Deep Q-Learning for No-Limit Texas Hold'em: Architecture and Convergence Analysis

Kacper Duda, Marek Dzierżawa, Kacper Karabasz
AGH University of Krakow

January 7, 2026

Abstract

We present a Reinforcement Learning approach to solving No-Limit Texas Hold'em (NLHE), a high-dimensional extensive-form game with imperfect information. Our solution implements a Deep Q-Network (DQN) agent utilizing a dense reward signal based on normalized chip accumulation. We engineer a compact 134-dimensional feature vector encapsulating hole cards, board state, and pot odds. The model architecture features a split-head Multi-Layer Perceptron (MLP) for simultaneous discrete action selection and continuous bet-sizing. We evaluate the agent's convergence properties over 2,000 episodes, demonstrating early-stage strategy acquisition against stochastic policies.

1 Introduction

No-Limit Texas Hold'em presents a significant challenge for conventional game-theoretic solvers due to its vast decision tree and hidden information states ($\approx 10^{161}$ decision points). While Counterfactual Regret Minimization (CFR) remains the state-of-the-art for Nash Equilibrium approximation in Poker, Deep Reinforcement Learning (DRL) offers a scalable alternative by approximating the optimal value function $Q^*(s,a)$ directly from self-play or gameplay experience.

This work implements a Model-Free RL agent using Deep Q-Learning (DQN) with Experience Replay. We focus on the feature engineering required to represent the poker game state efficiently for a feed-forward neural network and analyze the stability of learning in a 3-player environment.

2 Methodology

2.1 State Representation

The partial observation o_t at time t is mapped to a feature vector $\phi(o_t) \in \mathbb{R}^{134}$. The state space S is composed of:

- **Card Embeddings**: Hero's hand $H \in \{0,1\}^{52}$ and Community cards $C \in \{0,1\}^{52}$ represented via one-hot encoding.
- **Game Context**: Normalized stack sizes s_i, current bets b_i, andpot size P, such that all values lie approximately in $[0,1]$.
- **Heuristic Features**: To accelerate training, we inject domain knowledge via pre-computed hand strength metrics $E(H,C) \in \mathbb{R}^{14}$ (bucketed hand rank + normalized high cards).

2.2 Network Architecture

We employ a dense Multi-Layer Perceptron (MLP) parameterized by θ. The network architecture (Fig. 1) consists of:

$$\begin{aligned} h_1 &= \text{ReLU}(W_1 x + b_1), && W_1 \in \mathbb{R}^{512\times 134} \\ h_2 &= \text{ReLU}(W_2 h_1 + b_2), && W_2 \in \mathbb{R}^{512\times 512} \\ h_3 &= \text{ReLU}(W_3 h_2 + b_3), && W_3 \in \mathbb{R}^{256\times 512} \end{aligned}$$

The final layer splits into two heads:

1. **Action Value Head** ($Q(s,\cdot) \in \mathbb{R}^3$): Linearly estimates Q-values for discrete actions $A = \{\text{Fold, Call, Raise}\}$.
2. **Sizing Head** ($\sigma(s) \in [0,1]$): A Sigmoid unit determining the raise magnitude as a fraction of the legal betting interval (min-raise to all-in).

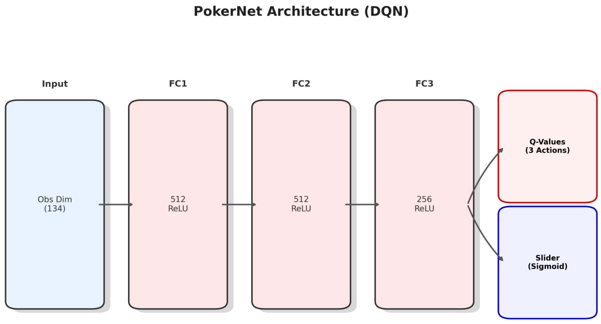

Figure 1: PokerNet Architecture. The network maps the 134-dim state vector to action-values and a continuous bet-sizing parameter.

2.3 Training Algorithm

The agent minimizes the temporal difference error using the Huber Loss to ensure robustness against outliers in variance-heavy poker rewards:

$$\mathcal{L}(\theta) = \mathbb{E}_{(s,a,r,s') \sim \mathcal{D}} \left[\delta_\theta^2 \right] \tag{1}$$

where $\delta_\theta = r + \gamma \max_{a'} Q(s', a'; \theta^-) - Q(s, a; \theta)$.

Table 1: Hyperparameters

Parameter	Value
Batch Size	128
Learning Rate (α)	1×10^{-4} (Adam)
Discount Factor (γ)	0.99
Target Update Period	500 steps
Replay Buffer Size	10^5 transitions
ϵ-decay strategy	Exp. decay ($1.0 \to 0.05$)

3 Experiments and Results

3.1 Convergence Analysis

Training was conducted over 2,000 episodes. The dense reward function is defined as the normalized net profit at the hand's conclusion.

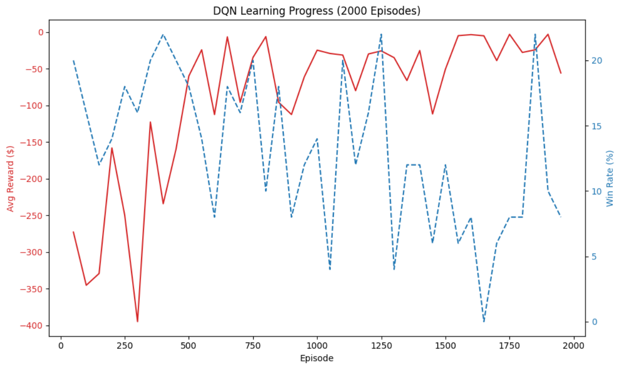

Figure 2: Training progression. Blue (dashed): Win Rate (%). Red (solid): Moving average of Normalized Reward.

Fig. 2 illustrates the training dynamics. The high variance is intrinsic to poker (luck factor). However, the upward trend in Average Reward indicates the policy π_θ is improving over the random baseline. The win rate oscillation suggests the agent is shifting between aggressive and passive strategies as it explores the state space.

3.2 Visual Verification

A custom UI (Fig. 3) validates the internal state tracking and decision-making process.

4 Conclusion

The proposed DQN architecture successfully parses the complex poker state space. Feature engineering significantly aids the MLP in recognizing hand strength. Future work addresses the discrete-continuous hybrid action space proper handling (e.g., via Parameterized Action Space Q-Network or PPO) and migrating to Self-Play (SP) to minimize exploitability rather than maximizing profit against fixed policies.

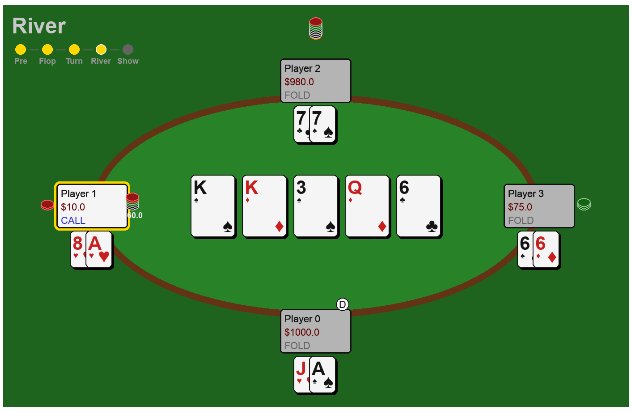

Figure 3: Environment Visualization.