Machine Learning Engineer Nanodegree

Capstone Proposal

Bowen Chen
January 20, 2020

Project Objective

Build a full-stack recommendation system that could recommend single accessories to complete the best fashion looking using techniques in both deep learning and recommender systems.

Domain Background

As a big-time fan of Nike shoes, I have always been wondering why certain Air Force 1s look better than others with a certain outfit. In fact, I noticed a majority of the fashion trends are started by a certain celebrity - a superstar started to wear a certain combination of outfits and shoes, posted on his/her social media, and people start to dress themselves the same way. To stay in the trend, more and more people found dressing up becomes a laborious task - they have to search social media to find celebrities or other people wearing similar combinations of clothes and decide what kind of look they would like to rock today. What if there is an application that could help you decide which pair of shoes goes well with your newly acquired off-white streetwear? The convenience of having such an application will spare people from spending unnecessary time browsing social media to find similar items. In addition, this application would also help many single item outfit production companies drawing inspirations from public approved composites if they wish to expand their business lines. The outfit completion recommendation system will have a trained deep learning model within a flask application, which will be accessible via both mobile and desktop devices.

Problem Statement

The outfit completion recommendation system will contain a trained deep learning model that takes in an image of a person within a certain scene along with a text search query, such as “shoes” and return a ranked response set of shoe images that are compatible with the user’s current outfit. The model will then be packaged into a flask application and made available to the frontend through a RESTful API. The whole application will be hosted in a docker container, which could be deployed on an AWS EC2 instance.

Dataset Description

The dataset will be obtained from Pinterest's "Shop the Look" Fashion image data, which contains 47,339 unique scenes, 38,111 products and 72,198 product pairs. The dataset also includes the bounding boxes of the product, product category information and labels that have similar styles or the current product. Examples of the dataset is the following,

```
Example (fashion.json):
{
    "product": "0027e30879ce3d87f82f699f148bff7e",
    "scene": "cdab9160072dd1800038227960ff6467",
    "bbox": [
        0.434097,
        0.859363,
        0.560254,
        1.0
    ]
}
```

The string in "product" and "scene" represents a signature, which could be joined to a URL to retrieve the actual image. The function that could accomplish this goal is the following,

```
def convert_to_url(signature):
  prefix = 'http://i.pinimg.com/400x/%s/%s/%s/%s.jpg'
  return prefix % (signature[0:2], signature[2:4], signature[4:6], signature)
```

There is another file that contains the product categories in a dictionary format,

```
{"8870b0d384138a81cfdc91f98d1dde9b": "Apparel & Accessories|Shoes",
"70d392390fa258ba9b5d6b83ea29857e": "Apparel & Accessories|Shoes",
"2c05db9ff1abbcfc4bfd50dd003aac80": "Apparel & Accessories|Clothing|
Pants",…}
```

This dataset will be used to map the text search query category when predicting which item fits with the input image.

Technical Solution Description

The images in the dataset will be cropped using the bounding box given in their corresponding metadata, which removes the single item from the scene image. A convolutional neural network will be used as a visual feature extractor that transforms all the images into a n-dimensional vector. Then a compatibility measure will be constructed and the deep learning model will be trained in TensorFlow 2.0. The hyperparameter involved in this model will be searched using efficient search algorithms such as Bayesian Optimization. After evaluating the model recommendations by a popularity recommendation baseline model, the trained model will be packaged into a flask application framework, making the model accessible through a RESTful API.

Benchmark Model

The benchmark model will simply be recommending the most popular items based on the number of scene and product pairs. The for each product, the number of times it appears in a product scene pair will be counted, and the top n recommended products will be ranked in reverse order with by the value of the count.

Evaluation Metric

The setup of this project makes defining evaluation metrics relatively simple. Since the single item has been removed from the scene image, we will treat that item as the ground truth. The metric will simply be the accuracy of the model on % of times the recommender has picked the item originally removed from the image (the ground truth item).

Project Design

- Goal: Build a full-stack single item recommendation system using TensorFlow and Flask
- Programming Language: Python 3.7
  - ✓ Libraries:
  - ✓ Pandas
  - ✓ Numpy
  - ✓ Seaborn
  - ✓ TensorFlow 2.0
  - ✓ Flask

✓ Workflow
  - ◉ Data Preprocessing
    - ✓ Using the bounding box to in the image metadata to crop out the single item in the image
    - ✓ Perform a 85-10-5 train, validation and test split on the entire dataset

  - ◉ Implement the baseline model

    - ✓ Rank the recommendation result by the popularity of the item (value count of the number of times the single item existed in the entire dataset)

- ✓ Implement the improved model
    - ✓ Visual feature extraction using either ResNet-50 or VGG-19 (pretrained on ImageNet), transform the visual features into a style embedding using the Graham Matrix transformation, which unifies the style vector length to 1
    - ✓ Define the compatibility measure using the triplet loss within the style embedding space
    - ✓ Model training using Adam optimization
    - ✓ Compatibility score assignment on all single items
    - ✓ Model archiving

- ✓ API construction using Flask

Reference

- Complete the Look: Scene-Based Complementary Product Recommendation (https://arxiv.org/pdf/1812.01748.pdf)

- Developing Art Style Embeddings for Visual Similarity Comparison (https://towardsdatascience.com/developing-art-style-embeddings-for-visual-similarity-comparison-of-artworks-7a9d4ade2045)

- Simply way to deploy machine learning models to cloud (https://towardsdatascience.com/simple-way-to-deploy-machine-learning-models-to-cloud-fd58b771fdcf)

- The brilliant beginner's guide to model deployment (https://heartbeat.fritz.ai/brilliant-beginners-guide-to-model-deployment-133e158f6717)