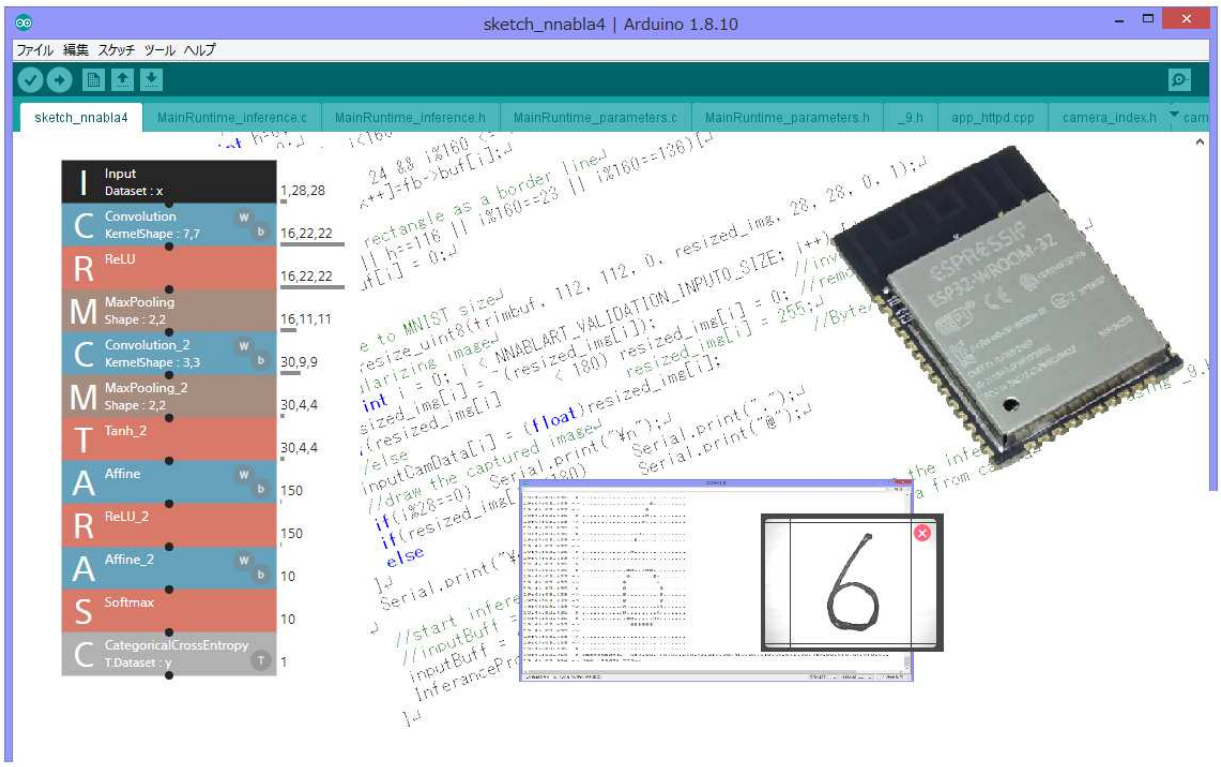
SAMPLE

ARDUINO 環境で始める
ディープラーニング講座

～ ESP32 と SONY Neural Network Console を使って
手書き文字認識までの手順～



むらよ iT-Style

はじめに

組み込みの世界でもマイコンがますます進化すると同時にコストも下がり、ディープラーニングや機械学習などの AI 機能を取り込む案件も出てきました。
ところが一般情報は高性能 PC を使った学習を前提とし、言語は Python が常識となっています。もちろん大きなニューラルネットワークを構築するには膨大な CPU パワーが必要であり小さなマイコンだけでこれを行うのは無理です。もちろん組み込みマイコンシステムでも大きな ROM・RAM を搭載し Linux や Android、Windows などの OS を走らせ Python もインストールして動作させるものもありますが、これらの大きなシステムは PC と同じなので対象としません。

本講座を受講される方の多くはマイコンを触ったことがあり、LED を点滅させるようなことはマニュアルを読めばなんとかなるという方です。ただディープラーニングはまったく経験が無く、どこから手を付けていけば良いか悩んでいるという方にはガイドラインを提供できます。ゴールは、あくまで小型のワンチップで納まる範囲の応用設計で AI 機能を搭載した応用を設計する筋道がわかることとします。ニューラルネットワークの学習そのものは PC で行い、推論エンジン部分をマイコンに搭載することで CPU パワーの小さいマイコンでもいろんなことが応用できるようになります。
つまり程度の差はあれどもマイコンに AI 機能を搭載する一番のショートカットパスをお伝えすることになります。

そこで、マイコンとしては IoT 分野で急速に普及している Espressif 社の ESP32 をターゲットとします。ESP-IDF を使わず Arduino マイコンとして、Arduino の統合開発環境で AI 機能を実装していく方法を採用します。ESP-IDF を使うと詳細なライブラリが全て使えますが、ソフトウエアの実装に大変な労力が必要となります。一方 Arduino 環境ではライブラリがしっかりと分類され、通常の開発では問題ないレベルにまとめられ、マイコンの初心者であっても使いやすい開発環境となっています。また一般書籍も数多く出版されており、Arduino での開発が実質的なスタンダードとなっています。マイコン技術者の方々が余計なトラブルに巻き込まれず AI 機能を搭載するときに必要な技術を学び、実際に実験して感じることができる最短ルートになるはずです。

本講座は、IT-STYLE が提供しているソフトウエア教室の 3 つの講座を一本にまとめてテキストで提供しているものです。
教室ではおよそ 6 か月のコースになります。対面での教室ではないので、できるだけ実践でそのまま動作するように構成していますので、最短で 20 時間程度で終了することができます。

■本マニュアルのライセンス条項

・本マニュアルは購入された方にのみ許諾されたものです。

・本来は教室での講座になります。
　複数名で知識共有される場合は複数名での講座ご購入をお願いいたします。

・著作権は、むらよ IT スタイルに属します。

・各社製品・商品名など登録商標は誤解のない範囲で著作権表示を省略しています。

■本講座の対象者

・Arduino 開発環境でマイコン ESP32 を使い AI 機能を実装してみたい方

・Arduino 開発環境を使って UNO などの LED 点滅動作をさせることができる、
　または他のマイコンで同等レベルの方

・C 言語がある程度わかる方。

■必要な環境

・Windows8.1 以上

・Wifi 環境 2.4GHz　　SSID パスワード必要

・Arduino 開発環境

・Anaconda

・ESP32 マイコンボード（M5Camera が最適。後述）

■用語

NNC・・SONY Neural Network Console

NNabla・・Sony 製、ディープラーニングフレームワーク　“Neural Network Libraries
　　コードでは nnabla 小文字表記

ArduinoIDE・・Arduino 統合開発開発

ESP-IDF・・Espressif 社提供 ESP32C++ライブラリ、ソースコード

スケッチ・・ArduinoIDE で作成したプロジェクト、プログラム

ESP32・・マイコンもしくは搭載モジュール代表名として表記

■作業フォルダ

筆者の環境では、C:¥users¥Yoichiro 以下のフォルダを作用フォルダとして使っています。ユーザネーム部分は適時読み替えてください。　Yoichiro は筆者のユーザ名です。

目次：

～事前知識～

AI について概論

AI については数多くの書籍が出版され、また多くの有料セミナーも開かれています。深くまで知るには相当な努力が必要かとは思いますが、本講座を一通り終了されると、何が必要で何が必要でないか。また何をマイコンソフトウエア・ハードウエア技術者が担当し、どの部分をデータ解析者が行うべきかが見えてくると思います。

PC 上だけで AI の実験をしていると、ついアカデミックな領域に入ってしまい悶々と悩む日々が続きがちになります。現在のマイコンと言われるものを搭載した機器の中で、比較的システムとして複雑でない応用製品に向けては、ハイエンドコンピュータで実装しているような AI 機能は要求されていません。ですからここ数年はひとまず AI と呼ばれることに対してはニューラルネットワークを構築して推論しているエンジンを搭載していれば問題なく AI 機能搭載と謳うことも可能ですし、今後の複雑なディープラーニングに向けての最初のステップにもなると考えます。現状 if 文があれば AI 機能と言ってしまおうかと迷っている話をいくつかお聞きした中で、まず Arduino レベルでもできるディープラーニングの講座を開きました。

筆者は AI の専門家ではありません。マイコンソフトウエアに 30 年以上携わっていますので、マイコンの組み込みソフトウエアからの目で見た解説に偏っていると思いますし、AI に関しては、多少の間違いはソフトウエアの実装上問題ない範囲で、省略やごまかしが含まれます。より本格的なディープラーニングに挑戦される際に技術の修正をされれば問題ありません。

さて、AI は一般的には、機械学習やディープラーニングと呼ばれますが、一番簡単な理解としては、ディープライニングが一番ややこしく最先端の研究がおこなわれている分野であり、その基本が機械学習と呼ばれているものです。機械学習は、数十年前から存在していて多くの研究者が実験を重ねていたと認識しています。それがコンピュータパワーの増大で一気に大量のベクトル計算ができるようになり、今日のディープラーニングブームが起きたものと考えます。
代表的なもので、Google 社の TensorFlow が恐らく一番有名で、Amazon,Microsoft,IBM など各社が提供しています。大抵は Python のライブラリとして API が揃っています。ただ API は統一されておらず、名前と機能がそれぞれ異なりますので、本格的に始めるには TensorFlow やラッパーの Keras などが情報も多く、書籍も一番多いと思います。 他の API はビジネス要素が強いため、会社業務でたずさわるなら、

ESP32 概論

次に、今回マイコンとして使う ESP32 について少し書きます。ESP32 の事をご存じであれば、読み飛ばしてください。

IoT という言葉がはやって数年経ちます。すべてのモノがインターネットに接続されることで、何か世界が変わるという流れではありますが、20 年前にはユビキタスであったり、ホームオートメーションであったりで、常に同じアイデアは存在していました。
ただ当時はデバイス、特に無線チップとマイコンの値段が高く、それを実装しても購入者のメリットをなかなかアピールできないままブームが去っていく感じでした。ところが昨今この手のマイコンの値段がどんどん下がって来て、大手半導体メーカーのチップも含めて、家庭用品にも実装可能なレベルにまで下がってきています。そんな状況で、ESP32 は 1 個単位かつ非常に安く入手できるマイコンモジュールです。
マイコンそのものは ESP-WROOM-32 という型番になりますが、USB モジュールや電源レギュレータなどを搭載して実験に使いやすくまとめたモジュールが各社から販売されています。

ESPr Developer 32

ESP32-DevKitC

紙面上で多くの解説を載せることは難しいので、ぜひインターネット検索でリサーチしてください。多くの情報があります。

～ひとまず体験～

NNC にサインイン

ひとまず体験と称して、ディープラーニングを体験していただき、その後の流れをつかんでいただきます。ここでディープラーニングと言うのは少々定義が違うというご批判も聞こえますが、本講座では機械学習や深層学習などの人工知能に関する表現をディープラーニングと呼んでしまうこととします。

さて、AI 概論でお伝えしたように、本講座ではディープラーニングのプラットフォームとして、SONY Neural Network Console を用います。有名な Google TensorFlow など

NNC で初めての画像認識

では次に実際にハンズオン講座として体験していくことで、ディープラーニングの学習と推論の流れを理解していきます。
よくあるサンプルでは Iris データセット、あやめの花のサイズからあやめの種類を推論するものや、ボストンの住宅価格予想が最初に出てきますが、マイコンとの親和性や本講座のゴールをふまえて最初から画像認識でスタートしたいと思います。お時間があれば本講座終了後に Iris データセットでのサンプルを使ってマイコン実装してみるなどの実験をされることをお勧めします。応用設計の幅が広がると思います。

さて、画像認識のなかで、数字の 4 と 9 を判別するニューラルネットワークを

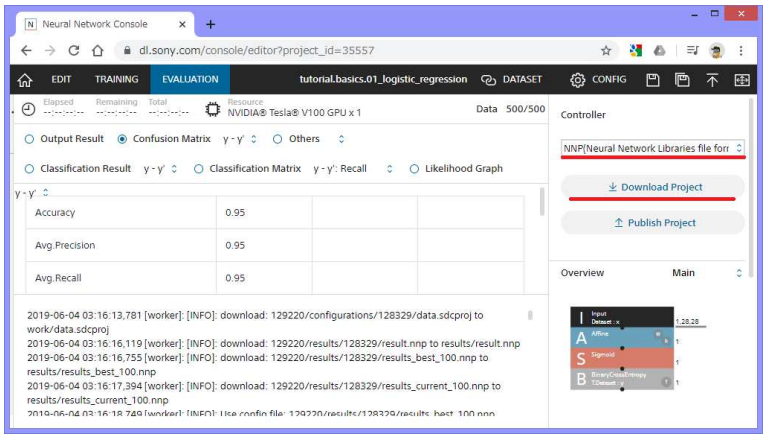
では PC で行った推論を Arduino マイコンに実装していくための手順に入ります。

～実践～

NNC で学習パラメータを取得

本講座の最初に体験した数字の 4 と 9 を判定する 1 層の Logistic Regression のニューラルネットワークの学習結果を使ってマイコンへ搭載して

の学習結果のパラメータとニューラルネットワークの構成を取得します。



このファイルを展開して、学習結果の重みなどのパラメータとニューラルネットワーク構成を含んでいる C 言語のソースコードを得ます。
複数のファイルが生成されるので、展開するフォルダを作成します。ここでは

学習結果のパラメータです。単純なニューラルネットワーク構成でも多量のパラメータがあります。

```
// Affine/affine/W
float MainRuntime_parameter1[] = {
      0.0890413299202919,
    0.020198114216327667,
     0.04940223693847656,
     0.11311008781194687,
```

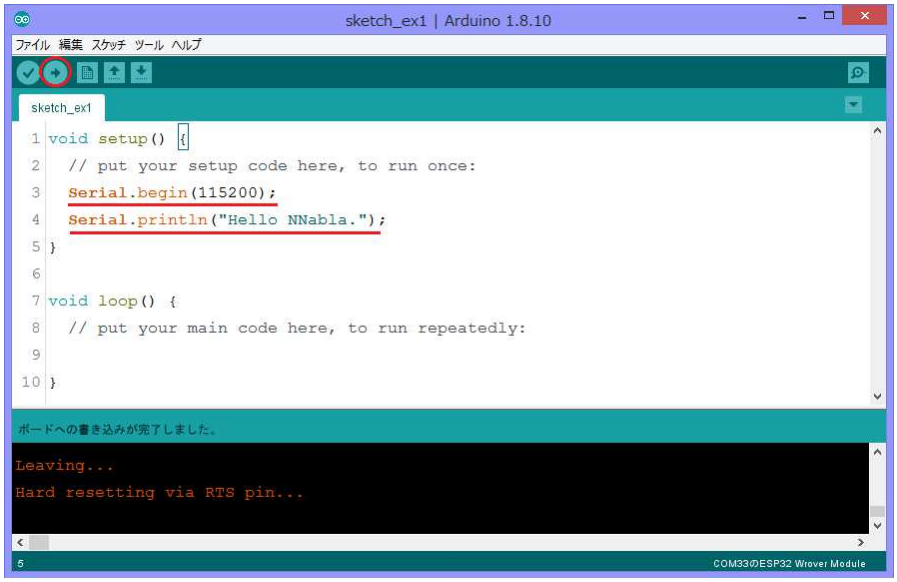
NNabla C ランタイムライブラリ準備

最初の目標の通り ESP-IDF の難易度の高い開発環境ではなく、ArduinoIDE を使って ESP32 を Arduino マイコンとして開発に使えるようにしていきます。そのためには、NNabla の C 言語 RuntimeLibrary が必要です。

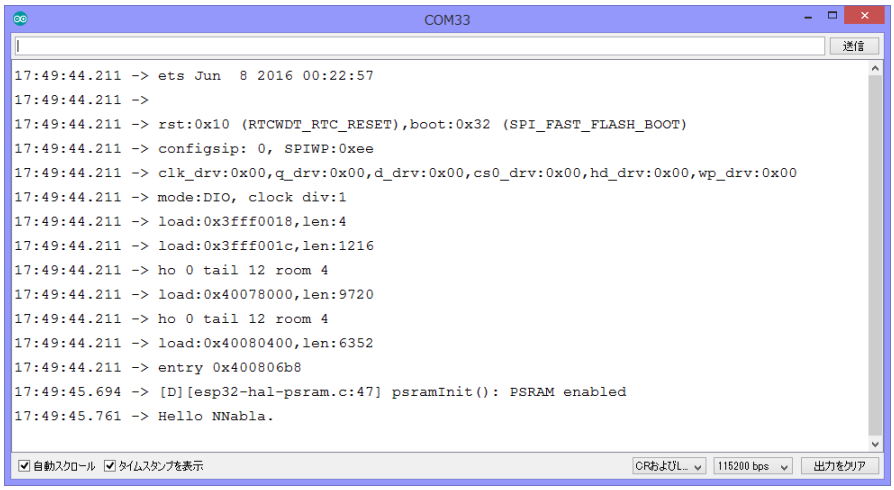
一般的に ArduinoIDE でライブラリと呼ばれるものには複数あり、ユーザが自分で使う共通コードとして使うライブラリ

最初のスケッチ作成

Arduino スケッチ（プログラム）を作成する準備ができたので、早速プログラムを作成します。Hello world. 相当のプログラムがマイコンに於いては L チカだと思いますが、残念ながら M5Camera には LED が搭載されていないので、代わりに



書き込み後、自動的にリセットされシリアルモニタの最後の行に“Hello NNabla.”と表示されれば OK です。



NNC コードとのマージ

先ほどの sketch_nnabla1.ino を拡張します。
「NNC で学習パラメータを取得」の章で展開した C ソースファイルを sketch_nnabla1 フォルダにコピーします。
講座の添付ファイルから、

```
// Copyright (c) 2017 Sony Corporation. All Rights Reserved.
//
// Licensed under the Apache License, Version 2.0 (the "License");
// you may not use this file except in compliance with the License.
// You may obtain a copy of the License at
//
//     http://www.apache.org/licenses/LICENSE-2.0
//
// Unless required by applicable law or agreed to in writing, software
// distributed under the License is distributed on an "AS IS" BASIS,
// WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
// See the License for the specific language governing permissions and
// limitations under the License.

// *WARNING*
// THIS FILE IS AUTO-GENERATED BY CODE GENERATOR.
// PLEASE DO NOT EDIT THIS FILE BY HAND!

#include "MainRuntime_inference.h"

#include "network.h"
#include "functions.h"

#include <string.h>
```

ここまででビルドが成功することを確認してください。

MNIST データ準備

Arduino 環境で NNC からの出力ファイルを C 言語変換してユーザアプリ（今は単に HelloNNabla の表示のみ）とマージしてビルドできるところまで確認しました。
実際に学習したニューラルネットワークとパラメータを使って、ESP32 マイコンで推論結果を得るプログラムを作成していきます。

確実にステップを踏むことと、全体の理解を深めるために、最初は入力データを固定データにします。具体的には、MNIST と呼ばれる手書き数字のデータセットから画像データを C 言語の配列データに変換します。

本講座では、MNIST データセットから 4 と 9 の 28*28 ピクセルグレイスケール値を C 言語のソースコードに変換したファイルを事前に用意しています。
用意しているサンプルファイルは数種類しかありませんが、ご自身で作成されたり、今後の画像入力アプリを作成される参考として、巻末に変換方法を記載しています。

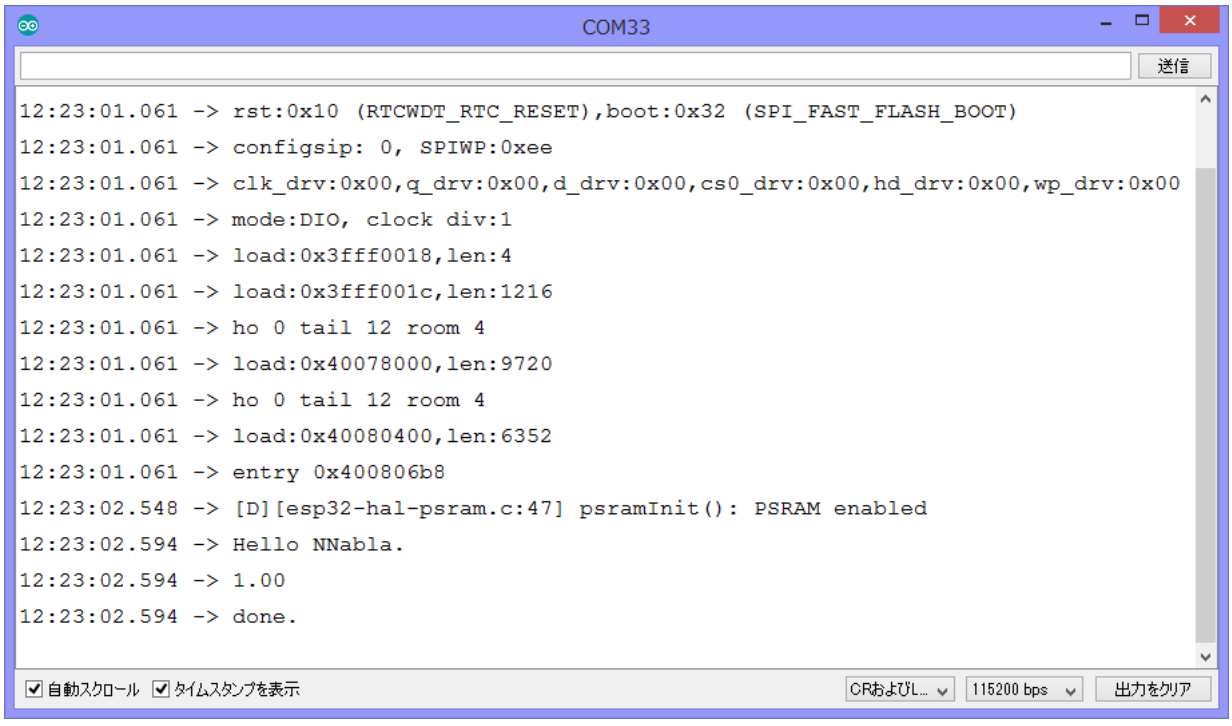

_4.h　　_9.h

0 から 9 までのデータを C 言語ソースコードに変換したファイルは

スケッチで推論実行

sketch_nnabla1 のフォルダをそのままコピー、リネームし sketch_nnabla2 を作成します。中にある sketch_nnabla1.ino も sketch_nnabla2.ino にリネームしてください。
そして、sketch_nnabla_misc フォルダにある、_4.h、 _9.h のファイルもコピー

そして実行結果は以下の通りです。



下から 2 行目が推論結果です。今回は_9.h で数字の 9 のデータを入力しましたので、出力は 9 である確かさ 1.00（100%）と推論されました。
スケッチで_9.h を_4.h に変更してみてください。結果は 0.0 になるはずです。9 である確かさが 0%となれば、学習結果が正しく動作していることになります。

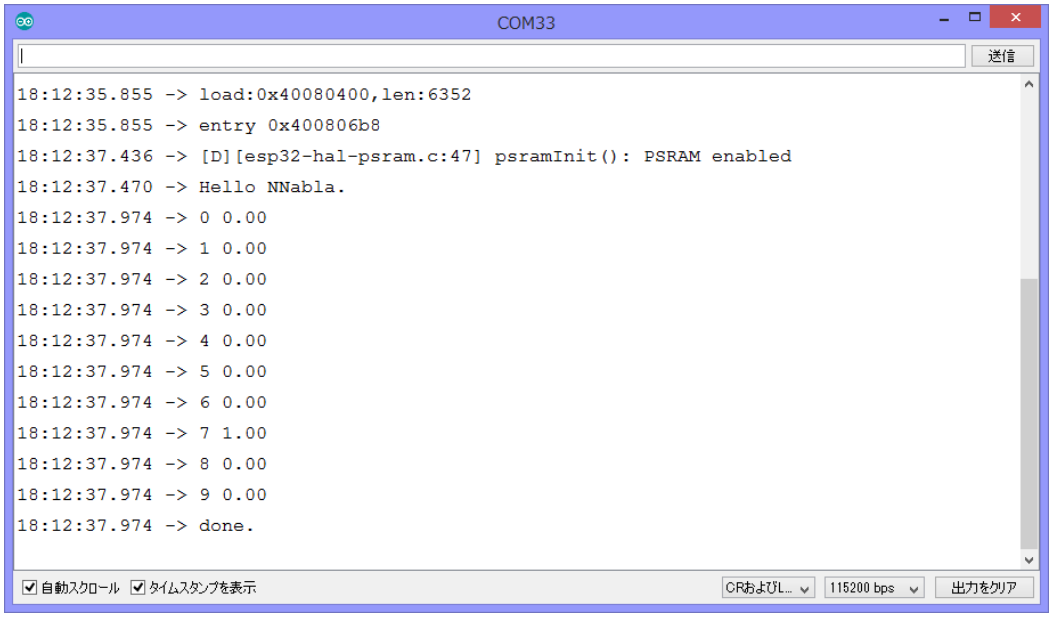
～応用編～

さてこれまでの講座で ESP32 マイコンを ArduinoIDE 上で扱い、NNC で学習したニューラルネットワークを C 言語ファイルに出力し、NNabla C 言語 Runtime library とリンクすることで、マイコン上で未知の入力データから結果を推論することができるようになりました。
ここでは、それをさらに拡張して、応用編としてビデオカメラで撮影している映像の手書き数字を入力として、連続して推論をするアプリケーションを作成します。

NNC で MNIST の学習

ではまず、NNC と MNIST データセットを使って、0 から 9 までの手書き数字を判定するニューラルネットワークを作成します。
このニューラルネットワークにはいろんな種類があるので、どのように構成しても構いませんが、ここは

ニューラルネットワーク 4 層で入出力を入れると 6 層になります。ここまで
_0.h、 _1.h など _9.h までの MNIST 手書き数字データを入力として使いますのでコピーします。



カメラ使用サンプルスケッチビルド

次にカメラからの入力を扱います。
ArduinoIDE から、[ファイル]-[スケッチ例] から [ESP32]-[Camera]-[CameraWebServer] を選択してください。
本講座では、このサンプルからの実装を sketch_nnabla4 として

```
// Select camera model
//#define CAMERA_MODEL_WROVER_KIT
//#define CAMERA_MODEL_ESP_EYE
//#define CAMERA_MODEL_M5STACK_PSRAM
#define CAMERA_MODEL_M5STACK_WIDE            //######################nnabla
//#define CAMERA_MODEL_AI_THINKER

#include "camera_pins.h"
const char* ssid = "atern[redacted]d-g";          //######################nnabla
const char* password = "[redacted]224ea";         //######################nnabla
```

ビルドが成功すれば、シリアルモニタにこのカメラにアクセスする IP アドレスが表示されていますので、ブラウザから URL を指定して接続してください。

```
19:30:52.916 -> WiFi connected
19:30:52.916 -> Starting web server on port: '80'
19:30:52.916 -> Starting stream server on port: '81'
19:30:52.916 -> Camera Ready! Use 'http://192.168.0.8' to connect
```

MNIST 推論部とサンプルスケッチ結合

MNIST の学習の章で作成した C ソースファイルとカメラのサンプルスケッチ、そして数点のファイルを合わせて 1 つのアプリとします。次章の前処理の実装と合わせて参照してください。
sketch_nnabla4　フォルダにスケッチは入って

以上の修正でビルドが成功するはずです。
カメラで手書き文字を撮影して、推論が正しくできるか確認してください。

シリアルモニタにはカメラから読み取った画像の 2 値画像と各数字 0 から 9 の推論結果を表示しています。ニューラルネットワークの最後に Softmax を使っていますので、

前処理の実装

カメラからの画像データは frame buffer に収められています。
camera_fb_t * fb;

中味は FORMAT を PIXFORMAT_GRAYSCALE 指定していますので、8 ビット明度のバイナリ配列です。さてこの生データを MNIST 画像ファイルで学習したニューラルネットワークに合わせるには

今回 MNIST データセットを使ったがために発生した前処理ですが、そもそも MNIST データセットではなく、トレーニングデータをすべてオリジナルで集めたとすれば、そのデータの加工でも問題が発生します。統一された規格を守ったトレーニングデータセットを作るだけでも相当な作業となります。

おわりに

これで講座は終了です。
ARDUINO 環境で始めるディープラーニング講座としてお話しをすすめてきましたがいかがだったでしょうか。
全体像をつかむという目的に於いてはほぼご理解いただけたと思います。

一通りの流れを作業してみると見えてくることがあります。 AI 技術を取り込ん

～おまけ～

CRuntimeLibrary の構築

NNabla には C 言語ライブラリが用意されています。GitHub 上にソースコードが

MNIST データセット

MNIST データセットは多くの AI 学習ステージでサンプルとして使われています。

http://yann.lecun.com/exdb/mnist/
このサイトからデータはダウンロードできますが、バイナリデータベースなので

本編　合計 53 ページ