# Design Defense

When it comes to intelligence and learning, humans and machines can have a lot of similarities in the way they seem to learn. However, it's still very important to understand that there are differences, and the way they differ is important to understand when analyzing machine learning algorithms and models. For the example of learning how to solve a maze, without any prior knowledge about how to solve it other than simply how to navigate around the maze, at the beginning stages of learning, both machines and humans would take similar approaches, making random guesses about which direction to go, until both learned that they were progressing further into the maze. Humans, on the other hand, have much more context about their surroundings, and would likely not make many mistakes like running into walls, or choosing to go in the same direction, which they know led to a dead-end on a previous attempt. In contrast, machines would take the trade-off of sometimes taking these incorrect or redundant paths, with the benefit of being able to run many more simulations of the maze than a human would be able to perform, in a much shorter amount of time, leading to a quick learning rate. However, it's also important to remember that the knowledge the machine is learning might be specific to the task at hand, whereas a machine might need to be completely re-trained, given a new environment.

Assuming the maze did not change, after having solved it the first time, humans would likely be able to solve the maze with a 100% success rate after only the first or second time solving it, as humans are just able to have much more context about the problem, knowing the exact path to take and no longer ever needing to take incorrect paths. Machines, on the other hand, need many more examples to adequately tune their model of the problem. For the maze, on each step or move for solving the maze, the "intelligent agent" will choose an action to take, or which move to make next, either by exploration or by exploitation. The machine will keep a table of values for each state/action combination of the environment (the maze) and update it so that when choosing an action by exploitation in the future, it will have a better understanding of what the best action to take will be, and over time this understanding will become greater and greater. At a certain point, the model will almost exclusively be using exploitation to solve the maze, until the point when it has also reached a 100% success rate.

Exploitation is the process of having the model choose the next action to take in the environment based on the knowledge the model itself has gained about the problem, based on all the previous events that have occurred. The model will keep a Q-Table of rewards (or values) associated with each action that can be taken in each state that the environment can be in. When using exploitation, the model will make use of this Q-Table, and based on all the rewards associated with each action, the model will take the best one, and make that the actual next action performed on the environment (the maze). Exploration, on the other hand, is the process of the model making a random choice, based on all the available actions there are to take, disregarding the rewards currently stored in the Q-Table. By doing this, it can potentially lead to the model improving, since it's taking actions that might not necessarily yield the best reward as of the current Q-Table, but might still be more beneficial overall. This also has the benefit of not getting too focused on a single part of the problem, and not fully "exploring" the entire space of actions that can be taken. For this problem, an exploration rate of about 10% seemed to yield the best results for me, meaning that for every 10 actions taken during solving the maze, for 1 of those actions, the model would choose a random move to make. If the exploration rate is too low, the model might not explore enough of the maze, and might not learn to take paths that would lead to better results overall. On the other hand, having an exploration rate that is too high would not rely on any of the learned information about the environment, and would not lead to good results overall.

To help solve this problem, we made use of neural networks and implemented the "Deep Q-Learning" algorithm in order to solve the maze. The input layer consisted a layer of nodes whose number of nodes matched the size of the maze we provided it, so that this model could be used for a maze of any size potentially. This was followed by a single hidden layer, with the same number of nodes as the input layer. Lastly, that hidden layer is fed into the final output layer, which consisted of only 4 nodes, since the only actions that could be taken in the maze are to move left, right, up or down. Then, on every step of every game we decide to play/simulate, we can train the model based on the state of the maze and over time obtain a model that will learn to solve the maze in an efficient and correct manner.

**References**

Fernandez, E. (2019, November 28). *Ai is not similar to human intelligence. thinking so could be dangerous*. Forbes. Retrieved April 12, 2023, from https://www.forbes.com/sites/fernandezelizabeth/2019/11/30/ai-is-not-similar-to-human-intelligence-thinking-so-could-be-dangerous/?sh=65883c726c22

Khan, T. (2022, January 2). *Reinforcement learning – exploration vs exploitation tradeoff*. AI ML Analytics. Retrieved April 12, 2023, from https://ai-ml-analytics.com/reinforcement-learning-exploration-vs-exploitation-tradeoff/#:~:text=Exploration%20is%20more%20of%20a,lead%20to%20long%20term%20benefit.&text=Exploitation%20basically%20exploits%20the%20agent's,to%20get%20the%20most%20reward.