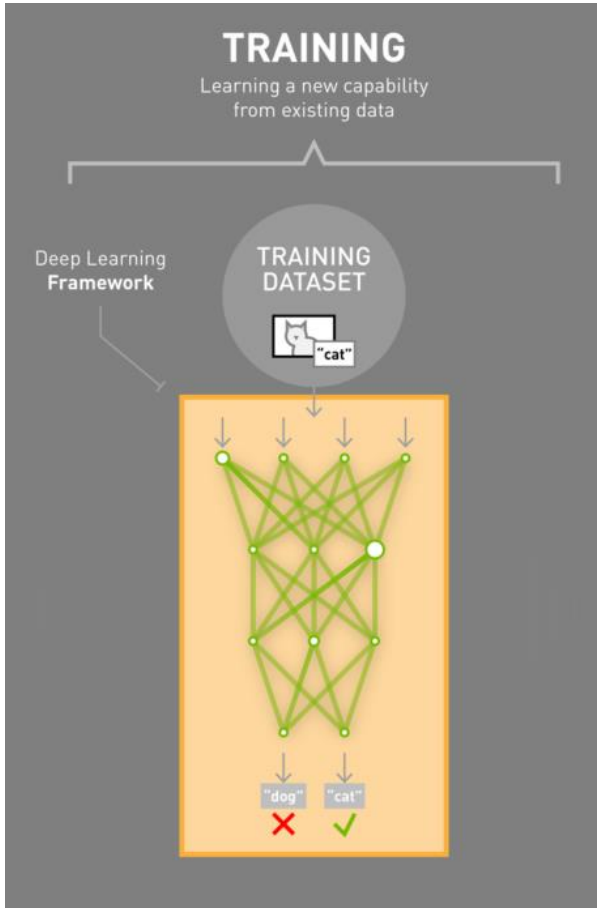
Image Classification with DIGITS

Dr. Alptekin Temizel
DLI Certified Instructor
Associate Professor, Graduate School of Informatics, METU

8 January 2018

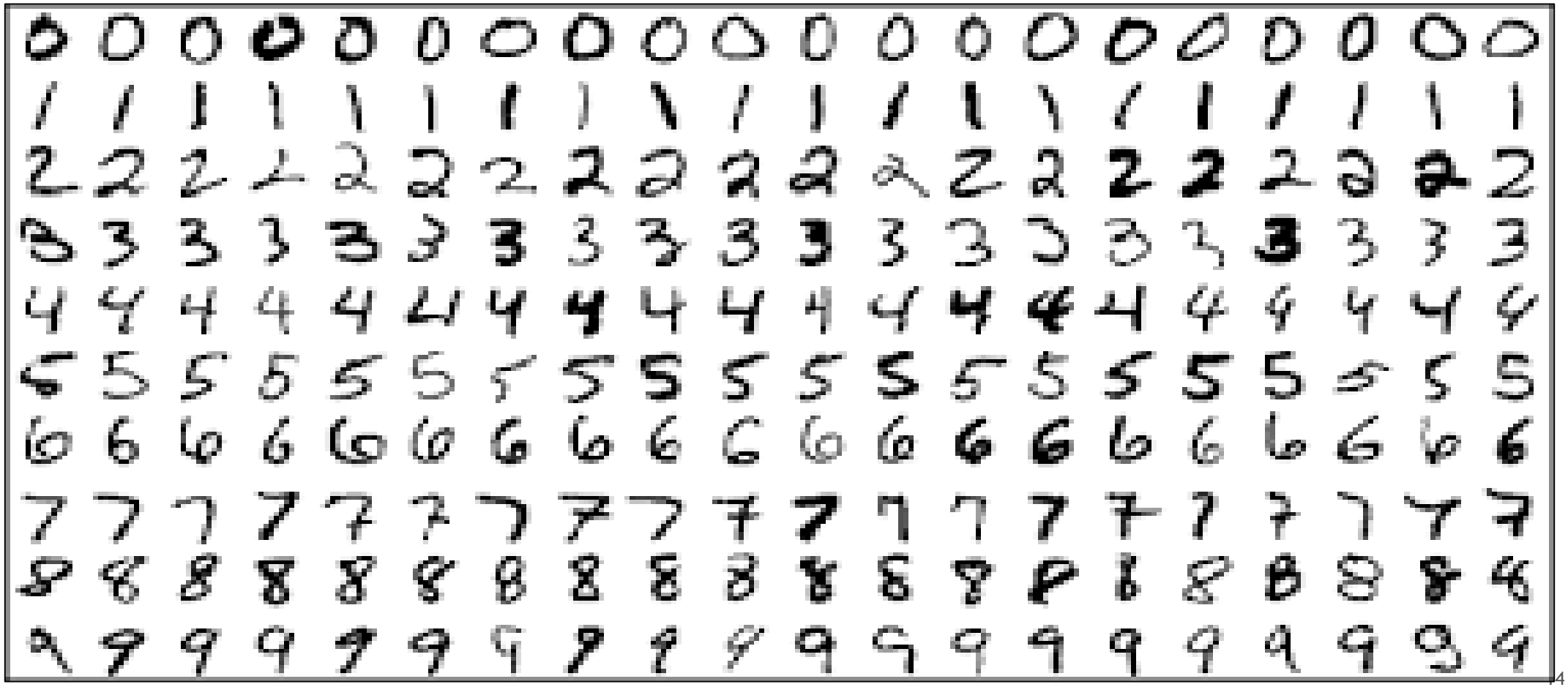
Training a network with data

Lab



HANDWRITTEN DIGIT RECOGNITION

HELLO WORLD of machine learning



WHAT THIS LAB IS

- An introduction to:
  - Workflow of training a network
  - Understanding the results

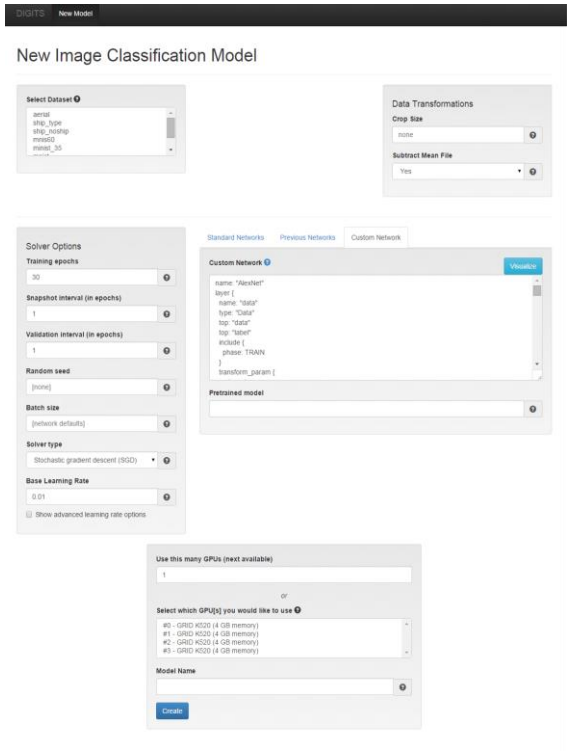
- Hands-on exercises using DIGITS for computer vision and classification

NVIDIA DIGITS

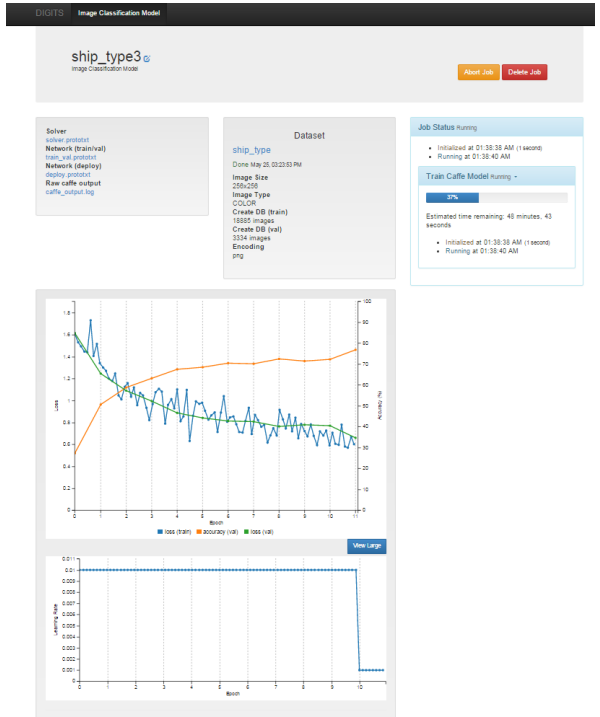
Interactive Deep Learning GPU Training System

Process Data

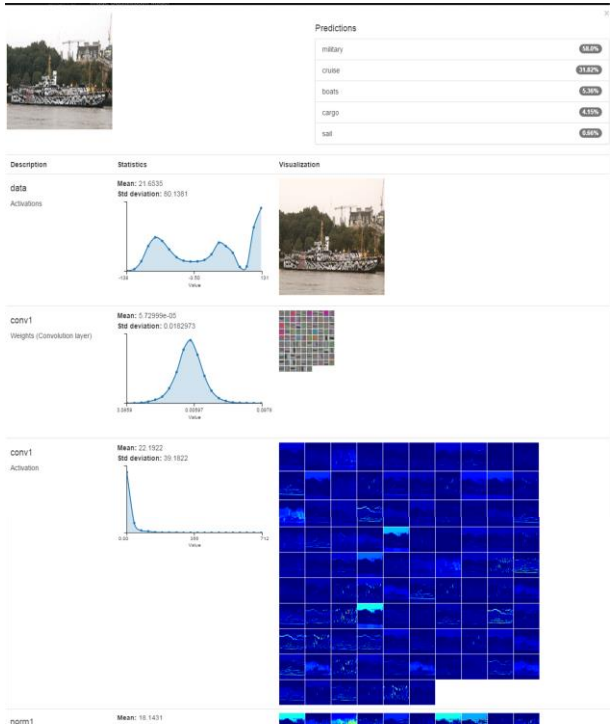
Configure DNN



Monitor Progress



Visualization



WHAT THIS LAB IS NOT

- Intro to machine learning from first principles
- Rigorous mathematical formalism of neural networks
- Survey of all the features and options of tools and frameworks

LAB OVERVIEW

- Learn about the workflow of Deep Learning
  - Load data
  - Expose a network to data
  - Evaluate model results
  - Try different techniques to improve initial results

LAB OVERVIEW

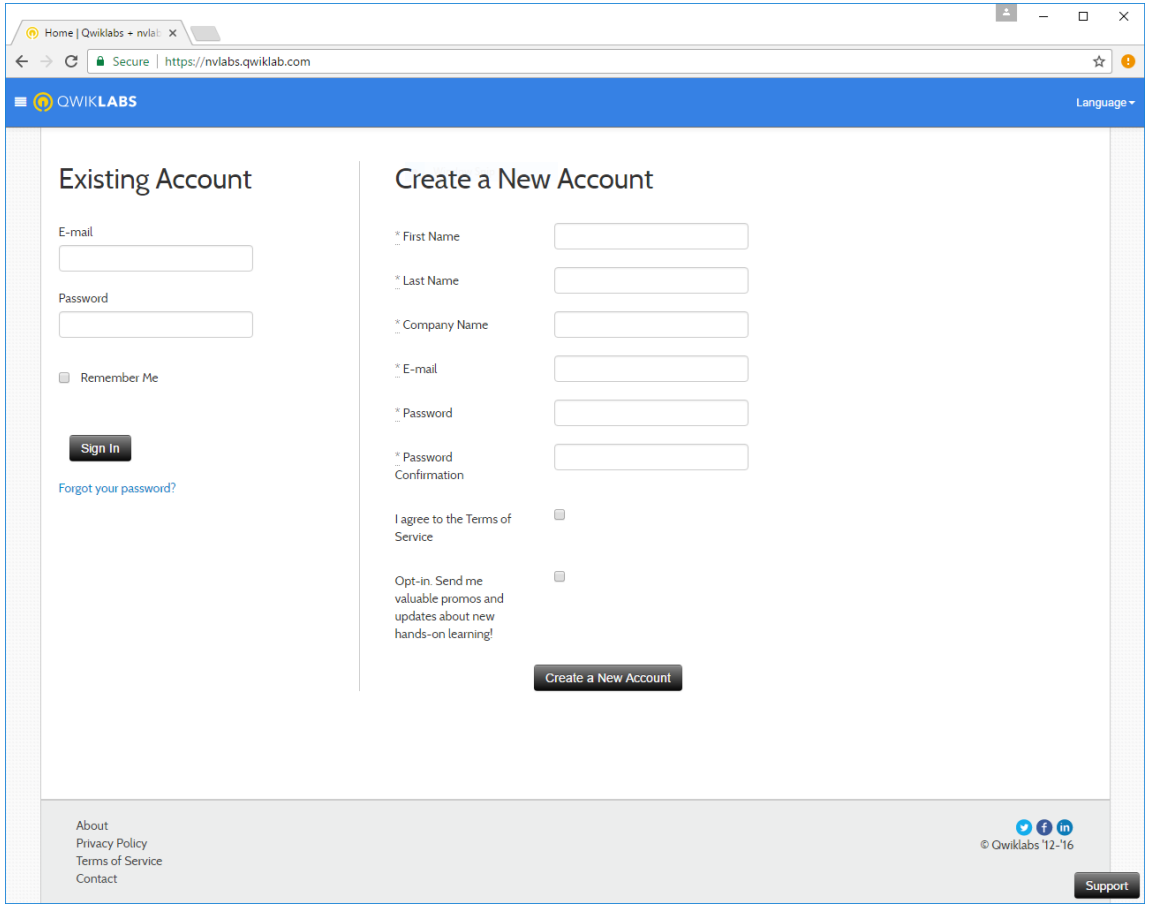
- We have clean, labeled data
- Our task is “supervised image classification”
- We will train a live network on GPU

LAUNCHING THE LAB

NAVIGATING TO QWIKLABS

1. Navigate to: https://nvlabs.qwiklab.com

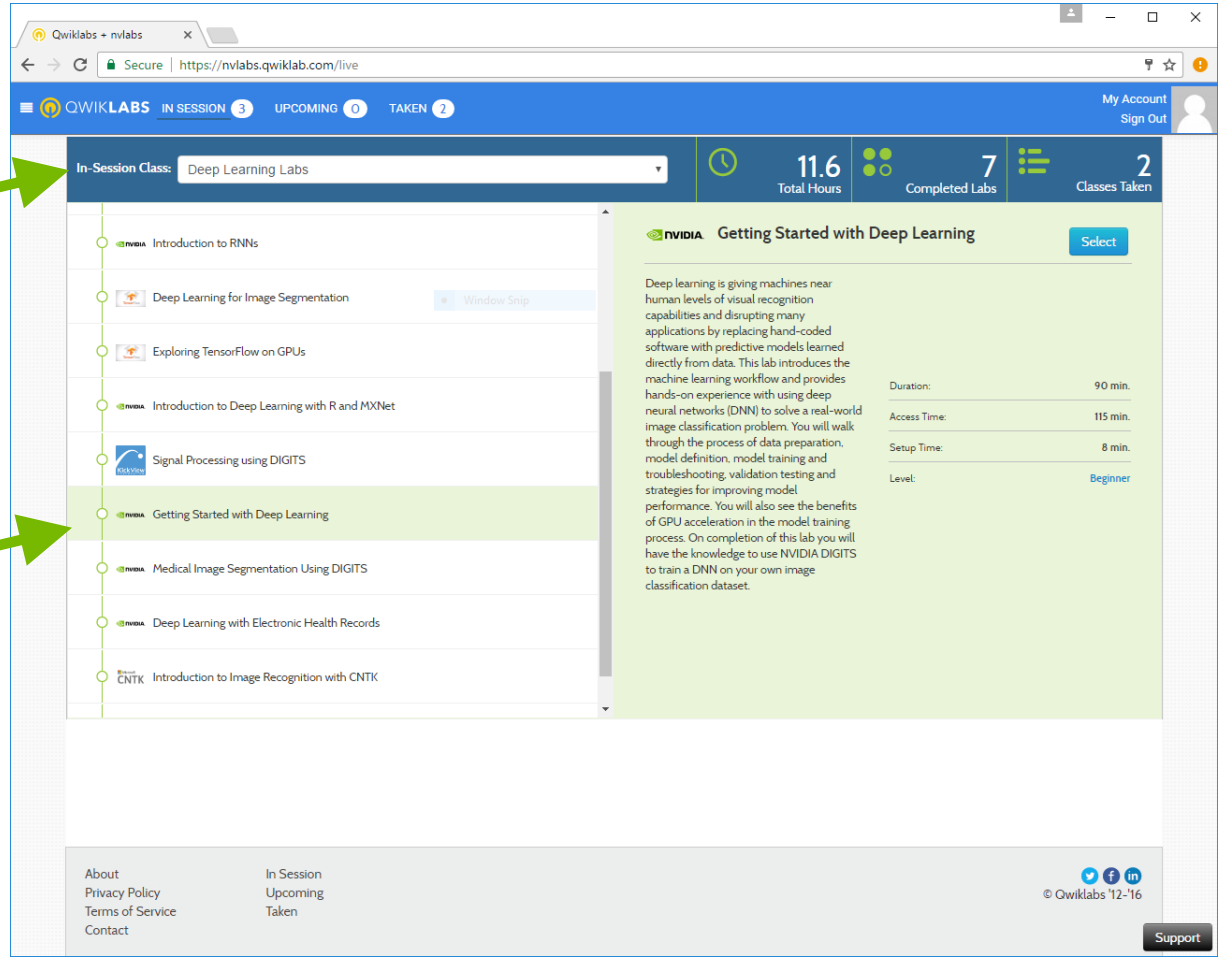
1. Login or create a new account



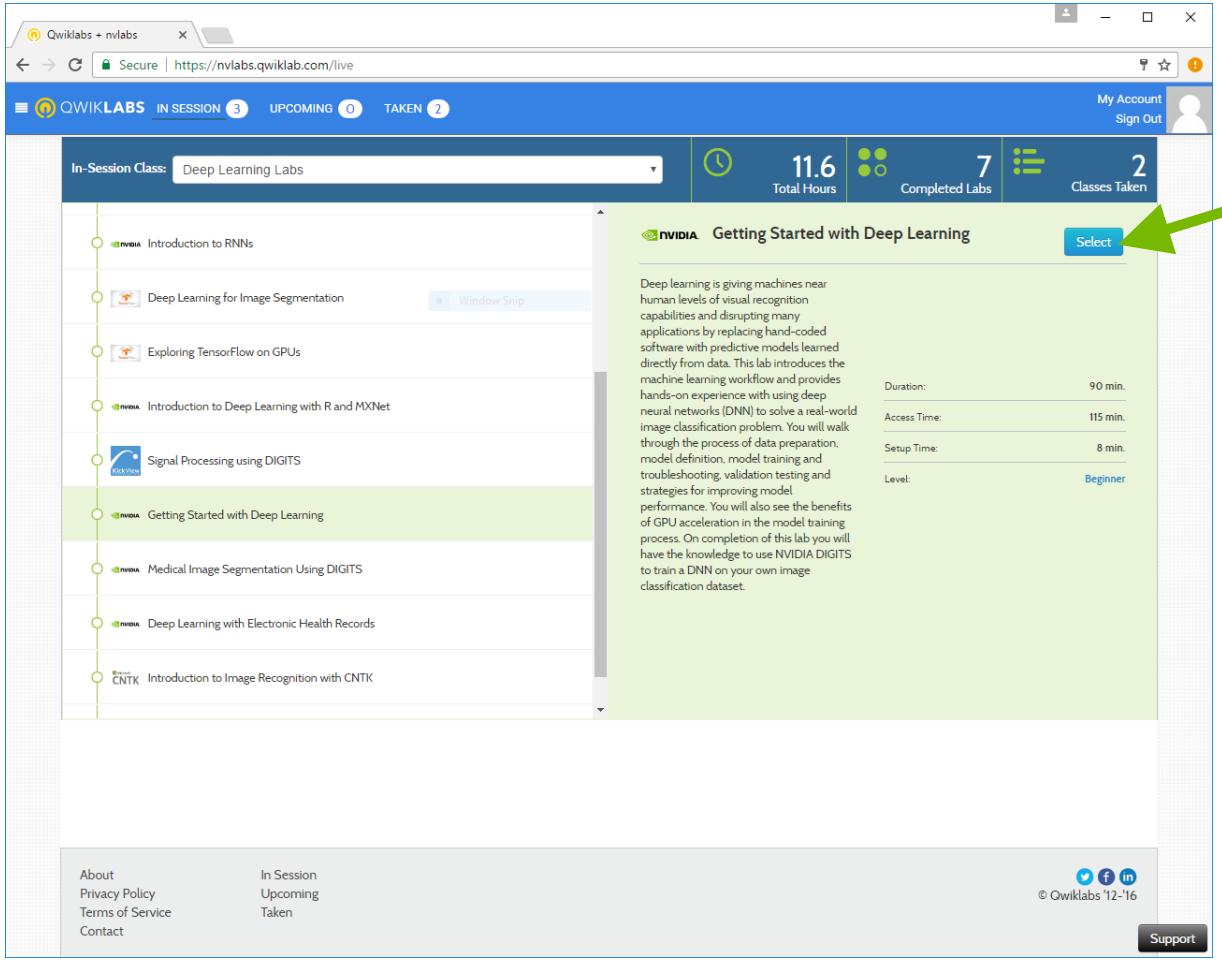
ACCESSING LAB ENVIRONMENT

3. Select the event specific In-Session Class in the upper left

3. Click the “Image Classification with DIGITS” Class from the list



LAUNCHING THE LAB ENVIRONMENT

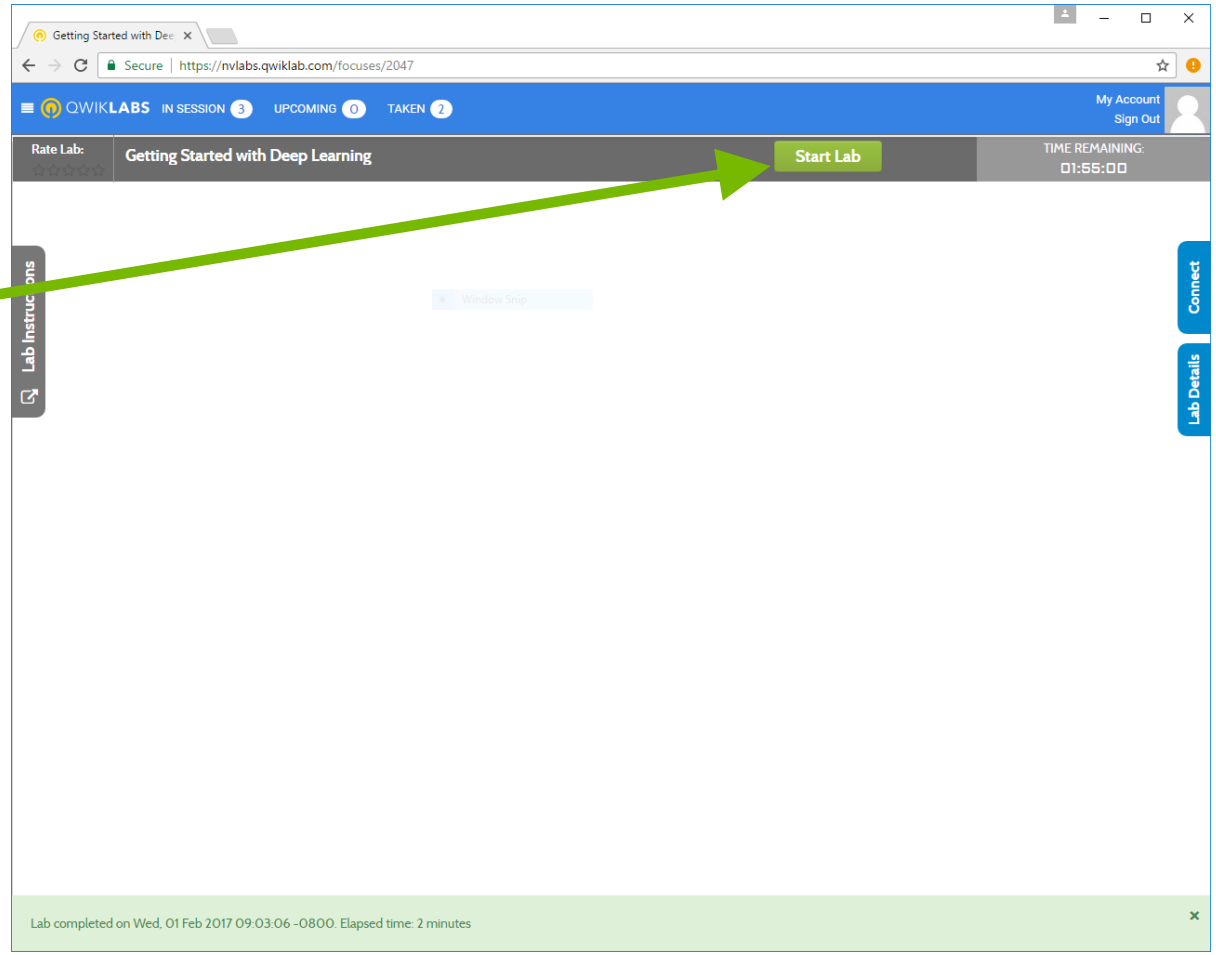


5. Click on the Select button to launch the lab environment

   - After a short wait, lab Connection information will be shown
   - Please ask Lab Assistants for help!

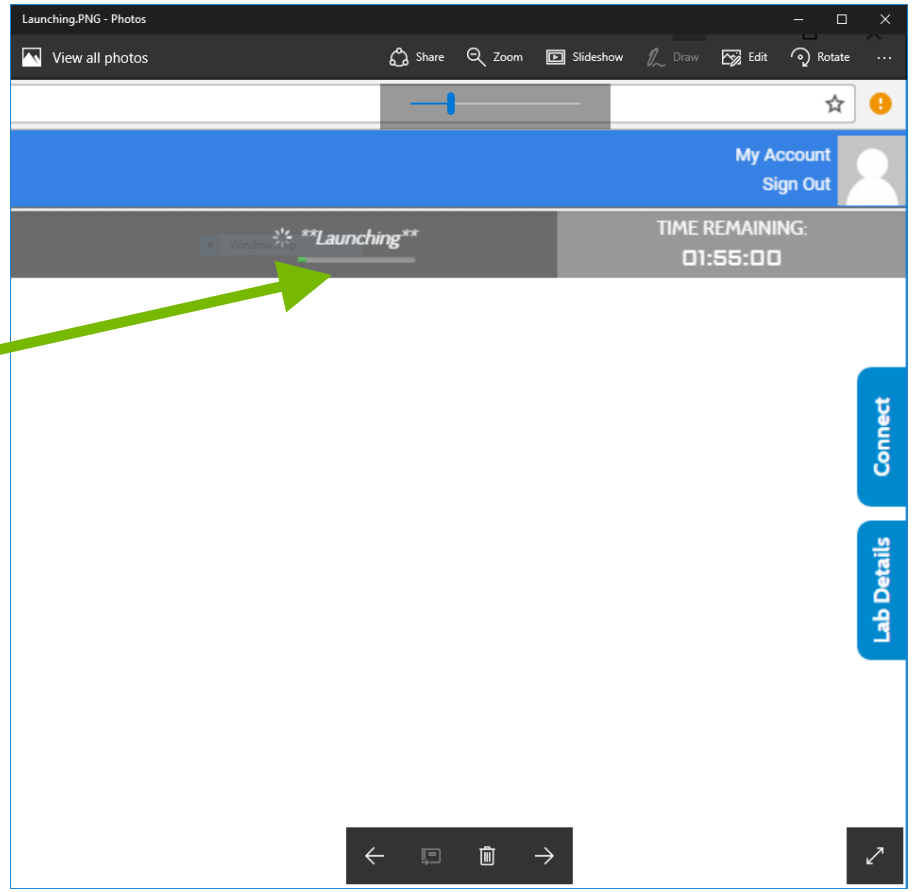
LAUNCHING THE LAB ENVIRONMENT

6. Click on the Start Lab button



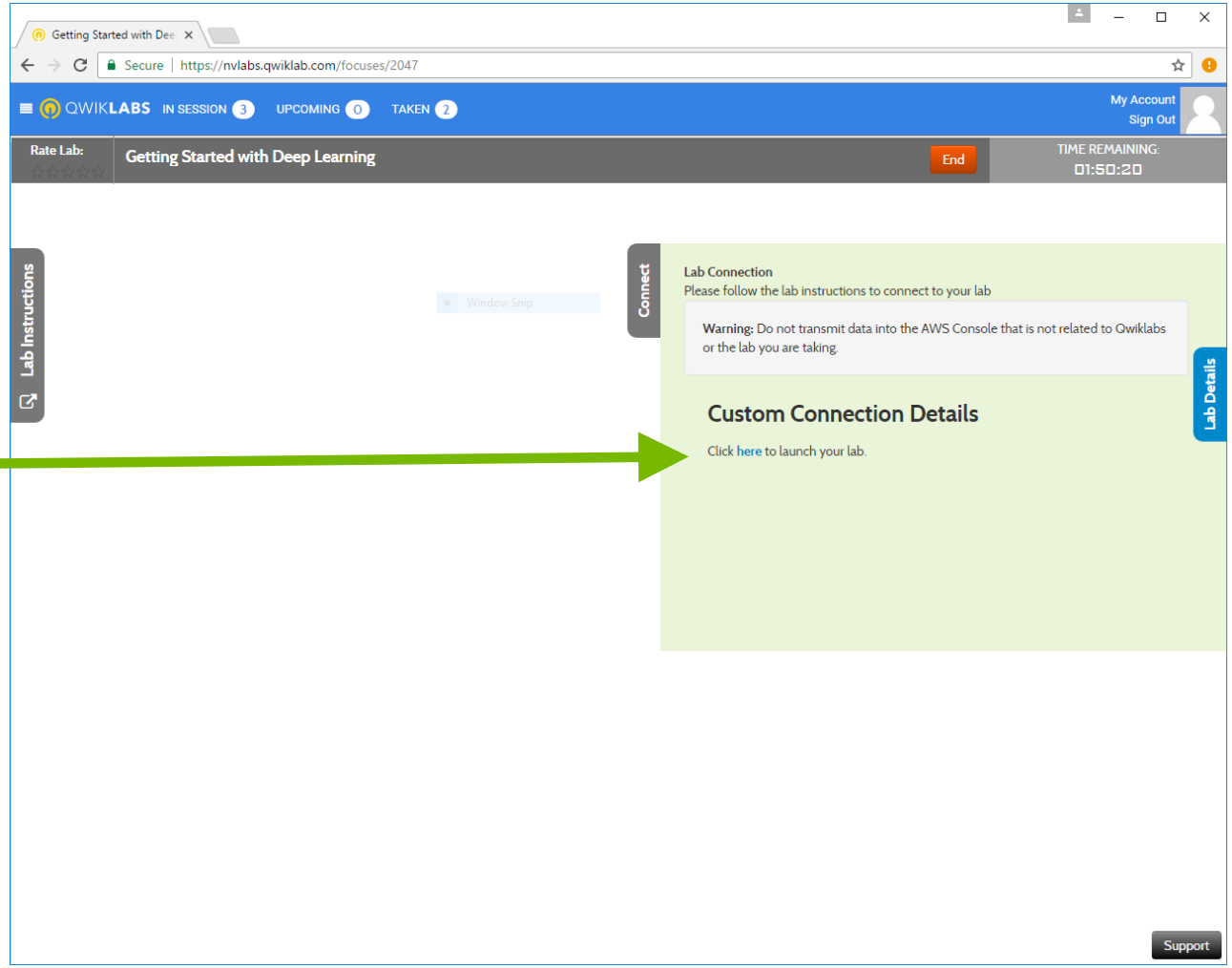
LAUNCHING THE LAB ENVIRONMENT

You should see that the lab environment is “launching” towards the upper-right corner



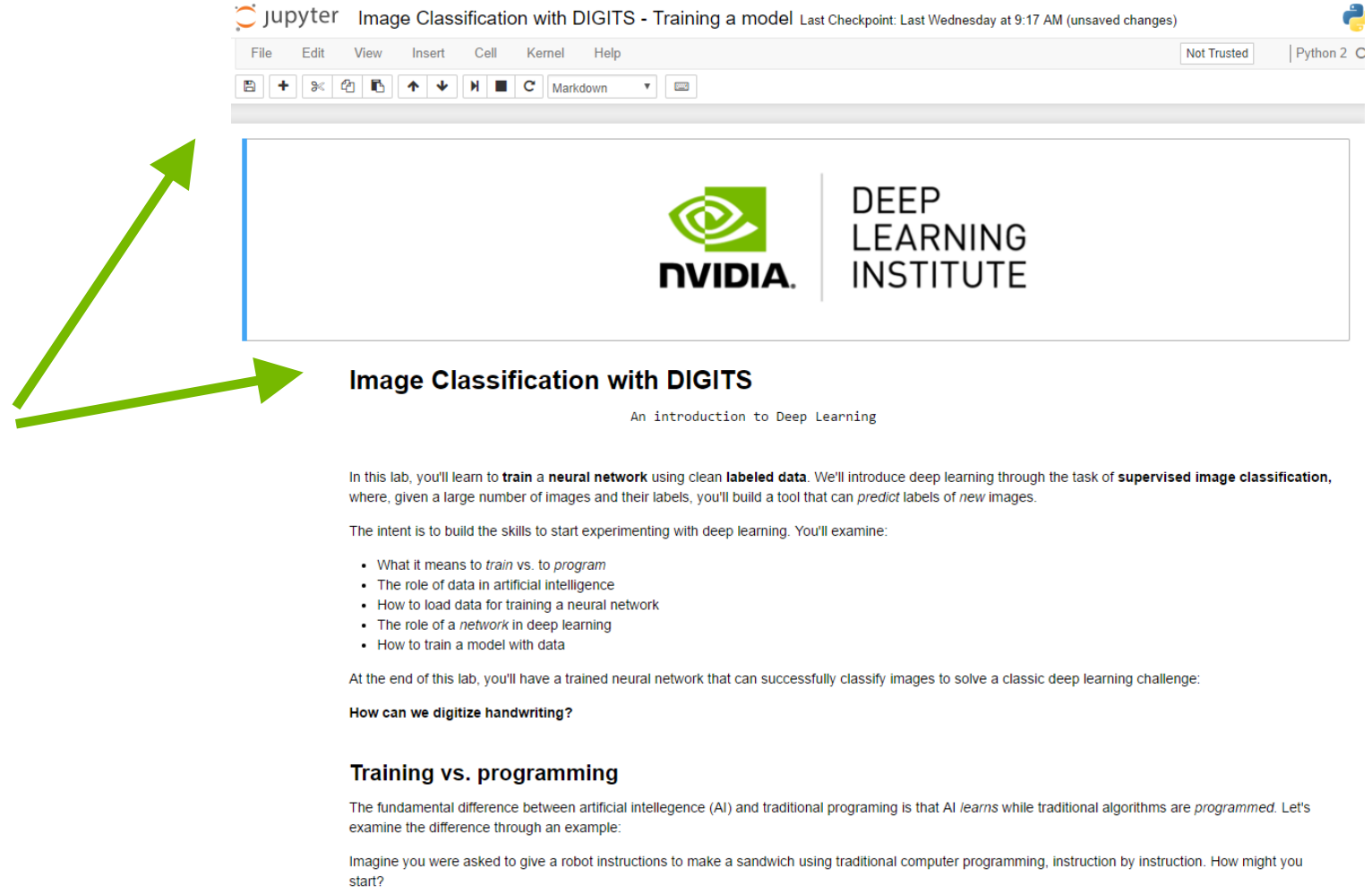
CONNECTING TO THE LAB ENVIRONMENT

7. Click on “here” to access your lab environment / Jupyter notebook



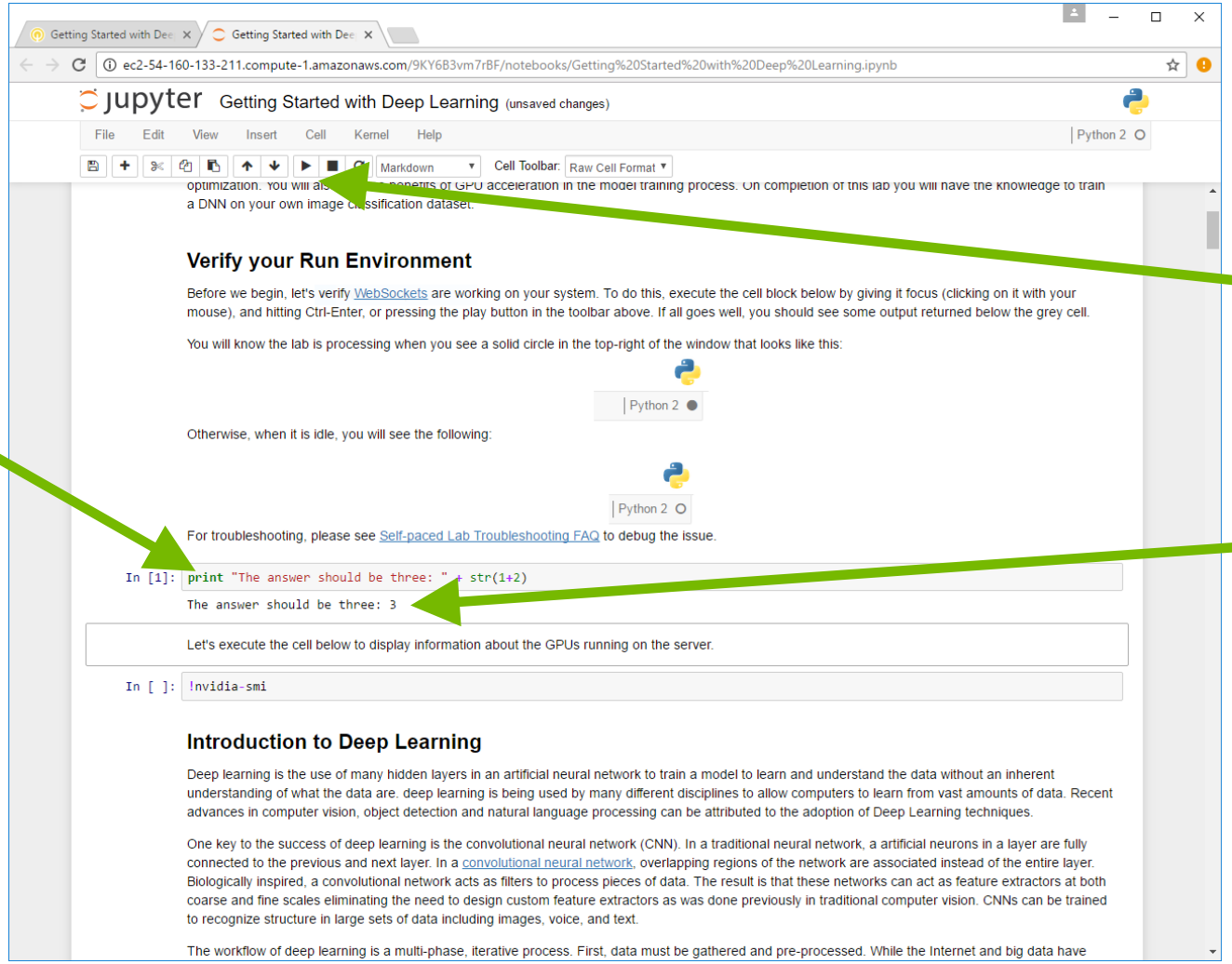
CONNECTING TO THE LAB ENVIRONMENT

You should see your "Image Classification with DIGITS" Jupyter notebook



JUPYTER NOTEBOOK

1. Place your cursor in the code

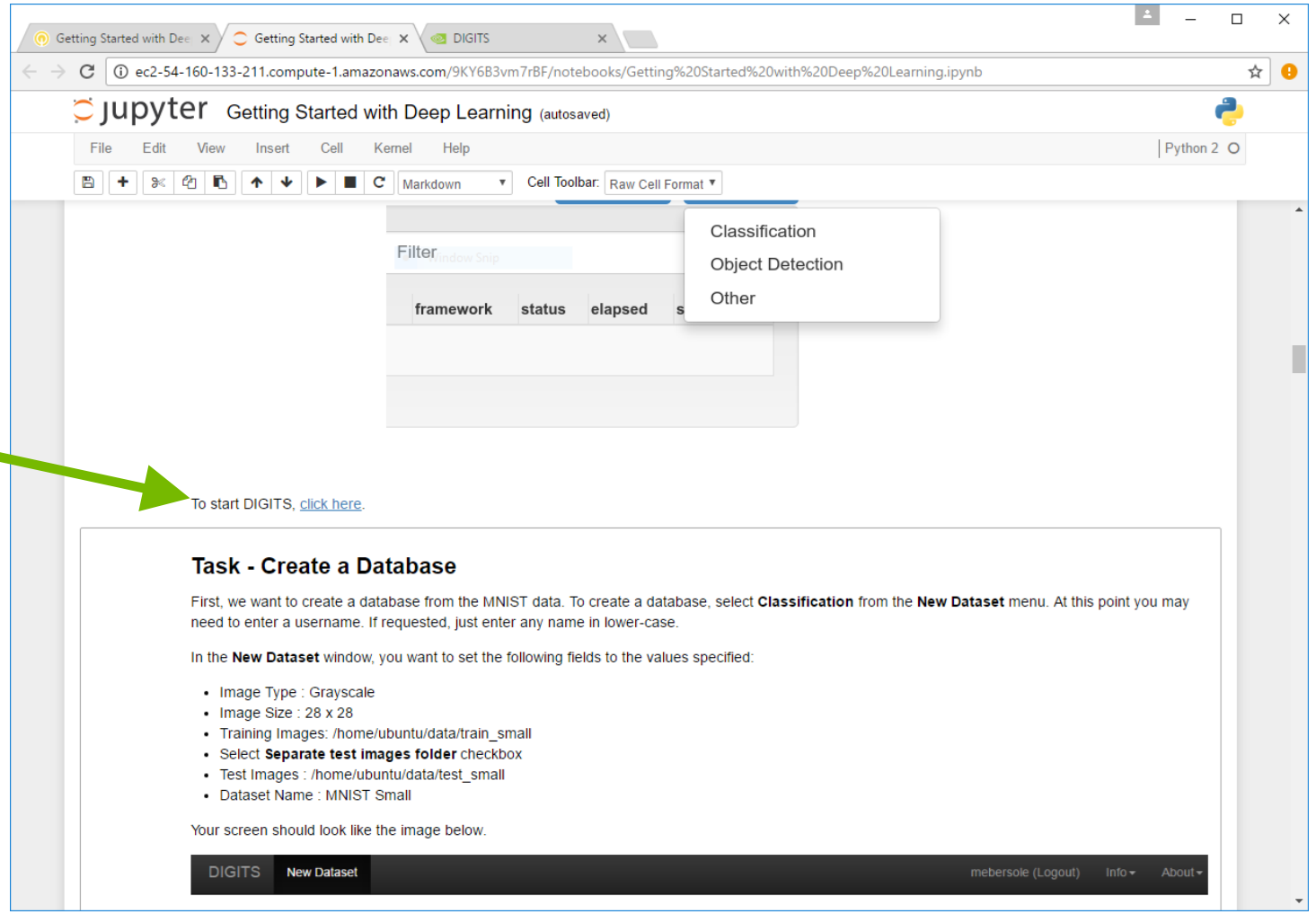


2. Click the "run cell" button

2. Confirm you receive the same result

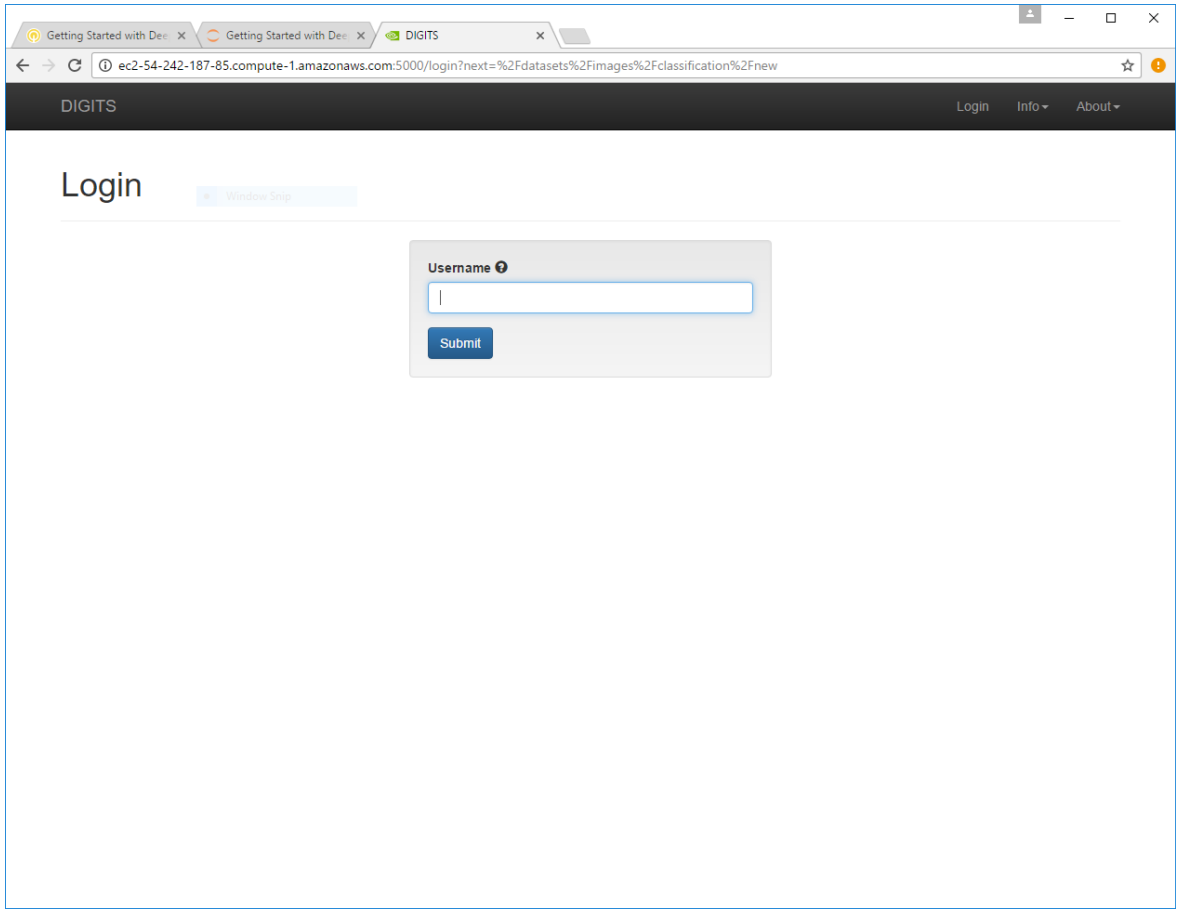
STARTING DIGITS

Instruction in Jupyter notebook will link you to DIGITS



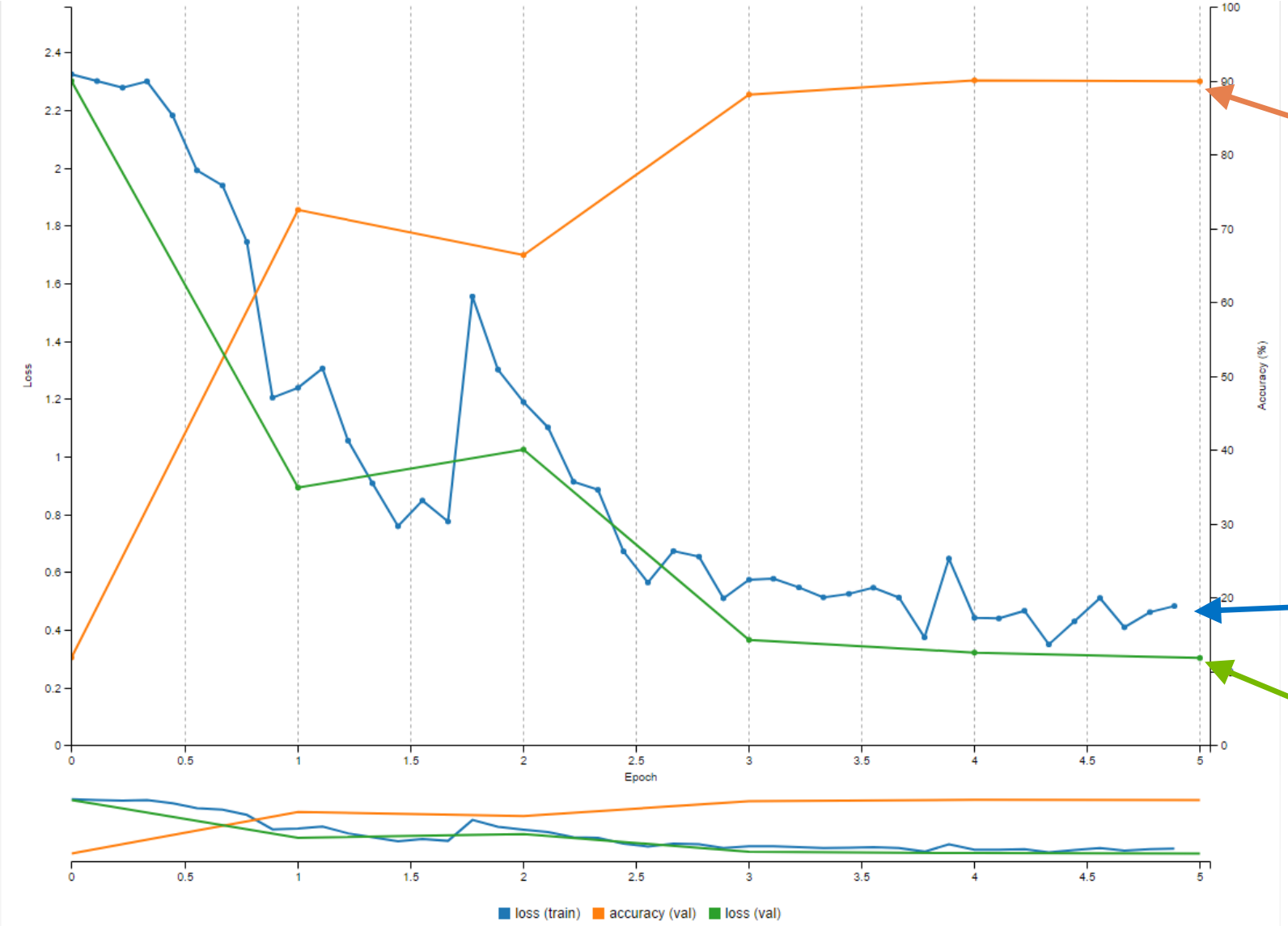
ACCESSING DIGITS

- Will be prompted to enter a username to access DIGITS
  - Can enter any username
  - Use lower case letters



Evaluating Performance

EVALUATE THE MODEL

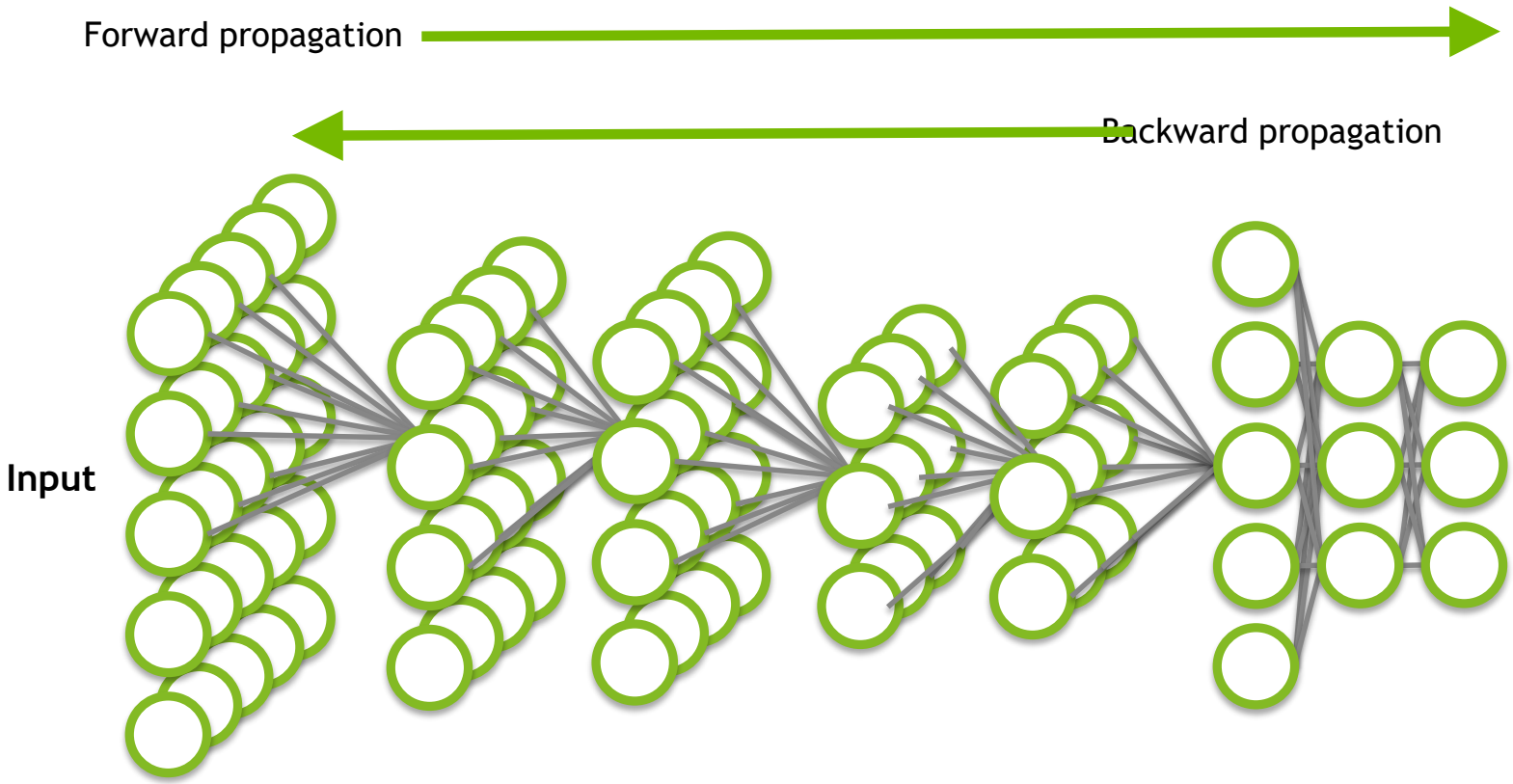


Accuracy obtained from validation dataset

Loss function (Training)

Loss function (Validation)

DEEP LEARNING APPROACH - TRAINING



Process

- Forward propagation yields an inferred label for each training image
- Loss function used to calculate difference between known label and predicted label for each image
- Weights are adjusted during backward propagation
- Repeat the process

Next Challenges

Ideas?

- Increase accuracy and confidence with similar data

- Generalize performance to more diverse data

Lab Review

More data

Full dataset (10 epochs)

- 99% of accuracy achieved
- No improvements in recognizing real-world images

	Defaults	Training+Data
	1 : 99.90 %	0 : 93.11 %
	2 : 69.03 %	2 : 87.23 %
	8 : 71.37 %	8 : 71.60 %
	8 : 85.07 %	8 : 79.72 %
	0 : 99.00 %	0 : 95.82 %
	8 : 99.69 %	8 : 100.0 %
	8 : 54.75 %	2 : 70.57 %

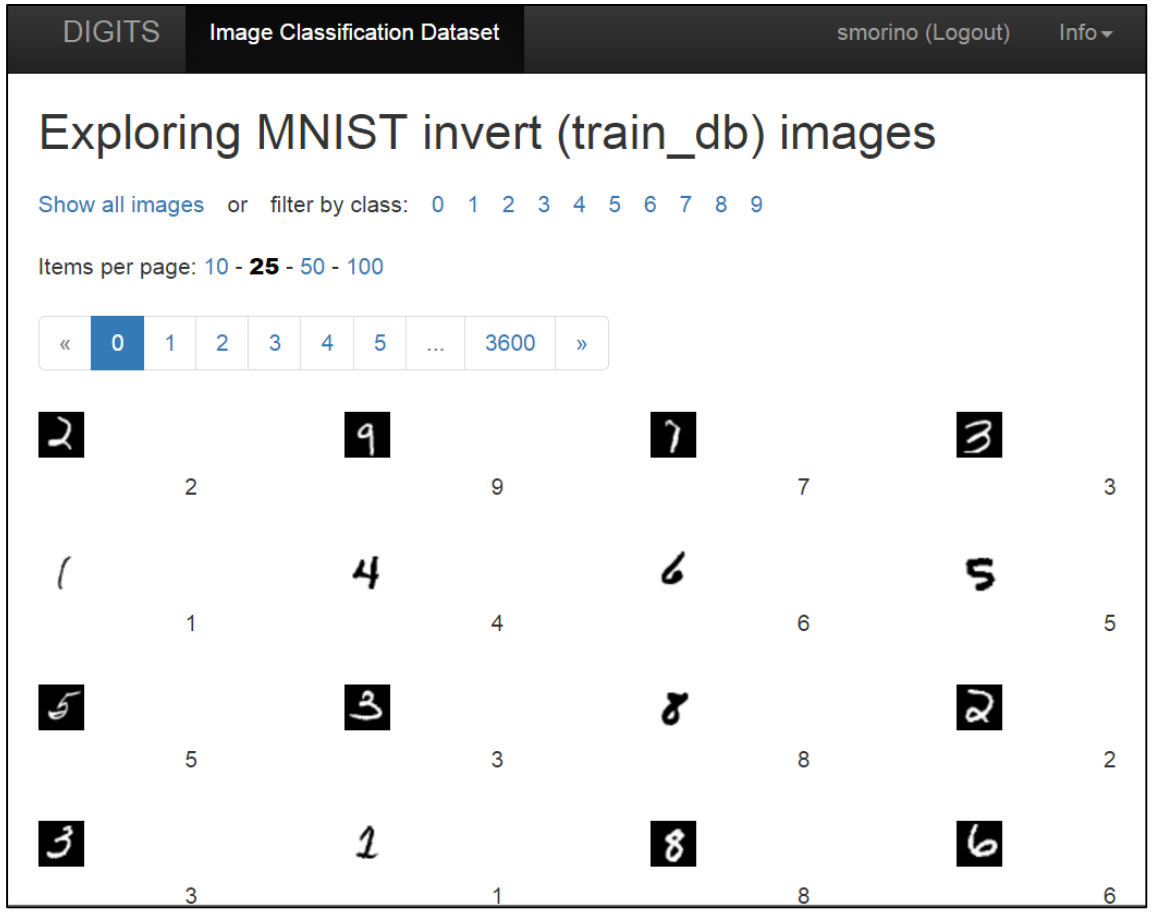
DATA AUGMENTATION

Adding inverted images (10 epochs)

	SMALL DATASET	FULL DATASET	+INVERTED
	1 : 99.90 %	0 : 93.11 %	1 : 90.84 %
	2 : 69.03 %	2 : 87.23 %	2 : 89.44 %
	8 : 71.37 %	8 : 71.60 %	3 : 100.0 %
	8 : 85.07 %	8 : 79.72 %	4 : 100.0 %
	0 : 99.00 %	0 : 95.82 %	7 : 82.84 %
	8 : 99.69 %	8 : 100.0 %	8 : 100.0 %
	8 : 54.75 %	2 : 70.57 %	2 : 96.27 %

DATA AUGMENTATION

Adding Inverted Images




```
keras.preprocessing.image.ImageDataGenerator(featurewise_center=False,
    samplewise_center=False,
    featurewise_std_normalization=False,
    samplewise_std_normalization=False,
    zca_whitening=False,
    zca_epsilon=1e-6,
    rotation_range=0.,
    width_shift_range=0.,
    height_shift_range=0.,
    shear_range=0.,
    zoom_range=0.,
    channel_shift_range=0.,
    fill_mode='nearest',
    cval=0.,
    horizontal_flip=False,
    vertical_flip=False,
    rescale=None,
    preprocessing_function=None,
    data_format=K.image_data_format())
```

MODIFIED NETWORK

Adding filters and ReLU layer (10 epochs)

	SMALL DATASET	FULL DATASET	+INVERTED	ADDING LAYER
	1 : 99.90 %	0 : 93.11 %	1 : 90.84 %	1 : 59.18 %
	2 : 69.03 %	2 : 87.23 %	2 : 89.44 %	2 : 93.39 %
	8 : 71.37 %	8 : 71.60 %	3 : 100.0 %	3 : 100.0 %
	8 : 85.07 %	8 : 79.72 %	4 : 100.0 %	4 : 100.0 %
	0 : 99.00 %	0 : 95.82 %	7 : 82.84 %	2 : 62.52 %
	8 : 99.69 %	8 : 100.0 %	8 : 100.0 %	8 : 100.0 %
	8 : 54.75 %	2 : 70.57 %	2 : 96.27 %	8 : 70.83 %

MODIFY THE NETWORK

Necessary for less “solved” challenges.

```
layer {
        name: "pool1“
        type: "Pooling“
        …
}

layer {
        name: "reluP1"
        type: "ReLU"
        bottom: "pool1"
        top: "pool1"
}

layer {
        name: "reluP1“
```

```
layer {
  name: "conv1"
  type: "Convolution"
        ...
        convolution_param {
        num_output: 75
        ...
layer {
        name: "conv2"
        type: "Convolution"
        ...
        convolution_param {
        num_output: 100
        ...
```

Next Steps

- Experiment with Image Classification
  - Different datasets
  - Increase performance
- Learn to train existing networks with data for other challenges
- Learn about network construction
- Learn about how to create an image classifier with other frameworks
  - Caffe/Keras
  - Tensorflow
  - Etc.