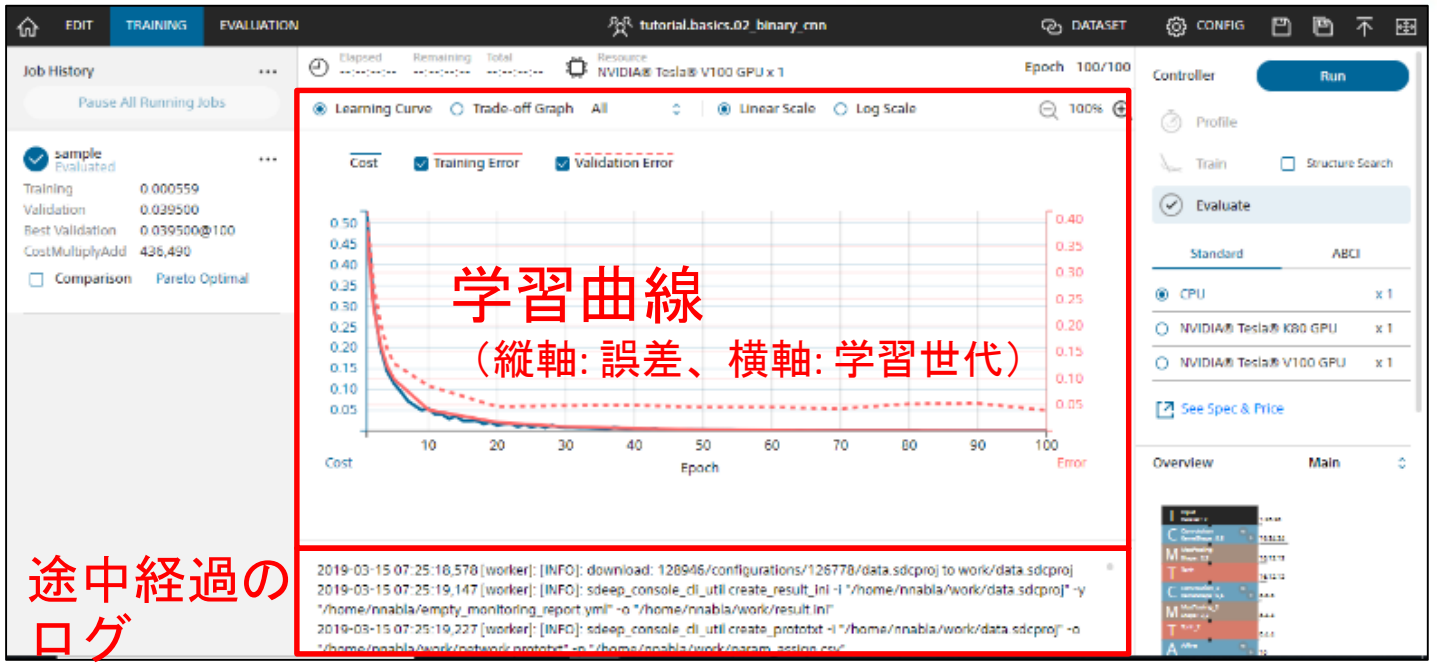
GPUを選択すると、高速に学習を行うことができます。（参考: 学習環境と処理時間）

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



※データ件数が少ない場合には、バッチサイズのエラーがポップアップで表示される場合があります。この場合には、バッチサイズの変更に従って、バッチサイズをデータ件数よりも小さい値に変更してください。

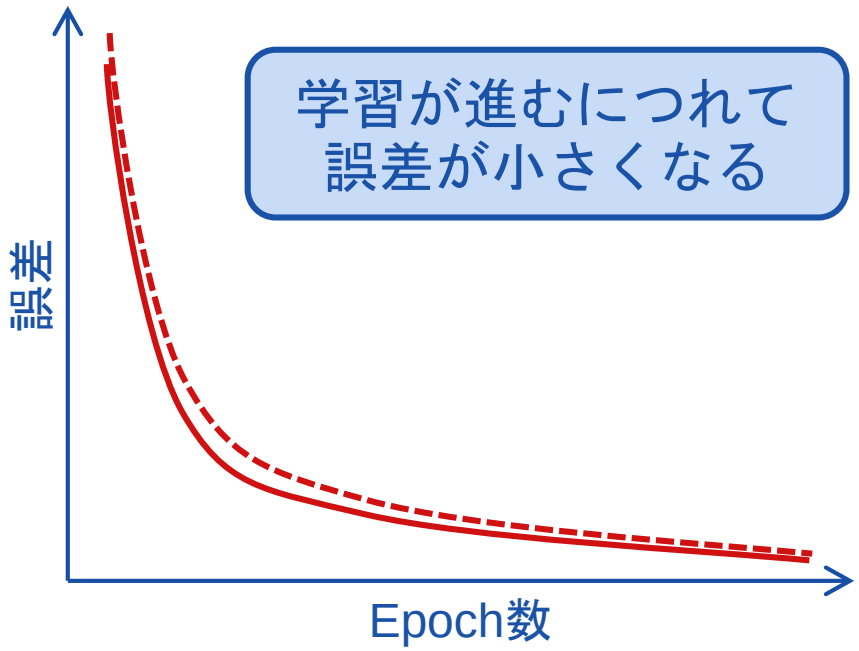
学習曲線の読み取り方

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。

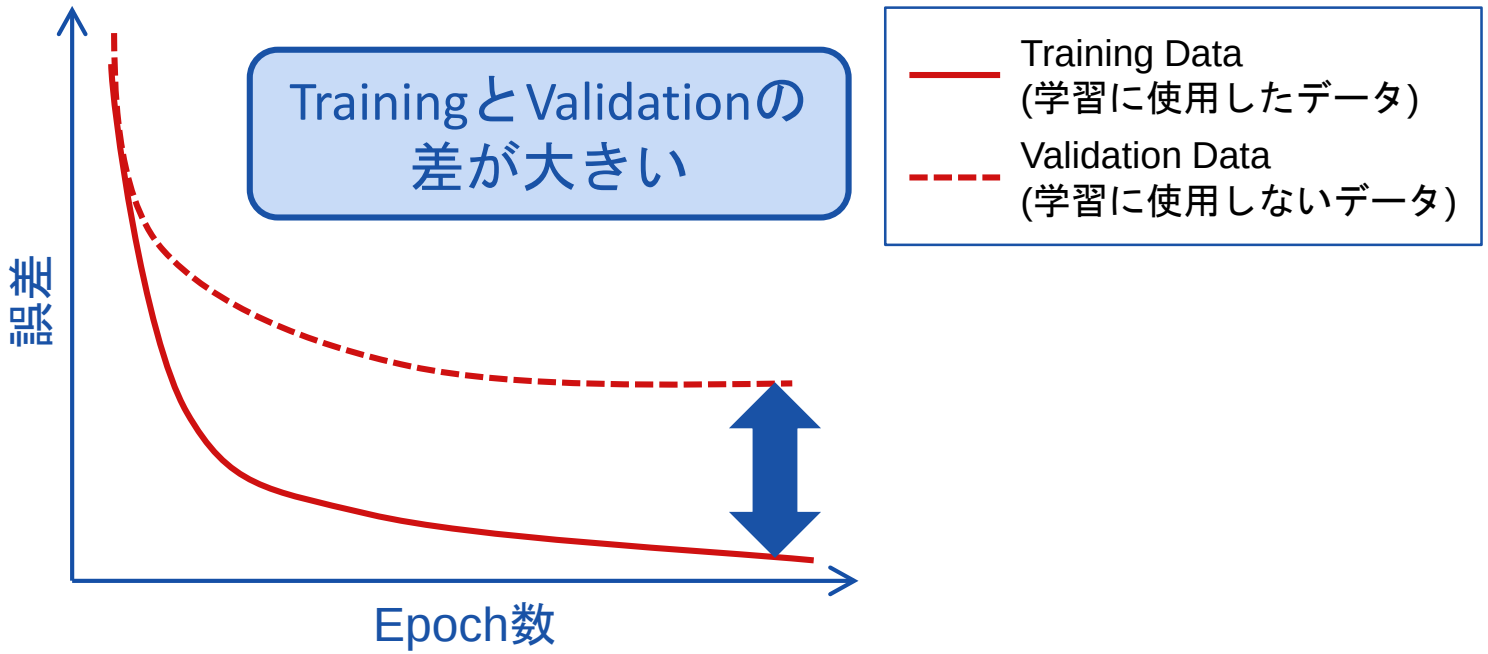
TrainingとValidationの差が大きい場合(過学習)は、モデルがTraining Dataに特化し過ぎた状態(教科書を丸暗記した場合に応用問題が解けないのと似た状態)です。

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、ResNetの層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



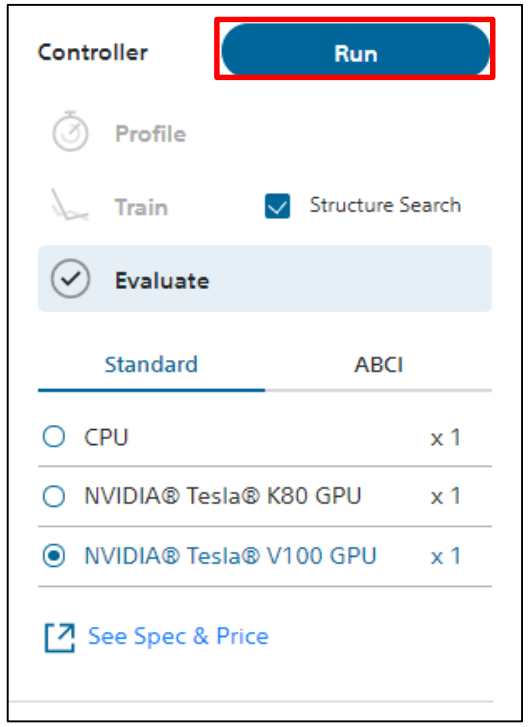
悪いモデルの学習曲線（過学習）



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判定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

クリックして選択

ページを変更し全ての結果を確認可能

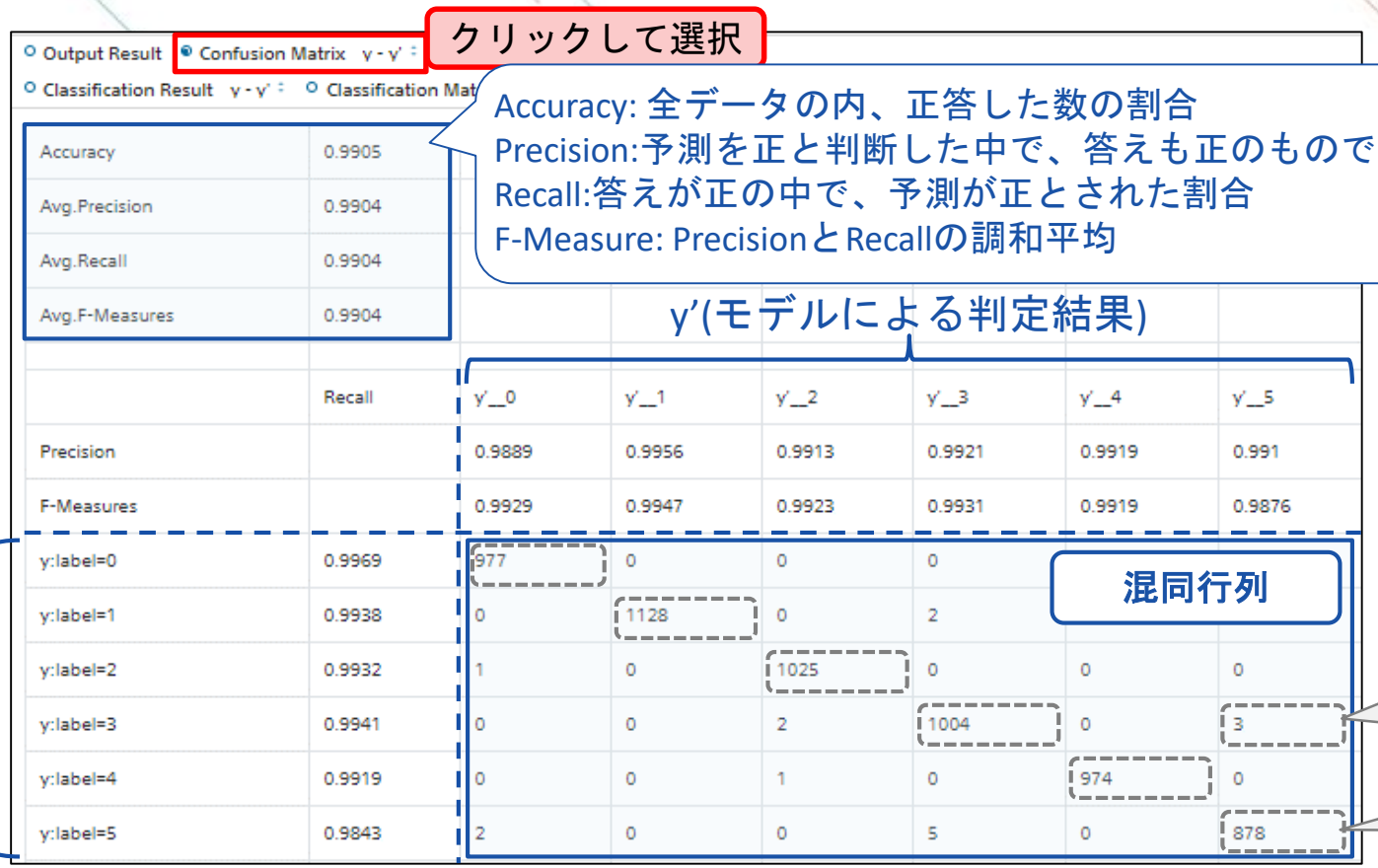
追記カラム名について

- 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

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



Accuracy	0.9905
Avg.Precision	0.9904
Avg.Recall	0.9904
Avg.F-Measures	0.9904

	Recall	y'__0	y'__1	y'__2	y'__3	y'__4	y'__5
Precision		0.9889	0.9956	0.9913	0.9921	0.9919	0.991
F-Measures		0.9929	0.9947	0.9923	0.9931	0.9919	0.9876
y:label=0	0.9969	977	0	0	0		
y:label=1	0.9938	0	1128	0	2		
y:label=2	0.9932	1	0	1025	0	0	0
y:label=3	0.9941	0	0	2	1004	0	3
y:label=4	0.9919	0	0	1	0	974	0
y:label=5	0.9843	2	0	0	5	0	878

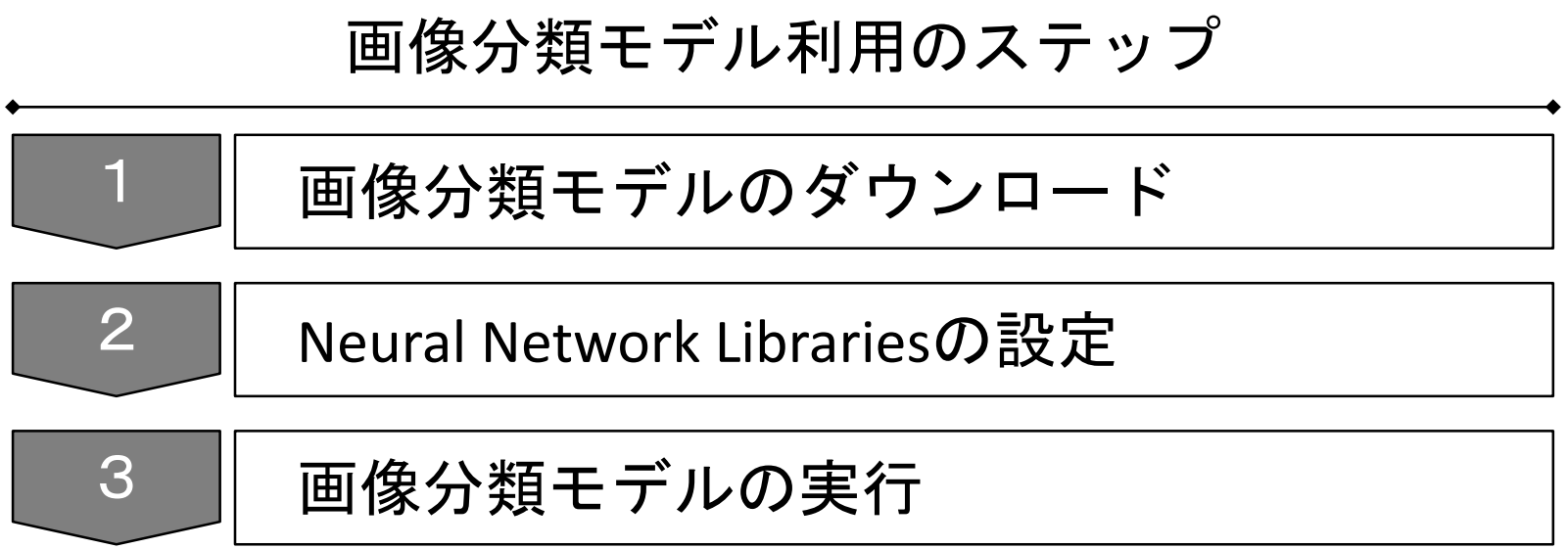
目次

画像分類モデル利用のステップ

Neural Network Consoleから画像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など様々な方法で画像分類モデルが実行可能となります。

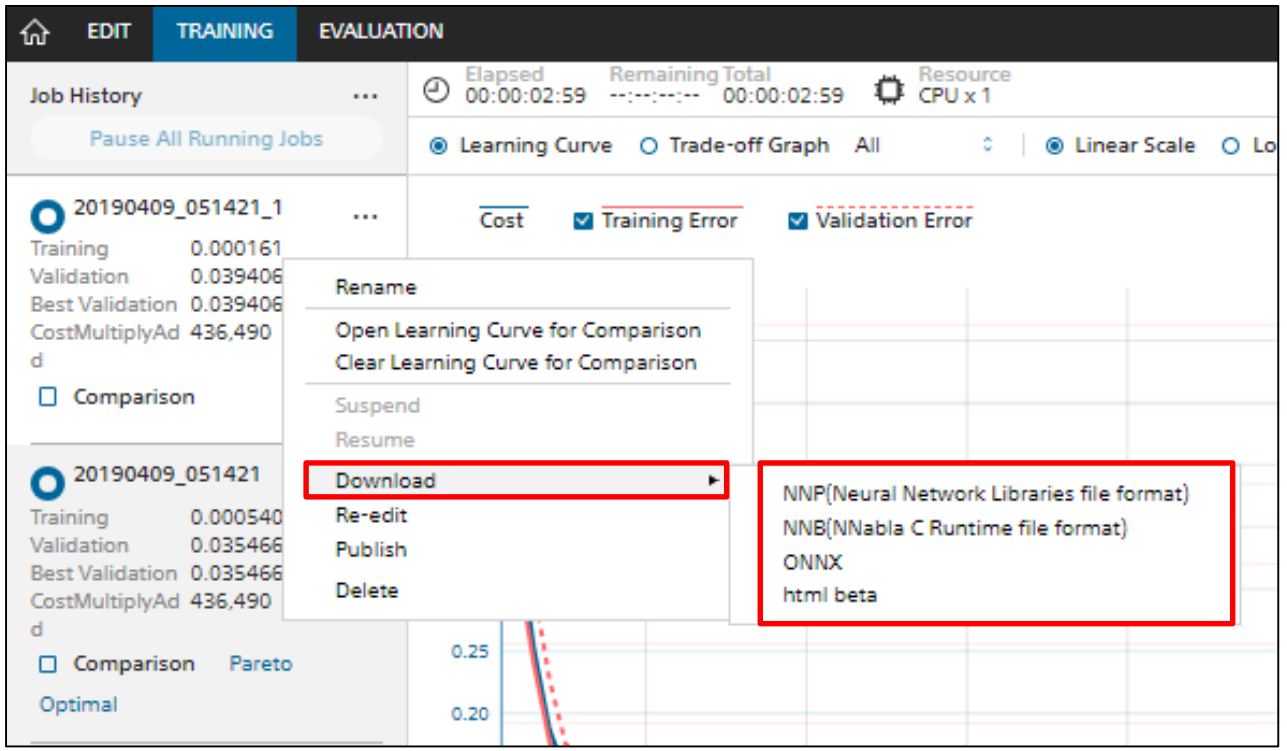


画像分類モデルのダウンロード

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/	次頁に解説あり
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	

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(img) * (1.0 / 255.0)
y.forward(clear_buffer=True)
# 推論結果の表示
print(y.d[0])
```

Appendix

SONYアカウントの取得方法

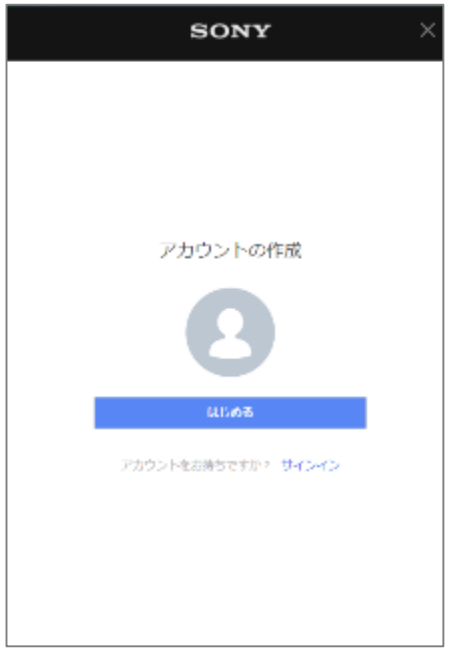
アカウント作成ページに移動し、メールアドレスやパスワードなどを設定します。

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アカウントの取得方法

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アカウントの取得方法

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

代表的なオープンデータ

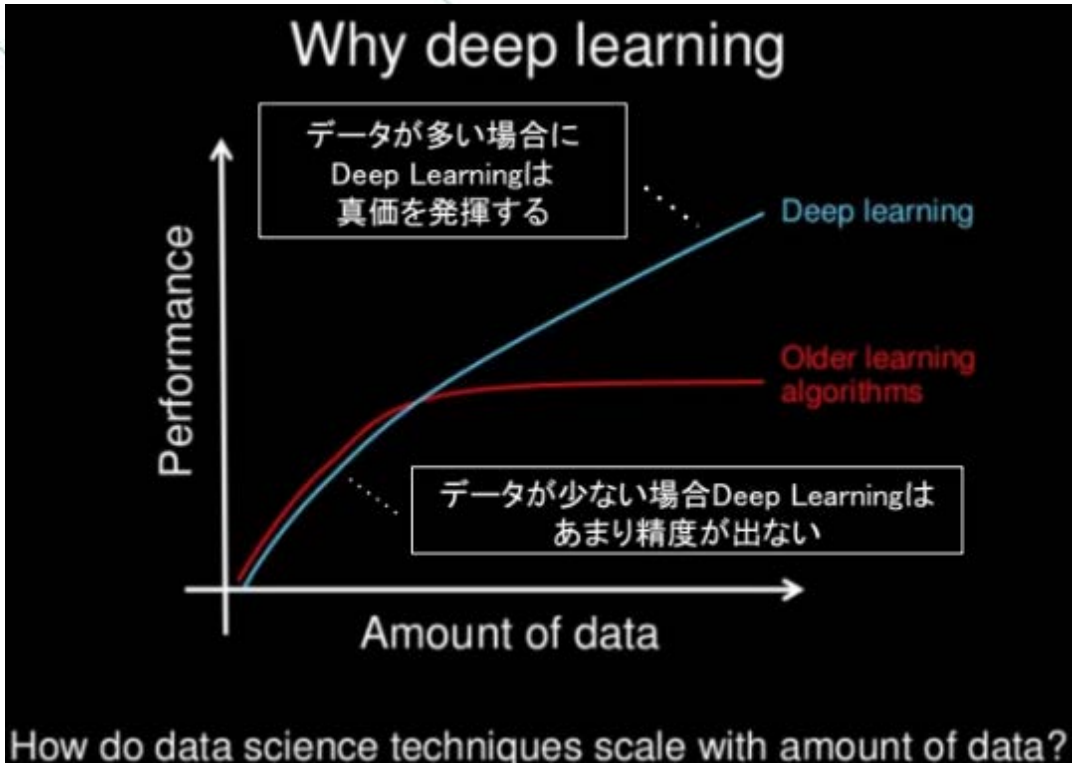
ウェブ上にある画像分類で利用可能なオープンデータとして以下のデータセットがあります。
このほかにもUCI Machine Learning RepositoryやKaggleには、様々なデータが公開されています。

	データセット名	概要	画像枚数	URL
1	ImageNet	約2万種類に分類された一般物体認識の画像データセット	約1,400万枚	http://www.image-net.org/
2	Cifar-10/Cifar-100	10または100クラスに分類された一般物体認識の画像データセット	6万枚	http://www.cs.toronto.edu/~kriz/cifar.html
3	DAGM2007	6種類の工業用部品の表面に傷を加工した画像データセット	6,900枚	http://resources.mpi-inf.mpg.de/conferences/dagm/2007/prizes.html
4	SDNET2018	コンクリートのひび割れの有無の画像データセット	56,000枚	https://digitalcommons.usu.edu/all_datasets/48/
5	Food-101	101カテゴリに分類された食べ物の画像データセット	約10万枚	https://www.vision.ee.ethz.ch/datasets_extra/food-101/

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。



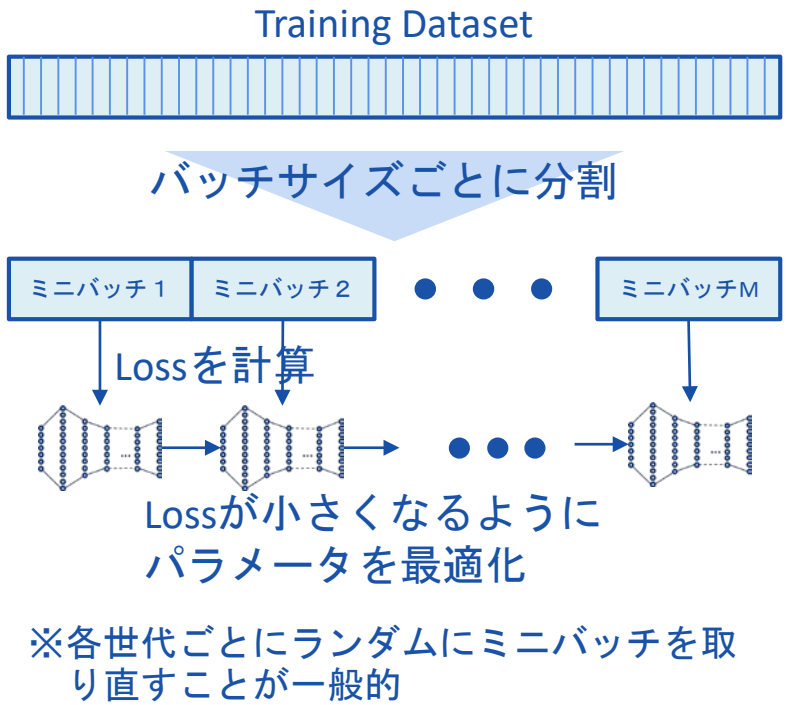
出典: https://www.slideshare.net/ExtractConf/andrew-ng-chief-scientist-at-baidu

バッチサイズの変更

バッチサイズとはモデルのアップデート時に用いるサブデータセット(ミニバッチ)のデータ件数です。
バッチイズはCONFIGタブから変更することができます。

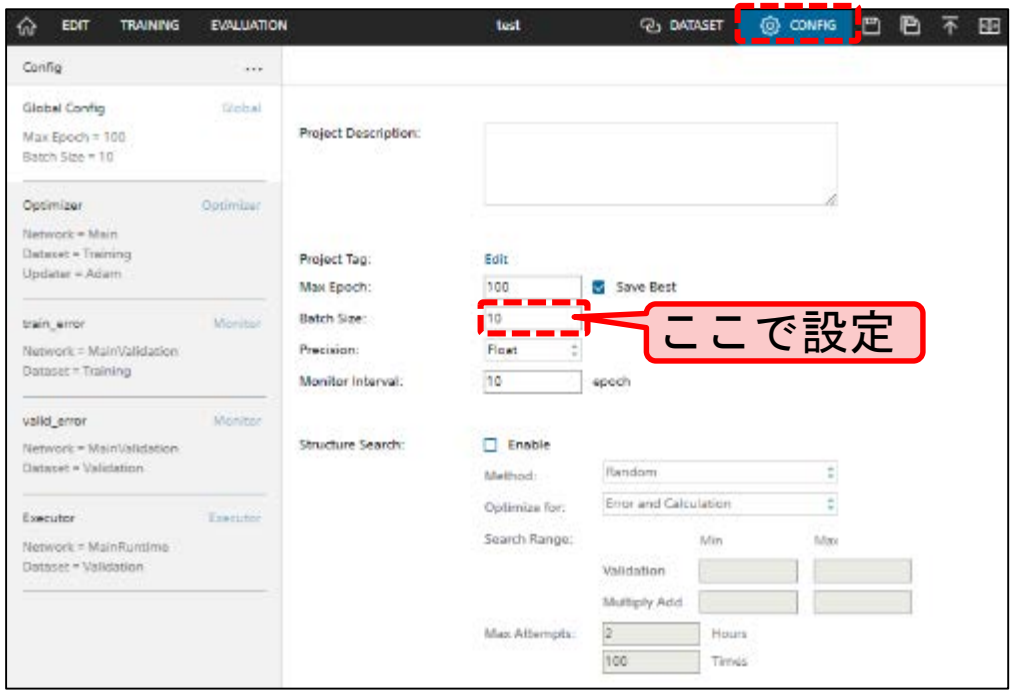
バッチサイズがデータ件数よりも大きすぎるとエラーが発生しますし、極端に大きな数値はメモリ不足でのエラーの原因にもなります。一方で小さすぎるとパラメータ更新が頻発し、学習が非効率になります。

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



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（一例）

	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