MLND Capstone Project Description - Deep Learning

Deep Learning Capstone Project

Build a Live Camera App

Description

Objective: Build a live camera app that can interpret number strings in real-world images.

→ 31

In this project, you will train a model that can decode sequences of digits from natural images, and create an app that prints the numbers it sees in real time. You may choose to implement your project as a simple Python script, a web app/service or an Android app.

Setup

Recommended setup for a simple Python script or web app/service:

- Python
- NumPy, SciPy, iPython
- TensorFlow™
- (Optional) OpenCV / SimpleCV / Pygame (to capture camera images)

(Optional) For deploying the model in an Android app:

- Android SDK & NDK (see this README)

Data

Street View House Numbers (SVHN): A large-scale dataset of house numbers in Google Street View images.

Tasks

There are several steps involved in achieving this. As you make progress, keep documenting your work in a project report. Look for specific *prompts* below that you must address.

Step 1: Design and test a model architecture that can identify sequences of digits in an image.

Design and implement a deep learning model that learns to recognize sequences of digits. Train it using synthetic data first (recommended) or directly use real-world data (see step 2).

There are various aspects to consider when thinking about this problem:

- Your model can be derived from a deep neural net or a convolutional network.
- You could experiment sharing or not the weights between the softmax classifiers.
- You can also use a recurrent network in your deep neural net to replace the classification layers and directly emit the sequence of digits one-at-a-time.

To help you develop your model, the simplest path is likely to generate a synthetic dataset by concatenating character images from notMNIST or MNIST. This can provide you with a quick way to run experiments. (Or you can go directly to the real-world dataset of Step 2.)

In order to produce a synthetic sequence of digits for testing, you can for example limit yourself to sequences up to five digits, and use five classifiers on top of your deep network. You would have to incorporate an additional 'blank' character to account for shorter number sequences.

Here is for example a published baseline model on this problem (video).

What approach did you take in coming up with a solution to this problem?
What does your final architecture look like? (Type of model, layers, sizes, connectivity, etc.)
How did you train your model? Did you generate a synthetic dataset (if so, explain how)?

Step 2: Train a model on a realistic dataset.

Once you have settled on a good architecture, you can train your model on real data. In particular, the SVHN dataset is a good large scale dataset collected from house numbers in Google Street View. Training on this more challenging dataset, where the digits are not neatly lined-up and have various skews, fonts and colors, likely means you have to do some hyperparameter exploration to do well.

How does your model perform on a realistic dataset?
What changes did you have to make, if any?

Step 3 (optional): Put the model into an Android app.

Do this step only if you have access to an Android device. If you don't, you may either:
i) take pictures of numbers that you find around you, and run them through your classifier on your computer to produce example results, or,
ii) use OpenCV / SimpleCV / Pygame to capture live images from a webcam.

Loading a TensorFlow model into a camera app on Android is demonstrated in the TensorFlow Android demo app, which you can simply modify.

Is your model able to perform equally well on captured pictures or a live camera stream?
Document how you built the interface to your model.

Step 4: Explore!

There are many things you can do once you have the basic classifier in place. One example would be to also localize where the numbers are on the image. The SVHN dataset provides bounding boxes that you can tune to train a localizer. Simply training a regression loss to the coordinates of the bounding box is one way to get decent localization.

Once you have the data localized, you can for example try turn it into an augmented reality app by overlaying your answer on the image like the Word Lens app does.

Those are just examples of extensions you can look into. Use your imagination!

Make sure to report what extension(s) you have implemented and how they worked.

Rubric

Please refer to the generic Capstone Project Rubric for the different components your submission will be evaluated upon. In addition, ensure that you have addressed all the *prompts* included above in your report. Only submissions that meet these requirements will be passed.

Deliverables

The following files should be included in your submission, and can be packaged as a single zip file for convenience:

- A 9 - 15 page report in PDF addressing the five major project development phases:
  - Definition
  - Analysis
  - Methodology
  - Results
  - Conclusion
- All development code necessary for your solution in a clean, commented format.
- Any required supporting files for your code (.csv datasets, input files, etc.).
  - *If these files are too large, please instead reference appropriate download links in the README.*
- A README documenting the software and libraries used in your project, including any necessary instructions required to execute your code.

Your project document should adhere to a similar structure as mentioned in the capstone project overview and in the project template.

Learning Resources

Courses

- Deep Learning, Vincent Vanhoucke (Google and Udacity).

Books

- Goodfellow, I., Bengio, Y. and Courville, A. *Deep Learning*. MIT Press, 2016 (in prep.).

Papers

● Netzer, Y., et al. Reading Digits in Natural Images with Unsupervised Feature Learning. *NIPS Workshop on Deep Learning and Unsupervised Feature Learning*, 2011.
● Goodfellow, I., et al. Multi-digit Number Recognition from Street View Imagery using Deep Convolutional Neural Networks. *ICLR*, 2014. [video]

Resources

● UFLDL Deep Learning Tutorial
● DeepLearning.net: A collection of papers, software, dataset and current events.
● Deep Learning Summer School, Montreal 2015