

Florida International University
Knight Foundation School of Computing and Information Sciences

Software Engineering Focus

Final Deliverable

Project Title: Hyperparameter Tuning for Deep Learning Neural
Networks

Team Members: Martin Luis Macho, Jacob Roman, Jose Lopez

Product Owner(s): Yanzhao Wu

Instructor: Masoud Sadjadi

The MIT License (MIT)
Copyright (c) 2016 Florida International University

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Abstract

This document presents the information necessary to gain a good understanding of an application developed to automatically tune the batch size used to train a convolutional neural network model to optimize the speed and accuracy of training of the model, and analyze how batch size impacts model performance.

Table of Contents

Table of Contents

INTRODUCTION.....	5
CURRENT SYSTEM	5
PURPOSE OF NEW SYSTEM.....	5
USER STORIES.....	6
IMPLEMENTED USER STORIES.....	6
PENDING USER STORIES.....	6
PROJECT PLAN.....	7
HARDWARE AND SOFTWARE RESOURCES	7
SPRINTS PLAN	8
<i>Sprint 1</i>	8
<i>Sprint 2</i>	9
<i>Sprint 3</i>	9
<i>Sprint 4</i>	10
<i>Sprint 5</i>	11
SYSTEM DESIGN	12
ARCHITECTURAL PATTERNS.....	12
SYSTEM AND SUBSYSTEM DECOMPOSITION	12
DEPLOYMENT DIAGRAM	12
DESIGN PATTERNS	13
SYSTEM VALIDATION.....	13
GLOSSARY	14
APPENDIX	14
APPENDIX A - UML DIAGRAMS	14
APPENDIX B - USER INTERFACE DESIGN.....	15
APPENDIX C – SPRINT REVIEW REPORTS	16
APPENDIX D – USER MANUALS, INSTALLATION/MAINTENANCE DOCUMENT, SHORTCOMINGS/WISHLIST DOCUMENT AND OTHER DOCUMENTS.....	19
<i>Shortcomings</i>	19
REFERENCES.....	19
REFERENCES.....	19

INTRODUCTION

Machine Learning is a current popular subject in the field of computing, thanks to advancements in computational resources and power, as well as the ease of access with libraries such as TensorFlow and Keras. With greater ease of access, more people are able to train incredibly complex models, such as neural networks, very quickly with little experience. However, due to this lack of experience, many of the hyperparameters used to tune machine learning models can be difficult to understand.

Current System

The current system is LRBench, a program that automatically tunes the learning rate function of a trained model to optimize training accuracy. This system was developed to research how to choose the best learning rate function for training a model. (Wu, 2023)

Purpose of New System

The new system is intended to do a similar process as LRBench for batch size, another Hyperparameter that can greatly impact accuracy and speed of training, but can be difficult to optimize manually without a fine and detailed understanding of how the value impacts training.

USER STORIES

The following section provides the detailed user stories that were implemented in this iteration of the Hyperparameter Tuning for Deep Learning Neural Networks project. These user stories served as the basis for the implementation of the project's features. This section also shows the user stories that are to be considered for future development.

Implemented User Stories

- User Story 1: Background Study and Research
- User Story 2: Plan based on Background Research
- User Story 3: Study existing project (LRBench)
- User Story 4: Write project proposal
- User Story 5: Develop backend for product
- User Story 6: Study and experiment with LRBench project to guide backend development
- User Story 7: Develop and test rudimentary algorithm for batch size tuning
- User Story 8: Continue working on backend for product
- User Story 9: Design and develop frontend UI/functionality for product
- User Story 10: Finalize product prototype
- User Story 11: Test product prototype
- User Story 12: Complete documentation on project

Pending User Stories

- User Story 13: Optimize backend algorithm
- User Story 14: Implement other datasets
- User Story 15: Dynamically adjust batch size during training

PROJECT PLAN

This section describes the planning that went into the realization of this project. This project incorporated the agile development techniques and as such required the sprints to be planned. These sprint plannings are detailed in the section. This section also describes the components, both software and hardware, chosen for this project.

Hardware and Software Resources

The following is a list of all hardware and software resources that were used in this project:

- **Python:** A procedural, general-purpose programming language used for developing the backend architecture used to implement the tuning algorithm.
- **C#:** A general-purpose scripting language used to script the frontend UI for the application.
- **Windows Forms:** A graphics class library included in Microsoft's .NET framework allowing for simplified development of Windows client applications, used to ease the development of the UI for the application.
- **Visual Studio 2022:** An IDE developed by Microsoft for the purpose of .NET development, used to build the application's UI and connect it to the Python backend.
- **Visual Studio Code:** A more general-purpose IDE similar to Visual Studio used to develop the Python backend for the application.
- **Keras:** An open-source library acting as a Python interface for the development of neural network models, built off of the TensorFlow library by Google.

Sprints Plan

Sprint 1

Sprint 1 Planning Meeting Minutes:

Attendees: Yanzhao Wu (Product Owner), Martin Macho (Main Lead), Jacob Roman, Abel Aguilera

Start time: 6:35 PM

End time: 6:55 PM

After discussion, the velocity of the team was estimated to be 20 story points.

The product owner chose the following user stories to be done during the next sprint. They are ordered based on their priority.

- User Story 1: Background Study/Research (5 story points)
- User Story 2: Plan based on Background Study (2 story points)
- User Story 3: Study existing project (LRBench) (3 story points)
- User Story 4: Write project proposal (10 story points)

The team members indicated their willingness to work on the following user stories.

- Martin Macho
 - User Story 1: Background Study/Research
 - User Story 2: Plan based on Background Story
- Jacob Roman
 - User Story 3: Study existing project
 - User Story 1: Background Study/Research
- Abel Aguilera
 - User Story 4: Write project proposal

Sprint 2
Sprint 2 Planning Meeting Minutes:

Attendees: Yanzhao Wu (product owner), Martin Luis Macho (main lead), Jacob Roman, Jose Lopez

Start time: 5:00 PM

End time: 5:30 PM

After discussion, the velocity of the team was estimated to be 20 story points.

The product owner chose the following user stories to be done during the next sprint. They are ordered based on their priority.

- User Story 1: Build model to be tuned on hyperparameter chosen (5 points)
- User Story 2: Tune model based on hyperparameter chosen (10 points)
- User Story 3: Test and evaluate model before and after hyperparameter tuning (5 points)

The team members indicated their willingness to work on the following user stories.

- Martin Luis Macho
 - User Story 2: Tune model based on hyperparameter chosen
- Jacob Roman
 - User Story 1: Build model to be tuned on hyperparameter chosen
- Jose Lopez
 - User Story 3: Test and evaluate model before and after hyperparameter tuning

Sprint 3
Sprint 3 Planning Meeting Minutes:

Attendees: Yanzhao Wu (product owner), Martin Luis Macho (main lead), Jacob Roman, Jose Lopez

Start time: 5:00 PM

End time: 5:30 PM

After discussion, the velocity of the team was estimated to be 20 story points.

The product owner chose the following user stories to be done during the next sprint. They are ordered based on their priority.

- User Story 1: Develop and test rudimentary algorithm for batch size tuning (10 points)
- User Story 2: Convert algorithm to backend for product similar to LRBench (10 points)

The team members indicated their willingness to work on the following user stories.

- Martin Luis Macho
 - User Story 1: Develop and test rudimentary algorithm for batch size tuning
 - User Story 2: Convert algorithm to backend for product similar to LRBench
- Jacob Roman
 - User Story 2: Convert algorithm to backend for product similar to LRBench
- Jose Lopez
 - User Story 1: Develop and test rudimentary algorithm for batch size tuning

Sprint 4 **Sprint 4 Planning Meeting Minutes:**

Attendees: Yanzhao Wu (product owner), Martin Luis Macho (main lead), Jacob Roman, Jose Lopez

Start time: 5:00 PM

End time: 5:30 PM

After discussion, the velocity of the team was estimated to be 20 story points.

The product owner chose the following user stories to be done during the next sprint. They are ordered based on their priority.

- User Story 1: Continue working on backend for product (10 points)
- User Story 2: Design and develop frontend UI/functionality for product (10 points)

The team members indicated their willingness to work on the following user stories.

- Martin Luis Macho
 - User Story 1: Continue working on backend for product
- Jacob Roman
 - User Story 2: Design and develop frontend UI/functionality for product

- Jose Lopez
 - User Story 1: Continue working on backend for product

Sprint 5
Sprint 5 Planning Meeting Minutes:

Attendees: Yanzhao Wu (product owner), Martin Luis Macho (main lead), Jacob Roman, Jose Lopez

Start time: 5:00 PM

End time: 5:30 PM

After discussion, the velocity of the team was estimated to be 20 story points.

The product owner chose the following user stories to be done during the next sprint. They are ordered based on their priority.

- User Story 1: Finalize product prototype (10 points)
- User Story 2: Test product prototype (5 points)
- User Story 3: Complete documentation on project (5 points)

The team members indicated their willingness to work on the following user stories.

- Martin Luis Macho
 - User Story 1: Finalize product prototype
 - User Story 3: Complete documentation on project
- Jacob Roman
 - User Story 1: Finalize product prototype
 - User Story 2: Test product prototype
- Jose Lopez
 - User Story 2: Test product prototype
 - User Story 3: Complete documentation on project

SYSTEM DESIGN

This section contains information on the design decisions that went into this project. The architecture patterns are outlined and explained. The entire system is shown in a package diagram and the subsystems are explained. Finally, the design patterns used in the project are discussed.

Architectural Patterns

Barring UI code automatically generated by Windows Forms, the application was built mostly from scratch, with some code inspired or lifted from Professor Wu's LRBench program. The application is a full-stack application, with a Python backend and a C# Windows Forms frontend. The backend is entirely built and compiled in Python; it comes in the form of a single Python file with a defined function, and can be run from the command line. The Windows Form frontend uses Windows Sockets to connect to this backend, allowing it to send input and receive output from the Python function. Since the backend and frontend are independent from each other, despite the frontend being Windows specific, the backend can easily be ported to other operating systems.

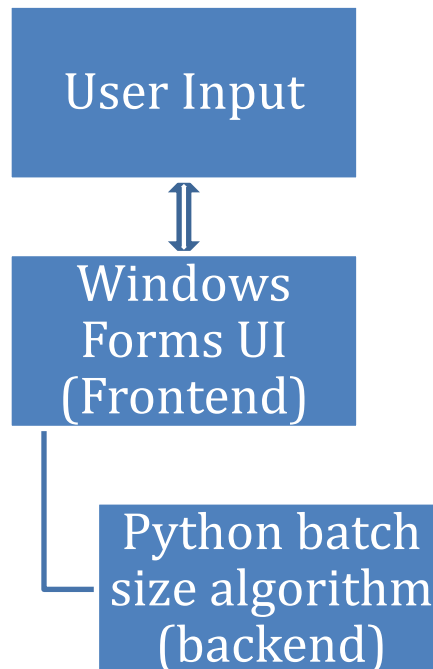
System and Subsystem Decomposition

As of now, the application is in the early prototype stages. As such, the system is rudimentary and not particularly complex; it simply takes input of how many epochs the user wants to train a neural network model for and whether they would like to prioritize speed or accuracy of training, chooses a batch size based on these choices, trains a model, and outputs the results to the user. The subsystems are the Windows Form frontend that enables user interaction and input, and the Python backend that contains the algorithm used to determine batch size and all the necessary functions and definitions needed to train a Keras convolutional neural network model.

Deployment Diagram

Due to the early nature of the product, we have no intentions of deployment or distribution at this time. Deployment is to be considered in future after considerable iterations and further development.

Design Patterns



SYSTEM VALIDATION

Testing was done locally on Visual Studio 2022. Testing focused on two aspects: functionality of the frontend, and performance of the backend. Backend performance was first tested by running the Python function independently in Visual Studio Code; when the frontend and backend were connected, testing was carried out through building and running the application through Visual Studio 2022. No major errors or bugs were found during testing.

GLOSSARY

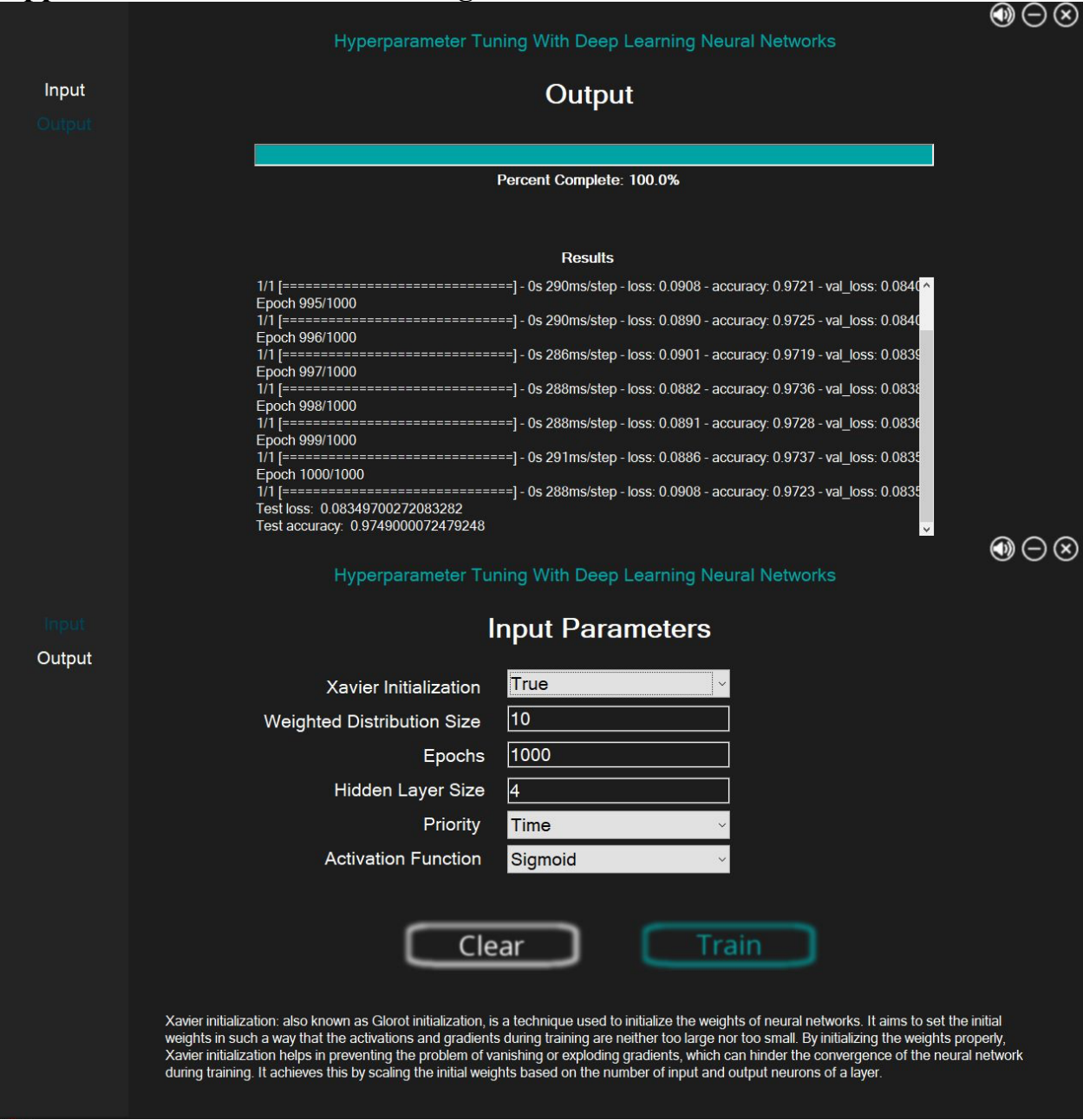
- **Neural Network:** A form of machine learning model based on tuning parameters to mimic the neural connections of a human brain.
- **Machine Learning:** A field of computer science focused on having machines process the optimal solution for a given problem by computing the algorithms itself, rather than using an algorithm developed by a human.
- **Keras:** An open-source library used to interface Python with the popular TensorFlow machine learning library by Google.

APPENDIX

Appendix A - UML Diagrams

Since the application was developed using Windows Forms and .NET Core, as well as a Python file with only one defined function, no UML diagram was created.

Appendix B - User Interface Design



Appendix C – Sprint Review Reports

Sprint 1 Review Meeting Minutes

Attendees: Yanzhao Wu (product owner), Martin Luis Macho (main lead), Jacob Roman, Jose Lopez

Start time: 9:00 PM

End time: 9:30 PM

After a show and tell presentation, the implementation of the following user stories were accepted by the product owners: All.

- User Story 1: Background Study and Research
- User Story 2: Plan based on Background Research
- User Story 3: Study existing project (LRBench)
- User Story 4: Write project proposal

The following ones were rejected and moved back to the product backlog to be assigned to a future sprint at a future Spring Planning meeting.

- None

Sprint 2 Review Meeting Minutes

Attendees: Yanzhao Wu (product owner), Martin Luis Macho (main lead), Jacob Roman, Jose Lopez

Start time: 9:00 PM

End time: 9:30 PM

After a show and tell presentation, the implementation of the following user stories were accepted by the product owners: All.

- User Story 4: Develop backend for product
- User Story 5: Study and experiment with LRBench project to guide backend development

The following ones were rejected and moved back to the product backlog to be assigned to a future sprint at a future Spring Planning meeting.

- User Story 1: Build model to be tuned on hyperparameter chosen
- User Story 2: Tune model based on hyperparameter chosen
- User Story 3: Test and evaluate model before and after hyperparameter tuning

Sprint 3 Review Meeting Minutes

Attendees: Yanzhao Wu (product owner), Martin Luis Macho (main lead), Jacob Roman, Jose Lopez

Start time: 8:30 PM

End time: 9:00 PM

After a show and tell presentation, the implementation of the following user stories were accepted by the product owners:

- User Story 1: Develop and test rudimentary algorithm for batch size tuning

The following ones were rejected and moved back to the product backlog to be assigned to a future sprint at a future Spring Planning meeting.

- User Story 2: Convert algorithm to backend for product similar to LRBench

Sprint 4 Review Meeting Minutes

Attendees: Yanzhao Wu (product owner), Martin Luis Macho (main lead), Jacob Roman, Jose Lopez

Start time: 8:30 PM

End time: 9:00 PM

After a show and tell presentation, the implementation of the following user stories were accepted by the product owners:

- User Story 1: Continue working on backend for product
- User Story 2: Design and develop frontend UI/functionality for product

The following ones were rejected and moved back to the product backlog to be assigned to a future sprint at a future Spring Planning meeting.

- None

Sprint 5 Review Meeting Minutes

Attendees: Yanzhao Wu (Product Owner), Martin Luis Macho (Main Team Lead), Jacob Roman, Jose Lopez

Start time: 10:30

End time: 10:40

After a show and tell presentation, the implementation of the following user stories were accepted by the product owners: All.

- User Story 1: Finalize product prototype (10 points)
- User Story 2: Test product prototype (5 points)
- User Story 3: Complete documentation on project (5 points)

The following ones were rejected and moved back to the product backlog to be assigned to a future sprint at a future Spring Planning meeting.

- None

Appendix D – User Manuals, Installation/Maintenance Document, Shortcomings/Wishlist Document and other documents

Shortcomings

Due to the new nature of the application, there are many aspects of the application that fell short that would should be focused on in future iterations.

- The system as currently designed is not very portable; future iterations would move away from Windows Forms for ease of installation on different operating systems.
- The system currently only operated on one neural network model and one dataset; future iterations should focus on expanding the system's capabilities to multiple datasets.
- The algorithm used to tune the batch size is still rudimentary; it only tunes the batch size once at the start of training, and uses a simple algorithm based on basic experimentation. Future iterations would ideally update batch size dynamically based on training performance on multiple datasets and models.

REFERENCES

REFERENCES

Wu, Y. (2023, February 16). Selecting and Composing Learning Rate Policies for Deep Neural Networks. *ACM Transactions on Intelligent Systems and Technology*, 14(2), 1-25.