UDACITY

‹ Back to Machine Learning Engineer Nanodegree

Machine Learning Capstone Project

REVIEW

CODE REVIEW

HISTORY

Meets Specifications

This is a very cool analysis and a great read to a very real world and practical problem. You have demonstrated a full understanding of the entire machine learning pipeline and your report definitely gets the readers attention with the results you have achieved.

Hopefully you have learned a bunch throughout this capstone project(as I can image that you have by reading your report) and you can take some of these techniques further.

If this is your final report, I would like to be the first one to congratulate you on completing this nano-degree! Wish you the best of luck in your future!

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.

Very solid opening section here, as you have done a great job describing the problem and that machine learning can solve this. Image classification is definitely growing in terms to different industries.

And you have provided good research to back your claims. It is always important to provide similiar research on such a topic.

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.

> "The problem this project addresses is how to build a model or algorithm that is able to capture the relationship between brain activity and EEG signals in such a way that building effective BMIs for disabled patients can become feasible."

Problem statement is clearly defined here. Would also recommend explicitly mentioning that this would be a classification problem in this section.

And very nice job mentioning you machine learning pipeline in your **Solution Statement** section, as this gives the reader some ideas in what is to come in your report and how you plan on solving this important task.

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

> "The reason for this choice is that the mean column-wise AUC discourages the model from not being very discriminative. This is specially valuable when one class (one type of movement) occurs for a very long period of time. If we were to use accuracy instead of AUC, the model could likely start to predict that in most instances (or most periods of time) one class of movement is the most likely to occur"

This is excellent justification for your choice in AUC. Really glad that you tie your metric choice into this particular problem domain and dataset.

Ideally, you should also describe how AUC is computed. A visual could be a nice touch here as well.

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.

Very nice job describing your dataset. Glad that you show some descriptive stats, show a sample of your data, go into a bit of detail in the features here and the distribution of the target variable. As this allows the reader to get an understanding of the structure of the data you are working with.

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.

Suggestion

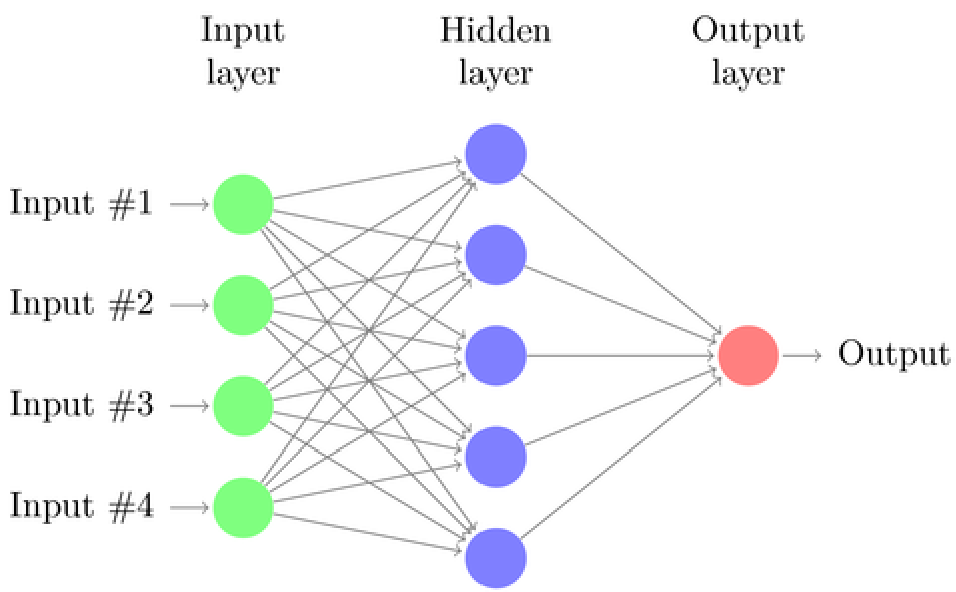
- Would be nice to include a title and X and Y labels in Figure 7 plots here. The reason for this is to clear depict to the reader what you are plotting.

You might also check out using some more advanced plotting libraries such as

- plot.ly: Modern Visualization for the Data Era. Where you can create really cool interactive visuals in jupyter notebooks and web apps!
- seaborn: statistical data visualization. Python data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics. My favorite method in seaborn is catplot(), where you can plot categorical features effortlessly!

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

Excellent job describing your main model here, as it is very clear that you have a solid understanding in how these models work. Maybe even some visuals could help explain these as well.



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

Good choice in benchmark models, I would agree that comparing the results both with similar and different models is a good idea.

Benchmarking is the process of comparing your result to existing method or running a very simple machine learning model, just to confirm that your problem is actually 'solvable'.

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.

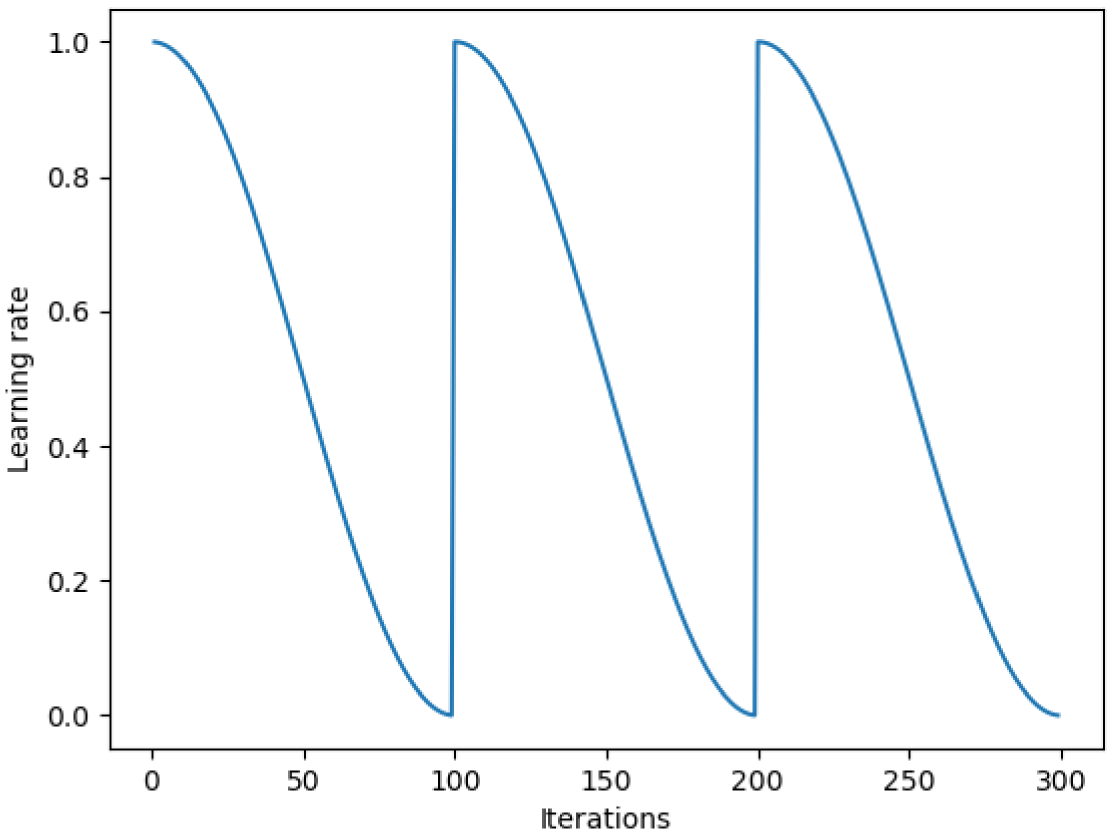
Awesome analysis of data standardization, this is always needed in NN models.

Another feature transformation idea, akin to standardization, for preparing continuous features for neural networks is with GaussRank. The purpose of this is to make the distribution of the transformed ranks Gaussian. Can check how this is done here.
(https://github.com/zygmuntz/gaussrank/blob/master/gaussrank.py)

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.

Very solid step by step process here, as it is quite clear in how you approached this problem. Your results would definitely be replicable.

Another idea would be to check out using Cyclical Learning Rates for Training Neural Networks. This is where we simply keep increasing the learning rate from a very small value, until the loss stops decreasing and then bump it up once more. We can plot the learning rate across batches to see what this looks like.



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.

Nice work with your hyper-parameter tuning ideas, as this is a great way to improve your model. And you have made it very clear in the parameter you tried and the results.

> "Another initial test consisted of evaluating a validation series from a specific subject using a model fit with only data from the same subject versus a model fit with all data. Initially we thought that fitting one model per subject could yield better results, but it was not the case."

Maybe one other idea would be to build multiple models (maybe one or two of each?) and 'ensemble' all of them together. As we can typically 'squeeze out' a few more percentage points by doing so. Check get an idea in how this would be done here (https://github.com/Hvass-Labs/TensorFlow-Tutorials/blob/master/05_Ensemble_Learning.ipynb)

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.

Nice discussion of your final model here.

And excellent way to validate the robustness of the model solution by analyzing the AUC in the validation set separately for each label and for each subject. Love that you also show the standard deviation!!!

Maybe one other idea would be to plot and 95% confidence interval to determine if the model is robust with bootstrapping.

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.

Conclusion

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

This is a great visual to show. Clearly depicts an important quality about the project.

This is typically the most important part in determining 'where' your model went wrong and how to improve it.

Another really cool idea would be to check out the SHAP library. SHAP (SHapley Additive exPlanations) is a unified approach to explain the output of any machine learning model. SHAP connects game theory with local explanations, uniting several previous methods and representing the only possible consistent and locally accurate additive feature attribution method based on expectations (see the SHAP NIPS paper for details).

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 work discussing your final end-to-end problem solution as this reads quite well. I can definitely tell that you have spent a long time on this project as it really shows.

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.

> "Regarding the model itself, applying batch normalization should be able to speed learning and provide a sort of regularization effect, helping to prevent overfitting"

Yes, batch normalization would be a good idea here, since you are overfitting here. Maybe also some more dropout could be included.

(https://www.quora.com/In-a-deep-neural-network-why-does-batch-normalization-help-improve-accuracy-on-a-test-set)

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.

Your writing is very clean and it is very easy to understand what you are saying. I personally thank you as this report is very easy to read :)

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

Code looks great. Nicely presented. You might also check out

- this post regarding Docstrings vs Comments.
- Google Style Python Docstrings
- This Best of the Best Practices" (BOBP) guide to developing in Python

DOWNLOAD PROJECT

RETURN TO PATH