

Final Report

03.12.2019

Martyn Wei and Richard Ren
UndefeatablePokemonTrainer

Word Count: 1996
Penalty: 0

Introduction

The goal of this project is to build a neural network that can win 50% of battles in the game of Pokemon. First, we will explain the mechanics of the game. Essentially, two competing trainers take turns deciding on one of four moves that their respective Pokemon can perform. The choice of Pokemon and their moveset are determined before the match begins. Each trainer seeks to reduce the health of the opponent's Pokemon to zero, and must strategically decide on a move depending on the conditions of the turn. As illustrated in Figure 1, they may choose the status move *Will-O-Wisp* to burn the opponent and reduce their health every turn for the rest of the battle by an incremental amount rather than *Shadow Sneak* which inflicts greater damage for a single turn.

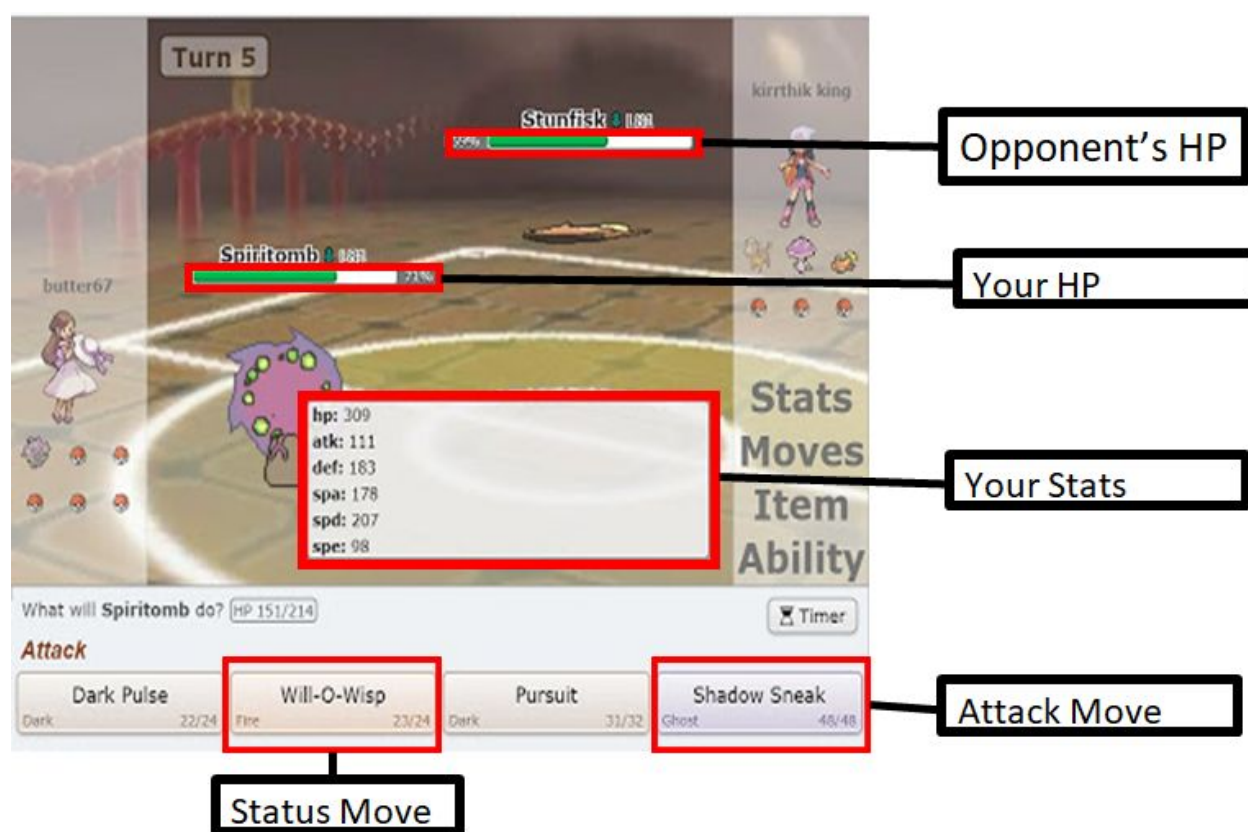



Figure 1: Screenshot of Pokemon Battle

To win 50% of battles, we developed a neural network to play a specified Pokemon with four moves. A neural network fits the task because it can accurately process a turn's numerous conditions. Using ideas from reinforcement learning, the neural network will learn from previously played battles.



To measure the performance of our neural network, we played on the online platform Pokemon Showdown against human players. Attracting over 100 000 players a day, we need to evaluate the performance of a neural network against human players and develop a practice tool for players.

Related Work

In this section, we describe an AI that plays Pokemon from the paper *Optimal Battle Strategy in Pokemon using Reinforcement Learning*[1]

While the authors of this paper shared our view of reducing the opposing Pokemon's health to zero, they also sought to "maximally retain[ing] the health of their Pokemon." To achieve this, they decided to use a model-free approach Q-learning, in which Q-values are assigned to each of the four moves. The AI has a 90% chance of selecting the move with the highest Q-value and a 10% chance of selecting a random move. Q-values were adjusted by rewarding the moves that resulted in wins or having greater health than the opponent and punishing moves that resulted in losses or having lower health than the opponent. With this Q-Learning model, the authors won 60% of matches. We were inspired by this work when we created labels that represent the optimal move. For instance, if we lost, we would reduce the used move's **F-value** (measures favourability to use the move) and increase the F-value of the other three moves by random amounts.

Models

We developed a baseline model that greedily selects the maximum damage move against a specific Pokemon. Since the greedy choice is not always globally optimal, we built the work-smart model to train on the battle data and learn from wins and losses.

Our final work-smart neural network model architecture consists of 3 layers: input, hidden, and output. By normalizing and one-hot encoding the features, we need 43 neurons in our input layer. The input data is processed by 20 neurons in the hidden layer. The output layer has four neurons that correspond to the favourability of using a particular move. The model plays the move with the greatest F-value.

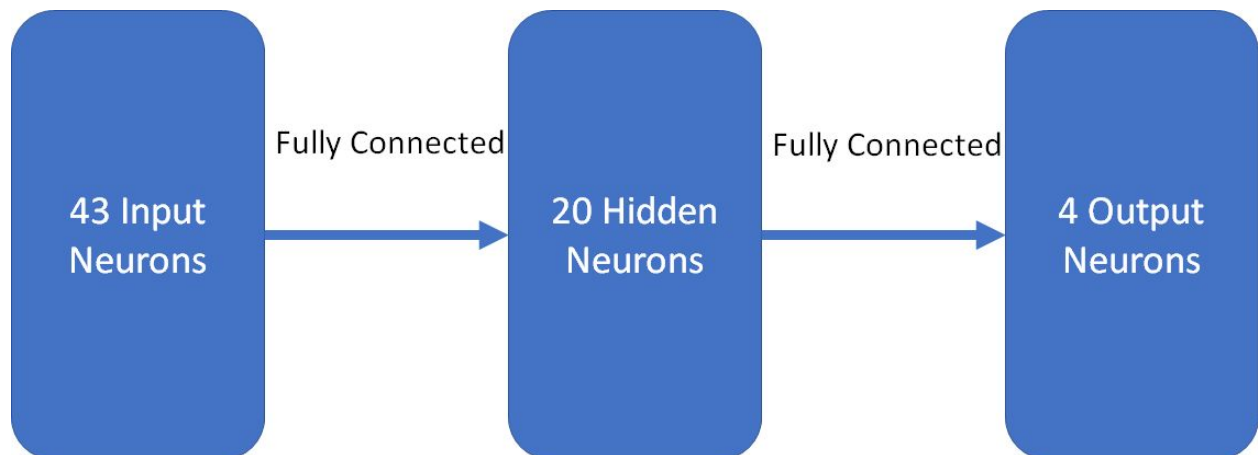


Figure 2: Neural Network Architecture

The neural network was trained with the following hyperparameters:

- Optimizer: Adam Optimizer
- Loss function: BCEwithLogits
- Batch-size: 995
- Learning Rate: 0.1
- Epochs: 50

The work-smart model has four new hyperparameters, α , β , γ , and Δ , further described in the data section. In this process we rewarded victories by increasing the maximum F-value move by a factor α and increasing/decreasing the F-value of the other moves by a factor β ; we punished losses by decreasing the maximum F-value move by a factor γ and increasing the F-values of other moves by a factor Δ .

For our smart model we used the following hyperparameters:

- $\alpha = 1.15$
- $\beta = 0.85$
- $\gamma = 0.85$
- $\Delta = 1.15$

Illustration/Figure

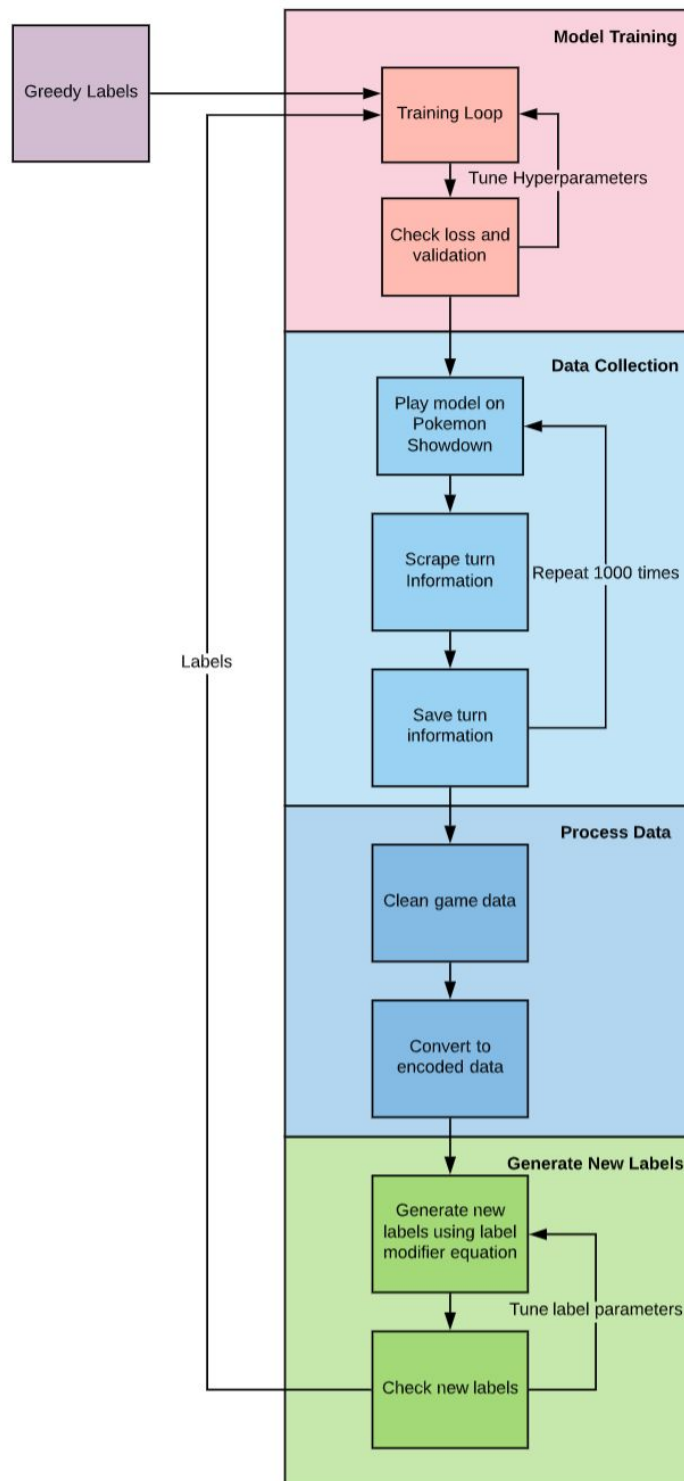


Figure 3: Project Diagram

Initially, the baseline is trained to select the maximum damage move. This model played on the Pokemon Showdown servers against human players to collect data about the opponent, the moves used and the result win/loss. The game data was then processed to generate new labels for the model's iterative training.

Data

We used one dataset for the baseline and another for the work-smart model. Both models received inputs consisting of stats (numerical data) and types (categorical data) by parsing a text file. Below is an entry for the Pokemon *Bulbasaur*:

```
bulbasaur: {
  num: 1,
  species: "Bulbasaur",
  types: ["Grass", "Poison"],
  genderRatio: {M: 0.875, F: 0.125},
  baseStats: {hp: 45, atk: 49, def: 49, spa: 65, spd: 65, spe: 45},
  abilities: {0: "Overgrow", H: "Chlorophyll"},
  heightm: 0.7,
  weightkg: 6.9,
  color: "Green",
  evos: ["ivysaur"],
  eggGroups: ["Monster", "Grass"],
},|
```

Figure 4: Information about Bulbasaur.

Stats are numerical data that describe the battling capability of the Pokemon. For instance, the attack stat correlates with how well it deals damage, while the defense stat indicates its resistance to damage. As numerical data, it was normalized with the following equation:

$$z = \frac{X - \mu}{\sigma}$$

Type effectiveness is another important concept in Pokemon, analogous to the hierarchy in rock-paper-scissors. For example, a fire-type Pokemon can easily defeat grass-type Pokemon. As categorical data, types are one-hot encoded.

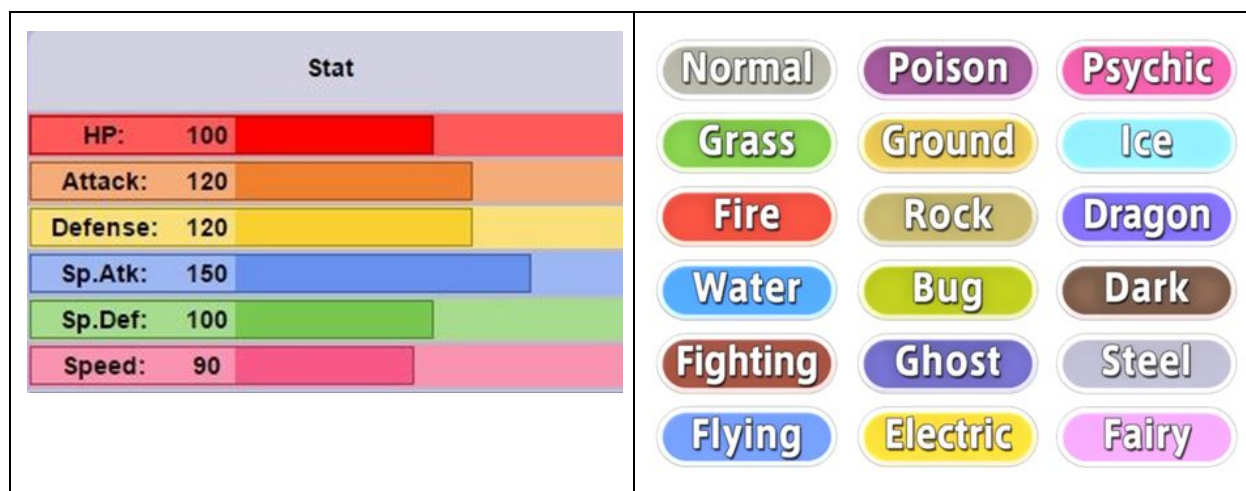


Figure 5: Stats

Figure 6: Types

Together, the numerical and categorical data feeds 43 inputs to the neural network. The label is a one-hot encoded 4-dimensional vector corresponding to the moves: Outrage, Rock Slide, Iron Tail, and Superpower.

For the baseline, we calculated the maximum damage move and assigned it the label of 1, assigning all other moves a label of 0.

For the “work-smart” model, the features are identical to the baseline, however, the label is modified with battle data. The battle data was cleaned to remove incomplete games and duplicate moves. After cleaning, the battle data consists of the opponent’s Pokemon, the move used, and the result of the battle.

Tapu Fini	Iron Tail	loss
Metagross	Superpower	loss
Dedenne	Iron Tail	win
Lopunny	Superpower	win
Dedenne	Iron Tail	loss
Tyranitar	Superpower	win
Greninja	Superpower	win
Kommo-o	Outrage	win
Dedenne	Iron Tail	win
Alakazam	Outrage	win
Serperior	Outrage	win
Alakazam	Outrage	win
Victini	Outrage	loss
Serperior	Outrage	win
Kommo-o	Outrage	win
Blaziken	Outrage	loss
Meloetta	Outrage	loss
Porygon2	Superpower	loss
Zygarde	Outrage	loss

Figure 7: Cleaned Battle Data

If we won, we rewarded the choice using the equation:

$$label = move * \alpha + move' * \beta + move' * r$$

Where:

label - 4-dimensional vector consisting of the F-values of each move

move - 4-dimensional vector with best move initialized as 1 and 0 otherwise

move' - 4-dimensional vector with a 0 for the current best move and 0.09 otherwise

α, β - scalar hyperparameters

r - 4-dimensional vector with random values between 0 and 0.01

If we lost, we punished the choice using the equation:

$$label = move * \gamma + move' * \Delta + move' * r$$

Where:

γ, Δ - hyperparameters

Results

As described in our introduction, we aimed to develop a neural network to surpass a 50% win rate.

However, we would also like to understand why our neural network won or lost a battle. We cannot use ideas such as sensitivity, precision, or confusion matrices because our labels are an estimate of the best move. Instead, we examine relationships in battle data. Since battles are won or lost based on the move, it is imperative to know its frequency with respect to the number of wins/losses and the frequency of the opponent's Pokemon.

In our project, we deployed four work-smart models that had different hyperparameters for the labelling equation.

Table 1: Selection of Hyperparameters for Four Work-Smart Models.

Model Name	α	β	γ	Δ
Model1	1.1	1.01	0.95	1.05
Model2	1.2	1.01	0.9	1.1
Model3	1.1	0.9	0.9	1.1
Model4	1.15	0.85	0.85	1.15

We obtained a holistic measure of performance by looking at the number of wins and losses for each model in Table 2.

Table 2: Win/Loss Information for Each Model.

Model Type	Model Name	Win	Loss	Win Rate
Baseline	Baseline	529	534	0.498
Work-smart	Model1	41	77	0.347
	Model2	54	92	0.370
	Model3	73	99	0.424
	Model4	70	85	0.452

Concentrating on the analysis of our most successful models: baseline and Model4, we compare the number of times each move is used based on the battle data collected by the baseline.

Table 3: Number of Times a Move is Used by the Baseline and Model4.

Move	# of times used by baseline	# of times used by Model4
Outrage	484	329
Superpower	383	356
Iron Tail	99	101
Rock Slide	40	100

Model4 uses a much evenner distribution of moves after training on new labels.

Depending on the distribution of opponent Pokemon, we expect a corresponding optimal distribution of moves. In the next table, we observe the relationship between the opponent's Pokemon with the number of wins/losses.

Table 4: Top 6 Frequently-Appearing Pokemon and Outcome of Match for Baseline.

In this dataset, the baseline has battled 203 different Pokemon.

Pokemon	Move Used	# of wins	# of losses
Gyrados	Outrage	7	35
Greninja	Superpower	19	12
Charizard	Rock Slide	15	12
Donphan	Outrage	2	21
Slaking	Superpower	1	20
Aegislash	Outrage	1	18

Model4 encountered a different set of frequent Pokemon in their dataset. But we would like to highlight the different moves chosen by each model.

Table 5: 6 Most Frequently-Appearing Pokemon and Corresponding Move Used.

Pokemon	Move used by Baseline	Model used by Model4
Gyrados	Outrage	Superpower
Greninja	Superpower	Superpower
Charizard	Rock Slide	Rock Slide
Donphan	Outrage	Outrage
Slaking	Superpower	Superpower
Aegislash	Outrage	Superpower

Discussion and Learnings

We were surprised that the baseline had the highest win rate of 49.8% and was closely followed by Model4 at 45.2%. The work-smart model was unable to learn effectively from the baseline battle data with our approach to achieve the desired 50% win rate.

From table 5, Model4 has some changes in the move label compared to the baseline. The change from using the move *Outrage* to *Superpower* against the Pokemon *Gyarados* was an improvement since it can change its type to become weak to *Superpower*. Although changing the move against the Pokemon *Aegislash* from *Outrage* to *Superpower* is unwise because *Aegislash* is unaffected by *Superpower*.

Comparing the results in table 2 with the hyperparameter values in table 1, we find that models with a β value above 1 have poor performance. Since the modifiers for *move'* are always positive, the label switches from the most damaging move to a new move easily, resulting in many suboptimal changes.

Unfortunately, the release of *Pokemon Sword and Shield* broke our script and hindered data collection. It also forced us to play on smaller servers with long wait times, and more experienced players. This was a large setback for us because we were unable to collect a sufficiently large dataset conducive for a higher win rate.

In the future, we would run the models and data collection scripts when Pokemon is not releasing a new update. This way, we play the models in a consistent environment. We would also replace our baseline neural network with a lookup table for data collection.

Ethical Framework

While artificial intelligence has far surpassed humans in games like Go and Starcraft; we lack ethical consideration. The main stakeholders, in this case, are the Pokemon players, developers, and administrators.

We consider reflexive principlism through each stage: data collection, model use and impact in society.

1. **Data Collection:** We infringe on the autonomy of human players by collecting their data without their consent. This process is maleficent as players lose enjoyment in facing a neural network that repeats its strategy[2]. Players may quit Pokemon Showdown while we collect data.
2. **Model use:** Based on online messages, repeated matches with our bot have either frustrated players or left them eager for a rematch. In this way, the neural network exhibits beneficence as a practice tool for players to improve against specific Pokemon. The major downside to releasing this neural network to the public is that someone with malicious intent can deploy numerous bots on the Pokemon Showdown website and ruin the experience for all players. This increased web traffic causes trouble for administrators who will then need to kick the bots.
3. **Impact on Society:** Machine learning in video games has led to improvements in quality of life. For instance, the network behind alphago optimizes power grids to reduce electricity consumption[3]. We imagine the decision-making paradigm in



Pokemon can also extend to larger applications like robotics. In addition, our neural network can help video game developers test new Pokemon and features.

References

[1] A. Kalose, K. Kaya, and A. Kim, "Optimal Battle Strategy in Pokemon using Reinforcement Learning."

[2] A. Barasch, "AI Ethics, Computer With Souls, Self-Playing Games," Variety, 31-Mar-2019.

[Online]. Available:

<https://variety.com/2019/gaming/features/ai-ethics-computer-with-souls-self-playing-games-1203176874/>. [Accessed: 03-Dec-2019].

[3] C. Metz, "Google's AlphaGo Levels Up From Board Games to Power Grids," Wired,

24-May-2017. [Online]. Available:

<https://www.wired.com/2017/05/googles-alphago-levels-board-games-power-grids/>.

[Accessed: 23-Oct-2019].

Permissions

- permission to post video: yes/no OR **wait till see video**
- permission to post final report: **yes**/no
- permission to post source code: **yes**/no