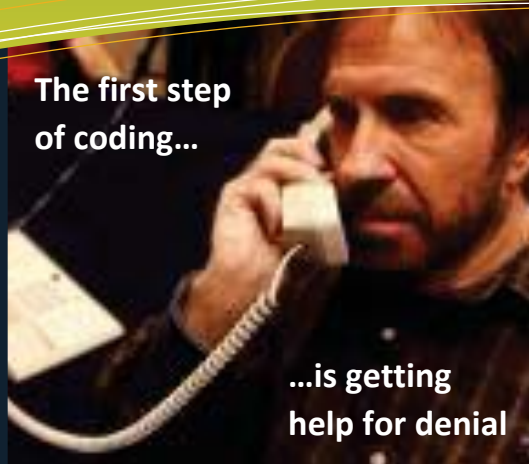


Data Analysis via Neural Networks

The first step
of coding...



...is getting
help for denial

WHERE CAN I GET A COPY OF YOUR SLIDES AND CODE?

Good question!
I would look here:

Why did you choose this?

To be honest, I wasn't sure exactly what I was getting myself into. I just knew that I liked the recent research on changing non-textual forms of data (e.g. image, audio) into a low-level computable data stream. I think it might have been the NLP assignment on three-lines in English that might have gave me the confidence for this to be my final assignment.

So, failure?

I had wanted to make my own ML-generated memes, but the code available was too complex to fix. I had to switch to a different application.

Flexibility is often key.

Exploring opportunities in Meme Text Generation.

In the beginning of searching for a topic to present for this course, I will admit I got stumped. I looked everywhere: pizza order databases on Kaggle[.com], Bollywood movie databases on github, UN population databases, and even average salary databases. But something about text did NOT catch my interest, and I wanted to try something different. Enter photos and images.

In the progress of doing first investigations on these databases, esp. the pizza ones, I found them unweildly and not so easy to code for with dedicated variables and classes in any TensorFlow compatible Python script. As the weeks of the course passed, so too did the amount of time I had left. At two weeks before the end of the course, I decided I was looking in the wrong direction and decided to look into a different area after hearing about it again on NHK Radio – image processing. I then considered what sort of images my Japanese co-classmates would never even imagine, and some western friends were quick to suggest that memes might be a great place to look (perhaps they were joking, in hindsight). Their joke was a goldmine anyway – one of the first Google Results was a recent Stanford CS224N NLP programming course report ... on memes.

In this Brief Report

Generating 'Meme' Text **P.1**

Where is your slides + code? **P.1**

pix2pix Neural Network **P.2**

The report talks about how humans have a way of generating certain text for memes, of course, primarily dependent on local and recent social influences. For example, as this report was being finalized during the early part of July, many memes that were shown as 'popular' and scraped by my code ([scraper.py](#)) focused on the American Independence Day. The week before, a preliminary run of a different scraper had produced results about the World Cup Football wins and losses.

The network, as shown in Peirson and Toulany's paper for the CS224N course, is described at being able to train with thousands of caption, photo, and basic labels... which when applied onto user-provided images, will identify features in the image, choose a similar meme, and then apply a fit-caption for the image.

That said, the process is shown to be expensive (even if all the code on the listed GitHub ran as expected) with the original authors running their code for a full almost 600 epochs. As I lack a GPU, time to do code refactoring, and could not even scrape from the websites without running into all sorts of other issues – I did abandon this idea. Still, as an idea it is interesting and holds sufficient merit to set-aside time to discuss its potential future applications.



pix2pix was the work of some architects who wanted to identify the most basic features of the world's buildings. Their solution was to sketch major features (shapes) of the building, color it in RGB depending on texture and depth aspects, and then train a neural network model.

Now, of course, pix2pix is a very useful software for a slightly different reason – it can create progressive scans of images w/ empty (whitespace) pixels. In fact, the tool is so good that the git repository I cloned the code from, [affinelayer/pix2pix-tensorflow](#),

has a photo of a cat in a tie with many pixels around the tie missing. Over generational analysis of the pixels aspects surrounding each white-area, we may see inputs on what a final image is predicted to be – indeed, more or less cat fur and a cotton tie.

My training model for the presentation used the classic architectural facades, mostly because I had previously used pix2pix on Google Earth Engine classified images in the past (via the MOLUSCE plugin). Additionally, I had seen from friends what to expect after one epoch, but wanted to see the result of a second epoch.

In the second epoch, I saw improved accuracy (which is to say, the computer algorithm's confidence) in the predictions and the images looked reasonably close to the building in a couple cases (like a blurry photo).

I also learned how to use TensorBoard to visualize the learning done by the Tensorflow algorithm. It did not disappoint. Above is data showing lowering uncertainty from the 1st-800th calculation. Indeed, another few thousand iterations wouldn't hurt.

“A cat in a tie, with pixels missing around the tie.”

Q: Why would anyone ever place a Hello World watermark on a cat?

A: Actually, I wish Marc Belmont had given an explanation in his github repository README file! While I am not sure, he mentioned it was created with a simple graphics editor in a GitHub issue, so I'm led to believe that he had added the watermark after selecting the image. In the future, I also hope that I can use this software, and one that can identify a perfect delta between my free image and watermarked image, so that I can remove some watermarks from images

depicting myself. Indeed, I think that the idea of using a TensorFlow convoluted -neural network (CNN) is a work of genius and admire the author for his thoughts on using a rotating rectangle shape as the initial variable in his python script. Perhaps with a touchup of modernization to solve for deprecated arguments from older versions of TensorFlow and it may be the best solution for my earlier issue. Wish me luck and follow my progress:



EYE ON IT

Good Resources not cited in the slide decks

Countless number of questions on errors and best-practices in scripting for python were searched and asked via [StackOverflow\[.com\]](#).

Google also provides many good resources for their products via their team of Developer Advocates making YouTube videos. One such video, by Yufeng Guo, was invaluable to allowing me to understand how to use TensorBoard with TensorFlow data. The video: www.youtube.com/watch?v=qEQ-_Eld-D0

Finally, some thanks to Dr. Wataru Takeuchi of the University of Tokyo for providing the idea for using pix2pix software in his course on Remote Sensing.

AFTERWORD

Last Thoughts

The course with Dr. Yamaguchi has given me some experience and insights I had previously lacked, including with programming languages such as SQL and a more in-depth theory discussion on data manipulation, esp. of Machine Learning. Hopefully I will also improve my skill of Python.

