**ReInforcement Learning workshop assets:**

**Exploration vs Exploitation**

**Goals:**

**Introduce to RL (diff of Deep/Shallow, Supervised/Unsupervised)**

**Basic components of RL**

**Reward Hypothesis – goal is to optimize by maximization of cumulative reward**

**Agent, Enviro, Algo**

**Episodic vs Continuous Task**

**Monte Carlo reward vs Temporal Difference Learning**

**Value-Based vs Policy-Based vs Model-based**

**Early Concept: Q-Learning**

The Q-table

Q is for quality

Q-Value = the choice with the best reward given the available options in a current state

Q-Learning Algorithm (Action Value Function

**First Evolution: Deep Q-Learning (DQN)**

The Q-table

Q Table values are determined by a Neural net

**Current Evolution: Pixel based, A3C, PPO?**

Andrew Ng is wrong. (https://www.oreilly.com/ideas/practical-applications-of-reinforcement-learning-in-industry)

Assets:

Spreadsheet with landscape map of RL

Vocabulary

**Genesis of RL:**

[Dynamic Programming](MarkovDecisionProcess/1960-howard-dynamicprogrammingmarkovprocesses.pdf) and [Markov Processes](MarkovDecisionProcess/markov_process.pdf) – Ronald Howard

RL does NOT equal real world results

Anatomy of an RL environment (diagram):

RL = exploration and exploitation. Exploring the environment in order to exploit a reward system. This is similar to the classic balance of Bias vs Variance in Supervised and Unsupervised learning. EvE must balance enough exploration for the agent to know the environment well enough to make a choice of optimal explotation.

**Agent** – sometimes called an Actor – does the exploring and received the reward

**Algo** – sometimes referred to as a “policy” the heuristics used to make a decision. i.e “Should I move forward or not? Is there a reward if I move forward?”

**State** – We could simply insert Markov Decision Processes here. State is meant to qualify the condition/location/status of the agent as a given point in time.

**Actions** – options available to the agent. I.e step forward, step sideways.

**Value/Reward** – The value of a state refers to how much reward the agent will receive for performing an action to achieve said state.

**ToolKit**

**Environment/Simulator** – An agent always operates within an environment. Environments are either Stochastic (elements of the environment are constantly changing) or Deterministic (the environment has known “rules” or limits”)

**Policy**

**On-Policy vs Off-Policy**

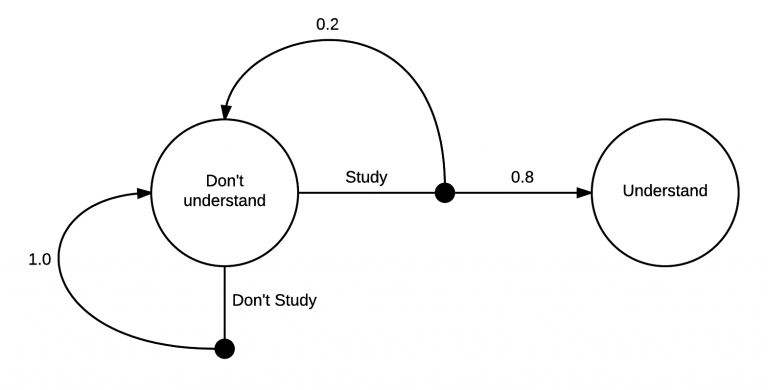
**Curriculum Learning**

Introduce a simplified version of a task to an agent, slowly increasing the complexity

**Domain Randomization**

Introducing variability into the simulation to assist models in generalizing to the real world.

**Probability distribution**



This MDP shows the process of learning about MDPs. At first you are in the state Don’t understand. From there, you have two possible actions, Study or Don’t Study. If you choose to not study, there is a 100% chance that you will end up back in the Don’t understand state. However, if you study, there is a 20% chance you’ll end up back where you started, but an 80% chance of ending up in the Understand state. The probability is distributed between 80% chance of going to the understand state, and 20% chance of the don’t understand state.

**Temporal Limitation** – in certain environments, a snapshot in time (TEMPORAL) may not provide enough information to predict the best action. In these instances multiple snapshots are need to understand the direction/momentum/inertia of objects in motion to choose the best action.

**Entropy** – Entropy in RL is a measurement of the unpredictability of possible actions in a given policy. Higher Entropy = higher randomness in action choice

**Adaptive Task Selection**

**Behavioral Cloning**

**Single Agent vs multi Agent**

**Discrete vs continuous Actions**

**Dense vs Spare Rewards**

**Continuous Control**

**learning curricula**

**memory systems**

**Episode –** One complete agent training session on a distinct environment. Training ends when either the goal is achieved by the agent or a pre-determined number of actions have been taken and the agent training is terminated.

**Run(s)**

**Evaluative vs Instructive feeback**

Instructive feedback tells you HOW to achieve your goal

Evaluative feedback tells you HOW WELL you achieved your goal

Markov state = future from that state is conditionally independent of the past, give that we know the current state. i.e a ball flying through the air. If state = position and velocity that can tell us where it has been and where it will go. However, if we do not know velocity state is no longer Markov. The current state does NOT summarize all past states