Solving WORDLE using Deep Reinforcement Learning

Anu Kumar, Hardik Patil, Rohan Sequeira, Shreyas Bhat, Sonali Mohanty

EECS 545

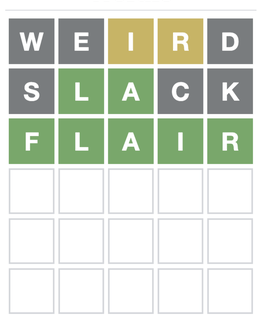
UNIVERSITY OF MICHIGAN

Introduction

In Wordle, the player gets six turns to guess a five-letter word. During each turn:

- Yellow – letter present but incorrect index
- Green – letter present at exact index
- Grey – letter not present in the word

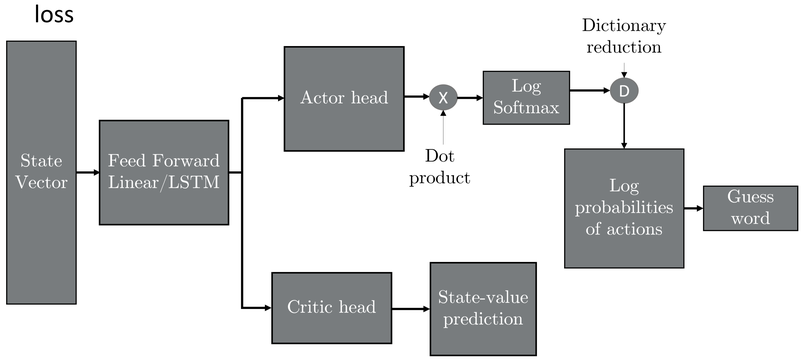
The game is won if the word is guessed within six attempts and lost otherwise.



Mathematically, an optimal winning strategy for Wordle is given by the information theory approach. In this project, we try to implement a deep reinforcement learning model to solve Wordle and achieve performance as close to the information theory approach as possible.

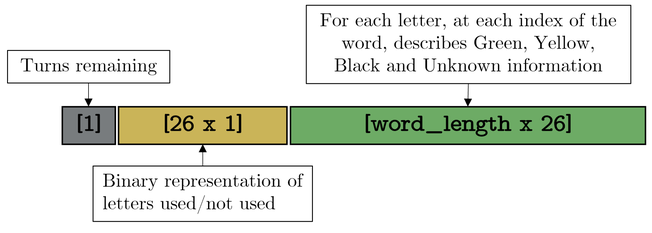
Advantage Actor-Critic Pipeline

- A2C is well suited for Wordle because it combines Value-based and Policy-based methods for *discrete action spaces*
- Actor generates a probability distribution over the action space given the current state
- Critic tries to learn the value function for the model
- Loss function is a weighted sum of the actor loss, critic loss, and entropy loss

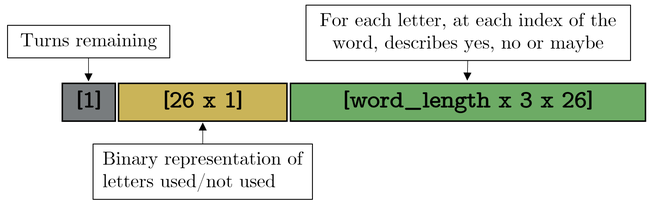


State Description

- **State 1** used for Word-level Wordle model



- **State 2** used for Character-level Wordle model



Reward Function

Reward Description	Reward
Win Reward	+ 30
Step Reward	- 1
Loss Reward	- 50
Green Reward	+ 2
Yellow Reward	+ 1
Grey Reward	+ 0.1
Repeated Grey Letters Reward (character level)	- 3

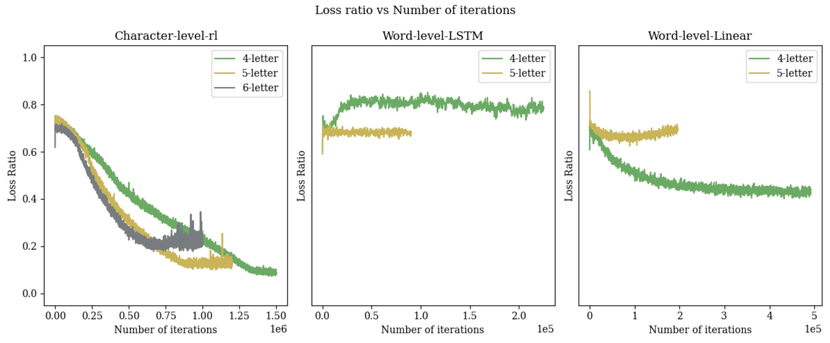
- Reward is handed out only for unique Greens, Yellows and Greys
- Step reward is applied from the 3rd turn onwards
- No reward for guessing off the bat in 1st turn

Model Variations

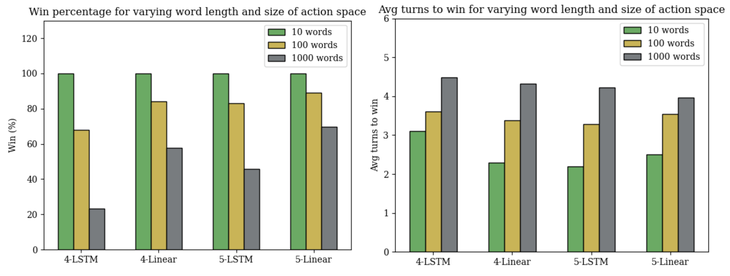
- We created a character-level model for 4, 5 & 6-letter versions of Wordle
- We also created word-level model for 4 & 5-letter versions of Wordle and trained them on linear and LSTM neural networks
- The action space for 4, 5 and 6-letter Wordle was 2348, 13000 (2315 targets) and 6000 (2000 targets) words long, respectively
- Dictionary reduction functionality was introduced to reduce number of possible actions at each step to help train the model better

Results

- For a **character-level model**, we observed that the model doesn't have to choose meaningful words and hence can prune the action space more effectively compared to dictionary reduction
- For a **word-level model**, even after dictionary reduction, the action space could have hundreds or thousands of possible words to choose from, rendering the model ineffective at solving Wordle

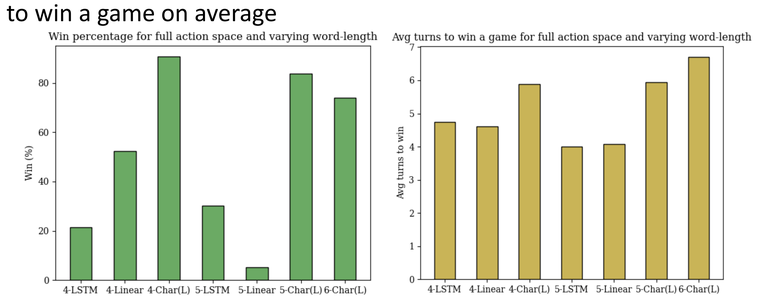


- To investigate the effect of size of action space, we trained our model where only a limited set of words formed the action space
- As the *size of action space increases, performance decreases,* shown in the plots below



Results (contd.)

- We found that the performance deteriorates for all models on the full scale Wordle game (max actions possible)
- The character-level model performs well but requires way too many turns to win a game on average



Example Games

- Character-level model wins more consistently in comparison to word-level model
- Character-level model doesn't always use a valid word in each turn

4 & 5 letter games from A2C Word-level

RIDE	LIVE	EVADE	CROAK
TWIN	THIN	UNMET	REACT
DIED	RILY	AGREE	CRASS
PIED	PILE	ABASE	CROAK
HIED	LIRE	CRATE	
TIED	LIVE	ABASE	
PIED		AWAKE	

4, 5 & 6 letter games from A2C Character-level model

LIVE	RIDE	EVADE	CROAK	OPTION	AGREED
LARE	LARE	SAITE	SAITE	LASIET	LASIET
NISE	DISE	CRADE	CRANL	MUTINY	GRADER
LIFE	RINE	GLADE	CROAD	NOTICN	ANARED
LIME	RIDE	OVADE	CROAK	ENTION	AORKED
LIVE		EVADE		DITION	AGREED
				OCTION	
				APTION	

Conclusion & Discussion

- With longer training periods the win rate increases for character level model but so does the average turns to win the game
- Character level model cheats it way through – at each turn, it doesn't necessarily use an actual word from the dictionary
- The word-level model suffers due to HUGE action space (dictionary reduction doesn't help much)
- Models do not learn as they get stuck in local minimums
- Word-level model also suffers from vanishing gradients problem
- Performance is better when the action space is small (<1000)
- All models perform poorly when target word rhymes with possible words from the dictionary

Future Steps

- Sequentially train the model on increasing size of action space
- Retraining on words that the model fails to learn
- Hyperparameter tuning (Discount factors, weights for actor/critic loss, batch size, hidden layers and hidden sizes)
- Improving reward function by incorporating a penalty for repeating words (word-level) and trying invalid words (Character-level)