System Requirements Specification

EasyNN/EasyAI

CS 491, Spring 2021

Team Name: EasyNN/EasyAI

Team Members:

- Jack Nguyen
- David Fadini
- Liam Kehoe
- Nathan Rose
- Nathan Foster

Contents of this Document

Section 1: Introduction

System to be Produced:

- EasyAI: This product is designed to allow users to create or log in to an account, allowing them access to the website and letting users access frameworks in the website. These frameworks will include the EasyNN framework and the EasyGA framework. The users will also be able to adjust the framework's attributes from a form on the website. On the website, users will be able to save their work to a database allowing access to their data at a later time.

- EasyNN: The goal of this product is to create a neural network framework that is easy to use and customize. This will be done in python and heavily leveraging default values for the user to start training immediately and then dive into various features as they need to.

Applicable Standards

- <You do not have to repeat the standards included in the project plan. Instead, cite any standards that are specific to the system requirements.>

Definitions, Acronyms, and Abbreviations

- <Include any that are needed to read this document or "none" if the document is self-explanatory and no acronyms or abbreviations will be used

- A model is a representation of another system.

- A(n) (Artificial) Neural Network (NN or ANN) is a simplified computational model of the human brain, loosely based on the idea of passing information between several nodes to get an output.

- A loss function is a numerical measurement of the performance of a model.

- A topology is a structure defined by an arrangement of nodes and their connections.

- Optimization is the process of either minimizing or maximizing a loss function.

- Neuroevolutionary methods are a class of general Neural Network optimization techniques for optimizing topologies based on Genetic Algorithms.

- Gradient descent methods are a class of numerical optimization techniques based on partial derivatives, including normal gradient descent, momentum-based gradient descent, SGD, ADAM, ADAGRAD, ADADELTA, and RMSPROP.

- Forward Propagation is the process of processing values through a Neural Network, starting from the input nodes and moving to the output nodes.

- Backward Propagation (BackPropagation) is the process of computing components of the gradient iteratively instead of entirely at once, starting from the output nodes and moving to the input nodes.

Section 2: Product Overview

Assumptions:

- <List all the assumptions the developers are making. For example assumptions about other systems this product will interface with; assumptions about the technological environment in which the product will operate (how much memory, what type of processor, ...); assumptions about availability and capability of COTS, GOTS, or other re-used products, ...>
- EasyGA:
  - The system shall attain better fitness values over time.
- EasyAI:
  - The system can accept new and returning users
  - The system can store users data in a database
  - The system is written in python using the Django framework

Stakeholders:

- <A stakeholder is anyone who has an interest in the system to be developed. For example, the customer, the various classes of users, applicable regulatory agencies, ... List each category of stakeholder and give a phrase or a sentence to describe their interest or concerns>
- EasyAI:
  - Users: This person wants a fully developed system to allow them to run projects through the neural network frameworks
  - Daniel, Product Owner: Wants the website to run the EasyNN and EasyGA and provide an interface for users to run their problems
- EasyNN:
  - User: The person who wants to be able to set up and customize the neural network to solve their problem

- Daniel, Product Owner: Wants the system developed to build on top of his EASYGA system.

Event Table:

Easy NN

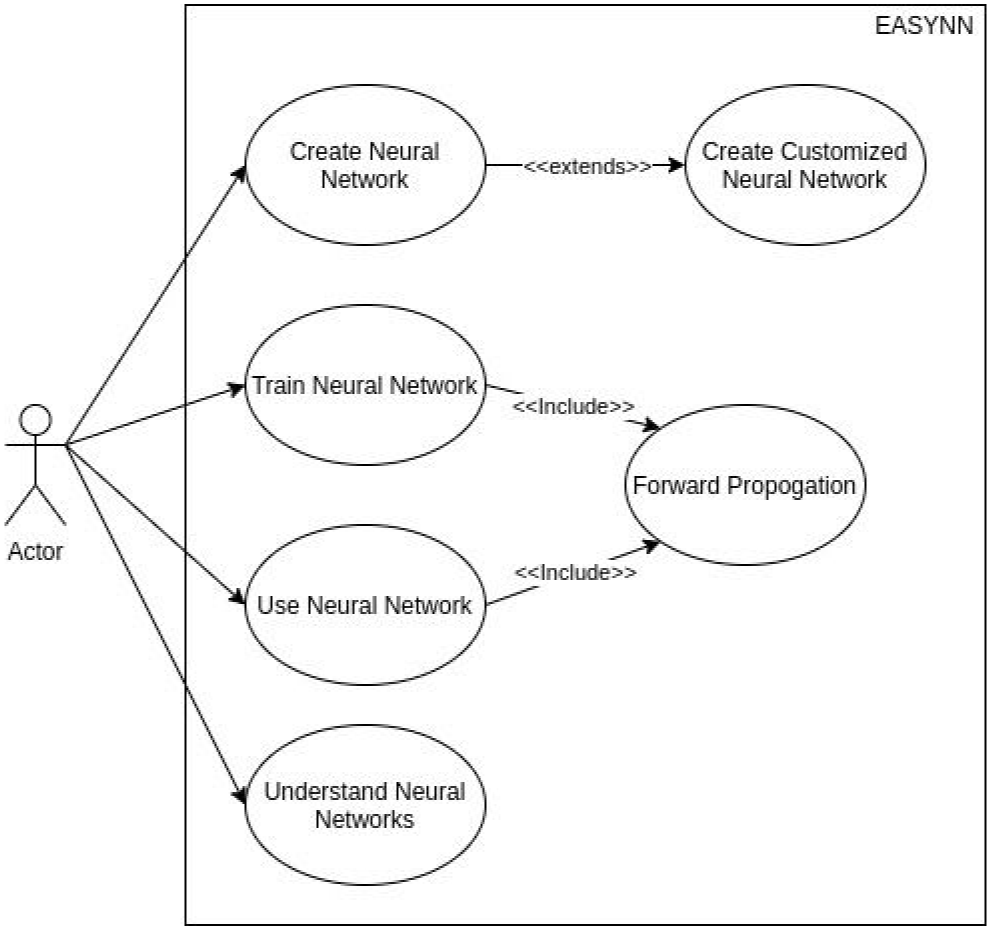
Event Name	External Stimuli	External Responses	Internal data and state
Creation	Constructor is called	A neural network is created	by default it is set up for good training of [[Whatever the database is called]]
Forward Propagation	Data is given to NN to process	The confidence interval is given	The values for each layer are determined(and stored)
Backward Propagation	The correct answer is given to NN	loss and accuracy are given for the last input/batch	All weights and biases are updated to improve the accuracy of the model

Easy AI

Event Name	External Stimuli	External Responses	Internal data and state
Create User	A database is filed	A user account is created	The values of the user ID, name, password [other credentials are changed]
Log In	A function with parameters are passed	The user profile is loaded into the system	The profile will load the user's tables of different projects
Algorithm Configuration	A form is filled with configuration parameters	The algorithm loads the selected parameters	the default will change parameters and become what the user wants

EasyNN

Use Case Diagram



Use Case Descriptions:

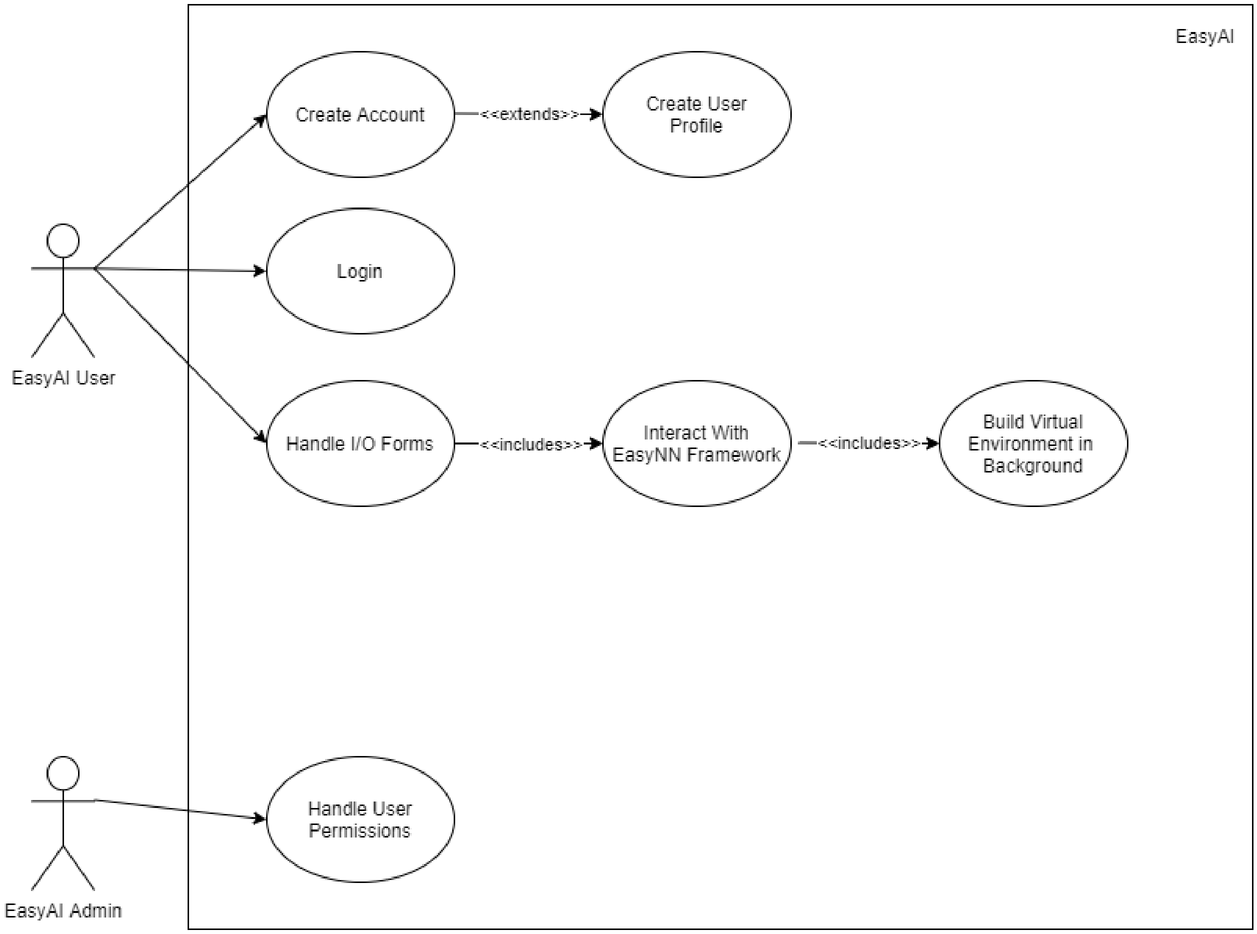
1. Create Neural Network
    a. The user created a basic Neural Network that will work with an MNIST dataset
2. Train Neural Network
    a. The user can give the Neural Network data so that it can train itself. Uses Forward Propagation. A correct answer is also given so the system can adjust itself
3. Use Neural Network
    a. The User can just use the Neural Network to try to determine the correct answer of their dataset without feeding in the correct answer for training
4. Understand Neural Network
    a. The user will want to understand not only how Neural Networks work, but how to customize this Neural Network so they can tweak it to their specifications
5. Forward Propagation

   a. The part of the Neural Network that determines what the output is given a set of inputs

6. Create Customized Neural Network
   a. The use can stray from the default values so that they can use it on their dataset.

EasyAI

Use Case Diagram



Use Case Descriptions:

1. Create Account
   a. The user can register an account with the website
2. Create User Profile
   a. A user profile is made for the user after they register an account
3. Login
   a. A user who has already registered for an account can log in to their account
4. Handle I/O Forms
   a. The EasyAI application can handle input and output forms that the user will fill out
5. Interact With EasyNN Framework
   a. The EasyAI application can interact with the EasyNN framework

6. Build Virtual Environment in Background
    a. The EasyAI application can create a virtual environment in the background so that users can input code without corrupting the site
7. Handle User Permissions
    a. An admin can log in to the admin section and handle user permissions

Section 3: Specific Requirements

<Use the following template for each requirement. >

No: <unique requirement number>
Statement: <the "shall" statement of the requirement>
Source: <source of the requirement>
Dependency: <list each other requirements on which satisfaction of this requirement depends. (Maybe "None")>
Conflicts: <list each other requirements with which this requirement conflicts. (Maybe "None")>
Supporting Materials: <list any supporting diagrams, lists, memos, etc.>
Evaluation Method: <How can you tell if the completed system satisfies this requirement? >
Revision History: <who, when, what>

3.1 Functional Requirements

- < Describe the fundamental actions that the system must perform. Functional requirements can be partitioned into subfunctions or subprocesses. Note: the System design partition does not have to correspond with the functional requirements partition. Functional requirements include:
    - validity checks on the inputs,
    - the exact sequence of operations,
    - responses to abnormal situations
    - relationship of outputs to inputs
        - input/output sequences, formulas for input to output conversion, etc.
    - ...>

EasyNN

- The EasyNN system shall return a float vector when given a float vector as input.

- The EasyNN system shall reduce the loss function over time until it hits a locally optimal Neural Network when given data to train on.

- The EasyNN system shall train numerical components of a Neural Network model using gradient descent methods.

- The EasyNN system shall produce a Neural Network model matching a given topology.

- The EasyNN system shall be able to save Neural Network models.

- The EasyNN system shall have built-in general, dense, recurrent, and markov chain types of Neural Network models.

- The EasyNN system shall have the option to improve Neural Network topologies using neuroevolution.

- The EasyNN system shall have built-in gradient descent methods including the ones listed in Section 1 Definitions, Gradient Descent.

- The EasyNN system shall use BackPropagation to perform gradient descent methods.

- The EasyNN system shall use analytic differentiation if it is provided with an analytic derivative.

- The EasyNN system shall use numerical differentiation if an analytic derivative is not provided.

EasyAI

- The EasyAI web application shall have a register page to allow users to have their page.
- The EasyAI web application shall have a secure login page.
- The EasyAI web application shall interact with the EasyNN and EasyGA framework.
- The EasyAI web application shall handle input/output forms that the user will fill out or have pre-filled in values.
- The EasyAI web application shall build a virtual environment in the background to allow users to input code without corrupting the site
- The EasyAI web application shall track and interact the user’s virtual environments folder with the database.
- The EasyAI web application shall have a user profile for the user.
- The EasyAI web application shall have an admin section that handles all users and permissions.

3.2 Interface Requirements

- < Describe the interactions of the system with other entities. Interface requirements include a precise description of the protocol for each interface:
  - what data items are input
  - what data items are output

  - what are the data type, the format, and the possible range of values for each data item? (i.e. what is the "domain" of this data item?)
  - how accurate must each data item be?
  - how often will each data item be received or sent?
  - timing issues (synchronous/asynchronous)>
  - how many will be received or sent in a particular period?
  - how accurate must the data be?
  - ...>

EasyNN

- The EasyNN system shall accept 2D arrays as vector inputs by flattening them.

- The EasyNN system shall reshape vector outputs to the 2D arrays as specified by user input.

EasyAI

- The EasyAI system shall run multiple users asynchronously.

- The EasyAI system shall handle the input data types by not allowing incorrect data types at the form input layer of the website.

- The EasyAI system shall have an interface that is easy to understand how to use the system. While also allowing the user to understand how the inputs affect the all overall system.

3.3 Physical Environment Requirements

- < Describe the environment in which the system must operate. Physical environment requirements include:
  - type of equipment/environment on which the system must run
  - location of the equipment
  - environmental considerations: temperature, humidity, ...
  - ...>

- Someone fill in the physical server information.

EasyAI

- EasyAI will have a server component that will run the Django website once a stable version is running but for the most part, all the development and advancement will be run on localhost on the developer computer.

3.4 User and Human Factors Requirements

- <Describe the users and their constraints:
    - What different types of users must the system support?
    - What is the skill level of each type of user? What type of training and documentation must be provided for each user?
    - Do any users require special accommodations (large font size, ...)
    - Must the system detect and prevent misuse? If so, what types of potential misuse must the system detect and prevent?
    - ...>

- The EasyAI system shall support any skill leveled python programmer.

- The EasyNN system shall save the model's data to the database.

3.5 Documentation Requirements

- <Describe what documentation is required:
    - on-line, printed, or both?
    - what is the assumed skill level of the audience of each component of documentation?
    - ...>

- The code shall follow PEP8's formatting standards.

- Classes, attributes, methods, and functions shall be documented using doc-strings and type-hints.

- The systems shall use duck typing instead of enforcing type hints.

- The built-in help() function shall print documentation for classes and functions.

- The GitHub repository shall include documentation in the wiki.

- In-line comments shall be used where determined necessary during code review.

- Source code documentation shall assume the reader is proficient with python.

- GitHub documentation shall assume the reader is proficient with python.

3.6 Data Requirements

- <Describe any data calculations: what formula will be used? to what degree of precision must the calculations be made? >
- <Describe any retained data requirements: exactly what must be retained?
- ...>

- The EasyNN system shall use NumPy's linear algebra operations to evaluate large vector calculations.

- The EasyNN system shall retain data used during calculations.

- The EasyAI system shall have a database to handle saving user inputs and project code to prevent the user from losing it.

- The EasyAI system will store all user

3.7 Resource Requirements

- <Describe the system resources:
    - skilled personnel required to build, use, and maintain the system?
    - physical space, power, heating, air conditioning, ...?
    - schedule?
    - funding?
    - hardware/software/tools?
    - ...>

3.8 Security Requirements

- <Describe any security requirements:
    - must access to the system or information be controlled?
    - must one user's data be isolated from others?
    - how will user programs be isolated from other programs and the operating system?
    - how often will the system be backed up?
    - must the backup copies be stored at a different location?
    - should precautions be taken against fire, water damage, theft, ...?
    - what are the recovery requirements?
    - ...>

- EasyAI shall handle user information. Django has a lot of pre-built handling of user creation frameworks to handle security so that is what we will rely on.

-

3.9 Quality Assurance Requirements

- <Describe quality attributes:
    - What are the requirements for reliability, availability, maintainability, security, portability ...?
    - How must these quality attributes be demonstrated?
    - Must the system detect and isolate faults? If so, what types of faults?
    - Is there a prescribed mean time between failures?
    - Is there a prescribed time the system must be available?
    - Is there a maximum time allowed for restarting the system after a failure?
    - What are the requirements for resource usage and response times?
    - ...>

- Easy AI will have to do error checking for input to prevent incorrect user interaction with the web application.

- EasyAI We will also have to test the frameworks using unit testing to see how the system handles large data tasks.

Section 4: Supporting Material

- <Here is where you put all your analysis work from which you derived the above requirements. It may include UML or other diagrams, notes, memos, etc.)