

Welcome to MOOC era!

My experiences with
Deep Learning Foundations Nanodegree at Udacity



Kamil Krzyk

Android Platform Engineer at



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 Chief 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

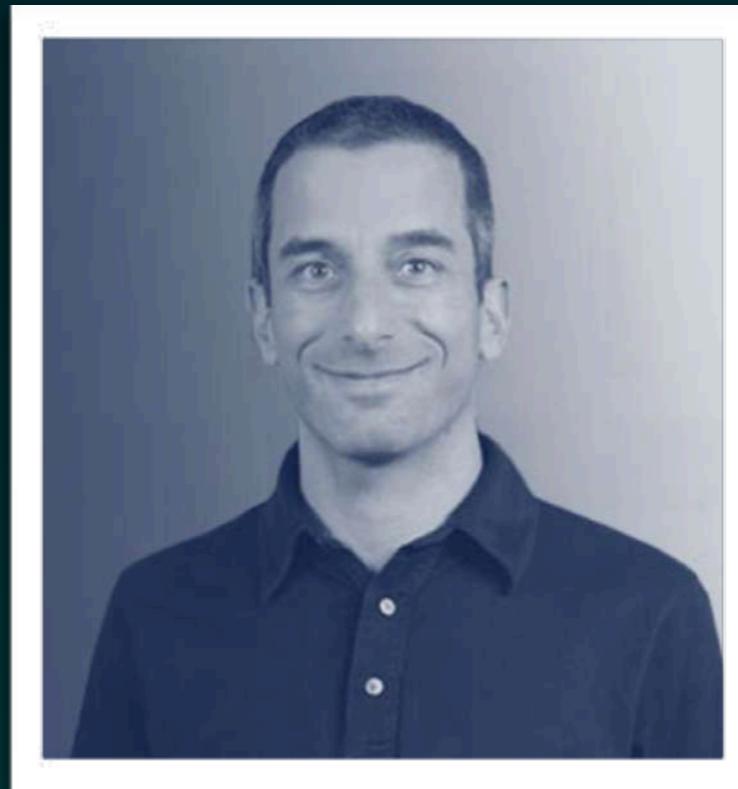
“It’s great that you have finished Deep Learning Nanodegree at Udacity. I think you should definitely apply for job in this area. This field is changing and you will have to keep learning anyway the same way as you do now.”



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

The screenshot shows a video player interface. On the right, a man with short hair and a dark t-shirt is speaking. To his left is a diagram illustrating the flow of a conversational AI system:

- 1. invoke:** A user interacts with a device (represented by a person icon and a microphone icon) which triggers an **invocation trigger**.
- 2. action responds:** The invocation trigger activates a **Conversation Action**, which then triggers a **fulfillment** process. This process includes **generate dialog output** and **process dialog input**.
- 3. user requests:** The fulfillment process sends a response back to the user.

Below the diagram is a flowchart showing the data path from the user to the fulfillment service:

- User:** Has **input methods** (keyboard, microphone) and **output methods** (speaker, screen).
- Your app/bot/device:** Receives input from the user and sends data to the API.AI.platform.
- API.AI.platform:** Receives **Actionable Data** and sends a **Query** to the Intent component. It also receives **Intent** data and sends **Actionable Data** back to the app.
- Your fulfillment service:** Receives **Intent** and **Actionable Data** from API.AI.platform, and sends **External API's** and **DB** data back to the platform.

At the bottom of the video player, there is a progress bar showing 1:20 / 8:34, and a row of control icons including play, pause, volume, and settings.

Create Your First Google Assistant App That Uses Your Server
9 411 wyświetleń 333 8 UDOSTĘPNIJ ...

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

The screenshot shows the Coursera website interface. At the top, there's a navigation bar with the Coursera logo, a 'Catalog' button, a search bar, and user profile options for 'For Enterprise' and 'Kamil'. A green banner below the navigation bar announces 'Unlimited access to 2000+ courses. Cancel anytime.' with a 'Get started' button. The main content area displays the 'Machine Learning' course by Andrew Ng. On the left, a sidebar provides links to 'Overview', 'Syllabus', 'FAQs', 'Creators', and 'Ratings and Reviews'. Below these is a large section for the course itself, which includes the title 'Machine Learning', a brief description about machine learning, and a 'More' link. It also lists 'Created by: Stanford University' with a Stanford University logo. At the bottom of the sidebar, there are buttons for 'Go to Course' and 'Already enrolled'.

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

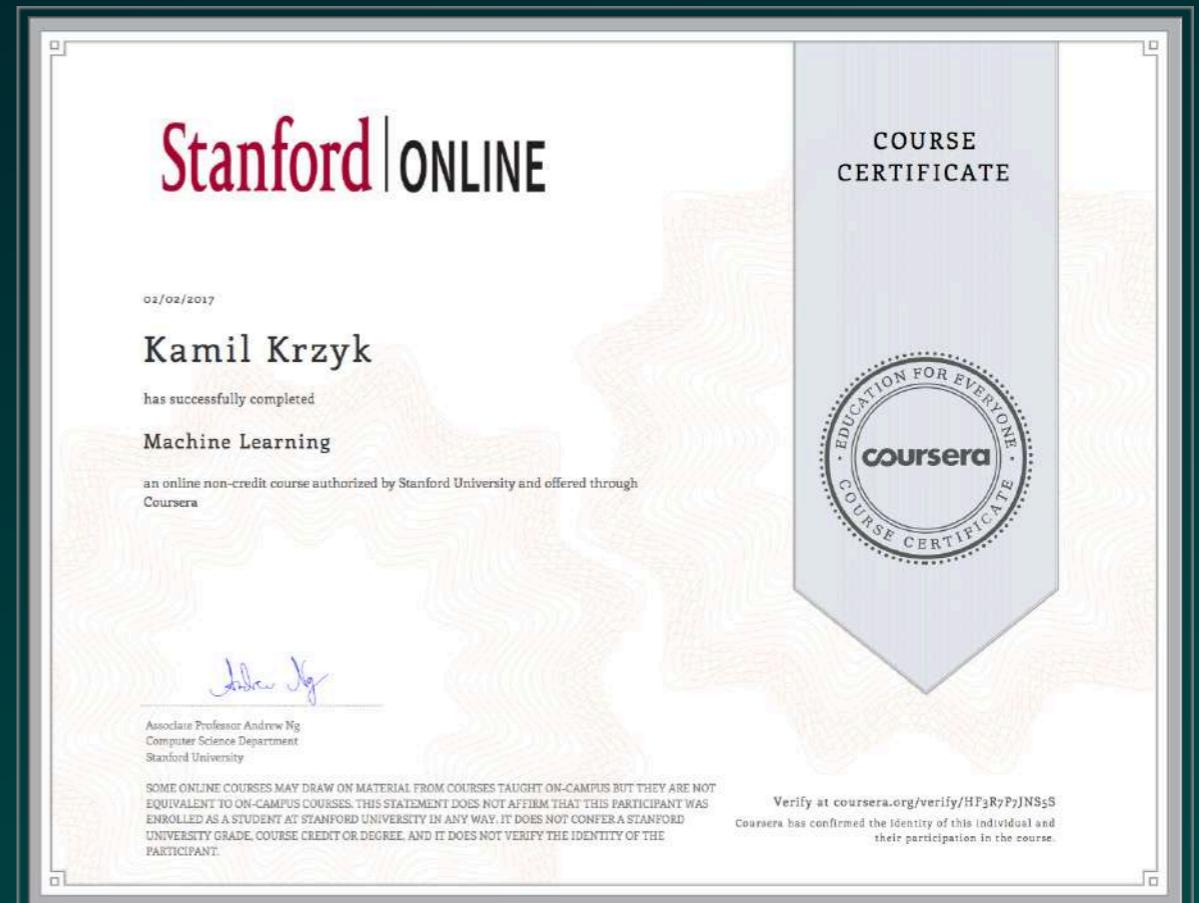
by Andrew Ng at Coursera

 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 ▾](#)



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 CC 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 CC 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 CC English

42 zł 595 zł
★★★★★ 4.6
(9,785 ratings)

Discounts everywhere... always!

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

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

BEST SELLER



4.5 (20,290 ratings) 130,425 students enrolled

Created by Kirill Eremenko, Hadelin de Ponteves, SuperDataScience Team Last updated 10/2017

English English [Auto-generated]



Preview This Course

42 zł 610 zł 93% off

⌚ 2 days left at this price!

Try to reset
your
cookies.

Buy Now

Add to Cart

30-Day Money-Back Guarantee

Includes:

- 40.5 hours on-demand video
- Top-responding instructor ⓘ
- 19 Articles
- 2 Supplemental Resources
- Full lifetime access
- Access on mobile and TV
- Certificate of Completion

What Will I Learn?

- Master Machine Learning on Python & R
- Make accurate predictions
- Make robust Machine Learning models
- Use Machine Learning for personal purpose
- Handle advanced techniques like Dimensionality Reduction
- Have a great intuition of many Machine Learning models
- Make powerful analysis
- Create strong added value to your business
- Handle specific topics like Reinforcement Learning, NLP and Deep Learning
- Know which Machine Learning model to choose for each type of problem

[View More](#)

How course looks like?

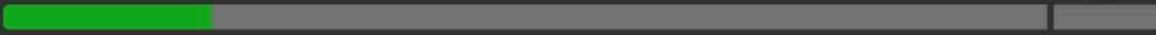
udemy Categories Search for Courses  Become an Instructor My Courses    



Data Science: Deep Learning in Python

[Continue to Lecture 14](#)

★★★★★ Edit Your Rating

12 of 60 items complete [Reset progress](#)  

Overview Course Content Q&A Bookmarks Announcements Options ▾

Search  Current Section All Sections All Resources

Section: 1 What is a neural network? 

Section: 2 Classification with the Outline method 

How course looks like?

Section: 3	6 / 6
Pandas	
▶ 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	6 / 6
Pandas	
▶ 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?

Visualizing what a neural network has learned usi...
Section 5, Lecture 42

Tinker With a **Neural Network** Right Here in Your Browser.
Don't Worry, You Can't Break It. We Promise.

Go to Dashboard

Iterations: 000,000 Learning rate: 0.03 Activation: Tanh Regularization: None Regularization rate: 0 Problem type: Classification

DATA
Which dataset do you want to use?

Ratio of training to test data: 50%
Noise: 0
Batch size: 10

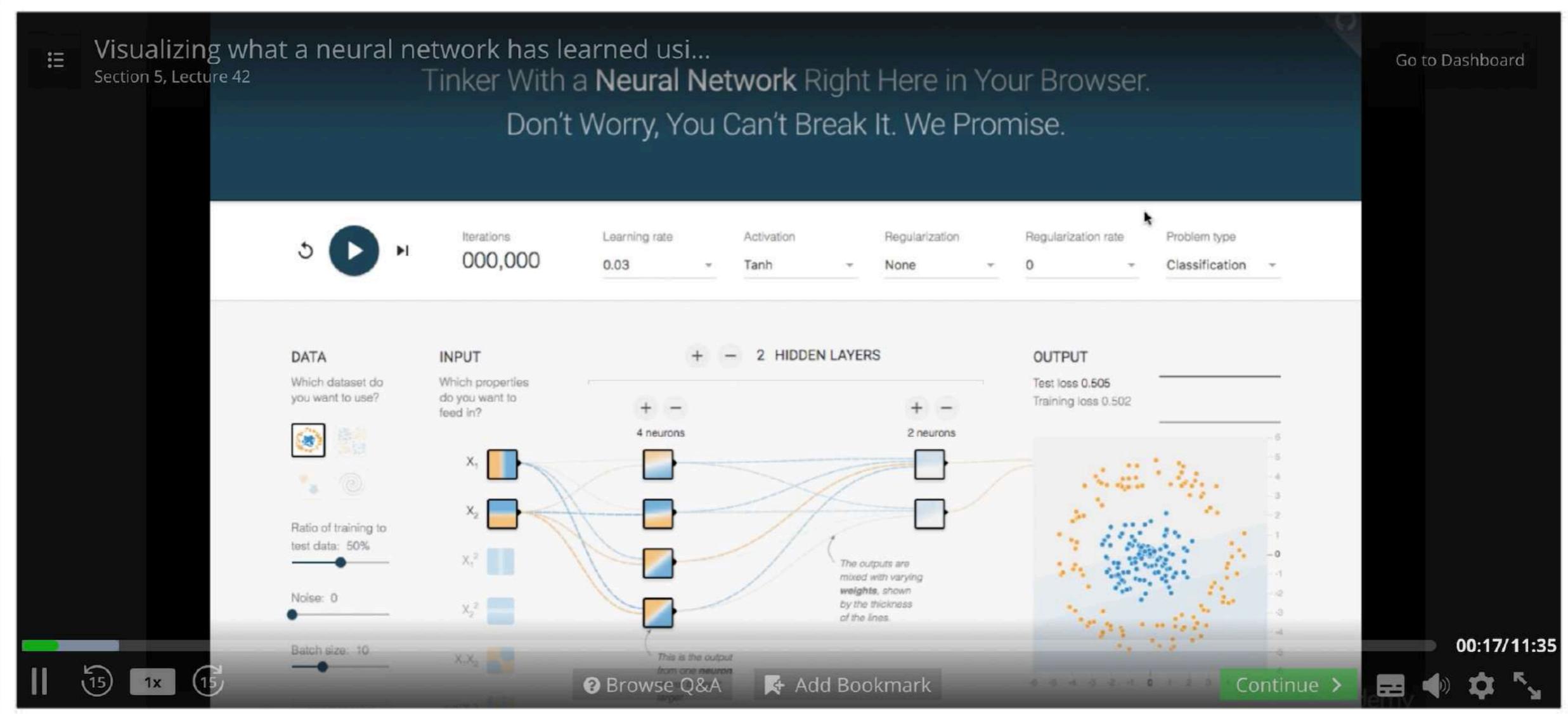
INPUT
Which properties do you want to feed in?
 X_1 , X_2 , X_1^2 , X_2^2 , $X_1 X_2$, $X_1^2 X_2^2$

2 HIDDEN LAYERS
+ - 4 neurons
+ - 2 neurons
The outputs are mixed with varying weights, shown by the thickness of the lines.
This is the output from one neuron

OUTPUT
Test loss 0.505
Training loss 0.502

00:17/11:35

|| 15x 15 Continue > Add Bookmark



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



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

My Experiences with Udacity

My Experiences with Udacity

UDACITY

Bookmarks Explore Nanodegree Catalog For Business | Blog **My Classroom**

Courses and Nanodegree Programs

Search

Category

- All
- Android
- Artificial Intelligence
- Data Analytics
- Data Science
- Deep Learning
- Developer Essentials
- Digital Marketing
- Georgia Tech Masters in CS
- iOS
- Machine Learning
- Mobile App Development
- Non-Tech
- Self Driving Car
- Software Engineering
- Virtual Reality
- Web Development

All Courses and Nanodegree Programs



Learn ARKit NEW

2 Projects

Put your skills to work in the exciting field of augmented reality! Learn the fundamentals of AR, build your very own AR app, and publish your app to the Apple Store.

IN COLLABORATION WITH **Unity**

110 



Intro to Self-Driving Cars NEW

8 Projects

This introductory program is the perfect way to start your journey.

180 



Data Foundations Nanodegree NEW

4 Projects

Beginner

My Experiences with Udacity



COURSE
Intro to Statistics

Get ready to analyze, visualize, and interpret data! Thought-provoking examples and chances to combine statistics and programming will keep you...

[VIEW COURSE](#)



COURSE
Technical Interview

Learn the skills technical interviewers expect you to know—efficiency, common algorithms, manipulating popular data structures, and how to explain a...

[VIEW COURSE](#)



COURSE
Intro Algebra Review

A brief review of introductory algebra topics including integer operations, scientific notation, algebraic expressions, linear equations, and...

[VIEW COURSE](#)



COURSE
Intro to Machine Learning

This class will teach you the end-to-end process of investigating data through a machine learning lens, and you'll apply what you've learned to a real-worl...

[VIEW COURSE](#)



COURSE
Linear Algebra Refresher Course

Learn linear algebra by doing: you will code your own library of linear algebra functions!

[VIEW COURSE](#)

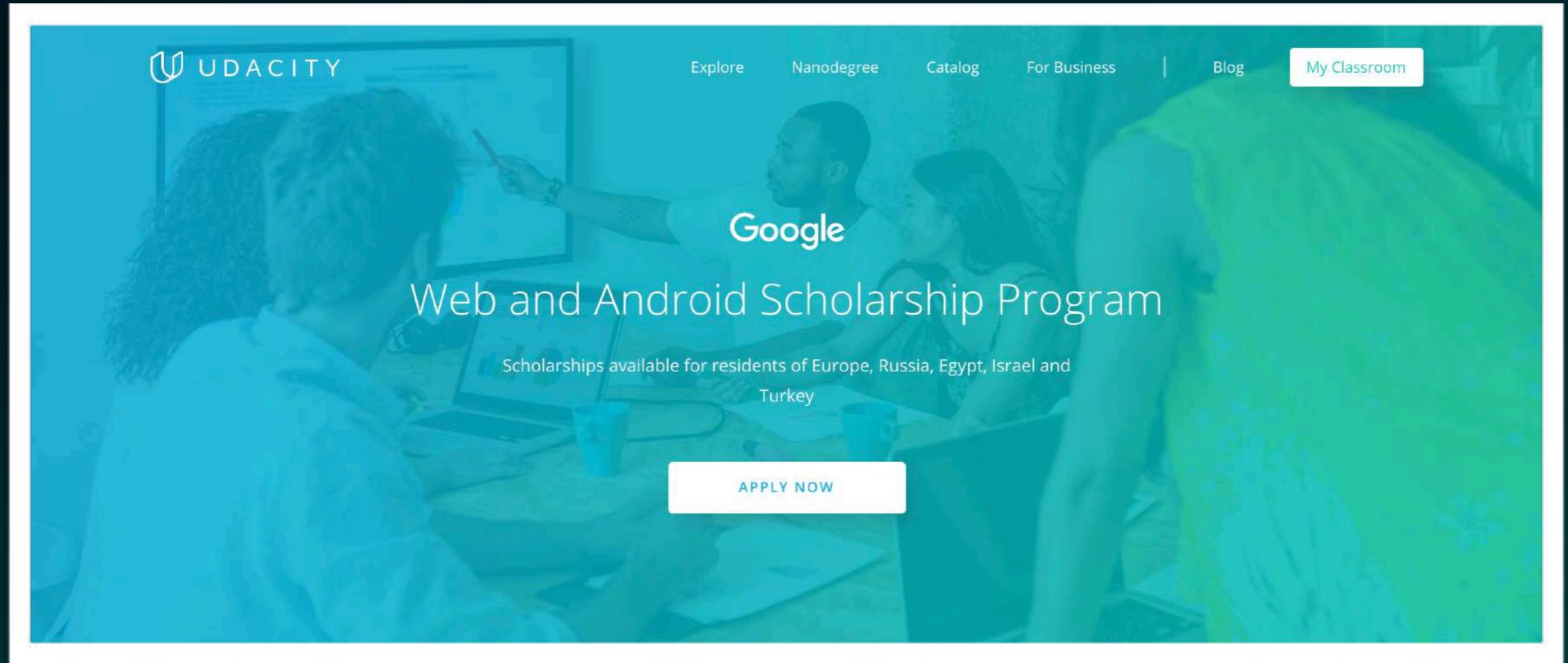


COURSE
Statistics

We live in a time of unprecedented access to information. You'll learn how to use statistics to interpret that information and make decisions.

[VIEW COURSE](#)

1 2



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

800\$, 4 months



600\$, 17 weeks

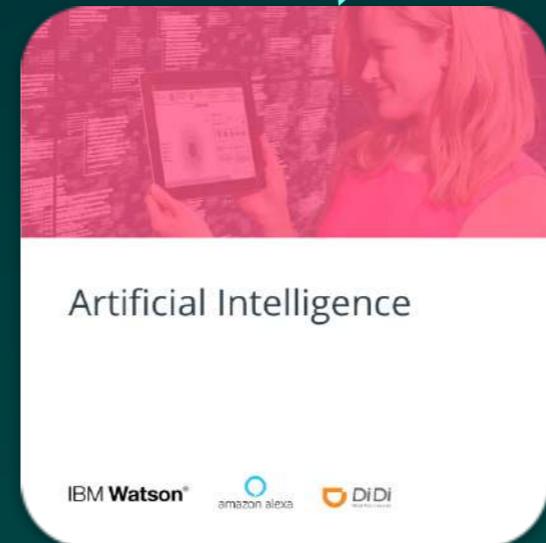


?

Granted seat,
200\$ discount

Granted seat,
200\$ discount

Granted seat,
200\$ discount



199\$/month,
in average 6 months

800\$ per term,
3 terms - 3 months each

800\$ per term,
2 terms - 3 months each

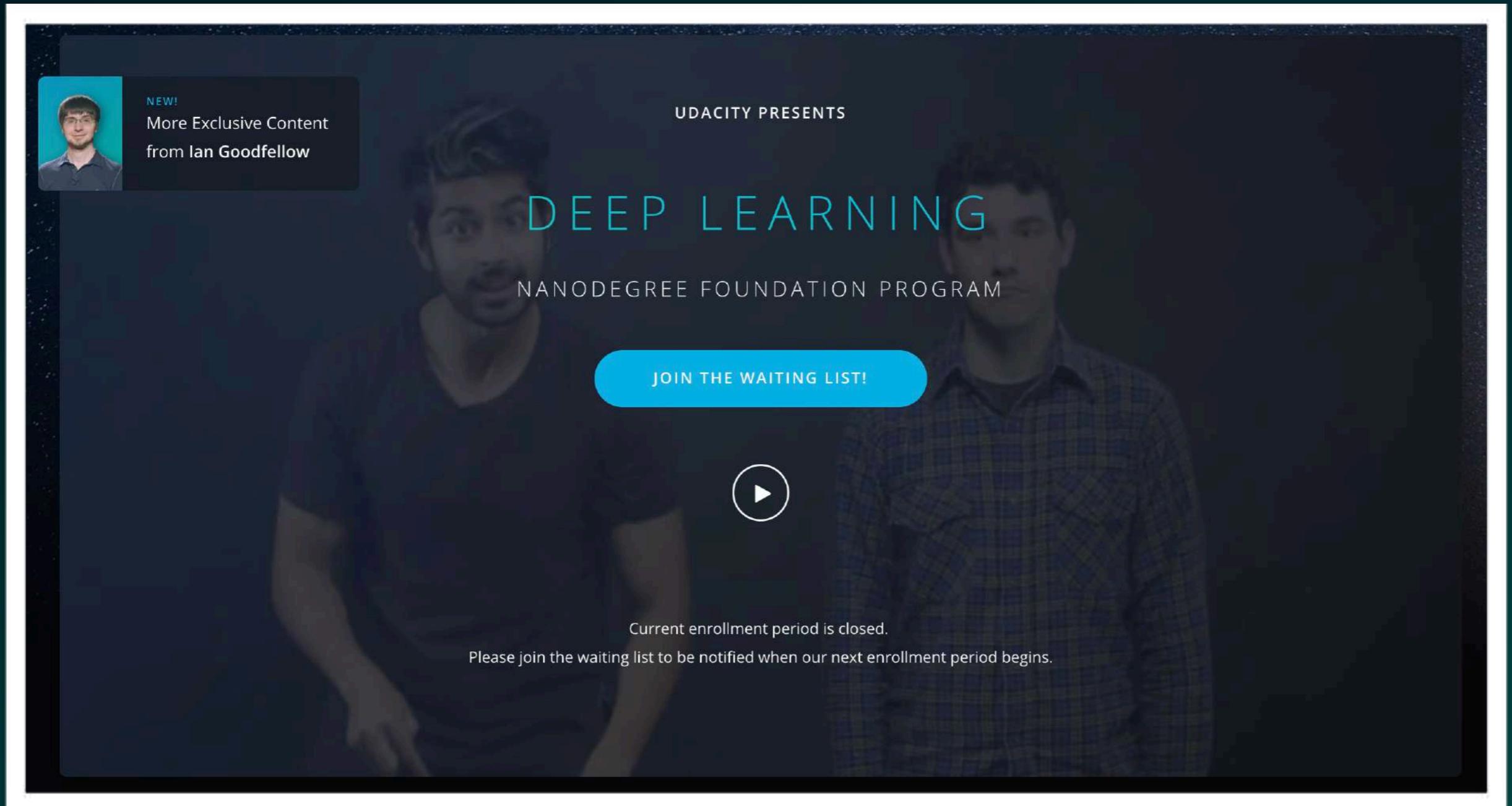
1200\$ per term,
2 terms - 3 months each

Incoming Nanodegrees at Udacity

The image shows the Udacity website homepage. At the top, there is a navigation bar with links for Explore, Nanodegree, Catalog, For Business, Blog, and My Classroom. The main feature is a large blue banner for the Flying Car Nanodegree Program. The banner includes the Udacity logo, a "NEW!" badge, and a brief description of the program: "Learn the latest in flying car technology and drones from the best in the field. Develop the software skills and conceptual understanding necessary to build an autonomous flight system for quadrotor and fixed-wing drones." Below the banner is a call-to-action button labeled "Be a pioneer!" with fields for "Email" and "KEEP ME UPDATED!".

Unconfirmed gossips about “Deep Learning for Cyber Security”

My experiences with Deep Learning Nanodegree at Udacity



The image shows the landing page for the Deep Learning Nanodegree Foundation Program. The background features a dark, slightly blurred photograph of two men, one with curly hair and one with short hair, looking towards the camera. In the top left corner, there is a small portrait of Ian Goodfellow with the text "NEW! More Exclusive Content from Ian Goodfellow". In the top right corner, it says "UDACITY PRESENTS". The center of the page has the title "DEEP LEARNING" in large, light blue capital letters, with "NANODEGREE FOUNDATION PROGRAM" in smaller white capital letters below it. A prominent blue button in the center contains the text "JOIN THE WAITING LIST!" in white. Below this button is a white play button icon. At the bottom of the page, a message reads: "Current enrollment period is closed. Please join the waiting list to be notified when our next enrollment period begins."

NEW!

More Exclusive Content
from Ian Goodfellow

UDACITY PRESENTS

DEEP LEARNING

NANODEGREE FOUNDATION PROGRAM

JOIN THE WAITING LIST!

Current enrollment period is closed.
Please join the waiting list to be notified when our next enrollment period begins.

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!

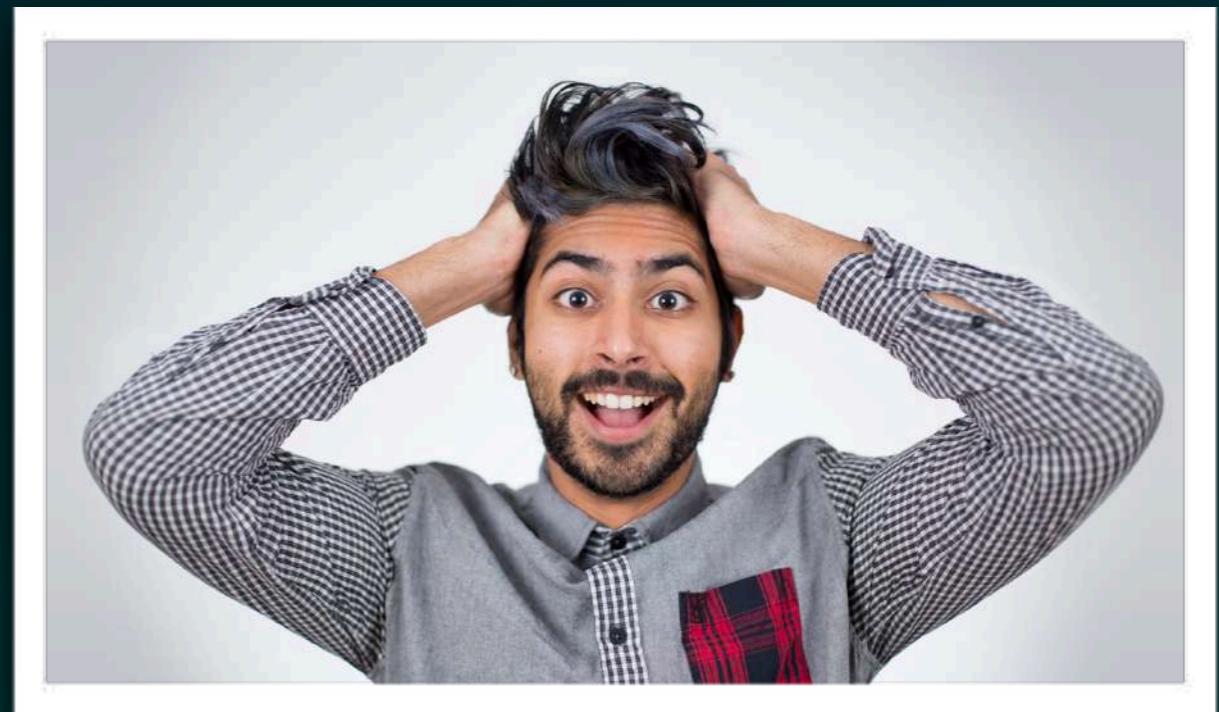
Most Seen Faces



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

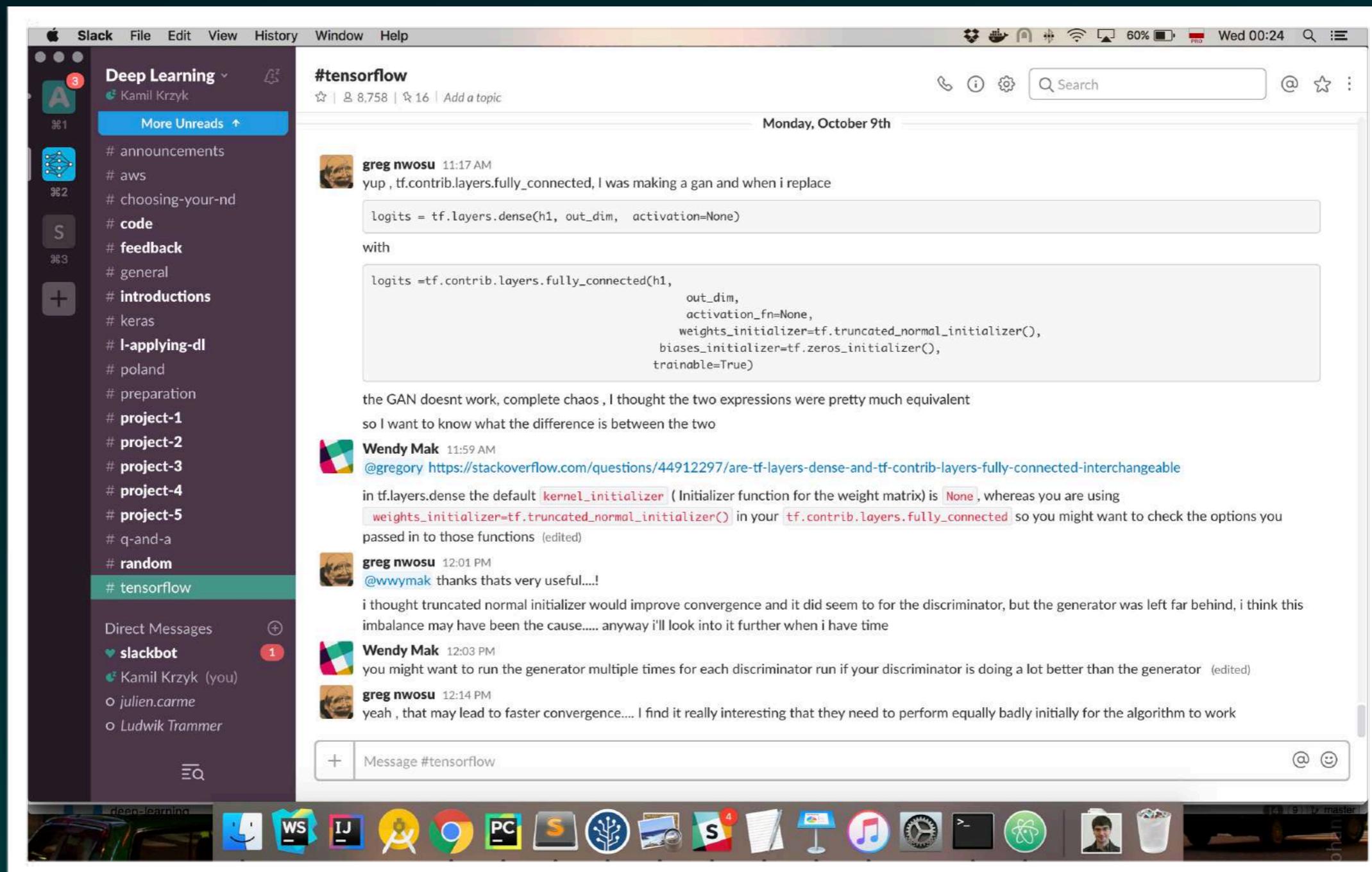
Research Scientist at Google Brain,
Doctoral Student at Oxford

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	# project 1
# feedback	# project 2
# general	# project 3
# random	# project 4
	# project 5

Access to Slack Channel

# announcements	# project 1	# january_class
# feedback	# project 2	# march_class
# general	# project 3	# june_class
# random	# project 4	# august_class
	# project 5	# ...

Access to Slack Channel

# announcements	# project 1	# january_class
# feedback	# project 2	# march_class
# general	# project 3	# june_class
# random	# project 4	# august_class
	# project 5	# ...
# tensorflow		
# keras		
# code		

Open DLFND GitHub Repo - 22 exercises/projects

udacity / deep-learning

Watch 180 Unstar 1,332 Fork 1,729

Code Issues 28 Pull requests 22 Projects 0 Wiki Insights

Repo for the Deep Learning Nanodegree Foundations program. <https://www.udacity.com/course/deep-l...>

320 commits 11 branches 0 releases 39 contributors MIT

Branch: master ▾ New pull request Create new file Upload files Find file Clone or download ▾

Icrucks committed on GitHub Merge pull request #211 from udacity/SmallWordChangesP1 ... Latest commit d949800 on 5 Sep

autoencoder	stride reduces the size by a factor	3 months ago
batch-norm	Fix some typos	5 months ago
dcgan-svhn	Fix overwritten parameter in GAN network	2 months ago
embeddings	Update Skip-Gram_word2vec.ipynb	3 months ago
environments	Fix tornado version in Mac environments	6 months ago
face_generation	Update dlnd_face_generation.ipynb	4 months ago
first-neural-network	Add small explanation changes for hyperparameters and a change to bac...	a month ago
gan_mnist	GAN_mnist: fix `generator` return doc	3 months ago
image-classification	Edit README and project notebook for new FloydHub dataset	2 months ago

Access to Classroom

The screenshot shows a web-based syllabus for the Deep Learning Foundation Nanodegree Program. The interface has a dark sidebar on the left and a main content area on the right.

Sidebar (Left):

- Home icon (blue)
- Deep Learning Foundation Nanodegree Program (highlighted in orange)
- Search icon (magnifying glass)
- Syllabus (selected)
- Core Curriculum (dropdown menu)
- 4. Generative Adversarial Networks (with a checkmark)
- Settings icon (gear)
- Logout icon (square with a left arrow)

Main Content Area (Right):

DEEP LEARNING FOUNDATION
Program Syllabus

RATE THIS PROGRAM

CORE CURRICULUM

Core Curriculum

This section consists of all the lessons and projects you need to complete in order to receive your certificate.

4 PARTS 5 PROJECTS

1 PART 1 Neural Networks

Neural network is the bedrock to deep learning. In this section, you'll learn how it works and test your ability by building a neural network from scratch.

2 PART 2 Convolutional Neural Networks

Convolutional neural network is the standard for solving vision problems.

Access to Classroom

The screenshot shows a web-based syllabus for the Deep Learning Foundation Nanodegree Program. On the left, a sidebar contains icons for profile, home, location, gear, and back. The main content area has a header: "Deep Learning Foundation Nanodegree Program". Below it are sections for "SYLLABUS" and "CORE CURRICULUM". The "SYLLABUS" section lists four modules: "1. Neural Networks", "2. Convolutional Neural Networks", "3. Recurrent Neural Networks", and "4. Generative Adversarial Networks", each with a green checkmark. The "CORE CURRICULUM" section is expanded, showing "DEEP LEARNING FOUNDATION Program Syllabus" and "RATE THIS PROGRAM" with five stars. The "CORE CURRICULUM" section itself has a title "Core Curriculum" with a red arrow pointing to it, a description, and icons for "4 PARTS" and "5 PROJECTS". It also lists two parts: "PART 1 Neural Networks" and "PART 2 Convolutional Neural Networks".

Deep Learning Foundation Nanodegree Program

SYLLABUS

CORE CURRICULUM

1. Neural Networks ✓

2. Convolutional Neural Networks ✓

3. Recurrent Neural Networks ✓

4. Generative Adversarial Networks ✓

DEEP LEARNING FOUNDATION Program Syllabus

RATE THIS PROGRAM ★★★★★

Core Curriculum

This section consists of all the lessons and projects you need to complete in order to receive your certificate.

4 PARTS

5 PROJECTS

PART 1

Neural Networks

Neural network is the bedrock to deep learning. In this section, you'll learn how it works and test your ability by building a neural network from scratch.

PART 2

Convolutional Neural Networks

Convolutional neural network is the standard for solving vision problems.

Access to Classroom

The screenshot shows a user interface for a learning program. On the left, there's a vertical sidebar with icons for profile, home, location, settings, and back/forward navigation. The main content area has a header for the "Deep Learning Foundation Nanodegree Program". It features a "SYLLABUS" section on the left with a dropdown menu for "CORE CURRICULUM" containing four items: "1. Neural Networks", "2. Convolutional Neural Networks", "3. Recurrent Neural Networks", and "4. Generative Adversarial Networks", each with a green checkmark. The right side has a "CORE CURRICULUM" section with a large box titled "Core Curriculum". This box contains a sub-section "PART 1 Neural Networks" with a description: "Neural network is the bedrock to deep learning. In this section, you'll learn how it works and test your ability by building a neural network from scratch." Below it is "PART 2 Convolutional Neural Networks" with the description: "Convolutional neural network is the standard for solving vision problems." To the right of the "Core Curriculum" box are two icons: "4 PARTS" and "5 PROJECTS". At the top right of the main content area, there's a "RATE THIS PROGRAM" button with five stars.

Deep Learning Foundation
Nanodegree Program

SYLLABUS

CORE CURRICULUM

1. Neural Networks ✓

2. Convolutional Neural Networks ✓

3. Recurrent Neural Networks ✓

4. Generative Adversarial Networks ✓

DEEP LEARNING FOUNDATION
Program Syllabus

RATE THIS PROGRAM ★★★★★

Core Curriculum

This section consists of all the lessons and projects you need to complete in order to receive your certificate.

4 PARTS

5 PROJECTS

PART 1

Neural Networks

Neural network is the bedrock to deep learning. In this section, you'll learn how it works and test your ability by building a neural network from scratch.

PART 2

Convolutional Neural Networks

Convolutional neural network is the standard for solving vision problems.

Access to Classroom

The screenshot shows a user interface for a learning program. On the left, a sidebar contains icons for profile, home, and search, followed by sections for 'SYLLABUS' and 'CORE CURRICULUM'. Under 'SYLLABUS', there are four items: '1. Neural Networks', '2. Convolutional Neural Networks', '3. Recurrent Neural Networks', and '4. Generative Adversarial Networks', each with a green checkmark. Under 'CORE CURRICULUM', the 'Core Curriculum' section is selected, showing a summary and two parts: 'PART 1 Neural Networks' and 'PART 2 Convolutional Neural Networks'. The 'Core Curriculum' summary includes text about completing lessons and projects for a certificate, and icons for '4 PARTS' and '5 PROJECTS'. The main content area has a header 'DEEP LEARNING FOUNDATION Program Syllabus' and a rating section 'RATE THIS PROGRAM' with five stars.

Deep Learning Foundation
Nanodegree Program

SYLLABUS

CORE CURRICULUM

1. Neural Networks ✓

2. Convolutional Neural Networks ✓

3. Recurrent Neural Networks ✓

4. Generative Adversarial Networks ✓

DEEP LEARNING FOUNDATION
Program Syllabus

RATE THIS PROGRAM ★★★★★

CORE CURRICULUM

Core Curriculum

This section consists of all the lessons and projects you need to complete in order to receive your certificate.

4 PARTS

5 PROJECTS

PART 1

Neural Networks

Neural network is the bedrock to deep learning. In this section, you'll learn how it works and test your ability by building a neural network from scratch.

PART 2

Convolutional Neural Networks

Convolutional neural network is the standard for solving vision problems.

Access to Classroom

The screenshot shows a web-based syllabus for the Deep Learning Foundation Nanodegree Program. On the left, a sidebar contains icons for profile, home, location, gear, and back. The main content area has a header "Deep Learning Foundation Nanodegree Program". Below it, "DEEP LEARNING FOUNDATION Program Syllabus" is displayed, along with a "RATE THIS PROGRAM" section showing five stars. The sidebar includes sections for "SYLLABUS" and "CORE CURRICULUM". The "CORE CURRICULUM" section lists four parts: 1. Neural Networks, 2. Convolutional Neural Networks, 3. Recurrent Neural Networks, and 4. Generative Adversarial Networks, each with a green checkmark. A large red arrow points from the sidebar towards the "CORE CURRICULUM" section. The main content area also features a "CORE CURRICULUM" section with a title "Core Curriculum", a description about completing lessons and projects for a certificate, and two parts: "PART 1 Neural Networks" and "PART 2 Convolutional Neural Networks". Each part has a description and a count of "4 PARTS" and "5 PROJECTS".

Deep Learning Foundation Nanodegree Program

DEEP LEARNING FOUNDATION Program Syllabus

RATE THIS PROGRAM ★★★★★

SYLLABUS

CORE CURRICULUM

1. Neural Networks ✓

2. Convolutional Neural Networks ✓

3. Recurrent Neural Networks ✓

4. Generative Adversarial Networks ✓

Core Curriculum

This section consists of all the lessons and projects you need to complete in order to receive your certificate.

4 PARTS

5 PROJECTS

PART 1

Neural Networks

Neural network is the bedrock to deep learning. In this section, you'll learn how it works and test your ability by building a neural network from scratch.

PART 2

Convolutional Neural Networks

Convolutional neural network is the standard for solving vision problems.

Part 1 - Neural Networks

1

Welcome

Welcome to the Deep Learning Nanodegree Foundations Program! In this lesson, you'll meet your instructors, find out about the field of Deep Learning, and learn how to make the most of the resources Udacity...

[VIEW >](#)

 100% VIEWED

[SHRINK CARD](#)



2

Applying Deep Learning

✓ COMPLETED

3

Anaconda

✓ COMPLETED

4

Jupyter Notebooks

✓ COMPLETED

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

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.

A diagram illustrating a neural network architecture. It consists of four horizontal layers of nodes. The first layer, on the left, is labeled "input layer" and contains three red-outlined circles. The second layer contains four blue-outlined circles. The third layer contains three green-outlined circles. The fourth layer, on the right, is labeled "output layer" and contains one yellow-outlined circle. Every node in one layer is connected by arrows to every node in the subsequent layer. The connections are represented by black lines with arrowheads pointing from left to right.

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

Saturday, September 30th



tharun gowrishankar 8:06 AM

uploaded this image: [StyleTrasfer_rain_princess.jpg](#) ▾



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

DeepTraffic

Main Page - Leaderboard - About DeepTraffic

Americans spend 8 billion hours stuck in traffic every year.
Deep neural networks can help!

```
1 //<![CDATA[  
2 // a few things don't have var in front of them - they update already  
3 // existing variables the game needs  
4 lanesSide = 0;  
5 patchesAhead = 1;  
6 patchesBehind = 0;  
7 trainIterations = 10000;  
8  
9  
10 var num_inputs = (lanesSide * 2 + 1) * (patchesAhead + patchesBehind);  
11 var num_actions = 5;  
12 var temporal_window = 3;  
13 var network_size = num_inputs * temporal_window + num_actions *  
14
```

Apply Code/Reset Net Save Code/Net to File Load Code/Net from File Submit Model to Competition

Run Training Start Evaluation Run

Part 1 - Neural Networks

Welcome

Applying Deep Learning

Anaconda

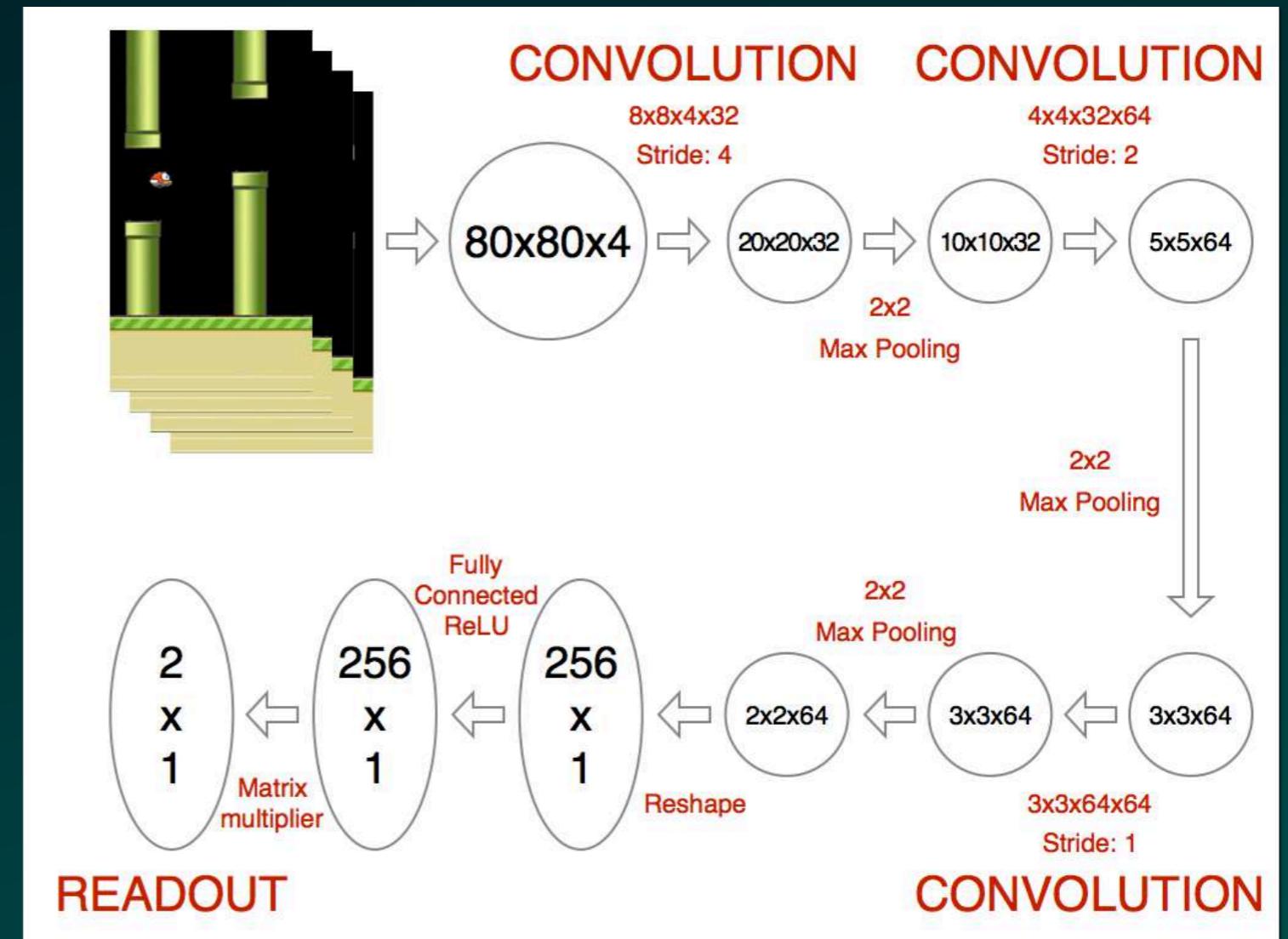
Jupyter Notebooks

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

Welcome

Applying Deep Learning

Anaconda

Jupyter Notebooks

Regression

Matrix Math and NumPy
Refresher

Your First Neural Network

Intro To 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

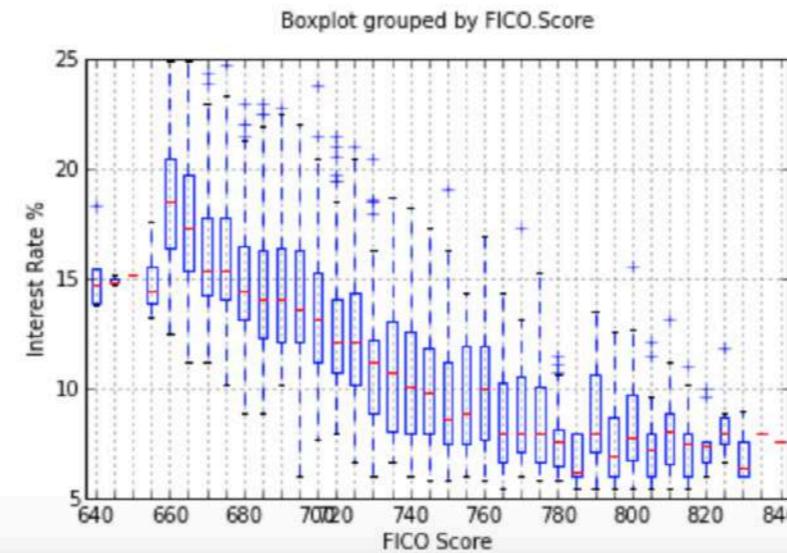
Box Plot

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', '840'])

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

```
gapminder1.py bmi_and_life_expectancy.csv solution.py
1 # TODO: Add import statements
2 import pandas as pd
3 import numpy as np
4 from sklearn import linear_model
5
6 # Assign the dataframe to this variable.
7 # TODO: Load the data
8 bmi_life_data = pd.read_csv("bmi_and_life_expectancy.csv")
9 life_expectancy_values = np.array(bmi_life_data["Life expectancy"]).reshape(-1, 1)
10 bmi_values = np.array(bmi_life_data["BMI"]).reshape(-1, 1)
11
12 # Make and fit the linear regression model
13 #TODO: Fit the model and Assign it to bmi_life_model
14 bmi_life_model = linear_model.LinearRegression()
15 bmi_life_model.fit(bmi_values, life_expectancy_values)
16
17 # Mak a prediction using the model
18 # TODO: Predict life expectancy for a BMI value of 21.07931
19 laos_life_exp = bmi_life_model.predict(21.07931)
20
```

RESET QUIZ

TEST RUN

SUBMIT ANSWER

Part 1 - Neural Networks

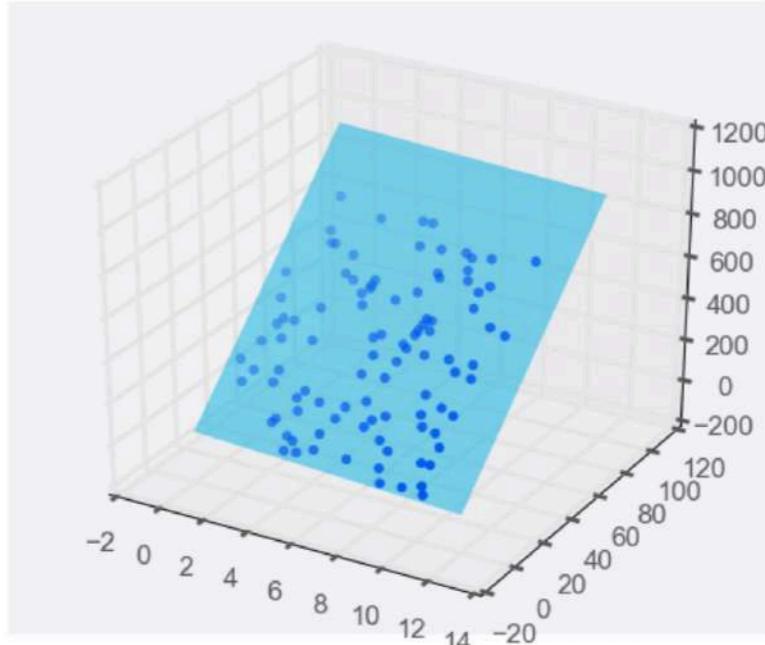
Lesson 5:
Regression

- 1. Welcome to Week One
- 2. Preparing for Siraj's video
- 3. Siraj's Intro to Deep Learning
- 4. Linear Regression
- 5. Linear Regression Warnings
- 6. Multiple Linear Regression
- 7. Siraj's Live Session

Multiple Linear Regression

$$y = m_1x_1 + m_2x_2 + b$$

To represent this graphically, we'll need a three-dimensional plot, with the linear regression model represented as a plane:



Linear regression with two predictor variables

You can use more than two predictor variables, in fact you should use as many as is useful! If you use n predictor variables, then the model can be represented by the equation

$$y = m_1x_1 + m_2x_2 + m_3x_3 + \dots + m_nx_n + b$$

As you make a model with more predictor variables, it becomes harder to visualise, but luckily, everything else about linear regression stays the same. We can still fit models and make predictions in exactly the same way - time to try it!

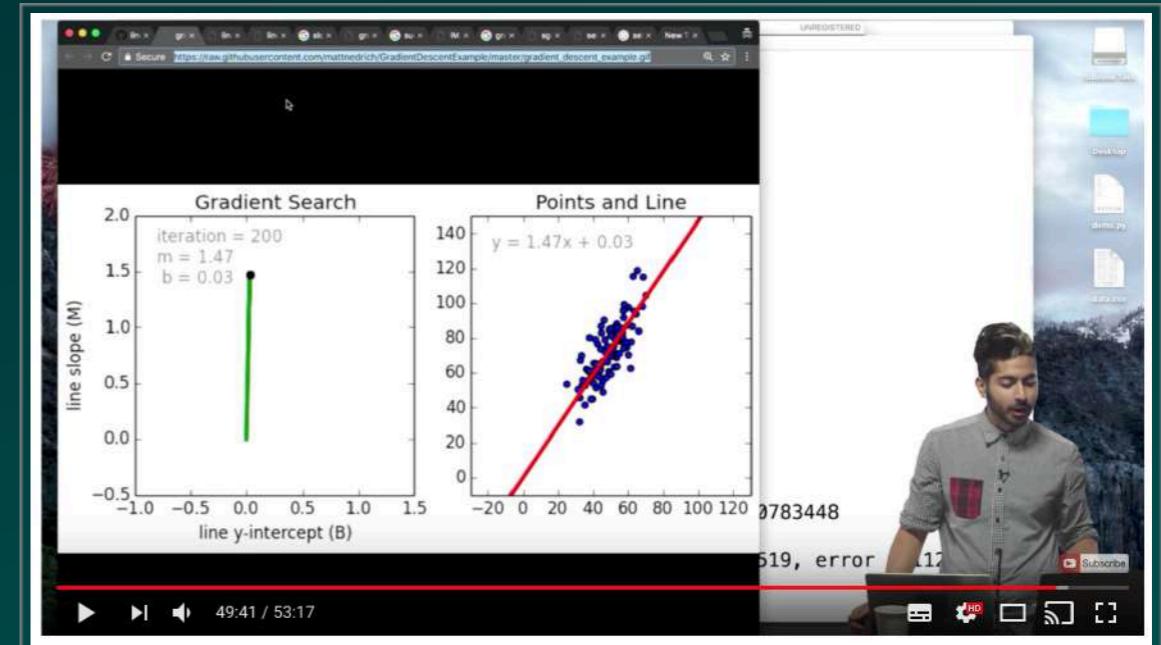
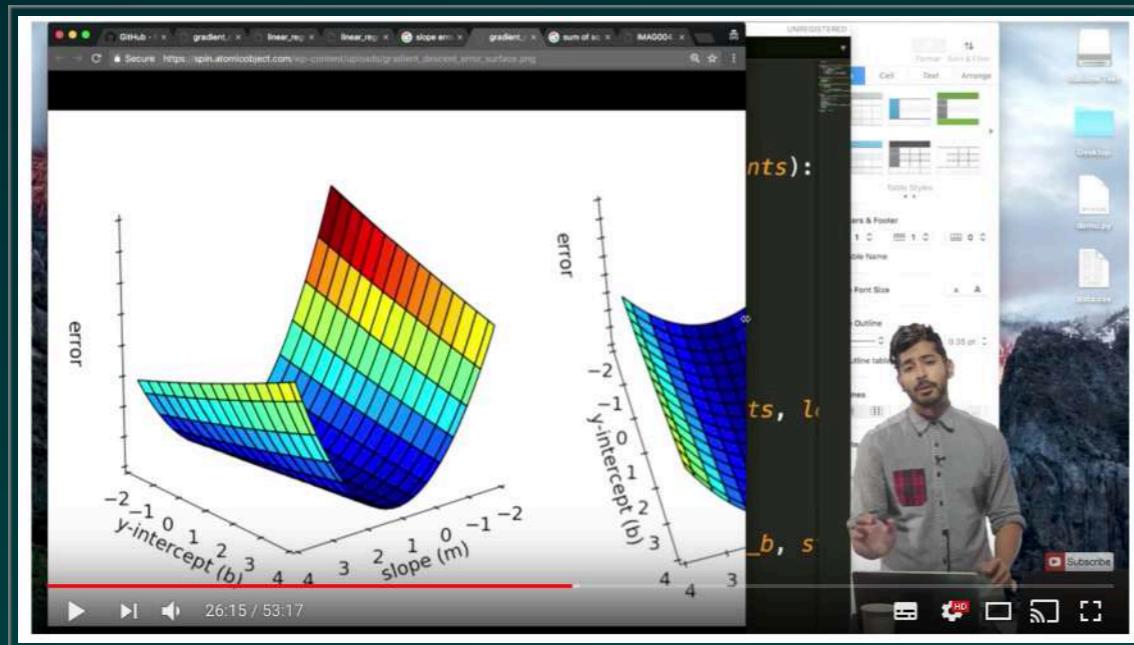
Part 1 - Neural Networks

A screenshot of a video player interface. The video frame shows a man speaking. The formula displayed is:

$$\text{Error}_{(m,b)} = \frac{1}{N} \sum_{i=1}^N (y_i - (mx_i + b))^2$$

A screenshot of a video player interface. The video frame shows a man speaking. The code displayed is:

```
1 from numpy import *
2
3
4 def compute_error_for_given_points(b, m, points):
5     totalError = 0
6     for i in range(0, len(points)):
7         x = points[i, 0]
8         y = points[i, 1]
9         totalError += (y - (m * x + b)) **2
10    return totalError / float(len(points))
11
12
13 def step_gradient(b_current, m_current, points, learningRate):
14     b_gradient = 0
15     m_gradient = 0
16     N = float(len(points))
17     for i in range(0, len(points)):
18         x = points[i, 0]
19         y = points[i, 1]
20
21     def gradient_descent_runner(points, starting_b, starting_m,
22         b = starting_b
23         m = starting_m
24         for i in range(num_iterations):
```



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

- 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

Welcome

Applying Deep Learning

Anaconda

Jupyter Notebooks

Regression

Matrix Math and NumPy
Refresher

Intro To Neural Networks

Your First Neural Network

- 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

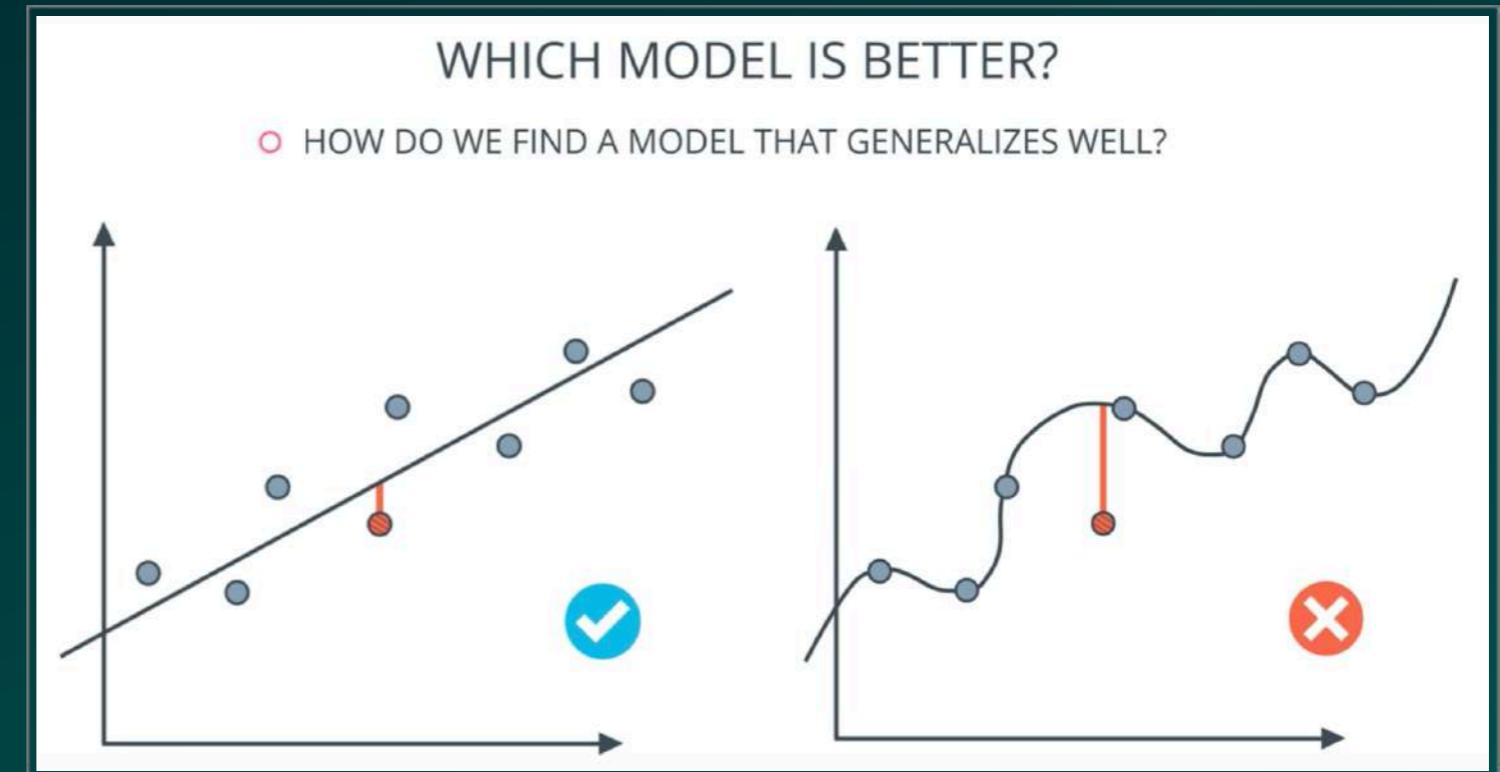
Convolutional Networks

Siraj's Image Classification

Weight Initialization

...

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



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 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.

CONFUSION MATRIX



1000 EMAILS

		SPAM	
		Spam Folder	Inbox
EMAIL	Spam	100 True Positives	170 False Negatives
	Not Spam	30 False Positives	700 True Negatives

Part 2 - Convolutional Neural Networks

Model Evaluation and Validation

MiniFlow

Sentiment Analysis with Andrew Trask

Intro to TensorFlow

Cloud Computing

Deep Neural Networks

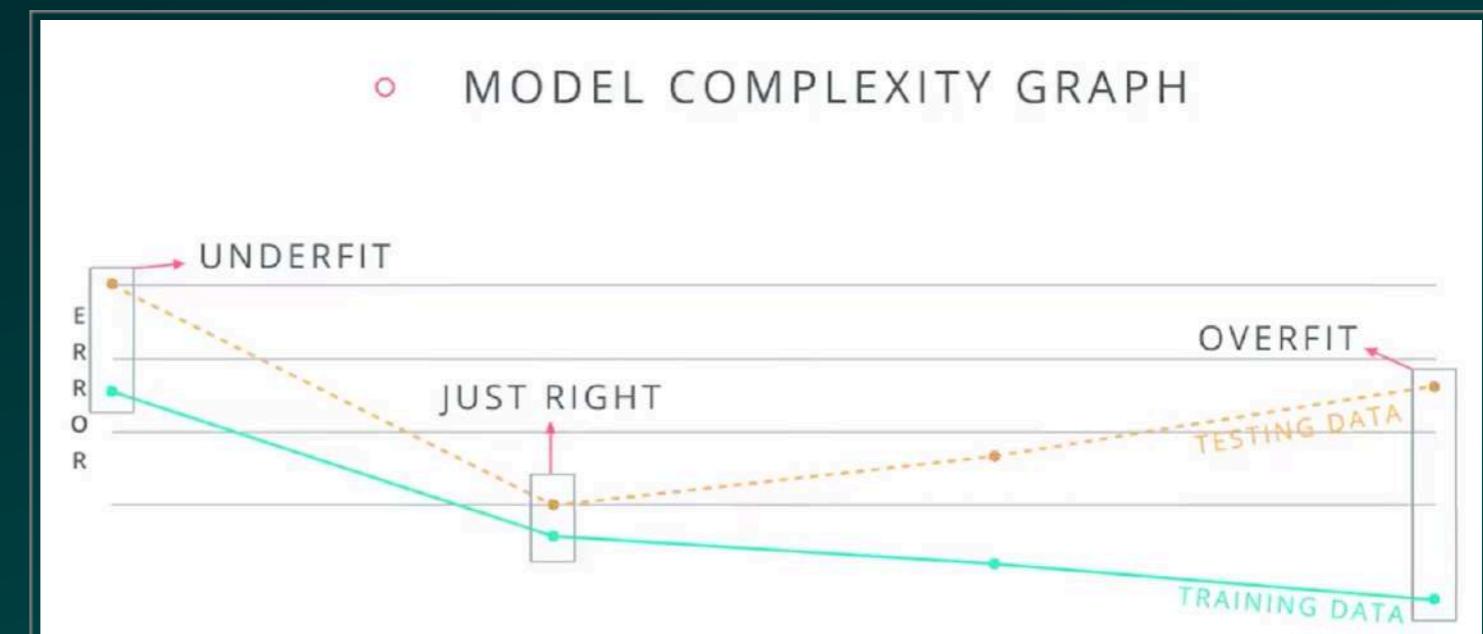
Convolutional Networks

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

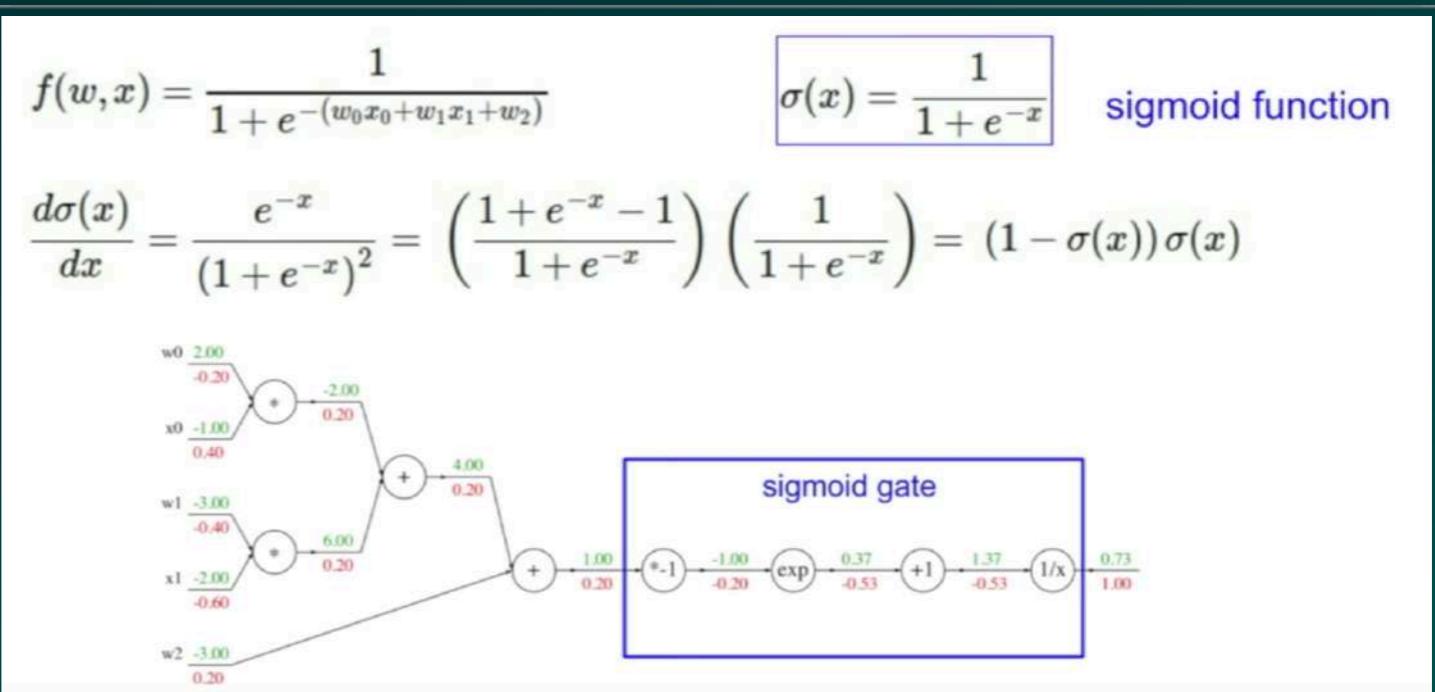
Convolutional Networks

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

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 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 outgoing 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

...

```
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

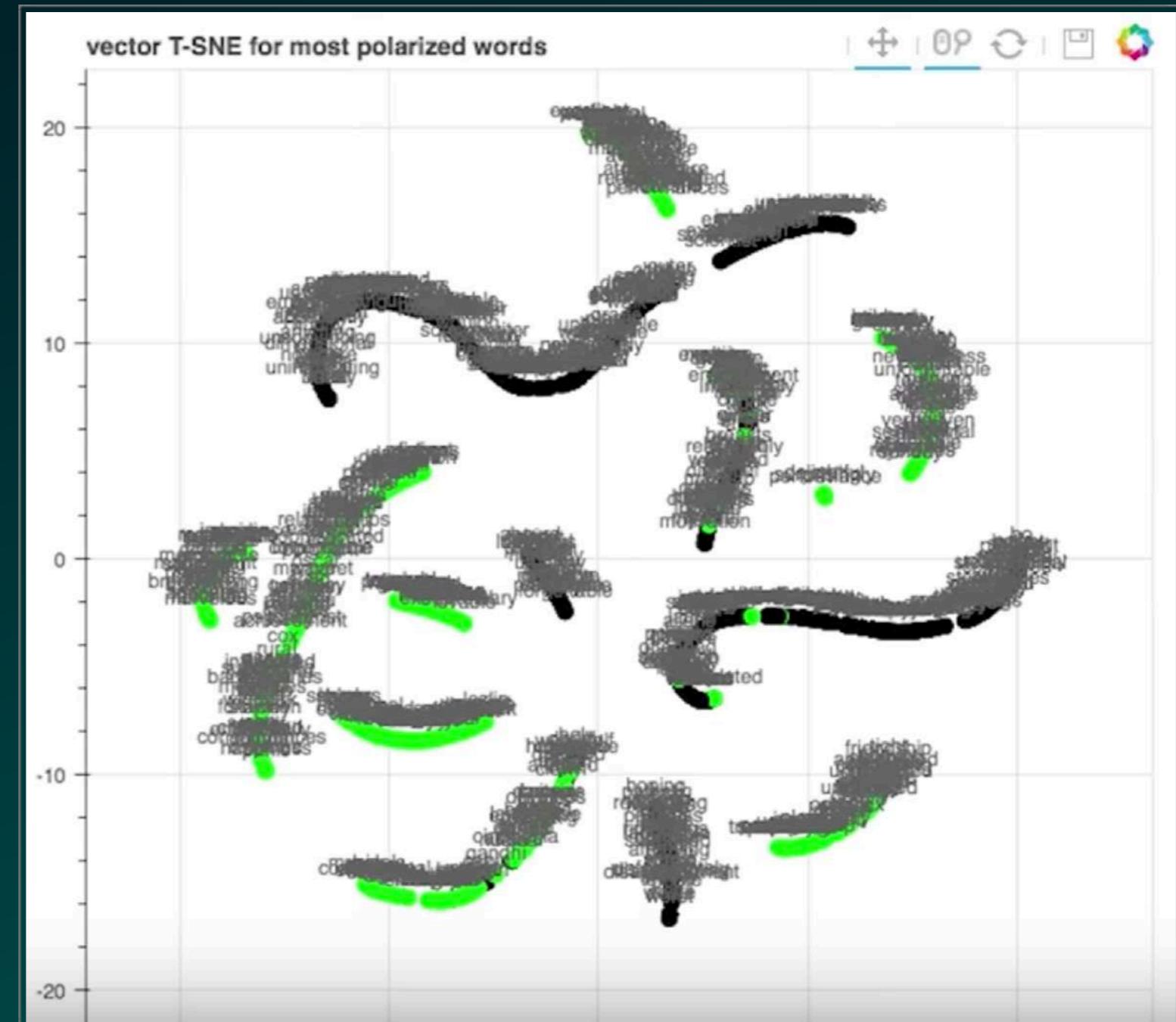
Deep Neural Networks

Convolutional Networks

Siraj's Image Classification

Weight Initialization

...



Part 2 - Convolutional Neural Networks

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

...

Lesson with Vincent Vanhoucke, Principal Scientist at Google Brain.

You will learn and implement:

- what is Deep Learning, what is it used for, what accelerated its 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

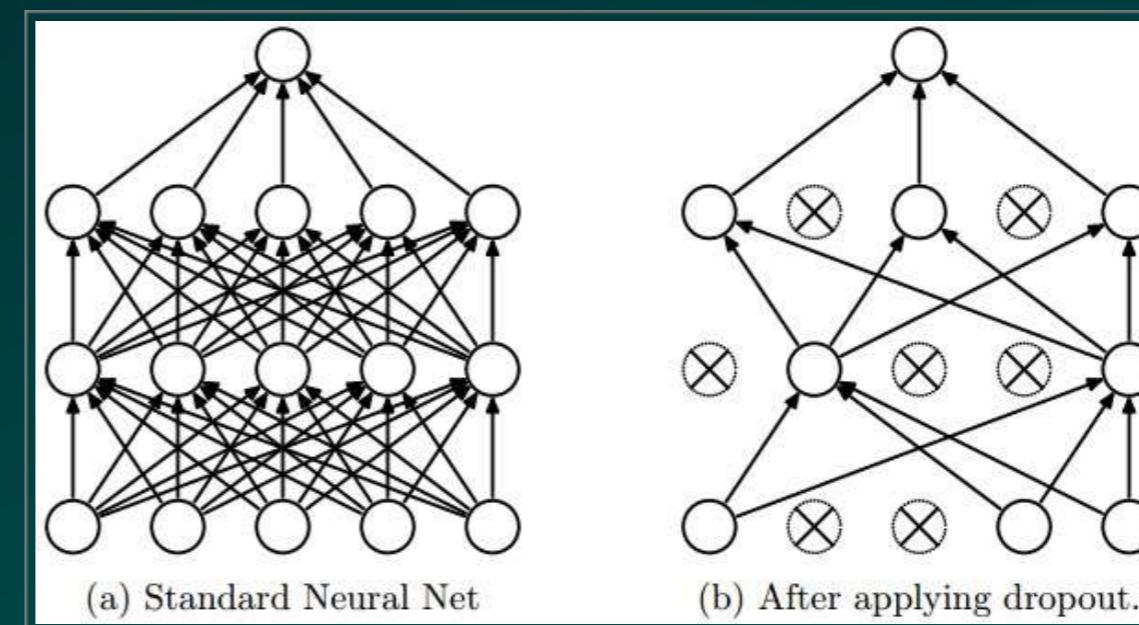
Convolutional Networks

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

Model Evaluation and Validation

MiniFlow

Sentiment Analysis with Andrew Trask

Intro to TensorFlow

Cloud Computing

Deep Neural Networks

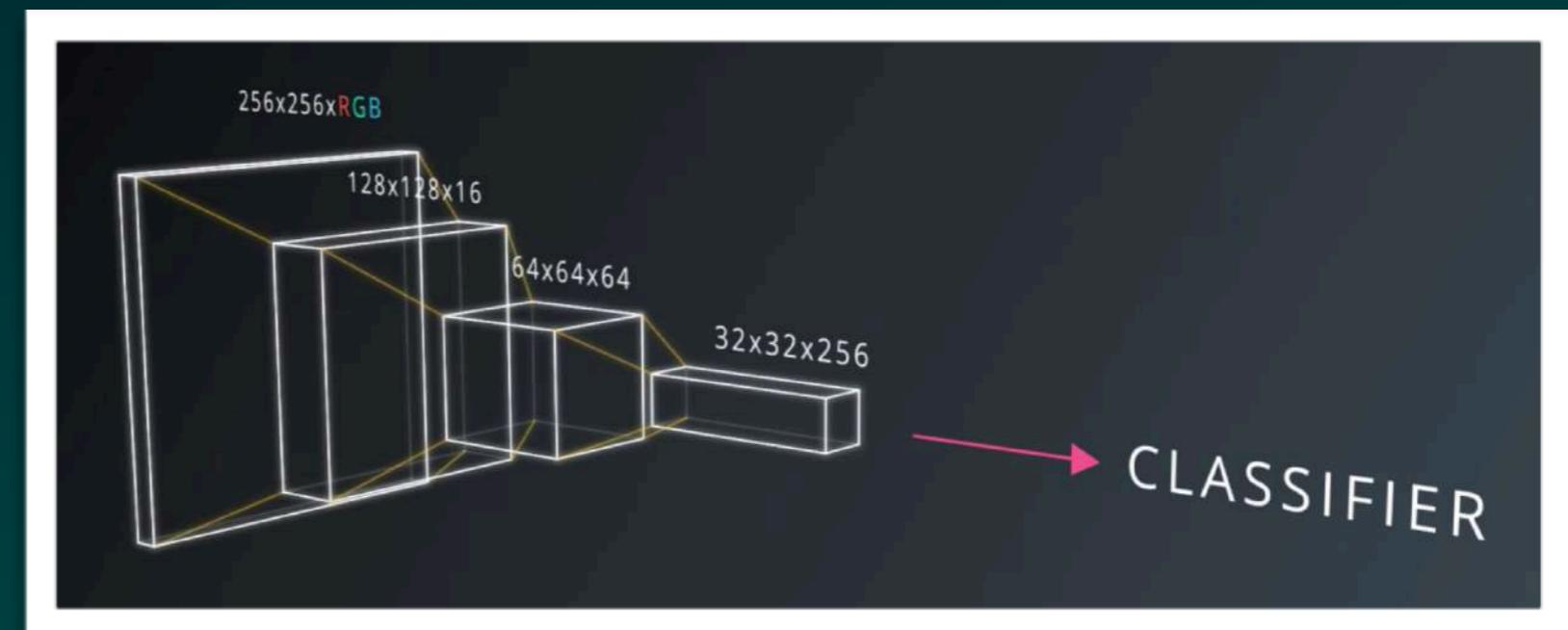
Convolutional Networks

Siraj's Image Classification

Weight Initialization

...

- Introduction to 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

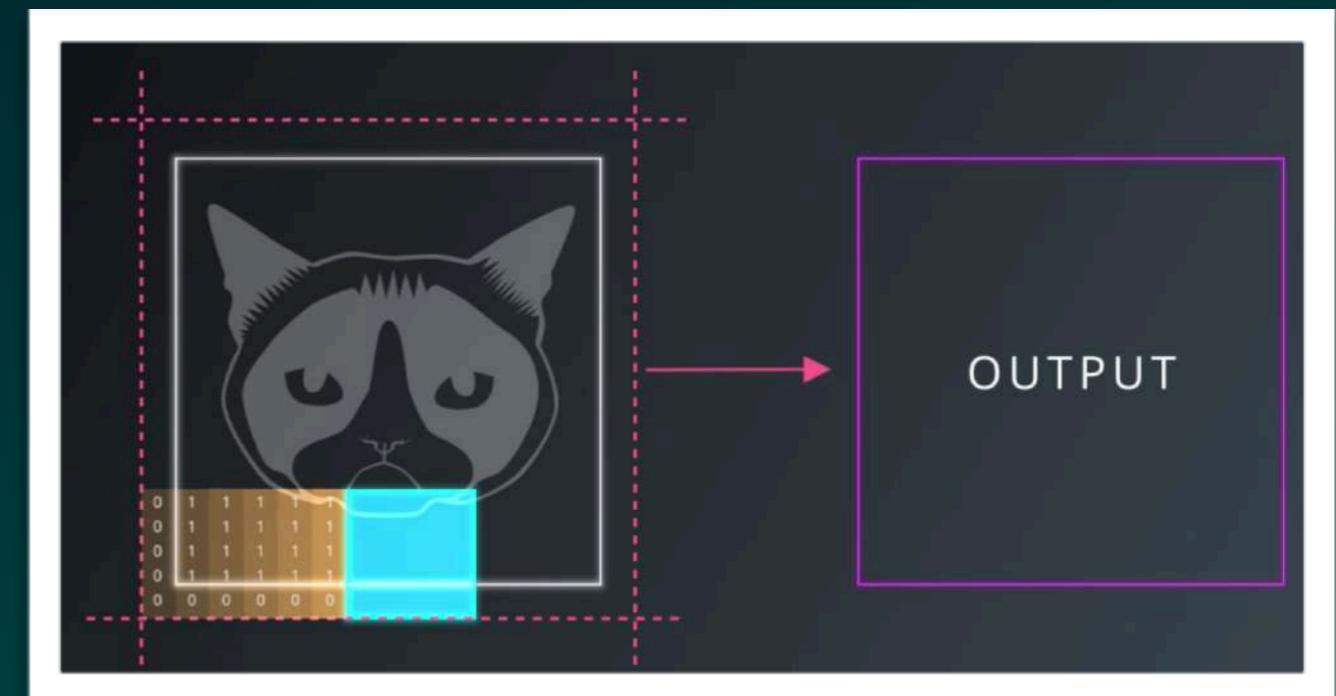
Convolutional Networks

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

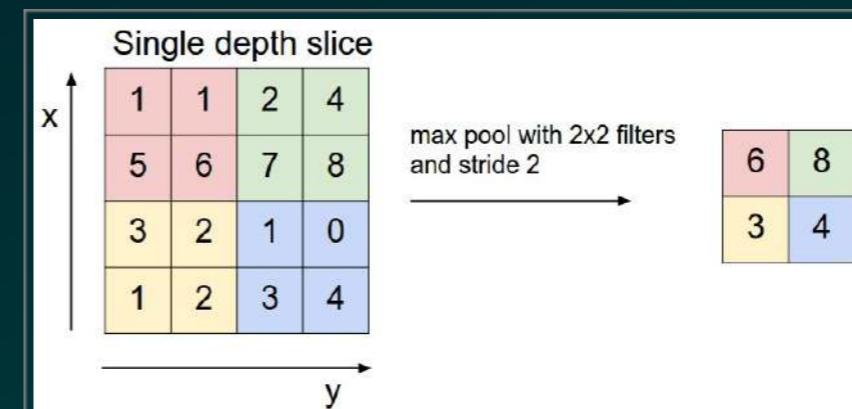
Convolutional Networks

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.

```
In [ ]: #Step 2 - Build Model
model = Sequential()
model.add(Convolution2D(32,3,3 input_shape=(img_width, img_height,3)))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2,2)))

model.add(Convolution2D(32,3,3))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2,2)))

model.add(Convolution2D(64,3,3))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2,2)))|
```

Input: every single pixel of the image

Are there patches of lines?
Are there ovals?
Are there triangles?
Is there fur?
Are there 2 eyes?
Are there 2 ears?
Is there a nose?

Is it a dog? yes/no

5:48 / 8:44

Subscribe

YouTube

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

DIRO, Université de Montréal, Montréal, Québec, Canada

Yoshua Bengio

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

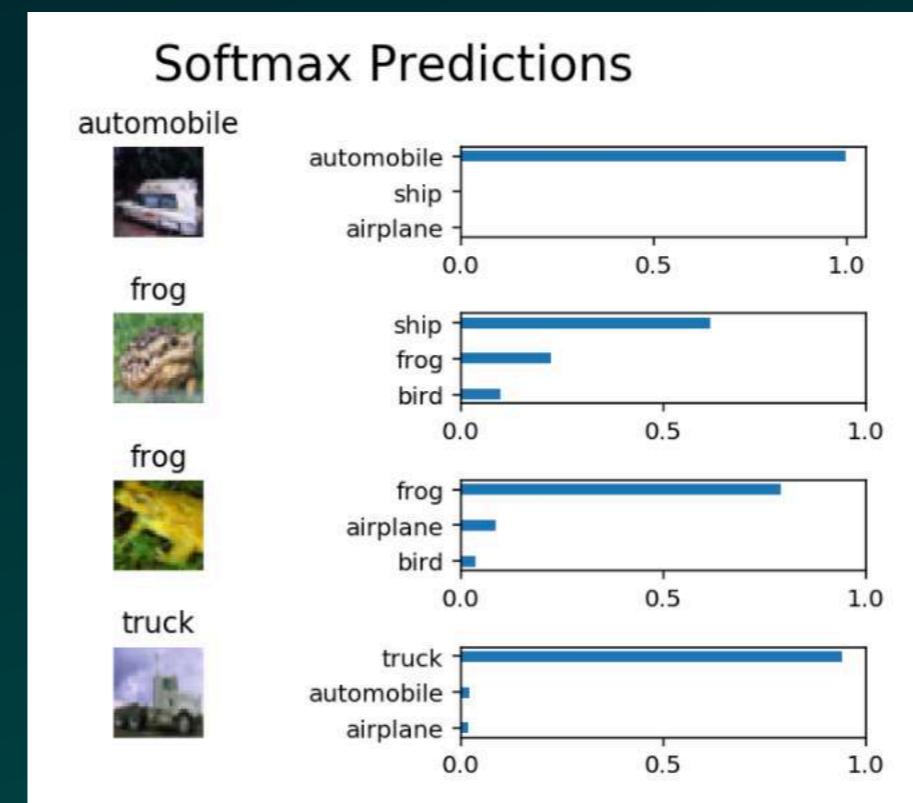
Deep Neural Networks

Convolutional Networks

Siraj's Image Classification

Weight Initialization

Image Classification



Part 3- Recurrent Neural Networks

Intro to Recurrent Neural Networks

Siraj's Stock Predictiton

Hyperparameters

Embeddings and Word2Vec

Siraj's Style Transfer

Q&A with FloydHub Founders

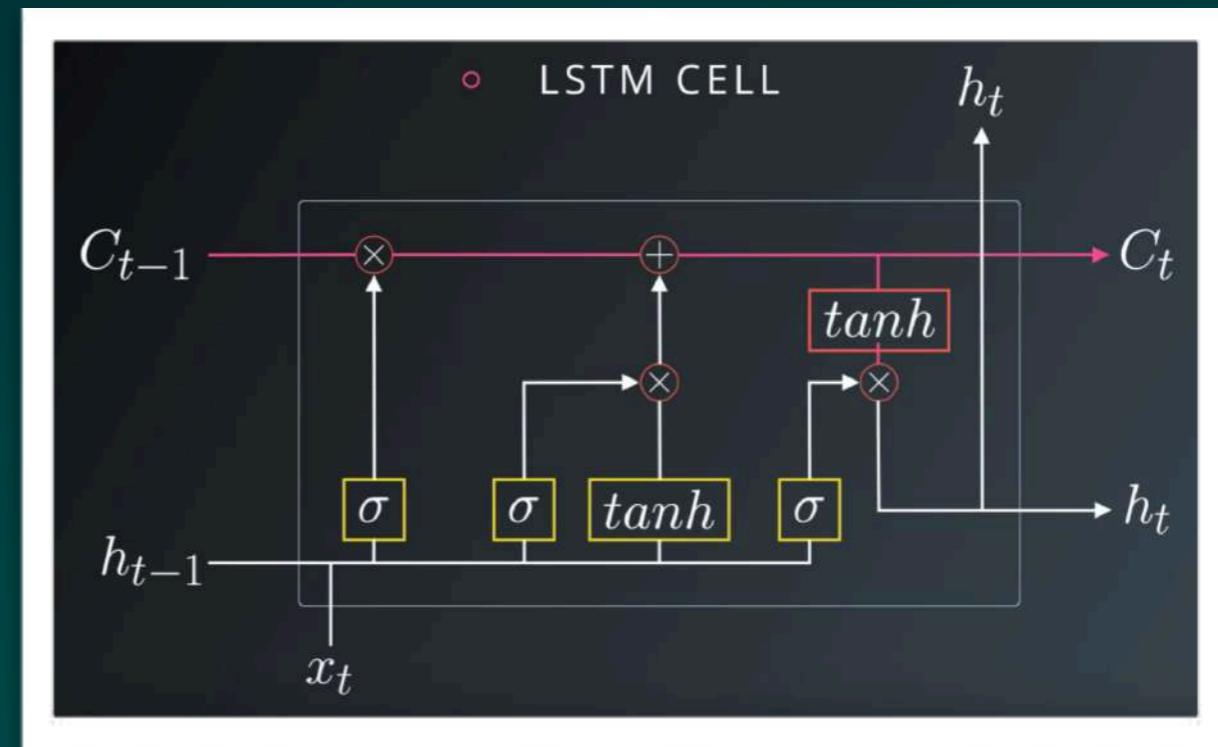
Tensorboard

Siraj's Music Generation

Siraj's Text Summarization

...

- 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_1512.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
streadds 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

Intro to Recurrent Neural Networks

Siraj's Stock Predictiton

Hyperparameters

Embeddings and Word2Vec

Siraj's Style Transfer

Q&A with FloydHub Founders

Tensorboard

Siraj's Music Generation

Siraj's Text Summarization

...

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

Intro to Recurrent Neural Networks

Siraj's Stock Predictiton

Hyperparameters

Embeddings and Word2Vec

Siraj's Style Transfer

Q&A with FloydHub Founders

Tensorboard

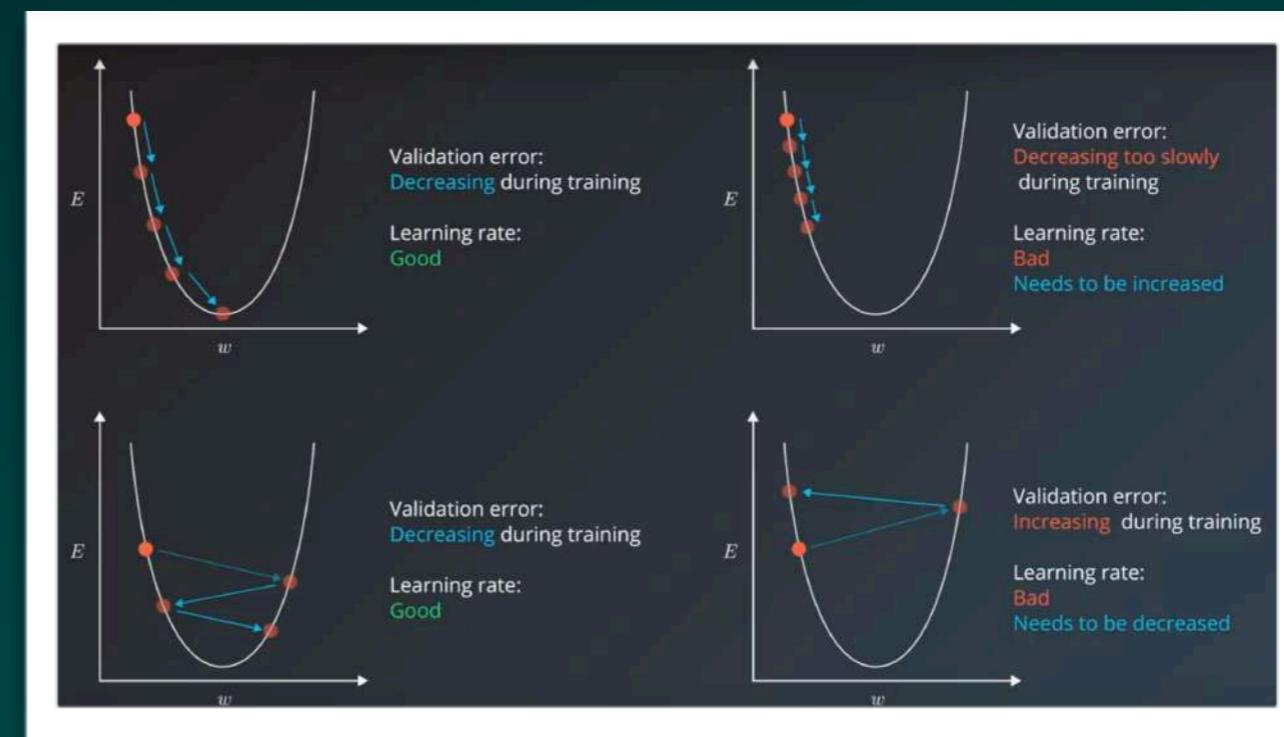
Siraj's Music Generation

Siraj's Text Summarization

...

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 Predictiton

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 data. However, despite their success, there is still a lack of understanding of how they work and how to best train them. In this paper, we present a visual approach to understanding RNNs and their training. We show how to visualize the hidden states of an RNN over time, and how these states can be used to interpret the model's predictions. We also show how to visualize the gradients during backpropagation through time, and how these gradients can be used to analyze the model's sensitivity to different inputs. Finally, we show how to visualize the hidden states of an LSTM, and how these states can be used to understand the model's memory and forgetfulness. Our visualizations provide new insights into the behavior of RNNs and LSTM, and help to explain why they are effective in solving sequential problems.

Part 3- Recurrent Neural Networks

Intro to Recurrent Neural Networks

Siraj's Stock Predictiton

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 Predictiton

Hyperparameters

Embeddings and Word2Vec

Siraj's Style Transfer

Q&A with FloydHub Founders

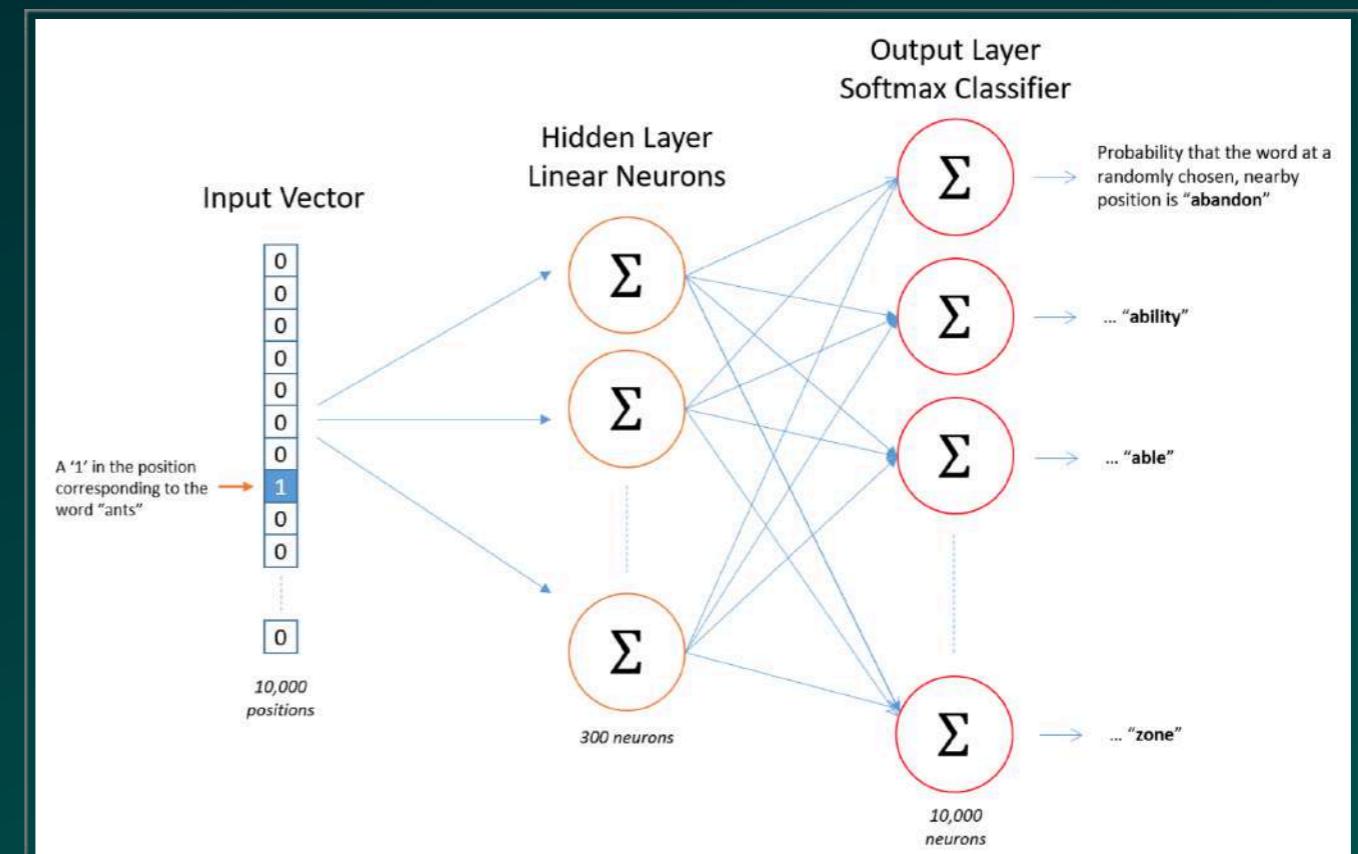
Tensorboard

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 Prediction

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 Predictiton

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 Predictiton

Hyperparameters

Embeddings and Word2Vec

Siraj's Style Transfer

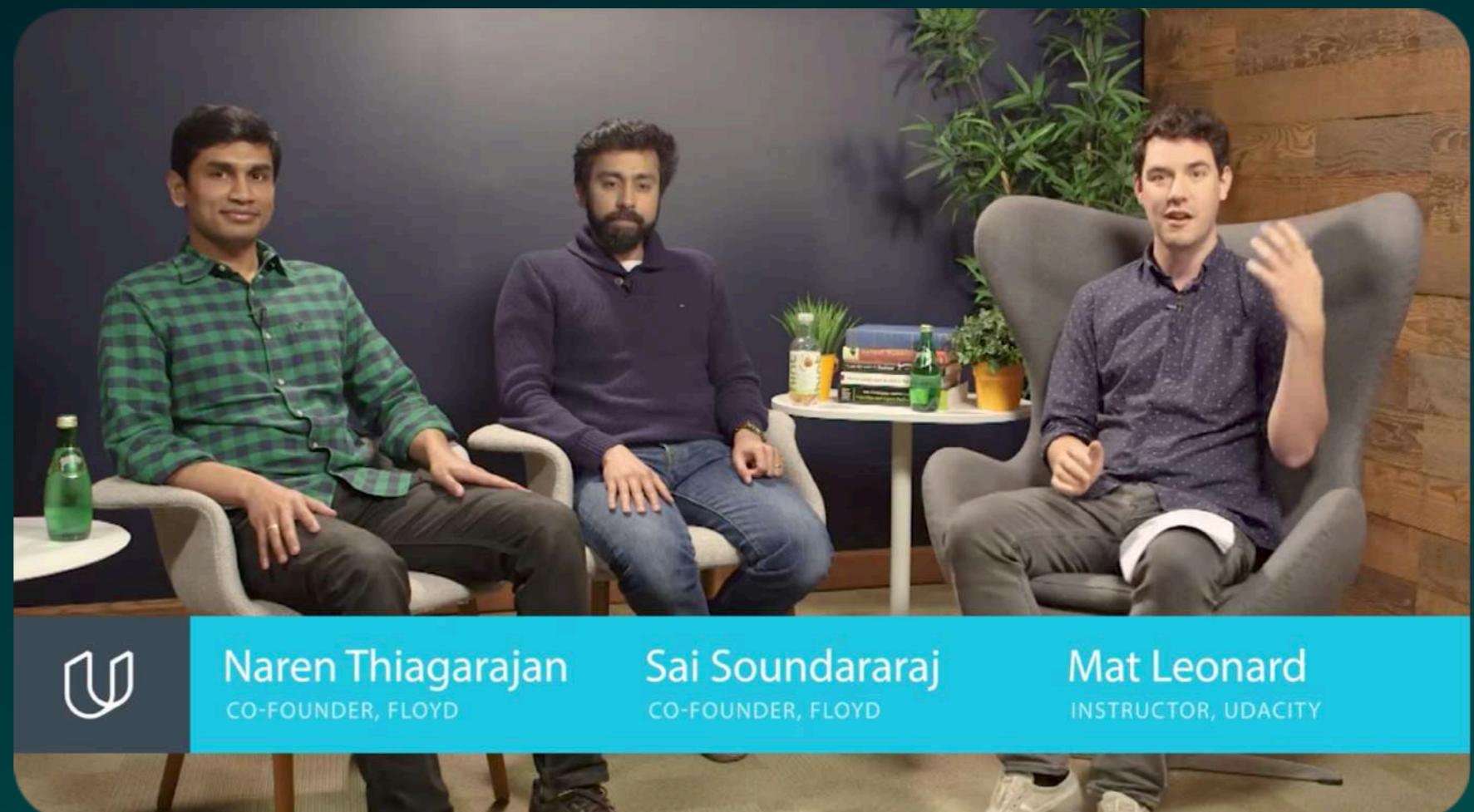
Q&A with FloydHub Founders

Tensorboard

Siraj's Music Generation

Siraj's Text Summarization

...



Part 3- Recurrent Neural Networks

Intro to Recurrent Neural Networks

Siraj's Stock Predictiton

Hyperparameters

Embeddings and Word2Vec

Siraj's Style Transfer

Q&A with FloydHub Founders

Tensorboard

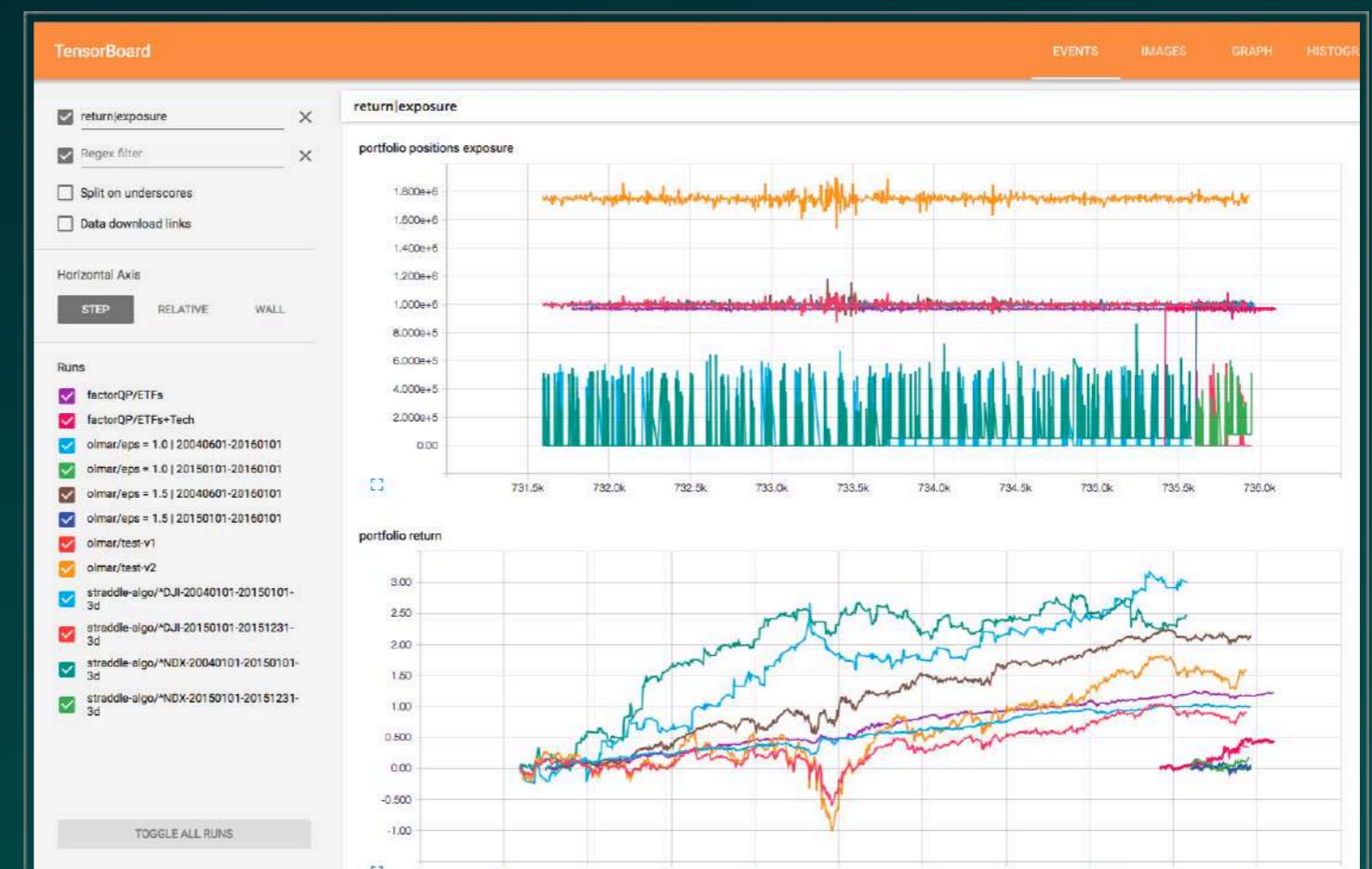
Siraj's Music Generation

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

Intro to Recurrent Neural Networks

Siraj's Stock Predictiton

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 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 Predictiton

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

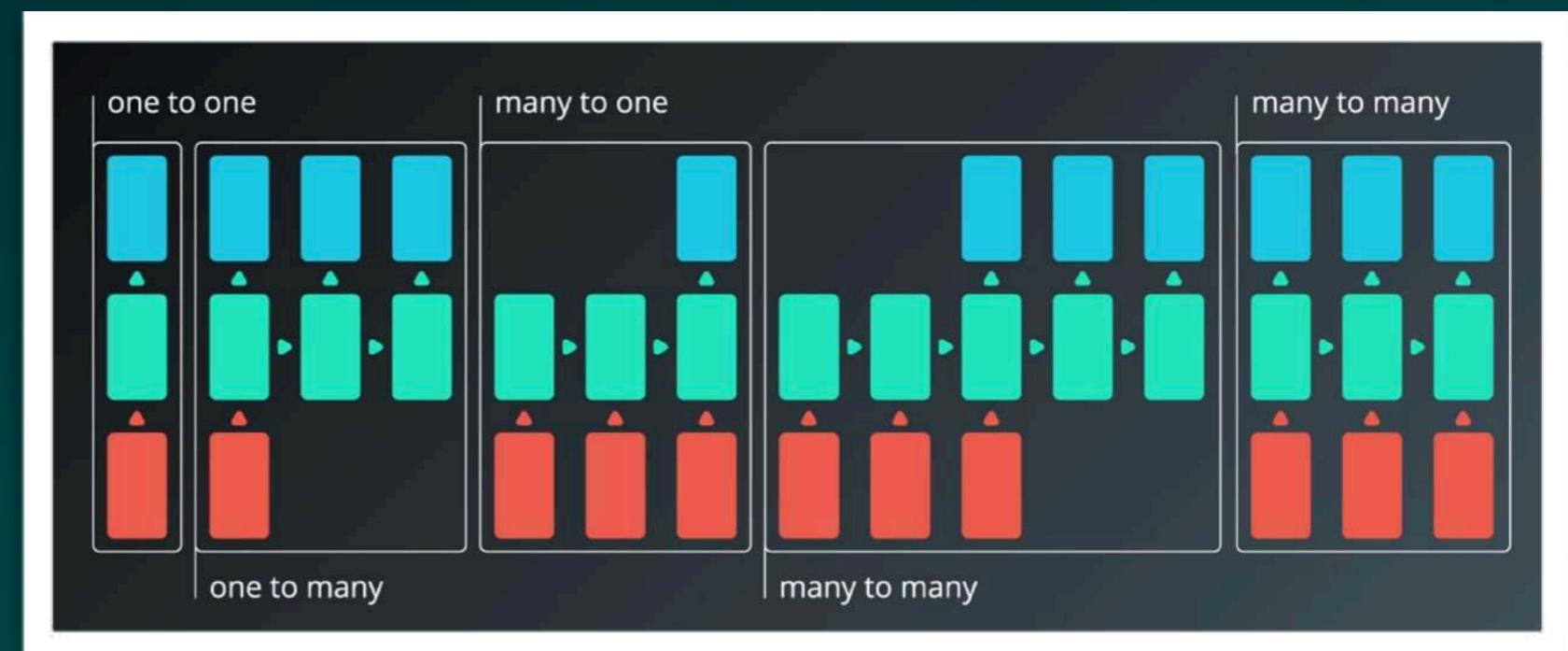
Reinforcement Learning

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

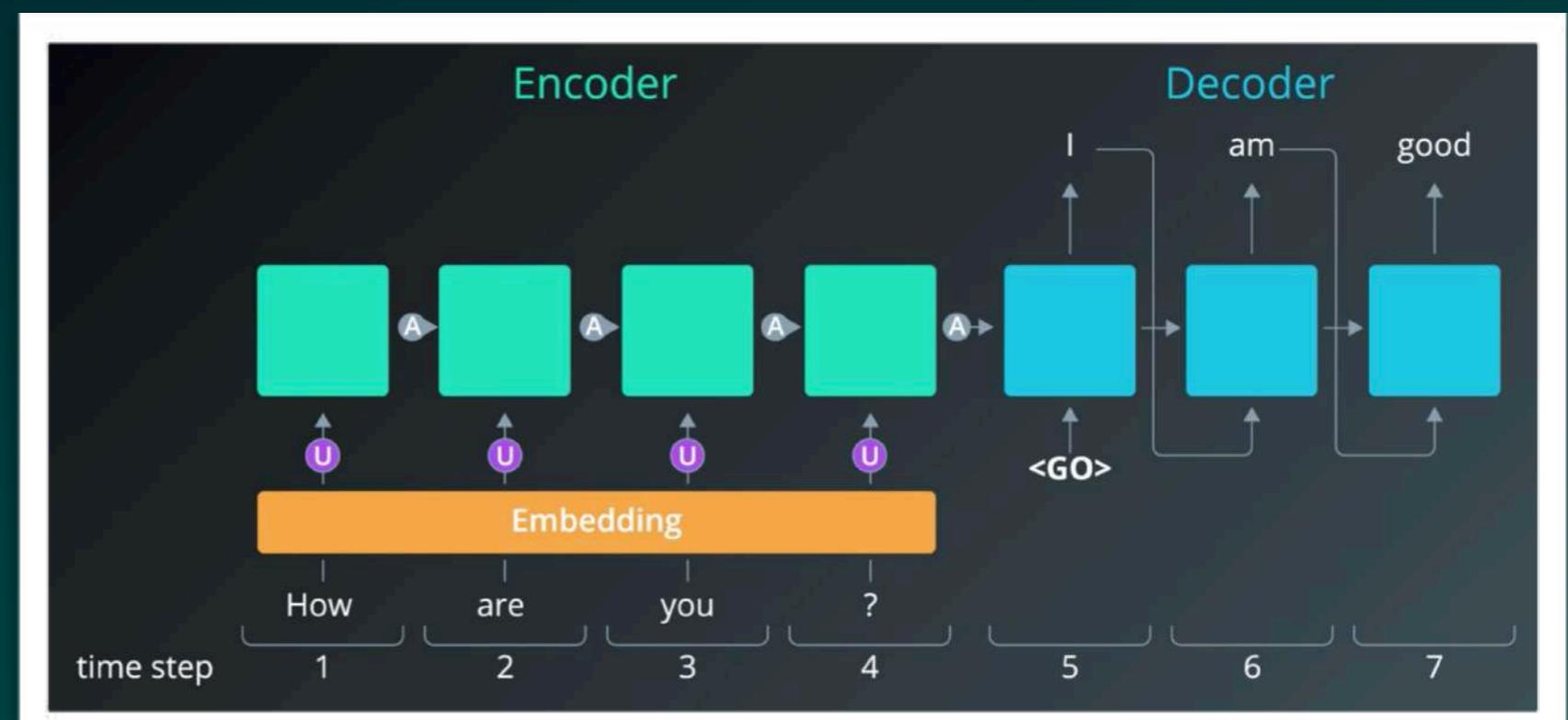
Reinforcement Learning

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

...

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 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

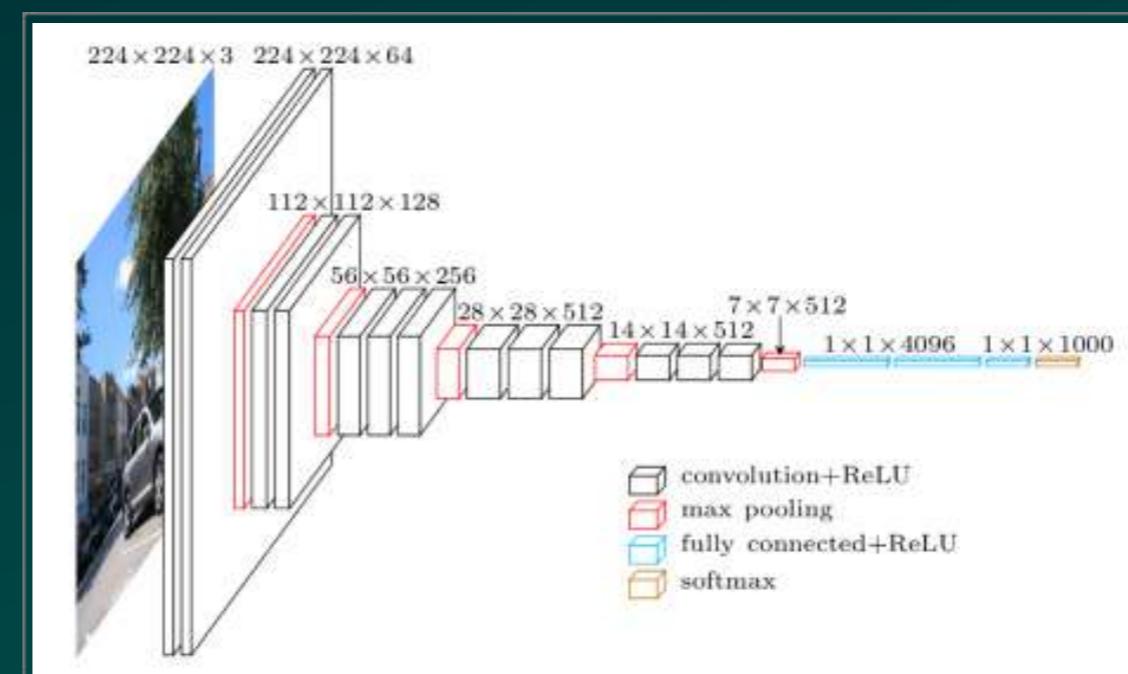
Reinforcement Learning

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

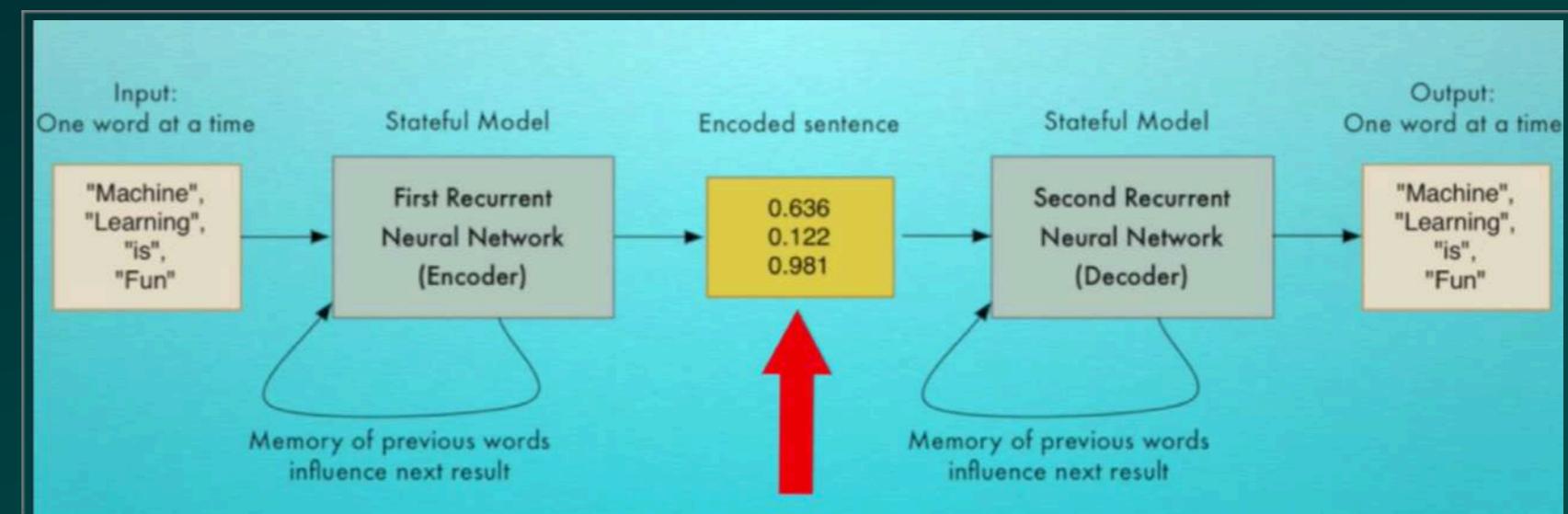
Reinforcement Learning

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

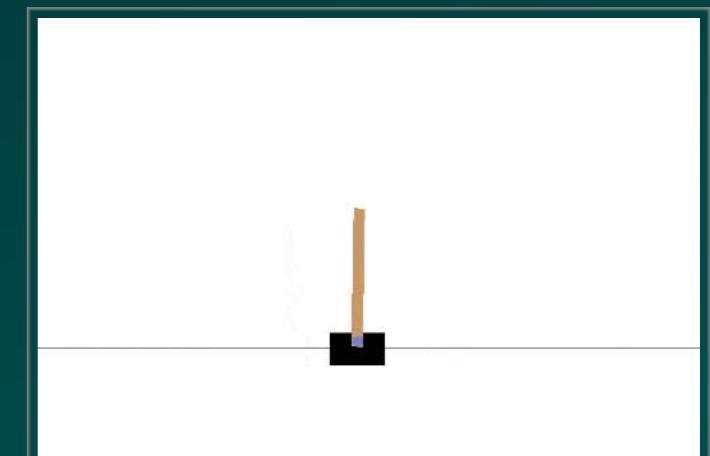
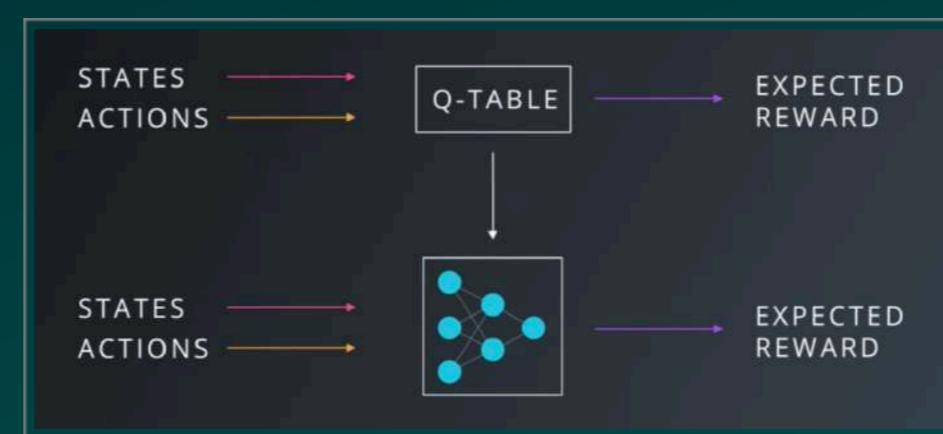
Translation Project

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

...

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 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

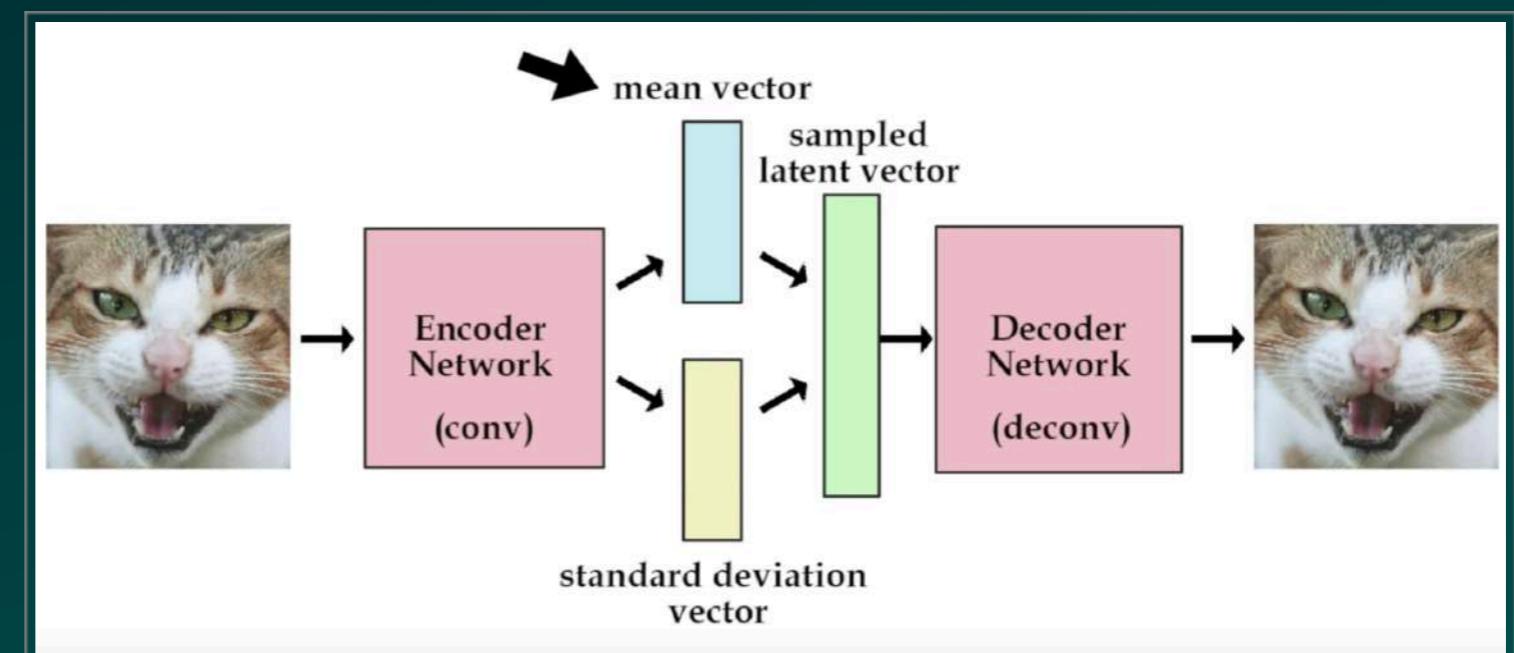
DCGANs

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

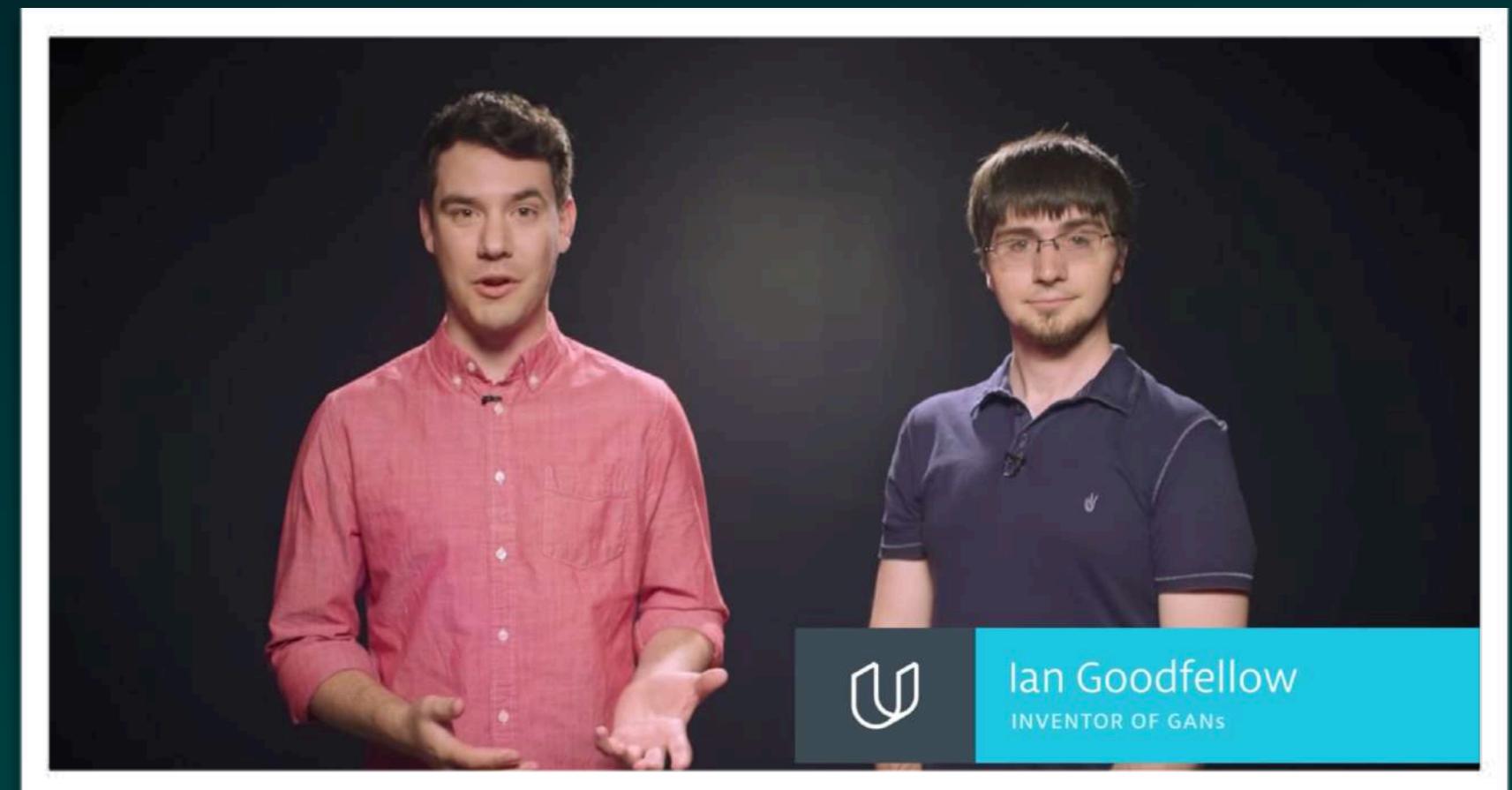
Siraj's Video Generation

DCGANs

Siraj's One Shoot Learning

Generate Faces

Lesson from creator of GANs - Ian Goodfellow!



Part 4 - Generative Adversarial Networks

Siraj's Image Generation

Autoencoders

Generative Adversarial
Networks

Siraj's Video Generation

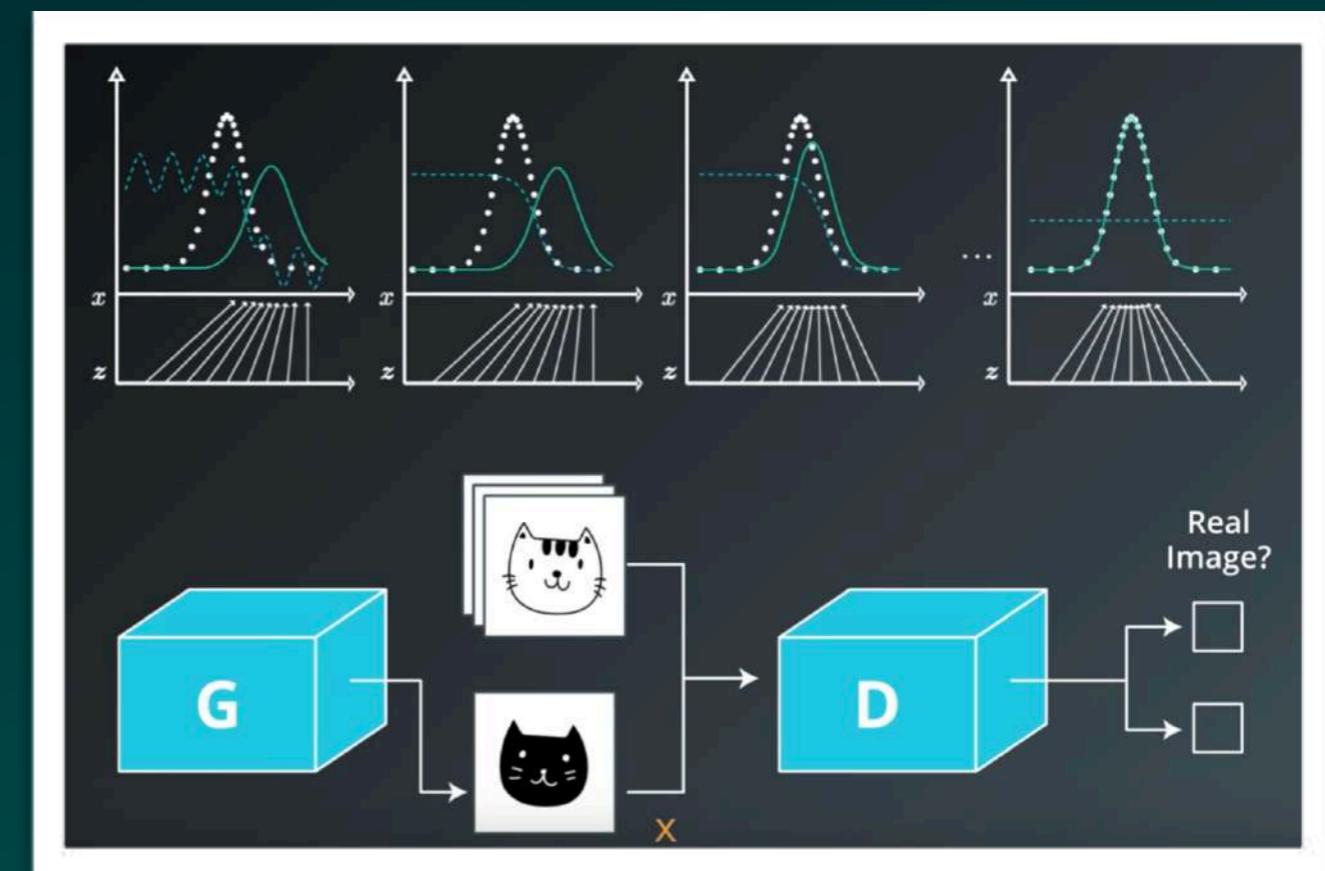
DCGANs

Siraj's One Shoot Learning

Generate Faces

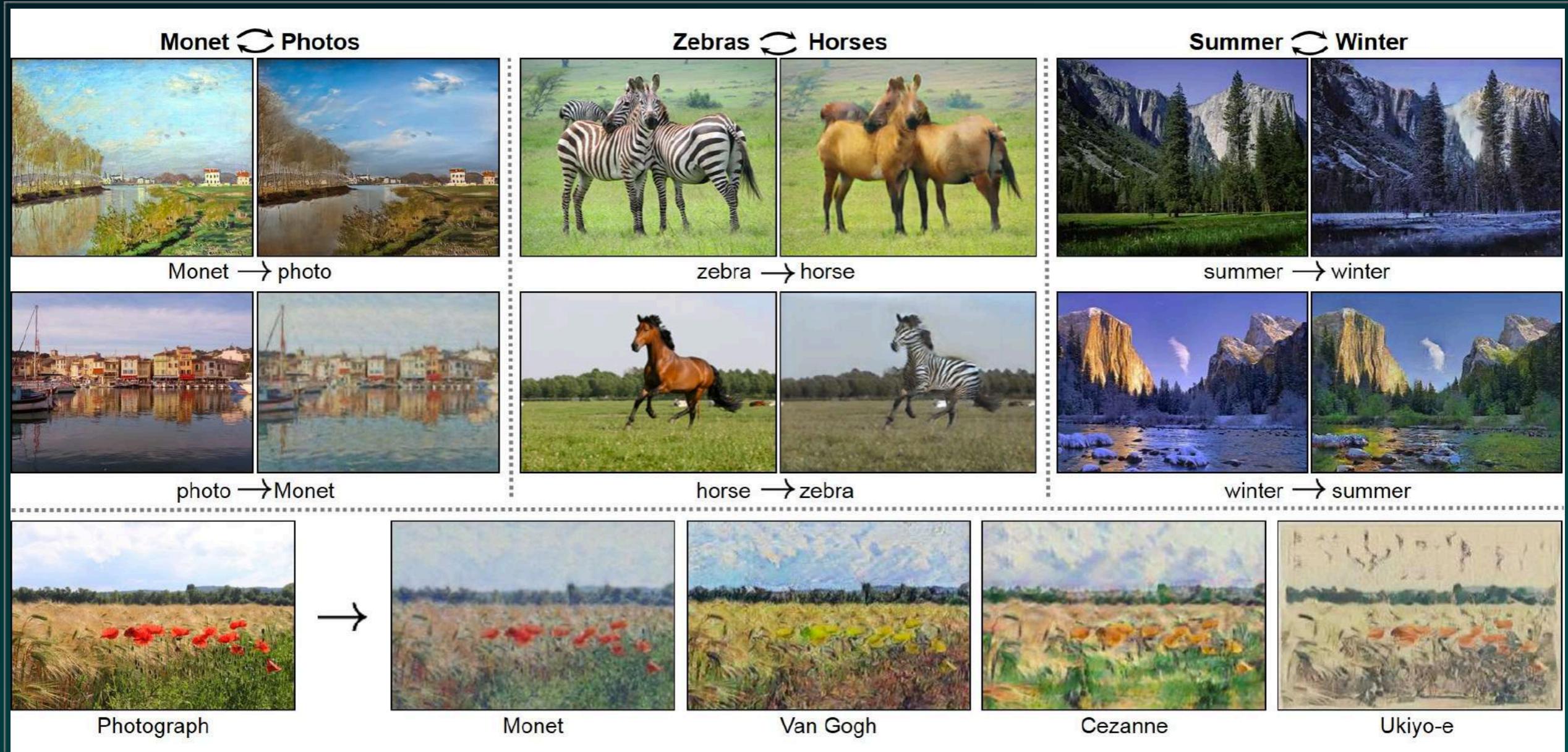
You will learn exactly how Generative Adversarial Networks work.

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



Part 4 - Generative Adversarial Networks

Cycle GAN usages



Part 4 - Generative Adversarial Networks

Siraj's Image Generation

Autoencoders

Generative Adversarial Networks

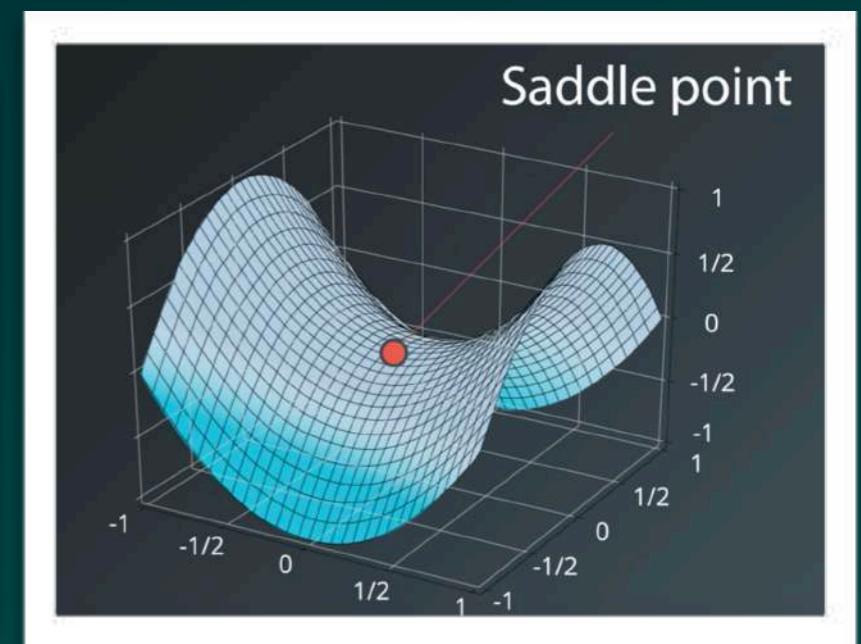
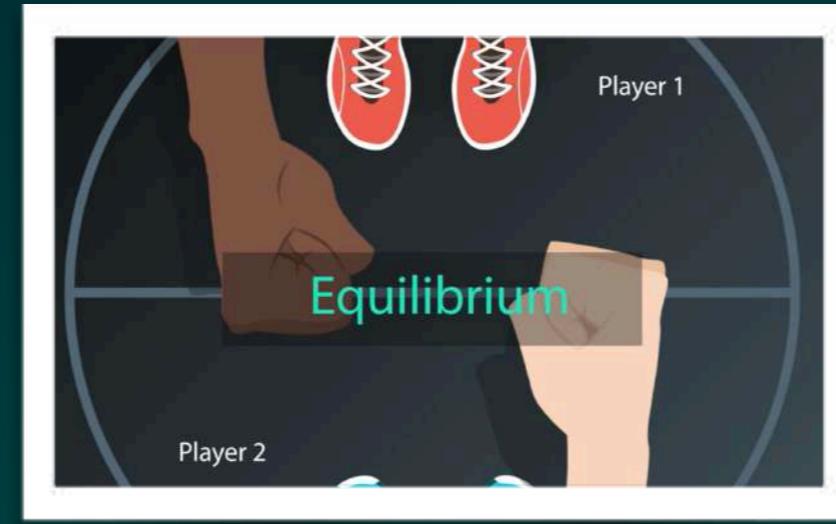
Siraj's Video Generation

DCGANs

Siraj's One Shoot Learning

Generate Faces

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

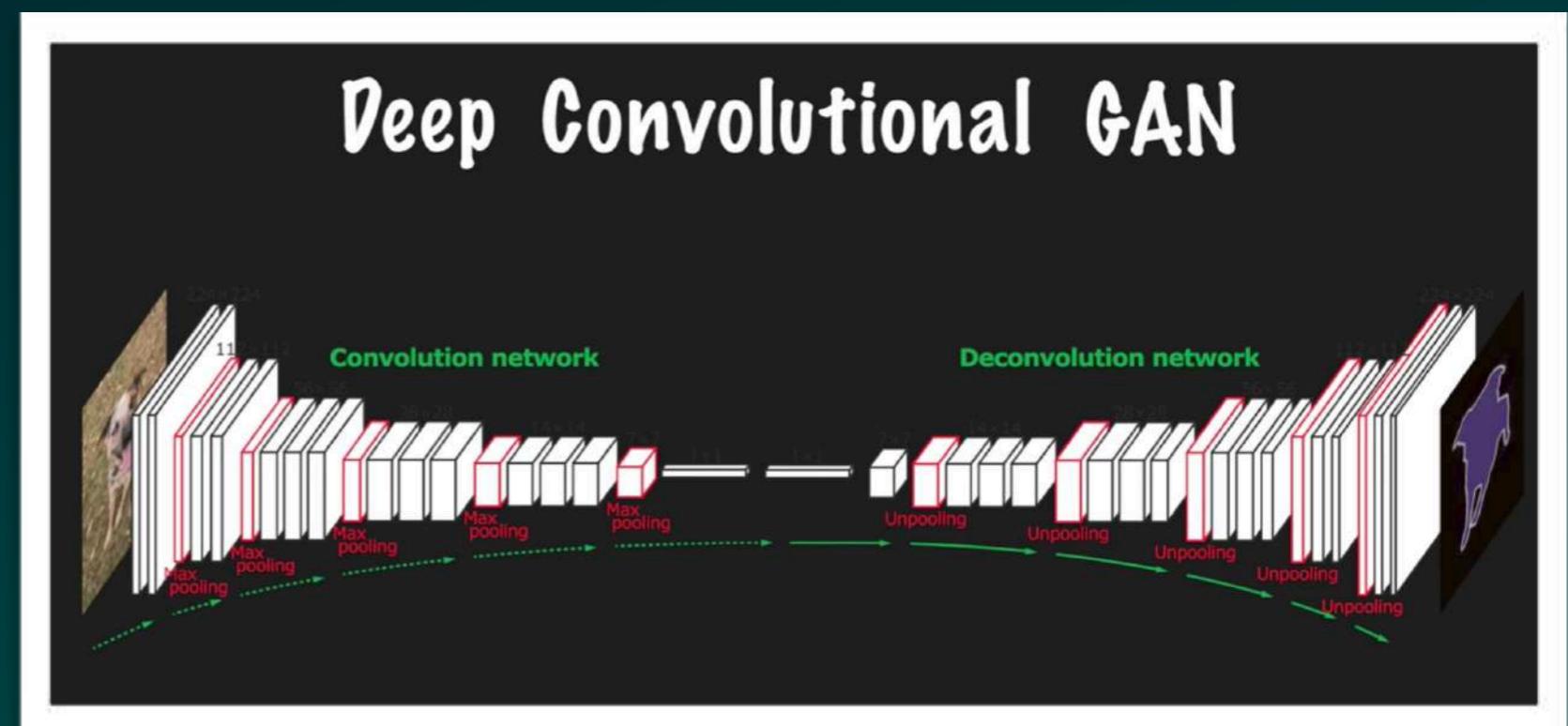
Siraj's Video Generation

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

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:



Part 4 - Generative Adversarial Networks

Siraj's Image Generation

Autoencoders

Generative Adversarial Networks

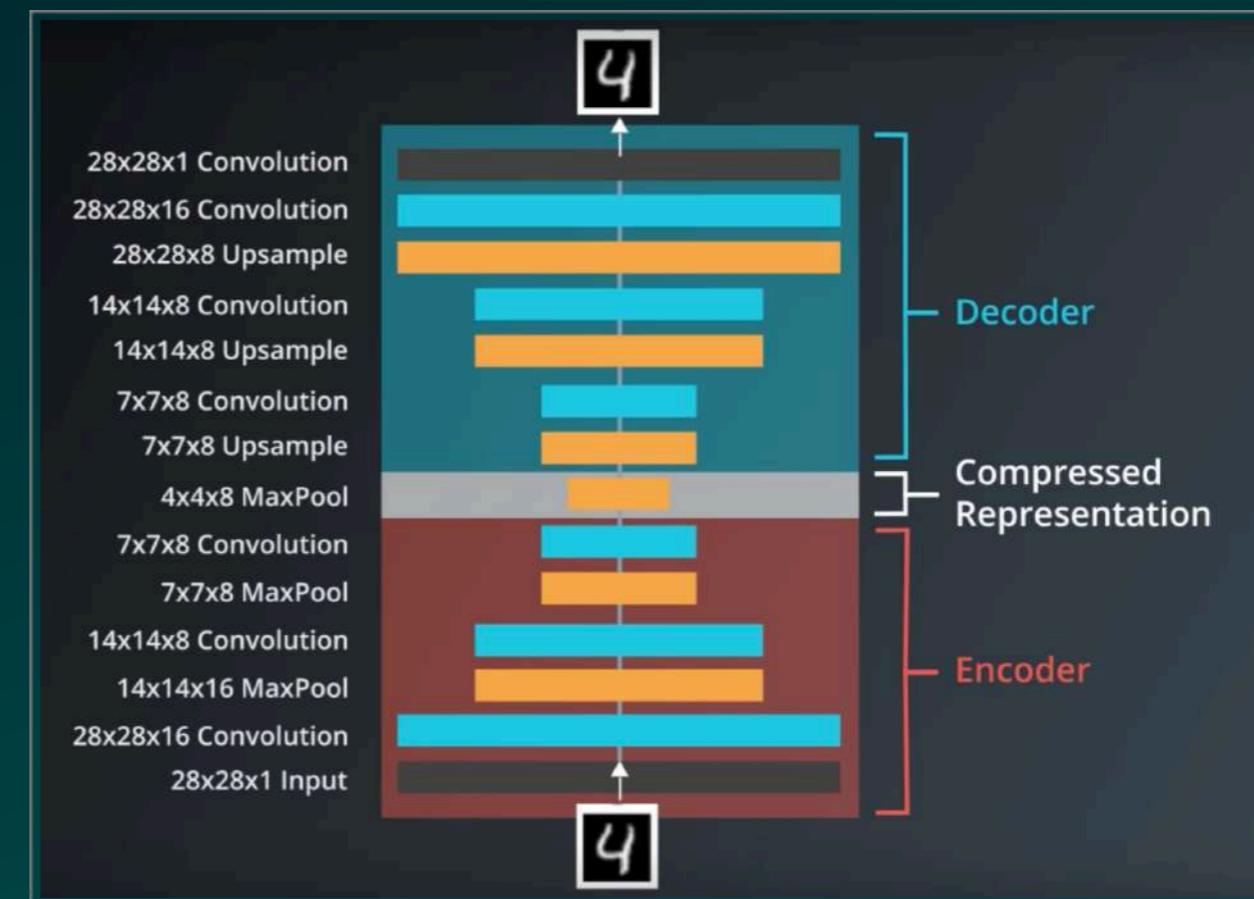
Siraj's Video Generation

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:



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

Google DeepMind

ADAMSANTORO@GOOGLE.COM

Sergey Bartunov

Google DeepMind, National Research University Higher School of Economics (HSE)

SBOS@SBOS.JN

Matthew Botvinick

Daan Wierstra

Timothy Lillicrap

Google DeepMind

BOTVINICK@GOOGLE.COM

WIERSTRA@GOOGLE.COM

COUNTZERO@GOOGLE.COM

Abstract

Despite recent breakthroughs in the applications of deep neural networks, one setting that presents a persistent challenge is that of “one-shot learn-

ing” from a single example. Such systems must be able to quickly adapt their behavior to new inputs based on a single observation. This kind of flexible adaptation is a celebrated aspect of hu-

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

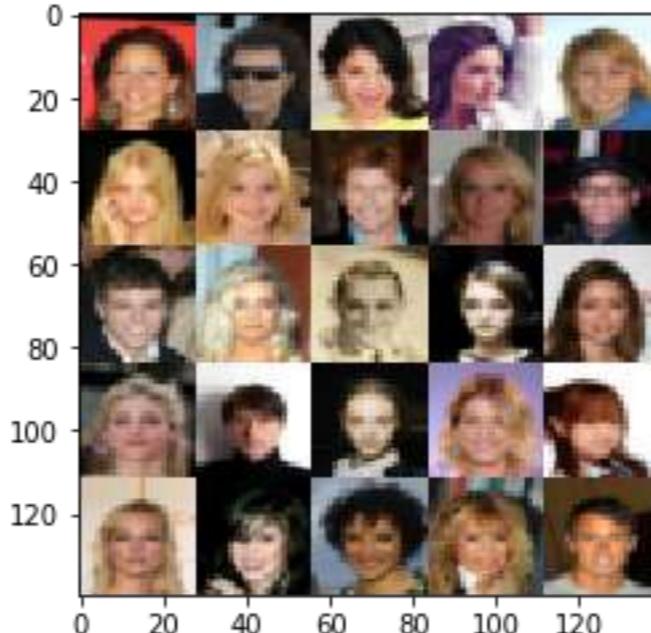
Siraj's Video Generation

DCGANs

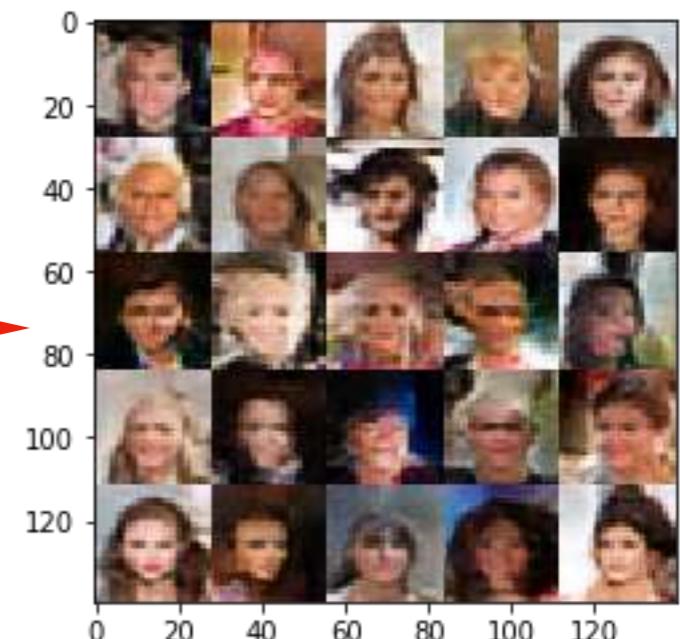
Siraj's One Shoot Learning

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



Kamil Krzyk

Has successfully completed the

Deep Learning Nanodegree Foundation

NANODEGREE FOUNDATIONS PROGRAM



A handwritten signature of Sebastian Thrun in black ink, enclosed in a blue oval.

Sebastian Thrun
Founder, Udacity

Thank you for attention!