

PROJECT

Machine Learning Capstone Project

A part of the Machine Learning Engineer Nanodegree Program

PROJECT REVIEW

CODE REVIEW

NOTES

SHARE YOUR ACCOMPLISHMENT!  

Requires Changes

11 SPECIFICATIONS REQUIRE CHANGES

Dear student

This was an excellent project to review! You've clearly put a lot of time and thought into this project and the result shows it. I've noted a few things you should change or add to the report, but you're quite close to passing at this point. Really great work so far...keep going!

Cheers!

Definition

Student provides a high-level overview of the project in layman's terms. Background information such as the problem domain, the project origin, and related data sets or input data is given.

This is a really great overview of the field! Please be sure to provide a link/citation to the dataset and a bit of background information on the dataset itself (for instance, how was it created?). It would also be great if you tailored your explanation of this type of machine learning to be a little more focused on this specific type of vision learning (but this is an optional consideration). Citing some

literature from the field would also be a great idea here!

The problem which needs to be solved is clearly defined. A strategy for solving the problem, including discussion of the expected solution, has been made.

Your current Problem statement section gives a great overview of the task to be accomplished. You should also give a 'top-to-bottom' overview of how you plan to solve it. This obviously doesn't need to be in any real depth, but please be sure to address what the steps are that you'll take to solve this problem and it's a good idea to mention anything innovative you'll be trying that might be interesting to the reader. You can also use this section to begin to justify why your approach is a good 'fit' for this problem or dataset.

(We're looking for something like your "General Approach" bullet points here, but with just a bit more detail)

Metrics used to measure performance of a model or result are clearly defined. Metrics are justified based on the characteristics of the problem.

These are great metrics to use for this problem. Please be sure to:

- Provide a formula for entropy loss.
- Provide a bit more justification for why you've chosen these metrics. For instance, is accuracy a good metric here? It doesn't perform well on unbalanced datasets...why did you choose accuracy instead of an F1 metric?

Analysis

If a dataset is present, features and calculated statistics relevant to the problem have been reported and discussed, along with a sampling of the data. In lieu of a dataset, a thorough description of the input space or input data has been made. Abnormalities or characteristics about the data or input that need to be addressed have been identified.

Great job exploring some of the peculiarities of this dataset. Please be sure to provide a few statistics about the dataset itself...perhaps mean intensity value or std. deviation for intensities?

A visualization has been provided that summarizes or extracts a relevant characteristic or feature about the dataset or input data with thorough discussion. Visual cues are clearly defined.

These (Distribution of length of sequences, Distribution of digits among the training images) figures are exactly what we're looking for. Please make sure that the axes are all labeled to clearly indicate what the numbers mean.

Algorithms and techniques used in the project are thoroughly discussed and properly justified based on the characteristics of the problem.

Please provide a bit more information about these algorithms. You should be identifying their strengths and weaknesses and talking a bit about why they are appropriate for this particular problem. Also, please be sure to provide a bit more detail about how they work (a visual aid might be useful for this). For instance, how does the convolution process work? What are regularization, optimizers, and dropout and what do they have to do with a processing an image?

Student clearly defines a benchmark result or threshold for comparing performances of solutions obtained.

This is a clear benchmark and I think your justification is a reasonable one. Another option you might consider is to pick a method as your benchmark. This allows you to run tests where you can compare methods objectively.

Methodology

All preprocessing steps have been clearly documented. Abnormalities or characteristics about the data or input that needed to be addressed have been corrected. If no data preprocessing is necessary, it has been clearly justified.

Looks great! Just a couple more things:

- Please be a bit more specific about your normalization process. Is this linear, exponential etc?
- How are the data shuffled to create training/validation/testing subsets? Is it random? Is anything done to prevent creating further imbalances in the dataset?

The process for which metrics, algorithms, and techniques were implemented with the given datasets or input data has been thoroughly documented. Complications that occurred during the coding process are discussed.

Please be sure to fully detail the parameters and settings that you've used. You address a number of

parameters like learning rate, batch size, dropout rates etc. in the report. Please be sure to document what these parameters were set to during the different implementations described in your report. Ideally, a skilled programmer should be able to (more or less) reproduce your results by just reading your report.

I'd recommend going through your code to make sure that each technique and parameter is documented in the report.

The process of improving upon the algorithms and techniques used is clearly documented. Both the initial and final solutions are reported, along with intermediate solutions, if necessary.

Great job! These are all logical steps to take to improve your model. Be sure to document the specific parameter settings that you used in this section for each implementation (for instance what was the dropout rate(s)?). Of course you don't need to document anything that didn't change from the original implementation.

Results

The final model's qualities — such as parameters — are evaluated in detail. Some type of analysis is used to validate the robustness of the model's solution.

Please be sure to address the qualities/parameters of the model as part of an explanation about why your solution is a robust one. For instance, the "Final Architecture" section would be perfect for this, if you used some of these bullet points to explain why you think your model is a really strong solution. Feel free to also address any parameters that you tweaked during the 'Refinement' section as well to bolster this answer.

The final results are compared to the benchmark result or threshold with some type of statistical analysis. Justification is made as to whether the final model and solution is significant enough to have adequately solved the problem.

I could achieve 75% test accuracy on 1000 test samples, training for only 300 epochs and using only about 30000 training samples. This sounds reasonable for a simple architecture and low end hardware...

Please be sure to specifically reference your benchmark here. You're reporting success! Also, it might be a good idea to document the results from your different implementations in a table so that the reader can look over all your results in an easy-to-compare format.

Conclusion

A visualization has been provided that emphasizes an important quality about the project with thorough discussion. Visual cues are clearly defined.

Great job! I think the visualization in the Justification section actually meets this requirement a bit better, and I'm marking this as meeting the specification since you discussed what it indicates about the project. I'd recommend providing a bit more discussion for the 'Free-form visualization' as well, but since I think the other Figure is meeting this requirement, I'm not marking this as required.

Student adequately summarizes the end-to-end problem solution and discusses one or two particular aspects of the project they found interesting or difficult.

Nice job! Note that this section is your chance to brag about anything innovative or exciting that you were able to implement, so feel free to highlight anything you're proud of.

Discussion is made as to how one aspect of the implementation could be improved. Potential solutions resulting from these improvements are considered and compared/contrasted to the current solution.

Quality

Project report follows a well-organized structure and would be readily understood by its intended audience. Each section is written in a clear, concise and specific manner. Few grammatical and spelling mistakes are present. All resources used to complete the project are cited and referenced.

This project report is well-written and demonstrates a good attention to detail. There are a few things that should be added or adjusted, but they're really quite minor. Nice work!

Code is formatted neatly with comments that effectively explain complex implementations. Output produces similar results and solutions as to those discussed in the project.

The code is implemented well. The preprocessing notebook is especially user-friendly (I love the progress hook!). The code appears to be well commented throughout and the code appears to run as described.

Suggestion: Please be sure to document all dependencies and libraries the reader will need in the README file.

 RESUBMIT

 [DOWNLOAD PROJECT](#)



Best practices for your project resubmission

Ben shares 5 helpful tips to get you through revising and resubmitting your project.

 [Watch Video](#) (3:01)

Have a question about your review? Email us at review-support@udacity.com and include the link to this review.

[RETURN TO PATH](#)