RL-Guided Adaptive Dose Optimization for Chemotherapy

NovaCure Therapeutics -- Personalizing Cancer Treatment Through Sequential Decision-Making

Section 1: Industry Context and Business Problem

The Landscape

Cancer treatment is one of the most consequential sequential decision-making problems in medicine. A patient undergoing chemotherapy receives a series of treatment cycles -- typically 4 to 8 rounds spaced 2 to 3 weeks apart. At each cycle, the oncologist must decide: what dose should the patient receive?

Too high a dose causes severe side effects -- neutropenia, organ damage, treatment discontinuation. Too low a dose allows the tumor to grow back between cycles. The optimal dosing strategy depends on the patient's current health state: tumor size, white blood cell count, kidney function, and overall toxicity level.

Traditional clinical practice uses **fixed dosing protocols**. A patient weighing 70 kg with a certain body surface area receives a standard dose, regardless of how their body is responding. Some oncologists adjust doses reactively -- reducing the dose after severe side effects appear -- but this is ad hoc and inconsistent.

NovaCure Therapeutics

NovaCure Therapeutics is a mid-stage pharmaceutical company with \$420M in annual revenue, focused on solid tumor oncology. They have a portfolio of three chemotherapy drugs in clinical trials and two FDA-approved treatments.

Their clinical data spans 12,000 patients across 8 clinical trials, with detailed longitudinal records: tumor measurements via imaging (every 3 weeks), complete blood counts (weekly), liver and kidney panels (biweekly), and patient-reported quality-of-life scores.

The Problem

NovaCure's lead drug, NC-4817, shows promising efficacy in Phase II trials for advanced non-small-cell lung cancer. However, the trial data reveals a critical pattern:

- 34% of patients require dose reductions due to toxicity
- 18% discontinue treatment entirely due to adverse effects

- Among patients who complete all cycles, tumor response varies enormously -- some achieve complete remission, others show minimal response

The oncology team suspects that a personalized, adaptive dosing strategy could simultaneously improve efficacy (better tumor control) and reduce toxicity (fewer dose reductions and discontinuations). But the challenge is that the optimal dose at cycle k depends on everything that has happened in cycles 1 through $k-1$.

This is a sequential decision-making problem under uncertainty. It is exactly the kind of problem that reinforcement learning was designed to solve.

Business Impact

If NovaCure can demonstrate that an RL-guided dosing protocol improves outcomes, the potential impact is:

- **Regulatory**: A companion dosing algorithm submitted alongside the drug could accelerate FDA approval and differentiate NC-4817 from competitors
- **Clinical**: Improved patient outcomes (higher response rates, fewer adverse events) directly impact the drug's commercial value
- **Financial**: Reducing treatment discontinuation from 18% to under 10% could increase revenue by \$45M annually for NC-4817 alone
- **Platform**: A validated RL dosing framework could be applied across NovaCure's entire drug portfolio

Section 2: Technical Problem Formulation

Why Reinforcement Learning?

This problem has three properties that make RL the right tool:

1. **Sequential decisions**: The dose at each cycle affects future states. This is not a one-shot prediction problem.
2. **Delayed rewards**: The ultimate outcome (tumor response) is not known until the end of treatment. Immediate biomarkers provide partial signals.
3. **Exploration-exploitation tradeoff**: We want to exploit known effective doses, but we also need to explore personalized adjustments.

A supervised learning model could predict "what dose will minimize toxicity?" at a single time point, but it cannot reason about the cumulative effect of a sequence of doses. An RL agent can.

MDP Formulation

We model chemotherapy dosing as a Markov Decision Process:

$$\text{MDP} = (S, A, P, R, \gamma)$$

States S : At each treatment cycle t , the state captures the patient's current condition:

$$s_t = (\text{tumor\_size}_t, \text{WBC}_t, \text{toxicity\_grade}_t, \text{kidney\_function}_t, \text{cycle\_numbe}$$

- Tumor size: continuous, measured in mm (from imaging)
- White blood cell count (WBC): continuous, measured in cells/uL
- Toxicity grade: discrete, 0-4 (CTCAE grading scale)
- Kidney function (eGFR): continuous, mL/min
- Cycle number: discrete, 1 to 8

Actions A : The dose level for the current cycle:

$$A = \{0.6D,\ 0.7D,\ 0.8D,\ 0.9D,\ 1.0D,\ 1.1D\}$$

where D is the standard protocol dose. Six discrete dose levels ranging from 60% to 110% of standard.

Transition Dynamics $P(s_{t+1}|s_t, a_t)$: The patient's state at cycle $t+1$ depends on: - Current tumor size (tumor growth/shrinkage dynamics) - Drug effect on tumor (dose-dependent kill rate) - Drug toxicity on healthy tissue (dose-dependent WBC suppression) - Recovery between cycles

We model these transitions using a pharmacokinetic/pharmacodynamic (PK/PD) simulator calibrated to NovaCure's clinical trial data.

Reward Function $R(s_t, a_t)$:

$$R(s_t, a_t) = w_1 \cdot \Delta\text{tumor} + w_2 \cdot \text{toxicity\_penalty} + w_3 \cdot \text{completion\_bonu}$$

where: - $\Delta\text{tumor} = \text{tumor\_size}_t - \text{tumor\_size}_{t+1}$ (tumor shrinkage is positive reward) - $\text{toxicity\_penalty} = -c$ if toxicity grade ≥ 3 (severe adverse event) - $\text{completion\_bonus} = +b$ if the patient completes the cycle without dose reduction

Weights $w_1 = 1.0$, $w_2 = 5.0$, $w_3 = 0.5$ are set by the clinical team to prioritize safety.

Discount Factor: $\gamma = 0.95$ -- the agent values future tumor response almost as much as immediate response, but slightly prefers earlier improvements.

The Markov Property

Does this formulation satisfy the Markov property? The state s_t captures the patient's current tumor size, blood counts, toxicity, and kidney function. Given this state, the transition to s_{t+1} does not depend on the full treatment history -- only on the current state and the chosen dose.

This is an approximation. In reality, cumulative drug exposure creates some history dependence. We handle this by including cycle number in the state (which implicitly captures how many doses the patient has received).

Baseline Comparison

We compare the RL agent against: 1. **Fixed-dose protocol**: Standard dose at every cycle (current clinical practice) 2. **Rule-based adjustment**: Reduce dose by 20% if toxicity grade ≥ 3 , increase by 10% if no toxicity (common clinical heuristic) 3. **RL agent**: Q-learning with the MDP formulation above

---

Section 3: Implementation Notebook Structure

The implementation notebook follows this structure:

Part A: Patient Simulator (PK/PD Model)

Build a realistic patient simulator that models: - Tumor growth dynamics (Gompertzian growth model) - Drug-tumor interaction (log-kill hypothesis) - Hematological toxicity (WBC suppression and recovery) - Inter-patient variability (random PK parameters)

```
class PatientSimulator:
    """
    Simulated patient for chemotherapy dose optimization.
    Models tumor dynamics, drug effect, and toxicity.
    """
    def __init__(self, patient_params=None):
        # TODO: Initialize patient-specific PK/PD parameters
        # - tumor_growth_rate: Gompertzian growth rate
        # - drug_sensitivity: how much dose affects tumor
        # - toxicity_sensitivity: how much dose affects WBC
        # - baseline_wbc: starting white blood cell count
        pass

    def step(self, dose_fraction):
        """
        Simulate one treatment cycle.
        Returns: next_state, reward, done
        """
        # TODO: Implement tumor dynamics
        # TODO: Implement WBC suppression and recovery
        # TODO: Compute toxicity grade
        # TODO: Compute reward
        pass
```

Part B: MDP Environment (Gymnasium-Compatible)

Wrap the simulator in a Gymnasium-compatible interface:

```
class ChemoDoseEnv(gym.Env):
    """
    Gymnasium environment for chemotherapy dose optimization.
    """
    def __init__(self):
        # TODO: Define action_space (6 discrete dose levels)
        # TODO: Define observation_space (5-dimensional state)
        pass

    def reset(self):
        # TODO: Sample a new patient and return initial state
        pass

    def step(self, action):
        # TODO: Map action to dose, simulate, return (obs, reward, done, info)
        pass
```

Part C: Baseline Agents

```
class FixedDoseAgent:
    """Always prescribes the standard dose."""
    def choose_action(self, state):
        return 4  # 1.0D (index 4 in action space)

class RuleBasedAgent:
    """Adjusts dose based on toxicity grade."""
    def choose_action(self, state):
        toxicity = state[2]
        # TODO: Implement rule-based dose adjustment
        pass
```

Part D: Q-Learning Agent

```
class ChemoQLearningAgent:
    """
    Q-learning agent for dose optimization.
    Discretizes continuous states for table-based Q-learning.
    """
    def __init__(self, n_actions=6, alpha=0.1, gamma=0.95, epsilon=1.0):
        # TODO: Initialize Q-table and hyperparameters
        pass

    def discretize(self, state):
        # TODO: Bin continuous state variables
        pass

    def choose_action(self, state):
        # TODO: Epsilon-greedy action selection
        pass

    def update(self, state, action, reward, next_state, done):
        # TODO: Q-learning update rule
        pass
```

Part E: Training and Evaluation

```
# TODO: Train the Q-learning agent over many simulated patients
# TODO: Evaluate all three agents (fixed, rule-based, Q-learning)
# TODO: Compare metrics: tumor response, toxicity events, completion rate
# TODO: Visualize dosing trajectories for sample patients
```

Section 4: Production and System Design Extension

From Prototype to Clinical Decision Support

Deploying an RL dosing algorithm in a clinical setting requires significant engineering beyond the prototype:

Regulatory Requirements: - The algorithm must be submitted to the FDA as a Software as a Medical Device (SaMD) - It requires clinical validation: a prospective trial comparing RL-guided dosing to standard protocol - The algorithm's decisions must be explainable -- oncologists need to understand why a particular dose was recommended

System Architecture:

```
Patient EHR Data --> Feature Extraction --> State Encoder
                                                |
                                                v
                                    RL Policy Server (Q-table / DQN)
                                                |
                                                v
                                    Dose Recommendation + Confidence
                                                |
                                                v
                                    Oncologist Dashboard (final decision)
```

The RL agent provides a **recommendation**, not a prescription. The oncologist always has final authority. The system displays: - Recommended dose level - Confidence score (based on how many similar patients the agent has seen) - Expected outcomes (predicted tumor response and toxicity probability) - Alternative dose options with predicted trade-offs

Safety Constraints: - Hard limits: dose can never exceed 1.1x standard or fall below 0.5x standard - Toxicity override: if current toxicity grade is 4, the agent is overridden and dose is reduced to 0.6x regardless - Human-in-the-loop: the oncologist can accept, modify, or reject the recommendation

Offline RL for Real Data: In production, the agent cannot explore randomly on real patients. Instead, we use **offline reinforcement learning** -- training the agent on historical clinical trial data without ever interacting with real patients during training. The agent learns from the existing data what the optimal dosing policy would have been.

Scaling Considerations

- **Multi-drug protocols**: Many cancer treatments involve combinations of 2-3 drugs. The action space expands to dose levels for each drug (e.g., 6 x 6 x 6 = 216 combinations). Deep Q-Networks handle this naturally.
- **Continuous monitoring**: Integrating real-time lab data (daily blood counts from wearable devices) could enable more frequent dose adjustments within a cycle.
- **Transfer learning**: An agent trained on NC-4817 data could be fine-tuned for a new drug with limited data, accelerating development of dosing algorithms across the portfolio.

Expected Outcomes

Based on simulation results and analogous published studies: - Tumor response rate improvement: +12-18% over fixed-dose protocol - Grade 3+ toxicity reduction: -25-35% - Treatment completion rate: 82% to 91% - Estimated revenue impact: \$45-60M annually for NC-4817

References:

1. Sutton, R.S. and Barto, A.G. (2018). *Reinforcement Learning: An Introduction*. 2nd Edition. MIT Press.
2. Padmanabhan, R. et al. (2017). "Reinforcement learning-based control of drug dosing for cancer chemotherapy treatment." *Mathematical Biosciences*, 293, 11-20.
3. Liu, Y. et al. (2020). "Reinforcement Learning for Clinical Decision Support in Oncology." *Nature Medicine*, 26, 1228-1234.
4. U.S. FDA. (2021). "Artificial Intelligence/Machine Learning (AI/ML)-Based Software as a Medical Device (SaMD) Action Plan."