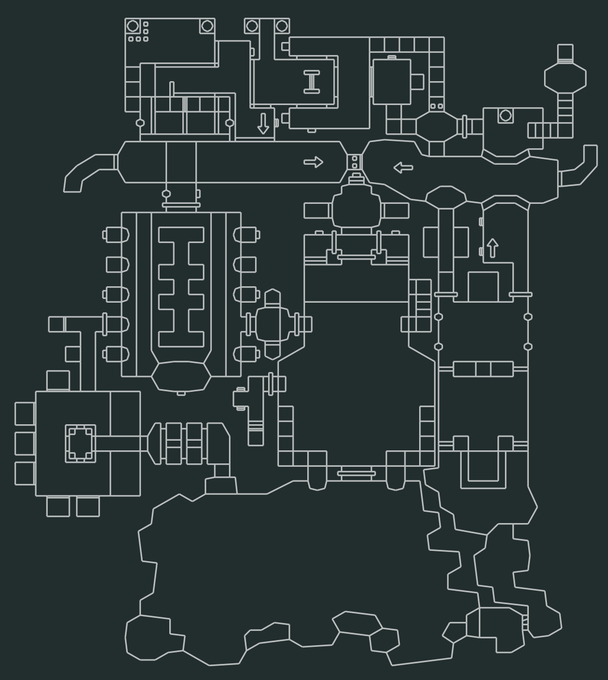
PLAYING WITH NEURAL NETWORKS

Dino Ratcliffe | dratcl@essex.ac.uk | DinoRatcliffe.com

/DinoRatcliffe/we-are-doomed

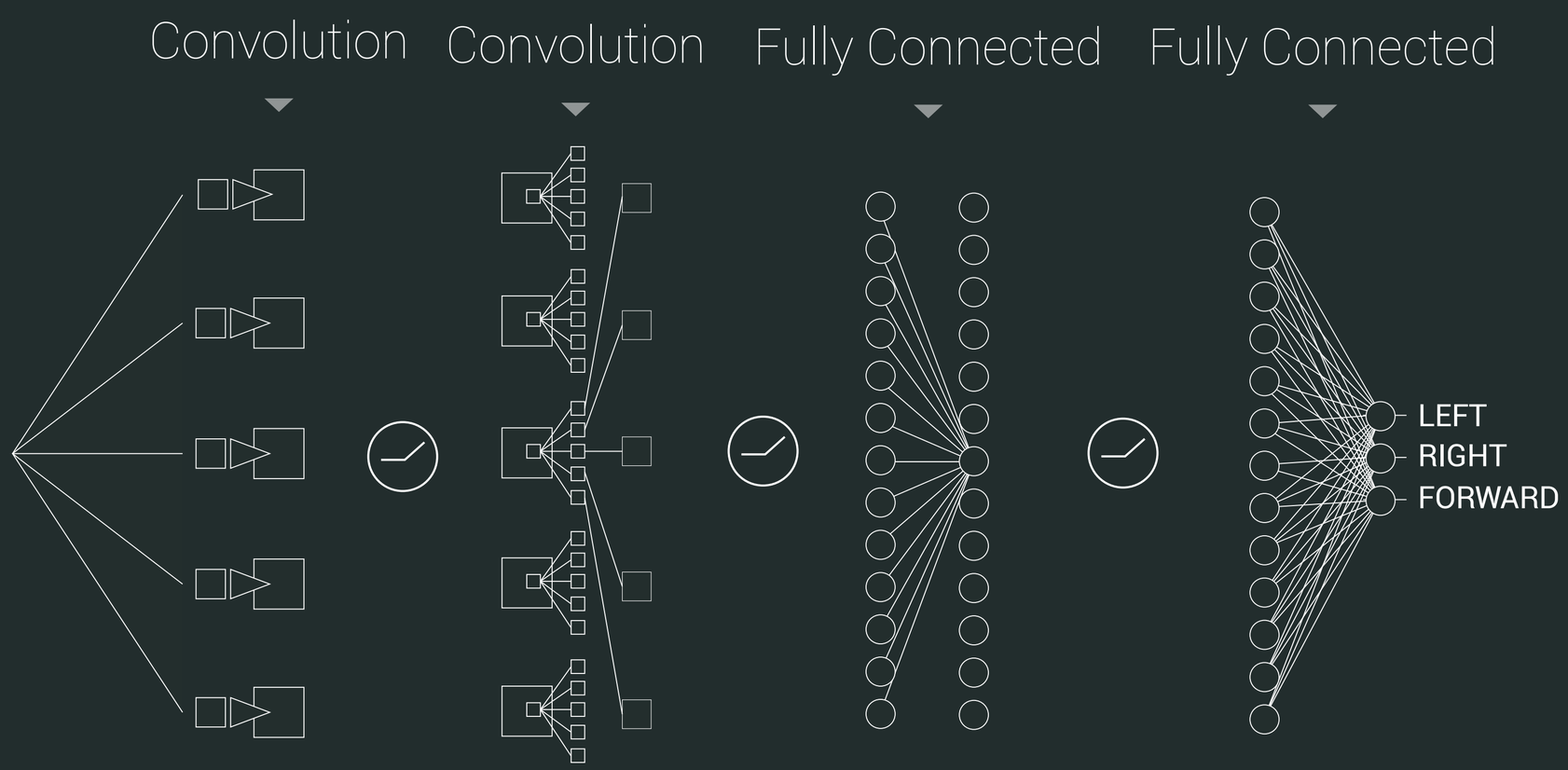
DOOM / ViZDOOM

DOOM is a FPS (First Person Shooter) game that was released in 1993. It released with a multiplayer network experience that allowed people to play against each other in a mode called Deathmatch. In this mode players need to kill each other until a time limit is reached. The player with the most kills, sometimes referred to as "frags" wins. ViZDOOM is a project and competition aiming to pit AI agents against each other in this game mode. This involves having an agent that can navigate the world, collect ammo and health and also acquire and shoot enemy targets.



Deep Q Networks

Deep Q Networks were developed by Google Deepmind and where, at the time, the state of the art in deep reinforcement learning. They use three fundamental techniques to achieve deep reinforcement learning, the first is a convolutional neural network to process the game images, the second is using this network to be a function approximation of Q-Learning and the third is using a working memory that allows it to train on experiences multiple times and also helps make the network more numerically stable.



Input

With this Deep Q Network the only information the network receives is the image of the game. In some games this may include the previous three frames in order for the network to get a sense of what direction objects in the images may be moving. This enables the network to lead the target when acquiring and firing projectile based weapons, such as the rocket launcher in DOOM.

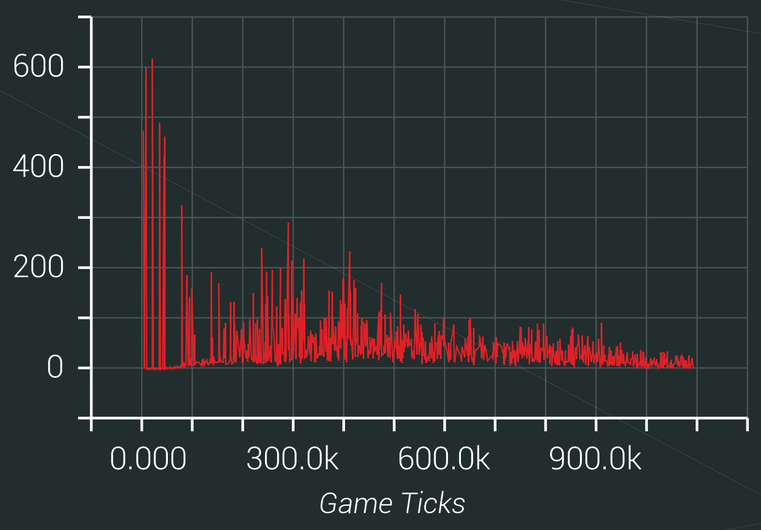
Process

The above diagram shows the architecture of the Deep Q Network. It consists of two convolutional layers that process the input images, this allows the network to learn to create feature vectors for game states. The two convolutional layers then feed into two fully connected layers that produce the estimated Q values for each action that the player can take.

Output

A Deep Q Network outputs an estimated Q value for the state that would occur if a specific action was taken, in this specific case that would mean 3 possible values. These values can then be used in the error calculation for the network, slowly dragging their values up by propagating rewards back through the state-action pairs that lead to beneficial outcomes.

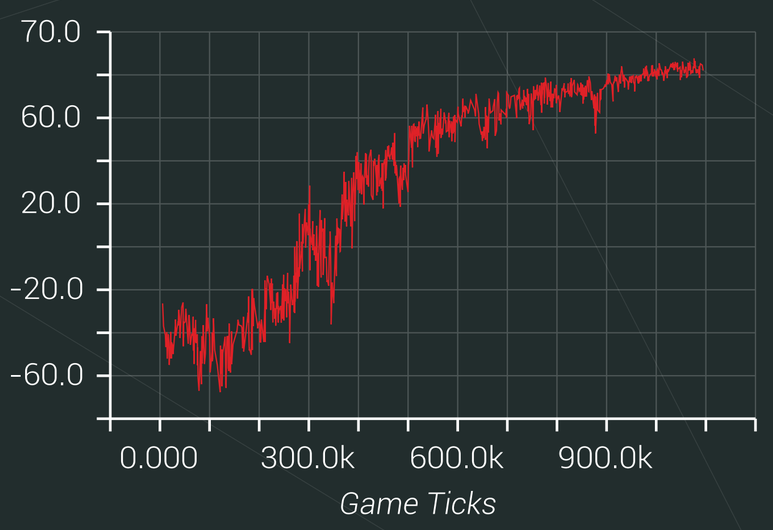
Mean/Error



Results

These two graphs show the training phase (1.1M ticks) of a simple scenario in doom. The more interesting results are on the right with the average Q-Value being shown during the training run, this is a good measurement of how well the network is training. As the values of the states back propagate they naturally increase the values associated with the actions. On the left is the error during the training phase.

Mean/Q-Value



EPSRC
Engineering and Physical Sciences Research Council

University of Essex

iGGi

This research is supported by EPSRC through the Centre for Doctoral Training on Intelligent Games and Game Intelligence, grant number EP/L015846/1