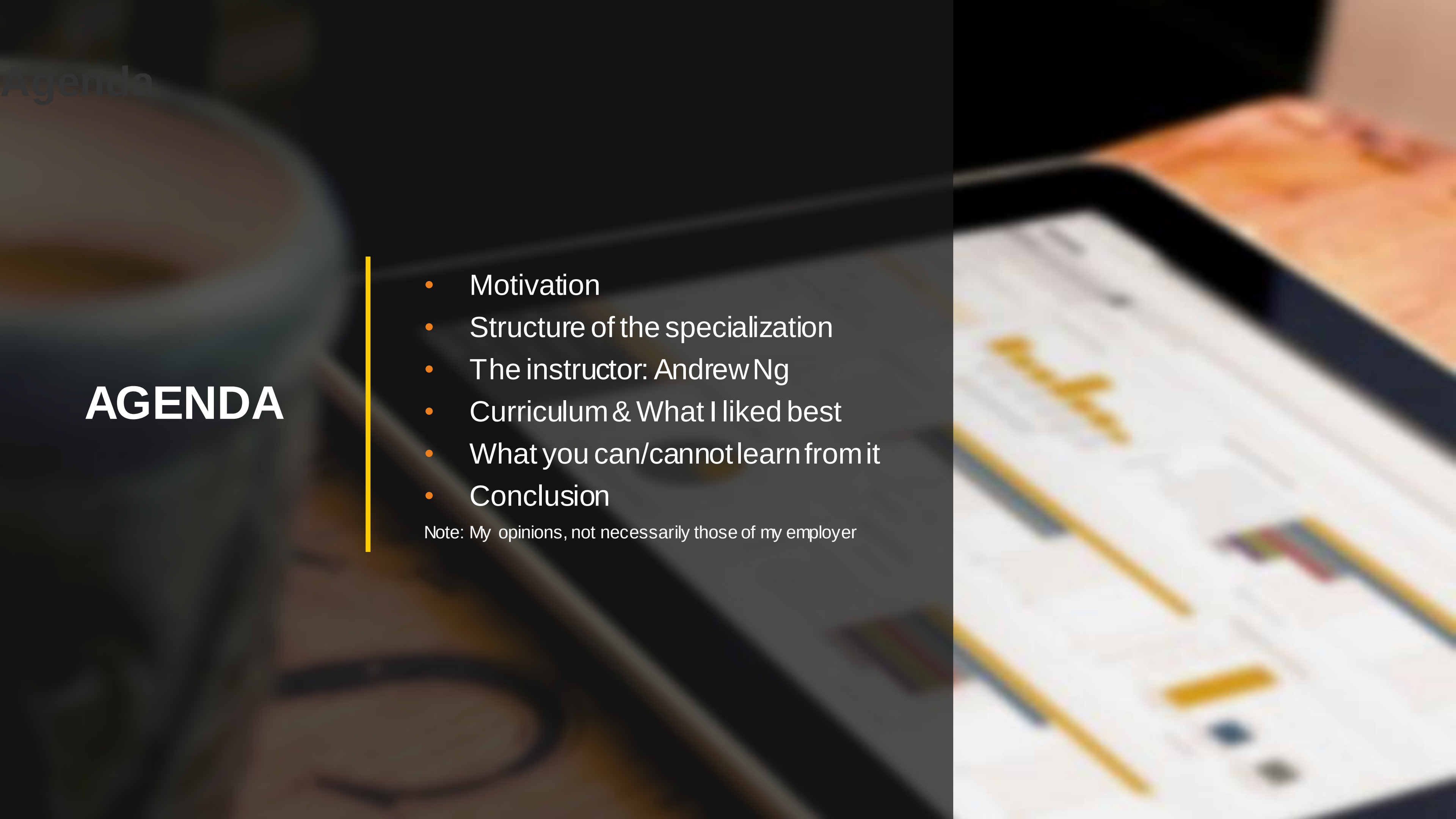
Feb. 2018

Review: Andrew Ng’s Deep Learning Specialization on Coursera

Christoph Bonitz christoph.bonitz@automic.com



Agenda

AGENDA

- Motivation
- Structure of the specialization
- The instructor: Andrew Ng
- Curriculum & What I liked best
- What you can/cannot learn from it
- Conclusion

Note: My opinions, not necessarily those of my employer

Motivation

- Deep Learning Meetup has been growing steadily
  - Large number of repeat visitors, talks that require basic knowledge of subject area
  - But also significant number of new attendees every time
- How to learn the basics?
- 1 ½ years ago I found it hard to find a single comprehensive source for getting started
- Nowadays it is easy to be overwhelmed with choices
  - Textbooks (e.g. Goodfellow/Bengio/Courville, Chollet)
  - University course videos (e.g. Stanford's cs231n)
  - Online courses – commercial and free
- Attempt to replicate: I found Oleg Leizerov's talk about the *Self Driving Car Engineer Nanodegree* helpful
- Help beginners make a choice, point experienced practitioners to good (free) parts of the course

Structure of the specialization

- Coursera: Commercial platform for online courses
- *Specialization*: Series of courses around a topic
- This one: 5 courses, 2-4 weeks each
- Video lectures with slides, with instructor writing/drawing on the slides a lot
- Programming assignments (Jupyter notebooks)
  - NumPy for implementing basic building blocks
  - Keras for higher level tasks
- Multiple choice quizzes
- Free to watch videos, pay subscription for full access (exercises, certificate): ~40 EUR/Month
- My experience: 4-10 hours per week

The instructor: Andrew Ng

Source: Andrew Ng

- Andrew Ng's career:
  - Stanford professor, teaching ML at least since 2008, leading researcher
  - Coursera cofounder, his first ML MOOC started the trend
  - Co-founded and led Google Brain
  - Head of Baidu research department (1500 employees)
- Trivia: Baidu market cap briefly fell by ~USD1.5 billion when he left
- => Course by a top researcher in ML field with broad industry experience, created right after his last industry gig

Source: Stanford University

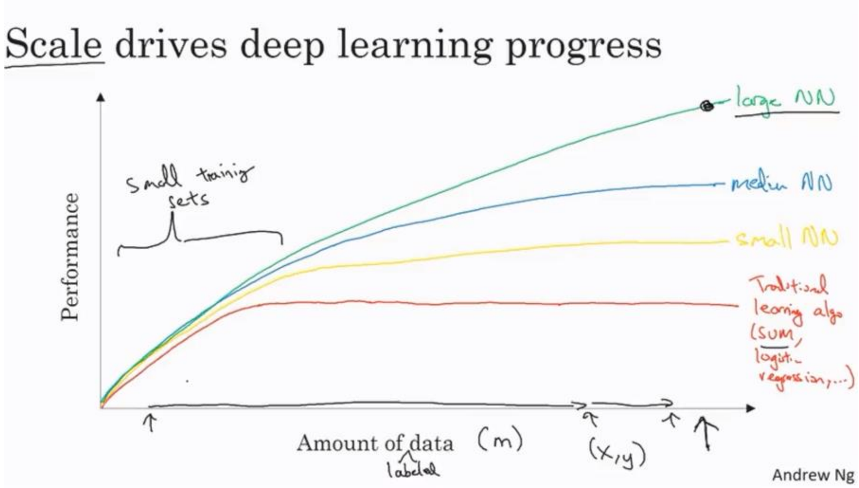
Source: Google Brain

Source: Andrew Ng

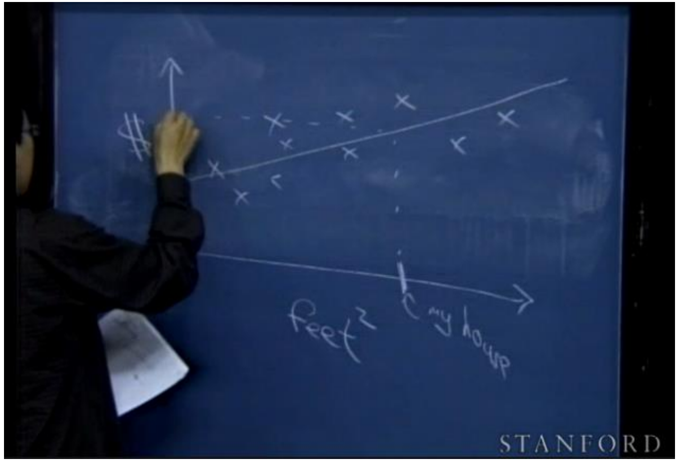
Source: landing.ai

1. Neural Networks and Deep Learning

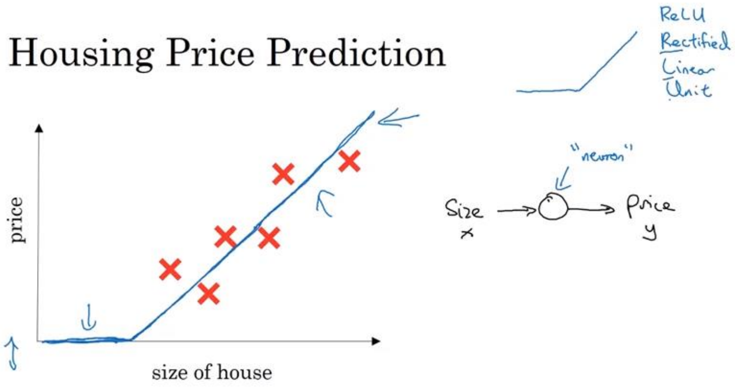
- What is Deep Learning all about?
- Content
  - Intro to ML in context of DL
  - What is DL, why is it taking off now
  - NN basics: simple FFNNs, gradient descent



What I liked best: Evolution & refinement of teaching style



2008 (Image: Stanford University)



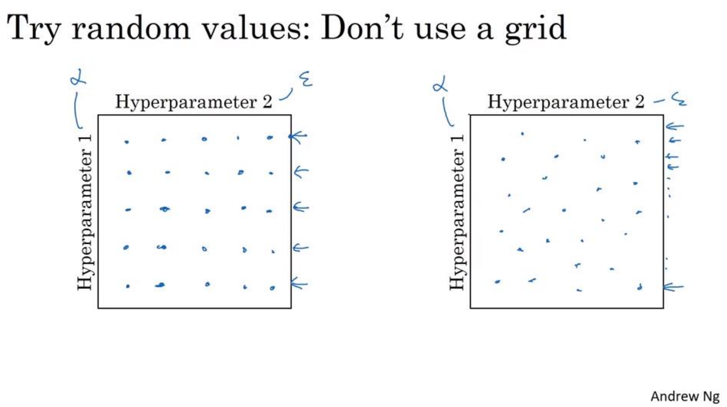
2018 (Image: Andrew Ng/Coursera)

Property of Automic Software. All rights reserved

2. Improving Deep Neural Networks: Hyperparameter tuning, Regularization and Optimization

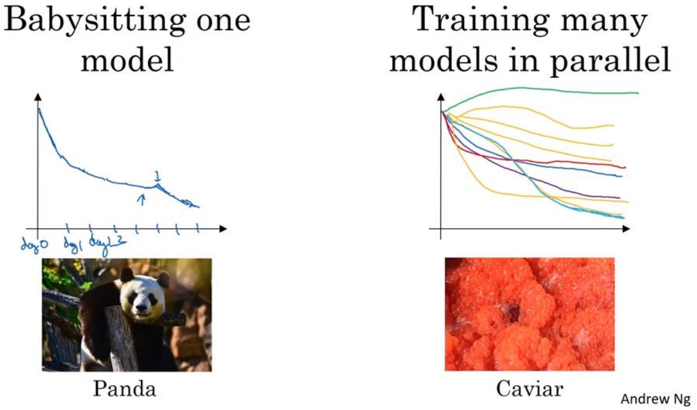
- How to make Neural Networks learn well
- Content
  - Bias/variance,
  - Optimization algorithms
  - Normalization
  - Regularization
  - Hyperparameter tuning process
  - Using frameworks

What I liked best: Two insights about hyperparameter tuning



Try random values: Don’t use a grid

Andrew Ng



Babysitting one model

Training many models in parallel

Panda

Caviar

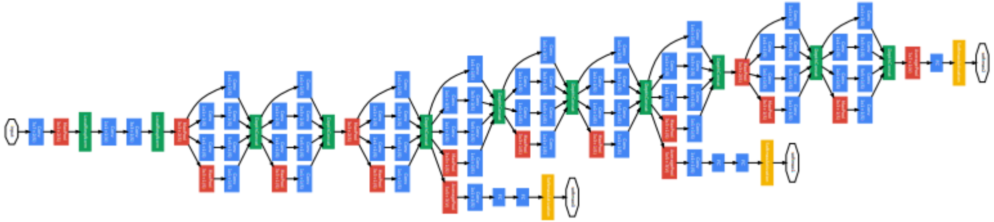
Andrew Ng

3. Structuring Machine Learning Projects

- How to successfully apply Deep Learning
- Content
  - Orthogonalization (of the problems you're solving)
  - Setting up your goal
  - Comparing (NNs) to human-level performance
  - Error analysis
  - Mismatched train/dev/test set distributions
  - Data augmentation
  - Transfer & multi-task learning
- What I liked best: Choose a single real-valued evaluation metric (that matches your business goals) to help teams focus and succeed

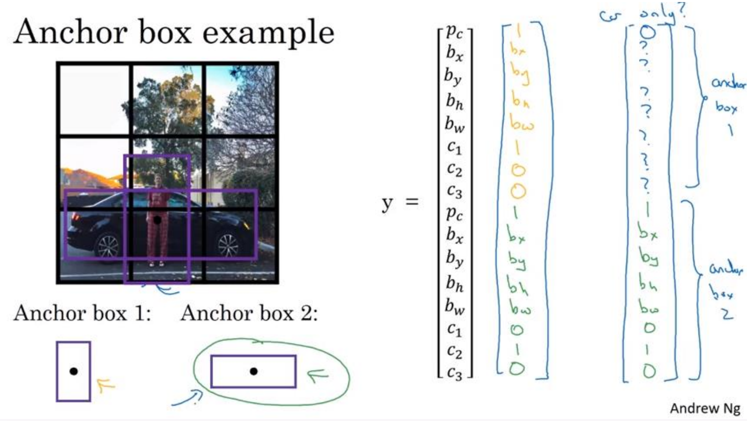
4. Convolutional Neural Networks

- How to solve computer vision problems with Deep Learning
- Content
  - Basic ConvNet theory
  - Classification + classic networks for the task
  - Object localization & detection (YOLO)
  - Face verification & recognition
  - Neural style transfer



Source: Szegedy et al. (Google)

What I liked best: YOLO made accessible



Anchor box example

Anchor box 1: Anchor box 2:

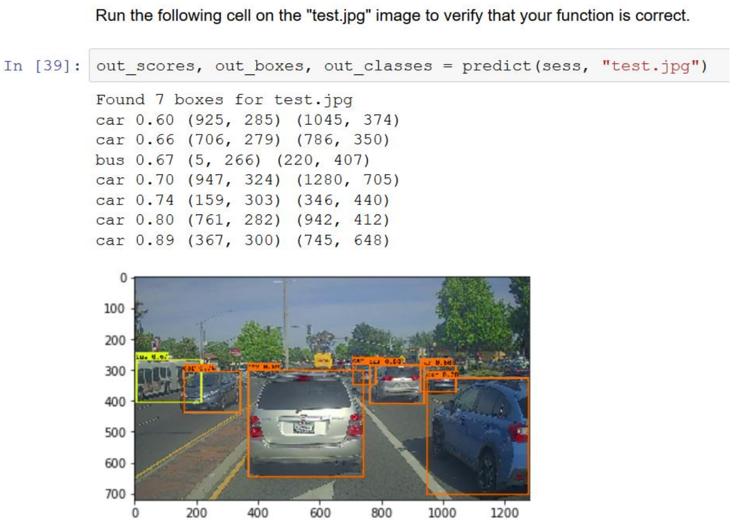
car only?

$$y = \begin{bmatrix} p_c \\ b_x \\ b_y \\ b_h \\ b_w \\ c_1 \\ c_2 \\ c_3 \\ p_c \\ b_x \\ b_y \\ b_h \\ b_w \\ c_1 \\ c_2 \\ c_3 \end{bmatrix} \quad \begin{bmatrix} 1 \\ b_x \\ b_y \\ b_h \\ b_w \\ 1 \\ 0 \\ 0 \\ 1 \\ b_x \\ b_y \\ b_h \\ b_w \\ 0 \\ 1 \\ 0 \end{bmatrix} \quad \begin{bmatrix} 0 \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \\ 1 \\ b_x \\ b_y \\ b_h \\ b_w \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

anchor box 1

anchor box 2

Andrew Ng



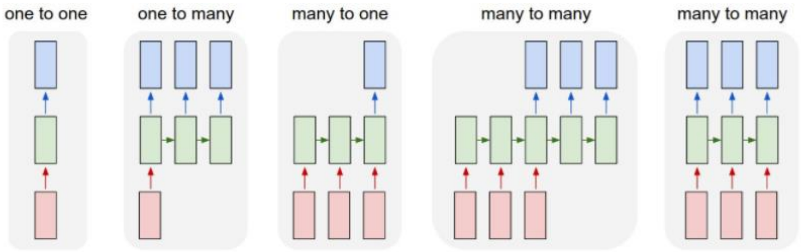
Run the following cell on the "test.jpg" image to verify that your function is correct.

In [39]:
```
out_scores, out_boxes, out_classes = predict(sess, "test.jpg")
```

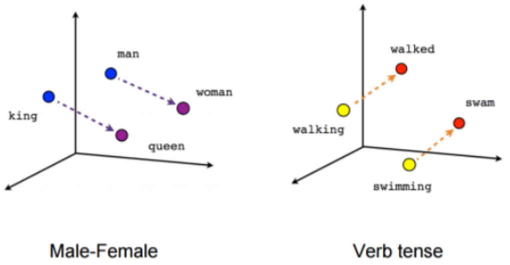
```
Found 7 boxes for test.jpg
car 0.60 (925, 285) (1045, 374)
car 0.66 (706, 279) (786, 350)
bus 0.67 (5, 266) (220, 407)
car 0.70 (947, 324) (1280, 705)
car 0.74 (159, 303) (346, 440)
car 0.80 (761, 282) (942, 412)
car 0.89 (367, 300) (745, 648)
```

5. Sequence Models

- How to solve questions around sequences (text, speech) with Deep Learning
- Content
  - Basic RNNS
  - LSTM & GRU
  - Attention
  - Word embeddings
  - Generative models
  - Speech recognition



Source: Andrej Karpathy



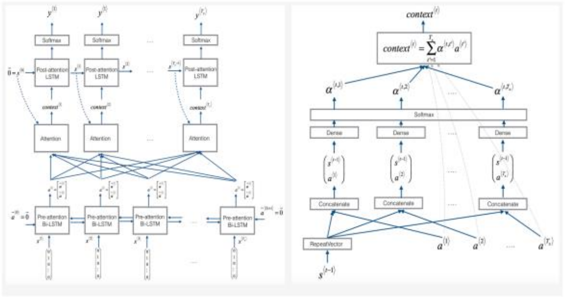
Source: Mikolov et al. (Google)

What I liked best: Applications of sequence models & Debiasing

Parsing dates with Bi-LSTMs and Attention

2.1 - Attention mechanism

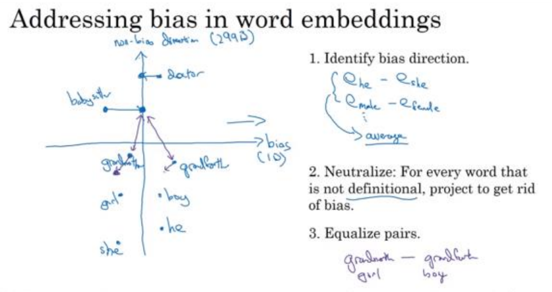
In this part, you will implement the attention mechanism presented in the lecture videos. Here is a figure to remind you how the model works. The diagram on the left shows the attention model. The diagram on the right shows what one "Attention" step does to calculate the attention variables $\alpha^{\langle t, t' \rangle}$, which are used to compute the context variable $context^{\langle t \rangle}$ for each timestep in the output ($t = 1, \ldots, T_y$).



Softmax
Post-attention LSTM
Attention
Pre-attention Bi-LSTM
$context^{\langle t \rangle} = \sum_{t'=1}^{T_x} \alpha^{\langle t,t' \rangle} a^{\langle t' \rangle}$
Softmax
Dense
Concatenate
RepeatVector

Figure 1: Neural machine translation with attention

De-biasing word embeddings



Addressing bias in word embeddings

1. Identify bias direction.

$e_{he} - e_{she}$
$e_{male} - e_{female}$
average

2. Neutralize: For every word that is not definitional, project to get rid of bias.

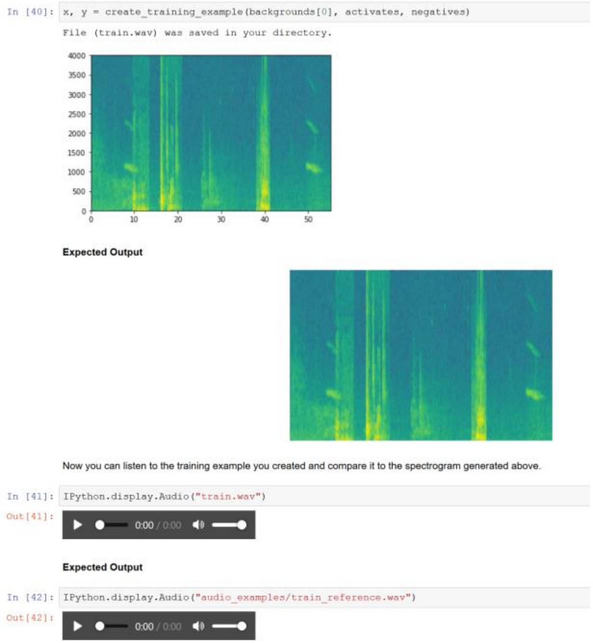
3. Equalize pairs.

grandmother — grandfather
girl — boy

[Bolukbasi et. al., 2016. Man is to computer programmer as woman is to homemaker? Debiasing word embeddings]

Andrew Ng

Trigger word detection



```
In [40]: x, y = create_training_example(backgrounds[0], activates, negatives)
         File (train.wav) was saved in your directory.
```

Expected Output

Now you can listen to the training example you created and compare it to the spectrogram generated above.

```
In [41]: IPython.display.Audio("train.wav")
Out[41]:
```

Expected Output

```
In [42]: IPython.display.Audio("audio_examples/train_reference.wav")
Out[42]:
```

Some more things I enjoyed, and my only real gripe

- Andrew Ng is really good at explaining concepts and making the topic feel accessible
- Heroes of Deep Learning interviews (Hinton, Bengio, Karpathy, …)
- Discussion Forums
- Only gripe: Fighting the autograder. Gets better the longer a course is online

Property of Automic Software. All rights reserved

What you can learn from this specialization

- Good understanding of terms, notation, fundamental algorithms and most important models of DL : Quizzes + Programming assignments ensure good understanding if you aim for a grade of 100%
  - you will be able to read research papers and enjoy the "what's new in Deep Learning" section of the meetup.
- Intuition about current state of DL, which problems you can apply it to, and how to set up a project to solve a business problem
- Andrew's up-to-date insights from his time in the industry

What you cannot learn from this specialization

- Data preparation
  => *Applied Data Science with Python* on Coursera
- End-to-end projects
  => *Deep Learning for Coders* at fast.ai (non-commercial), then Kaggle
- Applying DL to structured data
  => *Deep Learning for Coders* at fast.ai
- Deep understanding of the math behind DL => Read (many) textbooks or get a math degree

Applied Data Science with Python Specialization

Gain new insights into your data . Learn to apply data science methods and techniques, and acquire analysis skills.

About This Specialization

The 5 courses in this University of Michigan specialization introduce learners to data science through the python programming language. This skills-based specialization is intended for learners who have basic a python or programming background, and want to apply statistical, machine learning, information visualization, text analysis, and social network analysis techniques through popular python toolkits such as pandas, matplotlib, scikit-learn, nltk, and networkx to gain insight into their data.

Introduction to Data Science in Python (course 1), Applied Plotting, Charting & Data Representation in Python (course 2), and Applied Machine Learning in Python (course 3) should be taken in order and prior to any other course in the specialization. After completing those, courses 4 and 5 can be taken in any order. All 5 are required to earn a certificate.

Created by: UNIVERSITY OF MICHIGAN

Screenshot from Coursera.org

Screenshot from fast.ai

Conclusion

- I thoroughly enjoyed the specialization, it fit my personal learning style very well.
- It's really good at teaching the basic concepts. If you are trying to enter the field and enjoy a bottom-up style of learning, take the course
- If you enjoy a more practical and top/down approach: *Deep Learning for Coders* (fast.ai) might be for you
- If you're an experienced practitioner, consider skimming the *Structuring Machine Learning Projects* videos
- If you're interested in research, watch the *Heroes of Deep Learning* interviews