The Learning Triangle

Use Case Specification: Building Up Neuronal Network

Version <1.0>

Date	Version	Description	Author
06.04.2017	1.0	First set up	LearningTriangleTeam

1. Building Up a Neuronal Network

1.1 Brief Description

In this use-case we define the basic look of our neuronal network. The neuronal network will be capable of making a decision based on information given in a certain step (or time) in the game.

2. Flow of Events

2.1 Look of the neuronal network

A neuronal network basically consists of three parts:

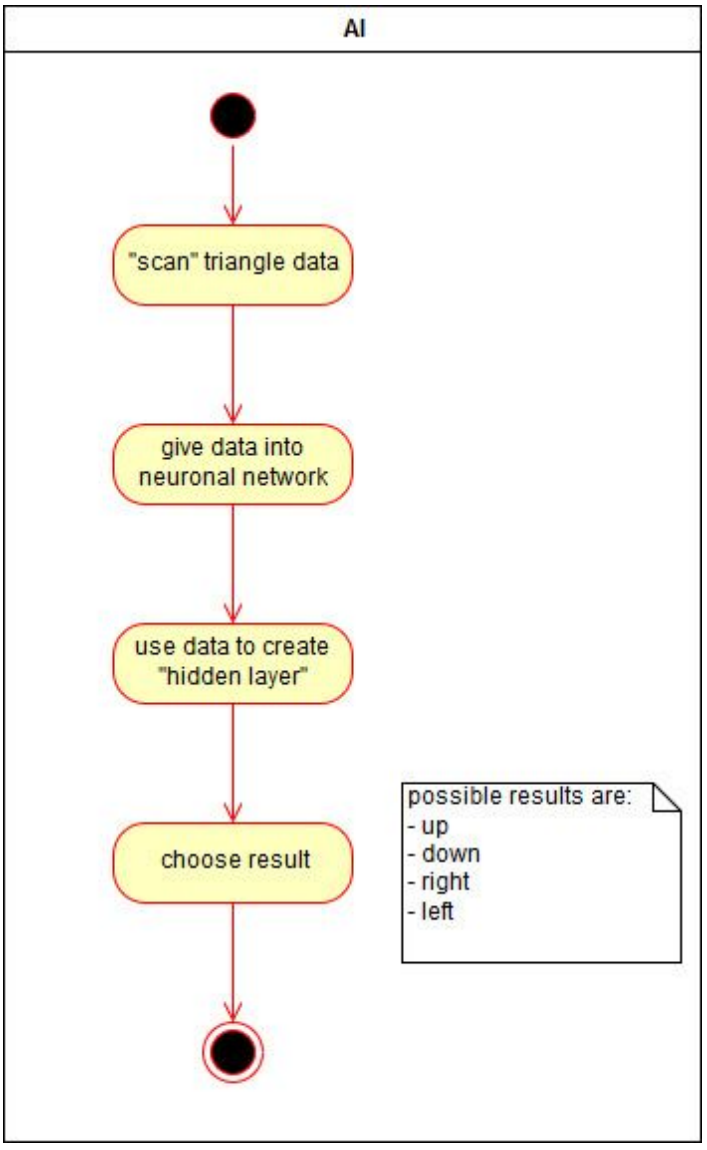
- Input layer
- Hidden layer
- Output layer

The input layer defines what a the learning algorithm sees. Therefore the output layer is the action it takes on certain circumstances. The hidden layer also consists of neurons connected with each other, but are not representing a result for itself. It is like an interconnected web of special features that are activated by the input. Therefore, if you use more neurons and hidden layers you get a much precise answer to your question. In our

case we ask for a direction the triangle should take and provide information of the environment.
In our case we feed the algorithm with the field of view and want to get the direction it should take. Consider variable ***vh*** as view height and ***vw*** as view width (centered on triangle).

So our input layer has ***vh * vw*** input neurons with possible values of the number of different ***field types***. The output layer will have **4** neurons one for each direction the triangle can take. The size of the hidden layer will be tested.
We will provide information about how many hidden layers we have used and how fast our algorithm converges to a certain state (for example: if triangles don't die anymore).
With this we will provide information about how fast a certain combination of hidden layers are influencing the behaviour of learning and how good it can become.

Activity diagram:



The Activity Diagram is this simple because explaining the algorithm and its steps in this diagram would be too much.

Mockup:

There isn’t a mockup necessary for the neuronal network, it's all based on algorithm behind our UI.

Feature File:

3. Special Requirements

4. Preconditions

There must be triangles which can be controlled by the AI. Only then the data from the view is providable. This data must have the format of an one-dimensional matrix.

5. Postconditions

For one specific triangle will be an effect, because it will recieve an result. Then the action will happen, defined in the Use Case *Control Triangle*.

6. Extension Points

n/a