# Neural Network Console クラウド版 スターターガイド -物体検出編-

ソニーネットワークコミュニケーションズ株式会社



本ドキュメントではNeural Network Console(NNC)で物体検出モデルを作成する一連の流れをまとめました。

サンプルプロジェクトを利用してモデル作成を行う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取り組み易い構成になっています。

また、ウェブにも物体検出の解説動画(<u>実践Deep Learning:物体検出</u>)を準備しておりますので、モデルのネットワーク設計の詳細を理解されたい場合には、こちらも併せてご覧ください。





### Deep Learning モデルを作るとは

Deep Learning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す。モデルは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。

#### Deep Learningの学習



### Deep Learningを用いた物体検出

Deep Learningを用いた物体検出では、画像の中にある複数の物体を認識し、おおよその位置がわかります。 Deep Learningを用いた画像認識技術には様々なものがあり、より正確な位置や形状を知る場合には領域抽出 が適切ですし、画像そのものを分類する場合には画像分類で十分です。

様々な画像認識技術の中から解決したい問題やモデル作成の期間・工数に応じて適切な手法を選択する必要 があります。

Deep Learningを用いた画像認識技術

画像分類	物体検出	領域抽出
入力画像の種類を予測する	入力画像の中にある物体の種類とその 位置を予測する	ピクセル単位で入力画像を物体の種類 を予測する
イヌ	Lemon	

## 物体検出の基本的な入出力

物体検出では画像データを入力として、検出対象である物体のラベルと物体を囲う長方形(バウンディング ボックス)を出力します。バウンディングボックスを定義するために、その中心位置と高さ、幅を記載するこ とが一般的です。







### サインインページへの移動

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



## 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<u>SONYアカウントの取得方法</u>に記載があります。

SONY X	Neural Network Consoleクラウド陥へようこそ)
	Noter Instruct Consists グラウド急速に通信した注意意味ないがようにないます。 Heard Servers Consists クラウド満定的に使用する。 てんそう ベラーのの数字、単葉、加速の支付すべきたのか 「ついて す。 別目目的はずきないなりたさ時間からない。
サインイン ーDのアカウントで、Sonyグループの復数サービスへアクセス もっと詳しく	<ul> <li>Marca Antonia Converte ディアド係は2000年になったのでは目前を認いします。他がプラウドへには含むすいことが みられま。</li> <li>ハー にはたいトリームの設定ではおけたいにとなった。ビータック、A. 利用くないークスペースの美国が構成、運動に 2000年10日にはないためにもなりないます。</li> <li>クロームの大きな「おいま」をついたので、Alexa Antonia Antonia</li></ul>
サインインID youraddress@example.com	
D サインインIDを記憶する パスワード 🔞	Neural Network Console クラウド振利用鉄約 Manal Matania (asser ヴァロット) キャービスリン(Miggi) (A. Mingaria (Mingaria (Mingaria)) そしたす (Min House (Mingaria 70) ビスチャータリードス)中国人の活動会社、Manal Analysia (Second Order)
•••••••	時には月時時には11日におお茶は、べいないない。ならればモードなの時代には月前後、はサービスの時代をまたか。ないな気をないない人 の気候が「空い」ではないないない。 読みのなきまでは良くいかい。
サインイン サインインでお回りですか?	部1集(注意数) 中期時間でありまたにために見します。 1日、日日日には、本地内市に対象のため、自分和に互体なられた中にしてもり、ビスを利用する目的といえた。 1日、日日日には、本地内市に対象のため、自分和に互体なられた中にいるのでありません。またので目前ので目のか
	【ITARE NO. LICE A

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

~	and the second
	ログイン
	「sony.com」(こ移動
	メールアドレスまたは電話語号
	xxxxx@gmail.com
	メールアドレスを忘れた場合
	音声または画面上のテキストを入力
	統行するにあたり、Google はあなたの名前、メール7 ドレス、言語設定、プロフィール写真を sony.com と 共有しよす。 このアプリを使用する前に、 sony.com のプライバシー ポリシーと利用規約をご確認くださ
	UV.
	The state of the s

ようこそ	
۲	
パスワードを入力	Ø
統行するにあたり、Google はあな ドレス、言語設定、プロフィール 共有します。このアプリを使用す	たの名前、メールア 写真を sony.com と る前に、sony.com 劇をご確認くださ

2. パスワードの入力

#### 3. 利用規約への同意

Ne	arel Network Consoleクラウド語へようこそ!
Neu Neu	rel Schwart Consert クラウドの通信ではしてきたきまでしょうではいます。 na National Consert クラウド語にない使用 ミニューマル・センジー なの感音・学校、特徴の目前等が行きたが、ドラスマ
す。 初月	<b>法教授学习市に以下者ご課題くだら</b> い。
- 1	inna Arbank Carolin クラフド後期 Sanak Carolin となれ目を意味します。他のブラフジム社会主要がしないことが - Park
·,	いいかす。 コートによるトリームの設計は無利用していたがたり、データングが、利用するロークスペースの各国が構成、準定時に 100210時期にした利用の資本分支になります。
0	PUNCPUNE目標準備に応じて期間のの構造になりますので、必要用小型の上ストで利用が可能です。
1	サービスとは急烈の利用や30回来時に1時間、リークスパース(1088、プロジェクト・10間の目前しています。
1	11)を起える時にななりを利用する場合はタレジットル。 PROS部でしたがな人体である時になります。 1) Mark A - Markanan Samanan
10	AND VERY EXPERIENCE SETTING ALL CONSIDER OF A VERY SUBJECT OF A VERY S
16	2010年2月11日時間1990年19月1日時間2月1日時間1月1日時間1月1日時代2月1日日時代2月1日日1月1日日1日日1日日 1月1日日月1日日日日日日日日日日日日日日日日
	19月1日はビステレードで、コススコートントンの加速などに開催したとうようという。20月1日はないという。 19月1日は1月2日には「日本の「お供加」」、「「日本の日本の日本の日本の日本の日本の日本の日本の日本の日本の日本の日本の日本の日
Ne 22	ural Network Console ⊐⇔ ka∋a antata
2	212 P/08/00/08/
Here	ral Selamb Consule クラフド族(な)「キリービス」といいます)は、ソニーネットリークスペルニシーションの特徴の
e 1	16年「旅行」とした後は、小説のはあり「Pスされ」とり「Pスが作用される場合法、Naural Satwark Constants つうつと

2 CATTERNET FLOAt 41 (1960年20) ドスフォースの「ドスパキ目はALSERAEL Name (Larsen Carsen STOP) 第21月前日 (117) (Labor House) 2014年-1220日年 21月前日、ホードンの単株を手かったのが見ていないか の前のの目的での目的な時時時間(10) ドネノービスの発展するという、本の自己自己で「本の内容」といいます。そのます 2014年5月 (ビント)

第1条(注意) 本物語であった時をためと思め注意します。 (2)時日日にとた。本地時代に開からた。自分時に支からみ時で訪らしており、ビスが明白の目に少しから。 (2)は、日本日によった時間を見、つい、「時代のように」、時代の人が一日になったとうために、たいたいと言語のの意味。

UTRACTACOUSE





### データの有無と本パートの進め方について

本ドキュメントではデータをお持ちの方とそうでない方の両方を読者として想定しています。 データがない方はNNCのサンプルデータをご利用いただくか、オープンデータをご利用ください。 以下でそれぞれにあった進め方を確認してください。

#### <u>1. データをお持ちの方</u>

アノテーション(物体のラベルや位置の入力)やデータセット分割などの作業が必要ですので、次頁以降を読み進めてください。ご準備いただいたデータの画像サイズは事前に統一していただく必要はありません。

#### 2. データがなく、サンプルデータを利用される方

データをお持ちでない場合でも、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。 NNCのサンプルデータの詳細は<u>サンプルデータの説明</u>に記載がありますので、ここから読み進めることも可 能です。

#### 3. データがなく、オープンデータを利用される方

データがお持ちでない場合でも、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、データセット作成から一連のモデル作成を体験できます。データをダウン ロードのうえ、「1.データをすでにお持ちの方」と同様に、次頁以降を読み進めてください。ダウンロー ドしたデータの画像サイズは事前に統一していただく必要はありません。

## データセット作成のステップ

画像データに物体のラベルとバウンディングボックスの対応付けをしたデータセットを作成し、本ドキュメ ントで用いるモデル構築手法を利用するためのフォーマット変換を行います。

データセットは学習用と検証用に分割したのちに、それぞれNNCにアップロードします。

ステップの2~4はNeural Network Libraries(NNabla)のコマンドラインを用いることで、一度に実施することができます。

	ステップ	概要概要	利用ツール
1	アノテーション作業	画像データと物体のラベル・バウンディングボックス の対応付け(アノテーション)作業を行います	アノテーションツール※
2	フォーマット変換 (グリッドの導入)	グリッドという学習効率化と速度向上のための手法を 用いるため、フォーマット変換をします	
3	フォーマット変換 (アンカーボックスの導入)	アンカーボックスという学習効率化と精度向上のため の手法を用いるため、フォーマット変換をします	NNablaのコマンドライ ン(NNabla_CLI)
4	データセット分割	データセットを学習用と検証用の2つに分割します	
5	データセットアップロード	2つのデータセットをNNCにアップロードします	NNC専用のアップロー ダ
※アノテ-	-ション作業は特化したツールをご利用	またない 詳細はアノテーション作業に記載	

### アノテーション作業

画像データと物体のラベル・バウンディングボックスの対応付けには、アノテーションに特化したツールを ご利用ください。

代表的なアノテーションツールは下表になります。ここでは、次のステップで変換が必要なため<u>YOLO</u> フォーマットで出力可能なツールのみを記載しています。

データセットの保存はYOLOフォーマットをご利用ください<sup>※1</sup>。

アノテーションツールのイメージ

アノテーションツールではGUIを通して、バウンディ ングボックスとラベルを設定することができます



YOLOフォーマットで出力が可能な 代表的なアノテーションツールの例

ツール名	URL
VoTT V1 <sup>%2</sup>	https://github.com/Microsoft/VoTT
labelImg	https://github.com/tzutalin/labelImg

※1 YOLOフォーマット以外で出力されるツールをご利用の場合には、プログラミングなどでフォーマットを変換いただく必要があります。 ※2 VoTT V2ではYOLOフォーマットでの出力機能がありませんのでご注意ください。





YOLOフォーマットでは画像と同じディレクトリに、画像ファイル名と同じ名称のテキストデータを作成し、 そこに物体のラベルとバウンディングボックスの情報を保存します。



※1 ファイル名は連番である必要はありませんが、NNCでは日本語などの2バイト文字に対応していませんので、ご注意ください。 ※2 NNCではpng、jpg、jpeg、gif、bmp、tifの画像フォーマットに対応しています。

### フォーマット変換: グリッド



画像を格子状に分割したグリッドを用いて、物体のおおよその位置を予測することで学習効率とモデル精度 を向上できます。

グリッドを用いることでフォーマットが2次元の表データとなり、バウンディングボックスの中心が含まれるセルにグリッド内での相対位置(詳細は<u>Appendix</u>参照)とバウンディングボックスのサイズが記載されます。 また、NNabla\_CLIの変換では、相対位置はグリッドサイズで規格化をし、バウンディングボックスのサイズ はグリッドサイズで規格化をした後に対数に変換します。

画像サンプル	YOLOフォー	ーマット	•						7	ブリ	ッ	ドに	53	とる	変換	與後	ファ	ተ —	マ	ッ	-						
	0 0.44 0.30 1 0.61 0.66	0.68 0.76 0.70 0.50	・ グ 	リッラ	ドズ	ごと ル	:の	グ 	リッ I対	╯ド∣ 位置	内て 置(積	ぎの 黄)	ク	ブリ 相対	ッド す位i	内 置(糸	での 従)	バ 	ウン	デクク	ィン くの	·グ 幅	バボ	ウン ック	ノデ フ ス	イン の福	ッグ うさ
	「 <sub>イ</sub> 」 「 ラベル 中心位置の		-1	-1	-1	-1	-1	0	0	0	0	0	(	o c	0	0	0	0	0	0	0	0	0	0	0	0	0
	横、縦	バックティン グボックスの 幅、高さ	-1	-1	0	-1	-1	0	0	0.2	0	0	(	0 C	0.1	0	0	0	0	0.53	0	0	0	0	0.58	0	0
	※YOLOフォーマット	ではそれぞれの	-1	-1	-1	-1	-1	0	0	0	0	0	(	o c	0	0	0	0	0	0	0	0	0	0	0	0	0
	大ささは画像サイ	「人で規格化	-1	-1	-1	1	-1	0	0	0	0.05	0	(	o c	0	0.3	0	0	0	0	0.54	0	0	0	0	0.40	0
			-1	-1	-1	-1	-1	0	0	0	0	0	(	0 C	0	0	0	0	0	0	0	0	0	0	0	0	0

※NNabla\_CLIによる変換では、物体の中心が含むセルにのみ値が出力され、含まないセルにはラベルは-1、それ以外は0で埋められます。

### フォーマット変換: アンカーボックス



物体検出では大小さまざまなサイズの物体を認識する必要があります。

検出するバウンディングボックスの大きさやアスペクト比に応じて、認識するネットワーク部分を複数用意 することで、学習効率や検出速度が向上します。

アンカーボックスと呼ばれる代表的な形状のバウンディングボックスを複数用意し、認識する物体をそれぞ れのアンカーボックスに割り当て学習を行います。

アンカーボックスの形状やデータセットの割り振りは、NNabla\_CLIの変換では学習データのクラスタリングにより、自動的に行われます。

最適なアンカーボックス(例:アンカーボックスを5個準備する場合)





### データセット分割

モデル作成には、学習に利用するデータセット(学習データ、Training Data)と、モデル精度を検証する データセット(検証データ、Validation Data)の2つが必要になります。

作成したモデル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を準備する必要があるため、 あらかじめ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。 NNabla\_CLIの変換では引数にそれぞれの割合を指定することができます。

データセットの件数については、Deep Learning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: <u>データの重要性</u>)



### NNabla\_CLIの実行

NNabla\_CLIを利用することで、グリッドとアンカーボックスを用いるためのフォーマット変換、入力画像サイズの統一とデータセット分割を一度に実行することができます<sup>\*1</sup>。 コマンドライン実行後の出力の詳細は、次頁で実行例とともに記載をしております。

《NNabla\_CLIの実行方法》

nnabla\_cli create\_object\_detection\_dataset -i データのディレクトリ -o 出力ディレクトリ -n 判別ク ラス数 -c モノクロ(1)/カラー(3) -w 入力画像の幅 –g 入力画像の高さ –a アンカー数 -d グリッドサ イズ[16<sup>※2</sup>] –m 画像修正方法(resize/trimming/padding) –s ランダム分割(true/false) -f1 学習データ ファイル名 -r1 学習データ割合 -f2 出力検証データ名 -r2 検証データ割合

黄字が変更箇所になります

- ※1 NNabla\_CLIはNNablaに付随するコマンドラインです。NNabla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
- ※2本ドキュメントの通りモデルを作成される場合にはグリッドサイズは16に指定してください。 16から変更される場合には、ご自身でグリッド数を計算し、ネットワーク構造を修正する必要があります。

### NNabla\_CLIの実行例

言어 HH

コマンドラインの実行後、出力ディレクトリに学習用と検証用のデータセットを取りまとめたCSVファイル が出力されます。また、サイズ統一をした入力画像と出力データを記載したCSVファイルがdataディレクト リの中に出力されます。

《コマンドラインの実行例》

nnabla\_cli create\_object\_detection\_dataset -i ./yolo -o ./ -n 4 -c 1 -w 112 -g 112 -a 5 -d 16 -m resize -s true -f1 training.csv -r1 80 -f2 validation.csv -r2 20

《出力ディレクトリの例》



		DIL PJ
1	training.csv	学習用データセットの入出力の対応関係を 取りまとめたCSVファイル
2	validation.csv	検証用データセットの入出力の対応関係を 取りまとめたCSVファイル
3	data/[画像ファイル 名].png	元画像を入力画像サイズにサイズ統一した 画像
4	data/[画像ファイル 名]_label.csv	画像にある物体のラベル情報を示したcsv ファイル
5	data/[画像ファイル 名]_region.csv	画像にある物体のバウンディングボックス の情報を示したcsvファイル

《出力ファイルの説明》

## データセットのアップロード NNCはクラウドサービスのため、専用のアップローダを利用し、モデルを作成するために必要なデータセッ

NNCはクラウドリービスのため、専用のアックロータを利用し、モテルを作成するために必要なテー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。 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.sonv.com/docs-ia/

 $rac{r}{r} = \frac{r}{r} + \frac{$ 

### 2. アップロードキーの取得

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

Neural Network Console	🗆 Name
Deshboard	□ mnist.small_mnist_4or9_training
Project	mnist.small_mnist_4or9_test
Dataset	豆 0.01 GB 《1 2017-10-30T09:30:05Z





### 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

2	Neural Network Console Uploader -	×
File	Help	
token	eyJhbGciOlJSUzIINIIsInR5cCJ6IkpXVCIsImtpZCI6IjEIMDU0NTg4OTQifQ.eyJpc3MiC	Paste
file	C:/neural_network_console_120/samples/sample_dataset/MNIST/small_mnist_4or	Select
	start	
Upload	der (2018-02-15 version)	- 2

#### ※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、画像などはサムネイルの形で確認できます。

アノテーション

(ガリッド)

**ル**アンカーボックス



データセット一覧に追加されたデータセット

※labelやregionが時系列表示 されますが、表データと して取り扱えますので意 識する必要はありません。

データセッ

分割

データセッ

アップロード

### 【参考】NNC上のサンプルデータ(synthetic\_image.object\_detection)の説明

正方形の背景の上に楕円、三角形、四角形、五角形を色、個数、形状をランダムに配置し、それらを検出す るサンプルデータです。

入力データが画像データ(image)となり、出力データがアンカーボックスとグリッドごとのラベル情報(label) とバウンディングボックスの情報(region)になります。

バウンディングボックスの中心が含まれる場所にデータが記載されています。中心が含まれない箇所は categoryは-1、regionは0で穴埋めがされています。

サンプルデータの入出力

カテゴリマスタ







## 物体検出モデル作成のステップ

NNC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、物体検出モデルのサンプルプロジェクト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。

物体検出モデル作成のステップ





### ネットワークの修正



## サンプルプロジェクトのコピー

サンプルプロ ジェクト選択 データセット 変更 終正 学習・評価

サンプルプロジェクト(tutorial.object\_detection.synthetic\_image\_object\_detection)を検索し、コピーします。

1				
	Neural Network Console	Group         Personal           + New Project <u>↑</u> Upload Project	Date uploader	ALL < Q object
Proj	ectをクリック	Name	Modified	Action 2 検索容に"object"
	Project	tutorial.object_detection.synthetic_image_object_detection	<sup>on</sup> sample クトをクリック	之. 候衆恋に Object 入力
	Job History			
	Sample Project			
	Sample Project Public Project			
	Sample Project Public Project Service Settings	4.ポップアップで好き オユキレズovさな	選択した ます。フ の文字に (¥,/,:	Eサンプルプロジェクトを元に新しいプロジェクトを作成し プロジェクト名を入力してください。1~255文字以内で以下 ま使用できません。 , *, ?, ', <, >,  , ;)

※学習結果やモデルも含めコピーを行うため、処理に時間がかかる場合があります。

## プロジェクトの起動

サンプルプロ	データセット	ネットワーク	学習・評価
ジェクト選択	変更	修正	

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。 以下のようなネットワークが表示されていることを確認します。

	Group Personal	
Neural Network Console	+ New Project 🗠 Upload Project	Date uploader 🗸 – ALL – 🗸
	Name	Modified
Dashboard	object_detection	2019-10-10T02:31:05Z
Project	2	
Dataset	1.コピーした	プロジェクト(前頁)
Job History	4で指定した名	る称)をクリック
Sample Project		
Public Project		
Service Settings		

#### Editタブ: ネットワークを作成するページ

CO EDIT TTAKING EVALUATION	2 2 Martine Charles Contract	SE) CONING	
Components	setanti x torong x videnos x encine x contint x +	Controller	fan
Q territ	ଏ ୧୬ ଅଧିର ସ୍ଲେକ୍ ସ୍ଥ୍ୟରେ ସ୍ଥରେ ସ	O Profile	
<ul> <li>KO</li> <li>Input</li> <li>Second Street</li> <li>Handline</li> <li>Also McDron</li> <li>Exclosition readers well are</li> <li>Handrid and McDron</li> <li>Exclosition readers</li> <li>Exclosition readers</li> <li>Chargestall Cost Charlot Ready</li> <li>Chargestall Cost Charlot Ready</li> <li>Chargestall Cost Charlot Ready</li> <li>Charles and Cost Charlot Ready</li> <li>Ender</li> <li>Antine</li> <li>Charles and Cost Charlot Ready</li> <li>Ender</li> <li>Charles and Cost Charlot Ready</li> <li>Charles and Cos</li></ul>	Marca 1       Unit AT         Marca 1       Unit AT <t< th=""><th>Twin     Twin     Twin     Surray      Surray</th><th>Nation Reach           ABDI           S.1           Lation Controll           Lation Control</th></t<>	Twin     Twin     Twin     Surray      Surray	Nation Reach           ABDI           S.1           Lation Controll           Lation Control
		Network Statistic	1

データセットの変更

サンプルプロ ジェクト選択 データセット 変更 修正 学習・評価

DATASETタブからTrainingとValidationのデータセットをそれぞれ変更します。 (サンプルデータを利用する場合には変更は不要です)

			8 N. 1	
슈 EDIT TRAINING EVALUATION 았어	ject_detection 🛛 DATASET {රිදු CONFIG		NING EVALUATION	J test
Training	image object detection training	training		Dataset List Q Search
Cumbhatic image object detectio		Num Data 0	_	Name
2. Trainingをクリック Shuffle	3. Link Datasetの横のハイ	パーリンクをクリック	Main	mnist.small_mnist_40r9_training 1500 Rows III 2 Cols <1 2017-10-30709:30:06Z
Num Column 3 Shuffle true Index x:image	サンプルでセットされている記載されています。	6既存のデータセット名た	ř	mnist.small_mnist_4or9_test = 500 Rows = 2 Cols () 2017-10-30109:30:052
Cache true Normalize false Main 1 1,112,112	Training: syntetic_image.object	_detection.training		mnist.mnist_training
	Validation: syntetic_image.obje	ct_detection.validation		
Validation	05	07-		mnist.mnist_test 副 10000 Rows (III 2 Cols ミン 2017-10-30709-30:032
Num Shur Shur Cache Me	35,7 35,7 35,7 35,28 35,28	● 4.一覧から学習 データセットのアッ を選択してください	に用い. , プロー `。	るデータセットを選択 ド時に指定したファイル名

1. Datasetタブをクリック

※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。



## ネットワークの構成

サンプルプロ ジェクト選択 データセット 変更 ネットワーク 修正 学習・評価

本サンプルでは学習用、検証用、推論用のネットワークが入出力や評価指数などを個別に設定するため、 Trainingタブ、Validationタブ、Runtimeタブと別々に記載しています<sup>※</sup>。 各ネットワークでは共通のNetworkタブが引用されており、NetworkタブはConvUnitタブを引用されています。

ネットワーク画面の説明





※それぞれのネットワークはCONFIGタブの設定画面で学習用、検証用、推論用に対応づけられています

### ネットワークの修正

サンプルプロ ジェクト選択 データセット 変更 **キットワーク** 修正 学習・評価

本サンプルではデータセットが変更された際に調整するパラメータを引数レイヤー(Argument)として抜粋しています。データセット作成時の入力画像・分類数・アンカー数をもとにパラメータを修正してください<sup>※1</sup>。 (サンプルデータを利用する場合には変更は不要です)

#### ネットワーク修正の操作説明

#### 修正が必要なパラメータの説明

	දිදි object_detection ල DATASET
. Trainingタブをクリック × 📷	ning X Validation X Runtime X ConvUnit X +
✓ IO Input ✓ Loss	5. Validation、Runtimeタブ も2~4の手順で修正 <sup>※2</sup>
SquaredError A Value : S HuberLoss AbsoluteError AbsoluteError EpsiloninsensitiveLoss InputWidth	2. NumClassをクリック
BinaryCrossEntropy A Value : 112 SigmoidCrossEntropy CategoricalCrossEntropy SoftmaxCrossEntropy	R <sup>Reshape_4</sup> 5,9,7,7
Layer Property A Argument 四 2、3	ラメータも の手順で修正 55.77 5.177 5.177
Name NumClass Value 4	G GreaterFqualScalar Value : 0
<sup>Type</sup> Pint G. Valueの値	を修正 SigmoidCrossEntropy 5.7.7

パラメータ	説明
NumClass	分類数
NumAnchor	アンカー数
InputCh	入力画像のモノクロ/カラーを 設定 1:モノクロ、3:カラー
InputWidth	入力画像の幅
InputHeight	入力画像の高さ
ScoreThrehold	推論時に物体を検出する閾値 0~1の値で、値を大きくする につれ判断基準が厳しくなる (Runtimeタブのみ設定)

※1その他のネットワーク部分の解説については、<u>ネットワーク解説 –物体検出編-</u>をご参照ください。 ※2引用元のConvUnit、Networkタブについては、Training、Validation、Runtimeタブでパラメータを設定することで、自動で反映されます。

## 学習の実行

新規プロジェ クト作成 データセット ネットワーク 学習・評価

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<sup>※</sup>。(参考: <u>学習環境と処理時間</u>) 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: <u>https://dl.sony.com/ja/business/</u>)

学習実行の方法





※サンプルプロジェクトのネットワークは複雑なため、CPU実行の場合にはGPU実行と比較して学習時間が長時間になります。 計算途中にウェブブラウザを閉じても計算が止まることはありませんので、長時間に及ぶ場合にはあらためて結果をご確認ください。

#### TRAININGページの概要

学習曲線の読み取り方

新規プロジェ クト作成 データセット ネットワーク 学習・評価

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、学習データ量を増やすなどの改善が必要です。



## 評価の実行

サンプルプロ ジェクト選択 データセット ネットワーク 学習・評価

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。 分類問題の場合には、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

Controller Run	◆ 評価グラフ	内容		問題	
Profile       Train     Structure Search	Output Result	各データの1つ1つの判定結果		すべて	
Evaluate	Confusion Matrix	データセット全体の統計的な指標と (分類ラベルごとに結果を集計した表	: 混同行列 表)	分類	
O CPU x 1	Classification Result	各データの判定確率上位3カテゴリ	リの確率	分類	物体検出は対象外
NVIDIA® Tesla® V100 GPU x 1	Classification Matrix	カテゴリごとのモデルの判定傾向		分類	ない
	Likelihood Graph	判定確率と正答率の傾向		分類	

### **Output Resultの見方**

サンプルプロ ジェクト選択 データセット 指定 修正 学習・評価

Output Resultでは検証データの1つ1つの判定結果を確認できます。 予想結果のScoreのピークが物体中心にあるかどうかで、モデル精度がよいかを確認できます。 その他の出力の説明は次頁をご覧ください。



Output Resultの概要

※label、region、category'、r'が時系列表示されますが、実際のデータは表データ(CSVファイル)として保存されています。

## サンプルプロジェクトの出力

サンプルプロジェクトの出力はグリッド内に物体中心を含む確率(score)とグリッド内に含まれる物体中心の ラベルのスコア(category)とバウンディングボックスの情報(region)の3種類になります。



アンカー数

※1 実際の出力データはscoreがアンカー数分の確率値を示したモノクロ画像、categoryとregionが2次元のCSVファイルになります 詳細はサンプルプロジェクトの出力ファイルを参照

※2 softmax変換により確率値に変換できます

※3 グリッドサイズにより規格化されており、バウンディングボックスのサイズはさらに対数変換した値です





## 物体検出モデル利用のステップ

NNC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、NNablaが必要になります。 NNablaを用いることでコマンドラインやPythonなど様々な方法で作成した物体検出モデルが実行可能となり ます。

物体検出モデル利用のステップ



## 物体検出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<u>モデルの実行方法</u>)。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モデルを利用することができます。

슈 EDIT TRAINING	EVALUAT	NON	
Job History Pause All Running Jo	 bs	Elapsed Remaining Tot     00:00:02:59     Control Curve O Trade-of	tal :00:02:59 CPU x 1 ff Graph All C I O Linear Scale O Lo
20190409_051421_1 Training 0.000161		Cost V Training Error	Validation Error
Validation 0.039406 Best Validation 0.039406 CostMultiplyAd 436.490	Renam Open L	e earning Curve for Comparison	
d Comparison	Clear L Susper	earning Curve for Comparison	
0 20190409_051421	Downlo	bad ►	NNP(Neural Network Libraries file format)
Training 0.000540 Validation 0.035466 Best Validation 0.035466 CostMultiplyAd 436,490	Re-edit Publish Delete	1	NNB(NNabla C Runtime file format) ONNX html beta
d Comparison Pareto Optimal		0.25	

## NNablaの設定



任意のPCにNNablaをインストールします。

NNabla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

NNabla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、コマンドラインと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	<u>https://support.dl.sony.com/docs-ja/</u> チュートリアル: neural-network- consoleによる学習済みニューラ/	次頁に
2	Python	可能	比較的容易に利用可能	NNPファイル	<u>https://support.dl.sony.com/docs-ja/</u> チュートリアル : neural-network- consoleによる学習済みニューラ/	「解説あり 」
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/tre e/master/examples/cpp/mnist_runti me	
4	С	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/la test/python/file format converter/f ile 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	

### コマンドラインでの推論実行

モデルの ダウンロード NNablaの設定 モデルの実行

NNabla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。

出力はグリッドごとのscore、category、regionになりますので(参照:<u>サンプルプロジェクトの出力、サンプル</u> <u>プロジェクトの出力ファイル</u>)、バウンディングボックスの描画には、scoreの高いグリッドのcategoryと regionのデータを取り出し、計算していただく必要があります。

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]

※ NNC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。 以下のサンプルでは入力画像を読み込み、モデルを実行し、バウンディングボックスを描画するまでの一連 の流れを記載しています。

```
#必要なmodulesのインストール
                                                                                                                      1/4
import nnabla as nn
from nnabla.utils import nnp graph
from nnabla.utils.image utils import imread
import numpy as np
import math
import matplotlib.pyplot as plt
import matplotlib.patches as patches
#ファイル名などの設定
nnp file = "./result object detection.nnp"
input figure = "./input.png"
output figure = "./output.png"
colorlist = ["r", "g", "b", "c"]
classlist = ["Ellipse", "Triangle", "Rectangle", "Pentagon"]
                                                            ここの部分を問題に応じて
threshhold = 0.6
                                                            変更する必要があります
num anchor = 5
num class = 4
img height = 112
img width = 112
figscale = 3
```

モデルの ダウンロード NNablaの設定 モデルの実行

## Pythonでの実行方法



※入力画像がモノクロの場合には、ソースコードの一部をコメントアウト部分と変更してください。

### Pythonでの実行方法

3/4

#スコアが閾値を超えたグリッドのインデックスを取得 indices anchor, indices grid y, indices grid x = np.where(y score>threshhold)# Softmax 関数を定義 def softmax(x): u = np.sum(np.exp(x))return(np.exp(x)/u)#描画のための推論結果の取り出し bboxes, texts = [], []for idx anchor, idx grid x, idx grid y in zip(indices anchor, indices grid x, indices grid y): # category 情報の取り出し prob = y category[idx anchor\*grid height num+idx grid y][num class\*idx grid x:num class\*idx grid x+num class] prob = softmax(prob)label = prob.argmax()texts.append("{0}: {1:.1f}%".format(classlist[label], 100\*prob[label])) # region情報の取り出し # dx, dyはGridの左上からのグリッドサイズで規格化したもの # ratio width, ratio heightはグリッドサイズで規格化し、対数変換したもの ratio dx, ratio dy, ratio width, ratio height = y r[idx anchor\*grid height num+idx grid y][4\*idx grid x:4\*idx grid x+4] cx = (idx grid x+ratio dx)\*grid width $cy = (idx grid y+ratio dy)^*grid height$ width = math.exp(ratio width)\*grid width height = math.exp(ratio height)\*grid height bboxes.append(patches.Rectangle(xy=(cx-0.5\*width, cy-0.5\*height), width=width, height=height, ec=colorlist[label], fill=False))

#### モデルの ダウンロード NNablaの設定 モデルの実行

4/4

## Pythonでの実行方法

#### #バウンディングボックスの描画

fig = plt.figure(figsize=(figscale\*img\_width/img\_height, figscale)) ax = plt.axes() ax = fig.add\_axes([0, 0, 1, 1]) ax.imshow(img) # モデルがモノクロの場合には下記 # ax.imshow(img, cmap='gray', vmin=0, vmax=255) for (r, text) in zip(bboxes, texts): ax.add\_patch(r) ax.text(r.xy[0], r.xy[1], text, ha='left', va='bottom', transform=ax.transData, color=r.get\_ec() ) ax.tick\_params(labelbottom=False,labelleft=False, labelright=False, labeltop=False) ax.tick\_params(bottom=False, left=False, right=False, top=False) plt.savefig(output\_figure)

保存画像の例



※入力画像がモノクロの場合には、ソースコードの一部をコメントアウト部分と変更してください。



## 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。

✓「はじめる」を押下

### 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

	SONY	×
	サインイン	
つの72 もっと詳	bウントで、Sonyグループの複数サービスヘアクセス ・バ	
9-12-1	NID	
EX-J	レアドレス	
1 <del>4</del> 75	インルトを使する	
-פגא	F 🚯	
パスワ	- 15	0
	900	
<del>9</del> 4240	ンでお困りですか?	
	新しいアカウントの作成	



<u>2. 作成ページへの移動2</u>

### 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

アカウントの作成 ・・・ サインインID youraddress@example.com パスワード・ パスワードの消気力	SON	1Y
・・      サインインID      youraddress@example.com      バスワード・      パスワードの第第      パスワードの第五カ	アカウント	~の作成
サインインID youraddress@example.com パスワード・ パスワードの減支 パスワードの減支力	••	•
youraddress@example.com パスワード ● パスワードの減済	サインインロ	
パスワード • • · · · · · · · · · · · · · · · · ·	youraddress@example.com	
	パスワード 🜖	
パルワードの残ま ――-		0
	バスワードの両入力	パス・リードの雑章 ―――
ーンのアサウントで、SonyグループのWe&サードスへアクセス もっとぼしく	ಬಂಗ್ ಸಾರ್ಯ ಎ. ಎಂಗ್ರ್ಯೆಸ್ಸ್- ಸಂ ಕಾರ್ಟಿಸ್	加藤サードスヘアシヤス

## 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

	so	NY	×			
アカウントの作成						
•••						
国/2385						
日本			•			
=a						
日本語						
生作月日 🔞	-	_				
≠ 1945	* 5	• 7	•			
20			<u> </u>			

#### 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

## グリッド内での相対位置

バウンディングボックスの中心が含まれるグリッドがわかれば、おおよその中心位置は決まります。 グリッド内での相対位置(δx, δy)をさらに予測することで、詳細にバウンディングボックスの中心位置を決定 することができます。

δx, δyは学習効率化のため、グリッドサイズで規格化されており、0~1の値をとります。



グリッド内での相対位置とバウンディングボックスの中心位置

実際のバウンディングボックスの中心位置(X,Y) =(x+[グリッドの幅] x δx, y+[グリッドの高さ] x δy)

## サンプルプロジェクトの出力ファイル

scoreは2次元データがアンカー数分あるため、確率値を表すモノクロ画像がアンカー数分出力します。 categoryとregionは3次元データがアンカー数分あるため、x方向に分類クラスの確率やバウンディングボック スの情報を、y方向にアンカー数分の情報をそれぞれ並べて、2次元のCSVファイルを出力します。

出力データの変換(グリッド:5x5、クラス数:3、アンカー数:3の場合)

