Module Interface Specification for ANN (Artificial Neural Newcounter)

Tanya Djavehrpour

March 18, 2024

1 Revision History

Date	Version	Notes
March 18	1.0	Initial Draft

2 Symbols, Abbreviations and Acronyms

See SRS Documentation Djavaherpour (2024b) at HERE.

Contents

3 Introduction

The following document details the Module Interface Specifications for ANN (Artificial Neural Network). This document specifies how every module is interfacing with every other parts.

Complementary documents include the System Requirement Specifications (SRS) Djavaherpour (2024b) and Module Guide (MG) Djavaherpour (2024a). The full documentation and implementation can be found at Github repository for ANN.

4 Notation

The structure of the MIS for modules comes from Hoffman and Strooper (1995), with the addition that template modules have been adapted from Ghezzi et al. (2003). The mathematical notation comes from Chapter 3 of Hoffman and Strooper (1995). For instance, the symbol := is used for a multiple assignment statement and conditional rules follow the form $(c_1 \Rightarrow r_1|c_2 \Rightarrow r_2|...|c_n \Rightarrow r_n)$.

The following table summarizes the primitive data types used by ANN.

Data Type	Notation	Description
character	char	a single symbol or digit
integer	$\mathbb{Z}$	a number without a fractional component in (-∞, ∞)
natural number	$\mathbb{N}$	a number without a fractional component in [1, ∞)
real	$\mathbb{R}$	any number in (-∞, ∞)

The specification of ProgName uses some derived data types: sequences, strings, and tuples. Sequences are lists filled with elements of the same data type. Strings are sequences of characters. Tuples contain a list of values, potentially of different types. In addition, ProgName uses functions, which are defined by the data types of their inputs and outputs. Local functions are described by giving their type signature followed by their specification.

5 Module Decomposition

The following table is taken directly from the Module Guide Djavaherpour (2024a) document for this project.

Level 1	**Level 2**
Hardware-Hiding	
Behaviour-Hiding	ANN Control Module Saved ANN Model Module Output Module Input Classifier Module Input Image Module Training Model Module Feedback Module
Software Decision	Input Preparing and Preprocessing Module Data Preparing and Preprocessing Module Training Module Testing Module

Table 1: Module Hierarchy

6 MIS of ANN Control Module

6.1 Module

`main`

6.2 Uses

- Hardware-Hiding Module
- Saved ANN Model Module (7)
- Output Module (**??**)

6.3 Syntax

6.3.1 Exported Constants

None.

6.3.2 Exported Access Programs

Name	In	Out	Exceptions
`main`	-	-	-

6.4 Semantics

6.4.1 State Variables

None.

6.4.2 Environment Variables

None.

6.4.3 Assumptions

- The ANN Control Module assumes that the Hardware-Hiding Module, Saved ANN Model Module, and Output Module are implemented according to their specifications. However, it does include error handling to manage unexpected behaviors or failures in these modules.
- The system environment (operating system, hardware) is assumed to be stable. Also, essential libraries and dependencies are presumed to be correctly installed and configured.

6.4.4 Access Routine Semantics

main():

- transition: Initializes the program.

Note: As the ANN Control Module mainly serves as a coordinator between different modules without maintaining its own state or producing output, its primary function is to ensure the correct sequence of operations and interactions between these modules. It relies on the robustness of the called modules' error handling.

6.4.5 Local Functions

None.

7 MIS of Saved ANN Model Module

7.1 Module

`model`

7.2 Uses

- Hardware-Hiding Module
- Training Module (**??**)

7.3 Syntax

7.3.1 Exported Constants

None.

7.3.2 Exported Access Programs

Name	In	Out	Exceptions
`save_model`	-	-	PermissionError
`load_model`	-	[weights, biases]	FileNotFoundError

7.4 Semantics

7.4.1 State Variables

modelData: structure holding the current ANN model's data. This is an array including weights and biases.

7.4.2 Environment Variables

modelFile: A file on the file system where the ANN model data is saved and from where it is loaded.

7.4.3 Assumptions

None.

7.4.4 Access Routine Semantics

`save_model`():

- transition: Writes the current state of the ANN model (weights and biases) to the `modelFile`.
- output: None.
- exception: Raises `PermissionError` if the module lacks the necessary permissions to write to `modelFile`. It may also raise an IOError if there are issues with the file system, such as insufficient storage space.

`load_model`():

- transition: Reads the model data from `modelFile`.
- output: Returns the trained model based on weights and biases from `modelData`.
- exception: Raises `FileNotFoundError` if `modelFile` does not exist or cannot be accessed. Additionally, an exception may be raised for data corruption or format mismatch, indicating issues with the integrity or compatibility of the stored model data.

7.4.5 Local Functions

None.

References

Tanya Djavaherpour. Module guide. https://github.com/tanya-jp/ANN-CAS741/blob/main/docs/Design/SoftArchitecture/MG.pdf, 2024a.

Tanya Djavaherpour. System requirements specification. https://github.com/tanya-jp/ANN-CAS741/blob/main/docs/SRS/SRS.pdf, 2024b.

Carlo Ghezzi, Mehdi Jazayeri, and Dino Mandrioli. *Fundamentals of Software Engineering*. Prentice Hall, Upper Saddle River, NJ, USA, 2nd edition, 2003.

Daniel M. Hoffman and Paul A. Strooper. *Software Design, Automated Testing, and Maintenance: A Practical Approach*. International Thomson Computer Press, New York, NY, USA, 1995. URL http://citeseer.ist.psu.edu/428727.html.

8 Appendix

[Extra information if required —SS]

9 Reflection

The information in this section will be used to evaluate the team members on the graduate attribute of Problem Analysis and Design. Please answer the following questions:

1. What are the limitations of your solution? Put another way, given unlimited resources, what could you do to make the project better? (LO_ProbSolutions)

2. Give a brief overview of other design solutions you considered. What are the benefits and tradeoffs of those other designs compared with the chosen design? From all the potential options, why did you select the documented design? (LO_Explores)