

Deep Learning

Introdução: Aplicando Deep Learning

Introdução

Nesta aula serão apresentadas algumas aplicações de *Deep Learning* usando modelos pré-treinados disponibilizados gentilmente por outros programadores no Github. O objetivo é mostrar o poder do *Deep Learning* em alguns contextos diferentes e também conhecer alguns destes modelos.

Transferência de Estilo

A [transferência de estilo](#) permite que você pegue suas fotos e as recrie à imagem de pinturas famosas. A rede aprende as técnicas subjacentes dessas pinturas e como aplicá-las corretamente sozinha. Esse modelo foi treinado nos modelos de estilos de pinturas famosas e então é capaz de transferir esse estilo para outras imagens e [até mesmo vídeos](#).

O exemplo ao lado transfere o estilo da obra [A grande onda de Kanagawa](#) de [Hokusai](#) para a imagem de um gato.



Transferência de Estilo

Para testar a aplicação, você pode encontrar o código no [repositório do fast-style-transfer no GitHub](#). Use então o `git` para clonar o repositório, ou então baixe tudo como um arquivo zipado e então extraia.

A rede foi treinada com alguns estilos diferentes ([aqui](#)) e salva nos [arquivos de checkpoint](#). Os arquivos de checkpoint contêm todas as informações sobre a rede treinada para aplicar em imagens novas.

Transferência de Estilo

Windows

No Windows, será necessário instalar o TensorFlow 0.12.1, Python 3.5, Pillow 3.4.2, scipy 0.18.1 e o numpy 1.11.2. Depois de instalar o Anaconda, abra o seu prompt de comando. Uma vez lá, digite os seguintes comandos uma linha de cada vez:

```
conda create -n style-transfer python=3.5
activate style-transfer
pip install tensorflow
conda install scipy pillow
```

A primeira linha cria um ambiente novo que irá conter todos os pacotes necessários para rodar o código do transfer de estilo. A linha seguinte (**activate style-transfer**) entra no ambiente, agora o nome do ambiente deveria aparecer antes do prompt de comando, escrito no começo da linha. As linhas seguintes instalam as bibliotecas TensorFlow, Scipy e Pillow (uma biblioteca específica para processamento de imagem).

Transferência de Estilo

OS X e Linux

Para o OS X e Linux, será necessário instalar o TensorFlow 0.11.0, Python 2.7.9, Pillow 3.4.2, scipy 0.18.1 e o numpy 1.11.2. No terminal, entre os seguintes comandos uma linha por vez:

```
conda create -n style-transfer python=2.7.9
source activate style-transfer
pip install tensorflow
conda install scipy pillow
```

A primeira linha cria um novo ambiente que irá conter os pacotes necessários para o código do transfer de estilo. A linha seguinte (`source activate style-transfer`) entra no ambiente, agora o nome do ambiente deveria aparecer antes do prompt de comando, escrito no começo da linha. As linhas seguintes instalam as bibliotecas TensorFlow, Scipy e Pillow (uma biblioteca específica para processamento de imagem).

Transferência de Estilo

Transferindo estilos

1. Baixe o arquivo Zip do repositório [fast-style-transfer](#) e então extraia. É possível baixá-lo ao clicar no botão verde brilhante na direita.
2. Baixe o checkpoint da Princesa da Chuva disponível [aqui](#). Coloque na pasta fast-style-transfer. Um arquivo checkpoint é um modelo que já tem os parâmetros ajustados. Ao usar esse arquivo checkpoint, não é necessário treinar o modelo e podemos então ir direto para sua aplicação.
3. Copie a imagem que você quer modificar para a pasta fast-style-transfer.
4. Entre no ambiente conda que você acabou de criar, caso tenha saído.
5. No terminal, navegue para a pasta fast-style-transfer e digite o seguinte comando

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

Transferência de Estilo

Os checkpoints foram treinados com as seguintes pinturas:

- Princesa da Chuva, de [Leonid Afremov](#)
- A Musa, de [Pablo Picasso](#)
- Udnie de [Francis Picabia](#)
- O grito, de [Edvard Munch](#)
- A grande onda de Kanagawa, de [Hokusai](#)
- O naufrago do Minotauro, de [J.M.W. Turner](#)

Material de apoio [Rain Princess checkpoint.](#) [La Muse checkpoint.](#) [Udnie checkpoint.](#) [Scream checkpoint.](#) [Wave checkpoint.](#) [Wreck checkpoint](#)

Deep Traffic

Essa [rede](#) está tentando aprender uma estratégia de direção que faça o carro se mover o mais rápido possível usando [aprendizado por reforço](#). A rede é premiada quando o carro escolhe ações que resultem em se mover mais depressa. É esse retorno que permite que a rede encontre a estratégia de ações que levem a uma velocidade otimizada.

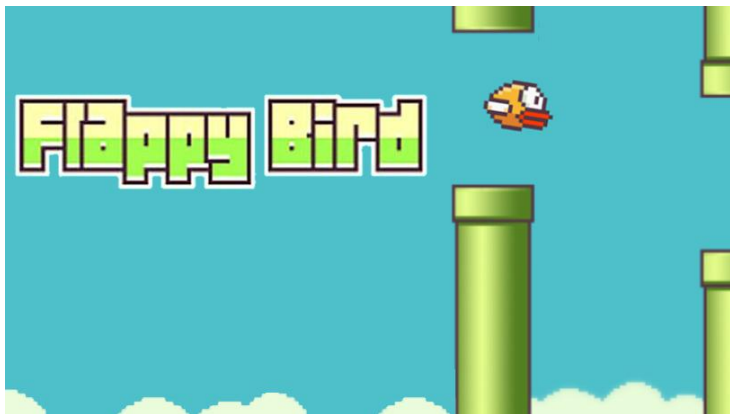
Assista o vídeo da Cezane Camacho no Youtube: https://www.youtube.com/watch?time_continue=12&v=az5ElmV4DhY

Para aprender mais sobre o ajuste de parâmetros e o treinamento da rede, leia o [resumo aqui](#).

Flappy Bird

Nesse exemplo, é mostrado um agente baseado em *Deep Learning* jogando Flappy Bird. O agente mostrado é capaz de jogar sem saber nenhuma informação sobre a estrutura do jogo ou suas regras. Ele automaticamente descobre as regras do jogo ao descobrir seu desempenho em cada interação.

Seguiremos [esse repositório](#) feito por Yenchen Lin.



Flappy Bird

Instruções

1. Entre no prompt Anaconda
2. Crie um ambiente para o flappybird
 - Mac/Linux: `conda create --name=flappybird python=2.7`
 - Windows: `conda create --name=flappybird python=3.5`
3. Entre no ambiente
 - Mac/Linux: `source activate flappybird`
 - Windows: `activate flappybird`
4. `conda install -c menpo opencv3`
5. `pip install pygame`
6. `pip install tensorflow`
7. `git clone https://github.com/yenchenlin/DeepLearningFlappyBird.git`
8. `cd DeepLearningFlappyBird`
9. `python deep_q_network.py`

Literaturas recomendadas

- [Grokking Deep Learning](#), de Andrew Trask.
- [Neural Networks And Deep Learning](#), de Michael Nielsen.
- [The Deep Learning Textbook](#), de Ian Goodfellow, Yoshua Bengio e Aaron Courville.