Style Transfer

As an example of the kind of things you'll be building with deep learning models, here is a really fun project, fast style transfer. Style transfer allows you to take famous paintings, and recreate your own images in their styles! The network learns the underlying techniques of those paintings and figures out how to apply them on its own. This model was trained on the styles of famous paintings and is able to transfer those styles to other images and even videos!

Mat Leonard, the product lead for the school of AI, used it to style his cat Chihiro in the style of Hokusai's *The Great Wave Off Kanagawa*.

Mat's cat in the style of Hokusai.

To try it out yourself, you can find the code in the fast-style-transfer GitHub repo. Either use `git` to clone the repository, or you can download the whole thing as a Zip archive and extract it.

The network has been trained on a few different styles (here) and saved into checkpoint files. Checkpoint files contain all the information about the trained network to apply styles to new images.

Dependencies

The easiest way to install all the packages needed to run this code is with Miniconda, a smaller version of Anaconda. Miniconda comes with Conda, a package and environment manager built specifically for data science. Install the Python 3 version of Miniconda appropriate for your operating system.

If you haven't used Conda before, please quickly run through the Anaconda lesson (Lesson 3 on this part).

Windows

For Windows, you'll need to install TensorFlow 0.12.1, Python 3.5, Pillow 3.4.2, scipy 0.18.1, and numpy 1.11.2. After installing Miniconda, open your command prompt. In there, enter these commands line by line:

```
conda create -n style-transfer python=3
activate style-transfer
```

```
conda install tensorflow scipy pillow
pip install moviepy
python -c "import imageio; imageio.plugins.ffmpeg.download()"
```

OS X and Linux

For OS X and Linux, you'll need to install TensorFlow 0.11.0, Python 2.7.9, Pillow 3.4.2, scipy 0.18.1, and numpy 1.11.2.

In your terminal, enter this commands line by line:

```
conda create -n style-transfer python=3
activate style-transfer
conda install tensorflow scipy pillow
pip install moviepy
python -c "import imageio; imageio.plugins.ffmpeg.download()"
```

Let’s take a quick look at what these commands do. The first line in both sets of instructions, creates a new environment with Python 3. This environment will hold all the packages you need for the style transfer code. The next line enters the environment. Next, we install TensorFlow, SciPy, Pillow (which is an image processing library), and moviepy. The last line here installs ffmpeg, an application for converting images and videos.

Transferring styles

1. Download the Zip archive from (or clone) the fast-style-transfer repository and extract it. You can download it by clicking on the bright green button on the right.
2. Download the Rain Princess checkpoint from here. Put it in the fast-style-transfer folder. A checkpoint file is a model that already has tuned parameters. By using this checkpoint file, we won't need to train the model and can get straight to applying it.
3. Copy the image you want to style into the fast-style-transfer folder.
4. Enter the Conda environment you created above, if you aren't still in it.

Finally, in your terminal, navigate to the fast-style-transfer folder and enter

```
python evaluate.py --checkpoint ./rain-princess.ckpt --in-path
<path_to_input_file> --out-path ./output_image.jpg
```

Note: Your checkpoint file might be named `rain_princess.ckpt`, notice the underscore, it's not the dash from above.

You can get more checkpoint files at the bottom of this page. Try them all!

Share what you create on Slack channel #general or with all of us using #MadeWithUdacity. We'd love to see what you come up with! Also, feel free to train the network on your own images, you can find instructions in the repository (although it does take some powerful hardware).

Note: Be careful with the size of the input image. The style transfer can take quite a while to run on larger images.

Style Transfer Checklist

Task List

- Apply style transfer to an image of yourself or something personal to you.
- Share your image on Twitter with the #MadeWithUdacity hashtag.

The checkpoints were trained on the following paintings:

- Rain Princesss, by Leonid Afremov
- La Muse, by Pablo Picasso
- Udnie by Francis Picabia
- Scream, by Edvard Munch
- The Great Wave off Kanagawa, by Hokusai
- The Shipwreck of the Minotaur, by J.M.W. Turner

Supporting Materials

Rain-Princess

La-Muse

Udnie

Scream

Wreck

Wave

DeepTraffic

Another great application of deep learning is in simulating traffic and making driving decisions. You can find the DeepTraffic simulator here. The network here is attempting to learn a driving strategy such that the car is moving as fast as possible using reinforcement learning. The network is rewarded when the car chooses actions that result in it moving fast. It's this feedback that allows the network to find a strategy of actions for optimal speed.

To learn more about setting the parameters and training the network, read the overview here.

Discuss how you built your network and your results with your fellow students in the #general channel on Slack.

Flappy Bird

In this example, you'll get to see a deep learning agent playing Flappy Bird! You have the option to train the agent yourself, but for now let's just start with the pre-trained network given by the author. Note that the following agent is able to play without being told any information about the structure of the game or its rules. It automatically discovers the rules of the game by finding out how it did on each iteration.

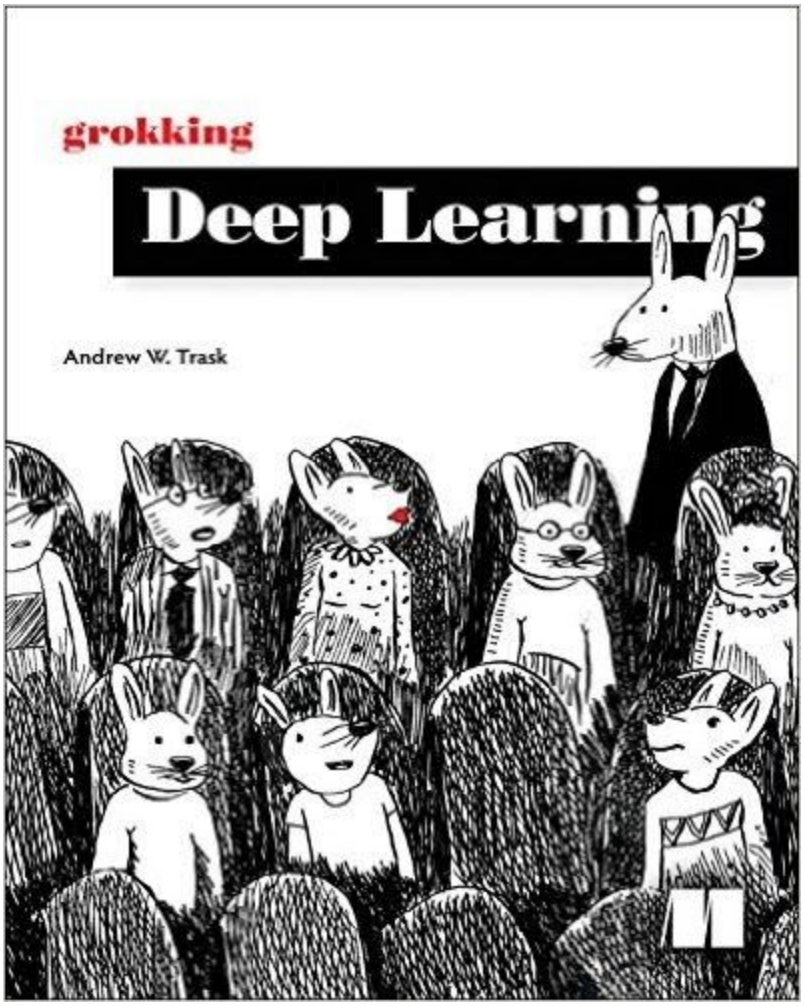
We will be following this repository by Yenchen Lin.

Instructions

1. Install miniconda or anaconda if you have not already.
2. Create an environment for flappybird
    - Mac/Linux: `conda create --name=flappybird python=2.7`
    - Windows: `conda create --name=flappybird python=3.5`
3. Enter your conda environment: `conda activate flappybird`
4. `conda install opencv`
    - If you encounter an error here, you may try an **alternate** download path and *instead* type `conda install --channel https://conda.anaconda.org/menpo opencv3`
5. `pip install pygame`
6. `pip install tensorflow==0.12`
7. `git clone https://github.com/yenchenlin/DeepLearningFlappyBird.git`
    - If you don't have git installed, you can download and extract the zip archive directly from the repository
8. `cd DeepLearningFlappyBird`
    - If you downloaded the archive, you will need to navigate to the extracted folder **DeepLearningFlappyBird-master** instead
9. `python deep_q_network.py`

If all went correctly, you should be seeing a deep learning based agent play Flappy Bird! The repository contains instructions for training your own agent if you're interested!

You can also, typically, force-quit out of the game by returning to your terminal window and typing `Command or Ctrl + C`.



Books to read

We believe that you learn best when you are exposed to multiple perspectives on the same idea. As such, we recommend checking out a few of the books below to get an added perspective on Deep Learning.

- Grokking Deep Learning by Andrew Trask. Use our exclusive discount code **traskud17** for 40% off. This provides a very gentle introduction to Deep Learning and covers the intuition more than the theory.
- Neural Networks And Deep Learning by Michael Nielsen. This book is more rigorous than Grokking Deep Learning and includes a lot of fun, interactive visualizations to play with.
- The Deep Learning Textbook from Ian Goodfellow, Yoshua Bengio, and Aaron Courville. This online book contains a lot of material and is the most rigorous of the three books suggested.