Welcome to MOOC era!

My experiences with
Deep Learning Foundations Nanodegree at **Udacity**

Kamil Krzyk

Android Platform Engineer at

AZIMO

Contact:

krzyk.kamil@gmail.com

@F1sherKK

AzimoLabs

FisherKK/F1sherKK-MyRoadToAI

Presentation plan

Part I – MOOC in general

- MOOC definition.
- Where to find MOOC?
- Few opinions of experienced people from AI industry about MOOC.
- My experiences with MOOC Services.
- My comparison of MOOC Services.

Presentation plan

Part I – MOOC in general

- MOOC definition.
- Where to find MOOC?
- Few opinions of experienced people from AI industry about MOOC.
- My experiences with MOOC Services.
- My comparison of MOOC Services.

Part II – Deep Learning Nanodegree

- Thorough feedback from Deep Learning Nanodegree at Udacity.

What's MOOC?

- MOOC stands for:

  Massive Open Online Course

- Term appeared for the first time in year 2006.
- MOOC popularity started to rapidly increase since 2012.
- Form of knowledge sharing by providing course attendant with access to materials such as filmed lectures, readings, problem sets and access to support community where interaction between students and teachers can occur.

coursera

edX

UDACITY

KHAN ACADEMY

udemy

DataCamp

coursera

UDACITY

udemy

Andrew Ng

"AI is the new electricity."

- Former Chef Scientist at Baidu
- Adjunct Professor at Stanford University
- Founded and led Google Brain project at Google
- Co-founder of world most popular MOOC service - Coursera

I could not find interview with Andrew Ng where he said that world is changing so fast, people need to be given possibility to re-learn new skills in order to adjust to changes in industry. He sees big potential in online courses like Coursera offers.

In interview at MIT Technology Review, 23.05.2016:

When asked by interviewer:

“Are there enough of you being train by universities, so that everyones company can hire an Chief AI Officer?”

Andrew Ng responded:

“I have one word for you... Coursera. No really there isn't, there is not nearly enough AI talent in the world today. But that's why the MOOC platforms like Coursera, Udacity, Udemy, edX and so one - I think those will help.”

Opinions I've heard at GDD Kraków

Andrew Gasparovic

Leader of Applied Machine Intelligence
team at Google Research Europe in Zurich

Presented:

Machine Learning with Tensorflow

Opinions I've heard at GDD Kraków

Andrew Gasparovic

Leader of Applied Machine Intelligence team at Google Research Europe in Zurich

Presented:

Machine Learning with Tensorflow

Mark Daoust

Developer Programs Engineer for TensorFlow, 9 years building embedded ML models for aircraft

Workshop:

Hands-on Running A TensorFlow Model on Android

Opinions I've heard at GDD Kraków

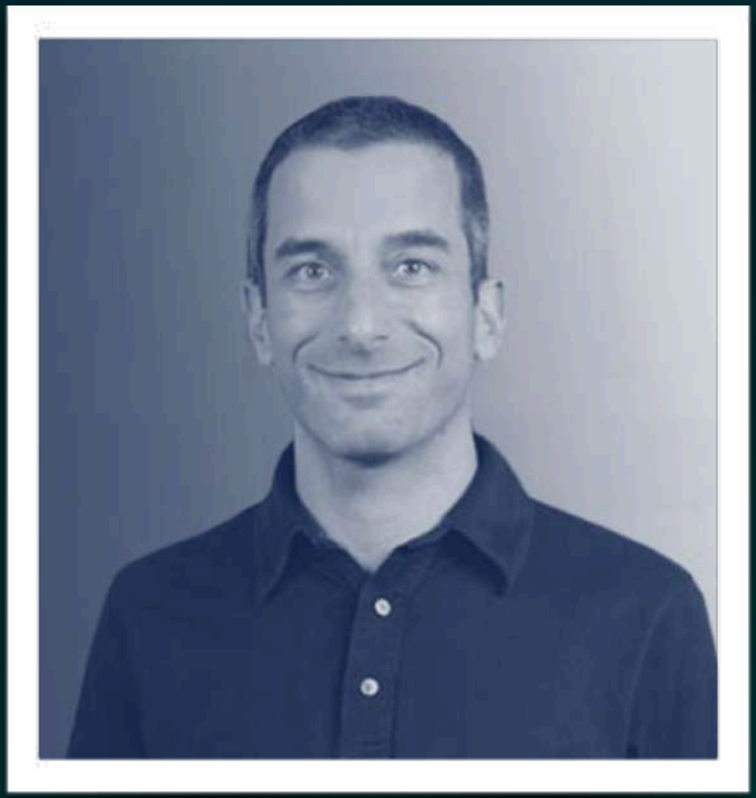
Andrew Gasparovic

"It's great that you have finished Deep Learning Nanodegree at Udacity. I think you should definitely apply for job in this area. This field is changing and you will have to keep learning anyway the same way as you do now."

Mark Daoust

"Before I started working at Google I have also taken online courses like Machine Learning on Coursera or Machine Learning Nanodegree at Udacity."

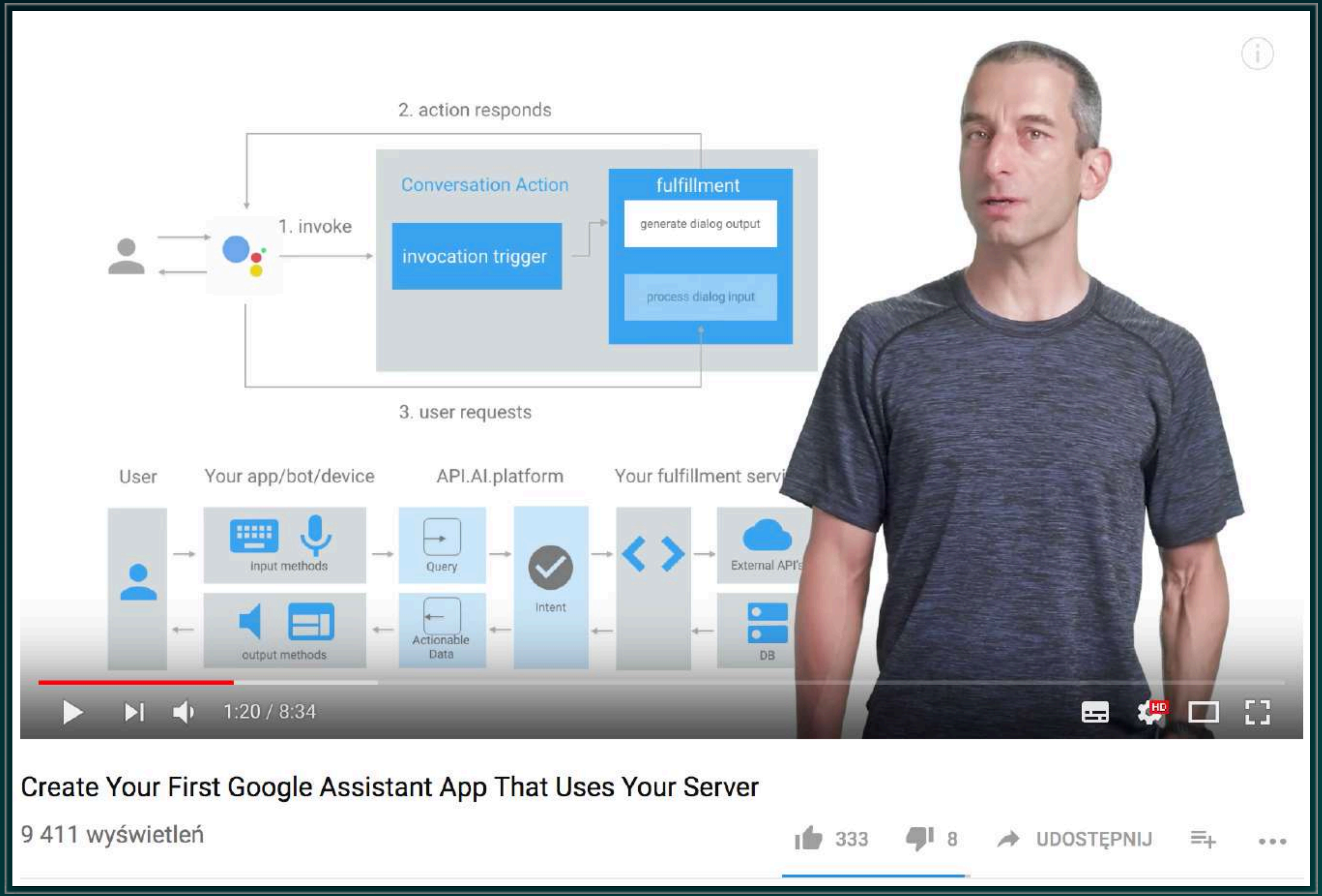
Opinions I've heard in podcast biznesmysli.pl



Ido Green

Developer Advocate at Google

Opinions I've heard in podcast biznesmysli.pl



He is focusing on sharing experience how to start use Google Home (and related topics, like Google Assistant).

Opinions I’ve heard in podcast biznesmysli.pl

When asked by Vladimir Alekseichenko:

"What recommendations do you have for companies who don't use Machine Learning now but they want start it?"

Ido Green responded:

“(…) I would try one of the courses, you have today, on Udacity, Coursera and other sites that let's you dive into this world of Machine Learning.It's really interesting to see what was the progress in the previous years and what are the best algorithms you could tap into and use for your use cases.”

Sebastian Thurn

- Founded Google X and Google's self-driving car teams
- Adjunct Professor at Georgia Tech and Stanford University
- Founder and President of Udacity
- In 2005 team led by him created DARPA autonomous car that completed and won 212km race through Mojave desert in Nevada

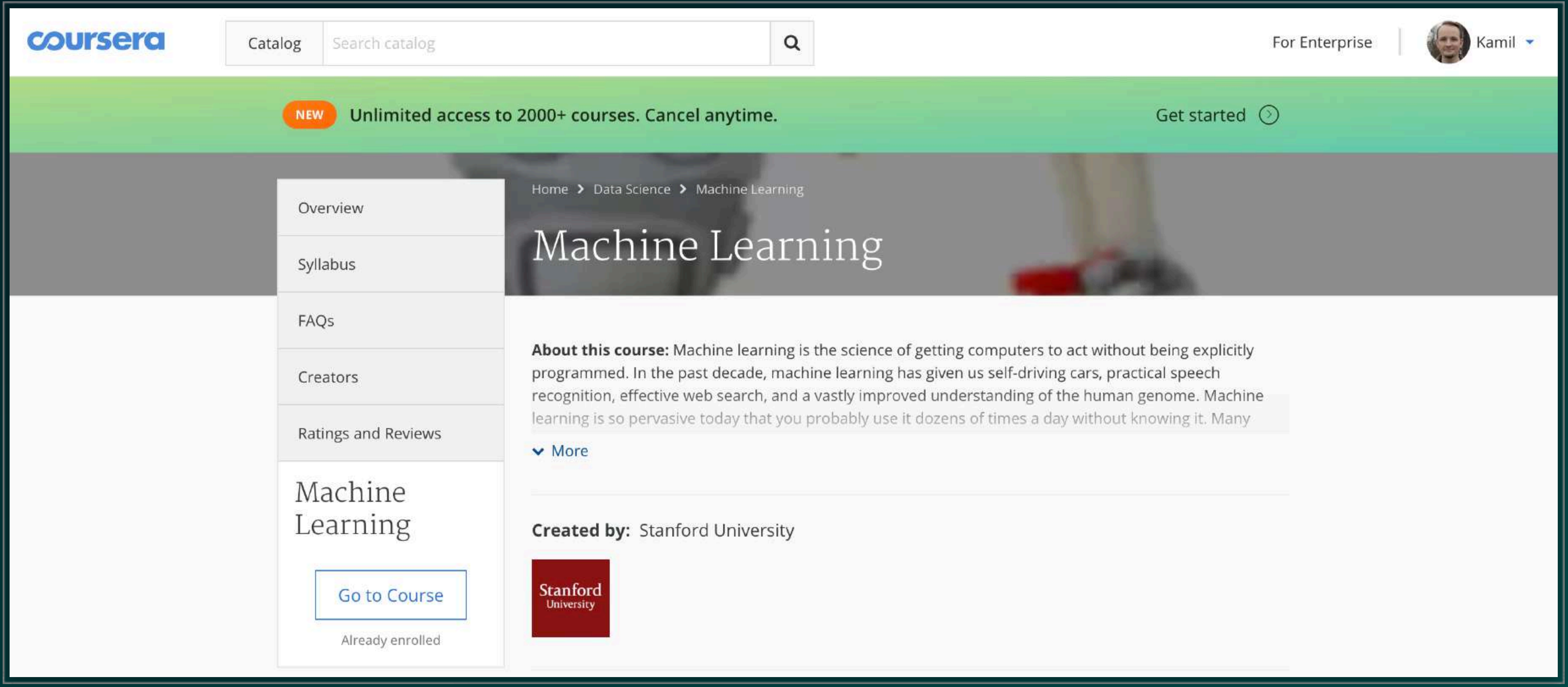
During Talk Education at Frontier Tech

"I think that world is moving from single education to lifelong education. We can't afford single education anymore. (...) The world is changing so rapidly that I believe that we (Udacity) will become lifelong service provider."

My Experiences with Coursera

Machine Learning

by Andrew Ng at Coursera



Machine Learning
by Andrew Ng at Coursera

Price: 79$ once

Expected completion time: 3 months

Knowledge validation by:

- quiz (10 questions, 3 tries every 8 hours)
- programming assignment in Matlab sent to Stanford University server where code is tested on different data than in assignment

- Linear Regression
- Logistic Regression
- Regularization
- Neural Network:
  - Multilayer Perceptron Intuition
  - Backpropagation
- Implementation and testing advices
- Support Vector Machines
- K-Means (basics of Unsupervised Learning)
- Principal Component Analysis
- Anomaly Detection
- Recommender Systems
- Some info how ML works at larger scale
- Introduction to Optical Character Recognition

Machine Learning

by Andrew Ng at Coursera

Machine Learning
Stanford University
Grade Achieved: 100.0%

Instructions

Enter the following in the Add certification section in LinkedIn:

- Certification name: Machine Learning
- Certification authority: Coursera
- License number: HF3R7P7JNS5S
- Time period - From: February 2017
- Time period - To: (blank - toggle the checkbox for *This certification does not expire*)
- Certification URL: https://www.coursera.org/account/accomplishments/verify/HF3R7P7JNS5S

Add to LinkedIn

Stanford | ONLINE

COURSE CERTIFICATE

02/02/2017

Kamil Krzyk

has successfully completed

Machine Learning

an online non-credit course authorized by Stanford University and offered through Coursera

EDUCATION FOR EVERYONE · coursera · COURSE CERTIFICATE

Associate Professor Andrew Ng
Computer Science Department
Stanford University

SOME ONLINE COURSES MAY DRAW ON MATERIAL FROM COURSES TAUGHT ON-CAMPUS BUT THEY ARE NOT EQUIVALENT TO ON-CAMPUS COURSES. THIS STATEMENT DOES NOT AFFIRM THAT THIS PARTICIPANT WAS ENROLLED AS A STUDENT AT STANFORD UNIVERSITY IN ANY WAY. IT DOES NOT CONFER A STANFORD UNIVERSITY GRADE, COURSE CREDIT OR DEGREE, AND IT DOES NOT VERIFY THE IDENTITY OF THE PARTICIPANT.

Verify at coursera.org/verify/HF3R7P7JNS5S
Coursera has confirmed the identity of this individual and their participation in the course.

My Experiences with Udemy

Discounts everywhere... always!

Machine Learning A-Z™: Hands-On Python & R In Data Science BEST SELLER in Data Science

Kirill Eremenko• Data Scientist & Forex Systems Expert

Learn to create **Machine Learning** Algorithms in Python and R from two Data Science experts. Code templates included.

276 lectures · 41 hours · All Levels · English [Auto-generated]

42 zł ~~610 zł~~

4.5 (20,290 ratings)

Deep Learning A-Z™: Hands-On Artificial Neural Networks BEST SELLER

Kirill Eremenko• Data Scientist & Forex Systems Expert

Learn to create Deep **Learning** Algorithms in Python from two **Machine Learning** & Data Science experts. Templates included.

179 lectures · 22.5 hours · All Levels · English [Auto-generated]

42 zł ~~610 zł~~

4.5 (4,581 ratings)

Python for Data Science and Machine Learning Bootcamp

Jose Portilla• Data Scientist

Learn how to use NumPy, Pandas, Seaborn , Matplotlib , Plotly , Scikit-Learn , **Machine Learning**, Tensorflow , and more!

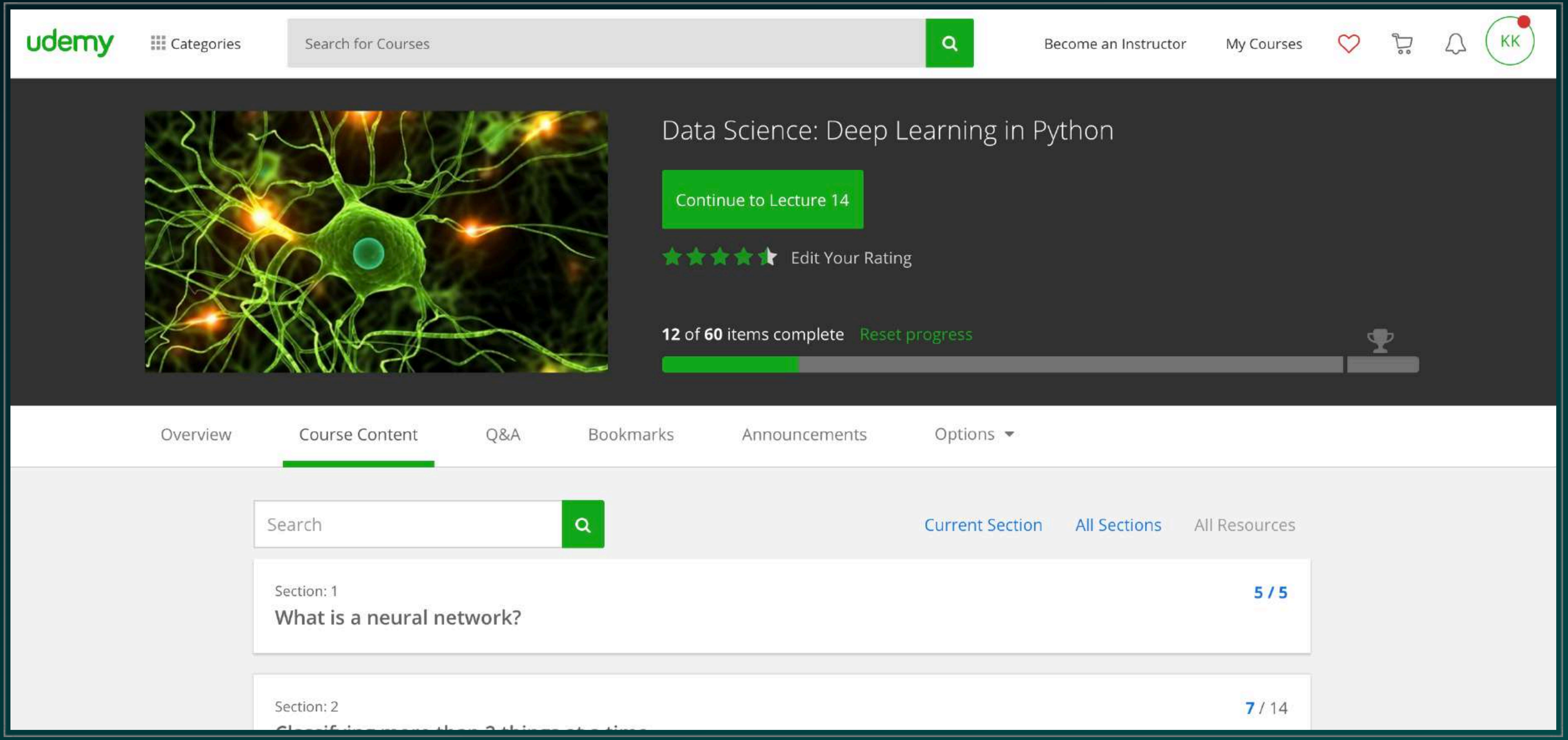
143 lectures · 21.5 hours · All Levels · English

42 zł ~~595 zł~~

4.6 (9,785 ratings)

Discounts everywhere... always!

How course looks like?



How course looks like?

Section: 3

Pandas **6 / 6**

12.	Manual Data Loading	4:33
13.	DataFrames	3:22
14.	More about DataFrames: Selecting Rows and Columns	4:33
15.	Even More about DataFrames: Column Names	3:21
16.	The apply() Function	3:14
17.	Joins	2:32

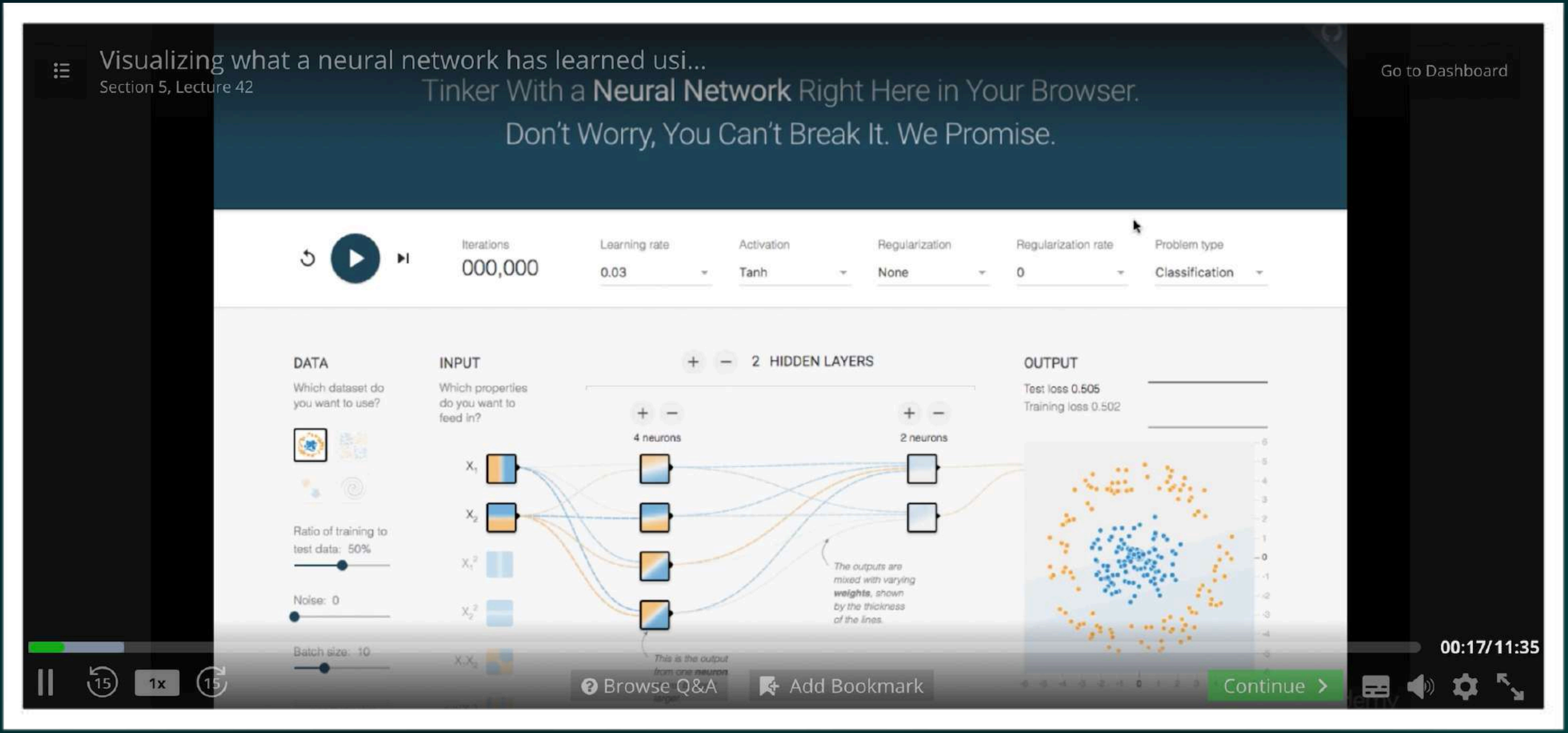
How course looks like?

Section: 3

Pandas 6 / 6

12.	Manual Data Loading	4:33
13.	DataFrames	3:22
14.	More about DataFrames: Selecting Rows and Columns	4:33
15.	Even More about DataFrames: Column Names	3:21
16.	The apply() Function	3:14
17.	Joins	2:32

How course looks like?



Certificate of Completion

This is to certify that Kamil Krzyk successfully completed the Deep Learning Prerequisites: The Numpy Stack in Python online course on March 12, 2017

Lazy Programmer Inc.

Lazy Programmer Inc., Instructor

&

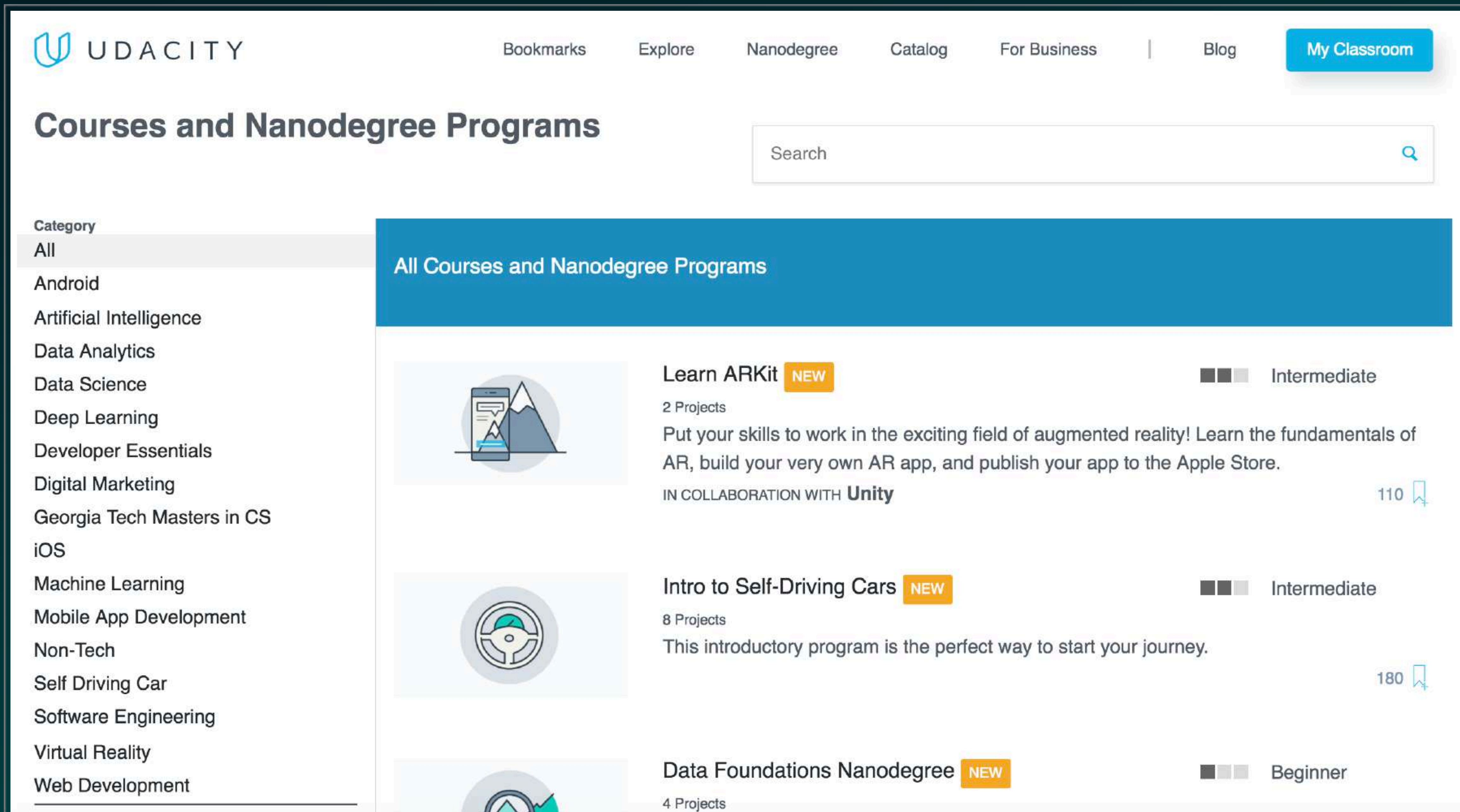
udemy

NEVER STOP LEARNING

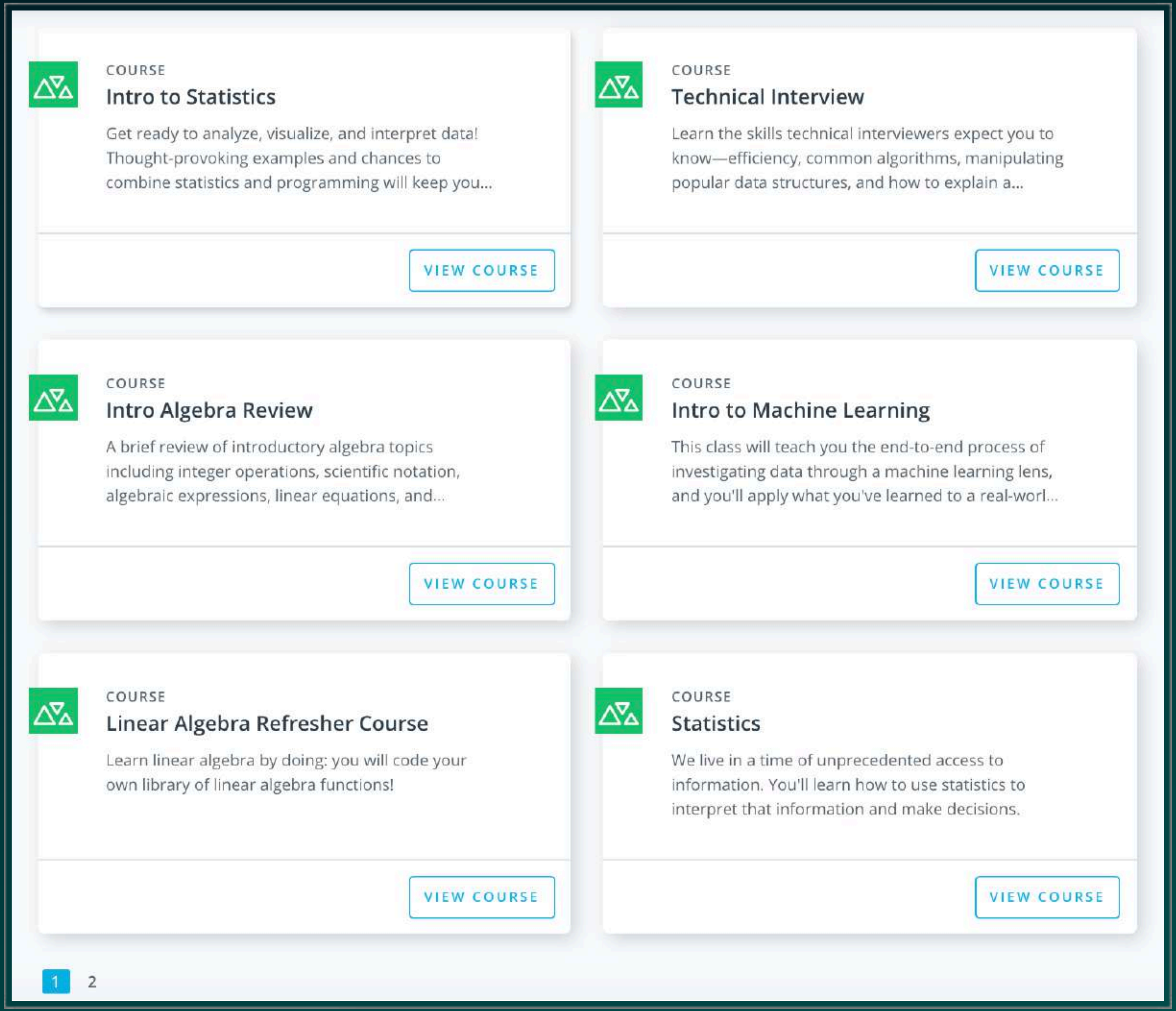
Certificate no: **UC-1WER9BMD**
Certificate url: **ude.my/UC-1WER9BMD**

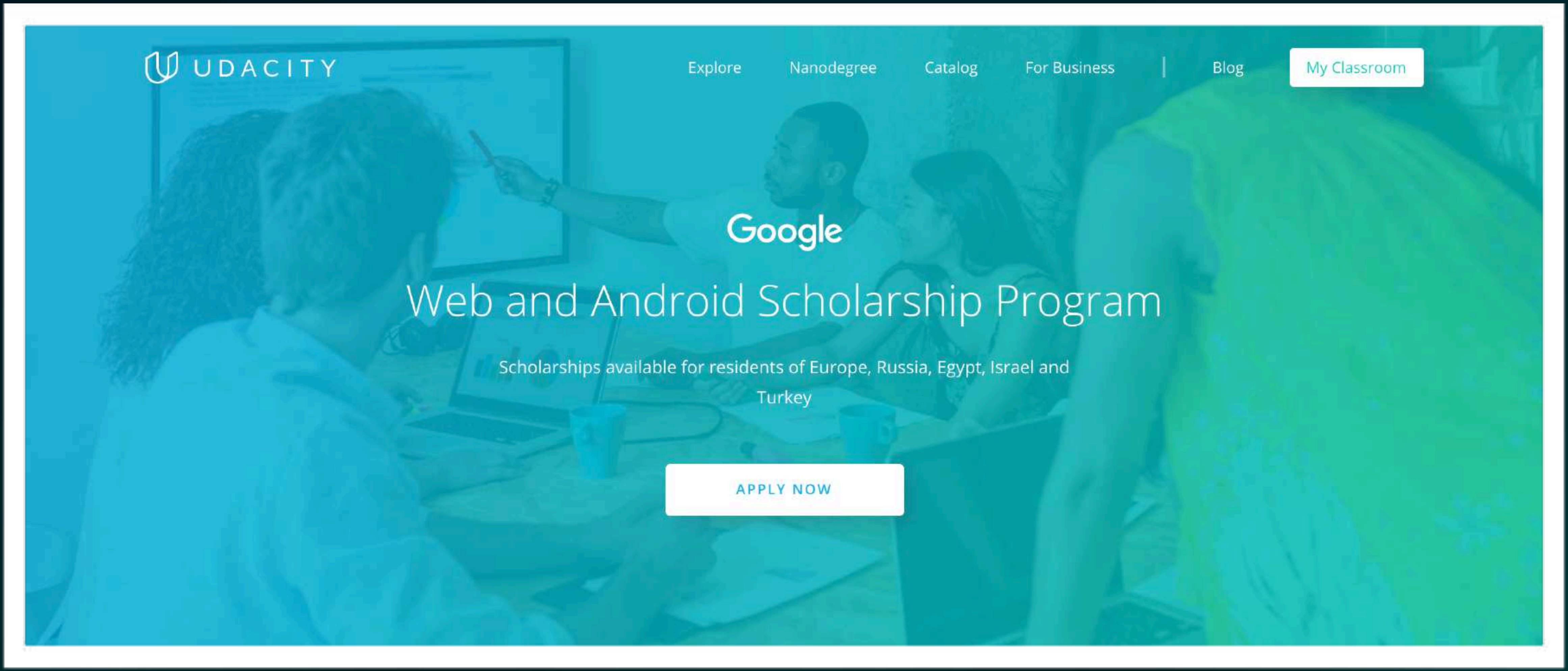
My Experiences with Udacity

My Experiences with Udacity



My Experiences with Udacity



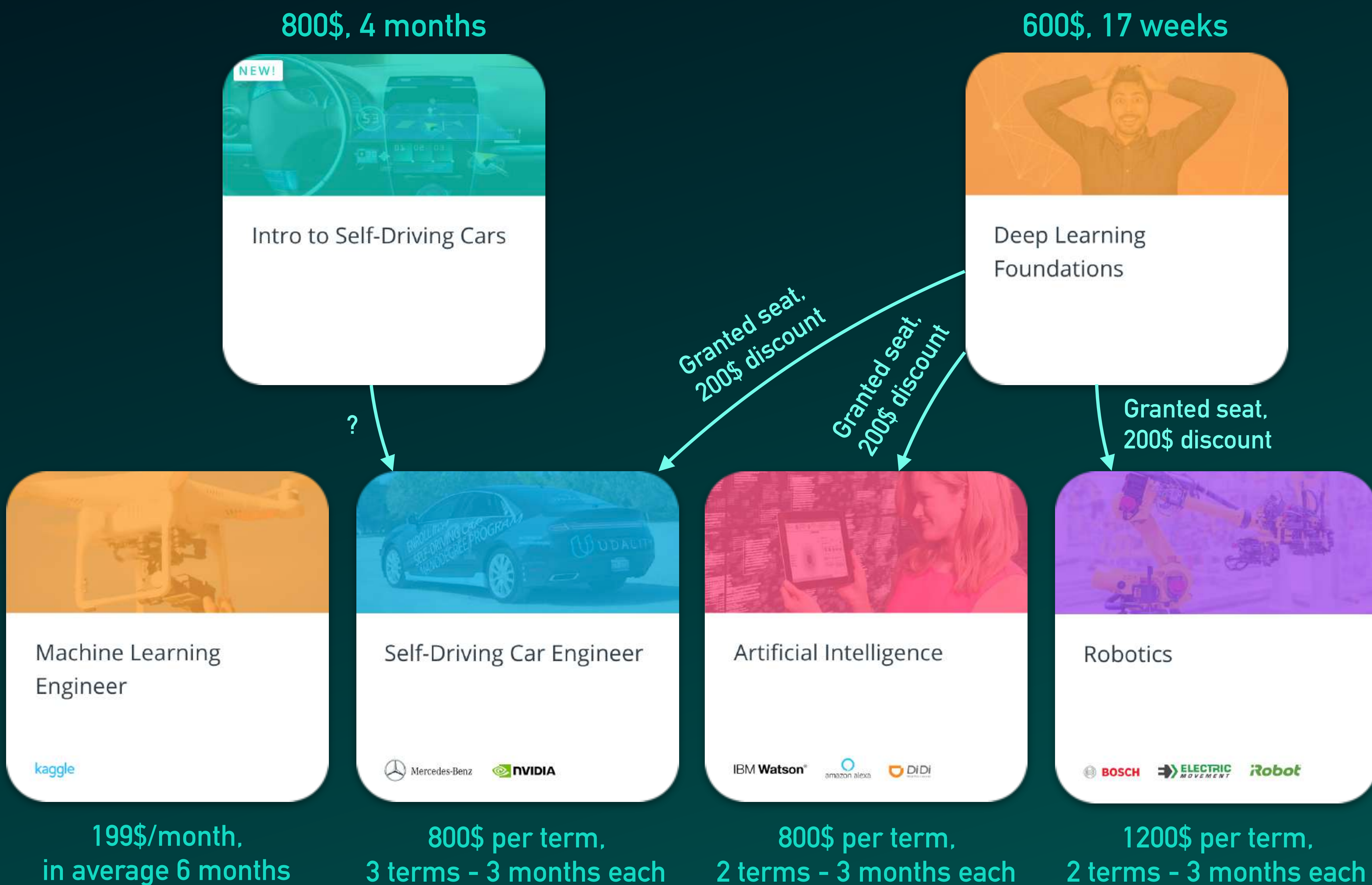


Application deadline: 15.11.2017 (unfortunately 3 days after my talk)
Results announcement: 30.11.2017

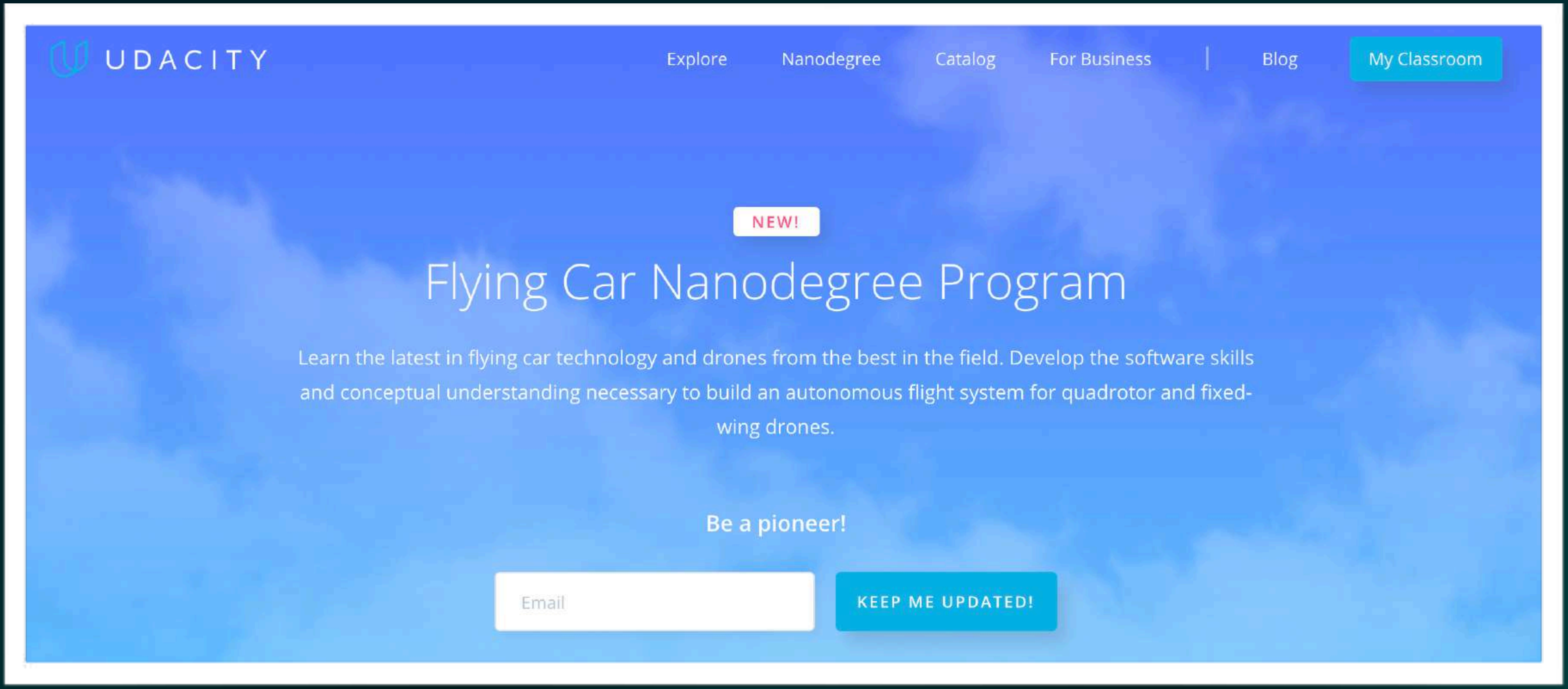
20 000 seats for Beginner Track for both Web/Android
10 000 seats for Intermediate Track for both Web/Android

3 months, access to Scholarship Course Content & Materials, Mentoring, Certificate

AI related Nanodegrees at Udacity

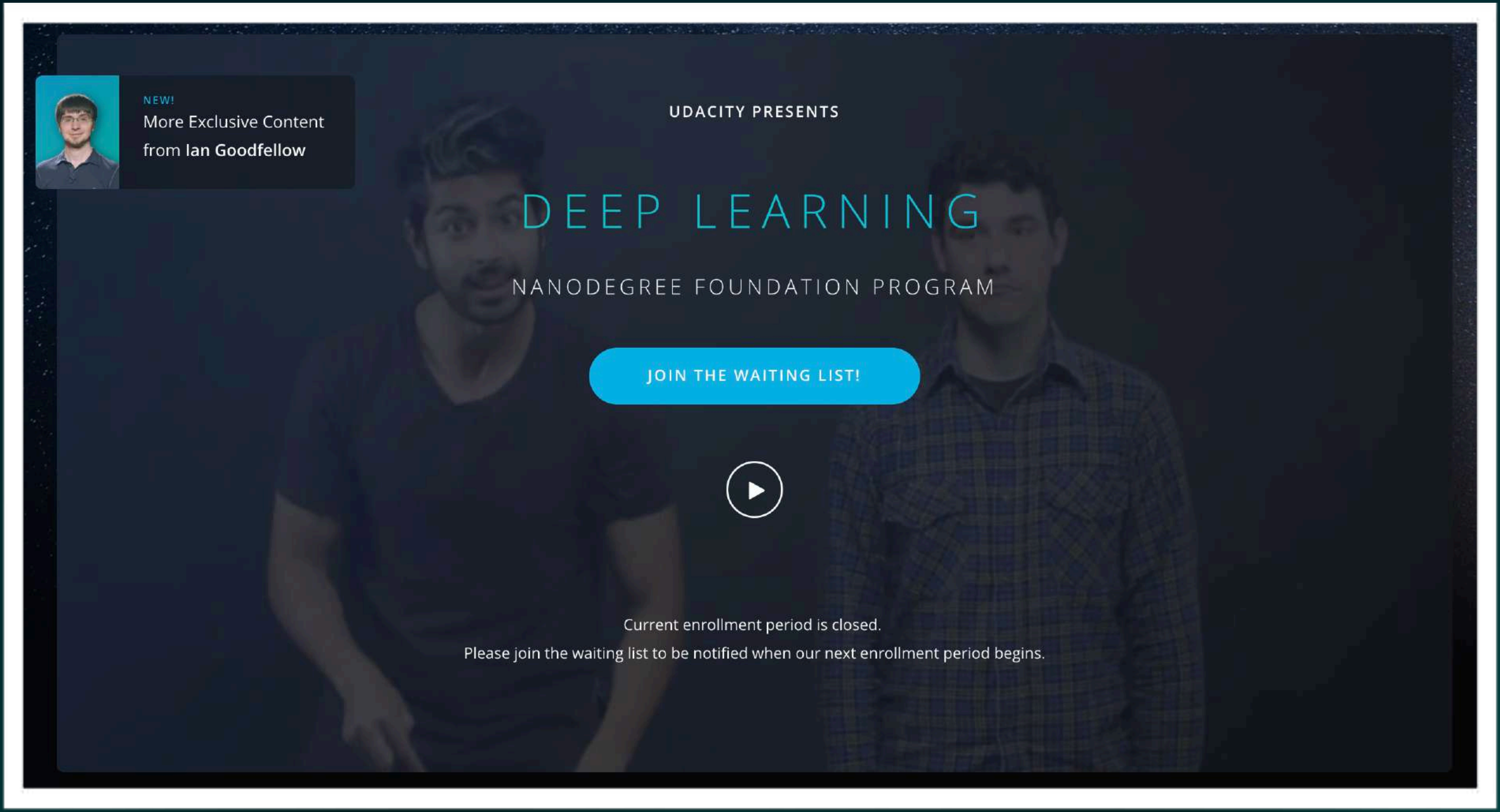


Incoming Nanodegrees at Udacity



Unconfirmed gossips about "Deep Learning for Cyber Security"

My experiences with Deep Learning Nanodegree at Udacity



General information about course

Recruitment for new cohort starts every 2 months.

There are ~1000 seats per cohort.

It costs 600$ (first 2 cohorts had discount 200$).

It lasts 17 weeks (+ bonus ~1 month to finish if you were behind).

It's not self-paced course - you receive new content every week.

If you finish all 5 projects before deadline you are granted slot for Robotics/AI/ Self-Driving Car Nanodegrees with 200$ discount.

LIFETIME ACCESS if you pass!

Most Seen Faces

Mat Leonard

Post-Doctoral Researcher at UC Berkeley

Nanodegree Lead and Senior
Content Developer at Udacity

Siraj Raval

YouTube Star - 200k Subscriptions

Developers educator and AI community builder

Self-employed owner of Siraj Raval Company

www.youtube.com/c/sirajology

Guests

Ian Goodfellow

Staff Research Scientist at Google Brain

Inventor of Generative Adversarial Networks

Andrew Trask

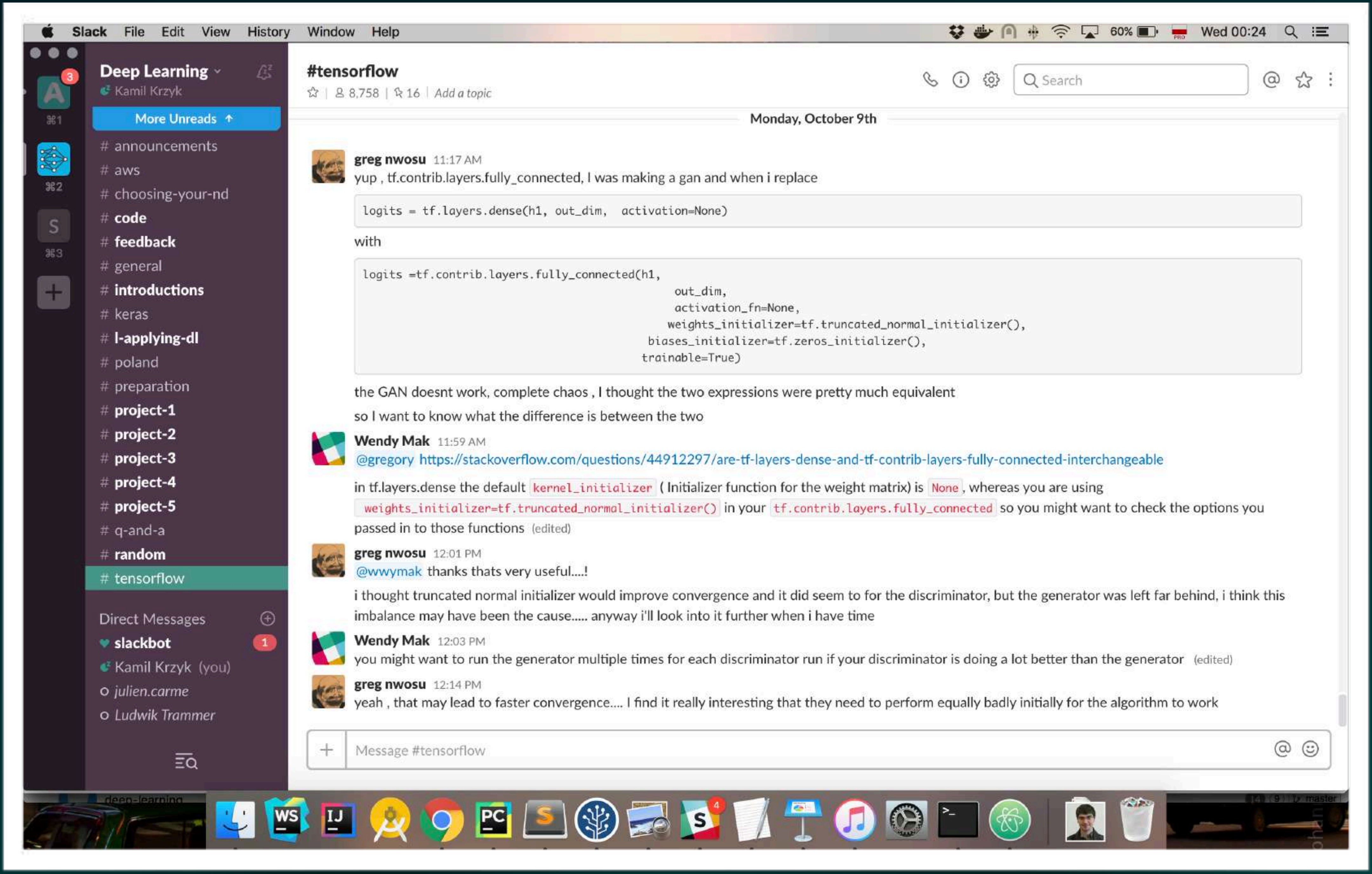
Research Scientist at Google Brain, Doctoral Student at Oxford

Author of Grokking Deep Learning book

And others which I haven't mentioned...

Thank you!

Access to Slack Channel



Access to Slack Channel

announcements

feedback

general

random

Access to Slack Channel

announcements
feedback
general
random

project 1
project 2
project 3
project 4
project 5

Access to Slack Channel

announcements
feedback
general
random

project 1
project 2
project 3
project 4
project 5

january_class
march_class
june_class
august_class
...

Access to Slack Channel

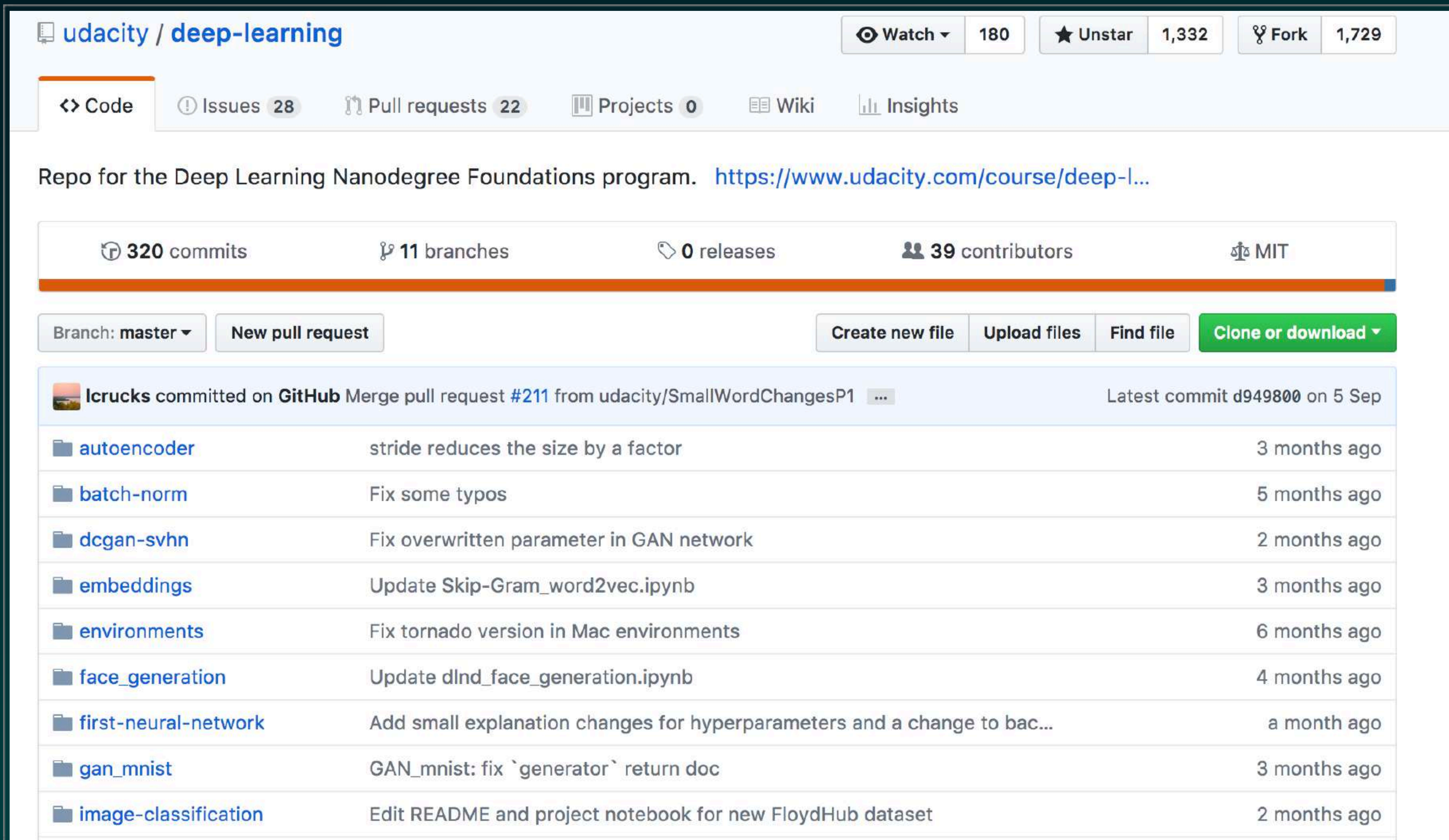
announcements
feedback
general
random

tensorflow
keras
code

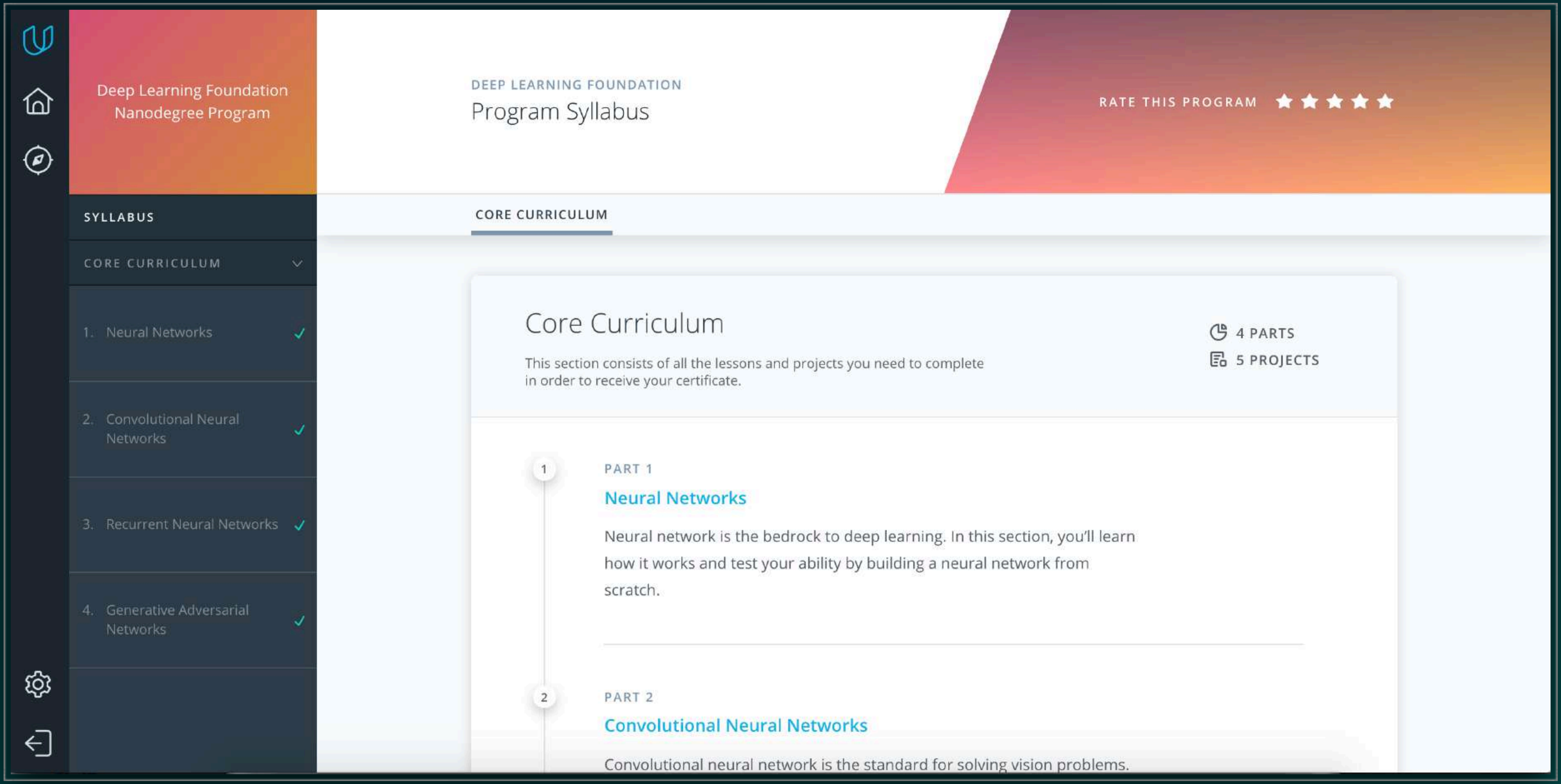
project 1
project 2
project 3
project 4
project 5

january_class
march_class
june_class
august_class
…

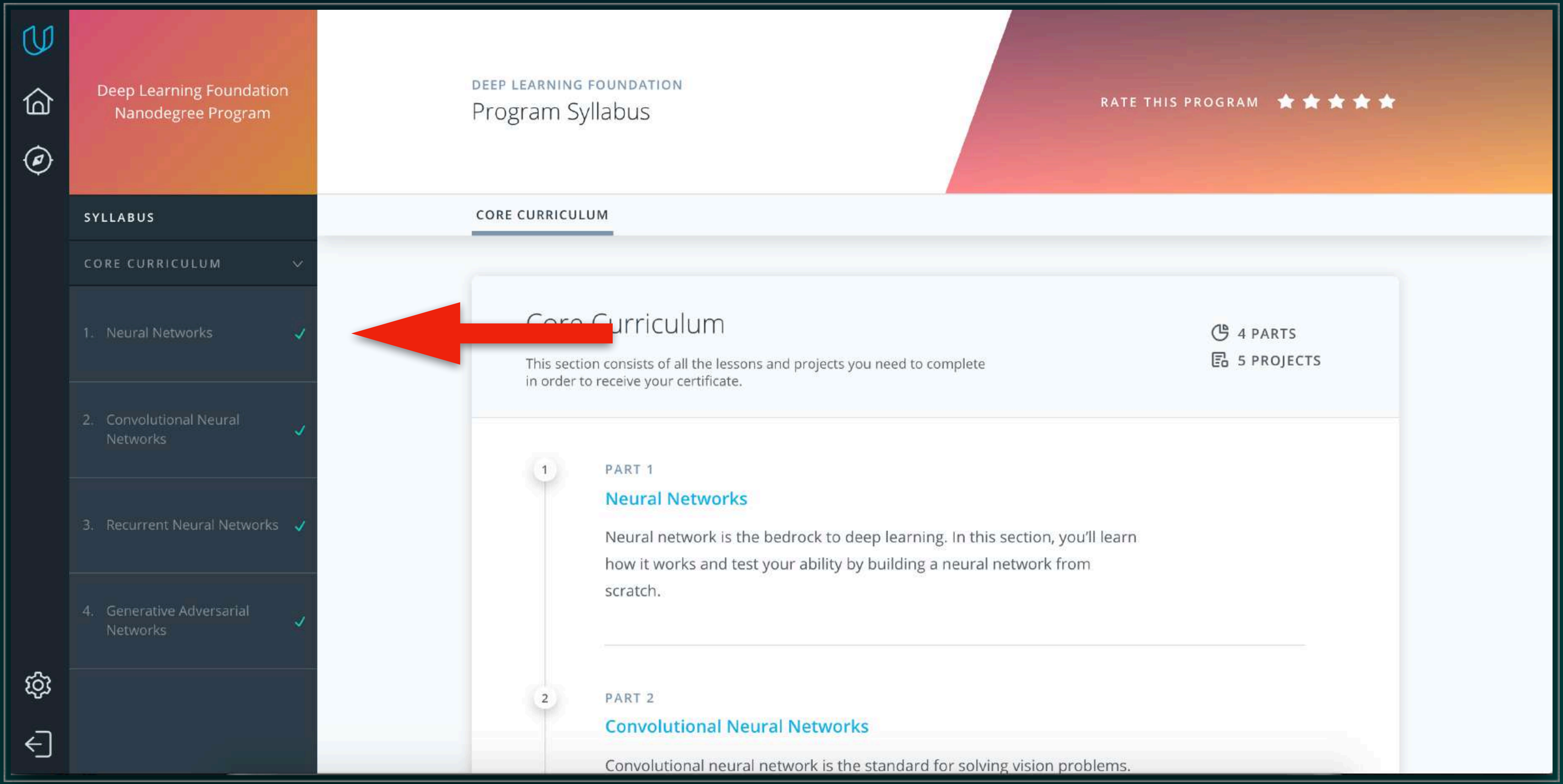
Open DLFND GitHub Repo – 22 exercises/projects



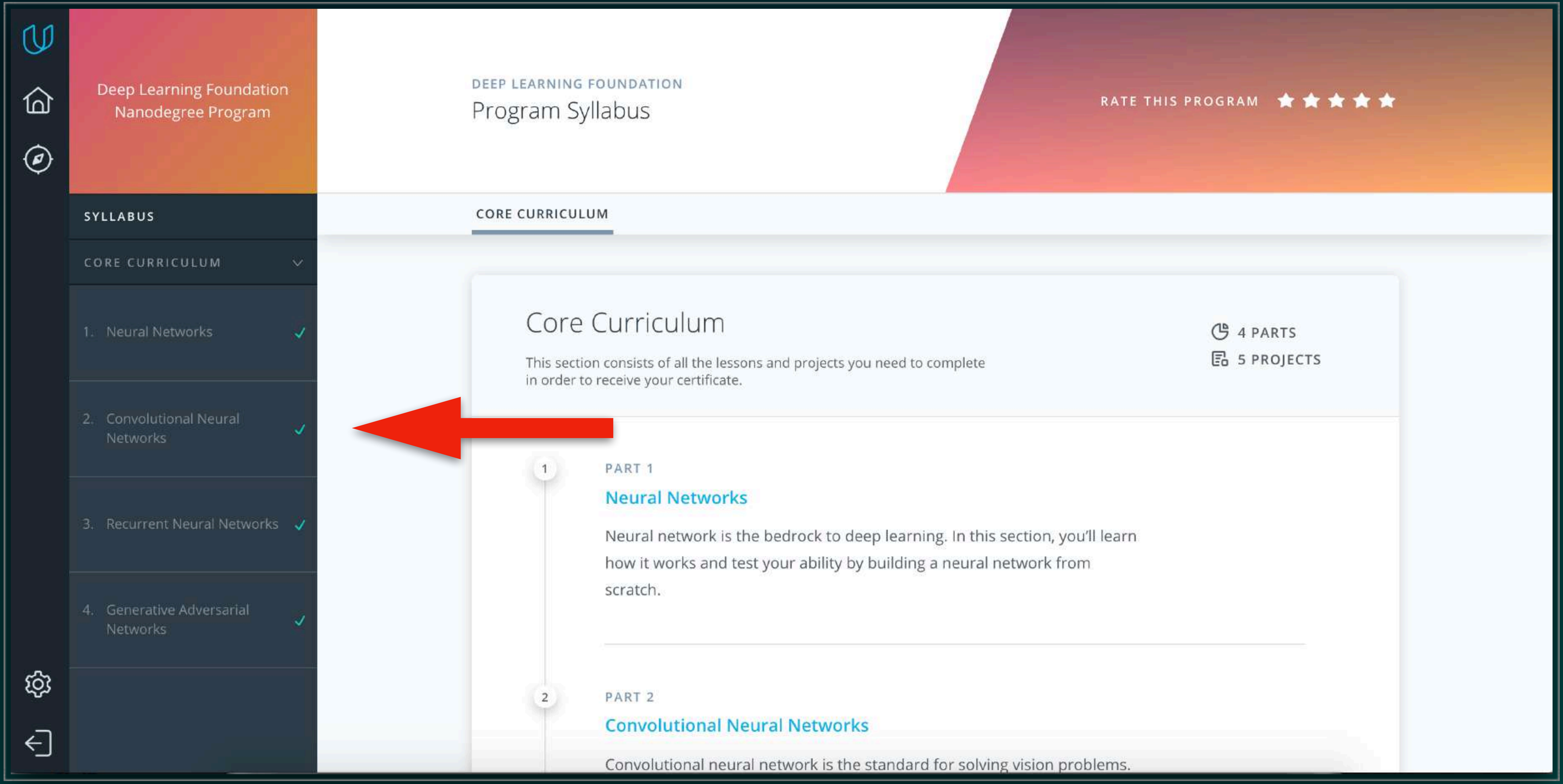
Access to Classroom



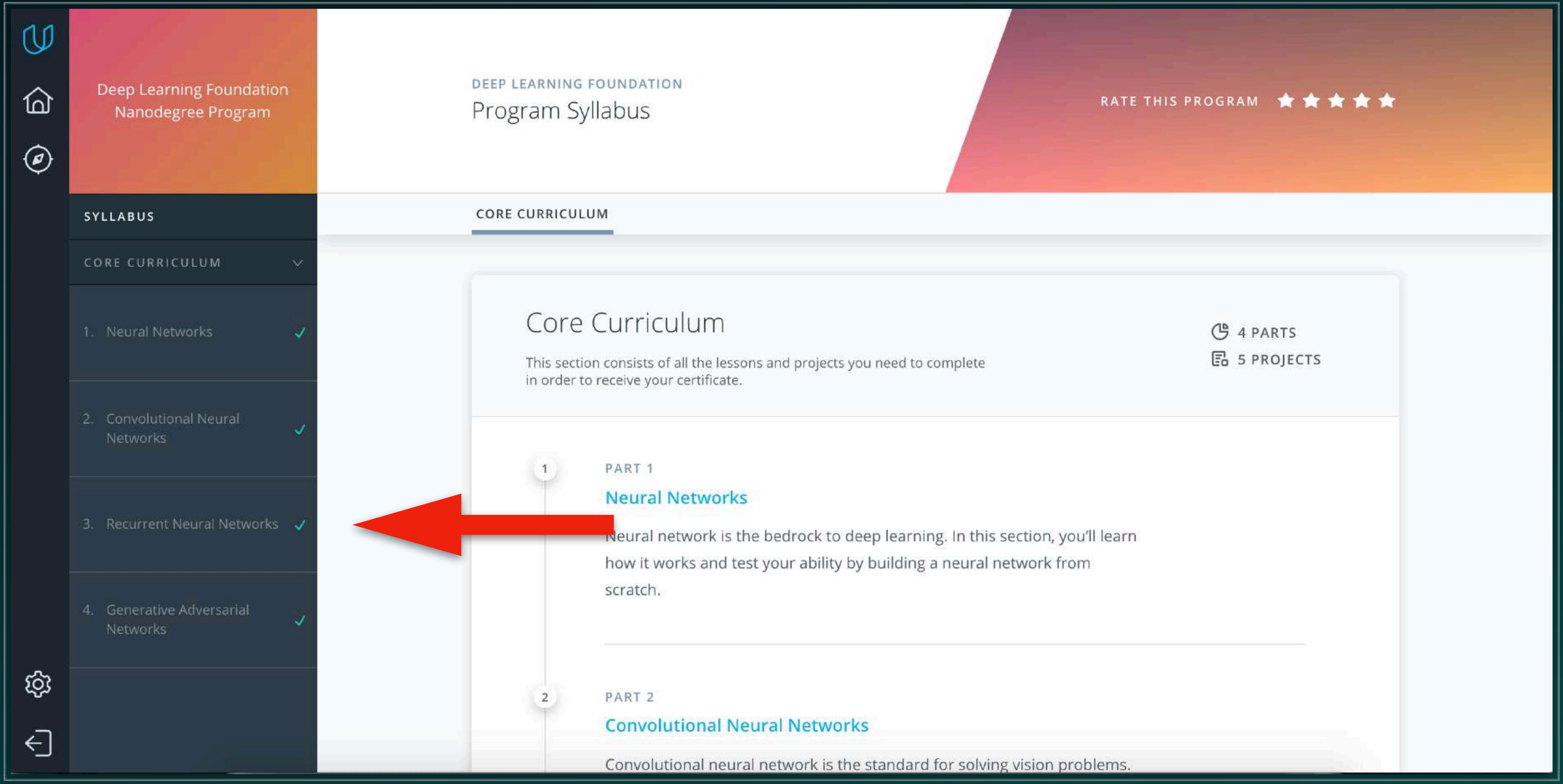
Access to Classroom



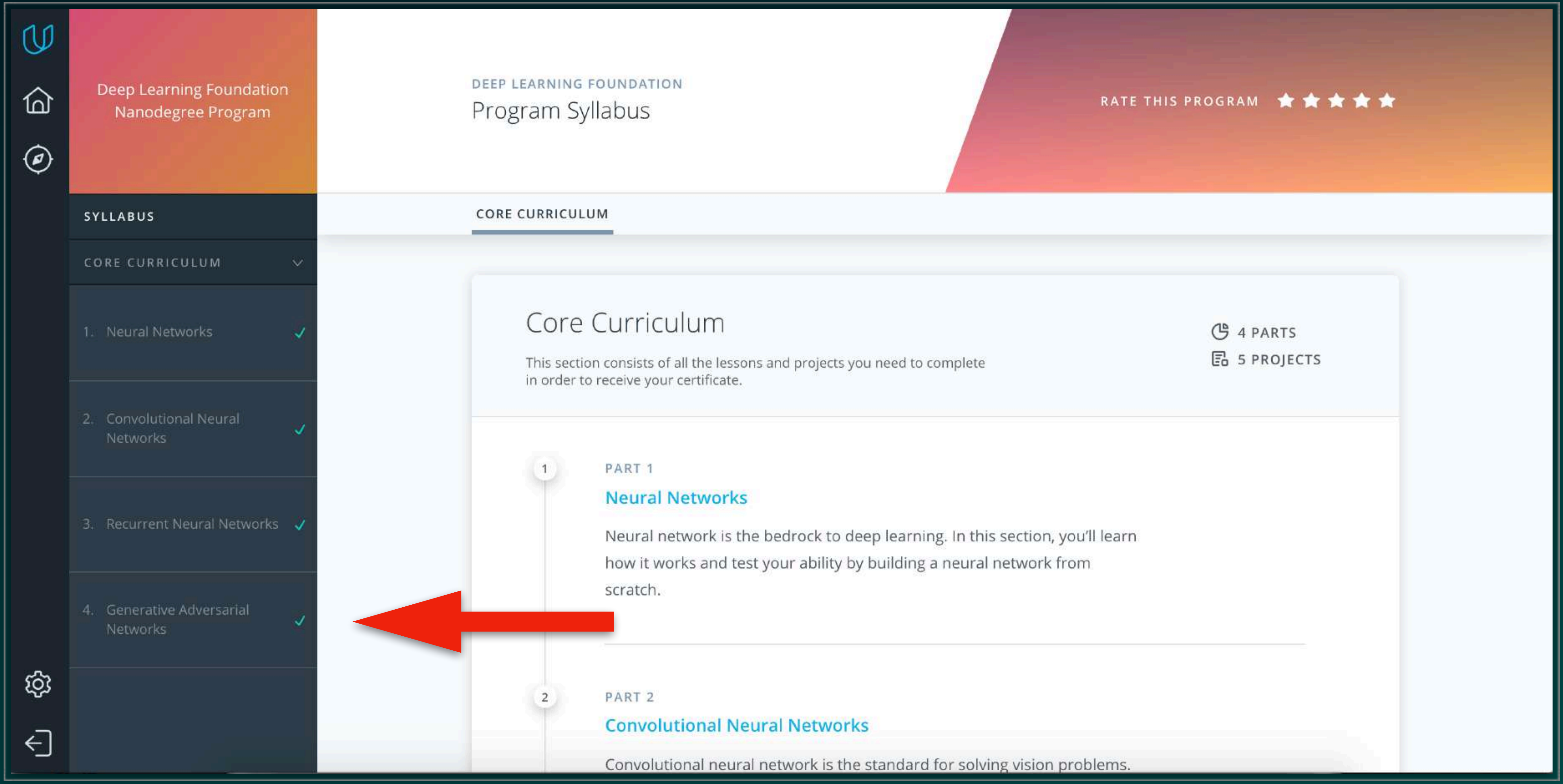
Access to Classroom



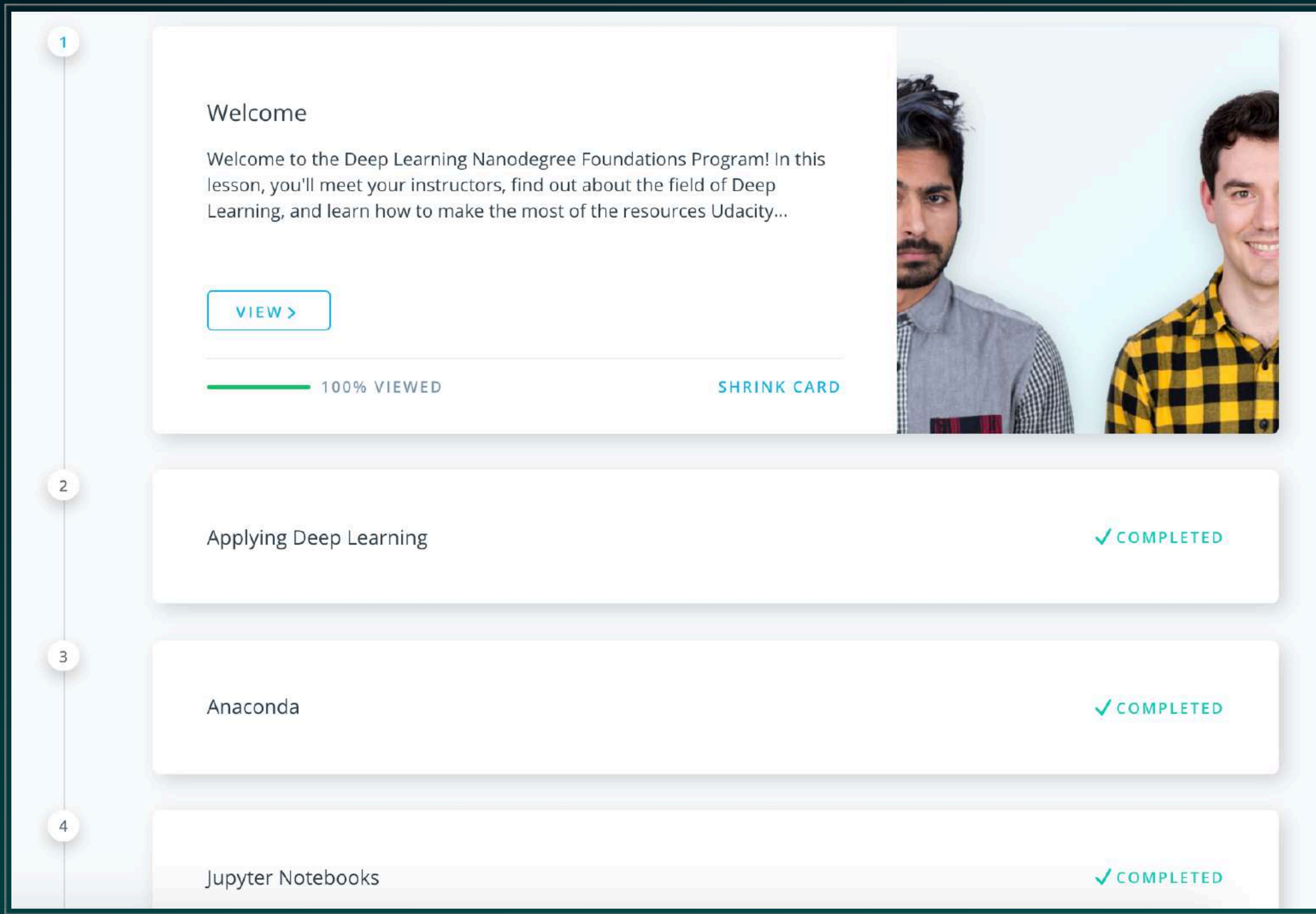
Access to Classroom



Access to Classroom



Part 1 – Neural Networks

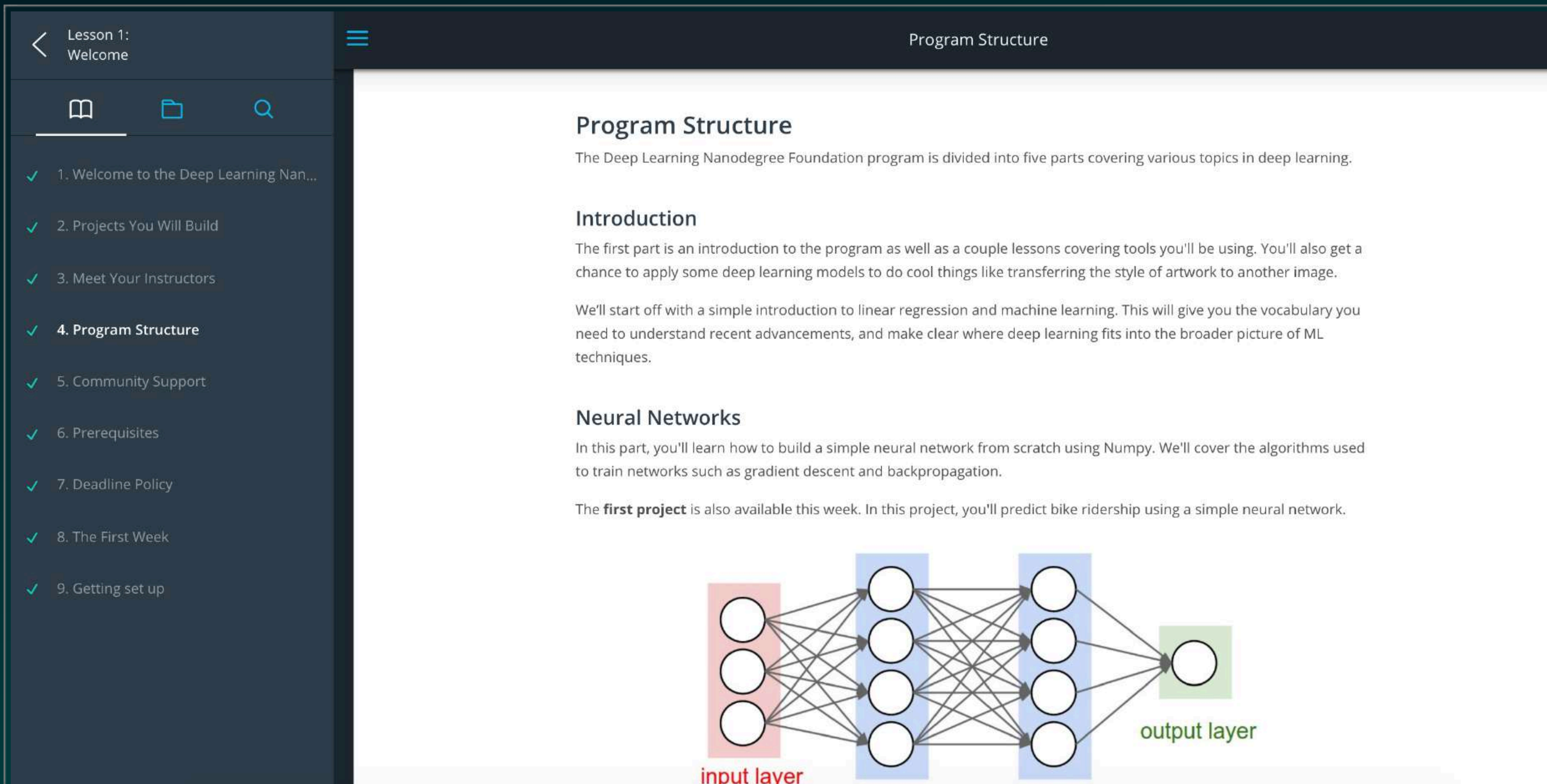


Part 1 - Neural Networks

- Welcome
- Applying Deep Learning
- Anaconda
- Jupyter Notebooks
- Regression
- Matrix Math and NumPy Refresher
- Your First Neural Network
- Intro To Neural Networks

- Welcome video
- Lecturers introduction
- General overview of course projects
- Course prerequisites (Khan Academy):
  - basic Python
  - basic Linear Algebra
- Deadlines
- Support

Part 1 – Neural Networks



Lesson 1:
Welcome

1. Welcome to the Deep Learning Nan...
2. Projects You Will Build
3. Meet Your Instructors
4. Program Structure
5. Community Support
6. Prerequisites
7. Deadline Policy
8. The First Week
9. Getting set up

Program Structure

Program Structure

The Deep Learning Nanodegree Foundation program is divided into five parts covering various topics in deep learning.

Introduction

The first part is an introduction to the program as well as a couple lessons covering tools you'll be using. You'll also get a chance to apply some deep learning models to do cool things like transferring the style of artwork to another image.

We'll start off with a simple introduction to linear regression and machine learning. This will give you the vocabulary you need to understand recent advancements, and make clear where deep learning fits into the broader picture of ML techniques.

Neural Networks

In this part, you'll learn how to build a simple neural network from scratch using Numpy. We'll cover the algorithms used to train networks such as gradient descent and backpropagation.

The **first project** is also available this week. In this project, you'll predict bike ridership using a simple neural network.

Part 1 - Neural Networks

- Welcome
- Applying Deep Learning
- Anaconda
- Jupyter Notebooks
- Regression
- Matrix Math and NumPy Refresher
- Your First Neural Network
- Intro To Neural Networks

Mini Projects:

- Fast Style Transfer
  - https://github.com/lengstrom/fast-style-transfer
- DeepTraffic
  - http://selfdrivingcars.mit.edu/deeptrafficjs/
- Flappy Bird
  - https://github.com/yenchenlin/DeepLearningFlappyBird

Part 1 - Neural Networks

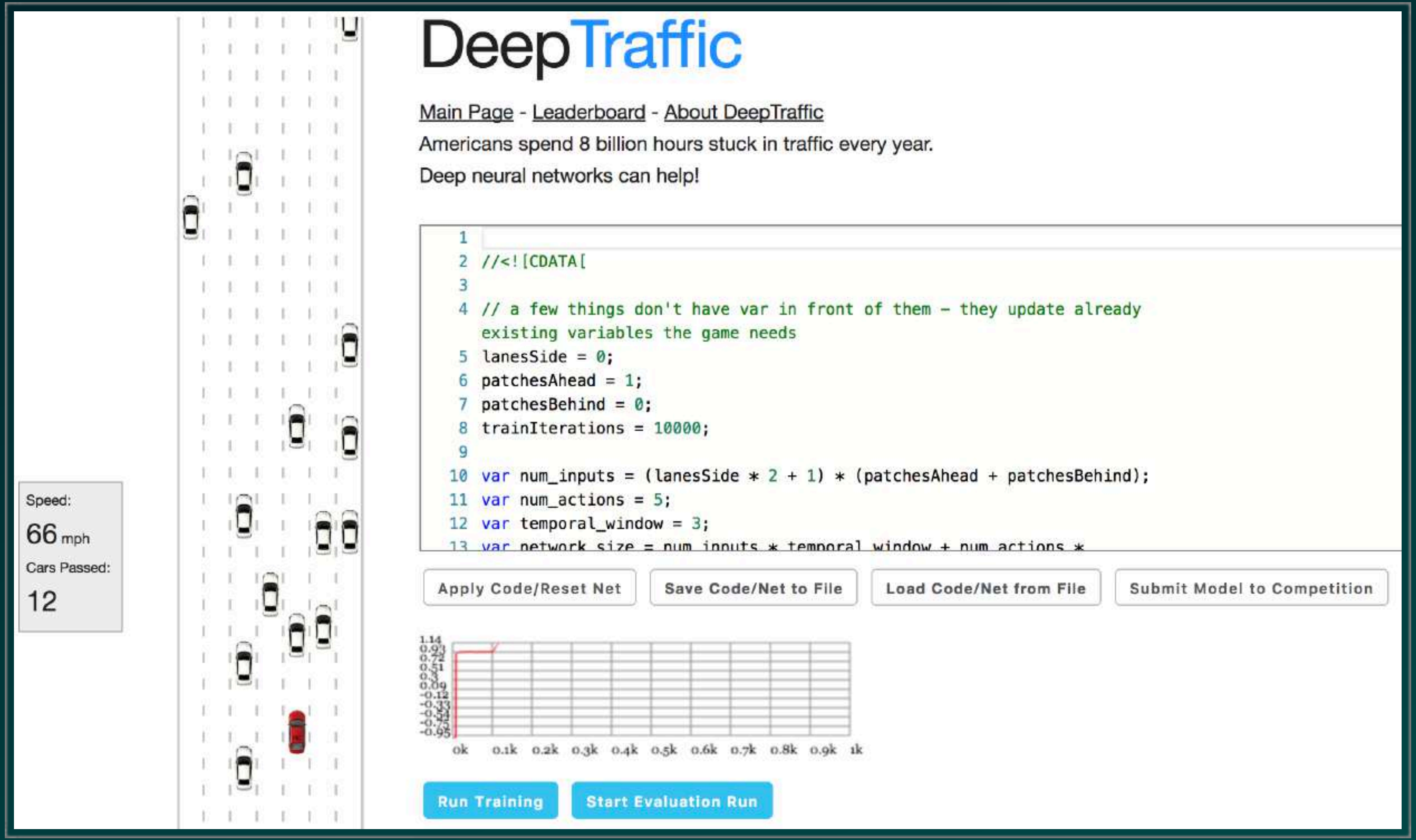
- Welcome
- **Applying Deep Learning**
- Anaconda
- Jupyter Notebooks
- Regression
- Matrix Math and NumPy Refresher
- Your First Neural Network
- Intro To Neural Networks

Part 1 – Neural Networks

- Welcome
- **Applying Deep Learning**
- Anaconda
- Jupyter Notebooks
- Regression
- Matrix Math and NumPy Refresher
- Your First Neural Network
- Intro To Neural Networks

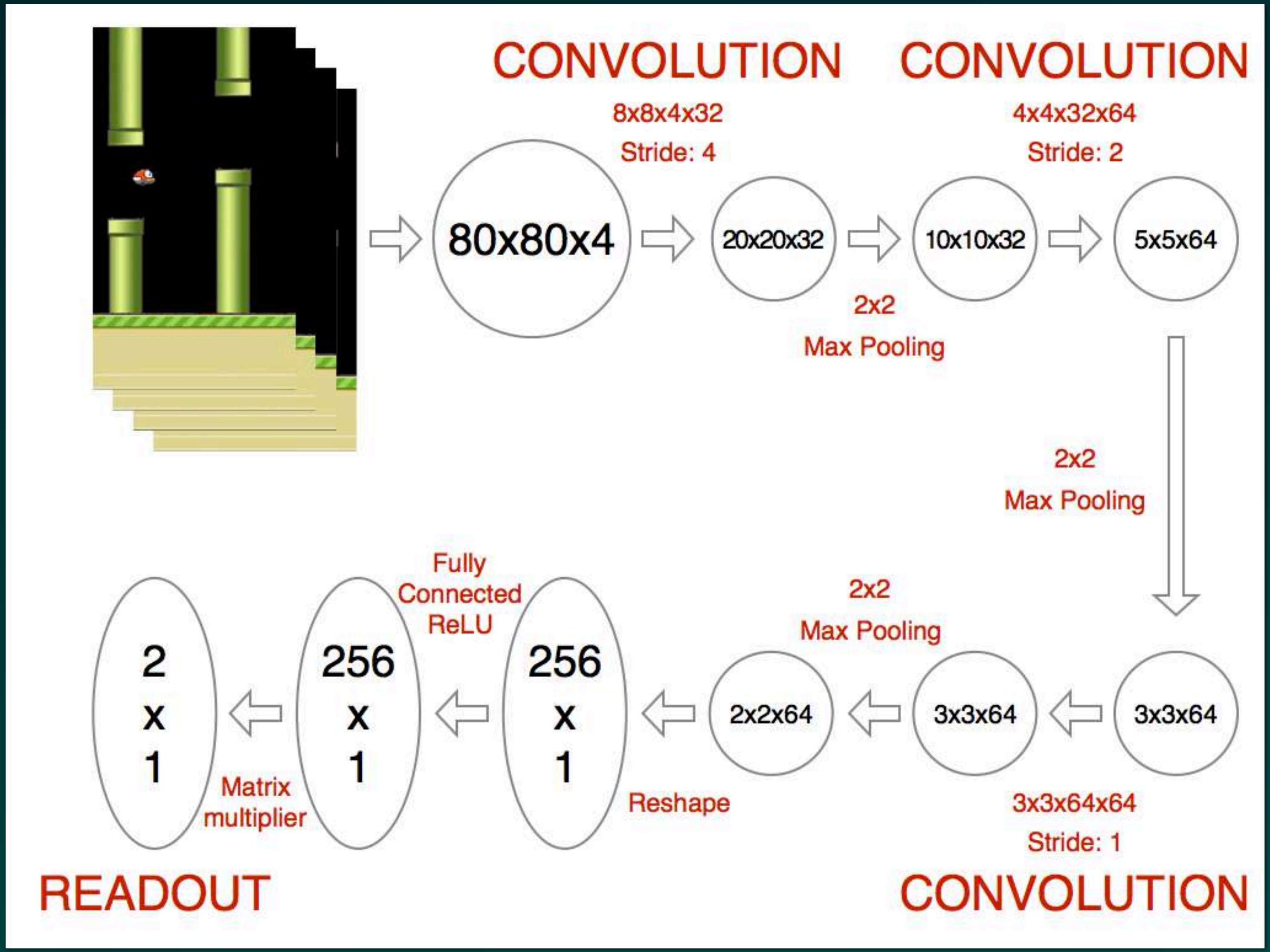
Part 1 – Neural Networks

- Welcome
- **Applying Deep Learning**
- Anaconda
- Jupyter Notebooks
- Regression
- Matrix Math and NumPy Refresher
- Your First Neural Network
- Intro To Neural Networks



Part 1 – Neural Networks

- Welcome
- **Applying Deep Learning**
- Anaconda
- Jupyter Notebooks
- Regression
- Matrix Math and NumPy Refresher
- Your First Neural Network
- Intro To Neural Networks



Part 1 - Neural Networks

- Welcome
- Applying Deep Learning
- Anaconda
- Jupyter Notebooks
- Regression
- Matrix Math and NumPy Refresher
- Your First Neural Network
- Intro To Neural Networks

Recommended books:

- Grokking Deep Learning by Andrew Trask
- Neural Networks And Deep Learning by Michael Nielsen
- The Deep Learning Textbook from Ian Goodfellow, Yoshua Bengio, and Aaron Courville

Part 1 - Neural Networks

ANACONDA®

Framework with Python/R libraries for large-scale data processing, predictive analytics, and scientific computing. Possible alternative for virtualenv.

Part 1 – Neural Networks

- Welcome
- Applying Deep Learning
- Anaconda
- **Jupyter Notebooks**
- Regression
- Matrix Math and NumPy Refresher
- Your First Neural Network
- Intro To Neural Networks

The Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations and narrative text.

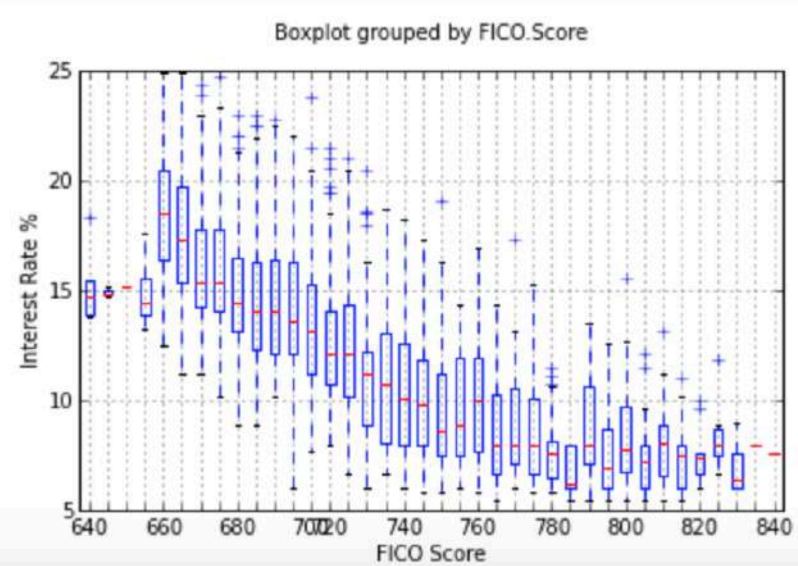
Part 1 – Neural Networks

- Welcome
- Applying Deep Learning
- Anaconda
- **Jupyter Notebooks**
- Regression
- Matrix Math and NumPy Refresher
- Your First Neural Network
- Intro To Neural Networks

Box Plot

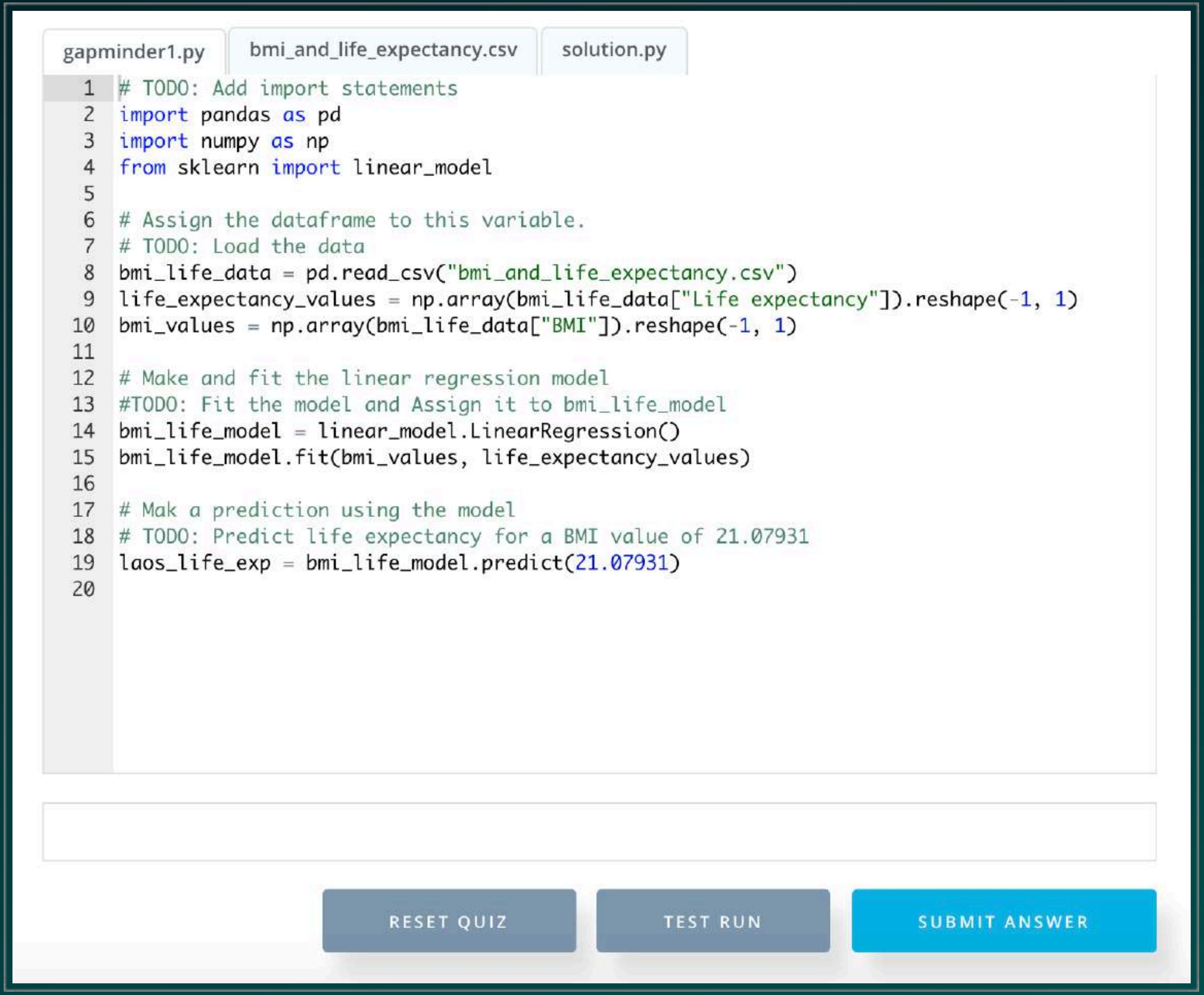
Next we take a box plot which allows us to quickly look at the distribution of interest rates based on each FICO score range.

```
import matplotlib.pyplot as plt
import pandas as pd
plt.figure()
loansmin = pd.read_csv('../datasets/loanf.csv')

p = loansmin.boxplot('Interest.Rate','FICO.Score')
q = p.set_xticklabels(['640','','','','660','','','','680','','','','700',
  '720','','','','740','','','','760','','','','780','','','','800','','','','820','','','',

q0 = p.set_xlabel('FICO Score')
q1 = p.set_ylabel('Interest Rate %')
q2 = p.set_title('                ')
```

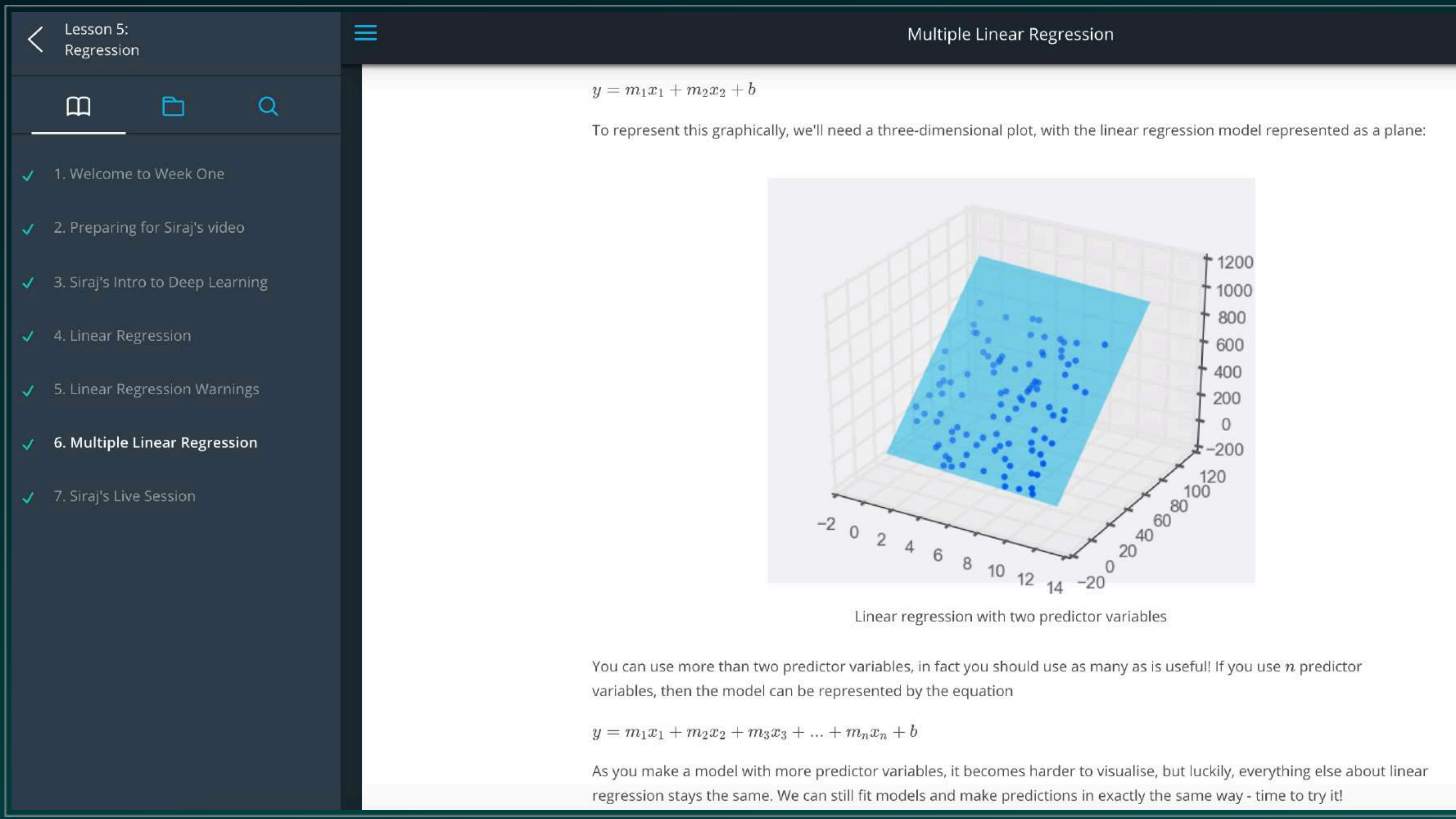


Part 1 – Neural Networks

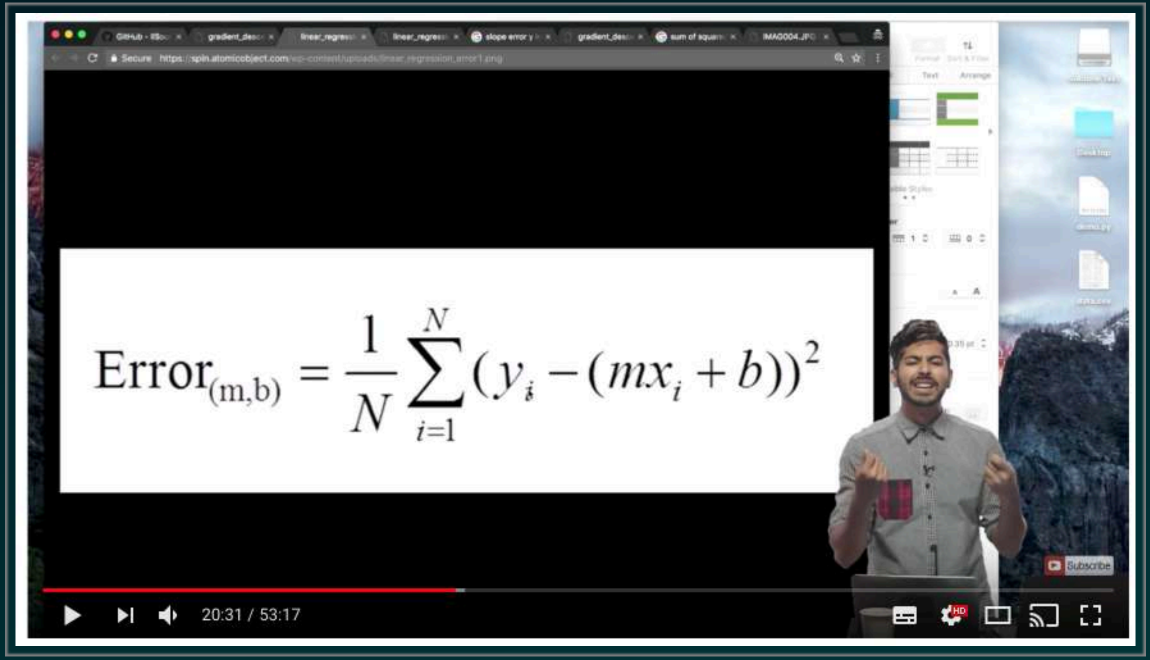
- Welcome
- Applying Deep Learning
- Anaconda
- Jupyter Notebooks
- **Regression**
- Matrix Math and NumPy Refresher
- Your First Neural Network
- Intro To Neural Networks

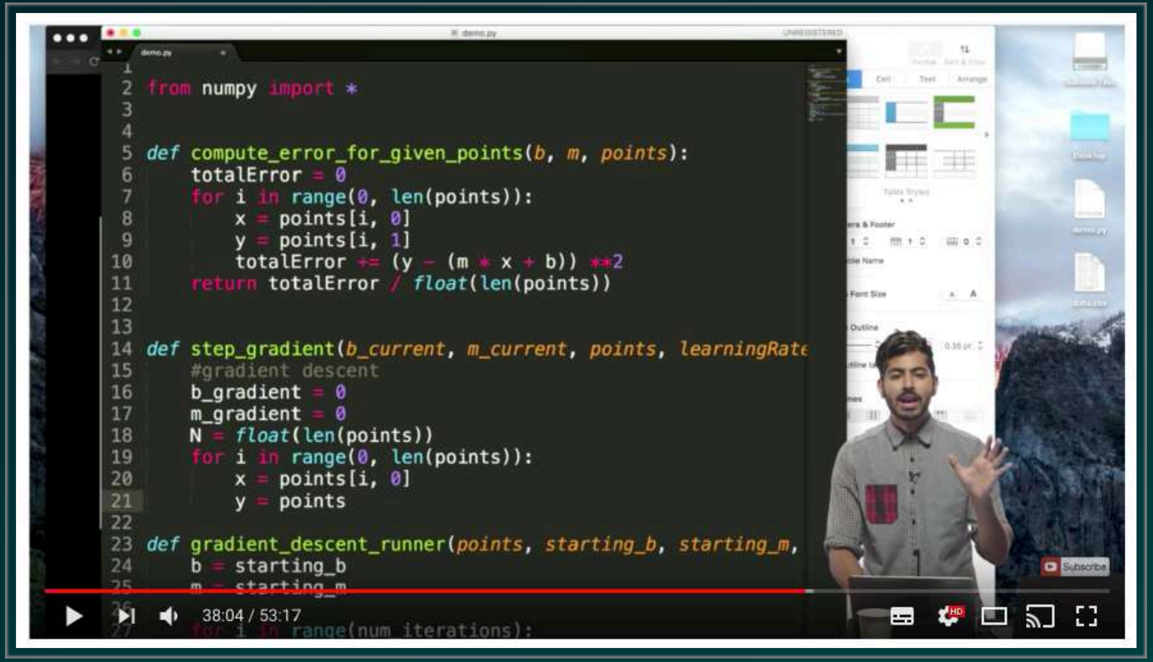


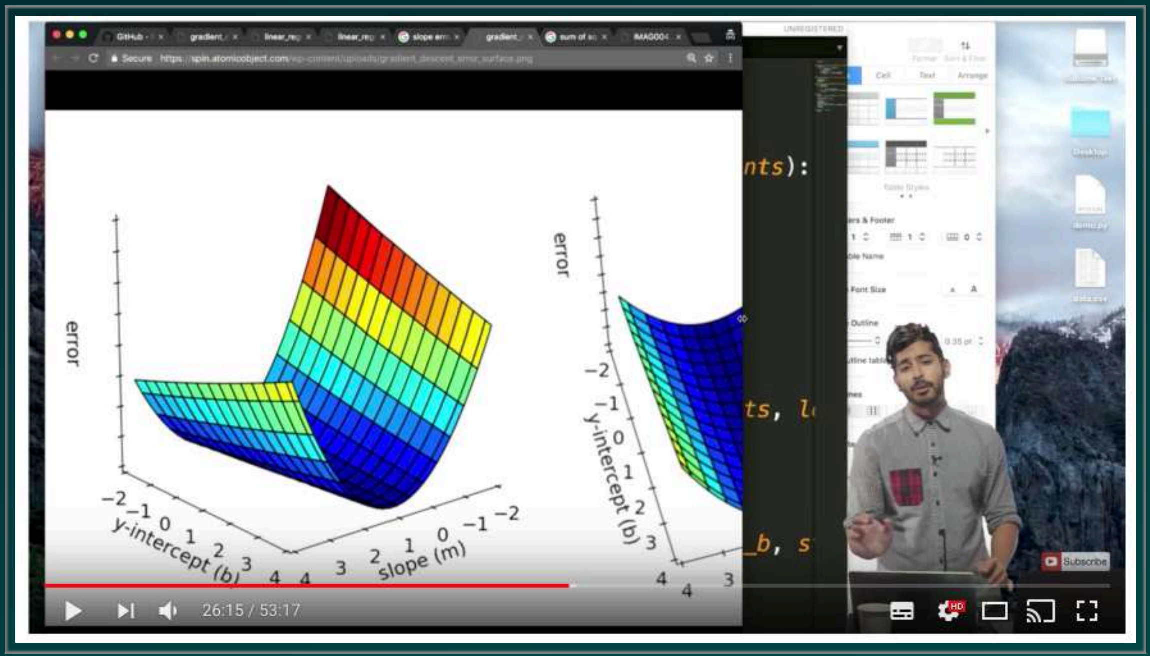
Part 1 – Neural Networks

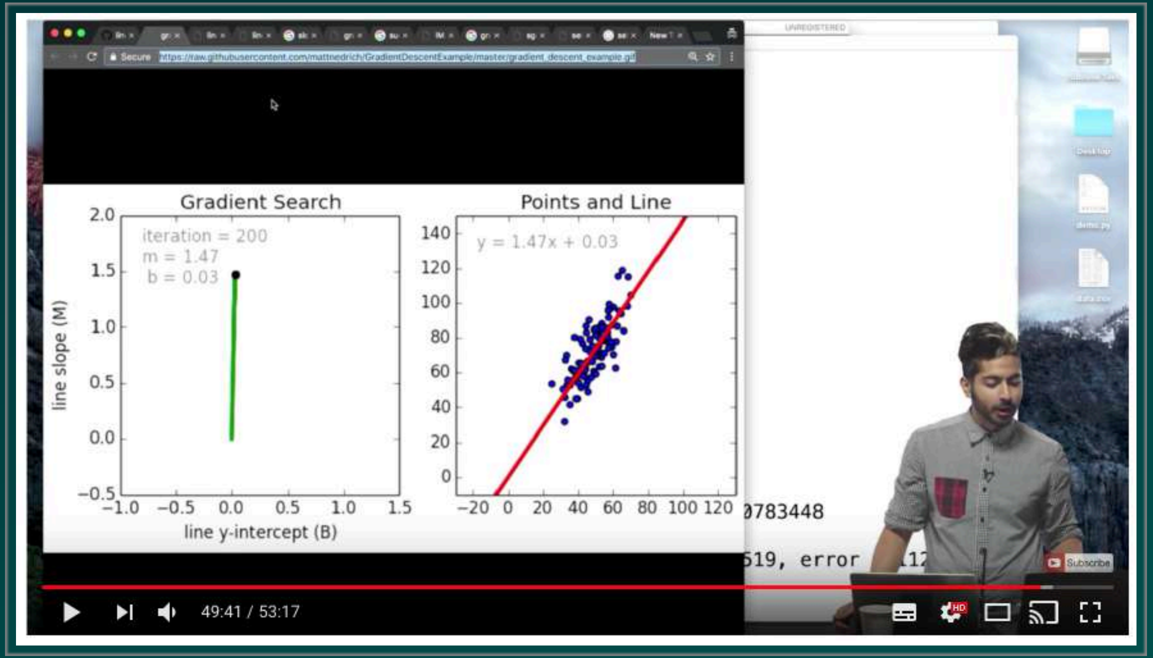


Part 1 – Neural Networks









Part 1 - Neural Networks

- Data dimensionality in general
- Representation of data dimensionality in NumPy and Tensorflow - from scalar to tensor
- Review of Matrix Multiplication algorithm
- Mathematical operations on matrices - comparison between raw Python and NumPy
- Matrix transpose

Part 1 - Neural Networks

- Learn Logistic Regression
- Understand how to calculate Loss for classification problem
- Understand Perceptron and implement AND. OR, XOR
- Introduction to Activation Function - Sigmoid
- Role of bias
- Understand Gradient Descent - equations, geometrical definition, code implementation
- Multiplayer Perceptron explained
- Understand and implement Backpropagation

Part 1 - Neural Networks

- Welcome
- Applying Deep Learning
- Anaconda
- Jupyter Notebooks
- Regression
- Matrix Math and NumPy Refresher
- Intro To Neural Networks
- **Your First Neural Network**

Project no. 1

Deadline: 4 weeks

You will build whole Neural Network from scratch in raw Python.

You will solve regression problem - where you will want to predict number of bikes that should be available in bikeshare store based on historical data.

You are given rubric with requirements your project has to meet in order to pass.

Project undergoes code review by specialist and you receive feedback with links/hints what was done well or could be done better.

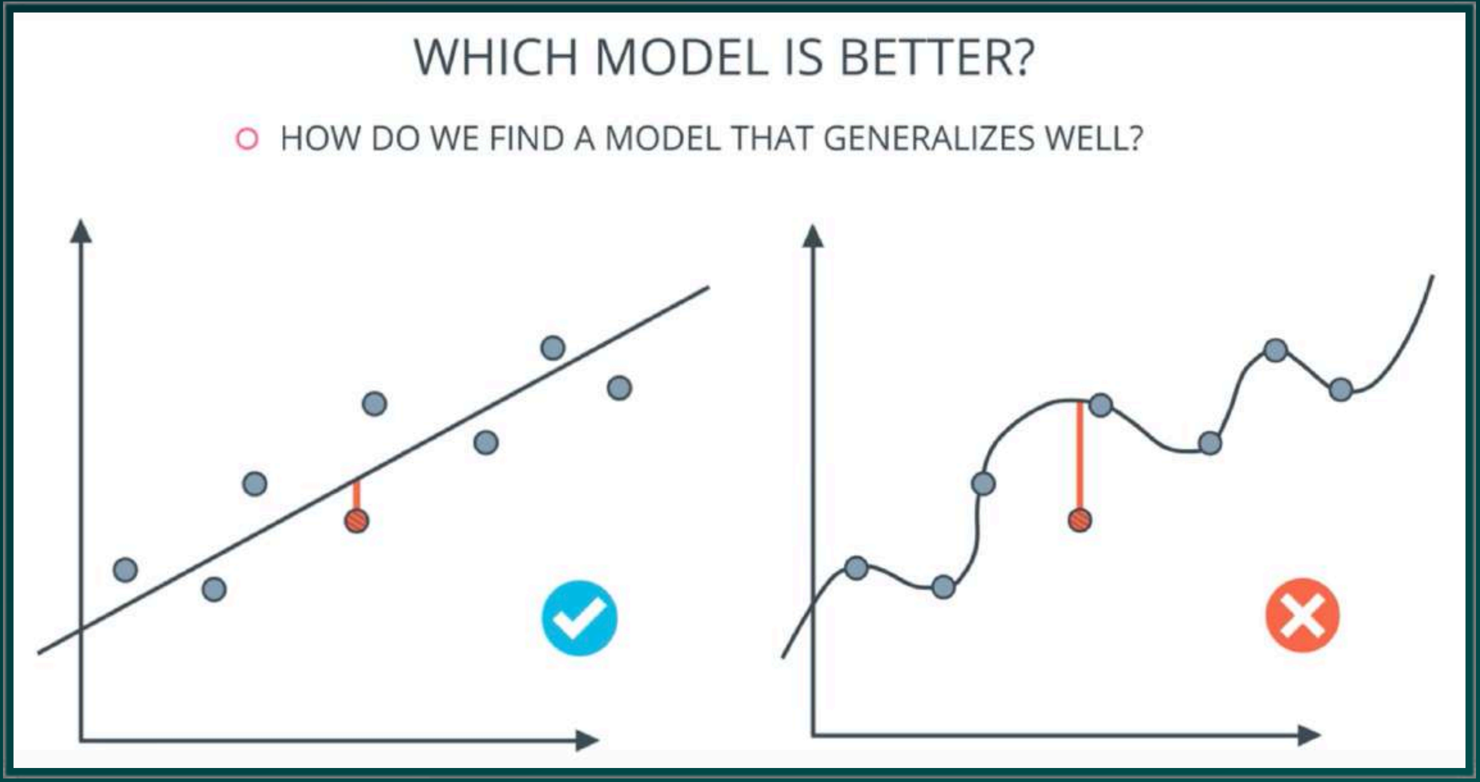
Part 1 - Neural Networks

Link to my project

Part 2 – Convolutional Neural Networks

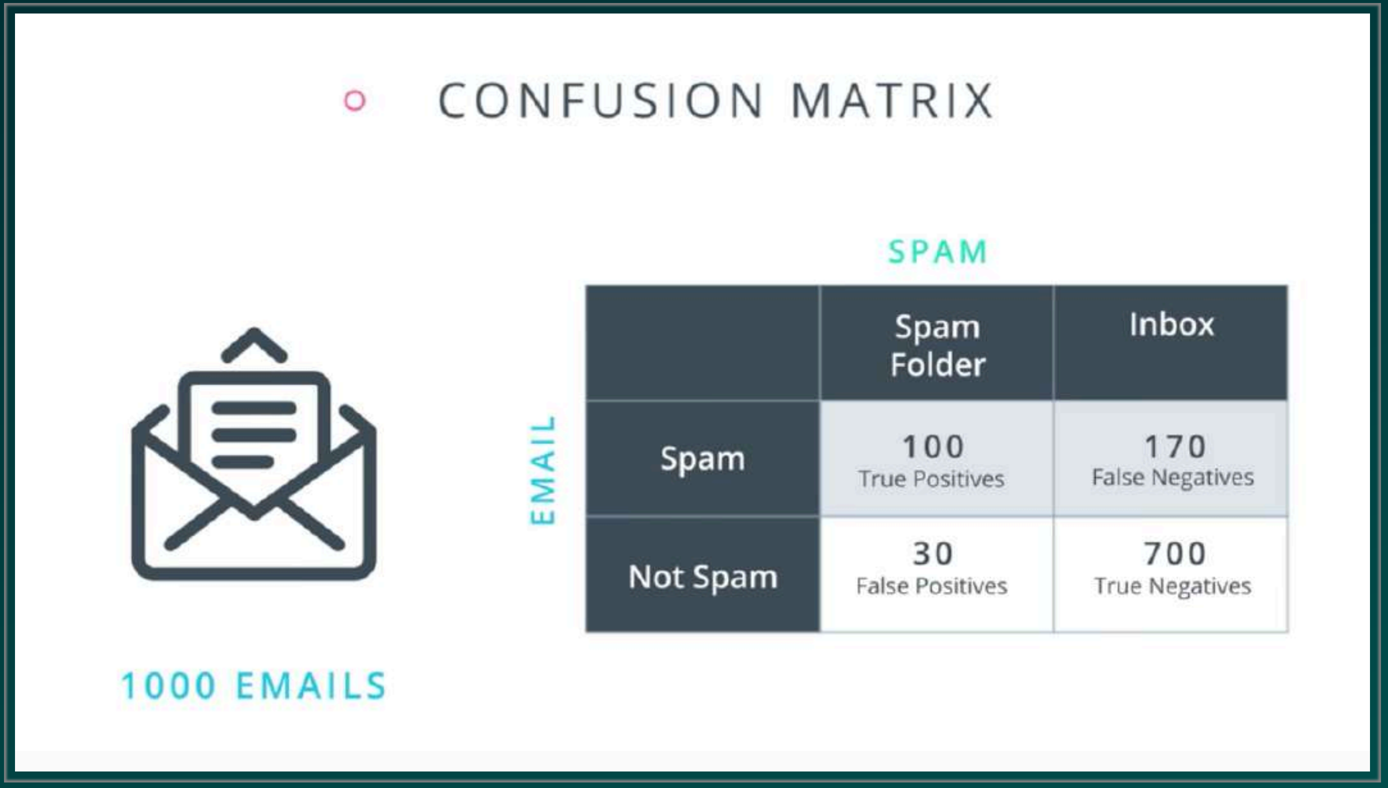
- **Model Evaluation and Validation**
- MiniFlow
- Sentiment Analysis with Andrew Trask
- Intro to TensorFlow
- Cloud Computing
- Deep Neural Networks
- Convolutional Networks
- Siraj's Image Classification
- Weight Initialization
- …

- Explaining when model is overfitting or under-fitting.
- How to split data for testing into "train", "validation" and "test" sets.
- K-Fold Cross Validation explained



Part 2 – Convolutional Neural Networks

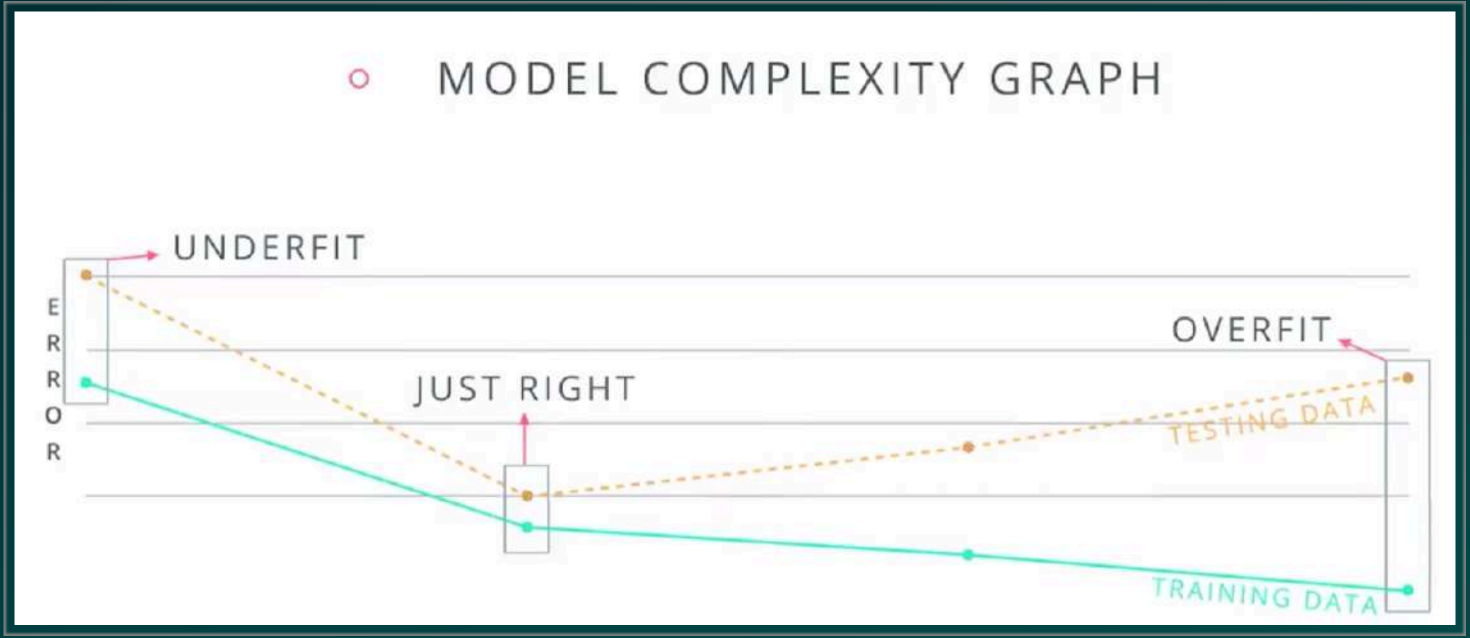
- You will learn how to assess model quality by creating Confusion Matrix, calculating accuracy or using R2 Score.
- How to split data for testing into "train", "validation" and "test" sets.



Part 2 – Convolutional Neural Networks

- Model Evaluation and Validation
- MiniFlow
- Sentiment Analysis with Andrew Trask
- Intro to TensorFlow
- Cloud Computing
- Deep Neural Networks
- Convolutional Networks
- Siraj's Image Classification
- Weight Initialization
- …

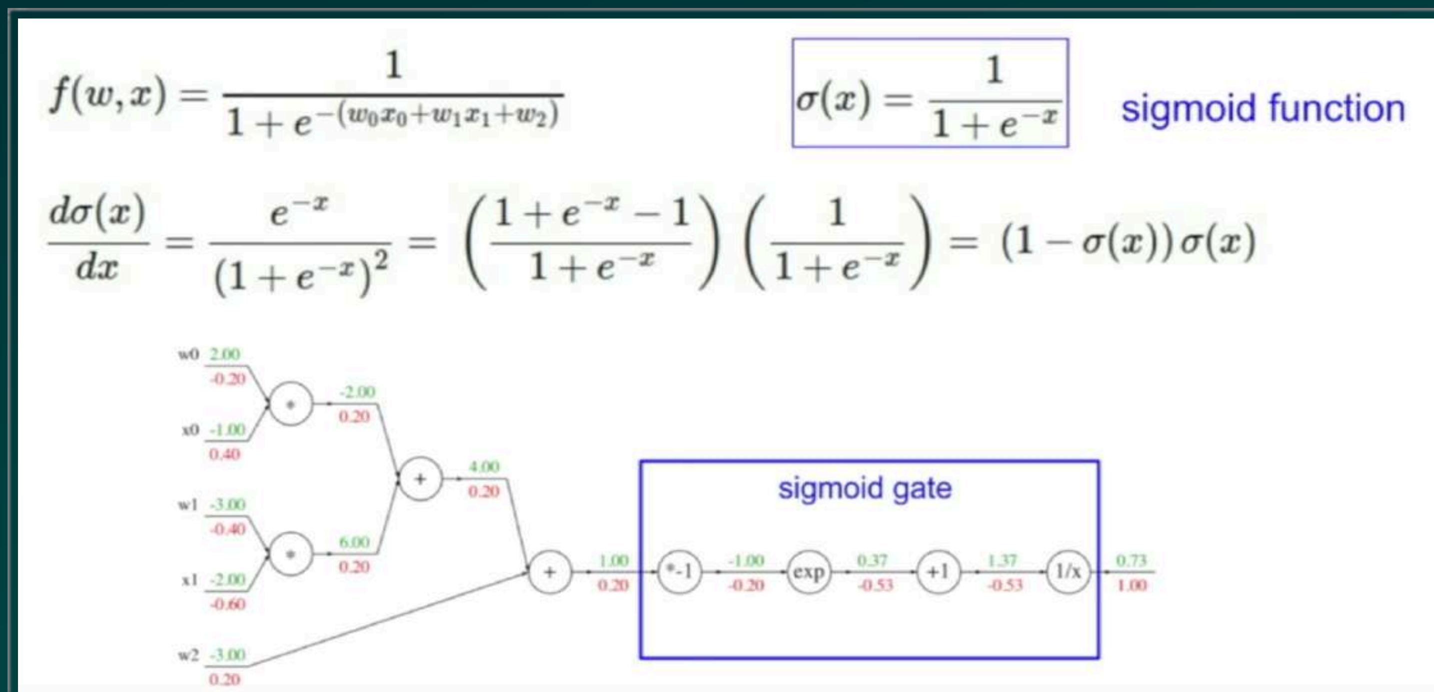
- Explained role of loss function (Mean Squared Errors) in model performance monitoring.
- How error values should behave during training.



Part 2 – Convolutional Neural Networks

- Model Evaluation and Validation
- **MiniFlow**
- Sentiment Analysis with Andrew Trask
- Intro to TensorFlow
- Cloud Computing
- Deep Neural Networks
- Convolutional Networks
- Siraj's Image Classification
- Weight Initialization
- …

- Miniflow is an training exercise for understanding how gradient flow in Neural Network.
- It is explained in course CS231n at Stanford University, lecture 4.



Part 2 - Convolutional Neural Networks

- You build model out of nodes which are implemented from scratch Python objects.
- Each node has list of incoming nodes that from which it takes values
- Each node has list of outcoming nodes to which it sends values.
- Each node sends data forward by performing specific mathematical operation e.g. multiplication or subtraction. This value is stored in node afterwards.
- Each implements backward method where derivative of mathematical operation in forward method is calculated - which is gradient

Part 2 – Convolutional Neural Networks

- Model Evaluation and Validation
- **MiniFlow**
- Sentiment Analysis with Andrew Trask
- Intro to TensorFlow
- Cloud Computing
- Deep Neural Networks
- Convolutional Networks
- Siraj's Image Classification
- Weight Initialization
- ...

```python
class Node(object):
    def __init__(self, inbound_nodes=[]):
        # Nodes from which this node receives values
        self.inbound_nodes = inbound_nodes

        # Nodes to which this note passes values
        self.outbound_nodes = []

        # List of nodes to which this node is connected
        for n in self.inbound_nodes:
            n.outbound_nodes.append(self)

        self.value = None
        self.gradient = None

    def forward(self):
        # Calculate and send data to outbound nodes
        raise NotImplemented

    def backward(self):
        # Calculate gradient by taking derivative
        # of forward function
        raise NotImplemented
```

Part 2 - Convolutional Neural Networks

- Model Evaluation and Validation
- MiniFlow
- **Sentiment Analysis with Andrew Trask**
- Intro to TensorFlow
- Cloud Computing
- Deep Neural Networks
- Convolutional Networks
- Siraj's Image Classification
- Weight Initialization
- ...

You will implement Neural Network in Raw Python which will be able to tell if given movie review is positive or negative.

IMdB reviews will be used in this exercise.

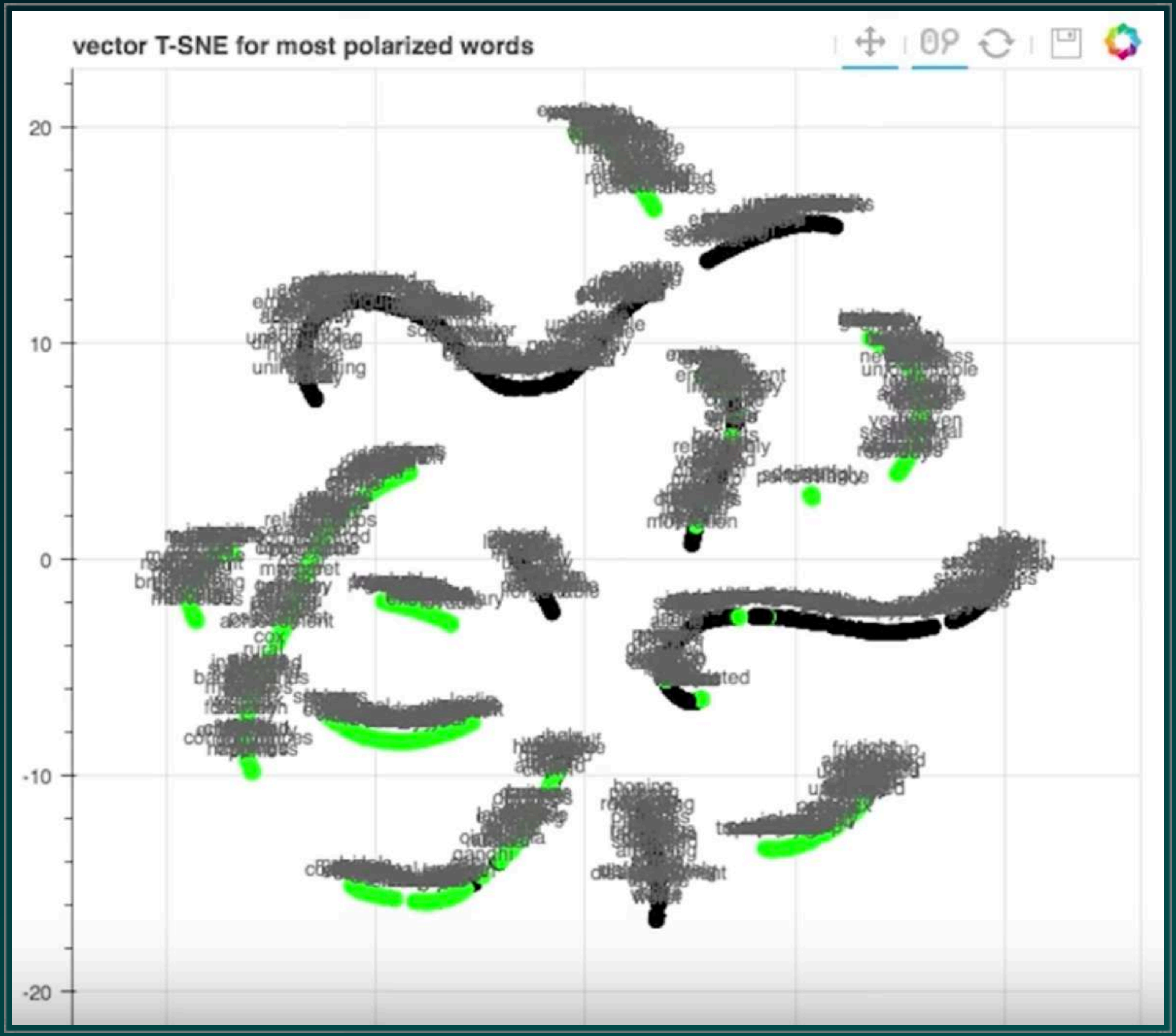
You will learn how to:

- clean data (remove noise) for NLP problems
- implement Bag of Words NLP technique
- monitor Neural Network computation speed
- optimise Neural Network computation speed
- display word polarisation

You will receive small mini-project from Andrew Track and implement in Jupiter Notebook given by him.

Part 2 - Convolutional Neural Networks

- Model Evaluation and Validation
- MiniFlow
- **Sentiment Analysis with Andrew Trask**
- Intro to TensorFlow
- Cloud Computing
- Deep Neural Networks
- Convolutional Networks
- Siraj's Image Classification
- Weight Initialization
- ...



Part 2 – Convolutional Neural Networks

Lesson with Vincent Vanhouke, Principal Scientist at Google Brain.

You will learn and implement:

- what is Deep Learning, what is it used for, what accelerated it's usefulness
- how to install Tensorflow
- write your first Hello World code
- how to create and initialize variables like tf.constants, tf.placeholders
- how to perform mathematical operations
- what is Tensorflow session and how to run it

Part 2 - Convolutional Neural Networks

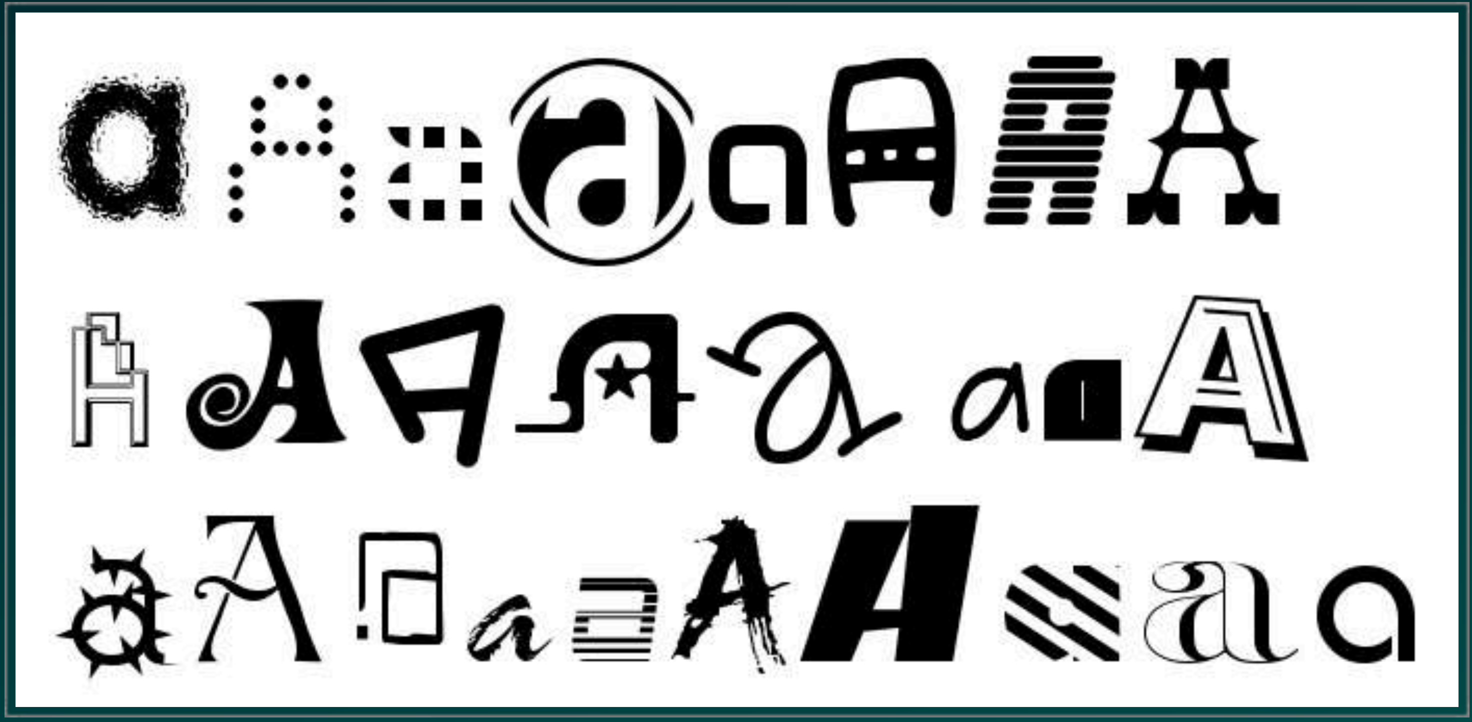
- Model Evaluation and Validation
- MiniFlow
- Sentiment Analysis with Andrew Trask
- **Intro to TensorFlow**
- Cloud Computing
- Deep Neural Networks
- Convolutional Networks
- Siraj's Image Classification
- Weight Initialization
- ...

- ReLU activation function
- how to use Neural Networks for Classification problem - Softmax activation function explained
- how to track loss of such model - introduction to Categorical Cross-Entropy cost function
- explanation of One-Hot Encoding
- general idea of how to set Neural Network weights
- explanation weights optimisation with usage of ADAGRAD (momentum, learning rate decay explained)
- explanation of Stochastic Gradient Descent
- introduction to all hyperparameters

Part 2 - Convolutional Neural Networks

- Model Evaluation and Validation
- MiniFlow
- Sentiment Analysis with Andrew Trask
- **Intro to TensorFlow**
- Cloud Computing
- Deep Neural Networks
- Convolutional Networks
- Siraj's Image Classification
- Weight Initialization
- ...

You will build Tensorflow Neural Network to solve notMNIST dataset:



Part 2 – Convolutional Neural Networks

- Model Evaluation and Validation
- MiniFlow
- Sentiment Analysis with Andrew Trask
- Intro to TensorFlow
- **Cloud Computing**
- Deep Neural Networks
- Convolutional Networks
- Siraj's Image Classification
- Weight Initialization
- …

In this lesson you will learn how to use AWS and FloydHub. Cloud computing platforms where you can rent machine with specific parameters and run your code online.

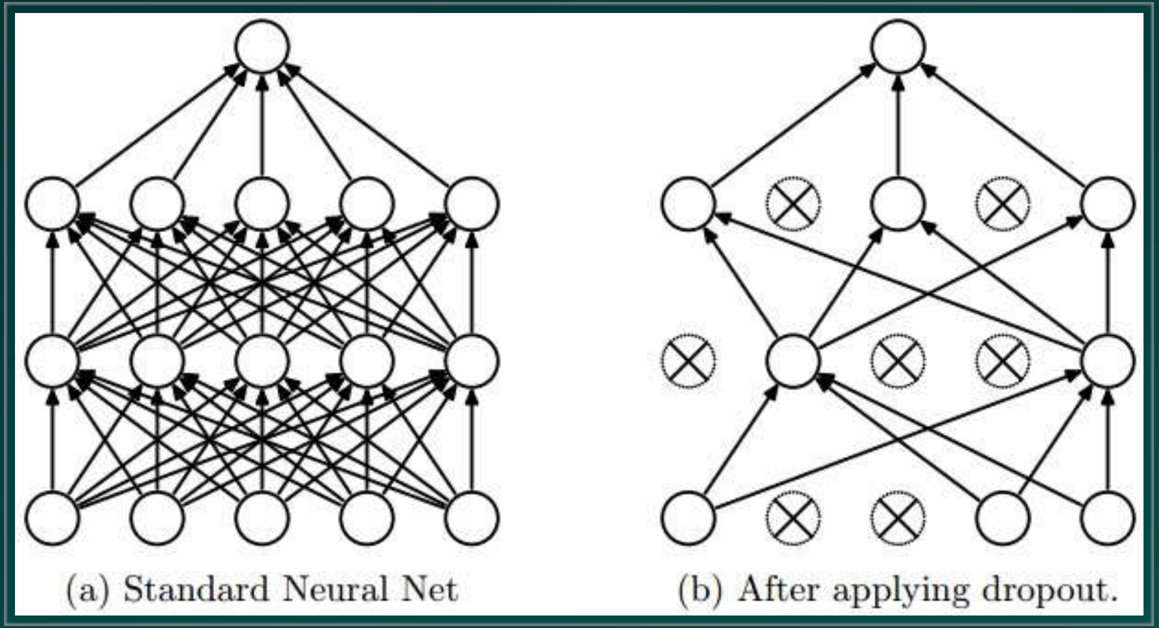
You will learn everything - from creating account, navigating through website, starting containers, sending code, launching it, properly closing machine and tracking your bill.

Every student gets 100$ present for using AWS.

Part 2 – Convolutional Neural Networks

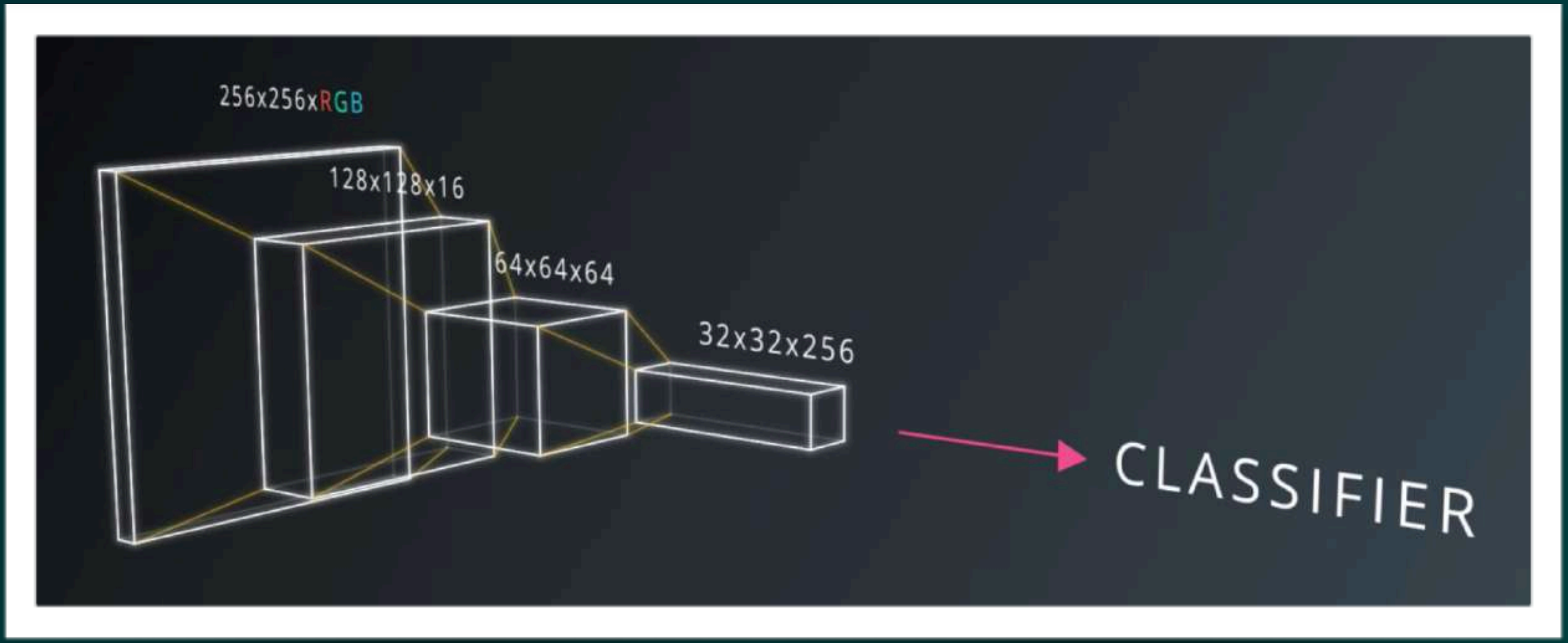
- Model Evaluation and Validation
- MiniFlow
- Sentiment Analysis with Andrew Trask
- Intro to TensorFlow
- Cloud Computing
- **Deep Neural Networks**
- Convolutional Networks
- Siraj's Image Classification
- Weight Initialization
- ...

- Learn how to build Neural Networks with more layers.
- Explanation of L2 Regularization.
- You will understand and implement Dropout by yourself.
- You will learn how to save/load Tensorflow Models.



Part 2 – Convolutional Neural Networks

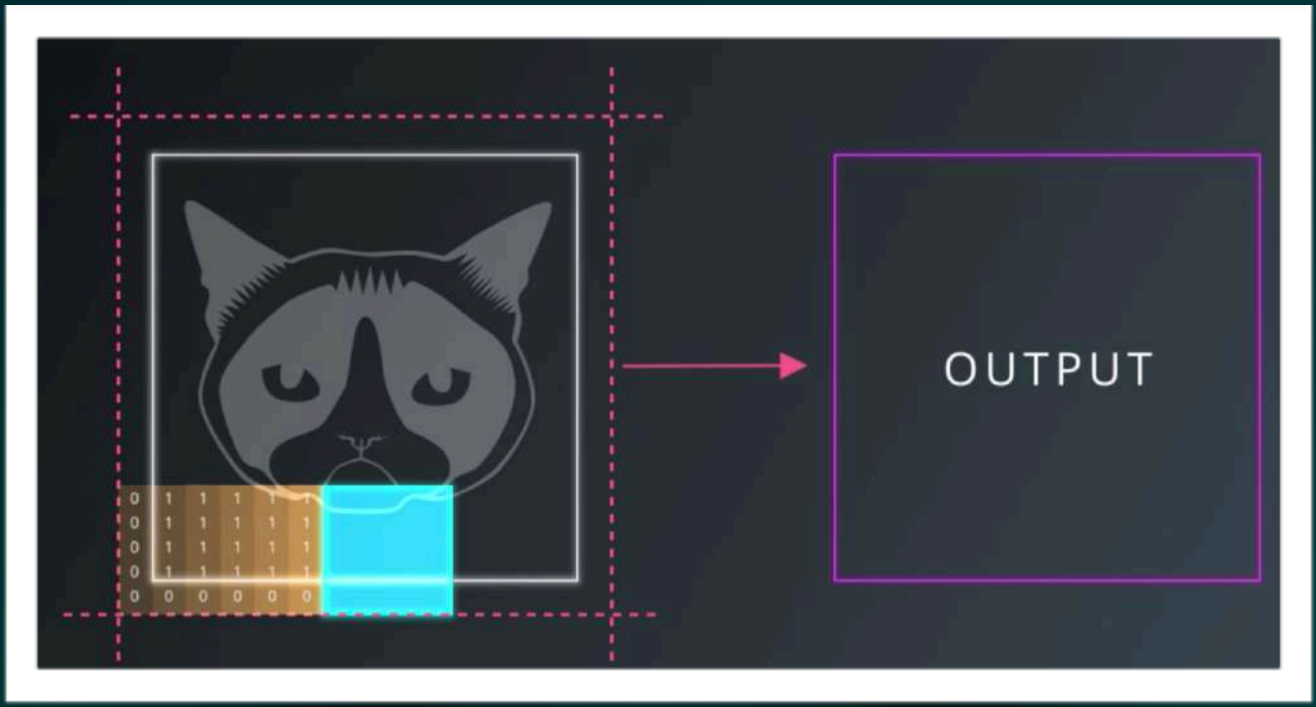
- Introduction to image classification and problems related to it
- Explanation how Convolution Neural Network works with a lot of visualisations:



Part 2 – Convolutional Neural Networks

- Model Evaluation and Validation
- MiniFlow
- Sentiment Analysis with Andrew Trask
- Intro to TensorFlow
- Cloud Computing
- Deep Neural Networks
- **Convolutional Networks**
- Siraj's Image Classification
- Weight Initialization
- …

- Explanation of CNN related parameters and how they work: kernel size, number filters, stride with valid/same padding

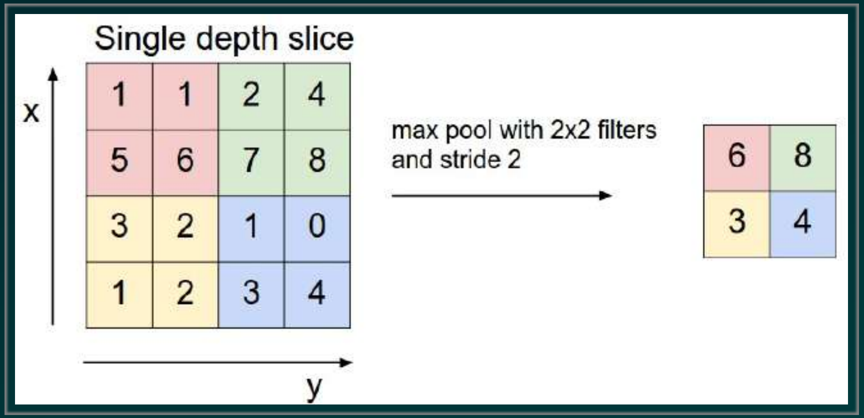


- Understanding how the output shape of each convolution looks like and how many parameters(weights) are learned in model

Part 2 – Convolutional Neural Networks

- Model Evaluation and Validation
- MiniFlow
- Sentiment Analysis with Andrew Trask
- Intro to TensorFlow
- Cloud Computing
- Deep Neural Networks
- **Convolutional Networks**
- Siraj's Image Classification
- Weight Initialization
- …

- Introduction to max/average pooling
- How to implement CNN in Tensorflow

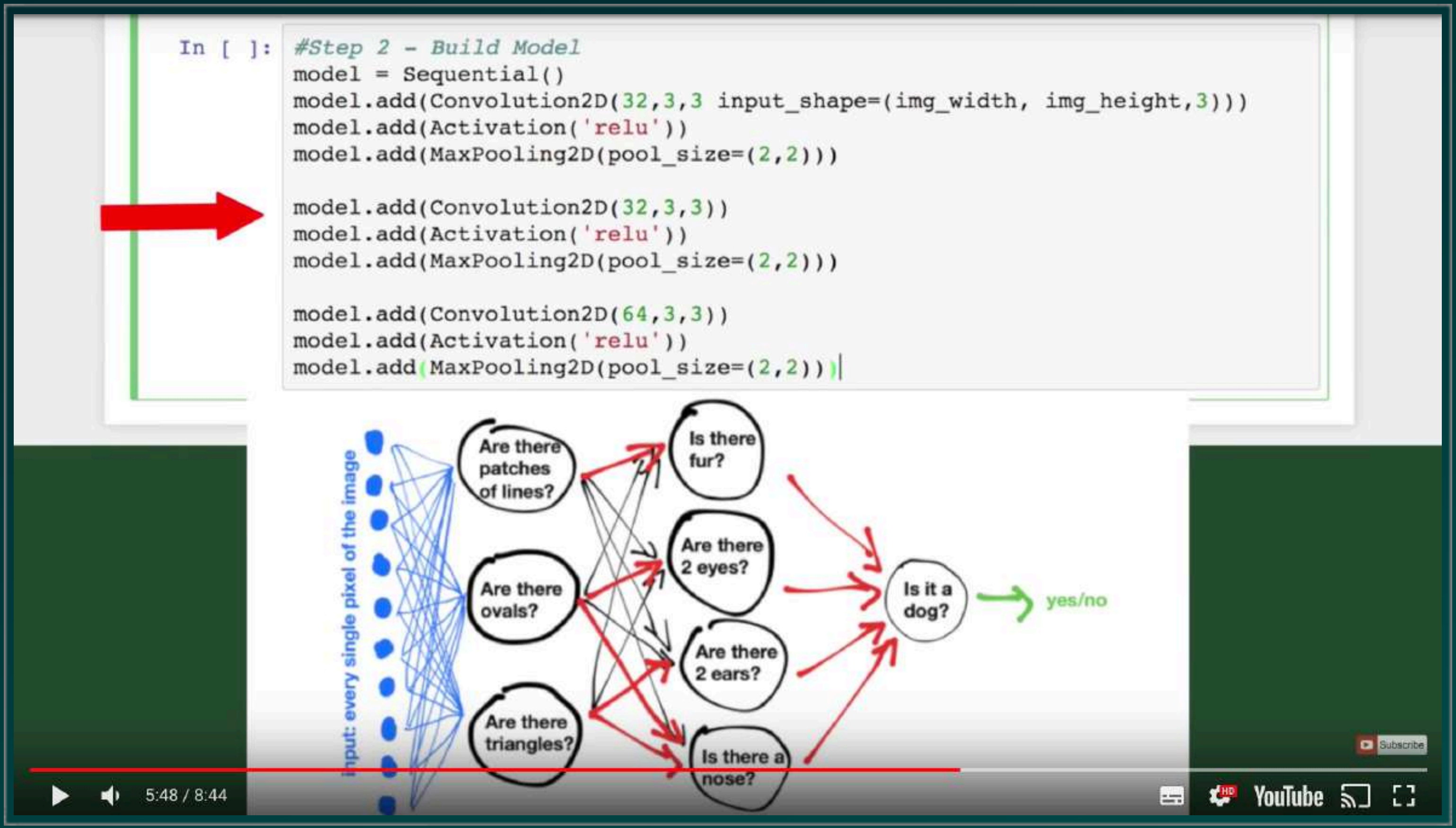



```
def conv_net(x, weights, biases, dropout):
    # Layer 1 - 28*28*1 to 14*14*32
    conv1 = conv2d(x, weights['wc1'], biases['bc1'])
    conv1 = maxpool2d(conv1, k=2)

    # Layer 2 - 14*14*32 to 7*7*64
    conv2 = conv2d(conv1, weights['wc2'], biases['bc2'])
    conv2 = maxpool2d(conv2, k=2)

    # Fully connected layer - 7*7*64 to 1024
    fc1 = tf.reshape(conv2, [-1, weights['wd1'].get_shape().as_list()[0]])
    fc1 = tf.add(tf.matmul(fc1, weights['wd1']), biases['bd1'])
    fc1 = tf.nn.relu(fc1)
    fc1 = tf.nn.dropout(fc1, dropout)

    # Output Layer - class prediction - 1024 to 10
    out = tf.add(tf.matmul(fc1, weights['out']), biases['out'])
    return out
```

Part 2 – Convolutional Neural Networks

- Model Evaluation and Validation
- MiniFlow
- Sentiment Analysis with Andrew Trask
- Intro to TensorFlow
- Cloud Computing
- Deep Neural Networks
- Convolutional Networks
- **Siraj's Image Classification**
- Weight Initialization
- ...

Siraj will explain how to build Convolutional Neural Network in Keras in 8 minutes.



Part 2 – Convolutional Neural Networks

- Model Evaluation and Validation
- MiniFlow
- Sentiment Analysis with Andrew Trask
- Intro to TensorFlow
- Cloud Computing
- Deep Neural Networks
- Convolutional Networks
- **Siraj's Image Classification**
- Weight Initialization
- …

In each Siraj's "lesson" you will receive his short video where he introduces topic and shows you the code how to implement it in ~10 minutes.

Additionally you are given bunch of links and materials under the video.

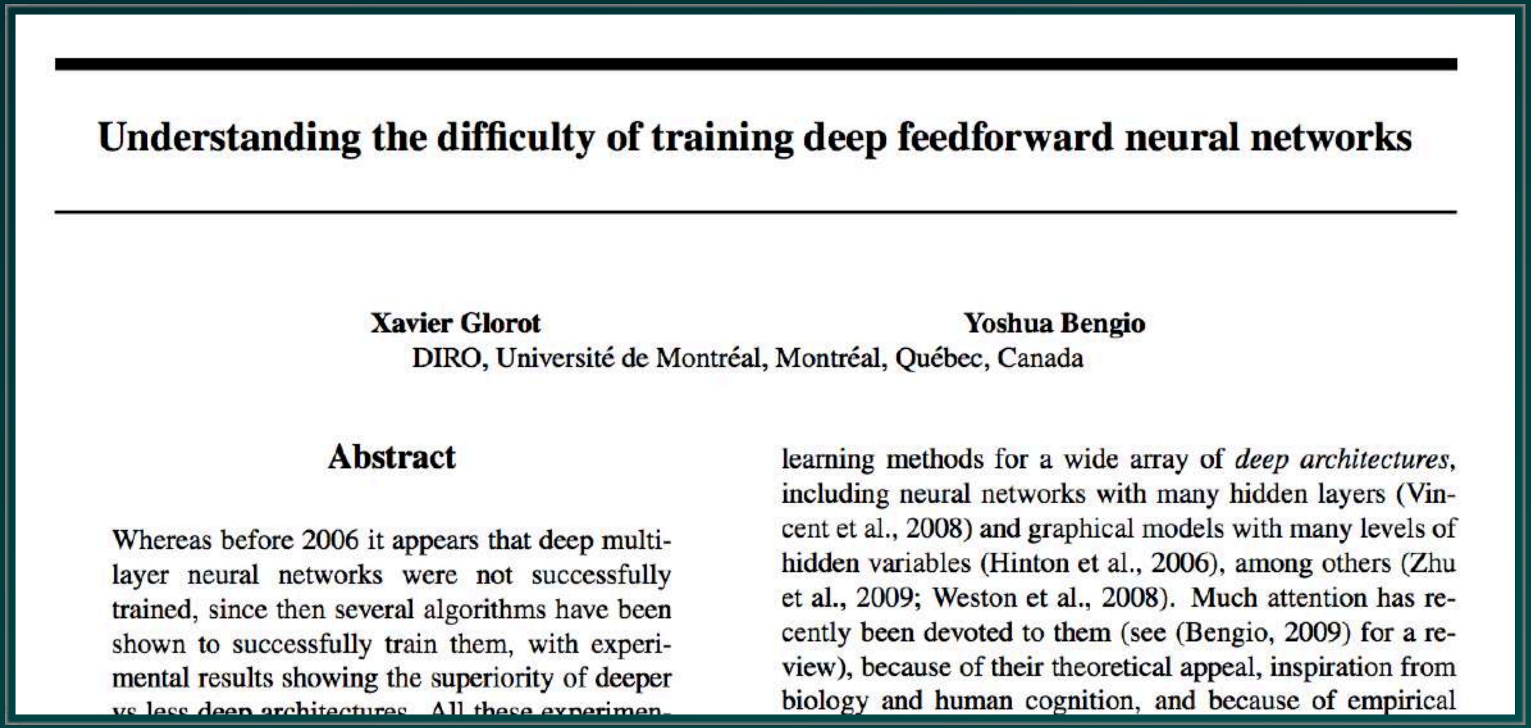
Under some of them there are recordings from coding live-sessions that lasts 1 hour and were organised during first two cohorts of DLFND programme.

Part 2 - Convolutional Neural Networks

- Model Evaluation and Validation
- MiniFlow
- Sentiment Analysis with Andrew Trask
- Intro to TensorFlow
- Cloud Computing
- Deep Neural Networks
- Convolutional Networks
- Siraj's Image Classification
- **Weight Initialization**
- ...

You will receive Jupiter Notebook to read where Neural Network weight initialization is explained.

Method explained there is called Xavier Initialization. Based on paper:



Understanding the difficulty of training deep feedforward neural networks

Xavier Glorot **Yoshua Bengio**
DIRO, Université de Montréal, Montréal, Québec, Canada

Abstract

Whereas before 2006 it appears that deep multi-layer neural networks were not successfully trained, since then several algorithms have been shown to successfully train them, with experimental results showing the superiority of deeper vs less deep architectures. All these experimen-

learning methods for a wide array of *deep architectures*, including neural networks with many hidden layers (Vincent et al., 2008) and graphical models with many levels of hidden variables (Hinton et al., 2006), among others (Zhu et al., 2009; Weston et al., 2008). Much attention has recently been devoted to them (see (Bengio, 2009) for a review), because of their theoretical appeal, inspiration from biology and human cognition, and because of empirical

Part 2 - Convolutional Neural Networks

...

- MiniFlow
- Sentiment Analysis with Andrew Trask
- Intro to TensorFlow
- Cloud Computing
- Deep Neural Networks
- Convolutional Networks
- Siraj's Image Classification
- Weight Initialization
- **Image Classification**

Project no. 2

Deadline: 4 weeks

You will build whole Convolutional Neural Network from scratch and preprocess data by yourself in Tensorflow.

You will solve classification problem - where you will want to teach CNN to recognise colour images of CIFAR-10 dataset.

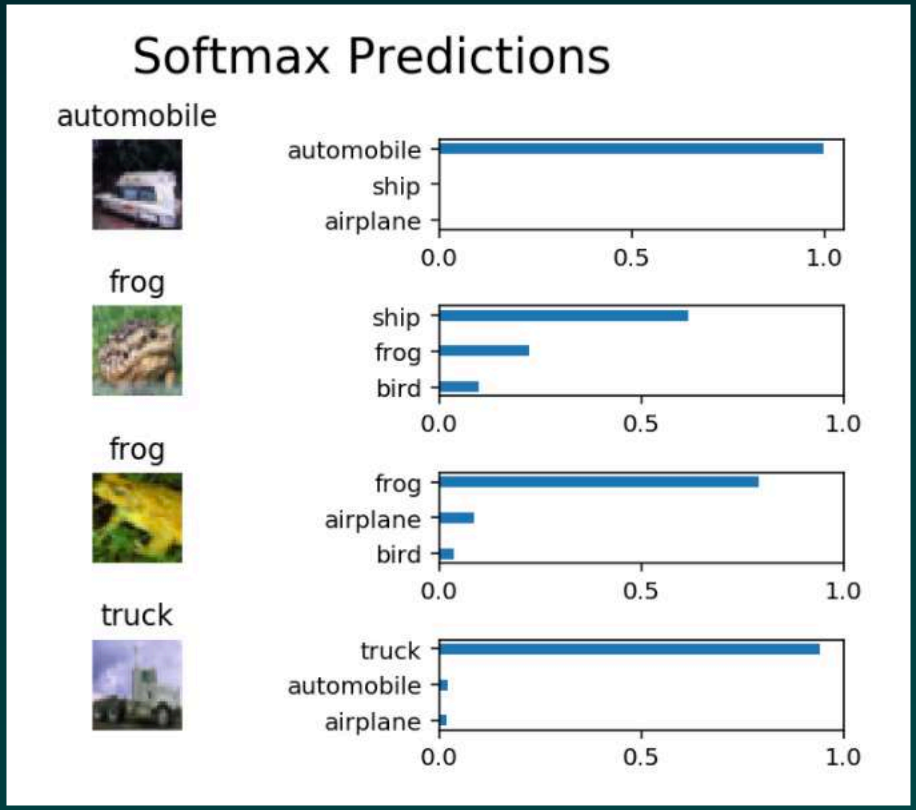
You are given rubric with requirements your project has to meet in order to pass.

Project undergoes code review by specialist and you receive feedback with links/hints what was done well or could be done better.

Part 2 – Convolutional Neural Networks

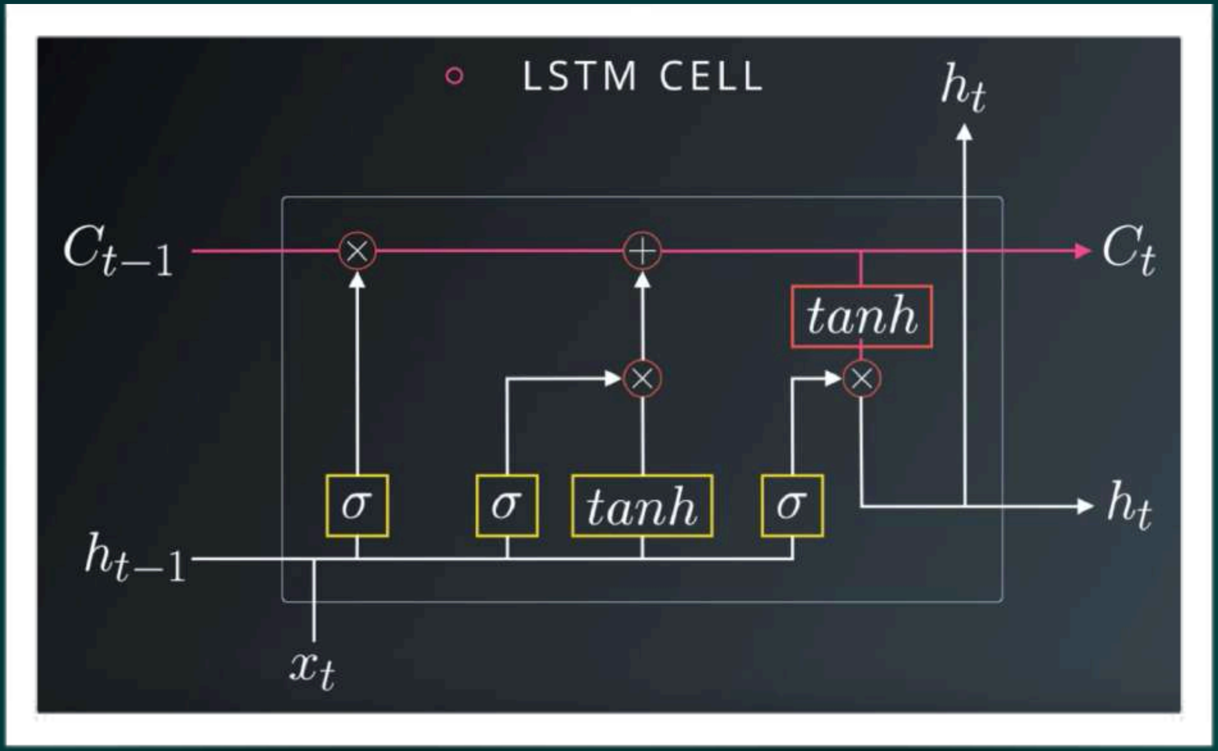
...

- MiniFlow
- Sentiment Analysis with Andrew Trask
- Intro to TensorFlow
- Cloud Computing
- Deep Neural Networks
- Convolutional Networks
- Siraj's Image Classification
- Weight Initialization
- **Image Classification**



Part 3- Recurrent Neural Networks

- Intuition of how Recurrent Neural Network works - no math
- Example how RNN can be used for series data - how to predict next letter of word or next word of sentence
- Intuition how LSTM works:



Part 3- Recurrent Neural Networks

- **Intro to Recurrent Neural Networks**
- Siraj's Stock Prediciton
- Hyperparameters
- Embeddings and Word2Vec
- Siraj's Style Transfer
- Q&A with FloydHub Founders
- Tensorboard
- Siraj's Music Generation
- Siraj's Text Summarization
- ...

- Explanation of how to perform mini-batch when working with RNN - Sequence Batching
- You will be given Jupiter Notebook exercise where you will feed Anna Karenina of Lew Tołstoj to RNN (exercise is named Anna KaRNNa). Model will be able to generate totally new text in style of fed book.

```
checkpoint = 'checkpoints/i1200_l512.ckpt'
samp = sample(checkpoint, 1000, lstm_size, len(vocab), prime="Far")
print(samp)
```

```
Farcial the
confiring to the mone of the correm and thinds. She
she saw the
streads of herself hand only astended of the carres to her his some of the princess of which he came him of
all that his white the dreasing of
thisking the princess and with she was she had
bettee a still and he was happined, with the pood on the mush to the peaters and seet it.

"The possess a streatich, the may were notine at his mate a misted
and the
man of the mother at the same of the seem her
felt. He had not here.

"I conest only be alw you thinking that the partion
of their said."
```

Part 3- Recurrent Neural Networks

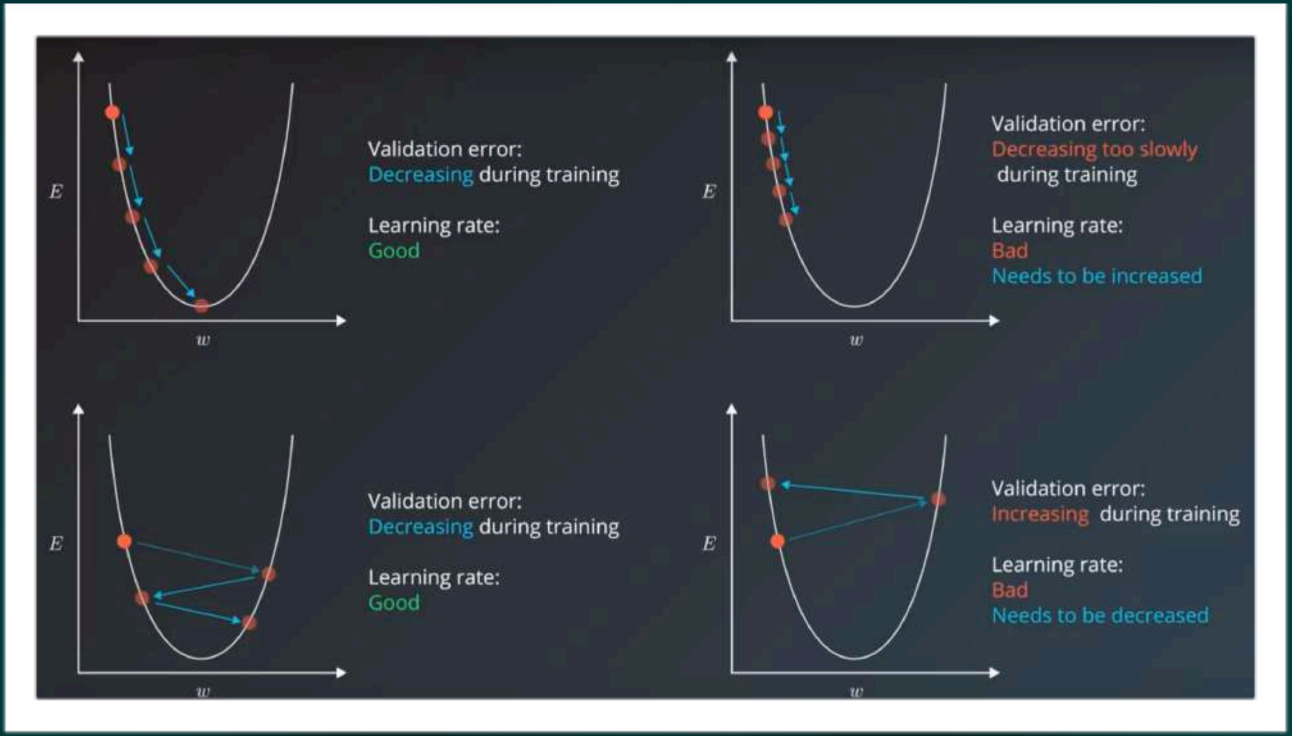
Siraj's YouTube video where he tries to predict stock price of S&P 500 with usage of Recurrent Neural Network in Keras.

You will receive coding challenge - try to predict stock price of Google with usage of data from 3 different inputs. You can send your results to him until next week and he will pick the best work and post on next weeks video.

Part 3- Recurrent Neural Networks

In this lesson you will learn more what impact does hyperparameters has on model.

- learning rate
- mini-batch size
- number of training iteration/epochs
- number of hidden units/layers

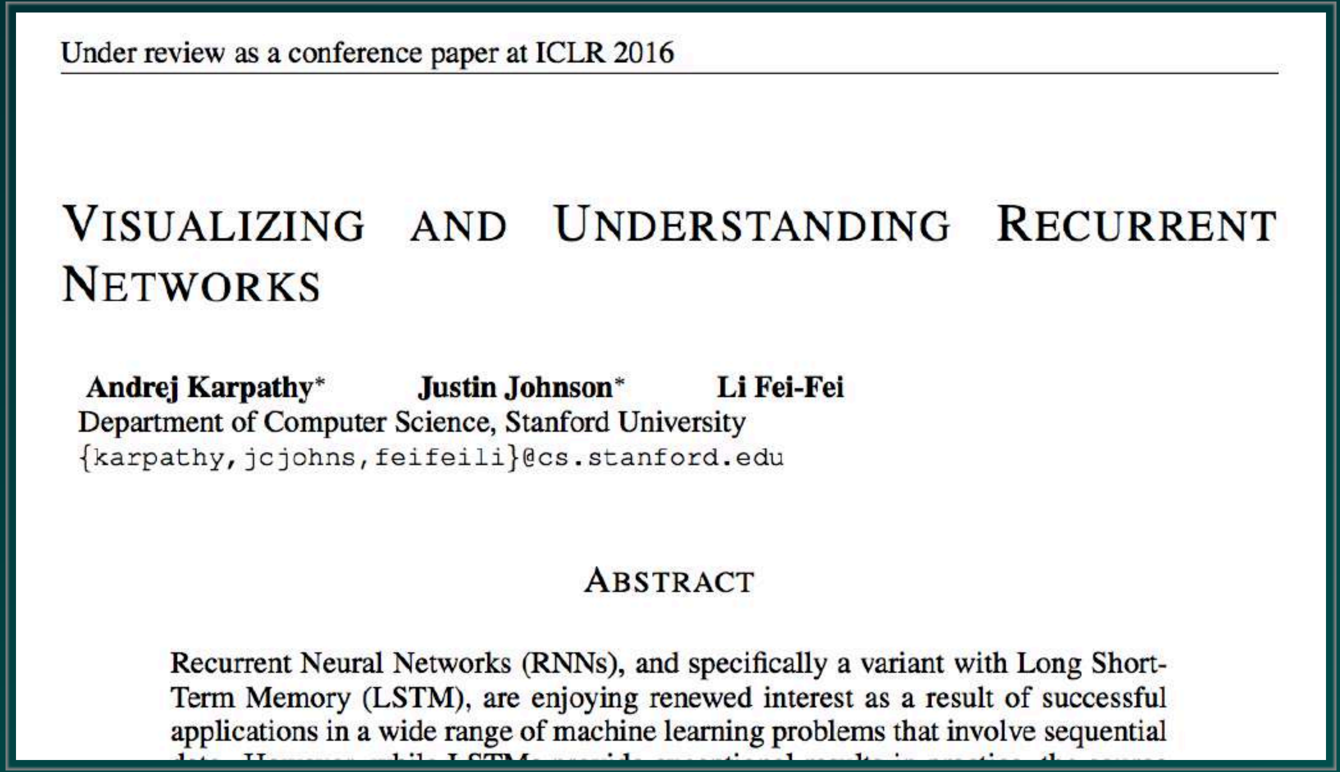


Part 3- Recurrent Neural Networks

- Intro to Recurrent Neural Networks
- Siraj's Stock Prediciton
- **Hyperparameters**
- Embeddings and Word2Vec
- Siraj's Style Transfer
- Q&A with FloydHub Founders
- Tensorboard
- Siraj's Music Generation
- Siraj's Text Summarization
- . . .

You will be also given a lot of research papers for your use to understand more how to tweak RNN parameters such us:

- number of RNN layers
- what kind of cell to use: LSTM, GRU, Vanilla etc.



Under review as a conference paper at ICLR 2016

VISUALIZING AND UNDERSTANDING RECURRENT NETWORKS

Andrej Karpathy* Justin Johnson* Li Fei-Fei
Department of Computer Science, Stanford University
{karpathy,jcjohns,feifeili}@cs.stanford.edu

ABSTRACT

Recurrent Neural Networks (RNNs), and specifically a variant with Long Short-Term Memory (LSTM), are enjoying renewed interest as a result of successful applications in a wide range of machine learning problems that involve sequential

Part 3- Recurrent Neural Networks

- Intro to Recurrent Neural Networks
- Siraj's Stock Prediciton
- **Hyperparameters**
- Embeddings and Word2Vec
- Siraj's Style Transfer
- Q&A with FloydHub Founders
- Tensorboard
- Siraj's Music Generation
- Siraj's Text Summarization
- ...

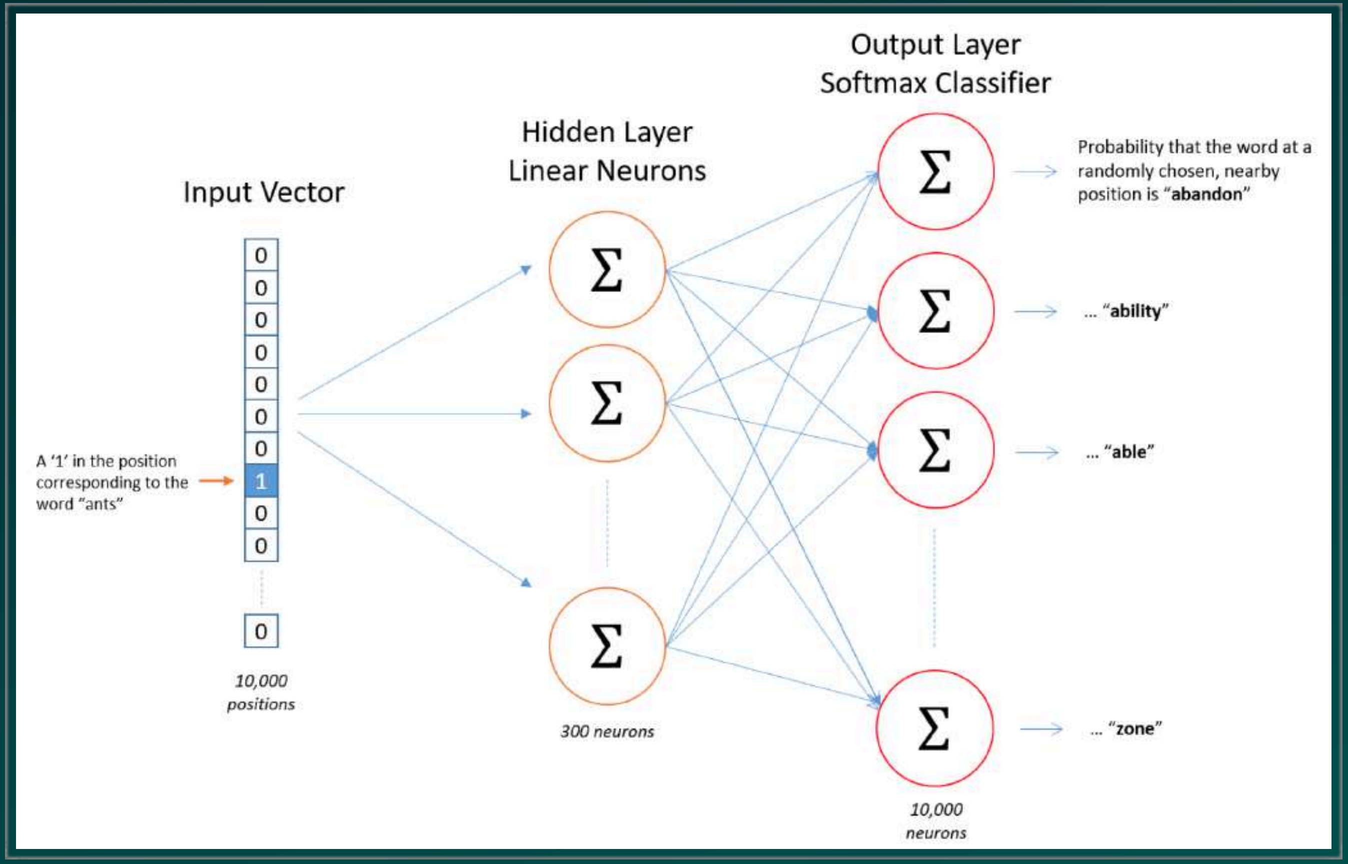
Example RNN Architectures

Application	Cell	Layers	Size	Vocabulary	Embedding Size	Learning Rate	
Speech Recognition (large vocabulary)	LSTM	5, 7	600, 1000	82K, 500K	--	--	paper
Speech Recognition	LSTM	1, 3, 5	250	--	--	0.001	paper
Machine Translation (seq2seq)	LSTM	4	1000	Source: 160K, Target: 80K	1,000	--	paper
Image Captioning	LSTM	--	512	--	512	(fixed)	paper
Image Generation	LSTM	--	256, 400, 800	--	--	--	paper
Question Answering	LSTM	2	500	--	300	--	pdf
Text Summarization	GRU		200	Source: 119K, Target: 68K	100	0.001	pdf

Part 3- Recurrent Neural Networks

- Intro to Recurrent Neural Networks
- Siraj's Stock Prediciton
- Hyperparameters
- **Embeddings and Word2Vec**
- Siraj's Style Transfer
- Q&A with FloydHub Founders
- Tensorboard
- Siraj's Music Generation
- Siraj's Text Summarization
- ...

You will be given Jupiter Notebook with exercise to implement RNN that will learn Word Embeddings with usage of Word2Vec with Skip-GRAM architecture. CBOW architecture will also be mentioned.



Part 3- Recurrent Neural Networks

- Intro to Recurrent Neural Networks
- Siraj's Stock Prediciton
- Hyperparameters
- **Embeddings and Word2Vec**
- Siraj's Style Transfer
- Q&A with FloydHub Founders
- Tensorboard
- Siraj's Music Generation
- Siraj's Text Summarization
- ...

Part 3- Recurrent Neural Networks

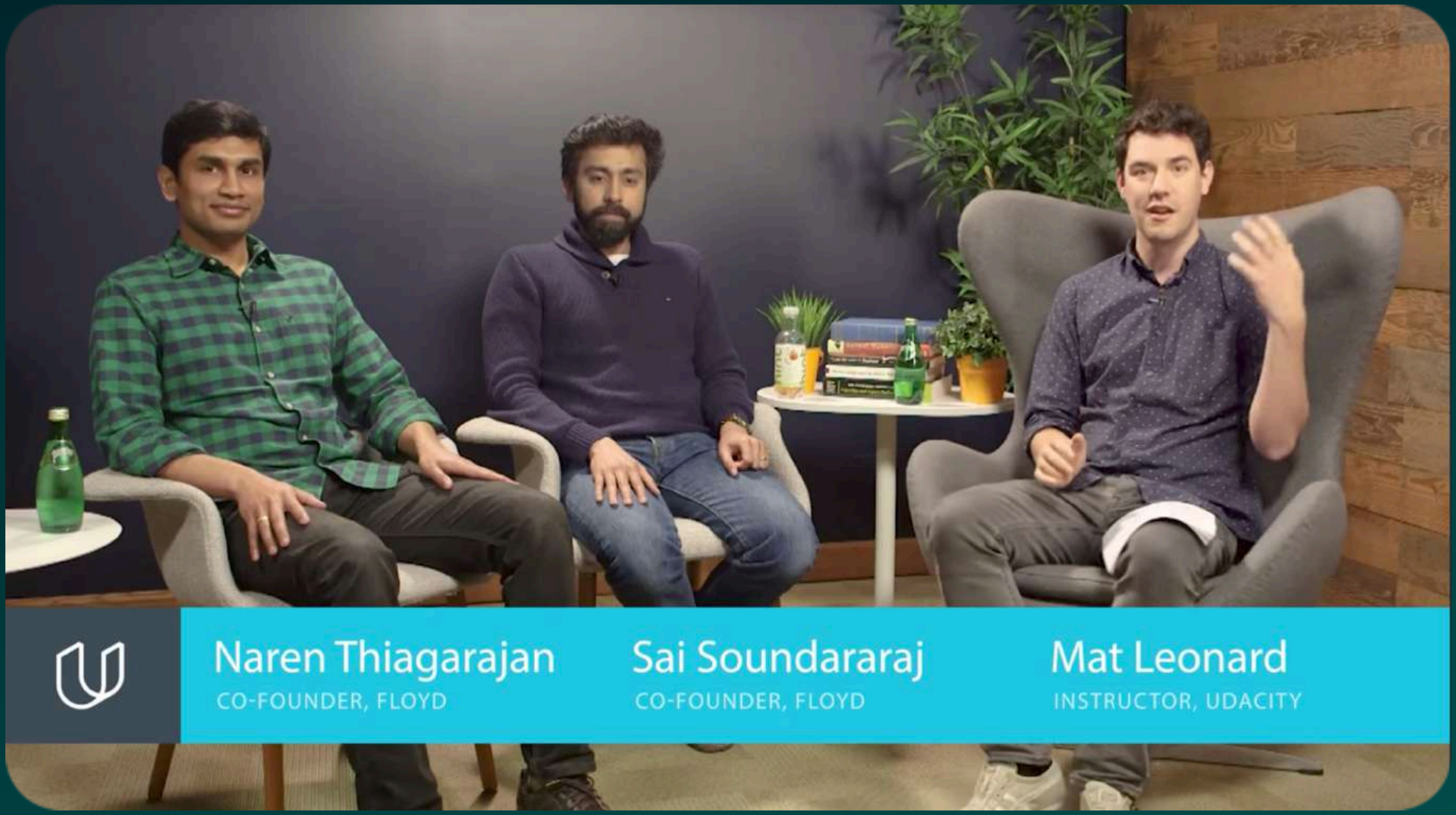
- Intro to Recurrent Neural Networks
- Siraj's Stock Prediciton
- Hyperparameters
- Embeddings and Word2Vec
- **Siraj's Style Transfer**
- Q&A with FloydHub Founders
- Tensorboard
- Siraj's Music Generation
- Siraj's Text Summarization
- ...

In this 10 minutes video Siraj will tell you how to implement Fast-Style-Transfer in Keras by using transfer learning - pretrained, embedded in Keras VGG16 network.

+

Part 3- Recurrent Neural Networks

- Intro to Recurrent Neural Networks
- Siraj's Stock Prediciton
- Hyperparameters
- Embeddings and Word2Vec
- Siraj's Style Transfer
- **Q&A with FloydHub Founders**
- Tensorboard
- Siraj's Music Generation
- Siraj's Text Summarization
- ...

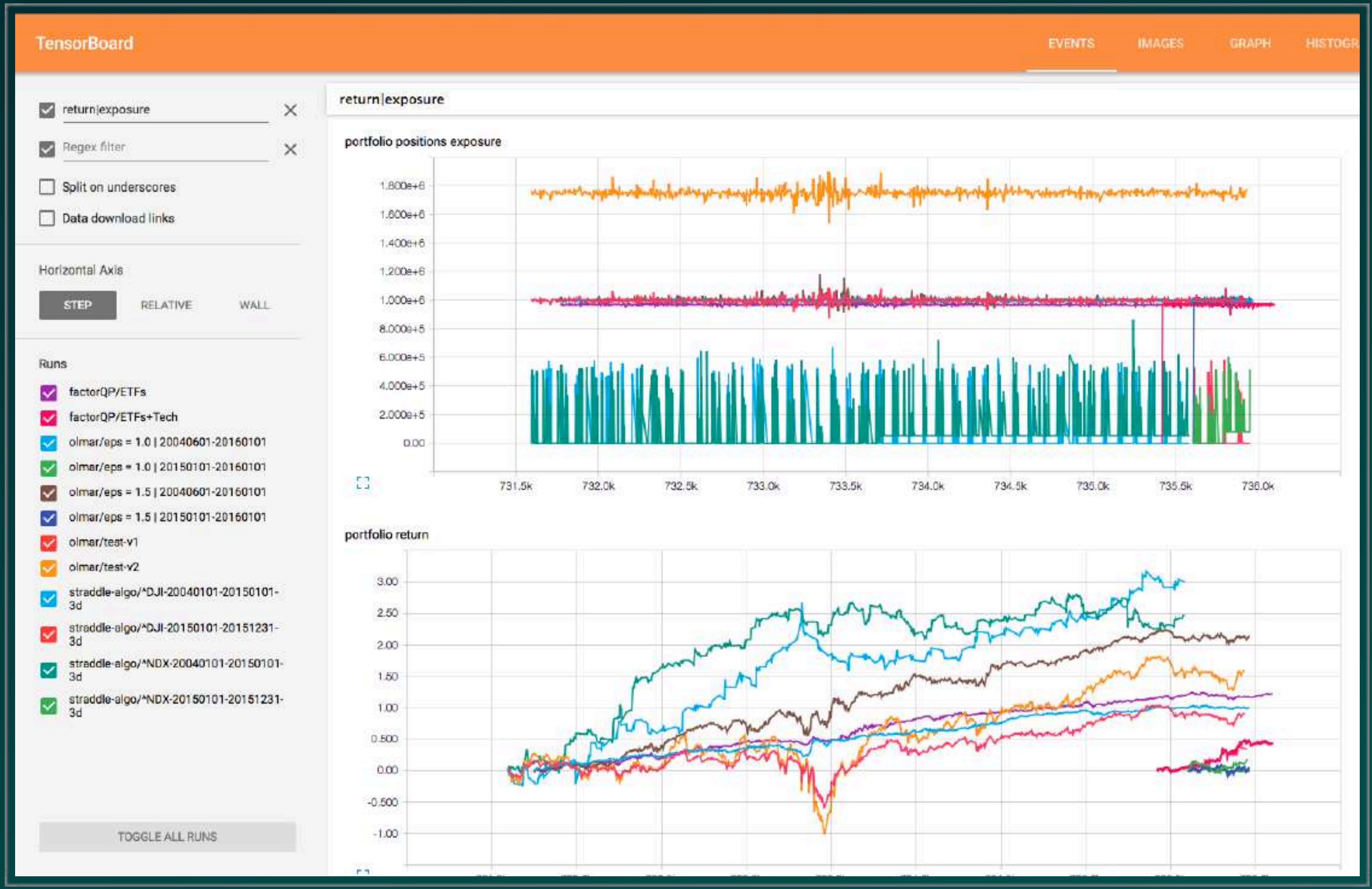


Part 3- Recurrent Neural Networks

- Intro to Recurrent Neural Networks
- Siraj's Stock Prediciton
- Hyperparameters
- Embeddings and Word2Vec
- Siraj's Style Transfer
- Q&A with FloydHub Founders
- **Tensorboard**
- Siraj's Music Generation
- Siraj's Text Summarization
- …

Learn how to use tf.summary module and tf.variable_scope of Tensorflow to visualise what's happening inside your model.

You will modify previous exercise - Anna KaRNNa.



Part 3- Recurrent Neural Networks

In this video Siraj will create Generate MIDI in Keras.

He will feed song of Pat Metheny to RNN to teach it how to generate new music.

There will be also overview of how LSTM cell works as well as explanation of Vanishing Gradient Problem.

Part 3- Recurrent Neural Networks

- Intro to Recurrent Neural Networks
- Siraj's Stock Prediciton
- Hyperparameters
- Embeddings and Word2Vec
- Siraj's Style Transfer
- Q&A with FloydHub Founders
- Tensorboard
- Siraj's Music Generation
- **Siraj's Text Summarization**
- ...

In this video Siraj will create Text Summarizer in Keras.

He will feed BBC News articles to RNN to teach it how to sum up the article in 10 words.

```
#Initialize Encoder RNN with Embeddings
encoder = build_model(word_embeddings)
encoder.compile(loss='categorical_crossentropy', optimizer='rmsprop')
encoder.save_weights('embeddings.pkl', overwrite=True)

#Initialize Decoder RNN with Embeddings
with open('embeddings.pkl', 'rb') as fp:
    embeddings = pickle.load(fp)
decoder = build_model(embeddings)

#Convert a given article to a headline
headline1 = pr.gen_headline(decoder, desc[1])

HEADS: 20.5102220631 How The Golden Globes Will Make You Cry

#Convert a given article to a headline
headline2 = pr.gen_headline(decoder, desc[2])

HEADS: 15.8060034323 The 10 Most Popular Songs Of All Time
```

Part 3- Recurrent Neural Networks

...

- **Sentiment Prediction RNN**
- Generate TV Scripts
- Sequence to Sequence
- Siraj's Chatbot
- Transfer Learning in TF
- Siraj's Language Translation
- Reinforcement Learning
- Siraj's Reinforcement Learning
- Translation Project

In previously done exercise with Andrew Trask, MLP was used to predict sentiment of text.

In this exercise RNN with word embeddings will be used instead of MLP.

This will give much better results as instead of assigning sentiment value to each word based on label of text in which word occurred - RNN will assign sentiment value to word based on context in which word appears.

Part 3- Recurrent Neural Networks

...

- Sentiment Prediction RNN
- **Generate TV Scripts**
- Sequence to Sequence
- Siraj's Chatbot
- Transfer Learning in TF
- Siraj's Language Translation
- Reinforcement Learning
- Siraj's Reinforcement Learning
- Translation Project

Project no. 3

Deadline: 4 weeks

You will build Recurrent Neural Network by using tf.layer module and reusing already implemented LSTM cells.

You will feed script of The Simpsons TV series to neural network in order to generate totally new TV script.

You are given rubric with requirements your project has to meet in order to pass.

Project undergoes code review by specialist and you receive feedback with links/hints what was done well or could be done better.

Part 3- Recurrent Neural Networks

```
moe_szyslak: oh, the heat's been on since them bush girls were in here.
homer_simpson: since i think about time they saw this.
homer_simpson: oh, moe. maybe i've learned your best drink it is... i couldn't you find the hand f
or a couple.
moe_szyslak:(to moe) hey, homer, if you got it," i aren't a glass up to my guy who shows up in the
bar.
moe_szyslak:(laughs) oh marge, i can't take that out?
lenny_leonard: that's the best break, the low...
moe_szyslak: barney, look! that was the longest gets every beer / and city third the house!
homer_simpson: hey, homer. here i come up, homer. i am changing the other day and water?
moe_szyslak: hey, you don't have to wait my new guy i ever. who am i got a big idea?
moe_szyslak:(starts to watch me go to the bar and die. you know, when they said no girl somewhere!
```

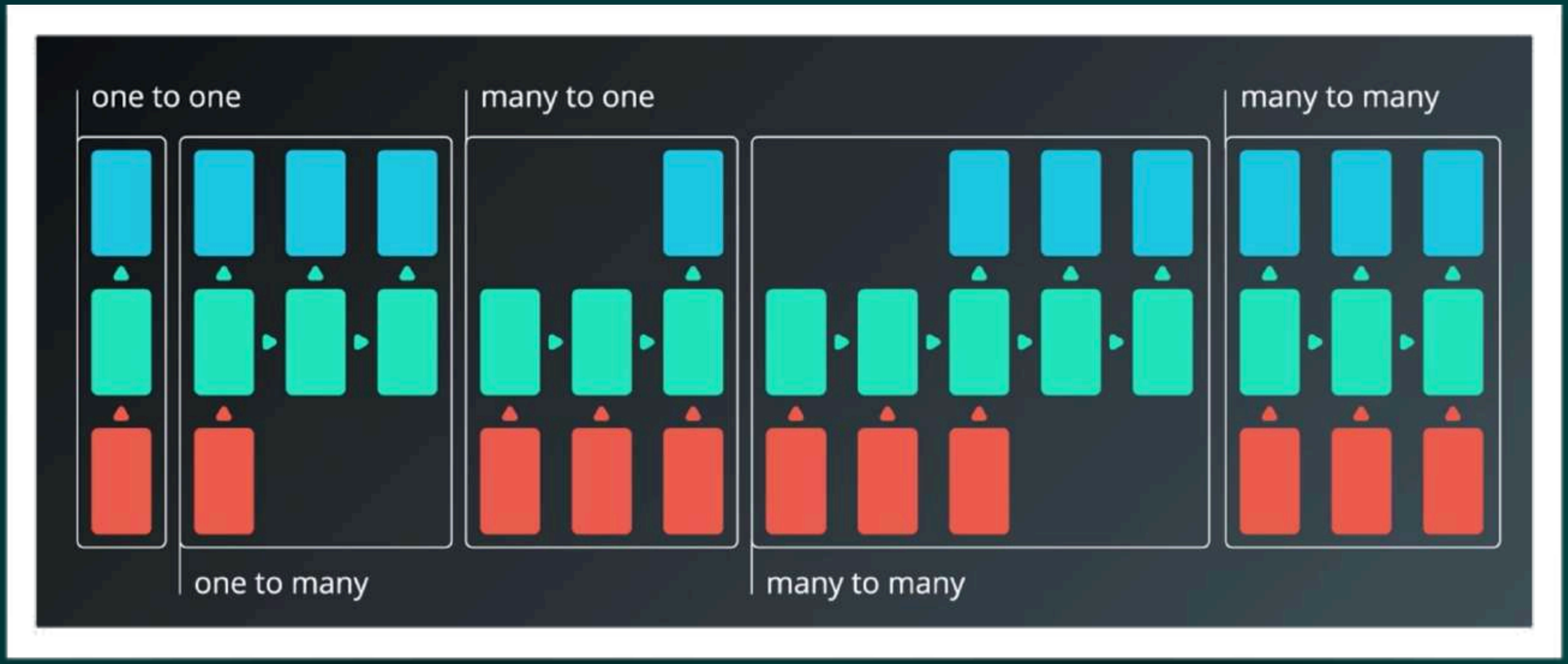
Part 3- Recurrent Neural Networks

...

- Sentiment Prediction RNN
- Generate TV Scripts
- **Sequence to Sequence**
- Siraj's Chatbot
- Transfer Learning in TF
- Siraj's Language Translation
- Reinforcement Learning
- Siraj's Reinforcement Learning
- Translation Project

Learn about different RNN structures. Exercises until now used Many to One architecture (many words to single sentiment value "positive" or "negative").

In this exercise we will implement Many to Many architecture (or just Sequence to Sequence).



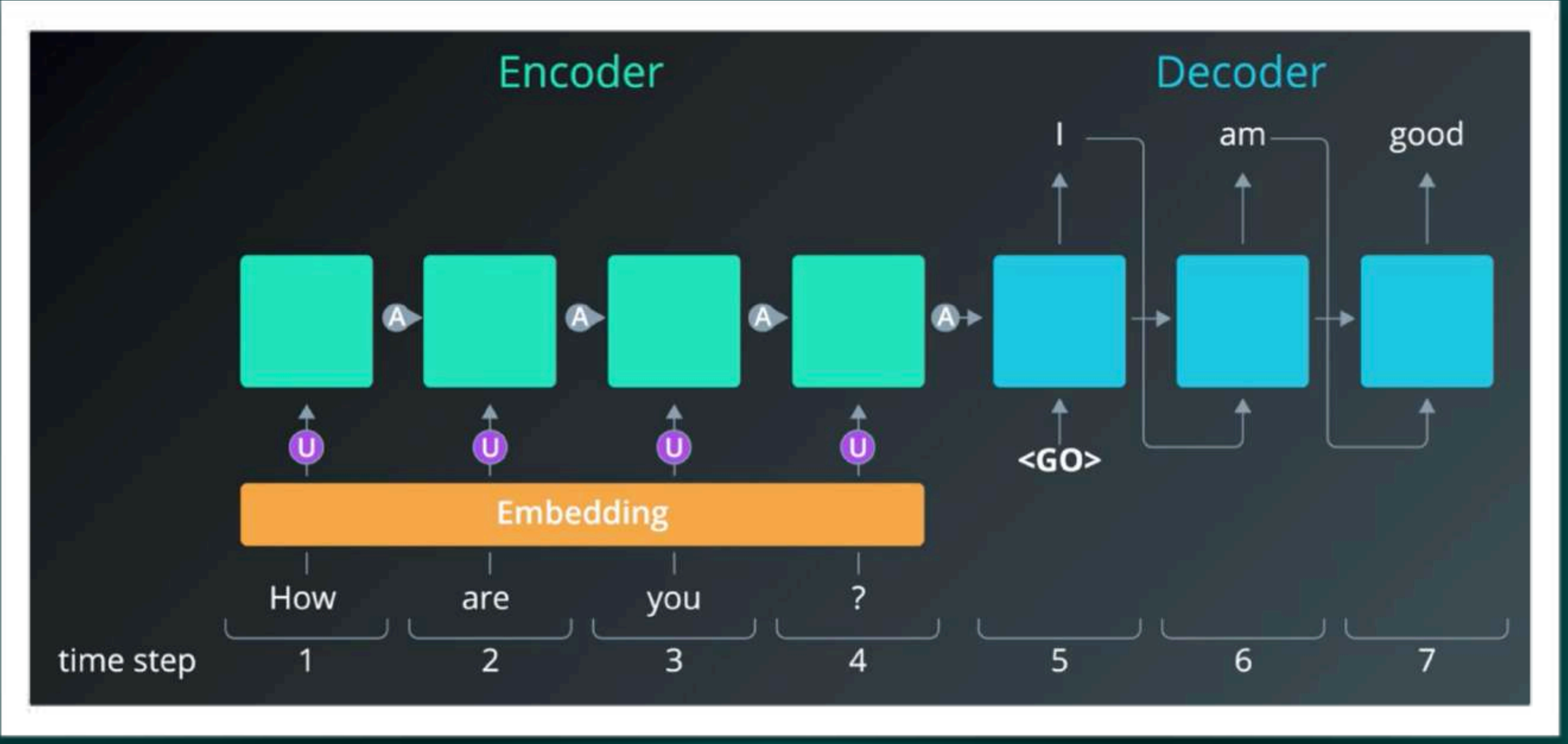
Part 3- Recurrent Neural Networks

…

- Sentiment Prediction RNN
- Generate TV Scripts
- **Sequence to Sequence**
- Siraj's Chatbot
- Transfer Learning in TF
- Siraj's Language Translation
- Reinforcement Learning
- Siraj's Reinforcement Learning
- Translation Project

You will also gain intuition how which structures are used to create chatbot or language translator.

You will become familiar with Encoder/Decoder Neural Network structures.



Part 3- Recurrent Neural Networks

In 8 minutes Siraj will introduce to you into history of creating chatbot and experimental models that were invented in order to create it - Memory Network (network that uses external memory to store it's data), Dynamic Memory Network.

He will build Dynamic Memory Network in Keras and show us work of person that have uploaded DMN on the web so we can play with it and test it.

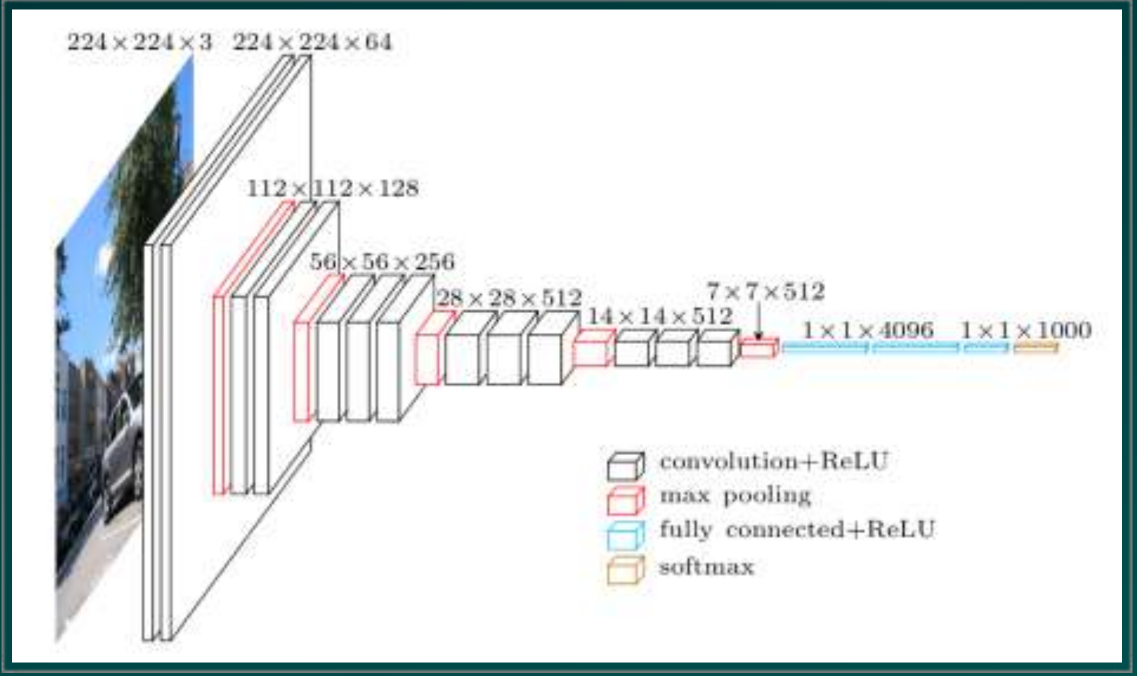
Part 3- Recurrent Neural Networks

...

- Sentiment Prediction RNN
- Generate TV Scripts
- Sequence to Sequence
- Siraj's Chatbot
- **Transfer Learning in TF**
- Siraj's Language Translation
- Reinforcement Learning
- Siraj's Reinforcement Learning
- Translation Project

You will receive Jupiter Notebook with exercise where you can learn how to do transfer learning in Tensorflow.

You will download pre-trained VGG16 Convolutional Neural Network and train fully connected layers to teach it how to recognise flower images.



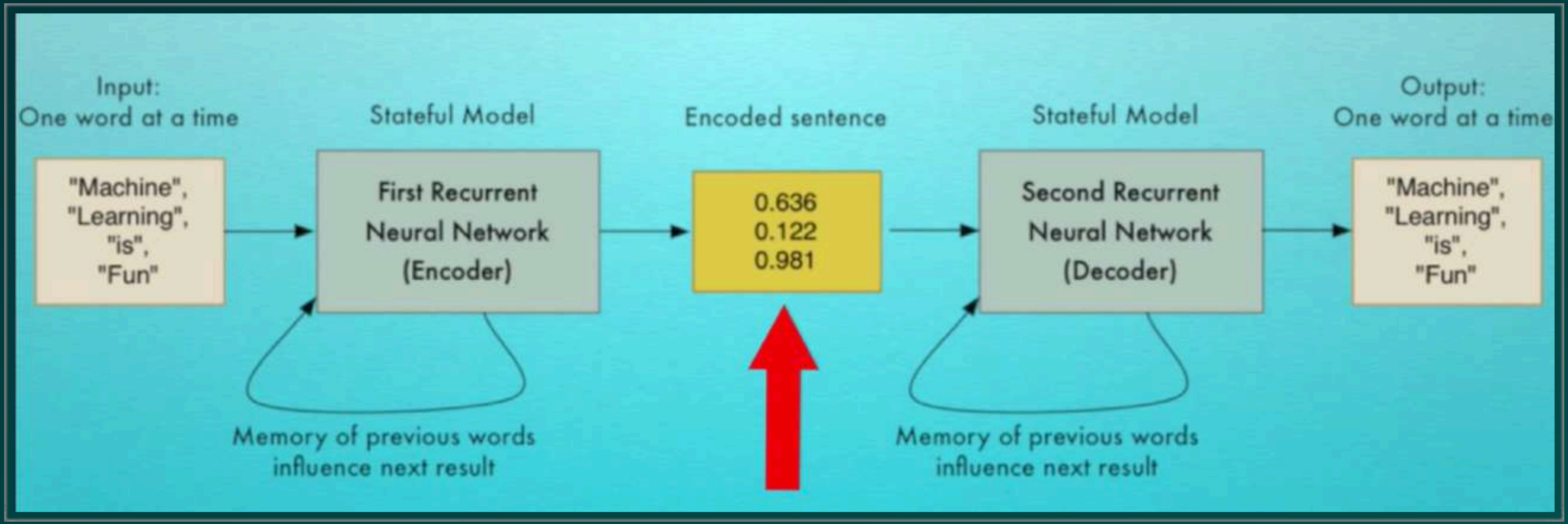
Part 3- Recurrent Neural Networks

...

- Sentiment Prediction RNN
- Generate TV Scripts
- Sequence to Sequence
- Siraj's Chatbot
- Transfer Learning in TF
- **Siraj's Language Translation**
- Reinforcement Learning
- Siraj's Reinforcement Learning
- Translation Project

Siraj will introduce you to very short language translator history.

He will implement simple language translator based on RNN Seq2Seq model with LSTM cells.



Part 3- Recurrent Neural Networks

...

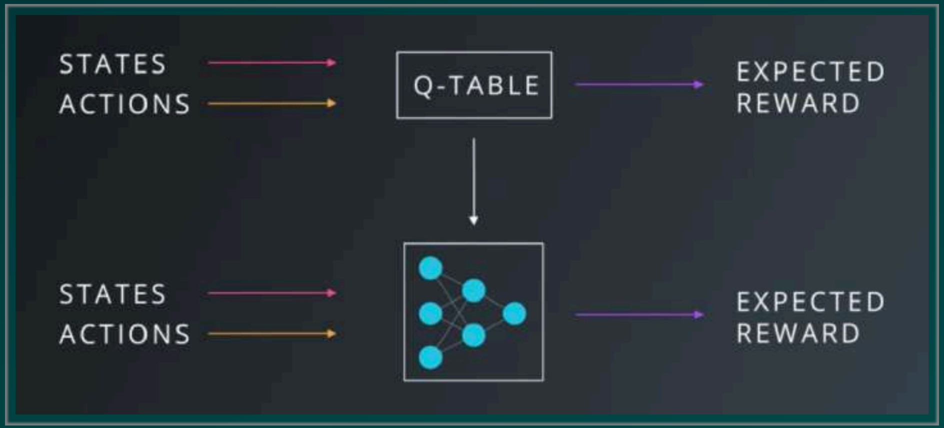
- Sentiment Prediction RNN
- Generate TV Scripts
- Sequence to Sequence
- Siraj's Chatbot
- Transfer Learning in TF
- Siraj's Language Translation
- **Reinforcement Learning**
- Siraj's Reinforcement Learning
- Translation Project

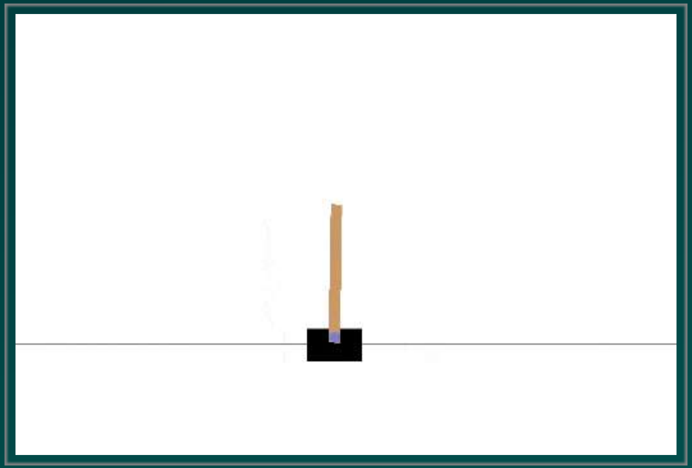
Introduction to Reinforcement Learning.

There will be more links to blog articles rather than Udacity videos.

You will learn how Bellman Equation and Q-Table can be applied for clearing simple game.

There will be Jupiter Notebook exercise where you will implement Deep-Q-Learning on Cart-Pole game from OpenAI Gym.





Part 3- Recurrent Neural Networks

In probability theory, the multi-armed bandit problem is a problem in which a gambler at a row of slot machines (sometimes known as "one-armed bandits") has to decide which machines to play, how many times to play each machine and in which order to play them

In next video Siraj will show how to implement policy gradients technique to solve this problem.

Part 3- Recurrent Neural Networks

...

- Sentiment Prediction RNN
- Generate TV Scripts
- Sequence to Sequence
- Siraj's Chatbot
- Transfer Learning in TF
- Siraj's Language Translation
- Reinforcement Learning
- Siraj's Reinforcement Learning
- **Translation Project**

Project no. 4

Deadline: 4 weeks

You will build Recurrent Neural Network with Sequence to Sequence architecture.

You will feed RNN with English and French sentences and teach it how to translate from English to French.

You are given rubric with requirements your project has to meet in order to pass.

Project undergoes code review by specialist and you receive feedback with links/hints what was done well or could be done better.

Part 3- Recurrent Neural Networks

...

- Sentiment Prediction RNN
- Generate TV Scripts
- Sequence to Sequence
- Siraj's Chatbot
- Transfer Learning in TF
- Siraj's Language Translation
- Reinforcement Learning
- Siraj's Reinforcement Learning
- **Translation Project**

```
Dataset Stats
Roughly the number of unique words: 227
Number of sentences: 137861
Average number of words in a sentence: 13.225277634719028

English sentences 0 to 10:
new jersey is sometimes quiet during autumn , and it is snowy in april .
the united states is usually chilly during july , and it is usually freezing in november .
california is usually quiet during march , and it is usually hot in june .
the united states is sometimes mild during june , and it is cold in september .
your least liked fruit is the grape , but my least liked is the apple .
his favorite fruit is the orange , but my favorite is the grape .
paris is relaxing during december , but it is usually chilly in july .
new jersey is busy during spring , and it is never hot in march .
our least liked fruit is the lemon , but my least liked is the grape .
the united states is sometimes busy during january , and it is sometimes warm in november .

French sentences 0 to 10:
new jersey est parfois calme pendant l' automne , et il est neigeux en avril .
les états-unis est généralement froid en juillet , et il gèle habituellement en novembre .
california est généralement calme en mars , et il est généralement chaud en juin .
les états-unis est parfois légère en juin , et il fait froid en septembre .
votre moins aimé fruit est le raisin , mais mon moins aimé est la pomme .
son fruit préféré est l'orange , mais mon préféré est le raisin .
paris est relaxant en décembre , mais il est généralement froid en juillet .
new jersey est occupé au printemps , et il est jamais chaude en mars .
notre fruit est moins aimé le citron , mais mon moins aimé est le raisin .
les états-unis est parfois occupé en janvier , et il est parfois chaud en novembre .
```

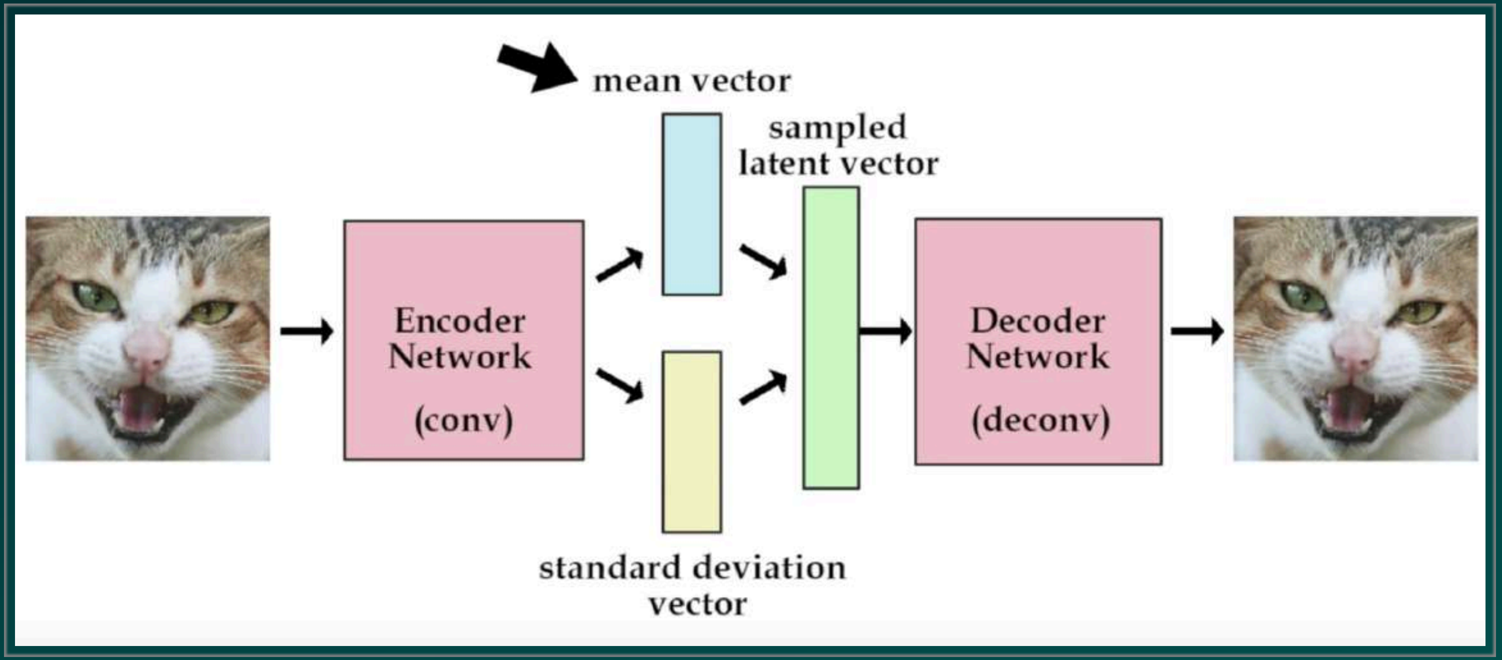
```
INFO:tensorflow:Restoring parameters from checkpoints/dev
Input
  Word Ids:      [18, 199, 217, 4, 83, 182, 97]
  English Words: ['he', 'saw', 'a', 'old', 'yellow', 'truck', '.']

Prediction
  Word Ids:      [85, 84, 248, 76, 142, 141, 206, 292, 1]
  French Words: il a vu un vieux camion jaune . <EOS>
```

Part 4 - Generative Adversarial Networks

- **Siraj's Image Generation**
- Autoencoders
- Generative Adversarial Networks
- Siraj's Video Generation
- DCGANs
- Siraj's One Shoot Learning
- Generate Faces

In this video Siraj will give you brief idea how Autoencoder can be used for image generation.

He will mention Convolutional Encoder which saves image in form of numbers - and Deconvolutional Decoder which generates image based on those numbers.



Part 4 - Generative Adversarial Networks

- Siraj's Image Generation
- **Autoencoders**
- Generative Adversarial Networks
- Siraj's Video Generation
- DCGANs
- Siraj's One Shoot Learning
- Generate Faces

You will receive Jupyter Notebook with exercise where you will create Autoencoder to generate digit images based on MNIST dataset.

- how to build Autoencoder
- how to implement Fully Connected and Convolutional Encoder
- how works and how to implement Fully Connected and Deconvolution Decoder
- how Autoencoder can be used for data encryption and image de-noising

Part 4 - Generative Adversarial Networks

Siraj's Image Generation

Autoencoders

Generative Adversarial Networks

Siraj's Video Generation

DCGANs

Siraj's One Shoot Learning

Generate Faces

Part 4 - Generative Adversarial Networks

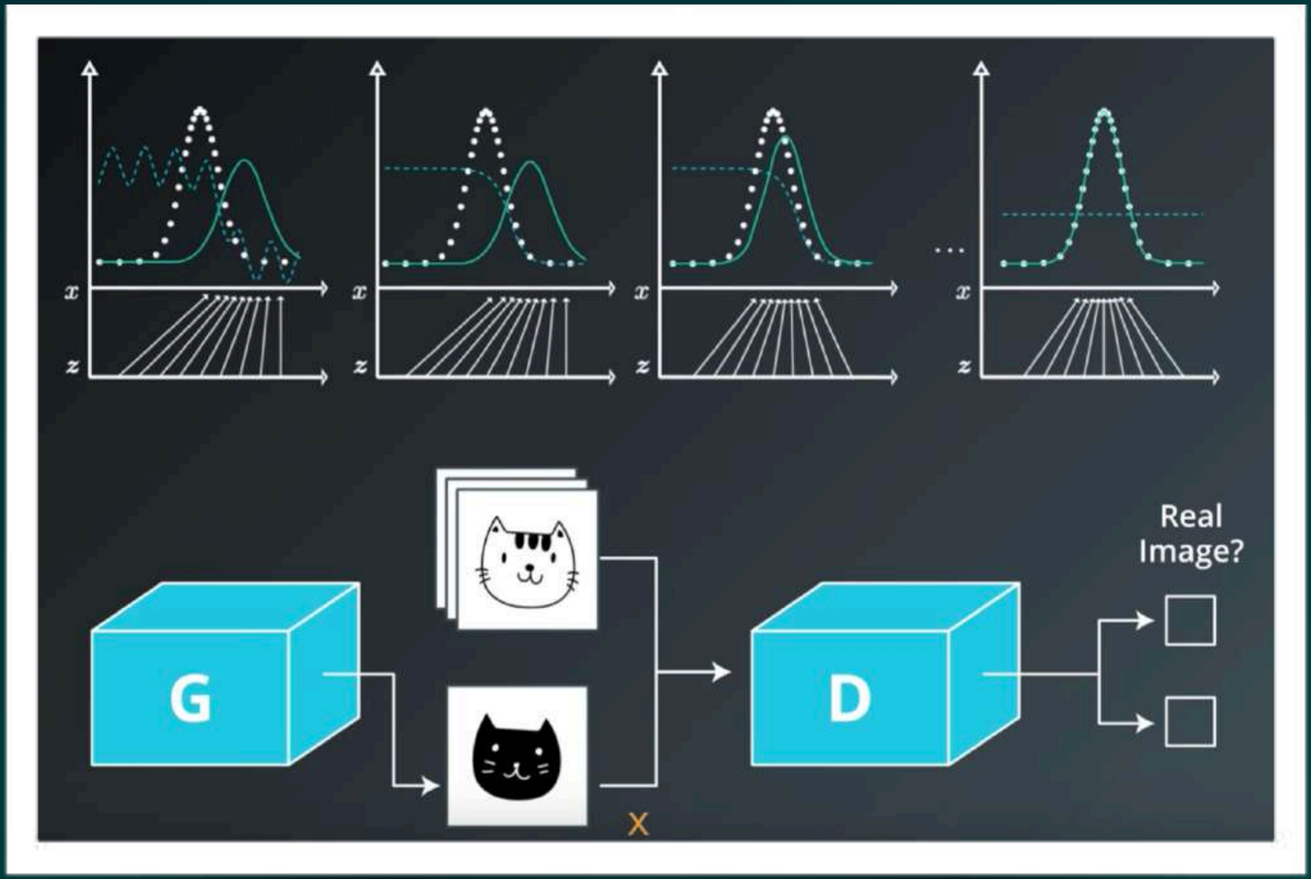
- Siraj's Image Generation
- Autoencoders
- **Generative Adversarial Networks**
- Siraj's Video Generation
- DCGANs
- Siraj's One Shoot Learning
- Generate Faces

Lesson from creator of GANs - Ian Goodfellow!

Part 4 - Generative Adversarial Networks

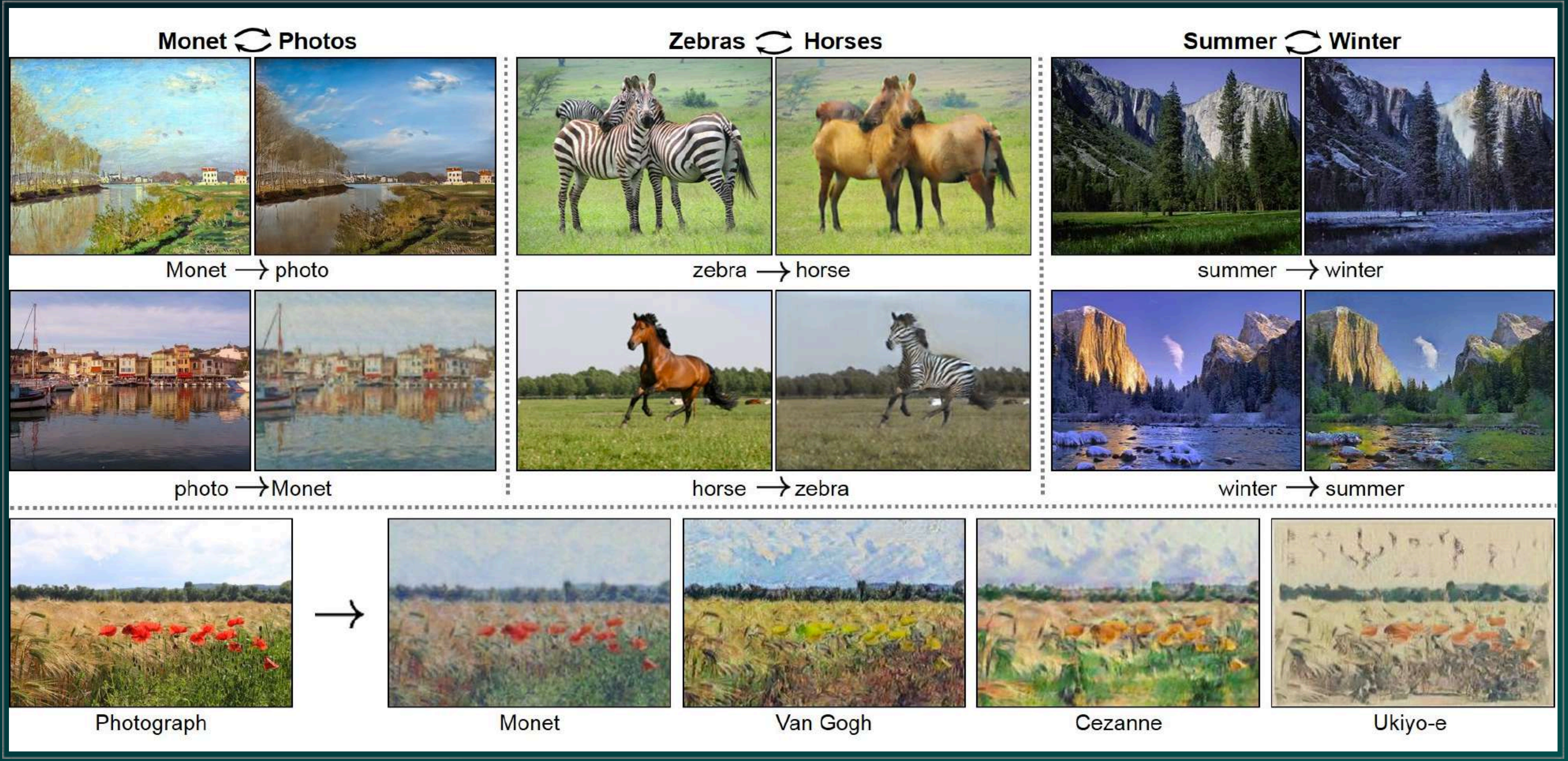
You will learn exactly how Generative Adversarial Networks work.

At start idea of Discriminator and Generator network will be explained.



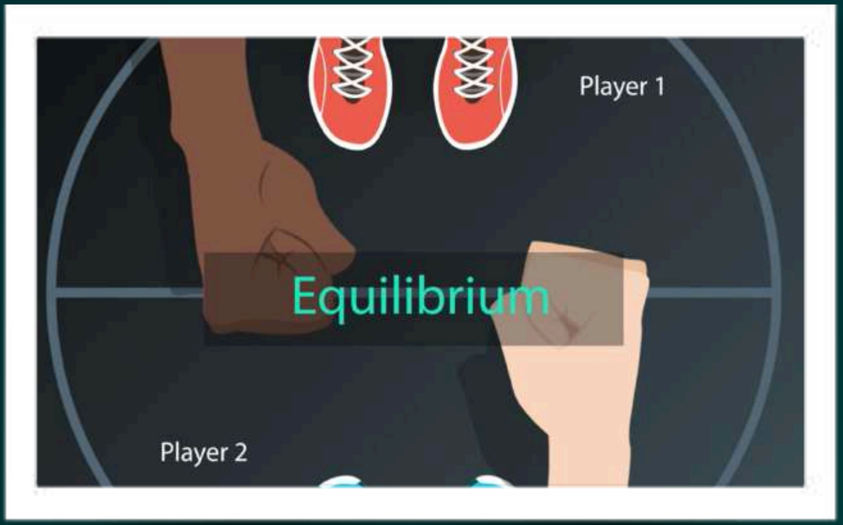
Part 4 - Generative Adversarial Networks

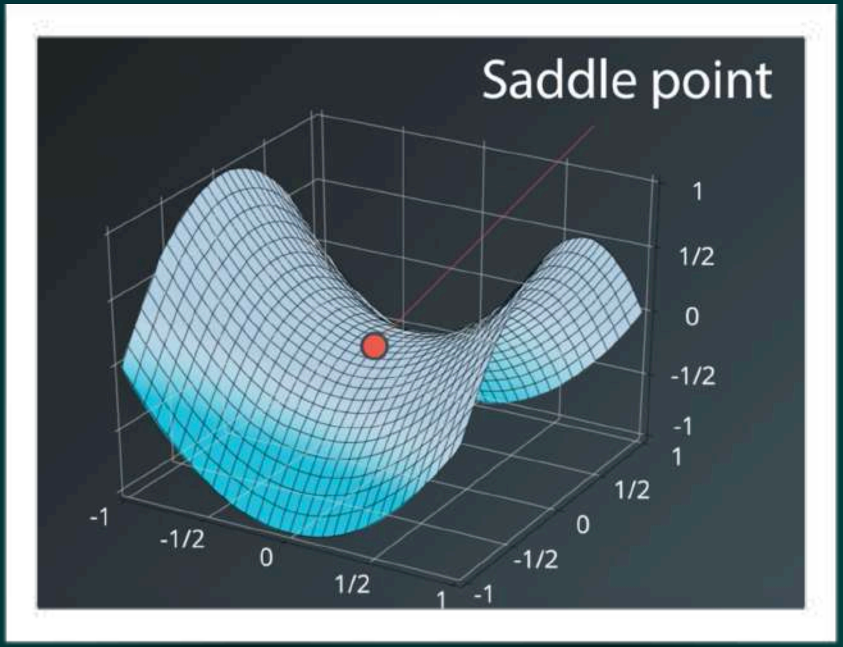
Cycle GAN usages



Part 4 - Generative Adversarial Networks

Then you will learn about connection between Generator and Discriminator errors and when equilibrium state between both networks is obtained - explained on Rock-Paper-Scissors game example.





Part 4 - Generative Adversarial Networks

- Siraj's Image Generation
- Autoencoders
- **Generative Adversarial Networks**
- Siraj's Video Generation
- DCGANs
- Siraj's One Shoot Learning
- Generate Faces

In next part you will receive a lot of good practices for GAN implementation.

- number of layers for Generator/Discriminator
- explanation of Leaky ReLU activation function
- activation function for output of Generator - Hyperbolic Tangent
- use Adam optimiser
- Numerically Stable Cross Entropy - apply smoothing to label values multiplying them by 0.9
- how to connect errors of both networks mathematically
- use BatchNorm on every layer of DCGAN but not last layer of Generator and first layer of Discriminator

Part 4 - Generative Adversarial Networks

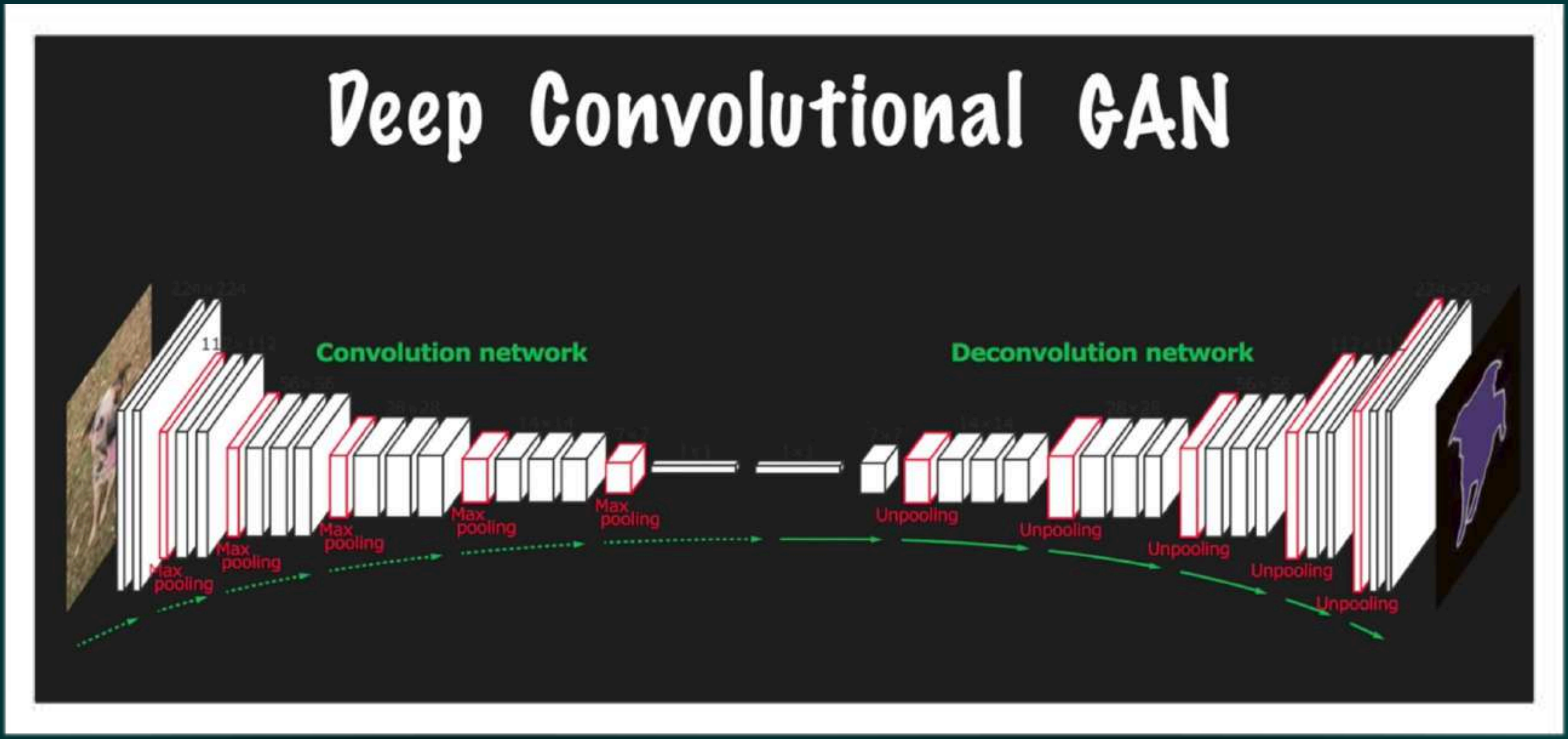
- Siraj's Image Generation
- Autoencoders
- **Generative Adversarial Networks**
- Siraj's Video Generation
- DCGANs
- Siraj's One Shoot Learning
- Generate Faces

In last part you will implement GAN with Mat Leonard to generate new digit images based on MNIST.

Part 4 - Generative Adversarial Networks

- Siraj's Image Generation
- Autoencoders
- Generative Adversarial Networks
- **Siraj's Video Generation**
- DCGANs
- Siraj's One Shoot Learning
- Generate Faces

Siraj will explain how Generative Adversarial Networks can be used for video generation frame by frame.

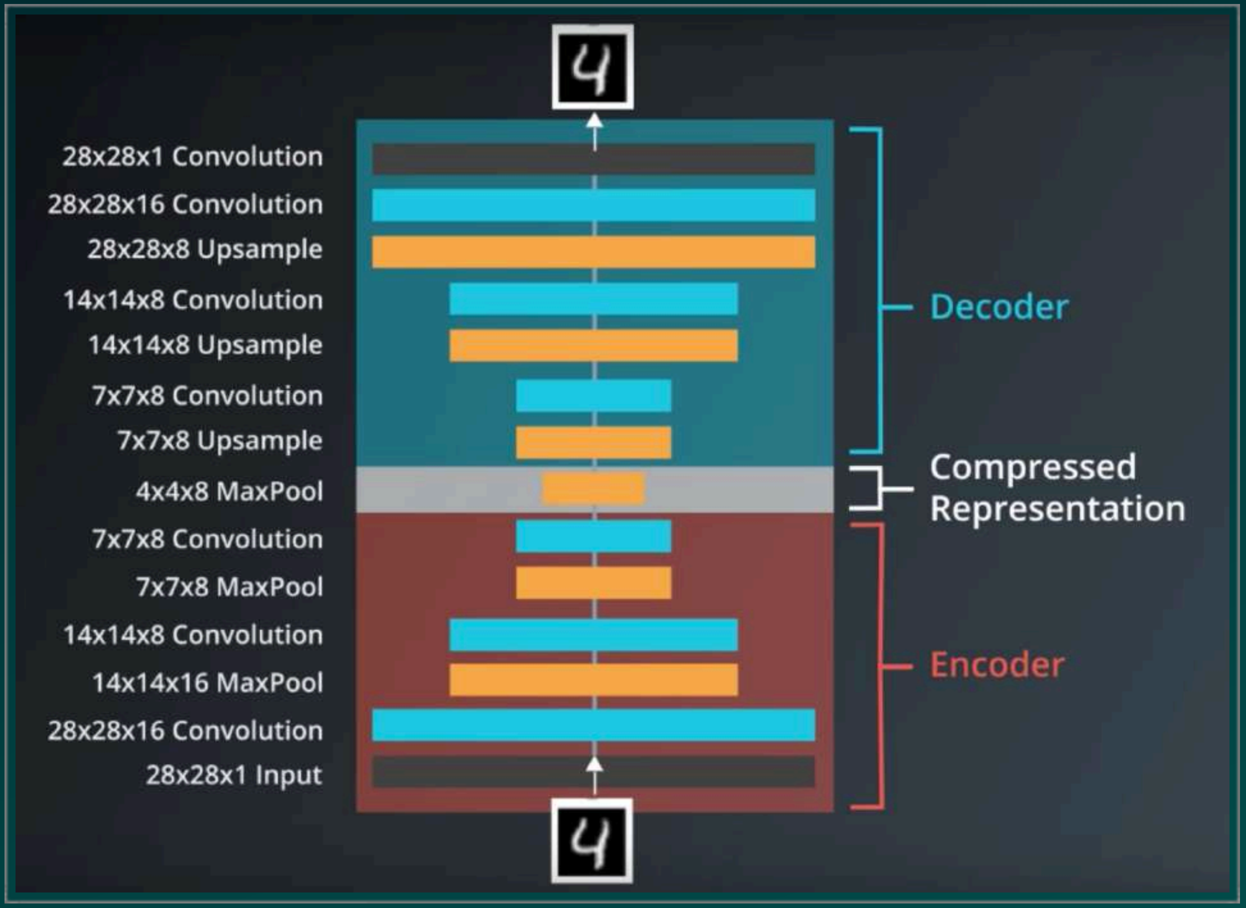


Part 4 - Generative Adversarial Networks

In this Jupyter Notebook exercise you will build Deep Convolutional Generative Adversarial Network to generate house numbers:

Part 4 - Generative Adversarial Networks

In this Jupyter Notebook exercise you will build Deep Convolutional Generative Adversarial Network to generate house numbers:



Part 4 - Generative Adversarial Networks

- Siraj's Image Generation
- Autoencoders
- Generative Adversarial Networks
- Siraj's Video Generation
- **DCGANs**
- Siraj's One Shoot Learning
- Generate Faces

In this Jupyter Notebook exercise you will build Deep Convolutional Generative Adversarial Network to generate house numbers.

Additionally there will be very thorough explanation of powerful regularization technique - Batch Normalization, which can be applied for MLP, CNN, RNN and GAN.

Part 4 - Generative Adversarial Networks

- Siraj's Image Generation
- Autoencoders
- Generative Adversarial Networks
- Siraj's Video Generation
- DCGANs
- **Siraj's One Shoot Learning**
- Generate Faces

In next 8 minutes video Siraj will talk about One-Shot learning on Neural Turing Machine (Memory Augmented Neural Network) model.

One-Shot-Learning is a method of training Neural Network with small amount of data.

One-shot Learning with Memory-Augmented Neural Networks

Adam Santoro ADAMSANTORO@GOOGLE.COM
Google DeepMind

Sergey Bartunov SBOS@SBOS.IN
Google DeepMind, National Research University Higher School of Economics (HSE)

Matthew Botvinick BOTVINICK@GOOGLE.COM
Daan Wierstra WIERSTRA@GOOGLE.COM
Timothy Lillicrap COUNTZERO@GOOGLE.COM
Google DeepMind

Abstract

Despite recent breakthroughs in the applications of deep neural networks, one setting that presents

quire rapid inference from small quantities of data. In the limit of "one-shot learning," single observations should result in abrupt shifts in behavior.

Part 4 - Generative Adversarial Networks

- Siraj's Image Generation
- Autoencoders
- Generative Adversarial Networks
- Siraj's Video Generation
- DCGANs
- Siraj's One Shoot Learning
- **Generate Faces**

Project no. 5

Deadline: 4 weeks

You will build Deep Convolutional Generative Adversarial Network in Tensorflow.

You will use CelebA dataset (200k celebrity images), to teach Discriminator to recognise human faces and guide Generator how to positively pass judgement of Discriminator.

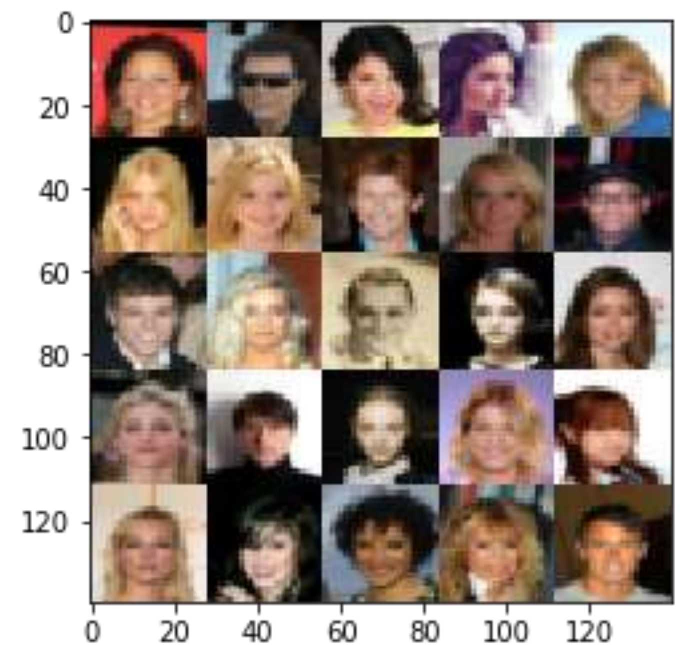
You are given rubric with requirements your project has to meet in order to pass.

Project undergoes code review by specialist and you receive feedback with links/hints what was done well or could be done better.

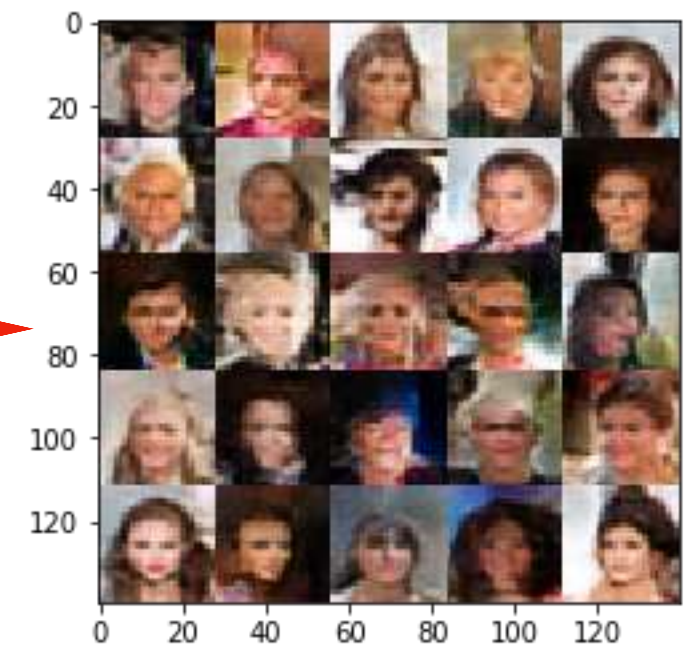
Part 4 - Generative Adversarial Networks

- Siraj's Image Generation
- Autoencoders
- Generative Adversarial Networks
- Siraj's Video Generation
- DCGANs
- Siraj's One Shoot Learning
- **Generate Faces**

Real images, downscaled to 28x28



Generated images



Results would be better:

- if you had more computation power because images could be larger then
- if you ran Network for larger time (this ran 8 hours on my CPU)

VERIFIED CERTIFICATE OF COMPLETION

August 5, 2017

UDACITY

Kamil Krzyk

Has successfully completed the

Deep Learning Nanodegree Foundation

NANODEGREE FOUNDATIONS PROGRAM

Sebastian Thrun
Founder, Udacity

Thank you for attention!