Lecture Notes for

Machine Learning in Python

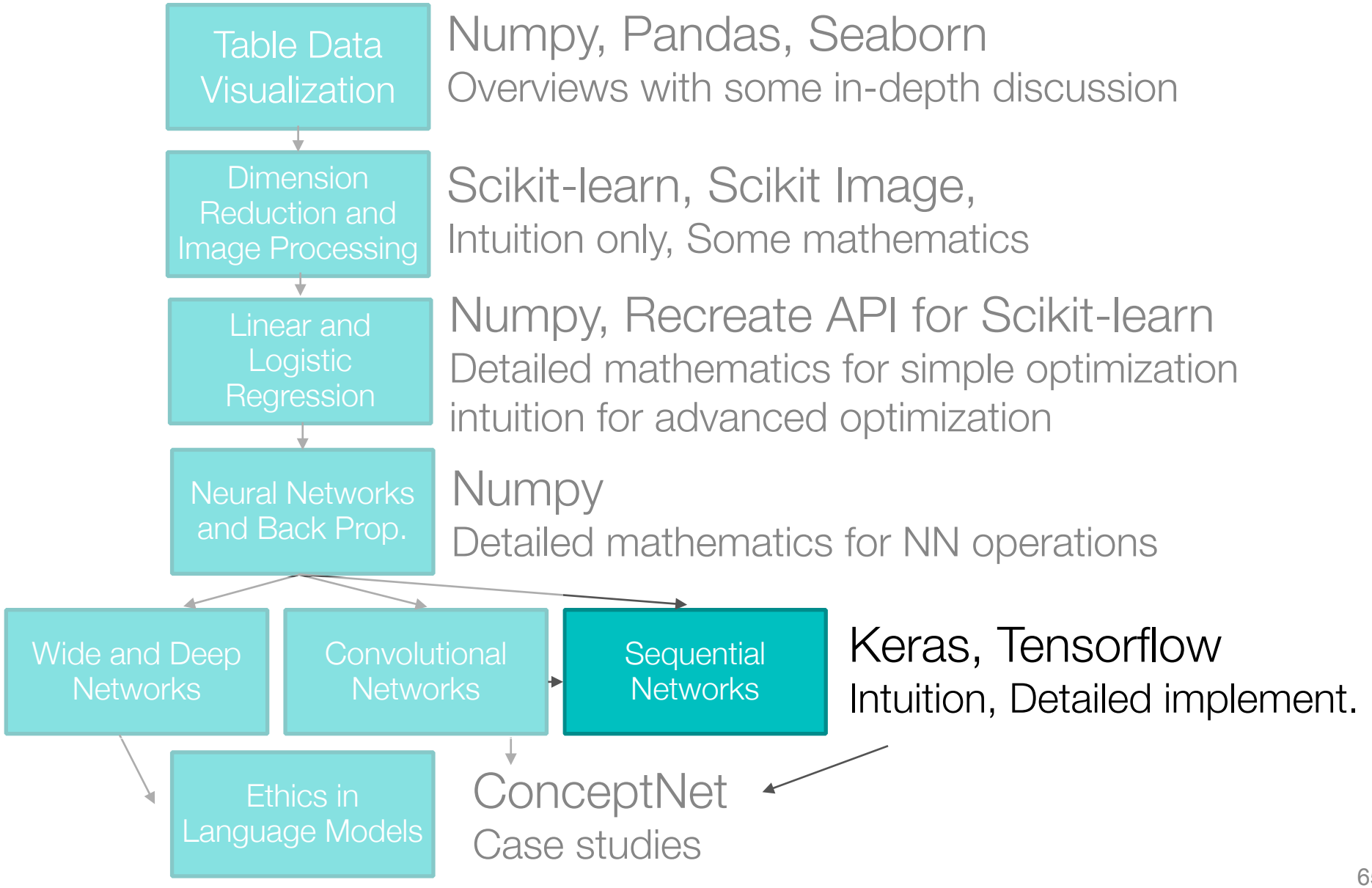
Professor Eric Larson

Sequential Networks Demonstration

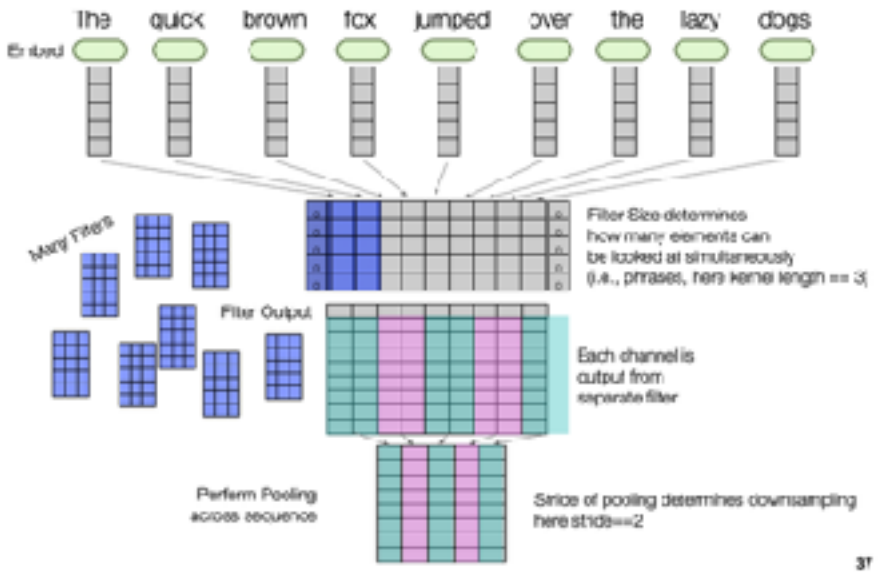
Lecture Agenda

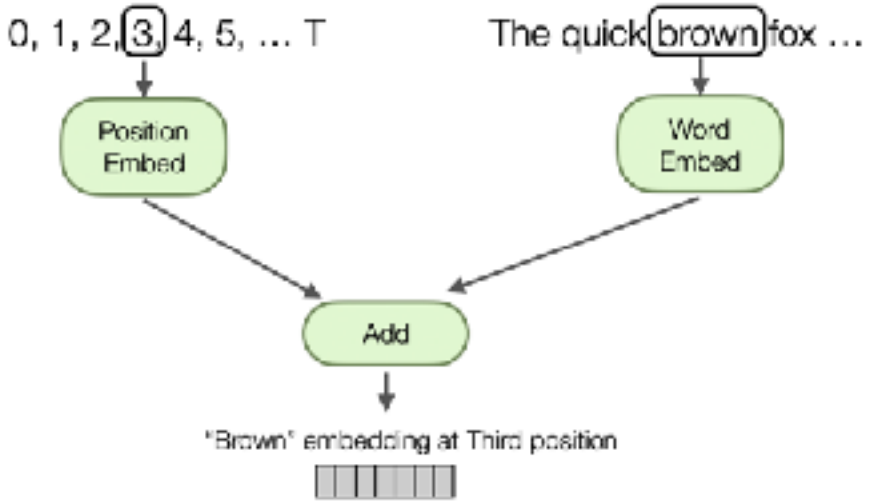
- Logistics
  - Grading Update
  - Sequential Networks due **see canvas**
- Agenda
  - Sequential Networks Demo
    - Extended Demo
  - Final Town Hall
    - (if time) Retrospective and Evaluations
  - Next time (if not behind):
    - Ethical Principles
    -

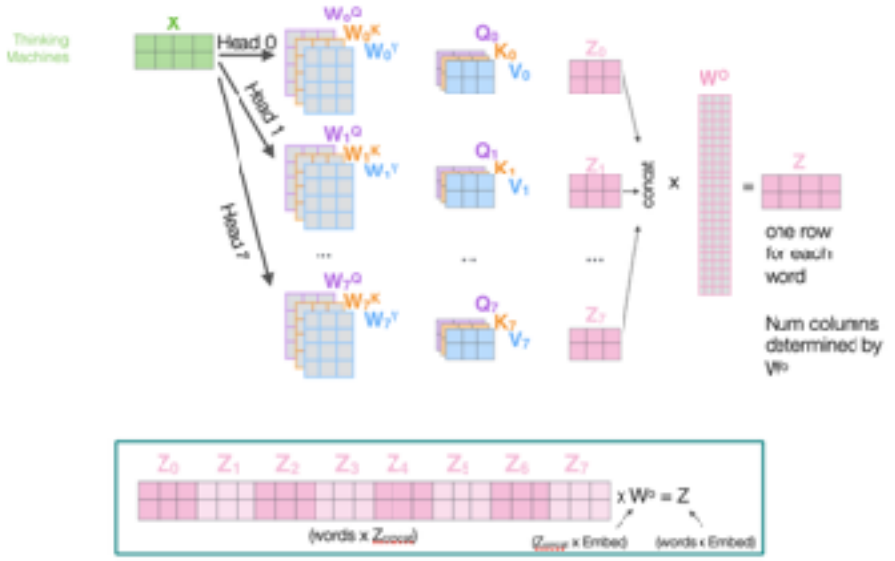
Class Overview, by topic

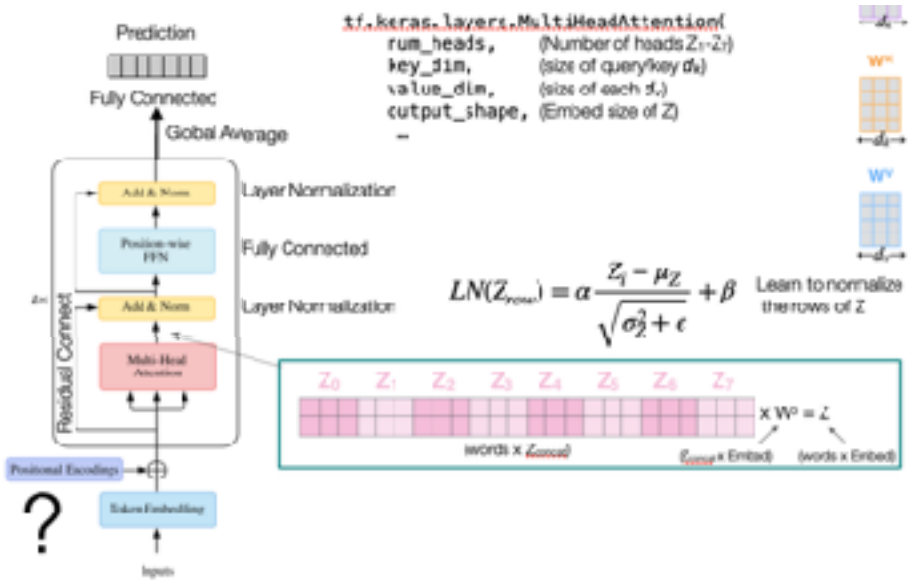


Last Time

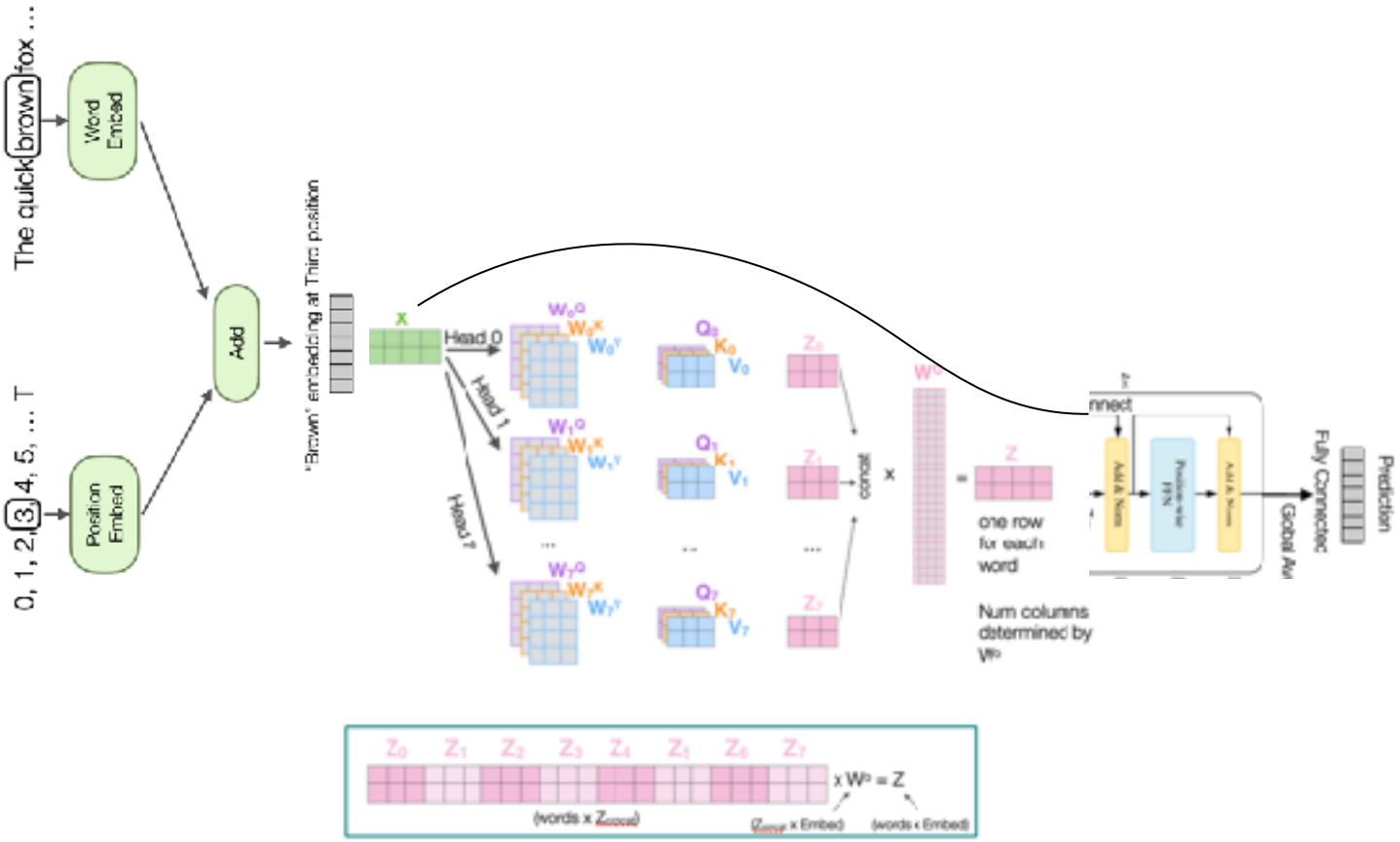

The quick brown fox jumped over the lazy dogs
Many Filters
Filter Size determines how many elements can be looked at simultaneously (i.e., phrases, here kernel length == 3)
Filter Output
Each channel is output from separate filter
Perform Pooling across sequence
Stride of pooling determines downsampling here stride==2
37


0, 1, 2, 3, 4, 5, ... T
The quick brown fox ...
Position Embed
Word Embed
Add
"Brown" embedding at Third position


Thinking Machines
X
Head 0
Head 1
Head 7
concat
one row for each word
Num columns determined by
(words x Z_concat)
x W^O = Z
(words x Embed)


Prediction
Fully Connected
Global Average
Layer Normalization
Fully Connected
Layer Normalization
Residual Connect
Positional Encodings
Inputs
tf.keras.layers.MultiHeadAttention(
num_heads, (Number of heads Z_1-Z_7)
key_dim, (size of query/key d_k)
value_dim, (size of each d_v)
output_shape, (Embed size of Z)
)
Learn to normalize the rows of Z
words x Z_concat
Z_concat x Embed
(words x Embed)
x W^O = Z

Transformers

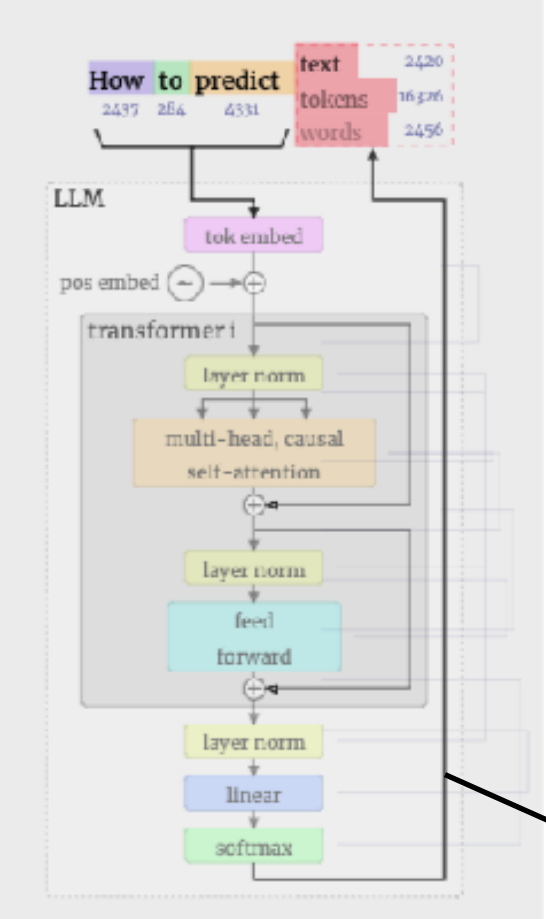


Do transformers have a lot of trainable parameters?

Do transformers have a relatively large memory footprint?

Sequential Networks in Keras

"Finish" Demo



Components

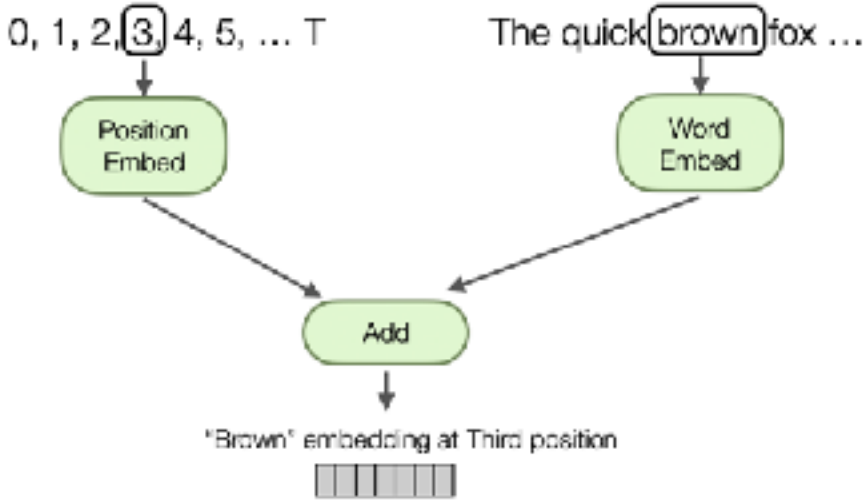
- Embedding
- Layer Norm
- Self Attention
- Projection
- MLP
- Transformer
- Softmax
- Output

Global Average

Fully Connected

Prediction

```
tf.keras.layers.MultiHeadAttention(
    num_heads,      (Number of heads Z1-Z7)
    key_dim,        (size of query/key dk)
    value_dim,      (size of each dv)
    output_shape,   (Embed size of Z, dims of Wo)
     …
```



The Transformer and 20 news groups with GloVe

`13a. Sequence Basics [Experimental].ipynb`

Sequential Networks Town Hall

CNN, RNN, LSTM, GAN,
Test time data,
Early stopping,
Data augmentation,
Dropout, Batch norm,
Gradient clipping

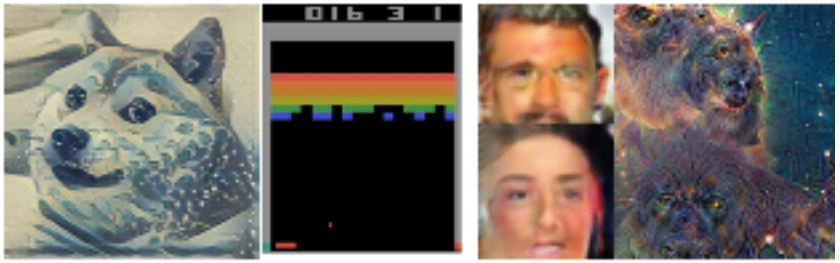
Attention

Topics review

- Data **munging** in pandas and numpy and **visualization** with matplotlib, pandas, seaborn
- Data preprocessing: **dim reduction**, images, text, categorical features, **embeddings**
- **Linear models**: linear regression, logistic regression, simple neural networks
- **Optimization** strategies: Gradient ascent, Quasi-Newton, Extensions of SGD (RMSProp, AdaM)
- **Back propagation** in MLP (from scratch)
- Tensorflow/Keras for **wide and deep networks**
- **Convolutional** neural networks (up to modern day)
- **Sequential** neural networks (scratched surface only)

Topics Not Covered

- Transfer/Multi-Task Learning
- Visualizing Deep Convolutional Networks
- Fully Convolutional Networks
- Style Transfer (if time)
- Generative Networks
- Large Language Models



Syllabus for CSE8321: Machine Learning and Neural Networks

Course Schedule

Week	Lecture A	Lecture B	Labs Due
1	Lecture: Course Introduction and Syllabus	Lecture: Brief Introduction to Neural Networks	
2	Student Presentations: Reading: Deep Learning, Chollet 2017 Lecture: Keras, TBD Lecture: Visualization Overview	Lecture: Convolutional Neural Networks Reading: Chapter 5, Section 4	
3	Lecture Demo: CNN Visuals	Lecture: Image Style Transfer Overview Reading: Chapter 6, Sections 2 and 3	Lab One: CNN Visuals
4	Student Presentations: Reading: A Neural Algorithm for Artistic Style, Gatys 2015 Lecture: Examples and Context, Jobs	Student Presentations: Reading: Perceptual Losses for Real-Time Style Transfer and Super-Resolution, 2016 Lecture: Performance Style Transfer Methods and Gyrus Consistency	
5	Lecture Demo: Image Style Transfer in Keras	Lecture: Transfer Learning in GANs Reading: Chapter 5, Section 2 and 3	Lab Two: Style Transfer
6	Lecture Demo: Transfer Learn in Mindspore	Lecture: Multi-task and Meta Learning Overview	
7	Student Presentations: Reading: Deep Multi-task Learning for Recommendations and Style, 2017 Lecture Demo: Demonstrate Multi-Task Learning	Student Presentations: Reading: An Overview of Multi-Task Learning for Sebastian Ruder, June 2017 Lecture Demo: Multi-task evaluation	
8	Lecture: Generative Adversarial Networks Overview Reading: Chapter 8, Section 1, 2, and 3	Student Presentations: Reading: Unsupervised Representation Learning with Deep Convolutional GANs, by Radford 2015 Lecture: Problems with GANs	Lab Three: Multimodal and Multi-task
9	Student Presentations: Reading: Recent's Principled Methods for Training GANs, Arjovsky and Bottou 2017 Lecture: TGAN for Domain Representations	Lecture Demo: GAN Training on Mindspore	
10	Lecture: Zero Variance Learning Overview	Lecture: Q-Learning	Lab Four: GANs
11	Lecture: Trust Region Policy Optimization	Lecture: Trust Region Policy Optimization cont.	
12	Student Presentations: Reading: Playing Atari with Deep Reinforcement Learning Policy Lecture Demo: OpenAI Gym: Pixel Shuttle	Lecture Demo: Training Deep Reinforcement Learning for Games	
13	Lecture: The Future of Deep Learning Reading: Chapter 9	Lecture: Conclusion	Lab Five: Reinforcement Learning
14	Work on Final Project Paper	Work on Final Project Paper	
15	Work on Final Project Paper	Work on Final Project Paper	Presents, Paper Due

Syllabus for CSE8321: Machine Learning and Neural Networks

Overview

This course extends basic knowledge of the use of Neural Networks in machine learning beyond simple prediction, especially targeted outputs that are generation or alteration of images, text, and audio. The course emphasizes topics of neural networks in the "deep learning" subdomain. This course will survey of important topics and current areas of research, including transfer learning, multi-task and multi modal learning, image style transfer, neural network visualization, deep convolutional generative adversarial networks, and deep reinforcement learning. For grading, students are expected to complete smaller team-based projects throughout the semester, present one research paper in a 15-20 minute group presentation (covering topics in the course), and complete a comprehensive final project that involves a number of different deep learning architectures.

Thank you for a great semester!

- but it could **have been better** somehow, right?
  - how could you learn better, more reliably for an interview?
  - what should **not be cut** or **not changed**?
    - **Already cut**: SVMs, Ensembles, RNNs, many-to-many RNNs,
  - X-formers?
  - More convolutional approaches/depth?
  - More APIs? Turi / PyTorch?
  - More flipped Assignments?
  - Self-guided Jupyter notebooks?

Thank You for an Excellent Semester!

Courtesy of Omar Roa

Please fill out the course evaluations!!!!