<Neural Compression Network & Version 0.1>

Design Documentation

* Paul Gildehaus <[peg3268@rit.edu](mailto:peg3268@rit.edu)>

Summary & Product Description 2

Domain Model 3

System Architecture 4

Subsystems 5

Name of the subsystem 5

Name of the subsystem 5

Status of the Implementation 6

Appendix 7

# Summary

The Neural Compression Network is a project that I am starting to teach myself about multiple areas of programming theory and new technologies. I originally set out with the goal to learn C# in depth with the project idea of making a manual compression program. I knew and still know I am not going to come up with anything that even comes close to the industry standard compression rate, however I wanted to do it for fun. Next, I wanted to learn more about how neural networks and machine learning works, which I can also do in C# comfortably. Then my roommate told me to combine the two projects; work on each of them and then implement them together with the end goal of creating a neural network that tries to make a compression algorithm more efficient.

This project has more than enough potential to be too difficult for my level of programming experience, however the worst-case scenario that I can foresee is that I create a manual compression algorithm and a basic neural network. Even if this is what ends up happening I will still be happy with this level of experience.

# Domain Model

# 

**Figure 1: High-level domain model of the entire project**

# System Architecture

# 

**Figure 2: System architecture of the project**

# Subsystems

This section provides detailed design for specific subsystems described in the system architecture.

## Name of the subsystem

In this section, provide the following information for the first subsystem.

* Class structure diagram and a narrative that describes the structure of this subsystem
* Sequence diagrams with associated narratives that describe the dynamic behaviors that are primarily located within this subsystem. Within your subsystem design descriptions, you must make sure to provide sequence diagrams for all features listed in the design project problem statement. You may also decide that other features require documentation within the subsystems.
* A description of all design patterns that are primarily located within this subsystem. Use the table below to describe each design pattern. If a design pattern cuts across the boundary of subsystems, place the pattern usage table in the section for the subsystem that holds the majority of pattern participants.

|  |  |  |  |
| --- | --- | --- | --- |
| **Name:** | | | **GoF pattern:** |
| **Participants** | | | |
| **Class** | **Role in GoF pattern** | **Participant's contribution in the context of the application** | |
|  |  |  | |
|  |  |  | |
|  |  |  | |
| **Deviations from the standard pattern:** | | | |
| **Requirements being covered:** | | | |

## Name of the subsystem

This section provides a similar description and diagrams for the model describing the second subsystem.

# Status of the Implementation

Provide a complete description of the status of your implementation. This should specify all known defects in the system, and indicate requirements that your implementation does not cover.

# Appendix

This section provides fine-grained design details for all of the classes in your design. You will capture this information using the CRC (Class-Responsibilities-Collaborators) card format below.

|  |  |
| --- | --- |
| **Class:** MyClass1 |  |
| **Responsibilities:** ... |  |
| **Collaborators:** ... |  |
| **Users:** ... | **Used by:** ... |
| **Author:** ... |  |