Hearthstone is a recently-released turn-based Trading Card Game that is gaining in popularity. This project utilizes **supervised deep networks** and **experience replay Q-learning** to develop an artificial intelligence agent capable of beating hand-coded heuristic agents.

### Methodology

Hearthstone is modelled naturally by a **Markov Decision Process**, with rewards only at terminal states and a discount of 0.8 to account for nondeterminism. Several techniques were developed in response to the unique challenges of the game, and two approaches to modelling the value of state-action pairs were used.

**Large State Space**: 2 relatively low-dimensional feature extractors were tested for their ability to effectively represent game states, and the best one selected. **Large Action Space**: The actions and transition space of the Hearthstone MDP is large enough to prevent direct evaluation. A **Monte-Carlo depth-limited A* search** was used to estimate the expected value of a game state.

### Feature Extraction

The representation used has the following properties:
- captures **critical** info: health/attack/mana etc.
- symmetrical, includes both players
- low-dimensional to allow **fast convergence**
- models only **general** player stats and not hero/minion/spell-specific interactions

### Supervised Learning

Linear and deep neural models were trained against expert guidance derived from a heuristic agent:

- **Game states**
- **Expert agent**
- **Predictive Model**
- **Oracle**
- **Reward values**

### Reinforcement Learning

Each epoch simulates a game, pitting the existing model against itself, with no online training involved:

- **Game states**
- **ε-greedy**

Then an **experience-replay** scheme updates the model:

- **Models**
- **Game history**
- **Experience**

This scheme allows the model to consolidate its experiences after each epoch and improving **convergence**. Less games are required to attain good performance.

### Results

<table>
<thead>
<tr>
<th>Win rate against heuristic agent</th>
<th>Convergence</th>
<th>Best</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear Learner</td>
<td>77%</td>
<td>81%</td>
</tr>
<tr>
<td>Deep Neural Learner</td>
<td>76%</td>
<td>78%</td>
</tr>
<tr>
<td>Q-learning (state difference)</td>
<td>40%</td>
<td>65%</td>
</tr>
<tr>
<td>Q-learning (final state)</td>
<td>50%</td>
<td>55%</td>
</tr>
</tbody>
</table>

**Learning curve of supervised learner**

**Learning curve of q-learner**

**Final State**

**State Difference**

**Training set data size**

**Training epochs**