**PROJECT #5 - Deep Learning Dog Breed Classifier**

| | | | |
|---|---|---|---|
| **Notebook:** | Artificial Intelligence NanoDegree | | |
| **Created:** | 1/8/2018 8:54 AM | **Updated:** | 1/12/2018 9:57 AM |
| **Author:** | Matthew R. Versaggi | | |
| **URL:** | https://hackernoon.com/learning-ai-if-you-suck-at-math-p4-tensors-illustrated-with-cats-27f0002c9b32 | | |

## PROJECT #5 - Deep Learning Dog Breed Classifier

**Agenda: -** ~~Not~~ a Primarily Theory Lecture, more a tactical / strategy overview + submission walk through.

1. Quick Overview of External Resources
2. Quick Survey of Theory Resources
3. Quick Intro to Practical Examples
4. Submission Walk Through
5. Lessons Learned
6. Q&A

## (1) Quick Overview of **External References**:

- **MLM Deep Learning w/ Python** - PDF
  - C:\Users\versaggi\Transfer\VERSAGGI-FOLDER\Desktop\AI Stuff\Deep_Learning\MLM_DeepLearning_Python
  - deep_learning_with_python.pdf
  - **NOTES**:
    - All of the code works out of the box, excellent examples of complicated stuff - great for future reference after reading all the rest of the materials.
    - **CH 19** - Simple Convolutional NN for MNIST (End to End Example)

- **DL Fundamentals** - PDF
  - C:\Users\versaggi\Transfer\VERSAGGI-FOLDER\Desktop\AI Stuff\Deep_Learning
  - Fundamentals of Deep Learning.pdf
  - **Notes:**
    - Very **good** general DL theory fundamentals (very **bad** code examples - pre TF 1.X code)
    - Read for the theory, don't stay for the code.

- **DL with TensorFlow** - PDF
  - Excellent general DL intro to CNN's (among other topics)
  - C:\Users\versaggi\Transfer\VERSAGGI-FOLDER\Desktop\AI Stuff\Deep_Learning\DeepLearning_Tensorflow
  - DEEP_LEARNING_WITH_TENSORFLOW.pdf
  - **Notes:**
    - One MUST understand Tensor Flow first to know / appreciate what is happening under the hood - decent reference.
    - **Ch 4** contains a decent intro to CNN's along with code that works. (ignore the first two chapters code - it's buggy).

- **DL with Keras** - PDF
  - C:\Users\versaggi\Transfer\VERSAGGI-FOLDER\Desktop\AI Stuff\Deep_Learning\DeepLearning_Keras
  - DEEP_LEARNING_WITH_KERAS.pdf
  - **Notes:**
    - Excellent general DL intro to CNN's (among other topics)
    - **Udacity uses Keras** (with TensorFlow on the backend) for the CNN project - so you should spend some time here.
    - **Ch 3 is CNN's** - Recognizing **CIFAR10** Images with DL is very much like the problem submission and the code is solid so it's beneficial to focus on that.

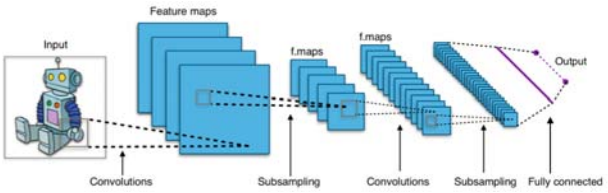
## (2) Quick Survey of **Theory Resources**:

- **NN Basics**

**NN PlayGround:**
http://playground.tensorflow.org/#activation=tanh&batchSize=10&dataset=circle&regDataset=reg-plane&learningRate=0.03&regularizationRate=0&noise=0&networkShape=4,2&seed=0.14585&showTestData=false&discretize=false&percT

- **Convolutional Basics**

- These nets use an ad **hoc architecture** inspired by biological data taken from **physiological experiments done on the visual cortex**. As discussed, **our vision is based on multiple cortex levels**, each one recognizing more and more structured information. **First**, we see single pixels; **then** from them, **we recognize** simple geometric forms. **And then...** more and more sophisticated elements such as **objects, faces, human bodies, animals, and so on**.

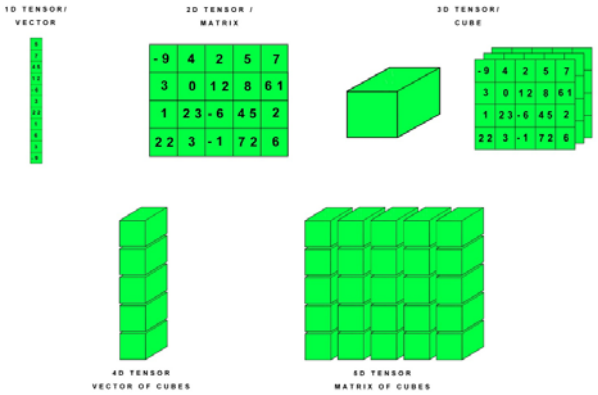



**DL and CNN's in Plain English**
https://hackernoon.com/learning-ai-if-you-suck-at-math-p5-deep-learning-and-convolutional-neural-nets-in-plain-english-cda79679bbe3

- **TensorFlow Basics**

**Tensor** is the core concept for TensorFlow:
*TensorFlow programs use a tensor data structure to represent all data — only tensors are passed between operations in the computation graph. You can think of a TensorFlow* ***tensor*** *as an* ***n-dimensional array or list****.*

**Learning AI if you suck at Math**
https://hackernoon.com/learning-ai-if-you-suck-at-math-p4-tensors-illustrated-with-cats-27f0002c9b32




### EverNote NoteBooks

- DL Fundamentals
- DL in TensorFlow
- DL in Keras

## (3) Quick Intro of Practical Examples:

**MLM DL in Python:**

- C:\Users\Matthew\workspaces\ML_Mastery\ML_Deep_Learning\code\chapter_19
- **mnist_cnn.py**

**DL In TensorFlow:**

- C:\Users\Matthew\workspaces\Deep_Learning\DeepLearning_Tensorflow\DeepLearningwithTensorFlow_Code\Chapter0**4**\MNIST_CNN\Python 3.5
- **mnist_cnn_1.py**

**DL in Keras:**

- C:\Users\Matthew\workspaces\Deep_Learning\DeepLearning_Keras\DeepLearningwithKeras_Code\Chapter0**3**
- **keras_CIFAR10_V1.py**

## (4) Submission Walk-Through:

- **Experiment**
  - I used only outside resources to do the submission - **none** of the Udacity Materials.
  - I probably spent 10 times the amount of time though.

**Jupyter Notebook**

- C:\Users\Matthew\workspaces\Udacity\AIND_DL\Project_5\dog-project-master
- dog_app.ipynb
- **Notes:**
  - There is a lot of python OS pathing wrangling going on so be forewarned. ("\\") is your friend.
  - Much of the work is not DL oriented directly, but integration of DL oriented per their strategy of showing complex real world examples instead of sterile academic ones. Good practice.
  - Make sure you understand how the "extract_bottleneck_features.py" works, it's simple.
  - Understand what Transfer Learning is and how to work with it.
  - Keep your Epocs less than 10 (more like 6).
  - Be SURE to comply with the Rubric.

## (5) Lessons Learned:

- T**he Art to this domain is far more important that the science**, which means you absolutely must obtain a solid grasp of the basic theory - particularly as implementations of DL differ (TensorFlow, Keras, Theano, BigDL, DL4J, etc...)
- The entry barriers to this domain are high - Which is a very good thing because:
  - **Hard to Learn but relatively easy to apply** after you reach a critical mass of core competencies.
  - You have to pay the price in terms of time invested to be good.
- Setting up your environment is half the learning curve
- Udacity is interested in showing integrated examples / work with practical applications over academic examples - be prepared for complex work
- Practical examples in good practitioners texts are invaluable for the newbie
- Online Searches are particularly rich in this space with materials
- It'll run on a 16GB RAM Box w/a decent CPU.
- You have to put in your time, but get the solid basics first.
- Run examples out of the Jupyter NB - it's faster, then back fill.

## (6) Q&A:

### Misc Stuff

**GitHub Project repository:**
https://github.com/udacity/dog-project

**Rubric:**
https://review.udacity.com/#!/rubrics/810/view

### Learning AI if You Suck at Math

**Learning AI If You Suck at Math—Part 1**—This article guides you through the **essential books** to read if you were never a math fan but you're learning it as an adult.

**Learning AI If You Suck at Math—Part 2—Practical Projects**—This article guides you through **getting started** with your first projects.

**Learning AI If You Suck at Math—Part 3—Building an AI Dream Machine**—This article guides you through getting a powerful deep learning machine setup and installed with all the latest and greatest frameworks.

**Learning AI If You Suck at Math—Part 4—Tensors Illustrated (with Cats!)**—This one answers the ancient mystery: What the he**ll** is a **tensor**?

**Learning AI If You Suck at Math—Part 5—Deep Learning and Convolutional Neural Nets in Plain English**—Here we create our first Python program and explore the inner workings of neural networks!

**Learning AI If You Suck at Math—Part 6—Math Notation Made Easy**—Sti**ll** struggling to understand those funny little symbols? Let's change that now!
**Learning AI If You Suck at Math—Part 7—The Magic of Natural Language Processing**—Understand how Google and Siri understand what you're mumbling.