Reinforcement Learning Documentation

#### Discounting factor information: <https://stats.stackexchange.com/questions/221402/understanding-the-role-of-the-discount-factor-in-reinforcement-learning>

#### Information on Terminal State:

<https://towardsdatascience.com/practical-reinforcement-learning-02-getting-started-with-q-learning-582f63e4acd9>

#### Introduction to DQN:

<https://towardsdatascience.com/introduction-to-various-reinforcement-learning-algorithms-i-q-learning-sarsa-dqn-ddpg-72a5e0cb6287>

<https://skymind.ai/wiki/deep-reinforcement-learning>

<http://deeplizard.com/learn/video/xVkPh9E9GfE>

#### DQN used by Atari documentation:

<https://storage.googleapis.com/deepmind-media/dqn/DQNNaturePaper.pdf>

##### Neural Network Setup:

Uses a hard ReLU (rectifier nonlinearity).

A.k.a Rectified Linear Unit

The purpose of using ReLU is to disrupt linearity between the input/outputs. Using relu is the more economical (computationally) to do this (as opposed to using an nonlinear rectifier)

<https://machinelearningmastery.com/rectified-linear-activation-function-for-deep-learning-neural-networks/>

Equation: (output is always either positive x or 0)

Classified as a type of activation function (associated mainly with neural networks)

#### Definition of Episode:

<https://www.quora.com/What-does-the-term-%E2%80%9Cepisode%E2%80%9D-mean-in-the-context-of-reinforcement-learning-RL>

#### Definition of Actions:

Incremental changes of parameters in each m-code.

Increments can be 0 (no action), 0.5, 1, 20 (etc.)

The incremental change is unit-less, applicable to all m-codes. Use the known significant interval changes of all m-codes to build options list.

Either extract m-codes from DWC (not sure how) or manually input initial m-code parameters.

#### Introduction into Neural Networks using Synaptic.js

<https://medium.freecodecamp.org/how-to-create-a-neural-network-in-javascript-in-only-30-lines-of-code-343dafc50d49>