SVHN Sequence Detector

Machine Learning Nanodegree Capstone Project

Summer 2016

—

Sergio Gordillo
@Sergio_Gordillo
github.com/archelogos/sequence-detector

Definition

Project Overview

Deep Learning is currently one of the most interesting fields in Machine Learning. The combination of Neural Networks and powerful computation systems is extremely useful to solve complicated Pattern Recognition or Text Classification problems.

The main goal of this project is, thanks to Deep Learning techniques, be able to detect and identify sequences of digits in a random picture. Especially in this project, the sequences correspond to the number of houses.

In order to simplify the explanation, the desired result is shown in the following picture.

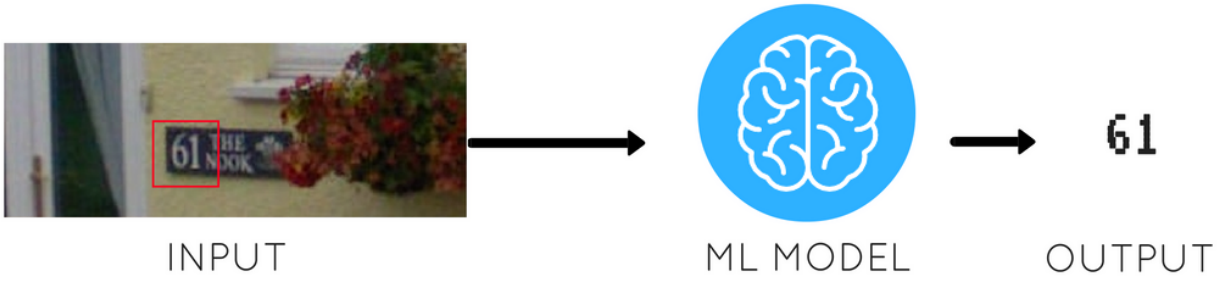

Figure 1: Project goal overview

The data used to train our model belongs to the SVHN dataset, which is a real-world image dataset for developing machine learning and object recognition and is obtained from house numbers in Google Street View images.

This is a real-world problem studied for many years and it wasn't until the application of neural networks to the images recognition problems when it could be considered as solved. At this moment it can seen real products and apps that use this technology (i.e. http://questvisual.com/).

Problem Statement

As it was said, the main goal is to define, build and train a stable and consistent ML Model which can identify sequences of numbers of house in a particular image.

Figure 2: Samples of images from the dataset

To reach this goal it was used the mentioned dataset and it was created a model based on a Neural Network. The final model has been built taking different ideas from investigation papers, tutorials and practices.

The model begins being as basic as it's possible based on a simple Logistic Regression. Along the project, the model is getting more complicated and sophisticated step by step, becoming a 3L ConvNet production-ready at the end of the project.

In the process, different techniques, algorithms and solutions are tried and properly discussed.

The project could be divided in 7 steps:

1. Familiarize with TensorFlow, applying from a Logistic Regression to a 3L ConvNet model to a **single-digit-MNIST** dataset.
2. Modify the previous model and apply it to a **single-digit-SVHN** dataset.
3. Improve the model and apply it to a **sequence-MNIST** dataset.
4. Modify the previous model and apply it to a **sequence-SVHN** dataset.
5. Define a stable and consistent model
6. Train the final model in Google Cloud Platform in order to boost its performance
7. Build and deploy a web app which demonstrates the objective of this project.

The human ability of identify correctly the numbers of SVHN dataset is approximately 98% and the most important results to this problem given by companies like Google have almost reached that goal.

Convolutional Neural Networks aren't extremely efficient and are not easy to train properly because they are expensive in terms of computation resources. For that reason the first steps of the project are developed in ipython notebooks and making small training sessions. Once it was clear that the model works fine, it was moved to the Public Cloud to train it in a better way (highcpu machines running py scripts). Anyway it's important to notice that the purpose of this project is not to improve the current best models built for this problem but it's to study and build a "not-bad" and production-ready model which can perfectly obtain a 96% of accuracy identifying sequences from pictures (which in the field of Neural Networks is a huge difference with the commented 98%).

Metrics

As it was mentioned, the accuracy is the main metric defined for this problem.

It can be defined as the probability of coincidence between the real number in the image and the number predicted from the model.

A prediction it's considered correct if all the digits in the sequence and the length of the sequence are exactly the same, in other cases the prediction is not correct.

To check that the model is working properly it was set a periodic checks on the training and validation dataset along the training. At the end of the training it's always estimate the current accuracy of the model on the test dataset.

```
## Pseudo python
def accuracy(labels, predictions):
    n_accuracy = summatory(predictions == labels) / length(labels)
    p_accuracy = 100 * n_accuracy
    return p_accuracy
```

Figure 3: Pseudo code of accuracy function used

Analysis

Data Exploration

In this section, you will be expected to analyze the data you are using for the problem. This data can either be in the form of a dataset (or datasets), input data (or input files), or even an environment. The type of data should be thoroughly described and, if possible, have basic statistics and information presented (such as discussion of input features or defining characteristics about the input or environment). Any abnormalities or interesting qualities about the data that may need to be addressed have been identified (such as features that need to be transformed or the possibility of outliers). Questions to ask yourself when writing this section:

- *If a dataset is present for this problem, have you thoroughly discussed certain features about the dataset? Has a data sample been provided to the reader?*
- *If a dataset is present for this problem, are statistics about the dataset calculated and reported? Have any relevant results from this calculation been discussed?*
- *If a dataset is not present for this problem, has discussion been made about the input space or input data for your problem?*
- *Are there any abnormalities or characteristics about the input space or dataset that need to be addressed? (categorical variables, missing values, outliers, etc.)*

Exploratory Visualization

In this section, you will need to provide some form of visualization that summarizes or extracts a relevant characteristic or feature about the data. The visualization should adequately support the data being used. Discuss why this visualization was chosen and how it is relevant. Questions to ask yourself when writing this section:

- *Have you visualized a relevant characteristic or feature about the dataset or input data?*
- *Is the visualization thoroughly analyzed and discussed?*
- *If a plot is provided, are the axes, title, and datum clearly defined?*

Algorithms and Techniques

In this section, you will need to discuss the algorithms and techniques you intend to use for solving the problem. You should justify the use of each one based on the characteristics of the problem and the problem domain. Questions to ask yourself when writing this section:

- *Are the algorithms you will use, including any default variables/parameters in the project clearly defined?*
- *Are the techniques to be used thoroughly discussed and justified?*
- *Is it made clear how the input data or datasets will be handled by the algorithms and techniques chosen?*

Benchmark

In this section, you will need to provide a clearly defined benchmark result or threshold for comparing across performances obtained by your solution. The reasoning behind the benchmark (in the case where it is not an established result) should be discussed. Questions to ask yourself when writing this section:

- *Has some result or value been provided that acts as a benchmark for measuring performance?*
- *Is it clear how this result or value was obtained (whether by data or by hypothesis)?*

Methodology

Data Preprocessing

In this section, all of your preprocessing steps will need to be clearly documented, if any were necessary. From the previous section, any of the abnormalities or characteristics that you identified about the dataset will be addressed and corrected here. Questions to ask yourself when writing this section:

- *If the algorithms chosen require preprocessing steps like feature selection or feature transformations, have they been properly documented?*
- *Based on the Data Exploration section, if there were abnormalities or characteristics that needed to be addressed, have they been properly corrected?*
- *If no preprocessing is needed, has it been made clear why?*

Implementation

In this section, the process for which metrics, algorithms, and techniques that you implemented for the given data will need to be clearly documented. It should be abundantly clear how the implementation was carried out, and discussion should be made regarding any complications that occurred during this process. Questions to ask yourself when writing this section:

- *Is it made clear how the algorithms and techniques were implemented with the given datasets or input data?*
- *Were there any complications with the original metrics or techniques that required changing prior to acquiring a solution?*
- *Was there any part of the coding process (e.g., writing complicated functions) that should be documented?*

Refinement

In this section, you will need to discuss the process of improvement you made upon the algorithms and techniques you used in your implementation. For example, adjusting parameters for certain models to acquire improved solutions would fall under the refinement category. Your initial and final solutions should be reported, as well as any significant intermediate results as necessary. Questions to ask yourself when writing this section:

- *Has an initial solution been found and clearly reported?*
- *Is the process of improvement clearly documented, such as what techniques were used?*
- *Are intermediate and final solutions clearly reported as the process is improved?*

Results

Model Evaluation and Validation

In this section, the final model and any supporting qualities should be evaluated in detail. It should be clear how the final model was derived and why this model was chosen. In addition, some type of analysis should be used to validate the robustness of this model and its solution, such as manipulating the input data or environment to see how the model's solution is affected (this is called sensitivity analysis). Questions to ask yourself when writing this section:

- *Is the final model reasonable and aligning with solution expectations? Are the final parameters of the model appropriate?*
- *Has the final model been tested with various inputs to evaluate whether the model generalizes well to unseen data?*
- *Is the model robust enough for the problem? Do small perturbations (changes) in training data or the input space greatly affect the results?*
- *Can results found from the model be trusted?*

Justification

In this section, your model's final solution and its results should be compared to the benchmark you established earlier in the project using some type of statistical analysis. You should also justify whether these results and the solution are significant enough to have solved the problem posed in the project. Questions to ask yourself when writing this section:

- *Are the final results found stronger than the benchmark result reported earlier?*
- *Have you thoroughly analyzed and discussed the final solution?*
- *Is the final solution significant enough to have solved the problem?*

Conclusion

Free-Form Visualization

In this section, you will need to provide some form of visualization that emphasizes an important quality about the project. It is much more free-form, but should reasonably support a significant result or characteristic about the problem that you want to discuss. Questions to ask yourself when writing this section:

- *Have you visualized a relevant or important quality about the problem, dataset, input data, or results?*
- *Is the visualization thoroughly analyzed and discussed?*
- *If a plot is provided, are the axes, title, and datum clearly defined?*

Reflection

In this section, you will summarize the entire end-to-end problem solution and discuss one or two particular aspects of the project you found interesting or difficult. You are expected to reflect on the project as a whole to show that you have a firm understanding of the entire process employed in your work. Questions to ask yourself when writing this section:

- *Have you thoroughly summarized the entire process you used for this project?*
- *Were there any interesting aspects of the project?*
- *Were there any difficult aspects of the project?*
- *Does the final model and solution fit your expectations for the problem, and should it be used in a general setting to solve these types of problems?*

Improvement

In this section, you will need to provide discussion as to how one aspect of the implementation you designed could be improved. As an example, consider ways your implementation can be made more general, and what would need to be modified. You do not need to make this improvement, but the potential solutions resulting from these changes are considered and compared/contrasted to your current solution. Questions to ask yourself when writing this section:

- *Are there further improvements that could be made on the algorithms or techniques you used in this project?*
- *Were there algorithms or techniques you researched that you did not know how to implement, but would consider using if you knew how?*
- *If you used your final solution as the new benchmark, do you think an even better solution exists?*

Before submitting your report, ask yourself…

Does the project report you've written follow a well-organized structure similar to that of the project template?

Is each section (particularly Analysis and Methodology) written in a clear, concise and specific fashion? Are there any ambiguous terms or phrases that need clarification?

Would the intended audience of your project be able to understand your analysis, methods, and results?

Have you properly proof-read your project report to assure there are minimal grammatical and spelling mistakes?

Are all the resources used for this project correctly cited and referenced?

Is the code that implements your solution easily readable and properly commented?

Does the code execute without error and produce results similar to those reported?