Training an ML Model

WM Joanne

Development Process

1. Model Selection
2. Data Preparation
3. Determine Pretraining Strategy
4. Build a Data Pipeline
5. Training the Model
6. Validate Inference and Deployment
7. Playtesting and Iterative Improvement

Development Process

1. Model Selection
2. **Data Preparation**
3. Determine Pretraining Strategy
4. Build a Data Pipeline
5. Training the Model
6. Validate Inference and Deployment
7. Playtesting and Iterative Improvement

Prerequisites for Training the Model

1. **Model Selection**

   Decide the type of model to be used based on the application:

   - Reinforcement Learning (RL) Strategies
   - Natural Language Models..

2. **Data Preparation**

   This is the most time-intensive phase, often requiring up to 50% of development resources.
   - Key considerations include:
     - **Data Types**
     - **Pipeline Design**
     - **Specific to selected model**

3. Model Selection

Reinforcement Learning (RL) Strategies

- Uses PPO (Proximal Policy Optimization) - it's fast and stable _2018
- Can learn player patterns in real-time
- Maintains consistent performance
- Alternative Options:
  1. DQN (Deep Q-Network)
  2. A3C (Asynchronous Advantage Actor-Critic)
  3. SAC (Soft Actor-Critic)

SLM, LLM

- Small mistral-nemo-minitron-8b-8k-instruct
  - -> Is it suitable?
    - Mir5 AI mechanics involve processing numerical data (like health, damage, or position) and making decisions based on rules or optimization algorithms.

3. Model Selection

recent RL strategies that were emerging in late 2023 ~ early 2024 for MMORPGs

1. **Hierarchical Transformer-based RL (HTRL)**
   - Uses hierarchical structure to handle long-term dependencies in MMO environments
   - Better at managing complex action spaces typical in MMORPGs

2. **Multi-Agent Population-Based Training (MA-PBT) _ 2023 Jun**
   - Particularly effective for MMORPGs as it can:
     - Handle diverse player behaviors by adapting to meta-game changes
     - Scale well with large player populations

3. **Decision Transformer Architecture**
   - Treats RL as a sequence modeling problem
   - Particularly good for:
     - Complex skill rotations
     - Group combat scenarios & Resource management

3. Determine Pretraining Strategy

Decide whether to use pre-trained models

1. **Use a pre-trained model – the common approach**
2. Pre-train a base model (essentially a shell) ourselves

4. Build a Data Pipeline

Create a data pipeline to input our own data

Keep benchmarking in mind during this step!

4. Build a Data Pipeline

Create a data pipeline to input our own data

Keep benchmarking in mind during this step!

what are the Key Metrics? -> **Challenge**, Fun, and Fairness

Benchmarking Focus:

- Validate that the AI provides a balanced player experience.
- Ensure the AI adapts **without being overly predictable or impossible to beat**.
- assess if the pipeline is delivering the desired gameplay outcomes.

5. Training the Model

Secure GPU resources (at least four small GPUs are typically needed, but determine how much you will actually need).

Training (`python train.py`)

- Fine-tune parameters while monitoring outputs and benchmarking performance.
  a. Performance metrics: Evaluate how much the players' HP decreases and analyze results from various angles.

6. Validate Inference and Deployment

Verify inference results

Check API calls and ensure that latency is reduced, and performance is optimal.

7. Playtesting and Iterative Improvement

You need to run at least 100 (100 is too low, ideally between 1000 to 10,000) playtests to improve the model's quality.

DATA NUM	ALL KILL	KILL #	WIN/LOSE	HP of players alive
1	O	12	W	
2	X	2	L	
3	X	7	L	
4	X	4	D	
5	X	2	D	
6	O	9	W	
...	...	...	...	...
100	X	4	O	

- As the number of columns above increases, the quality of the model improves.

2. Data Preparation

1. **Data Types**: Player stats (position, HP, MP, damage, character type, skills, history, rules).
2. **Pipeline Design**: use tools like Hugging Face's `pipeline` for efficient tokenization and processing (e.g., converting JSON to text).
3. **Specific to RL**: Diverse input types are critical. Pretrained models might not exist, so it's recommended to adopt the latest RL architectures.

(widely depending on the model used)

For language models,**Hugging Face's Pipeline**

- JSON → String: Hugging Face automatically tokenizes sentence-form inputs.

Data Preparation - Categorization

- **Core Entity Data:**
  - General attributes such as health, mana, position, and attack power.
- **Behavioral Traits:**
  - AI-specific details like movement modes, destinations, and skill execution.
- **Combat & Buff Interaction:**
  - Tracks engagement capabilities and status effects.

{"ai_data":{"event_data":{},"state_data":[{"common":{"buff_states":{},"direction":139.55166625976562,"entity":4294967773,"fly_stamina":10000,"hp":4000,"magic_attack":150,"mp":600,"position":{"x":-1349400.0,"y":255800.0,"z":19542.800508},"velocity":{}},"specific":{"bot":{"char_id":2}}},{"common":{"buff_states":{},"destination":{"x":-1351285.1850464446,"y":260415.0980473864,"z":19544.950401658396},"direction":-28.604249954223633,"entity":4294967793,"fly_stamina":10000,"hp":4000,"magic_attack":150,"move_mode":10,"mp":600,"position":{"x":-1349400.0,"y":255800.0,"z":19542.800508},"skill_info":{"end_ticks":1732499507.48468,"skill_id":20103,"start_direction":{"y":331.395751953125},"start_position":{"x":-1351451.246506502,"y":259630.62149201651,"z":19544.95041004952}},"velocity":{}},"specific":{"bot":{"char_id":2}}},{"common":{"buff_states":{},"destination":{"x":-1351440.8836280536,"y":259753.20679473906,"z":19544.972588965866},"direction":-151.91177368164062,"entity":4294967776,"fly_stamina":10000,"hp":4000,"mp":600,"physical_attack":150,"position":{"x":-1349400.0,"y":255800.0,"z":19542.800508},"velocity":{}},"specific":{"bot":{"char_id":1}}},{"common":{"buff_states":{},"destination":{"x":-1351603.6372117384,"y":259535.91656024722,"z":19544.950408401426},"direction":-142.31626892089844,"entity":4294967788,"fly_stamina":10000,"hp":4000,"mp":600,"physical_attack":150,"position":{"x":-1349400.0,"y":255800.0,"z":19542.800508},"velocity":{}},"specific":{"bot":{"char_id":1}}},{"common":{"buff_states":{},"destination":{"x":-1351553.045596201,"y":260196.17705654478,"z":19545.105234935578},"direction":-151.92300415039062,"entity":4294967786,"fly_stamina":10000,"hp":4000,"mp":600,"physical_attack":150,"position":{"x":-1349400.0,"y":255800.0,"z":19542.800508},"velocity":{}},"specific":{"bot":{"char_id":3}}},{"common":{"buff_states":{},"destination":{"x":-1351249.960545602,"y":259465.56732787297,"z":19999.67315356906},"direction":-170.49795532226562,"entity":4294967802,"fly_stamina":10000,"hp":4000,"mp":600,"physical_attack":150,"position":{"x":-1349400.0,"y":255800.0,"z":19542.800508},"velocity":{}},"specific":{"bot":{"char_id":3}}},{"common":{"buff_states":{},"direction":-82.12332153320312,"entity":4294967805,"fly_stamina":10000,"hp":4000,"magic_attack":150,"mp":446,"position":{"x":-1349400.0,"y":255800.0,"z":19542.800508},"velocity":{}},"specific":{"bot":{"char_id":2}}},{"common":{"buff_states":{},"destination":{"x":-1351157.3728768209,"y":259610.6823160432,"z":19544.95041056385},"direction":165.24473571777344,"entity":4294967796,"fly_stamina":10000,"hp":4000,"mp":600,"physical_attack":150,"position":{"x":-1349400.0,"y":255800.0,"z":19542.800508},"velocity":{}},"specific":{"bot":{"char_id":1}}},{"common":{"buff_states":{},"destination":{"x":-1351522.6479955136,"y":259806.66209650098,"z":19544.950409851903},"direction":-110.99695587158203,"entity":4294967817,"fly_stamina":10000,"hp":4000,"magic_attack":150,"mp":600,"position":{"x":-1349400.0,"y":255800.0,"z":19542.800508},"velocity":{}},"specific":{"bot":{"char_id":2}}},{"common":{"buff_states":{},"destination":{"x":-1351232.9492413823,"y":259732.58952397257,"z":19544.950408398512},"direction":-92.914710998853516,"entity":4294967812,"fly_stamina":10000,"hp":4000,"mp":600,"physical_attack":150,"position":{"x":-1349400.0,"y":255800.0,"z":19542.800508},"velocity":{}},"specific":{"bot":{"char_id":1}}},{"common":{"buff_states":{},"destination":{"x":-1351448.1485312178,"y":259547.514881667,"z":19544.95041324158},"direction":-175.02862548828125,"entity":4294967820,"fly_stamina":10000,"hp":4000,"mp":600,"physical_attack":150,"position":{"x":-1349400.0,"y":255800.0,"z":19542.800508},"velocity":{}},"specific":{"bot":{"char_id":1}}},{"common":{"buff_states":{},"destination":{"x":-1351481.083963569,"y":259472.16773559124,"z":19544.95041133114},"direction":175.50711059570312,"entity":4294967810,"fly_stamina":10000,"hp":4000,"magic_attack":150,"mp":600,"position":{"x":-1349400.0,"y":255800.0,"z":19542.800508},"velocity":{}},"specific":{"bot":{"char_id":2}}}],"timestamp":1732499507},"dungeon_inst_Id":65000001,"winner":{}}

Data Preparation - Categorization

1. **General Metadata**

- **`dungeon_inst_Id`**:
  - Represents the unique ID of the dungeon instance.
  - Used to differentiate between game sessions or environments.
- **`timestamp`**:
  - Indicates the time of data capture, useful for synchronization and debugging.
- **`winner`**:
  - Placeholder for storing the winner or outcome of the game.

2. AI Data

- **`ai_data`**:
  - Core section containing detailed AI-related information.
  - Split into `event_data` and `state_data`.

3. Event Data

- **`event_data`**:
  - Placeholder for capturing triggered events (currently empty).
  - Would typically include interactions such as attacks, buffs, or debuffs.

4. State Data

Common Attributes (Shared by all entities):

- **`buff_states`**:Tracks active buffs or debuffs affecting the entity.
- **`direction`**:Entity's facing direction, typically in degrees.
- **`entity`**:Unique identifier for the entity.
- **`fly_stamina`**:Stamina available for flying or aerial movement.
- **`hp`**:Current health points.
- **`mp`**:Current mana points.
- **`magic_attack`**:Entity's magical attack power (if applicable).
- **`physical_attack`**:Entity's physical attack power (if applicable).
- **`position`**:Current position in 3D space (`x`, `y`, `z` coordinates).
- **`destination`**:Targeted destination in 3D space.
- **`velocity`**:Current movement velocity (empty, implying static velocity in this dataset).

Data Preparation - Categorization

For RL, Example Schema for RL Dataset:

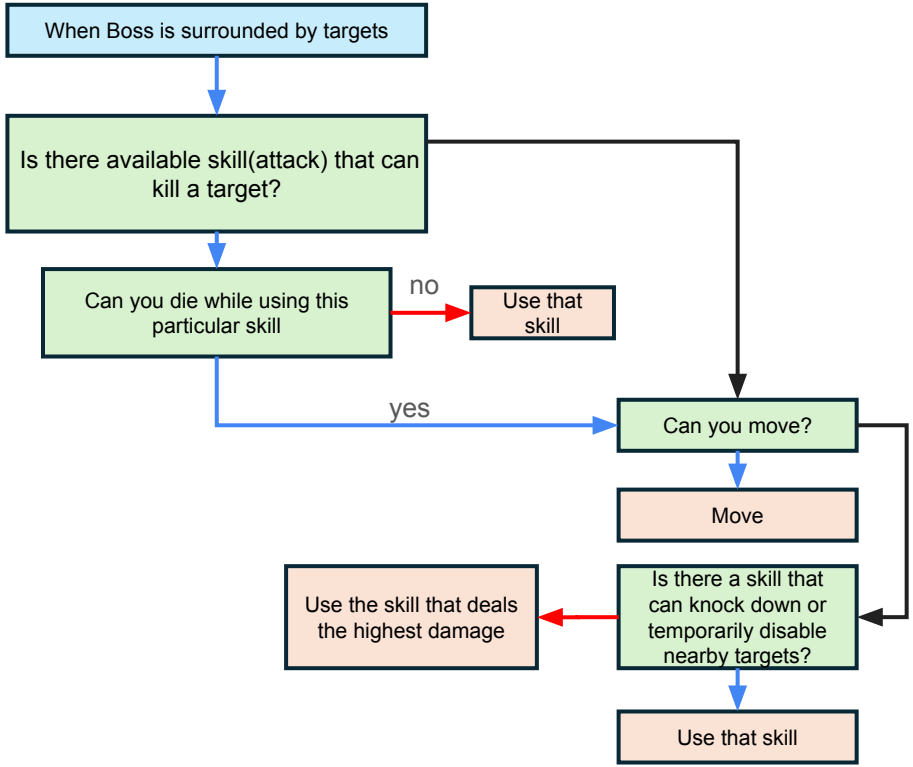
Structuring Data for RL

- **States:**
  - Define the state space, such as the entity's current position, health, buffs, and nearby entities.
- **Actions:**
  - Map available actions (e.g., move, attack, cast a skill) to the decision-making process.
- **Rewards:**
  - Craft a reward signal that encourages the AI to exhibit desired behaviors (e.g., successful attacks or avoiding damage)

Boss decision making flowchart

Scenario when surrounded by targets(players):

In a predictive AI model, instead of making sequential judgments, the system predicts various outcomes based on the current situation and executes the best possible result.



Situation Assess

- Are you surrounded by targets?
- Which skill will deal the most damage to the enemy?
- Is there a target you can kill with a single skill?
- Do you have any skills that benefit from attack attribute bonuses due to your active buffs?
- Are there any skills that work well when used consecutively?
- Is there a risk of you dying while using the skill?

Skill Selection

Select the most efficient skill based on the status of yourself and players

Checklist before Boss uses a skill

- Information to check based on the flowchart(on the previous page)
- Check all available skills and individual targets.

Category	Checkpoints	Purpose
Skill Information	Number of targets within the range of available skills	Determine which target to use a skill on among multiple targets to hit the most targets.
	Estimated Damage to Targets in Skill Range	Check skills with the highest damage output in relation to the number of targets within the skill's range.
	Required Resources for Skill	Compare the resources remaining after using a skill with the resources required for other skills to determine if additional skills can be used.
	Boss's Final Position	Check Boss's expected position after the attack finishes.
location information	Distance Between You and Target	Check if the skill's range includes the target. If outside of range, check the time required to track the target.
	Positions of All Targets in Combat	Check if Boss is Surrounded by Multiple Targets
Boss Information	Health	Check how much time Boss can survive based on the average damage (DPS) you receive.
	Active Buffs on You	Check the effects and remaining time of active buffs and debuffs on you.
Target Information	Target basic stats	Check the target's task and level.
	Target's Health	Check if you can reduce the target's health to zero with the expected damage.
	Active Buffs on the Target	Check the effects and remaining time of active buffs and debuffs on the target.