DEEP
LEARNING
INSTITUTE

Image Classification with DIGITS

NVIDIA Deep Learning Institute

DEEP LEARNING INSTITUTE

DLI Mission

Helping people solve challenging problems using AI and deep learning.

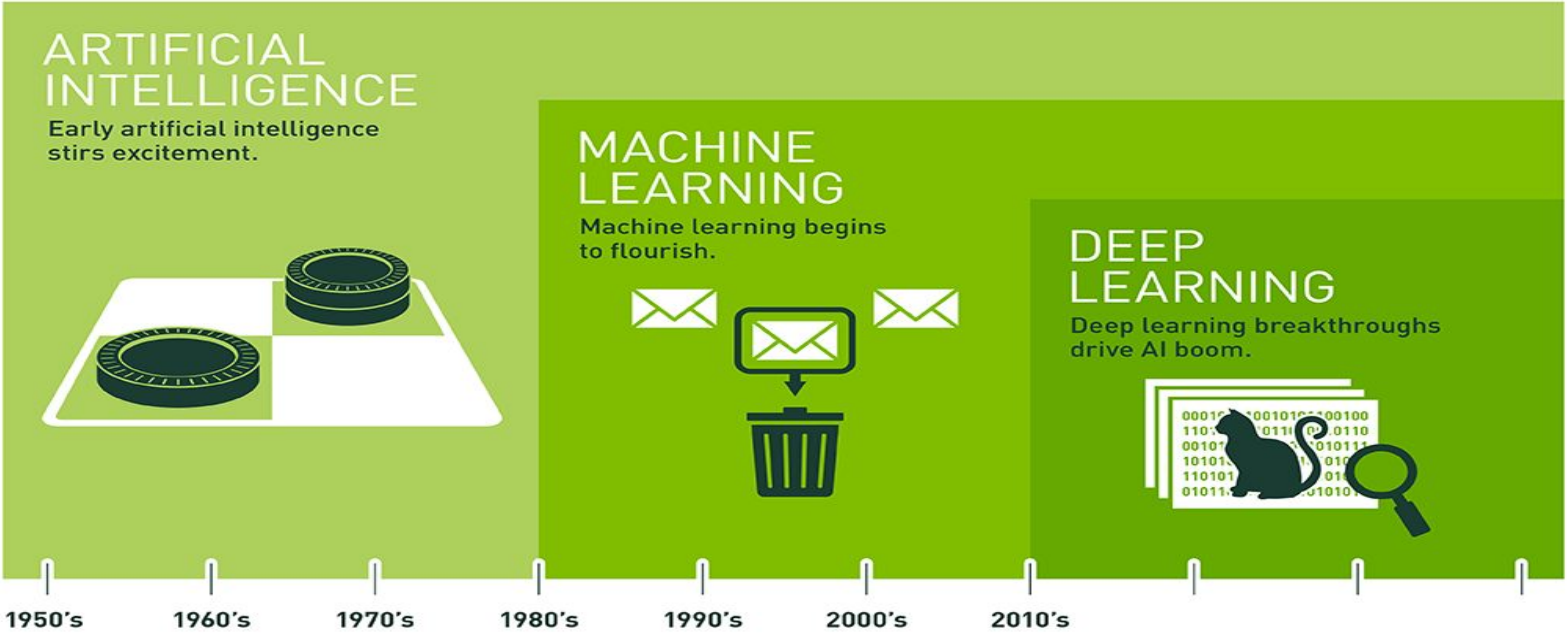
- Developers, data scientists and engineers
- Self-driving cars, healthcare and robotics
- Training, optimizing, and deploying deep neural networks

Agenda

- Intro to Deep Learning
- Training vs. Programming
- Train our first neural network - Lab
- How networks “learn”
- Increasing performance - Lab
- Next Steps

WHAT IS DEEP LEARNING?

ACCOMPLISHING COMPLEX GOALS

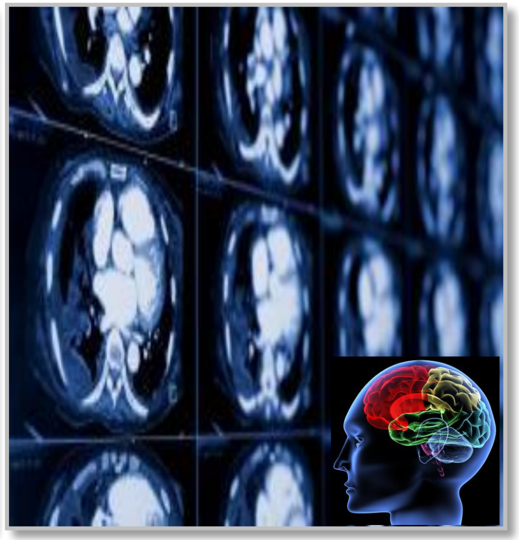


SWEEPING ACROSS INDUSTRIES

Internet Services

- Image/Video classification
- Speech recognition
- Natural language processing

Medicine



- Cancer cell detection
- Diabetic grading
- Drug discovery

Media & Entertainment

- Video captioning
- Content based search
- Real time translation

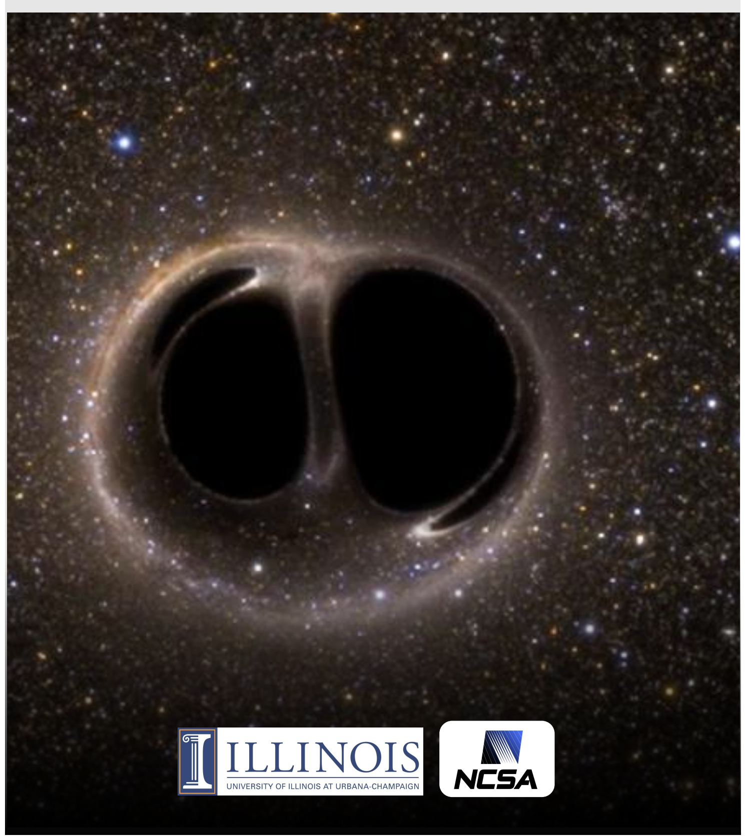
Security & Defense

- Face recognition
- Video surveillance
- Cyber security

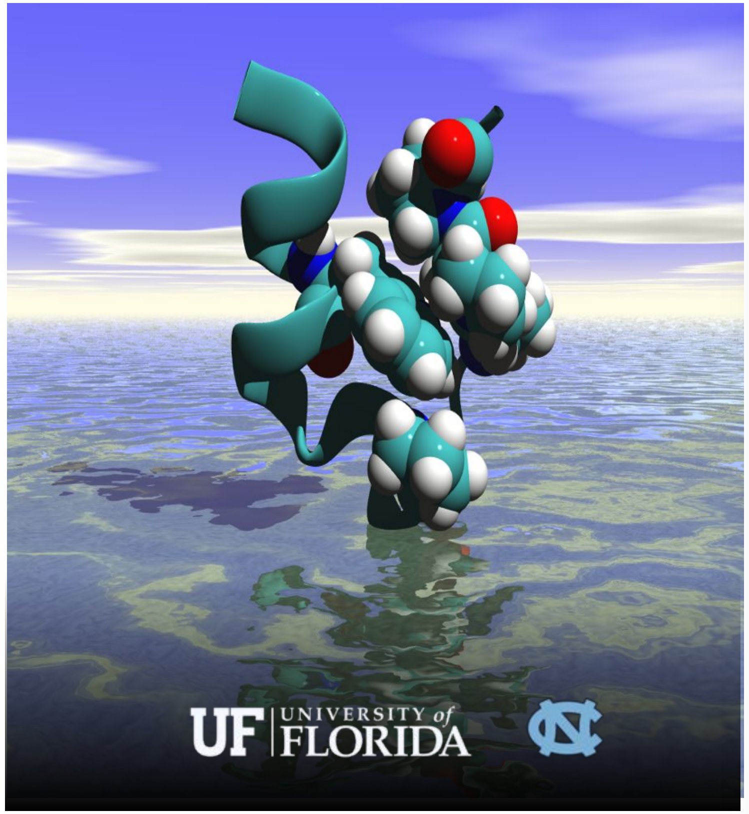
Autonomous Machines

- Pedestrian detection
- Lane tracking
- Recognize traffic signs

TRANSFORMING RESEARCH



"Seeing" Gravity In Real Time



Accelerating Drug Discovery

92%
believe AI will impact their work

93%
using deep learning seeing positive results

inside HPC

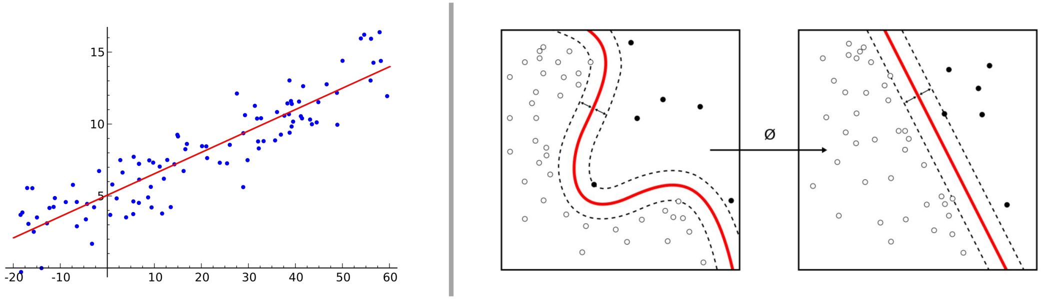
insideHPC.com Survey
November 2016

Difference in Workflow

Classic Machine Learning [1990 : now]

Input → Hand Designed **Features** → Model / Mapping → Output

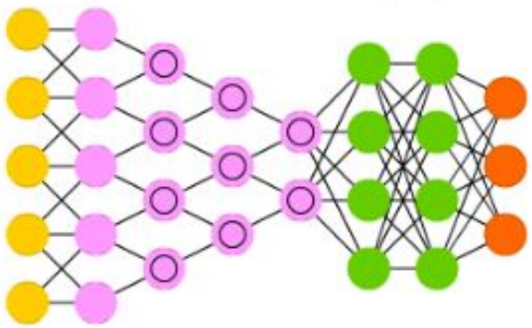
Examples [Regression and SVMs]



Deep/End-to-End Learning [2012 : now]

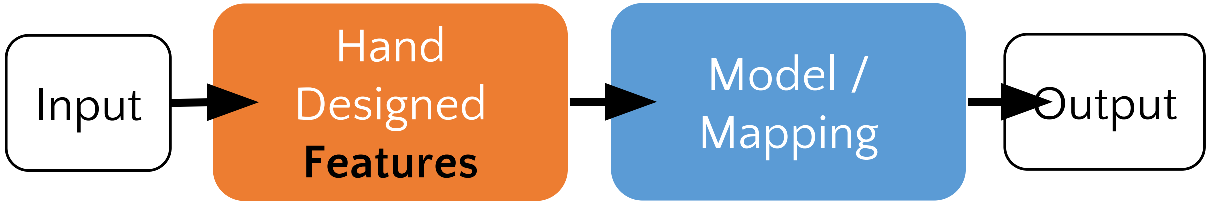
Input → Simple Features → Complex Features → Model/ Mapping → Output

Example [Conv Net]

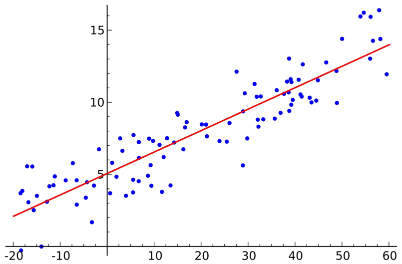


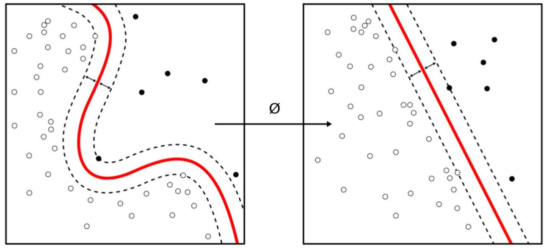
Traditional Workflow

Classic Machine Learning [1990 : now]



Examples [Regression and SVMs]





Challenge in Slack channel: How would you describe this image to someone (or something) blind?

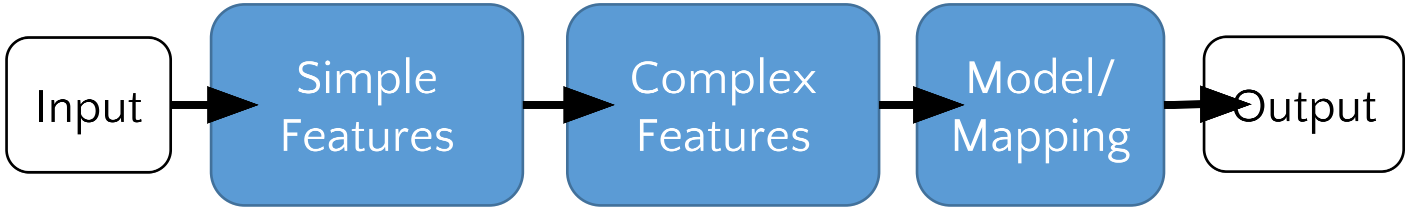
Difficult: From it's raw pixels.
Medium: From geometric primitives (lines, curves, colors)
Easy: Using any words that you may know

Deep Learning Workflow

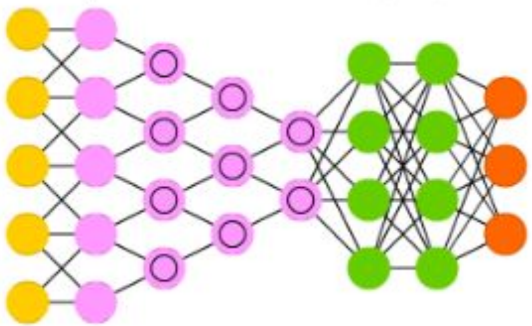
Experience: <u>Trust</u> Neural Network to learn features and model by providing inputs and outputs.

Key Skill: Experience (data) creation

Deep/End-to-End Learning [2012 : now]



Example [Conv Net]



INPUT TO OUTPUT

Louie or Not Louie?

1 = Louie
0= Not Louie
.85 = 85% confident Louie

INPUT TO OUTPUT

Louie or Not Louie?

1 = Louie
0= Not Louie
.85 = 85% confident Louie

Yes, this beagle is Louie!

INPUT TO OUTPUT

Louie or Not Louie?

1 = Louie
0= Not Louie
.85 = 85% confident Louie

No, not Louie!

INPUT TO OUTPUT

Louie or Not Louie?

1 = Louie
0= Not Louie
.85 = 85% confident Louie

No, not Louie!

INPUT TO OUTPUT

Louie or Not Louie?

1 = Louie
0= Not Louie
.85 = 85% confident Louie

Yup, that's Louie!

INPUT TO OUTPUT

Louie or Not Louie?

1 = Louie
0= Not Louie
.85 = 85% confident Louie

Yea, that's Louie!

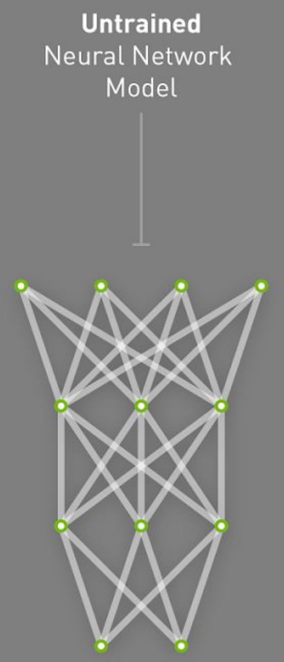
INPUT TO OUTPUT

Louie or Not Louie?

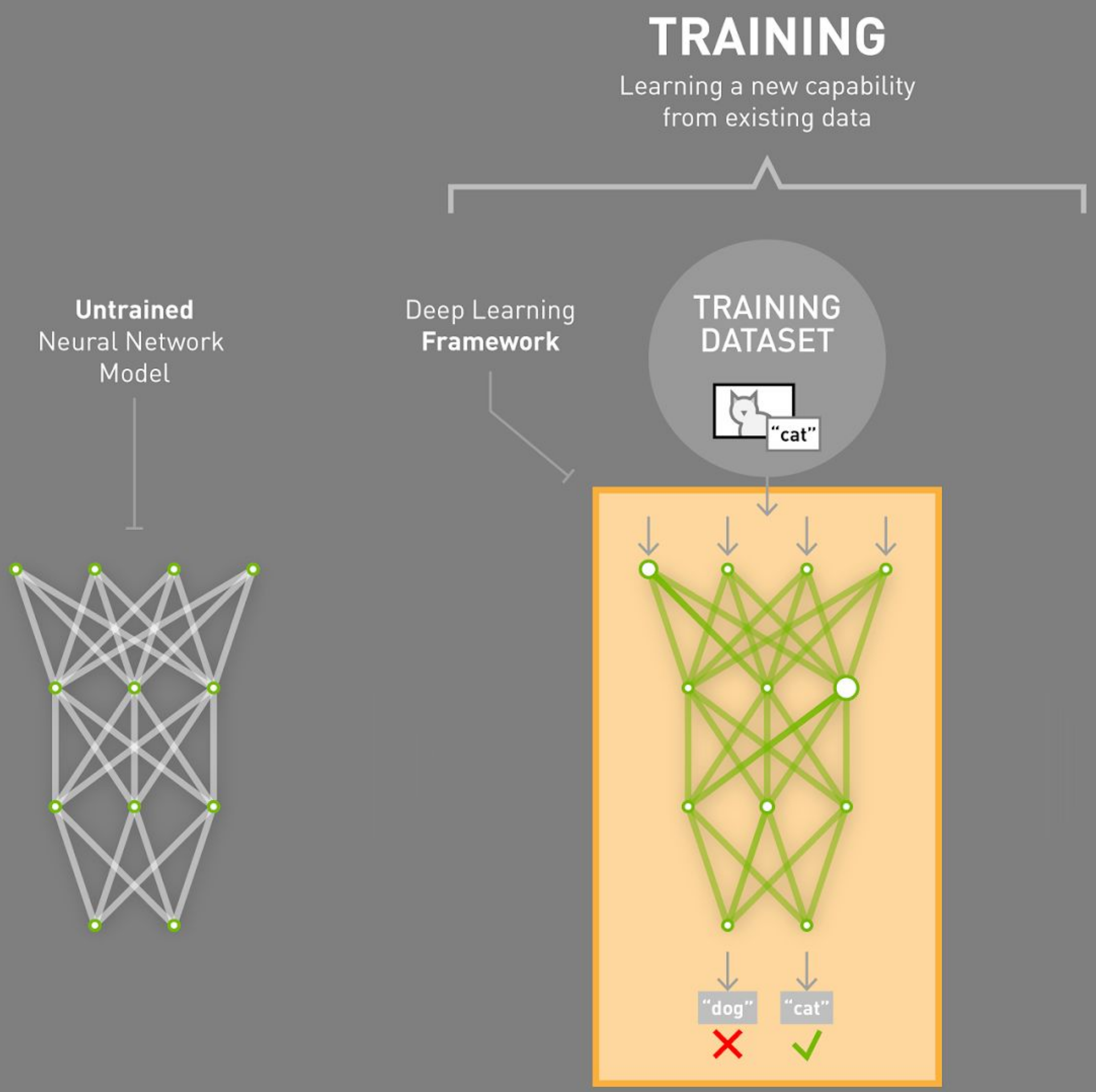
1 = Louie
0= Not Louie
.85 = 85% confident Louie

Yes! Another epoch?

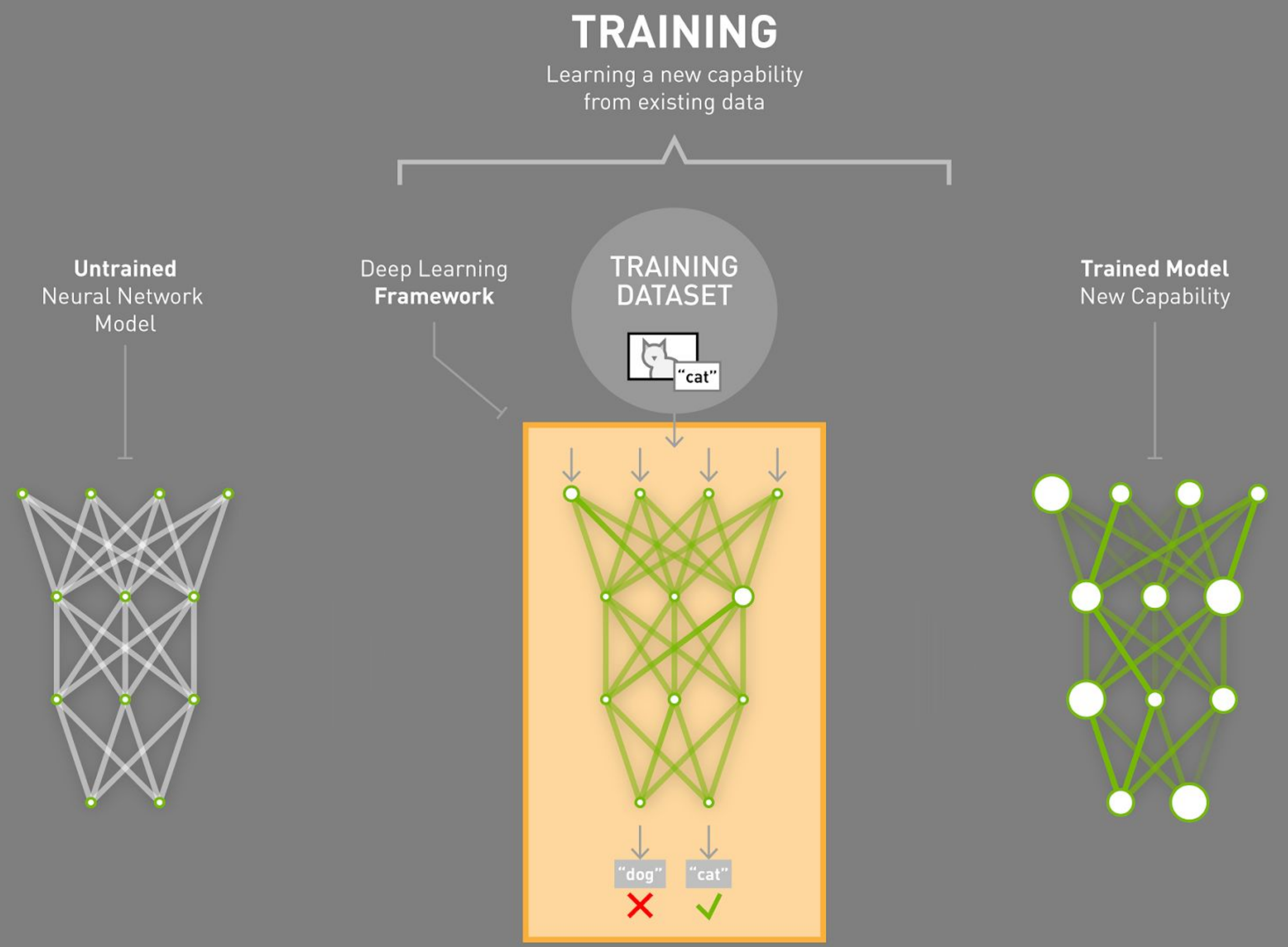
DEEP LEARNING

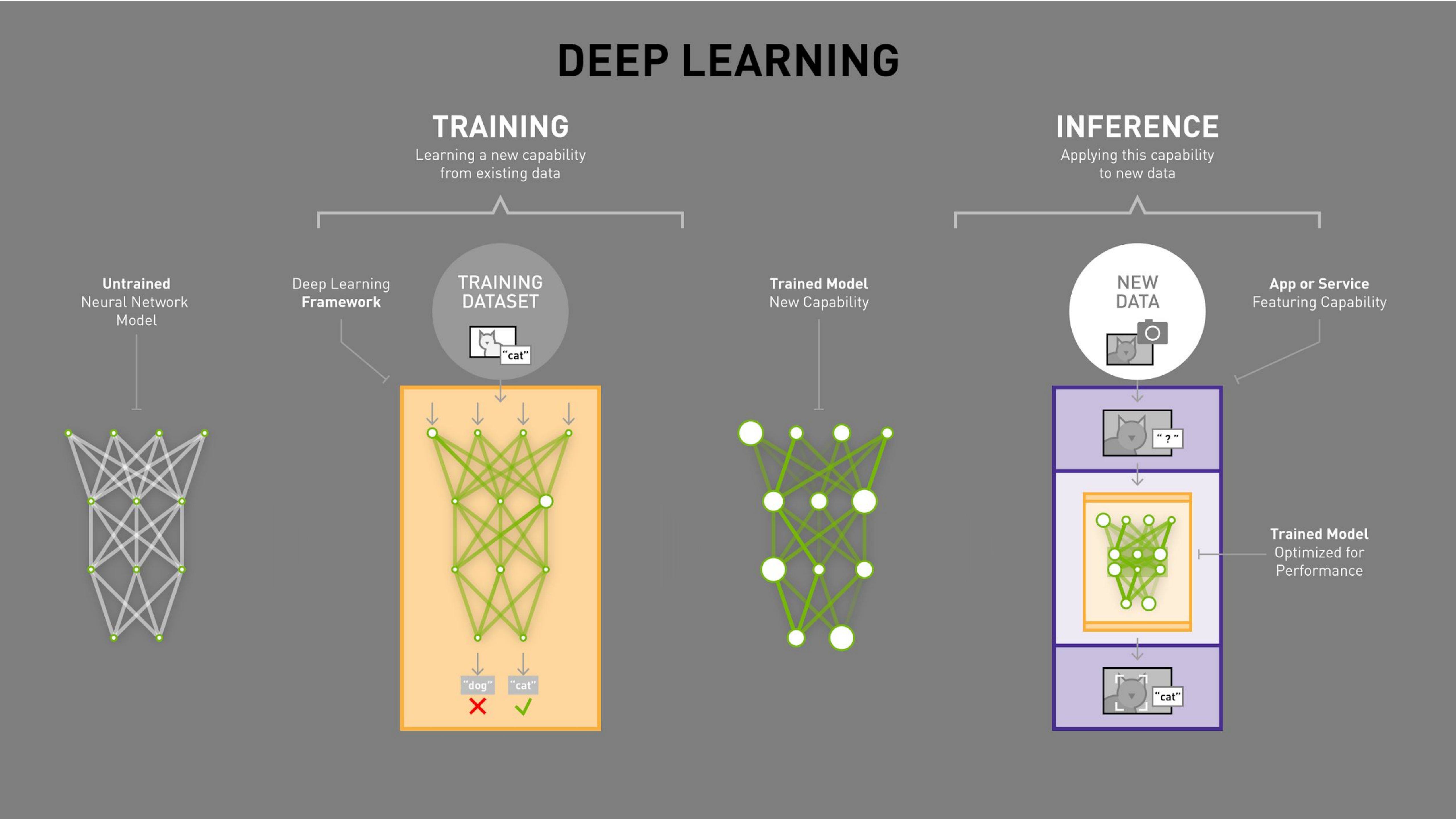


DEEP LEARNING



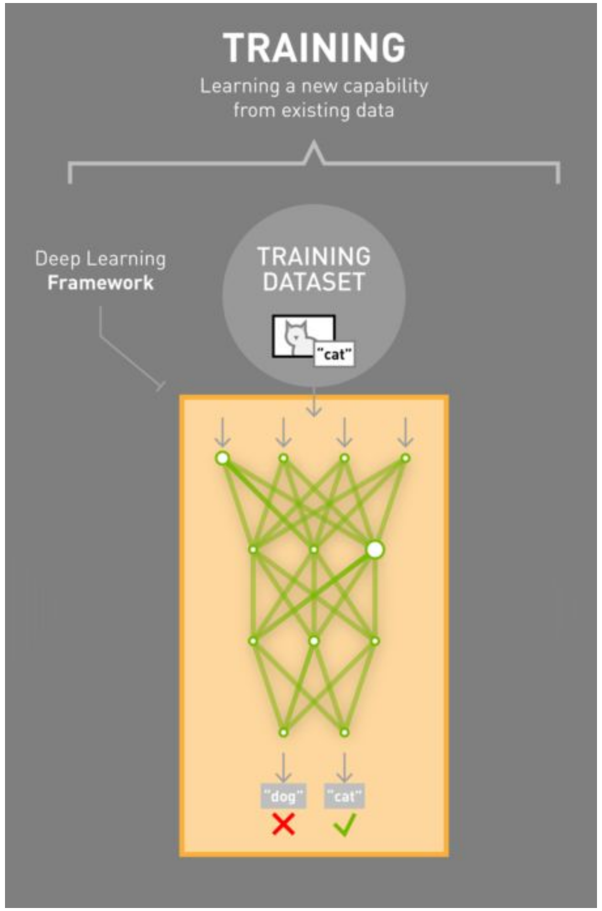
DEEP LEARNING




DEEP LEARNING
TRAINING
Learning a new capability from existing data
INFERENCE
Applying this capability to new data
Untrained Neural Network Model
Deep Learning Framework
TRAINING DATASET
"cat"
"dog"
"cat"
Trained Model New Capability
NEW DATA
App or Service Featuring Capability
" ? "
Trained Model Optimized for Performance
"cat"

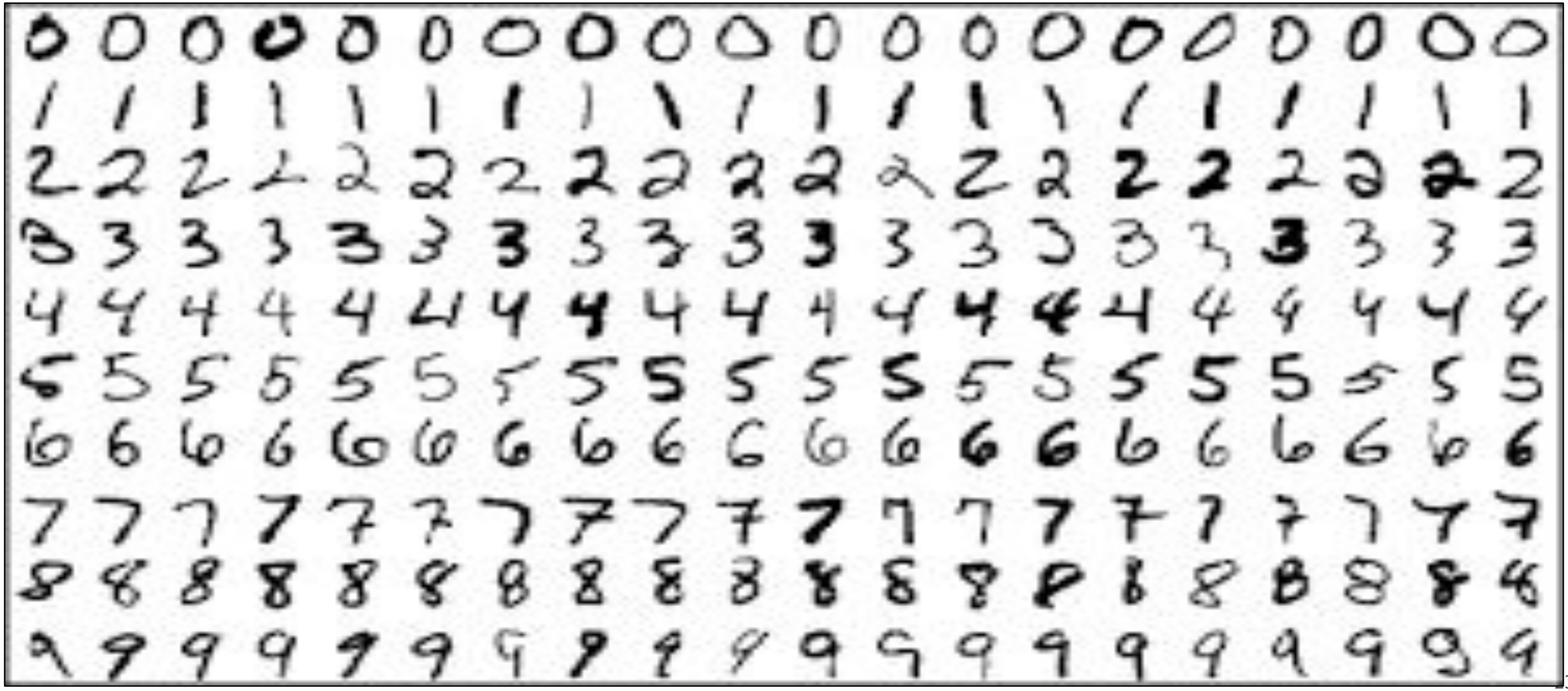
Training a network with data

Lab



HANDWRITTEN DIGIT RECOGNITION

HELLO WORLD of machine learning



WHAT THIS LAB IS

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

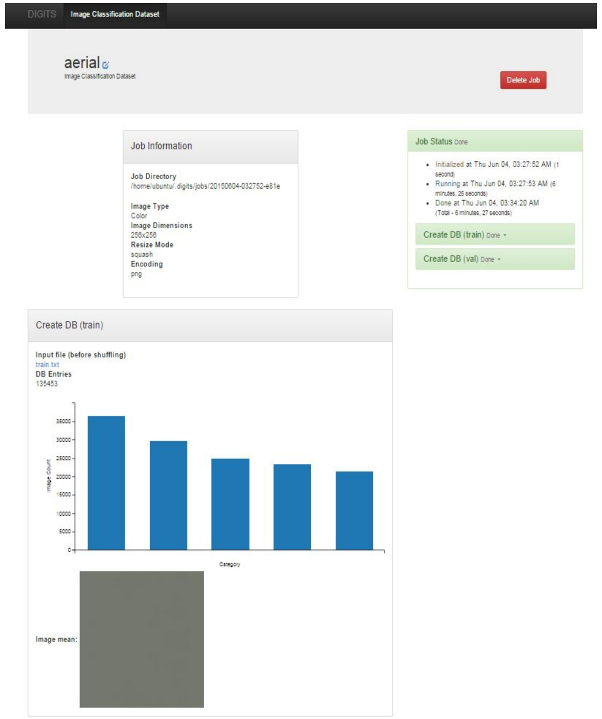
- Hands-on exercises using DIGITS for computer vision and classification

NVIDIA’S DIGITS

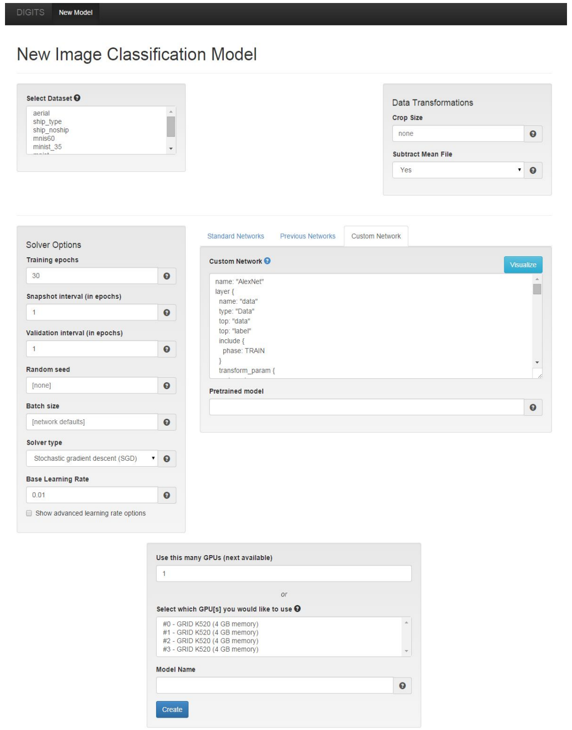
NVIDIA DIGITS

Interactive Deep Learning GPU Training System

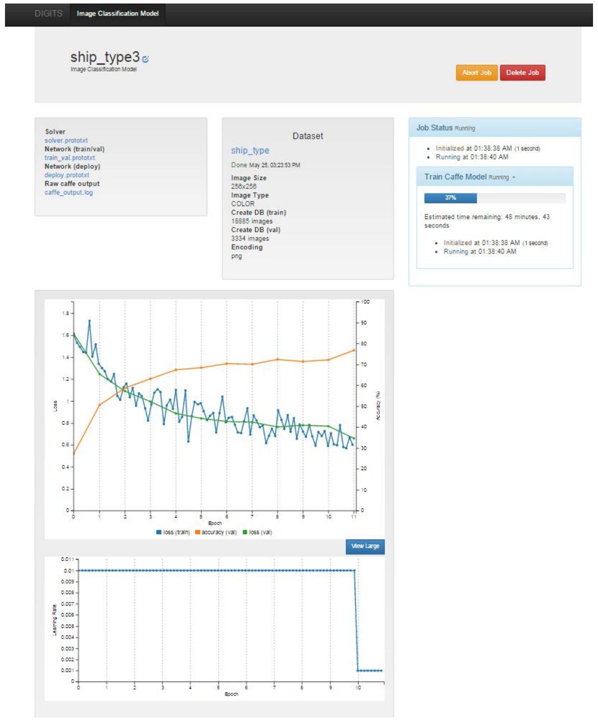
Process Data



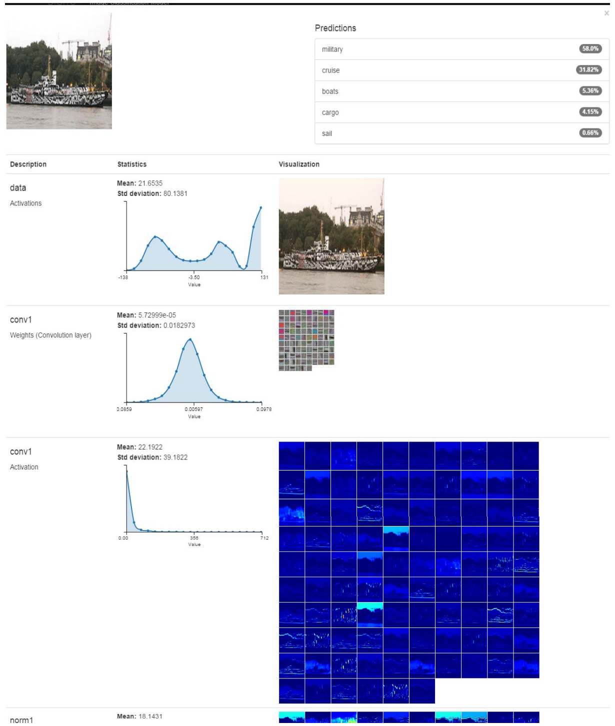
Configure DNN



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

ASSUMPTIONS

- No background in Deep Learning needed
- Understand how to:
  - Navigate a web browser
  - Download files
  - Locate files in file managers

LAB OVERVIEW

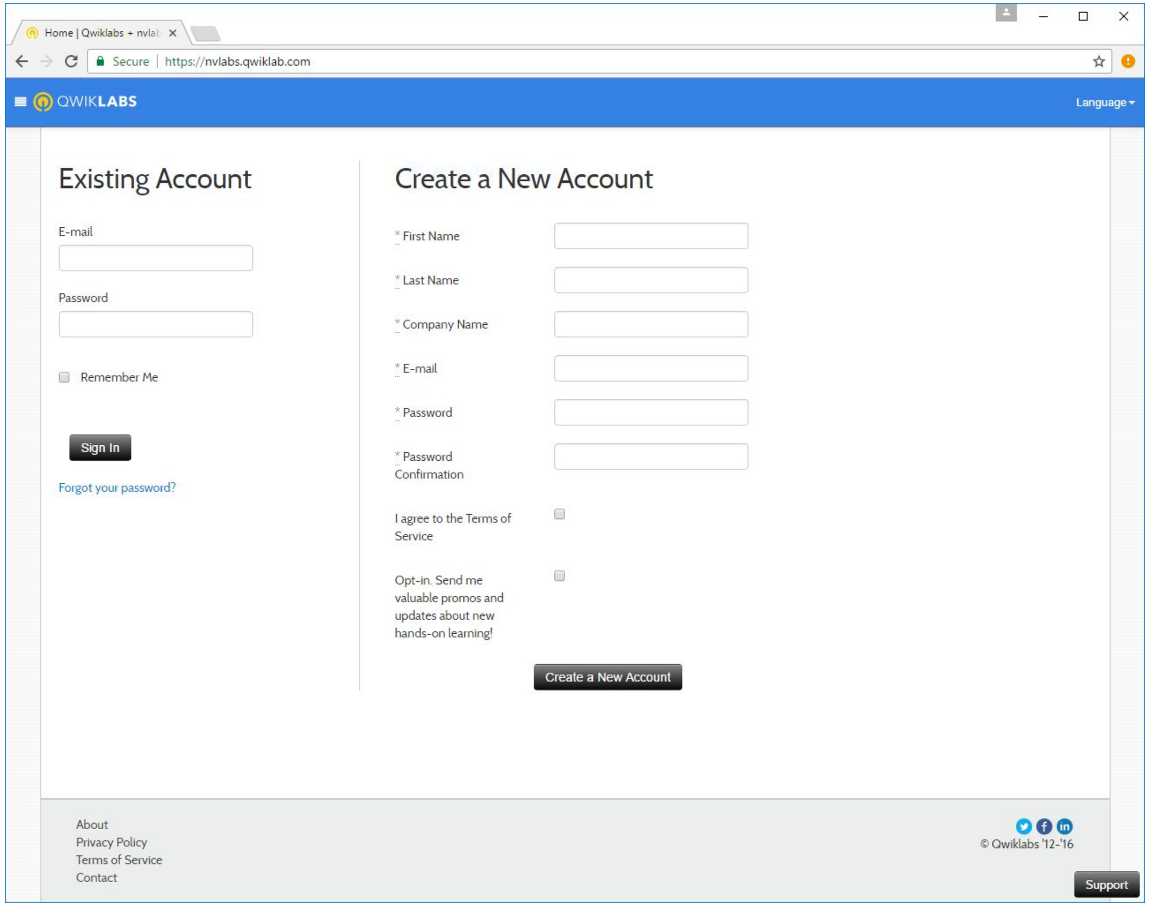
- Learn about the workflow of Deep Learning
  - Load data
  - Expose a network to data
  - Evaluate model results
  - Try different techniques to improve initial results

LAUNCHING THE LAB

NAVIGATING TO QWIKLABS

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

2. Login or create a new account

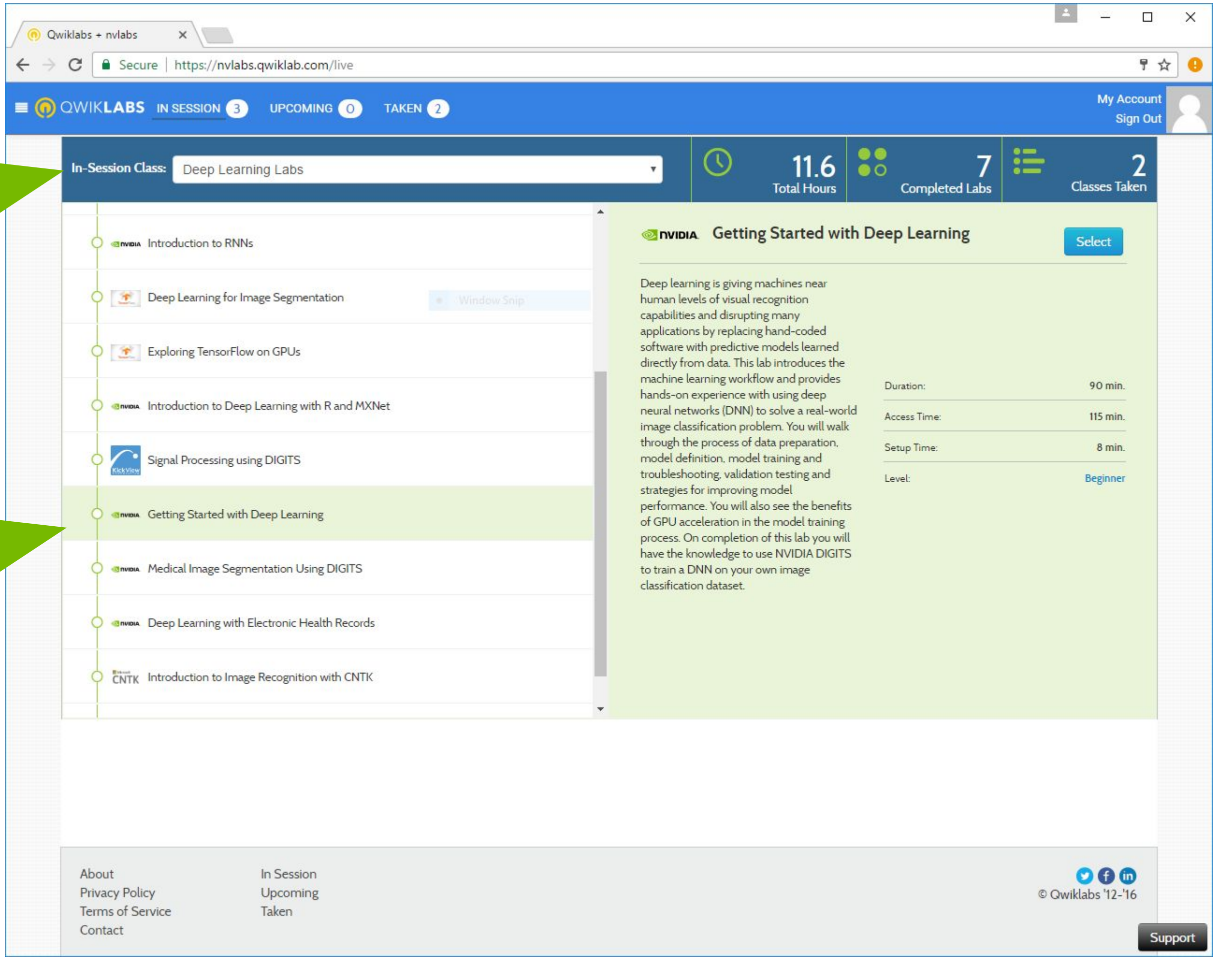


NVIDIA DEEP LEARNING INSTITUTE

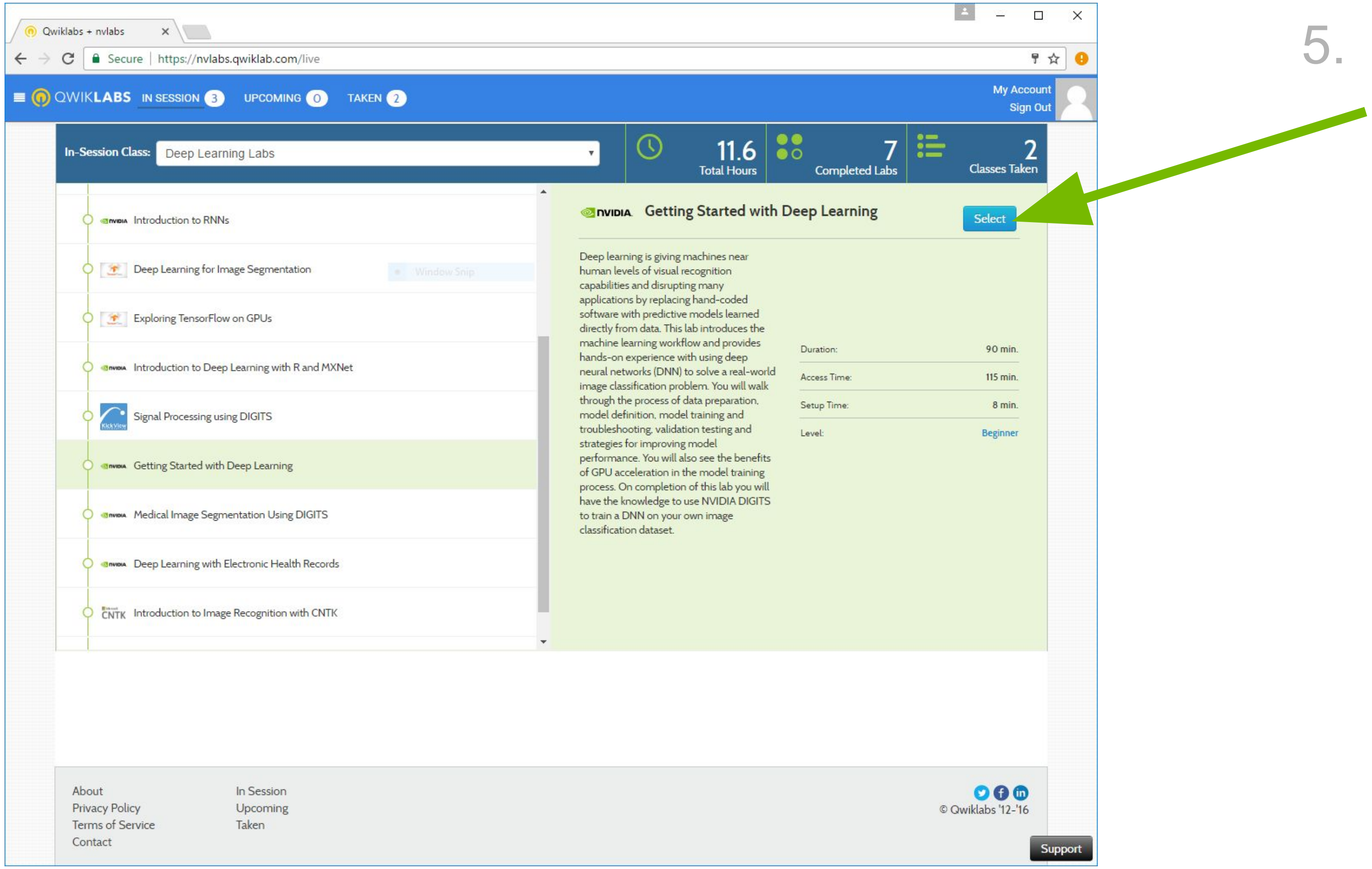
ACCESSING LAB ENVIRONMENT

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

4. Click the "Image Classification with DIGITS" Class from the list



LAUNCHING THE LAB ENVIRONMENT

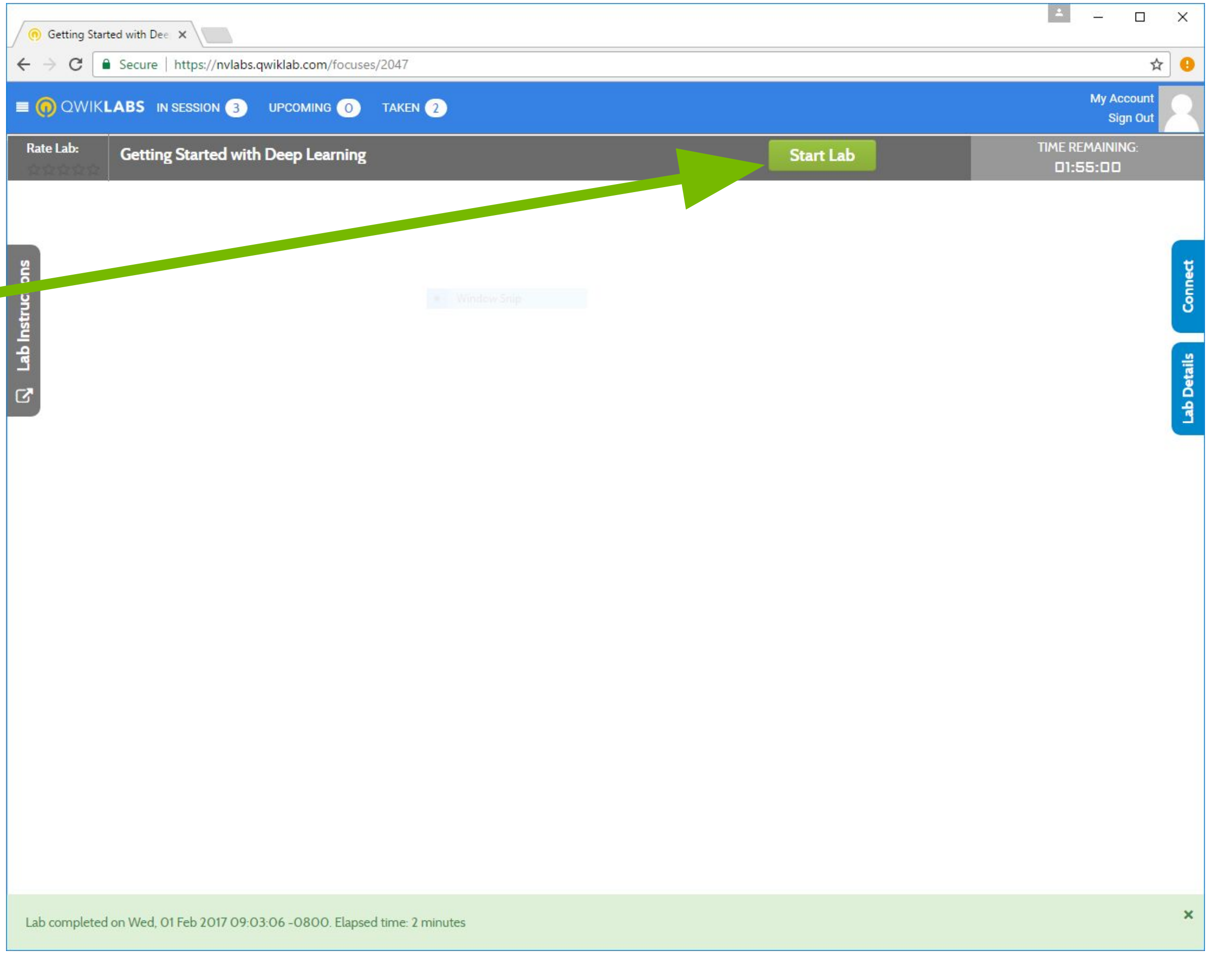


5. Click on the Select button to launch the lab environment

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

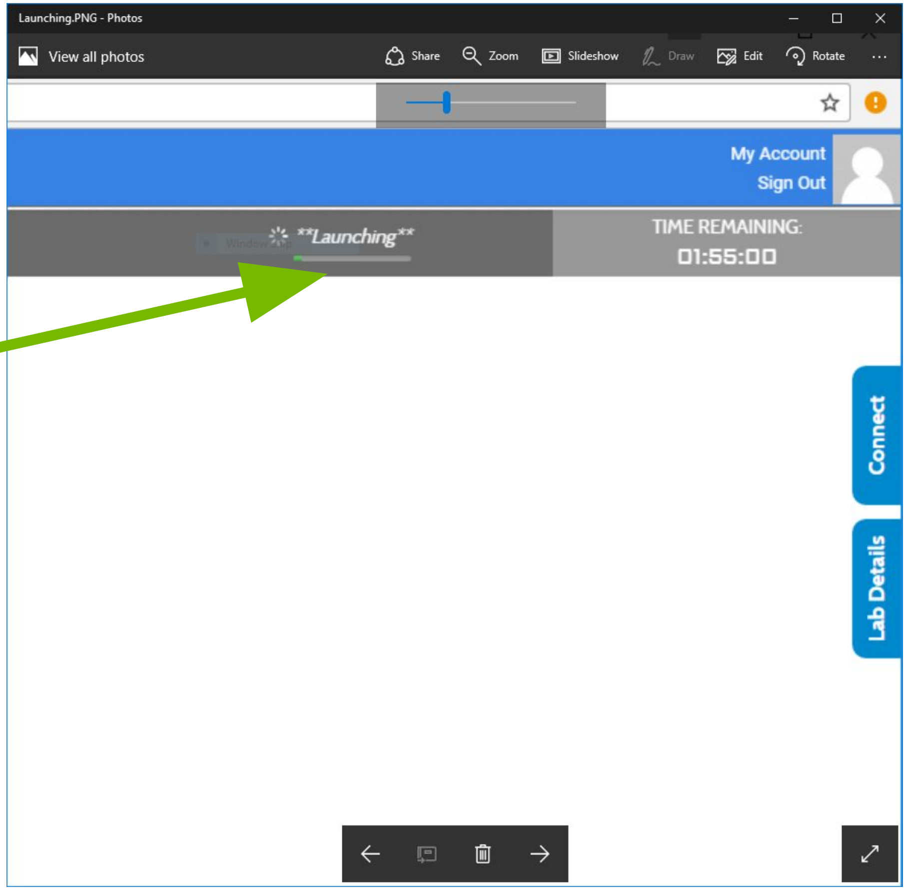
LAUNCHING THE LAB ENVIRONMENT

6. Click on the Start Lab button



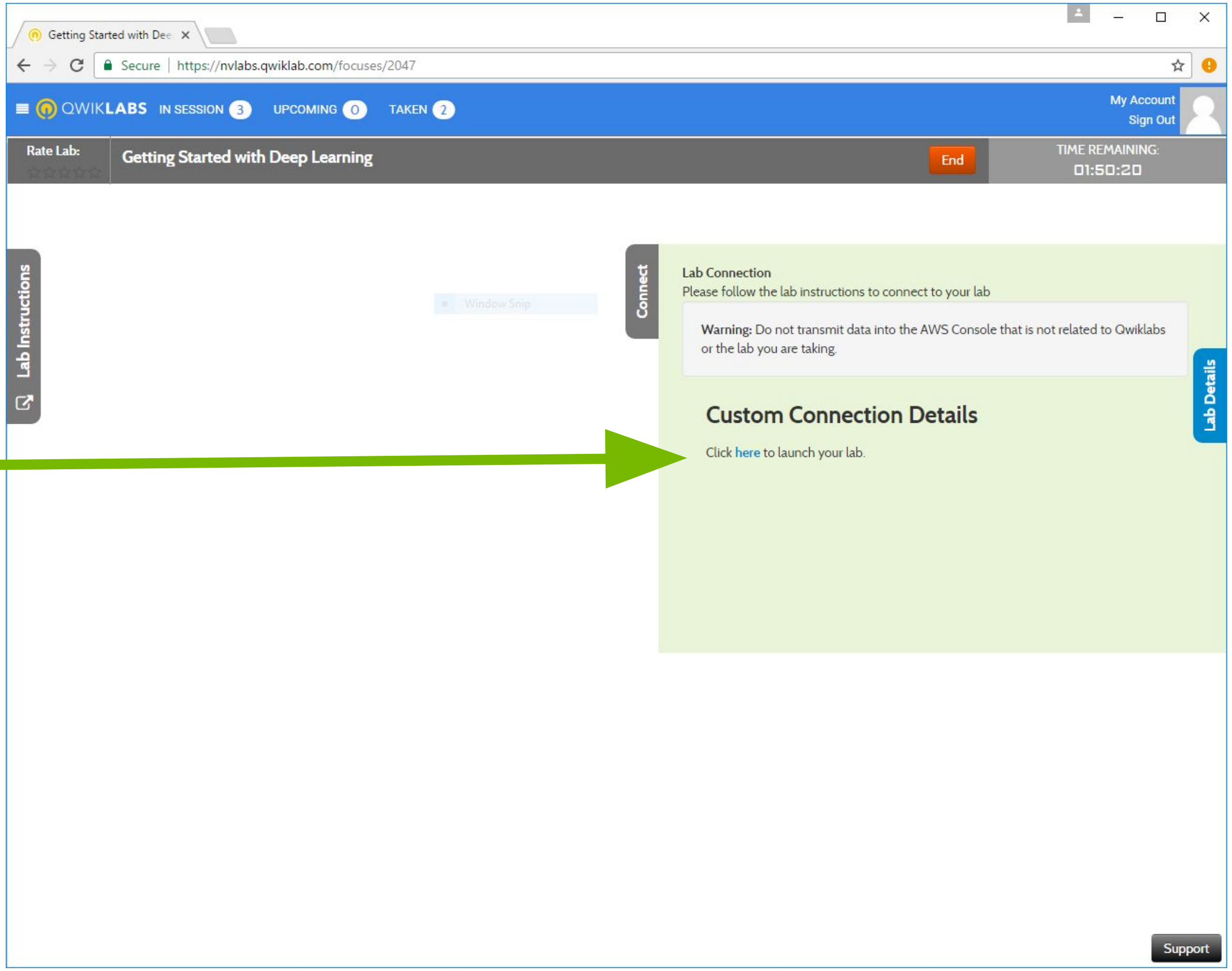
LAUNCHING THE LAB ENVIRONMENT

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



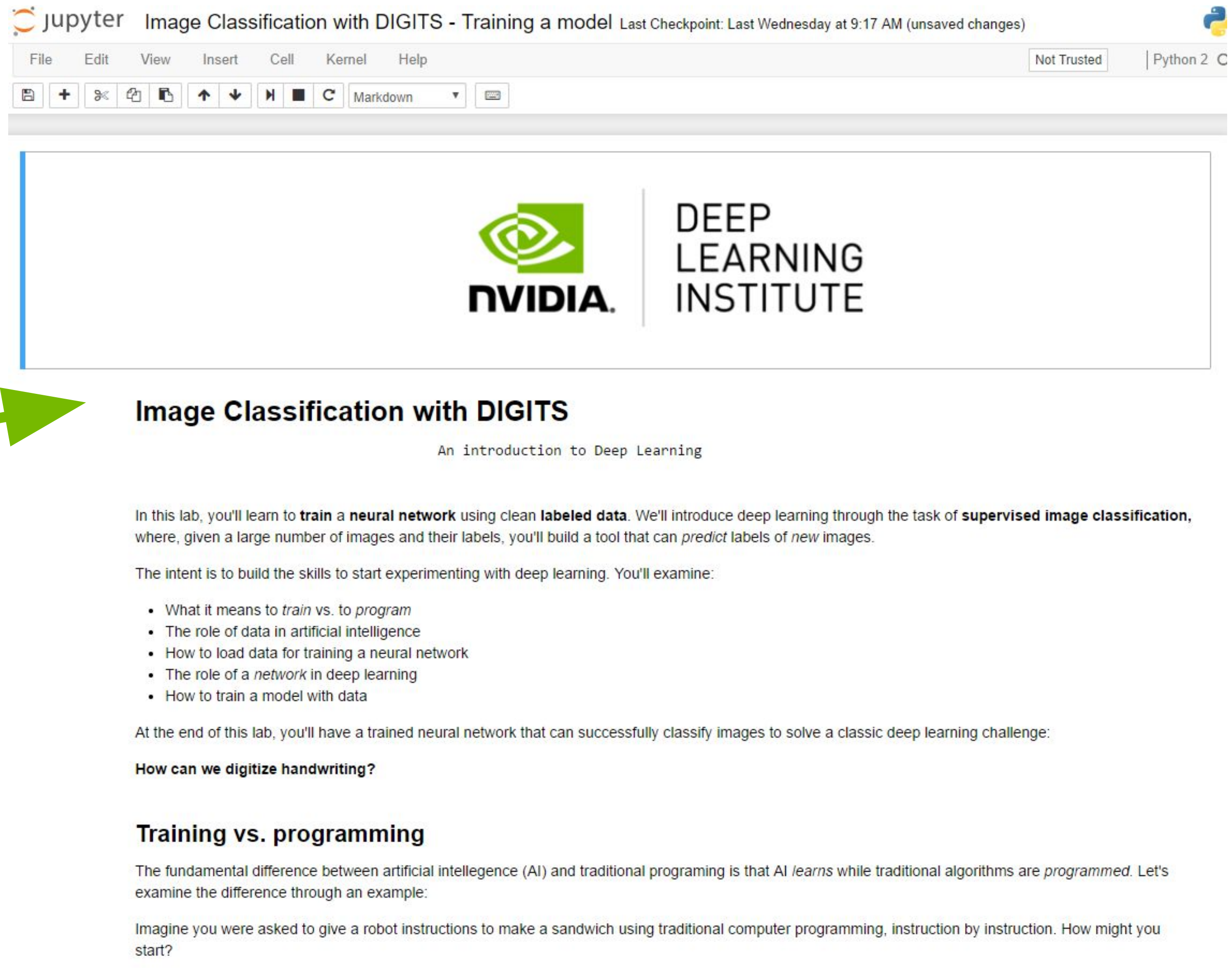
CONNECTING TO THE LAB ENVIRONMENT

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



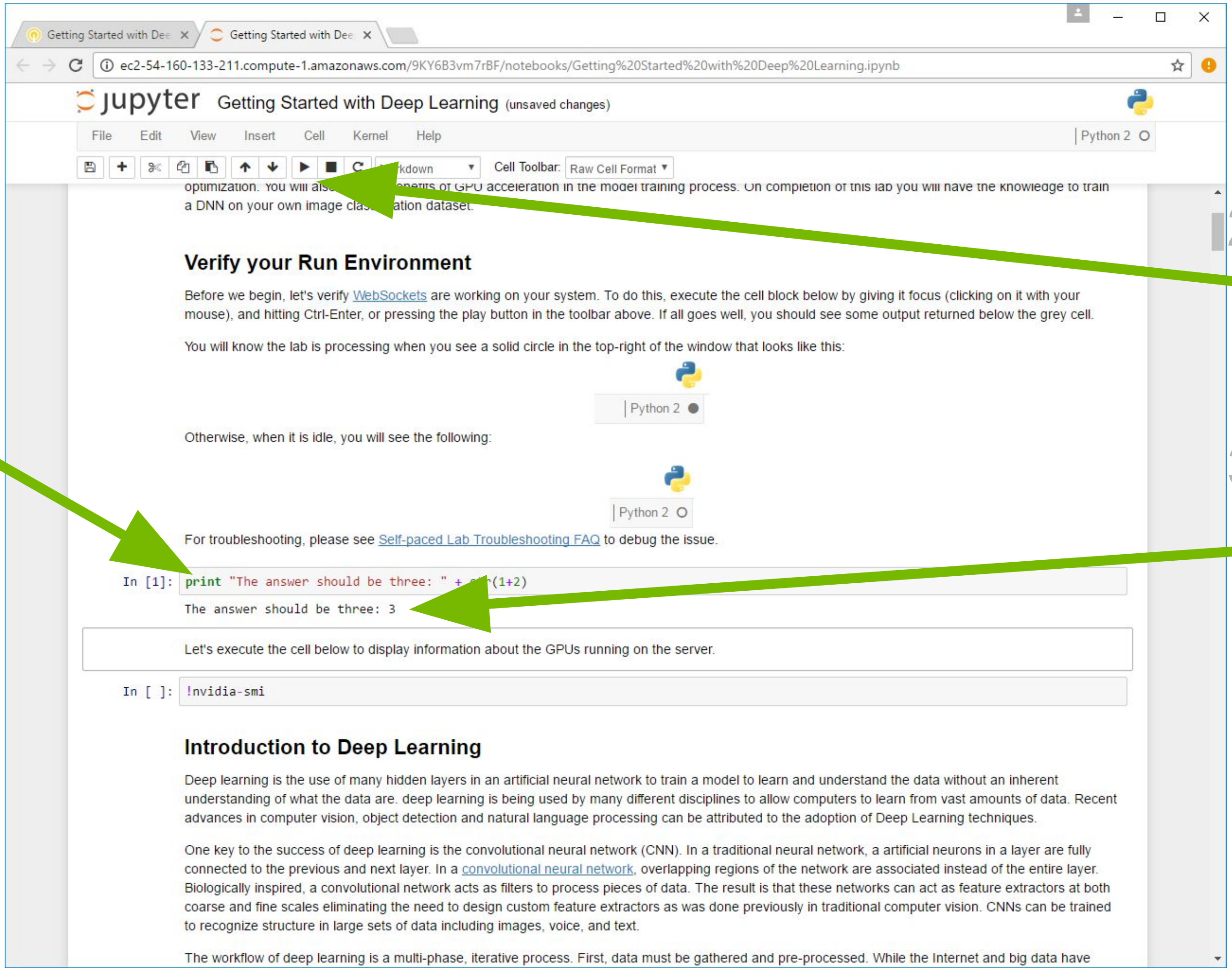
CONNECTING TO THE LAB ENVIRONMENT

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



JUPYTER NOTEBOOK

1. Place your cursor in the code

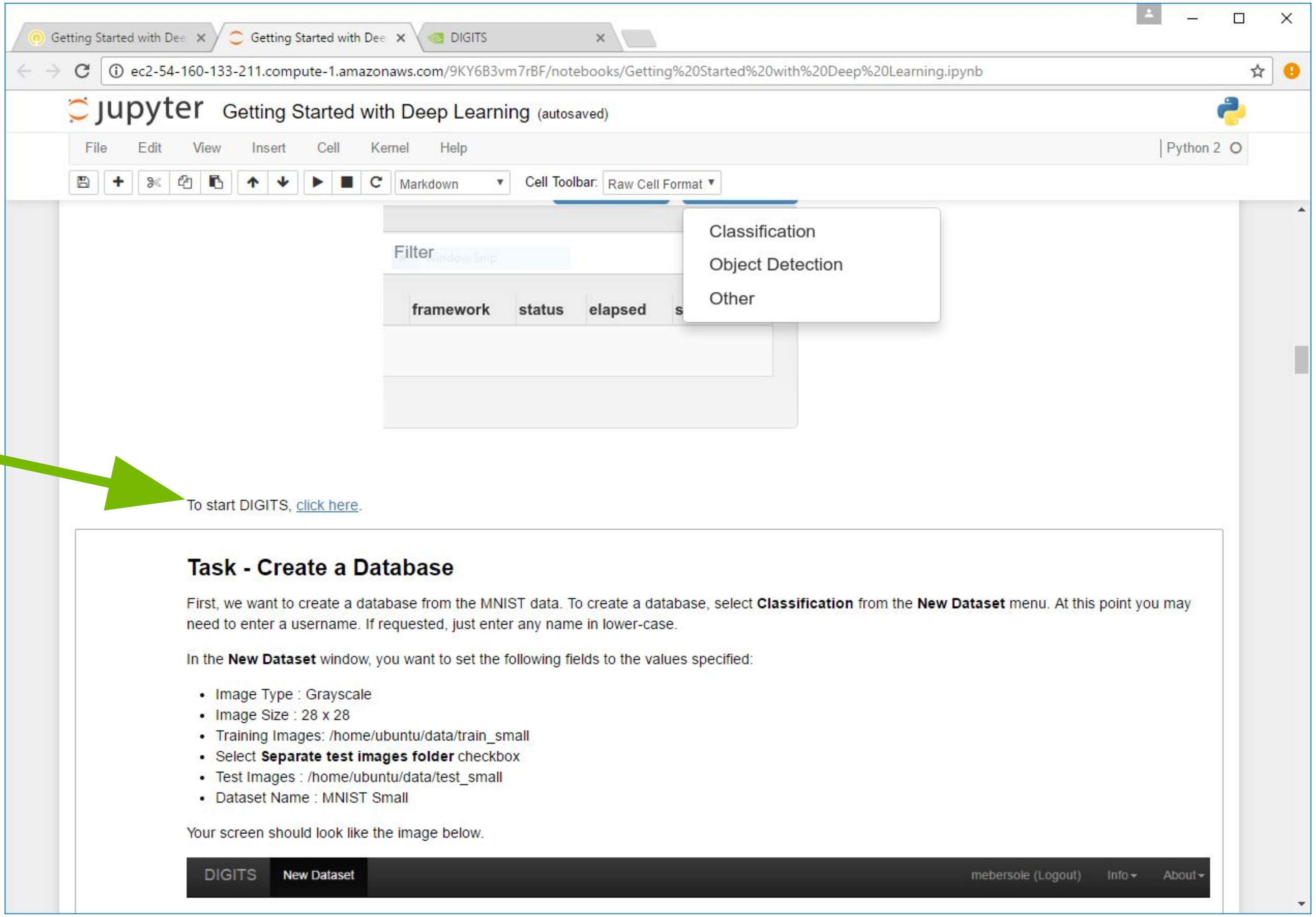


2. Click the “run cell” button

3. Confirm you receive the same result

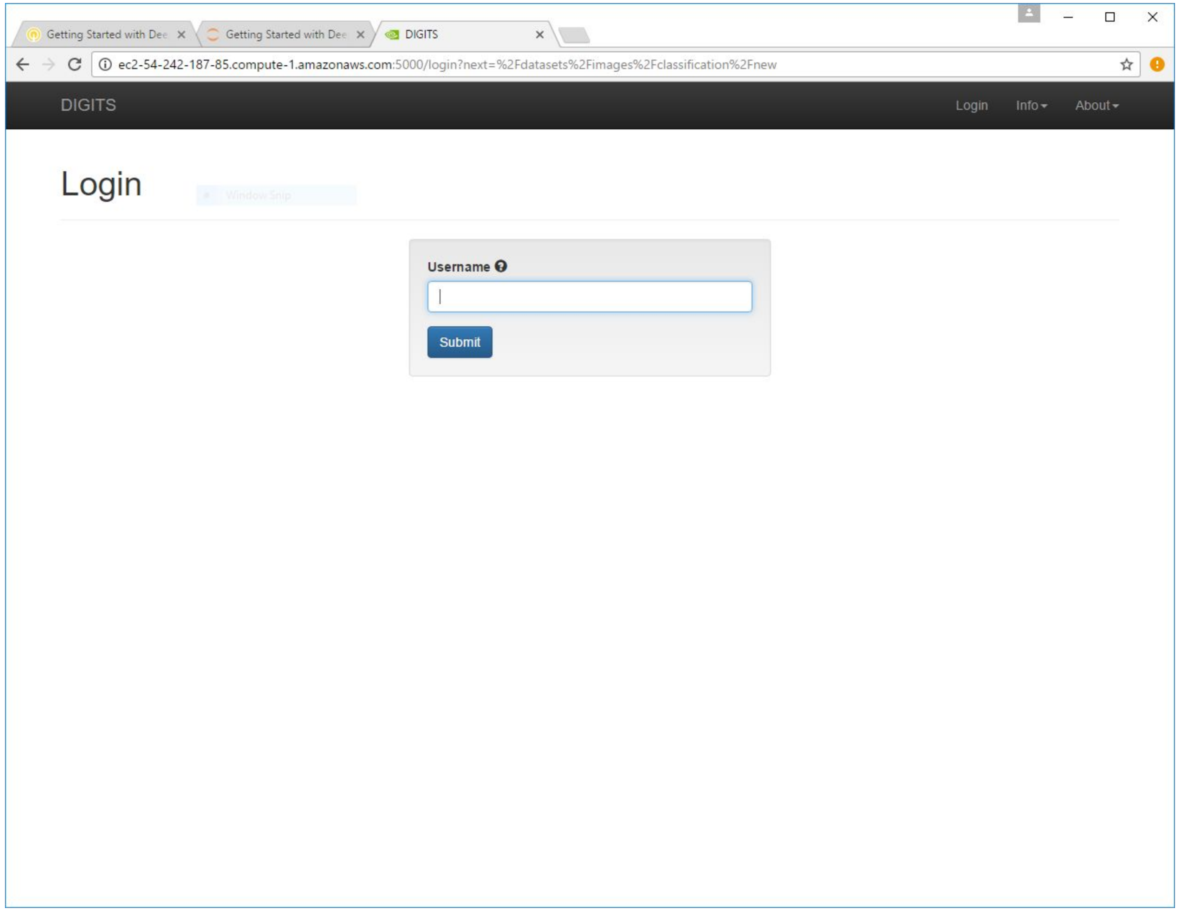
STARTING DIGITS

Instruction in Jupyter notebook will link you to DIGITS



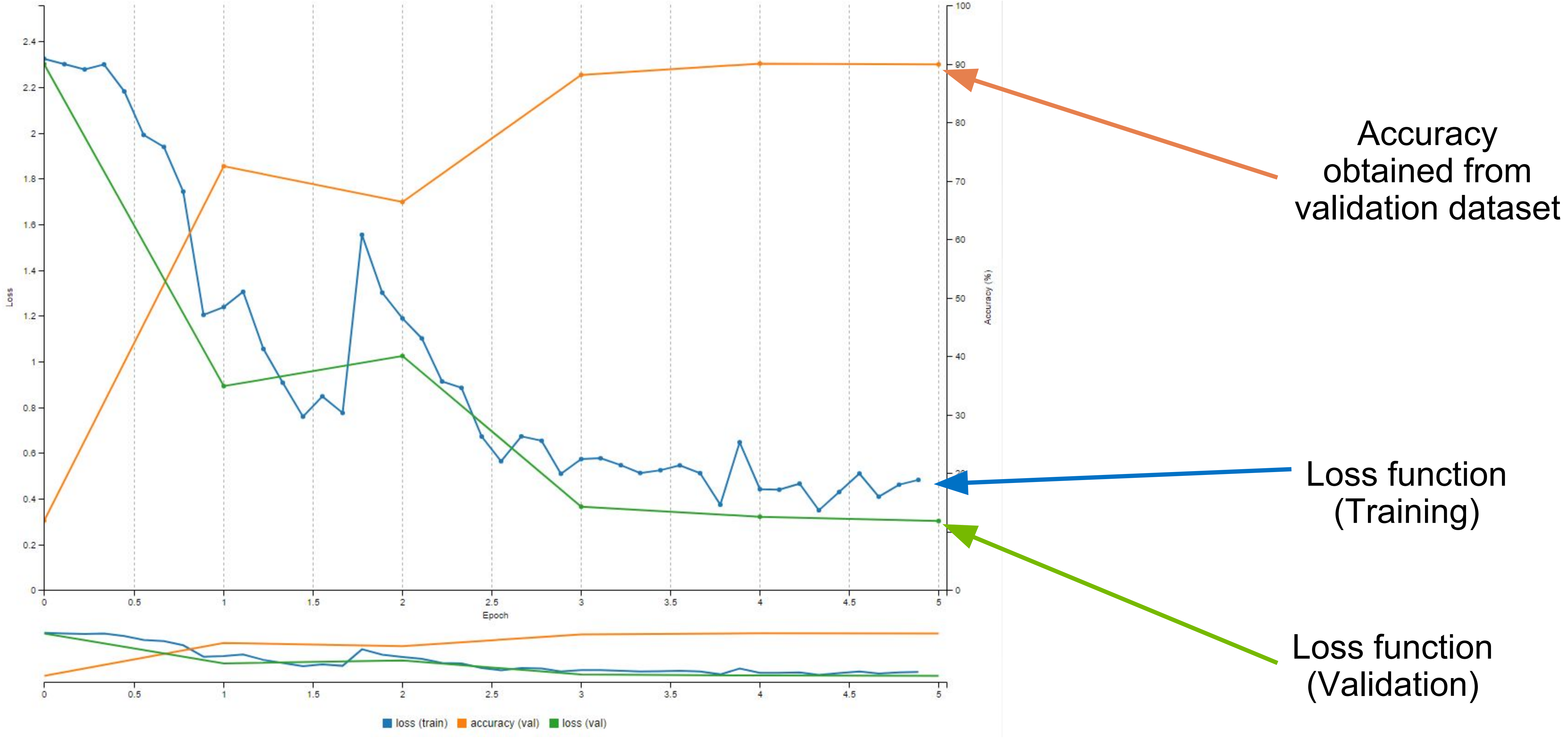
ACCESSING DIGITS

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

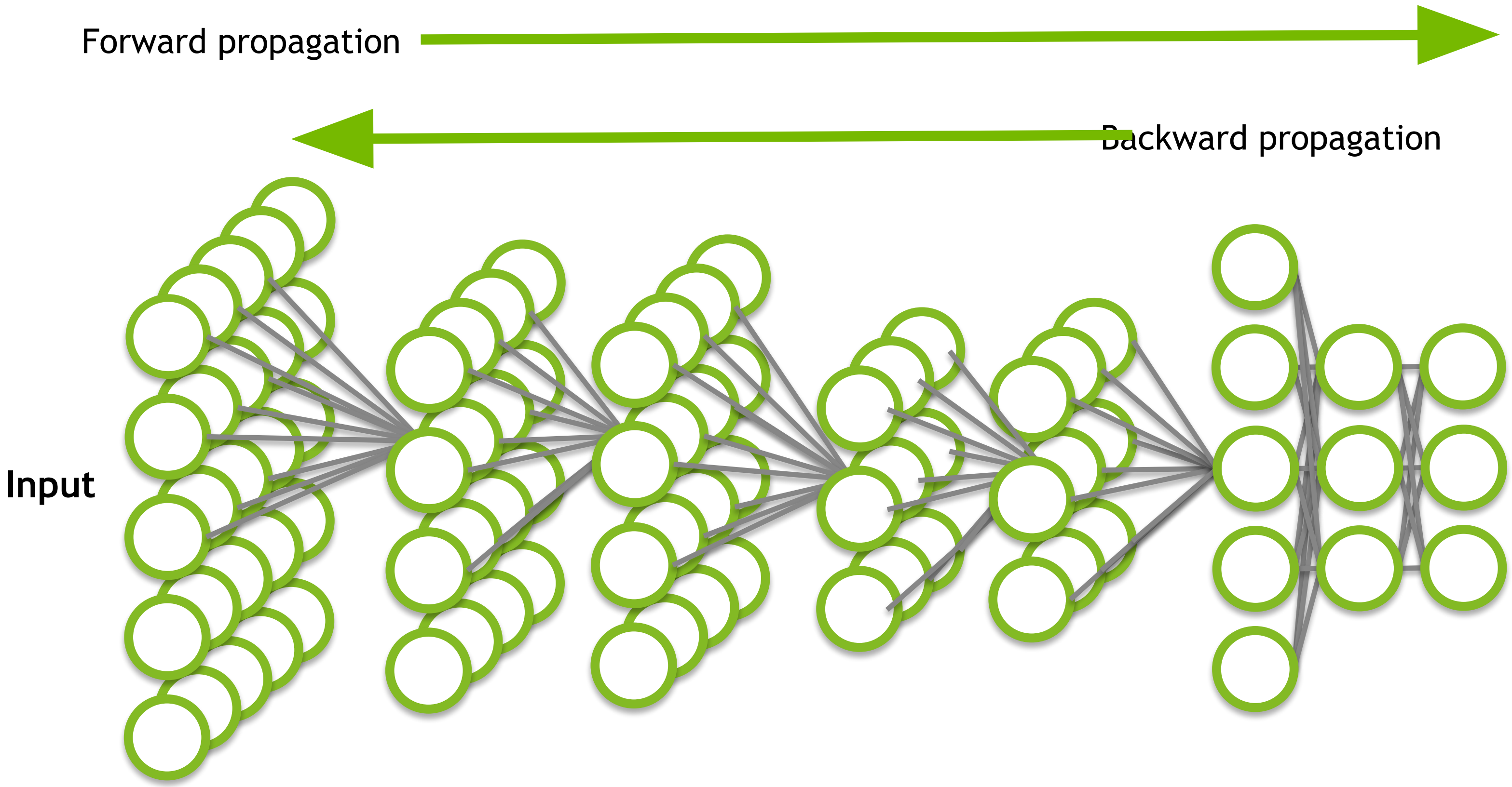


Evaluating Performance

EVALUATE THE MODEL



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?

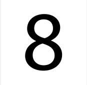
- Increase accuracy and confidence with similar data
- Generalize performance to more diverse data

Lab Review

More data

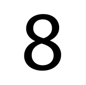
Full dataset (10 epochs)

- 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 %

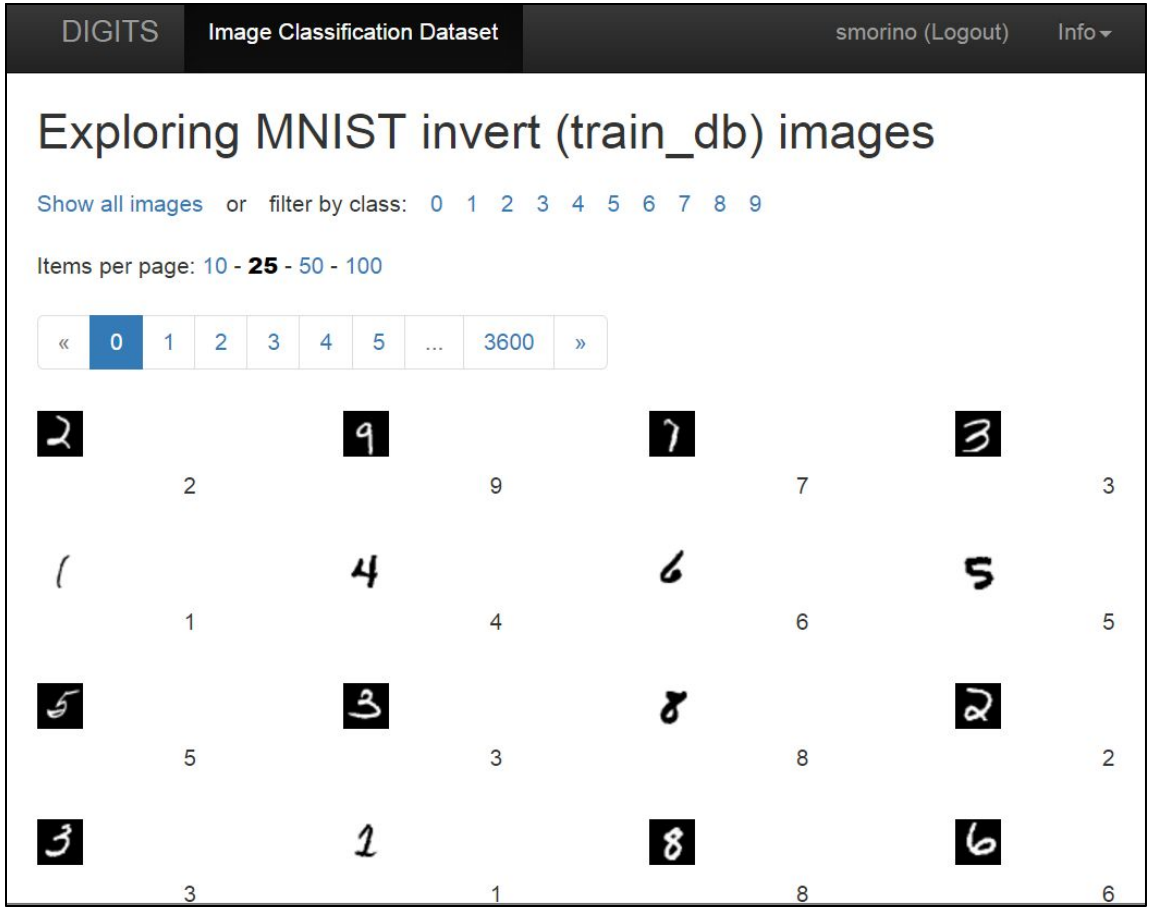
DATA AUGMENTATION

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“
```

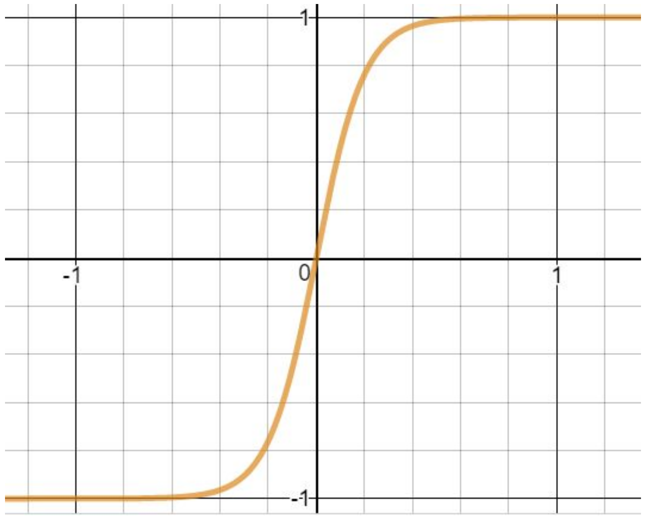
```
layer {
  name: "conv1"
  type: "Convolution"
    ...
    convolution_param {
    num_output: 75
    ...
layer {
    name: "conv2"
    type: "Convolution"
    ...
    convolution_param {
    num_output: 100
    ...
```

Next Steps

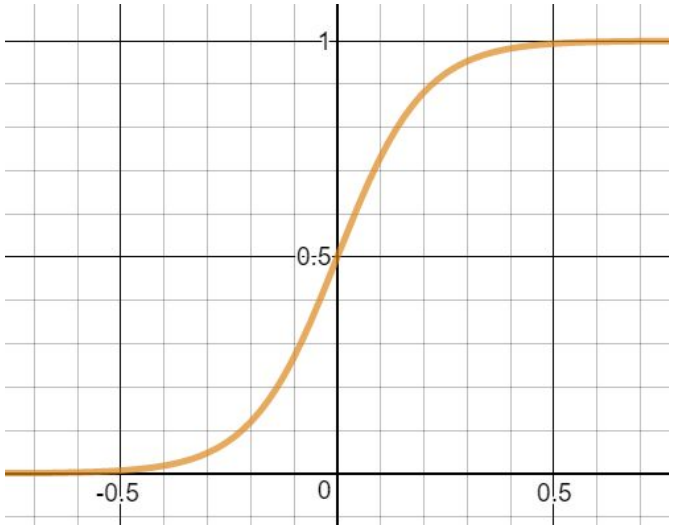
- Experiment with Image Classification
  - Different datasets
  - Increase performance
- Learn to train existing networks with data for other challenges
- Learn about network construction
- Learn about how to create an image classifier with other frameworks
  - Caffe/Keras
  - Tensorflow
  - Etc.

Appendix

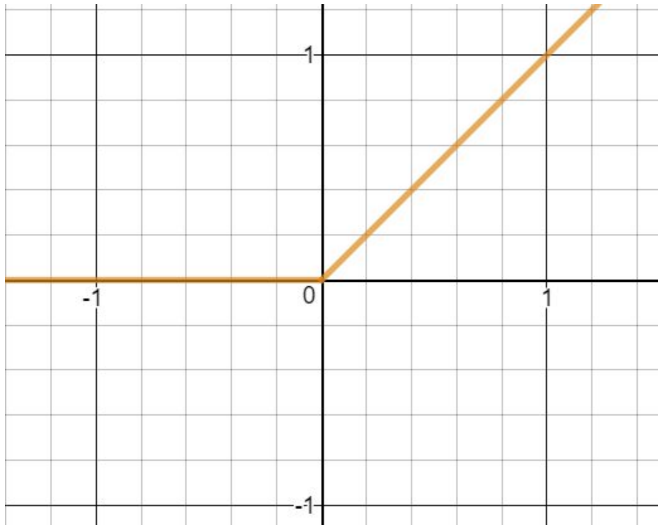
Activation functions



tanh



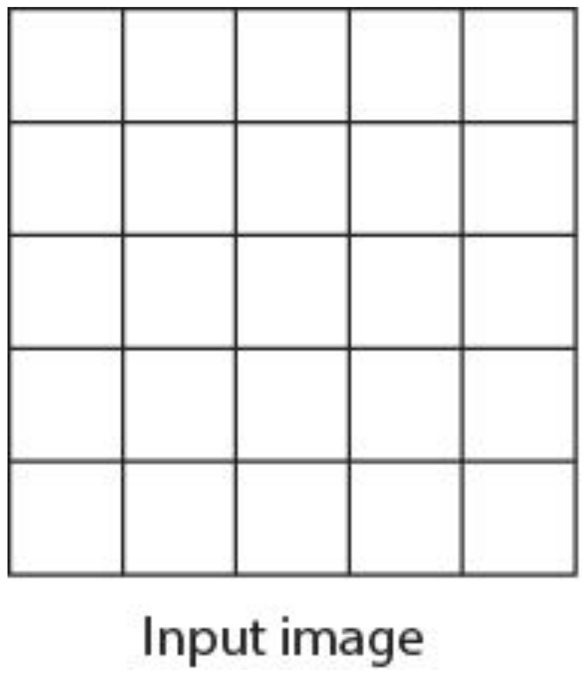
Sigmoid

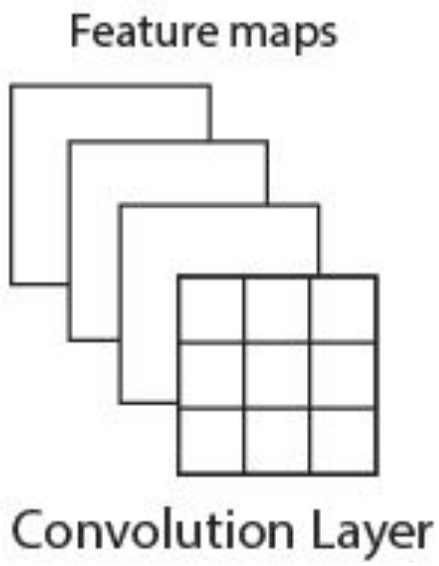


ReLU

CNN - Example

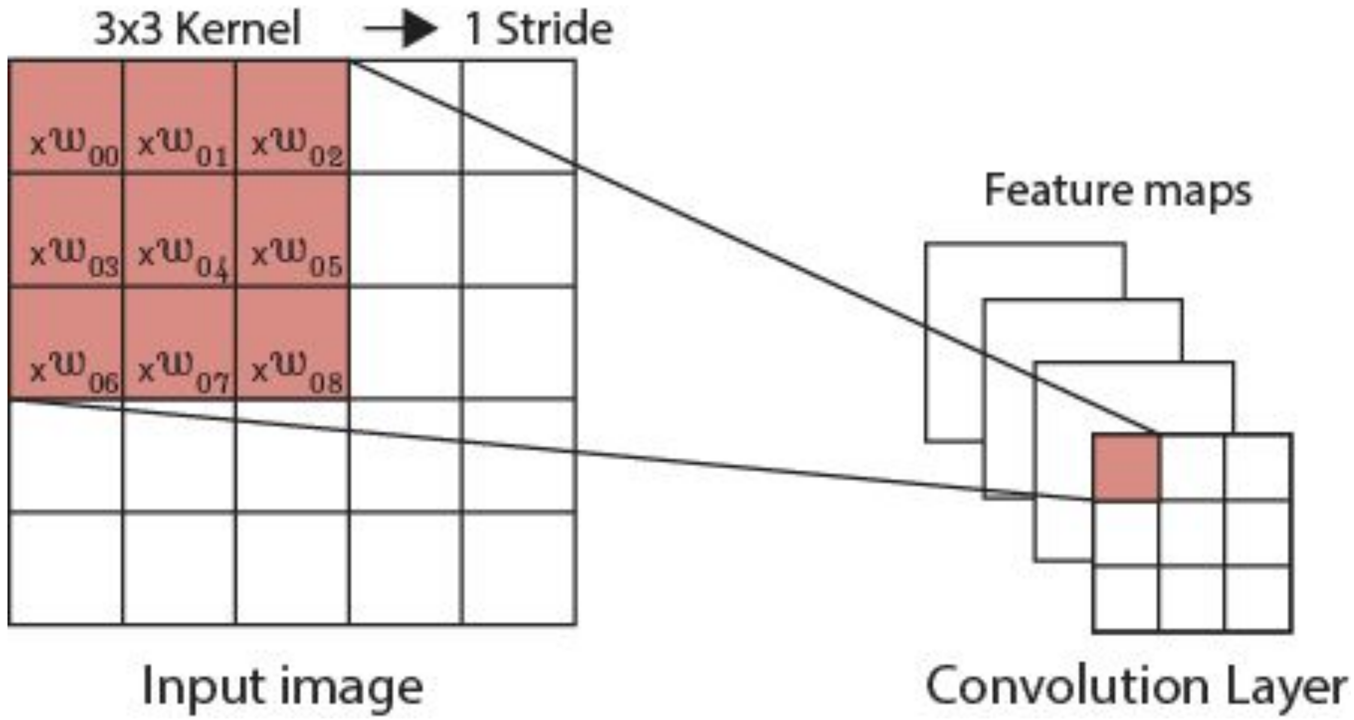
Each pixel is a neuron





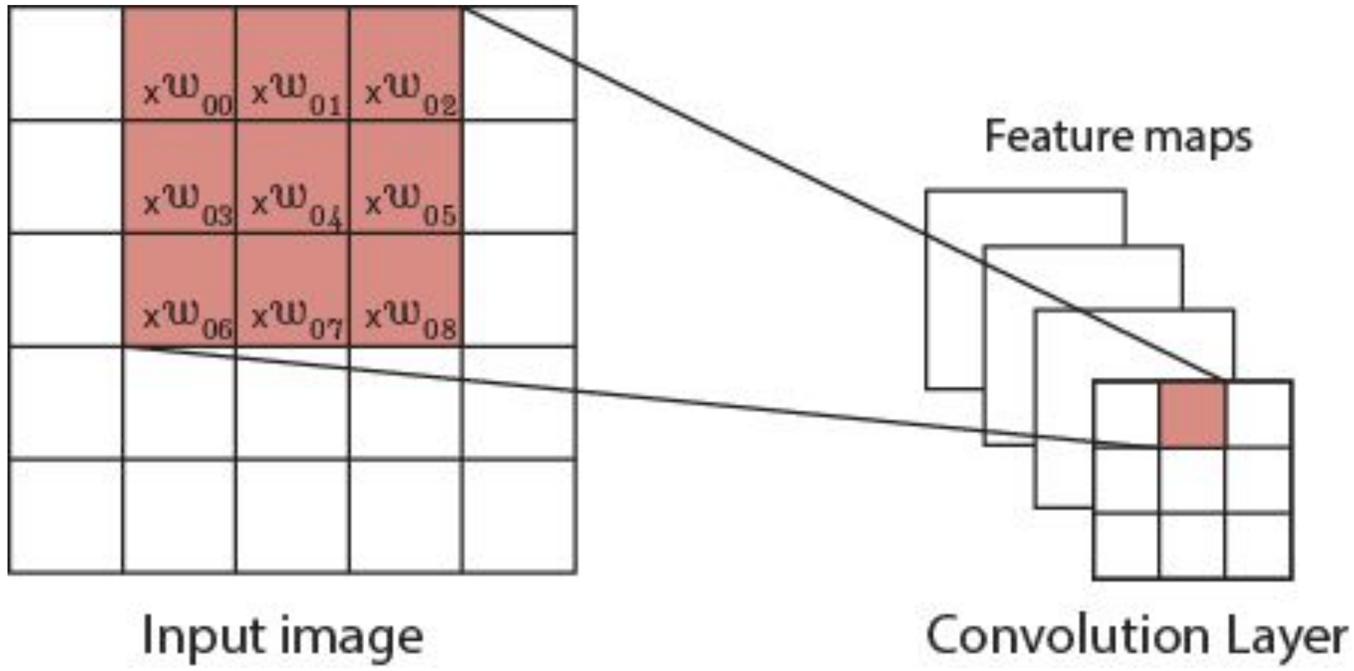
CNN - Example - 1st Feature Map

3x3 Kernel, 1 Stride, weights constant per kernel



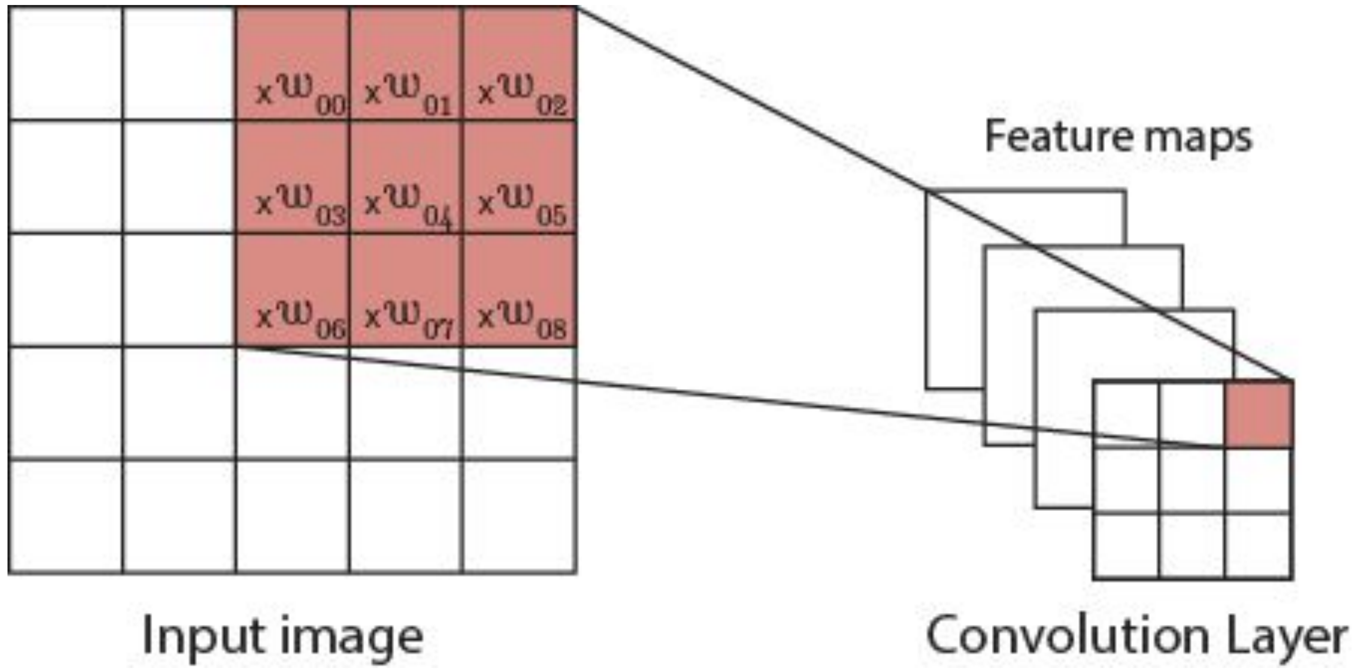
CNN - Example - 1st Feature Map

3x3 Kernel, 1 Stride, weights constant per kernel



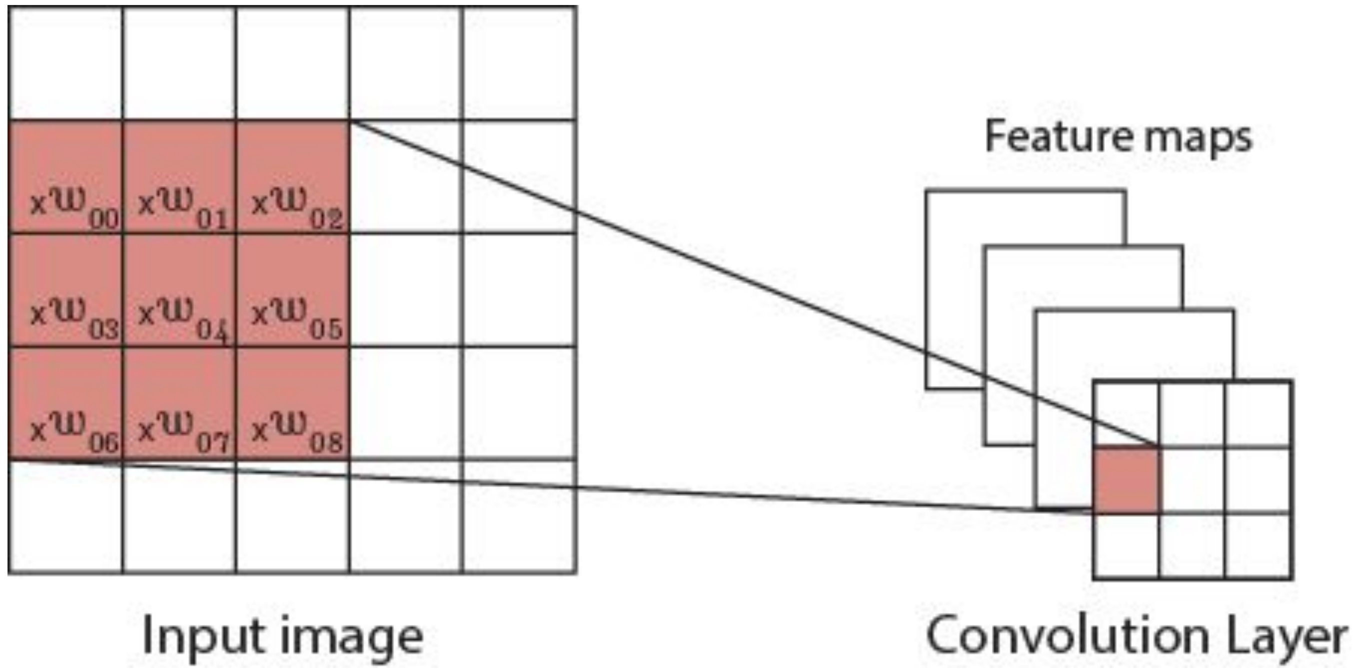
CNN - Example - 1st Feature Map

3x3 Kernel, 1 Stride, weights constant per kernel



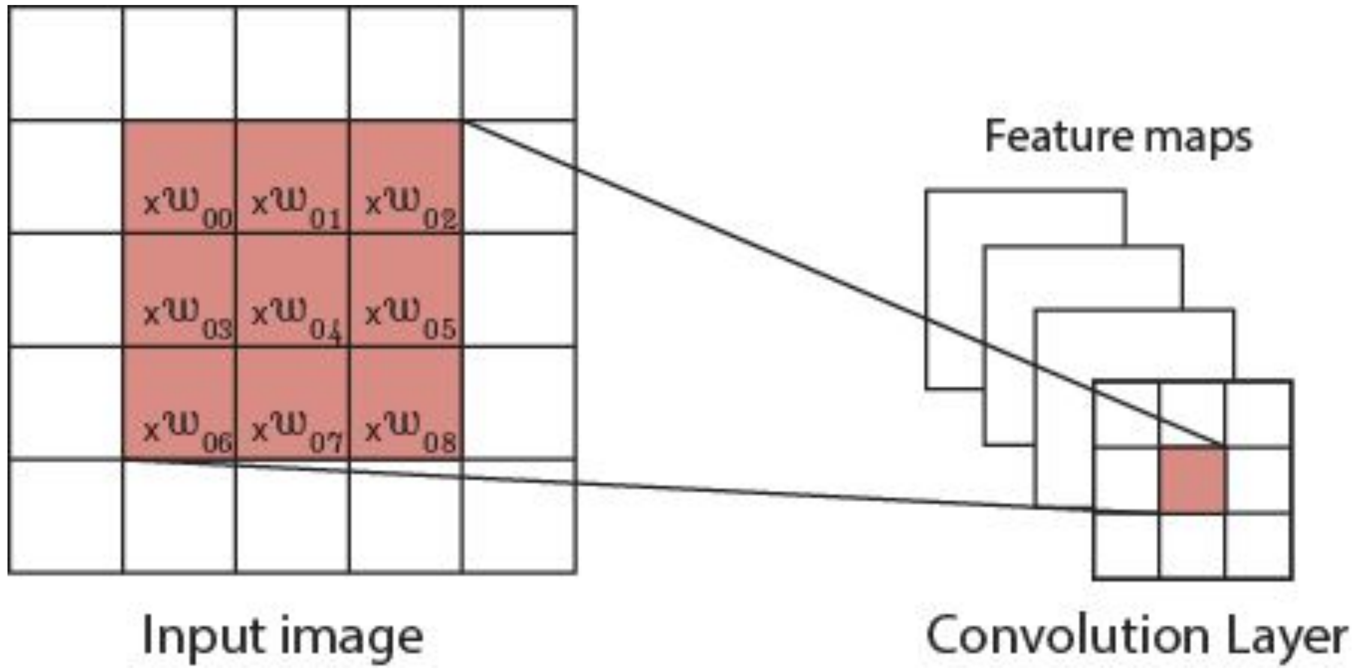
CNN - Example - 1st Feature Map

3x3 Kernel, 1 Stride, weights constant per kernel



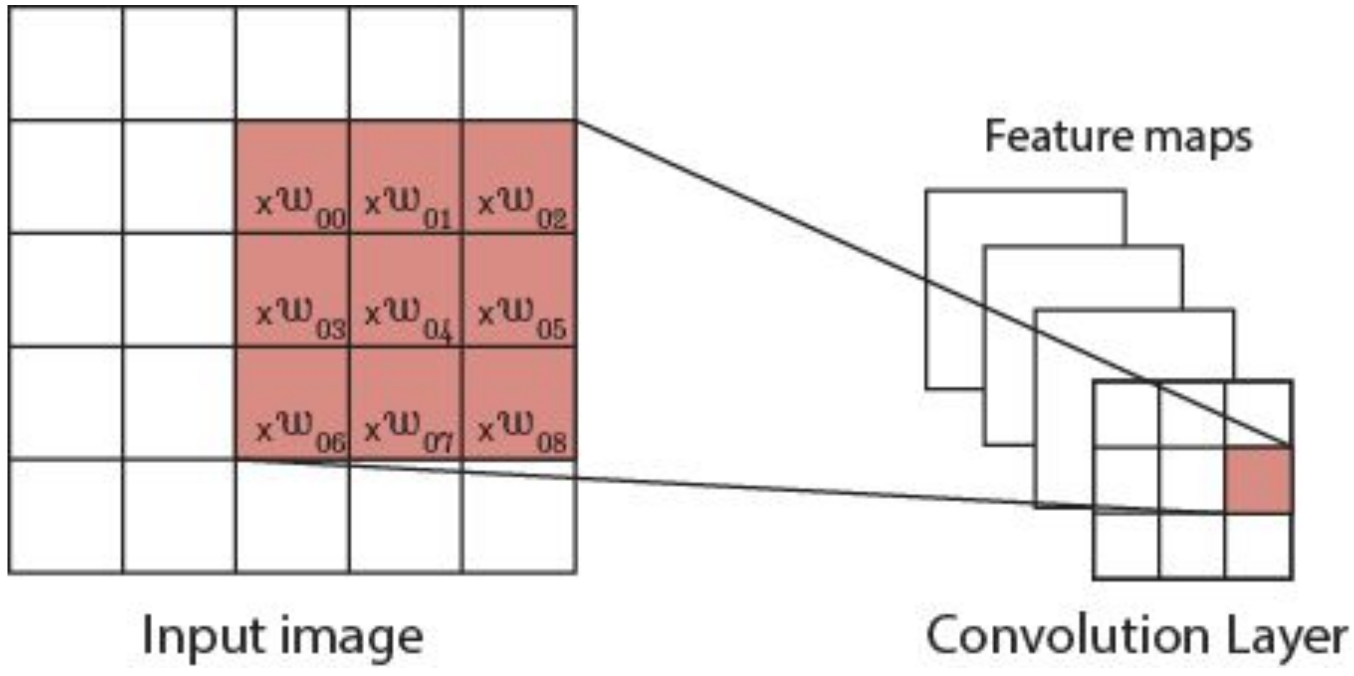
CNN - Example - 1st Feature Map

3x3 Kernel, 1 Stride, weights constant per kernel

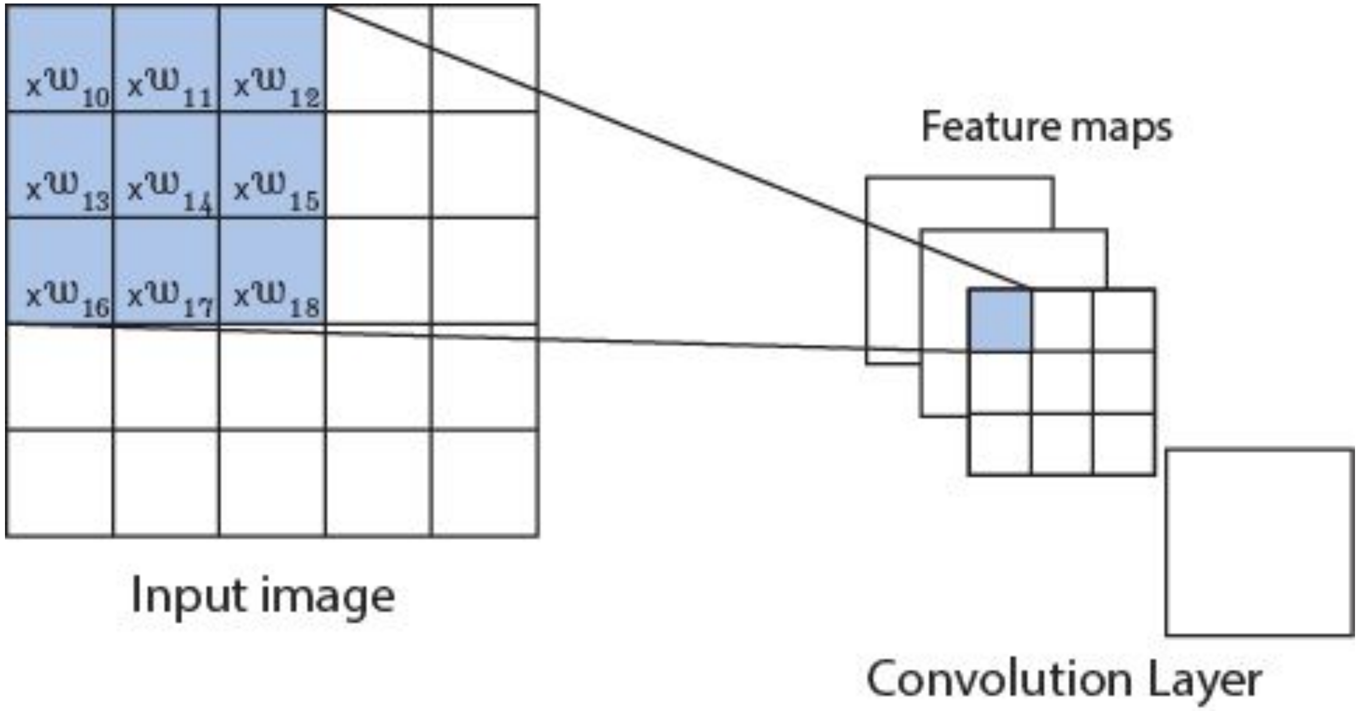


CNN - Example - 1st Feature Map

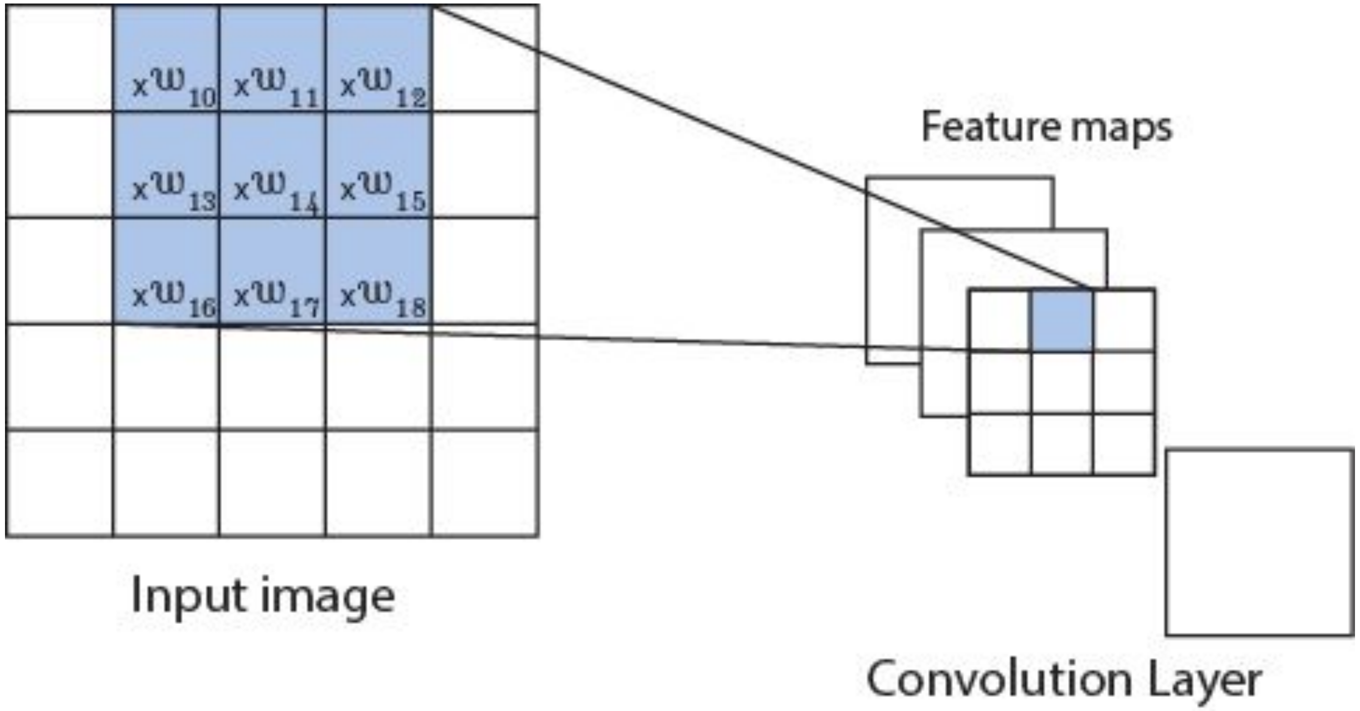
3x3 Kernel, 1 Stride, weights constant per kernel



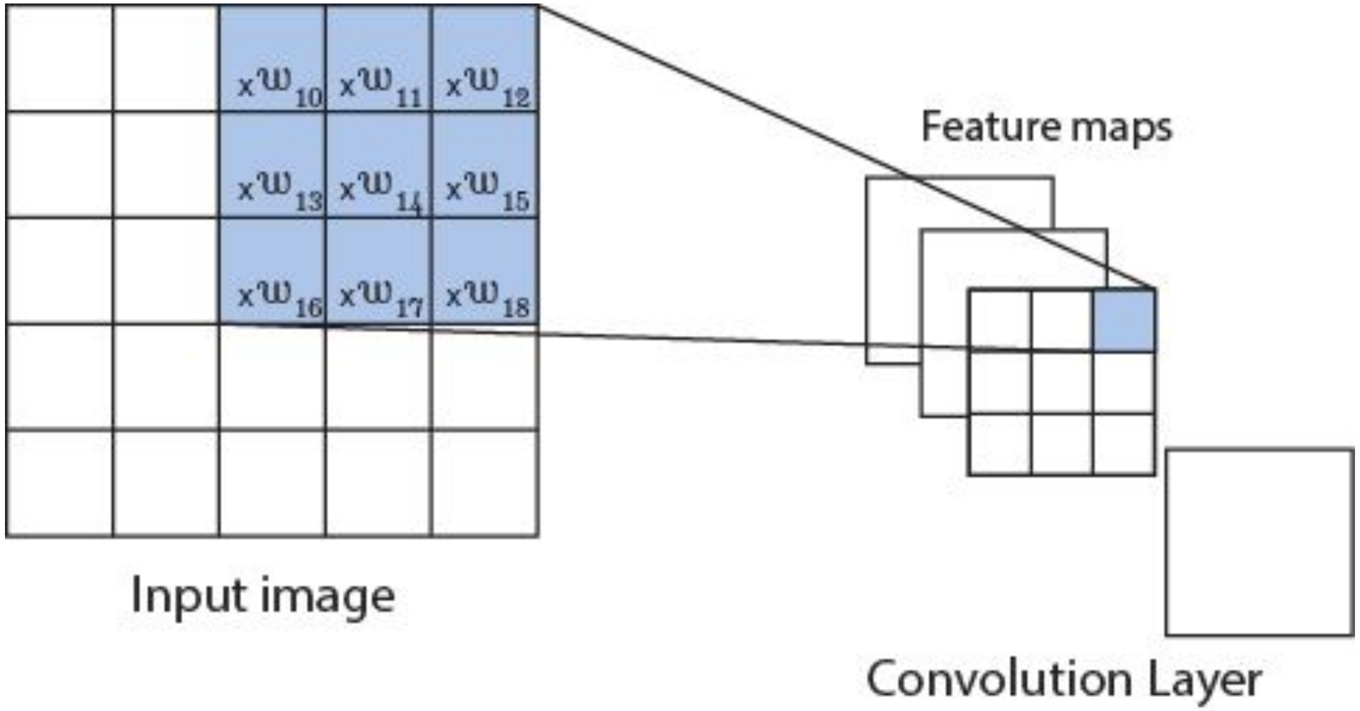
CNN - Example - 2nd Feature Map



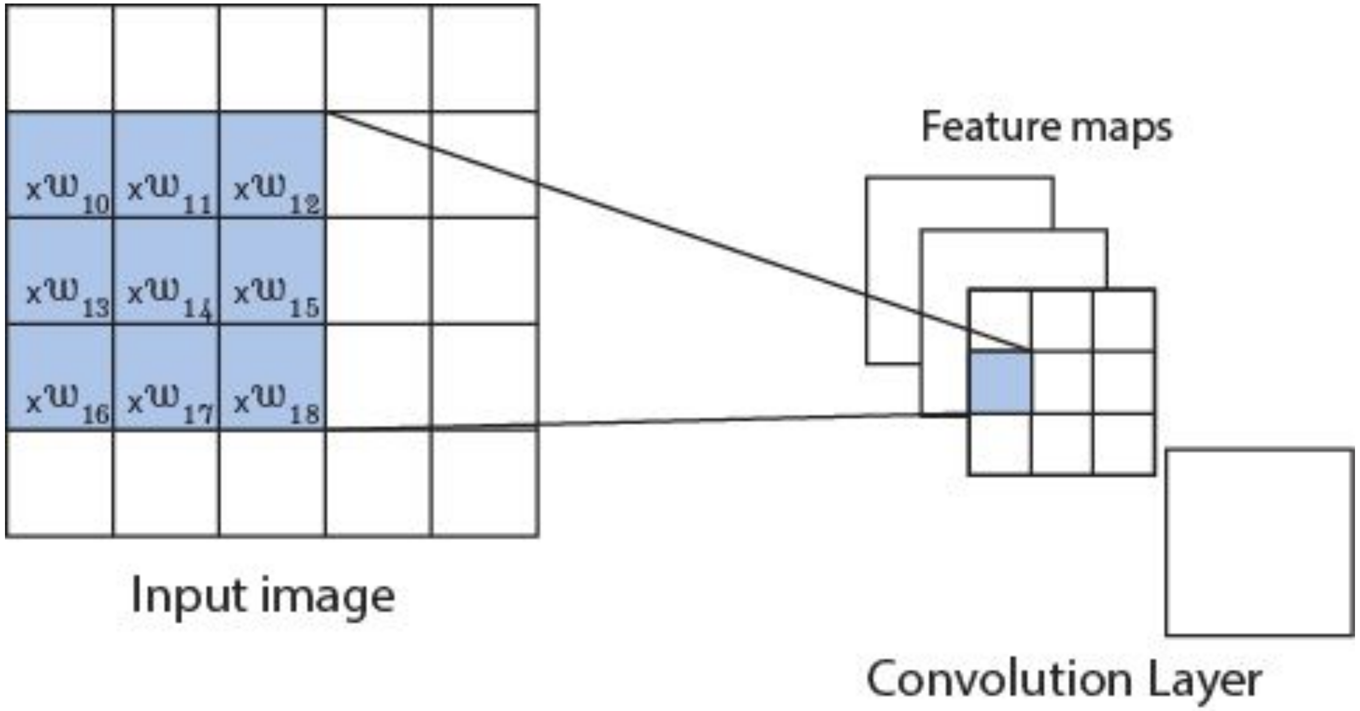
CNN - Example - 2nd Feature Map



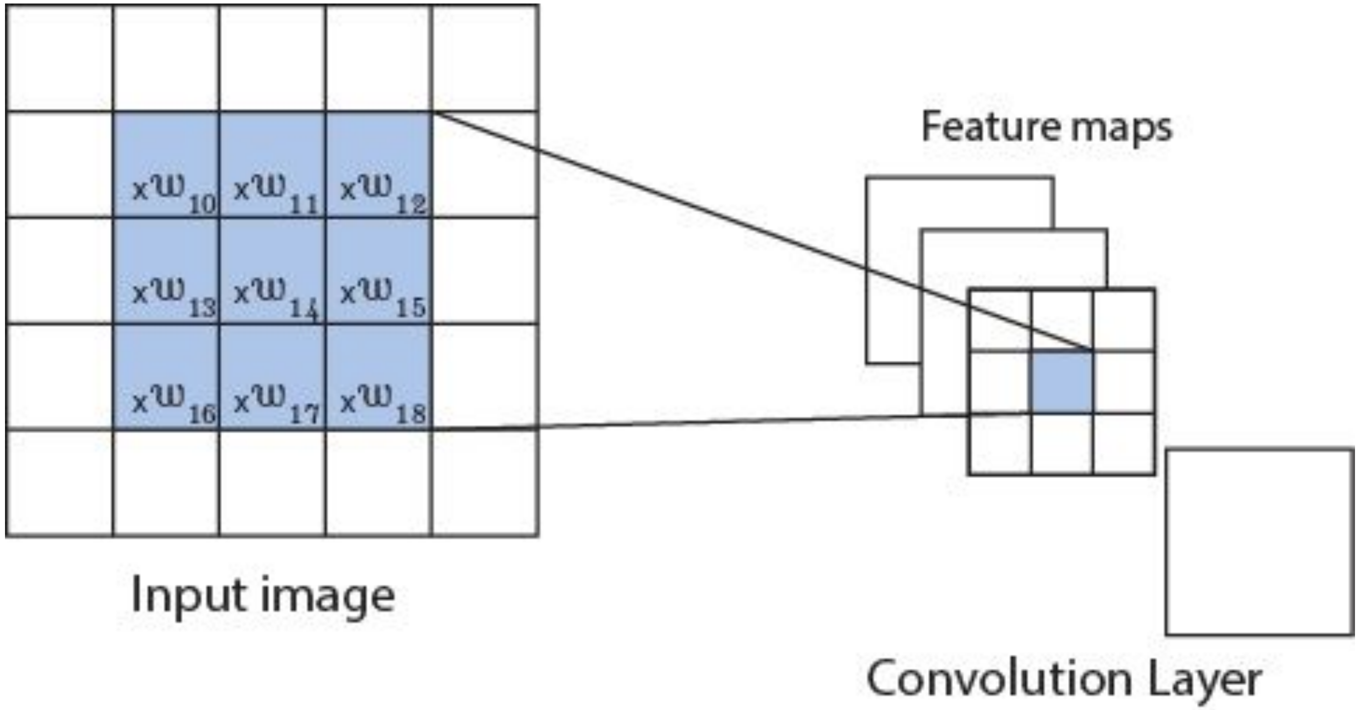
CNN - Example - 2nd Feature Map



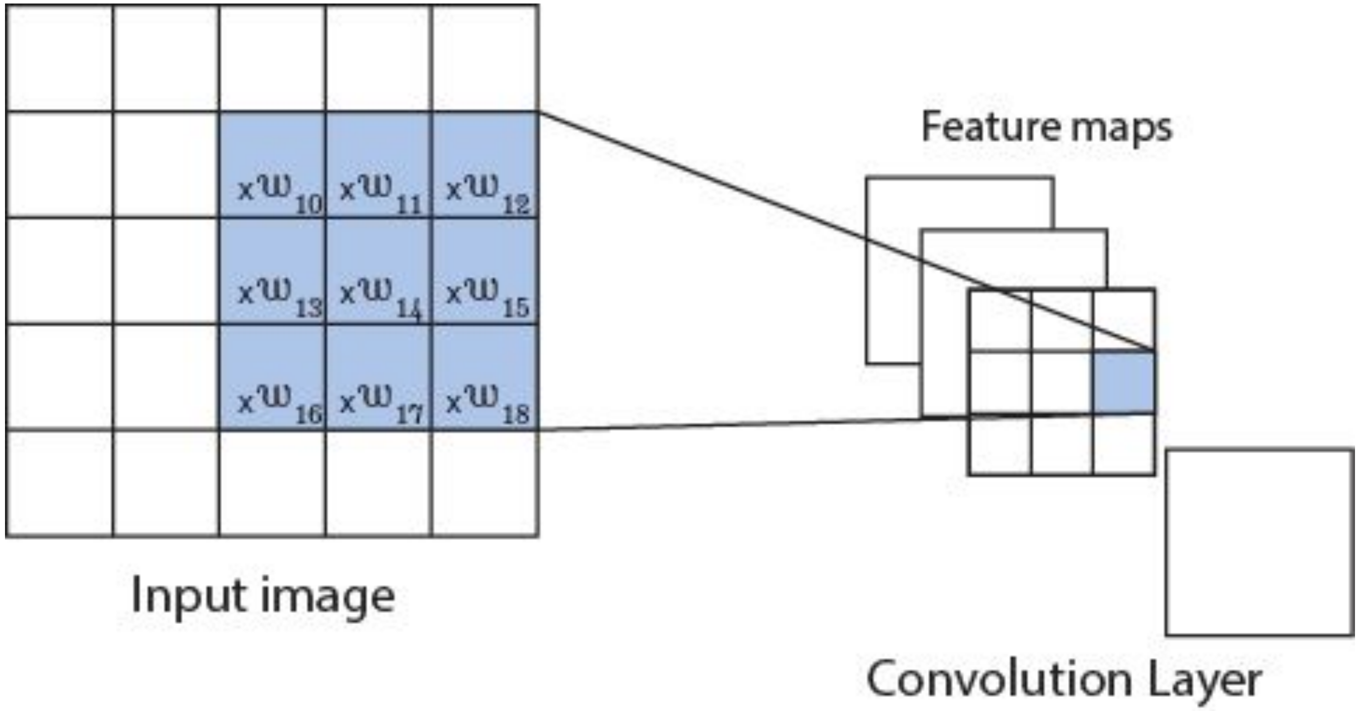
CNN - Example - 2nd Feature Map



CNN - Example - 2nd Feature Map

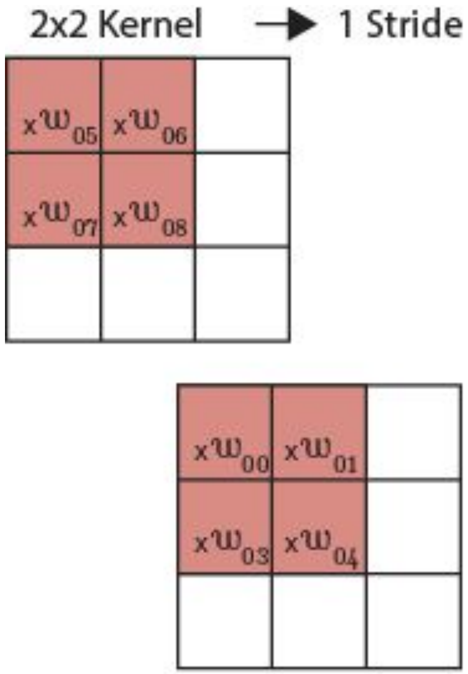


CNN - Example - 2nd Feature Map

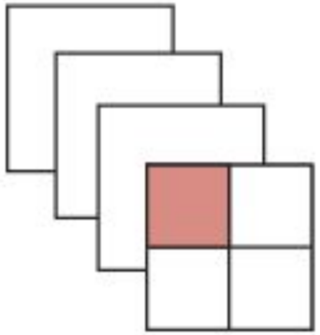


CNN - Example - Consecutive Convolutions

- Each filter in above layer performs convolution on all filters in previous layer, same for colour channels.



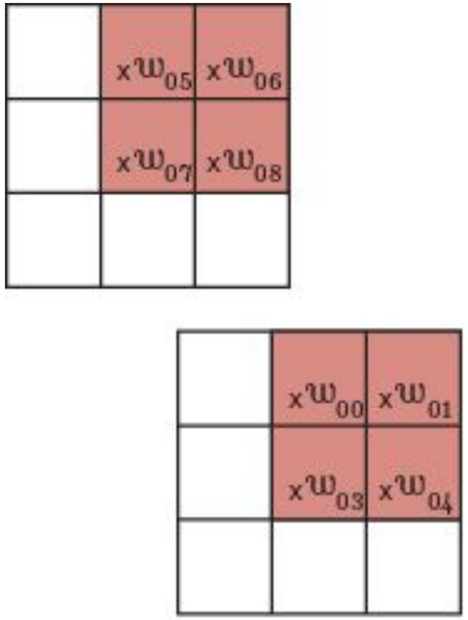
Convolution with 2 feature maps



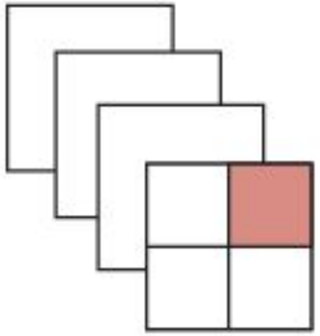
Convolution Layer

CNN - Example - Consecutive Convolutions

- Each filter in above layer performs convolution on all filters in previous layer, same for colour channels.



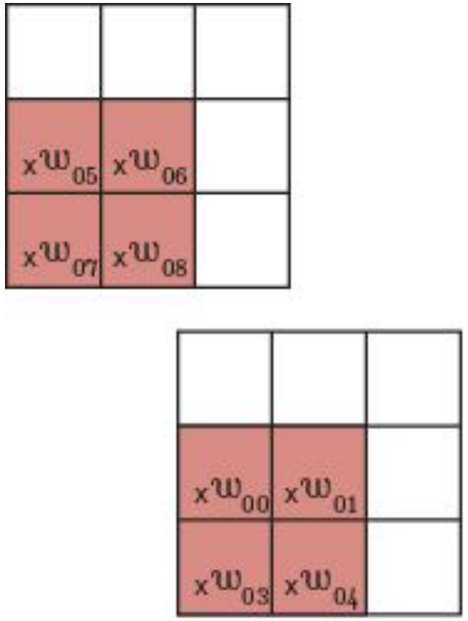
Convolution with
2 feature maps



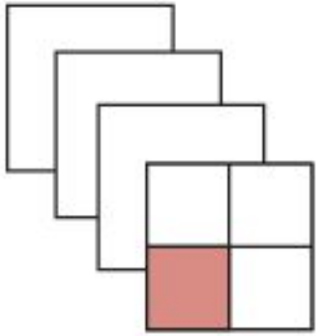
Convolution Layer

CNN - Example - Consecutive Convolutions

- Each filter in above layer performs convolution on all filters in previous layer, same for colour channels.



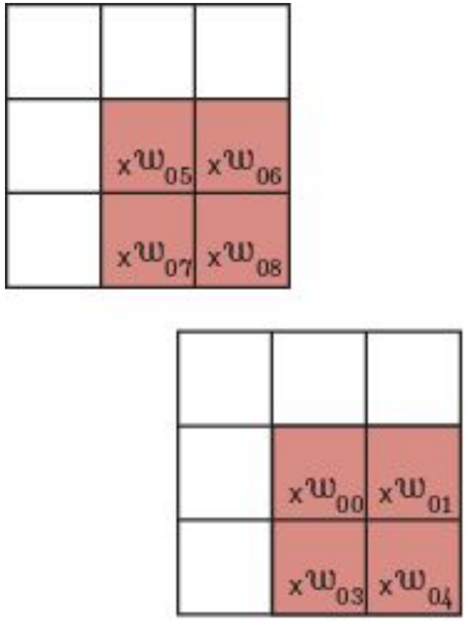
Convolution with 2 feature maps



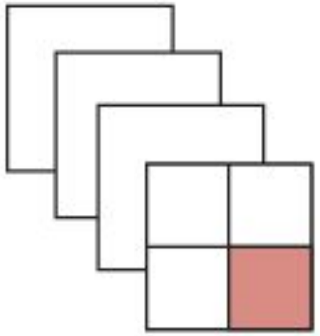
Convolution Layer

CNN - Example - Consecutive Convolutions

- Each filter in above layer performs convolution on all filters in previous layer, same for colour channels.



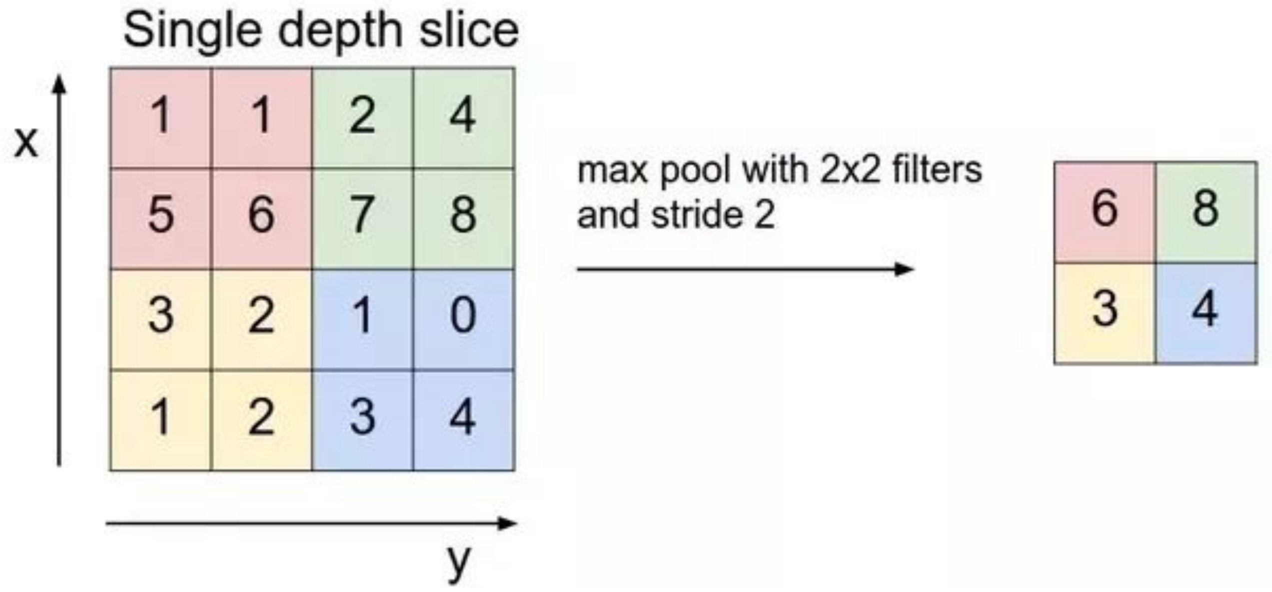
Convolution with
2 feature maps



Convolution Layer

Pooling

- Pooling performs subsampling and reduces network size
- Example of MAX pooling (selecting the maximum value)



[http://cs231n.github.io/]

Instructor:

DEEP
LEARNING
INSTITUTE

www.nvidia.com/dli