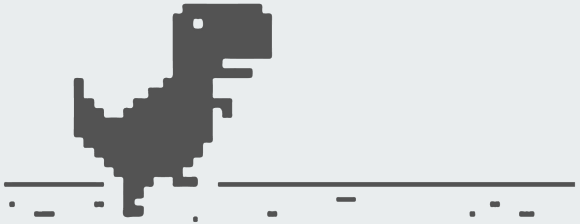
Saving the Dinosaurs with Reinforcement Learning

Austin Poor



Goal

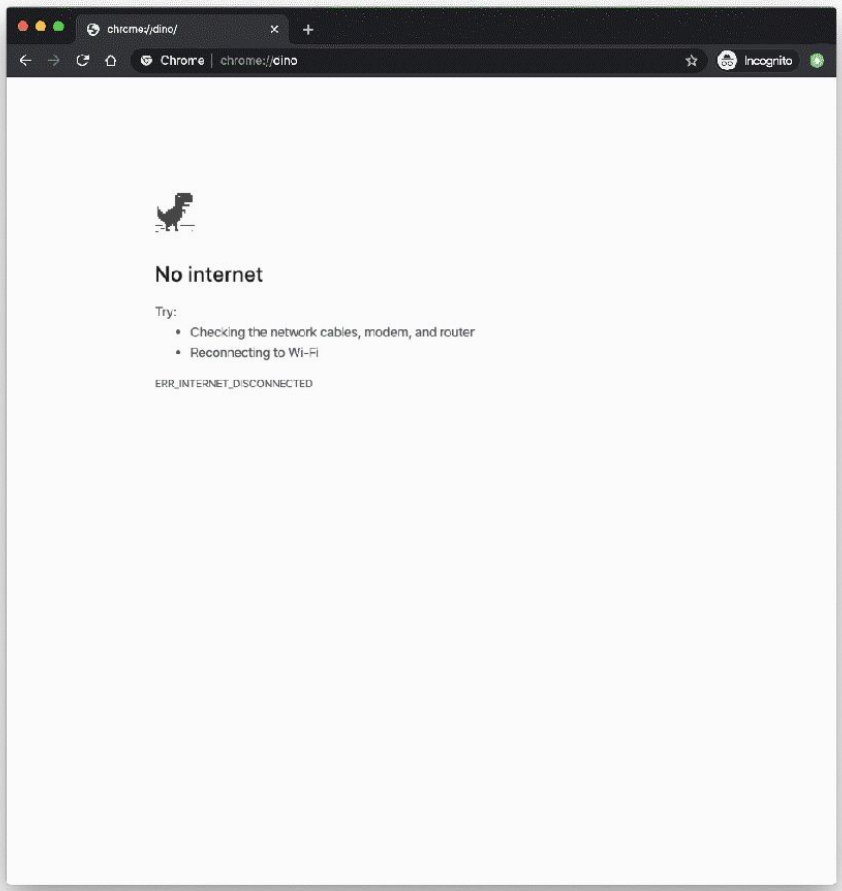
Chrome Dino Game

Chrome's "No internet" page has a hidden game.

The dino game is an **infinite runner** where the dinosaur dodges cacti and pterodactyls.

The player can **jump** with the 🔼(*up arrow*) and **duck** with the 🔽(*down arrow*).

Can a program be taught to play?

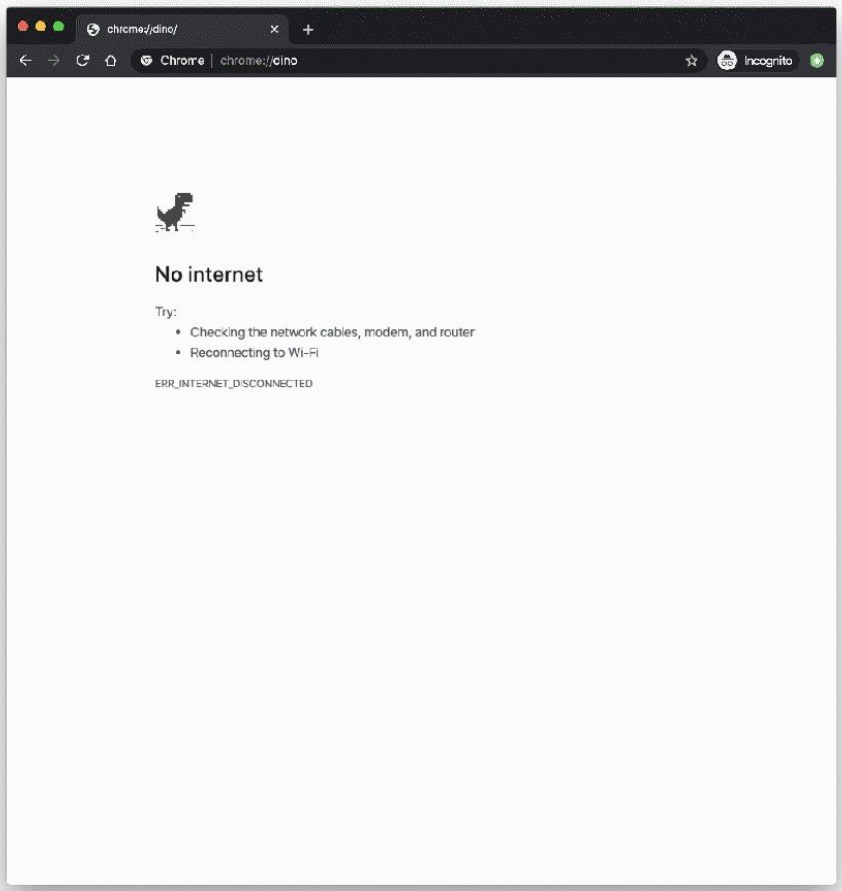


Chrome Dino Game

Multiple possible approaches.

Hand-coded Heuristic approach – I specify the rules for when to perform an action.

ML approach – I train an algorithm to play the game (Reinforcement Learning)



Agents

The Agents

Agents:

- Heuristic Bot
- Reinforcement Learning Bot

Environment: Bots have access to information about the *(X, Y)* coordinates of the dinosaur and the nearest obstacle.

Actions:

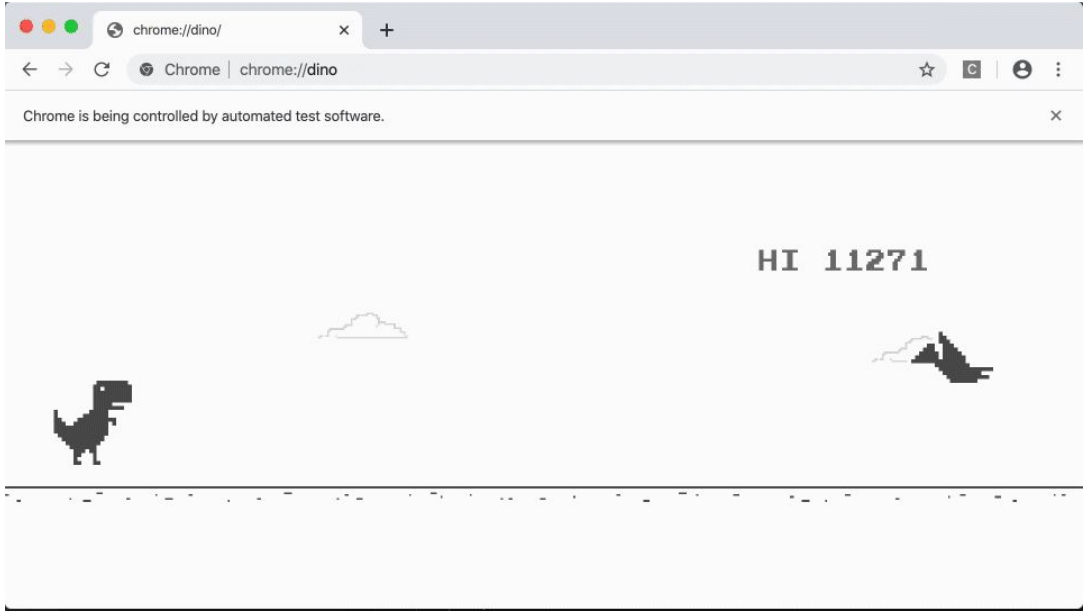
- Jump
- Duck
- Do Nothing

Heuristic Bot

Heuristic Bot

Heuristic bot used a hand-coded set of rules.

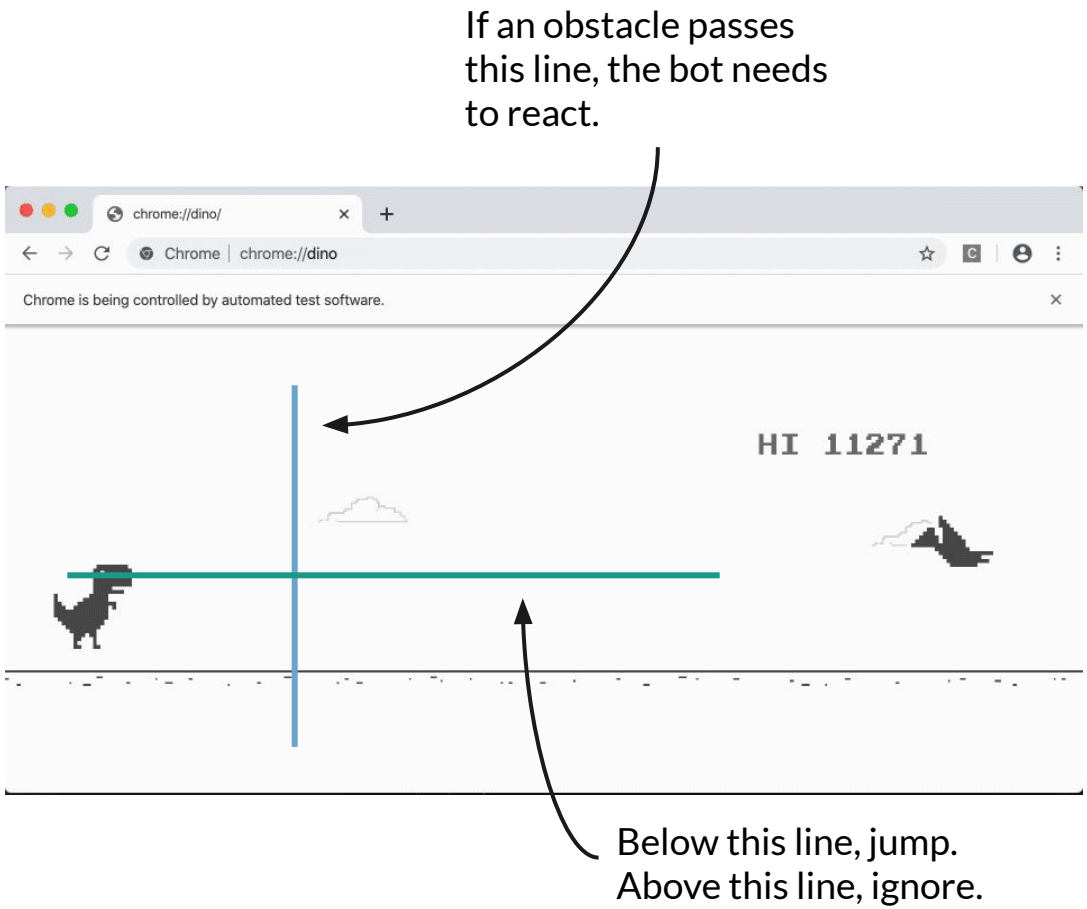
Uses thresholds for when to jump or duck.



Heuristic Bot

Heuristic bot used a hand-coded set of rules.

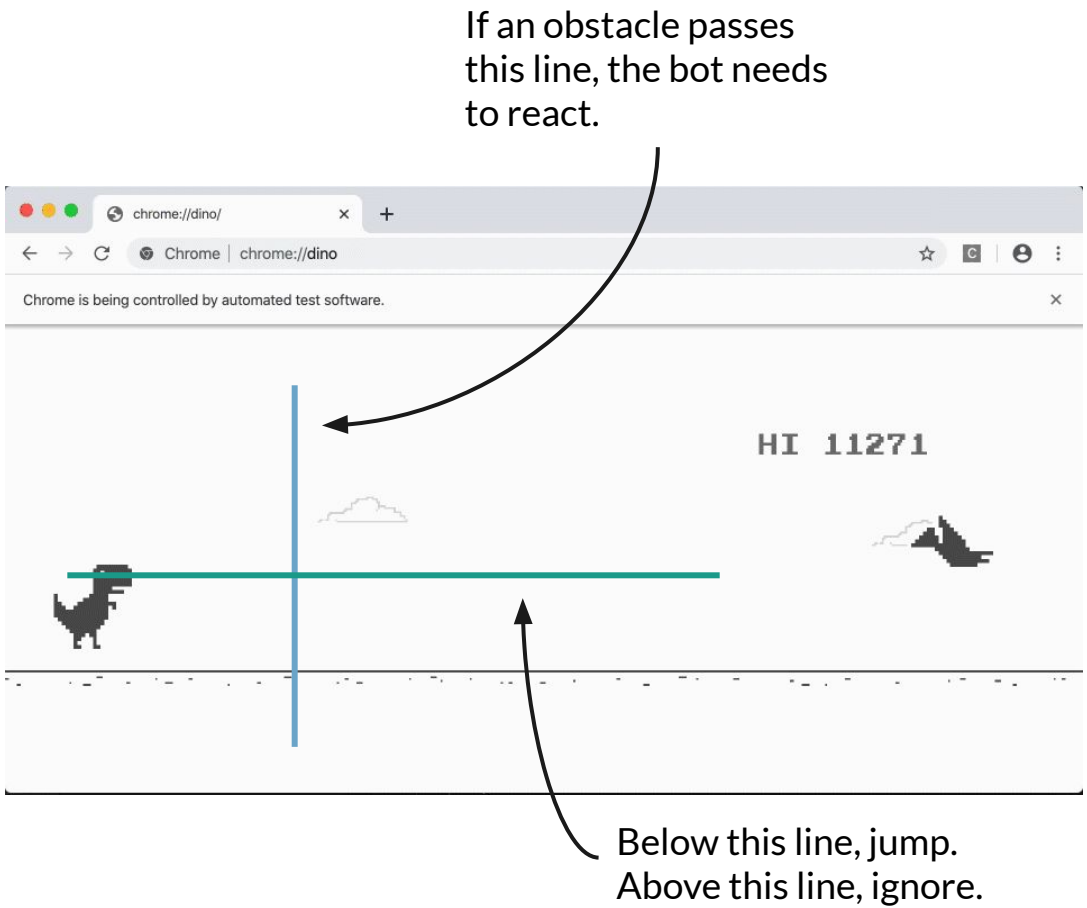
Uses thresholds for when to jump or duck.



Heuristic Bot: Results

High Score: 16,000

(For Comparison) I am able to get a high score of about 2,500

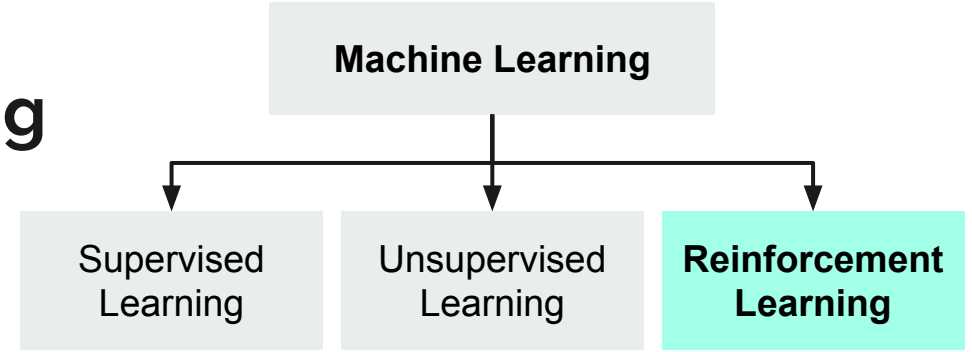


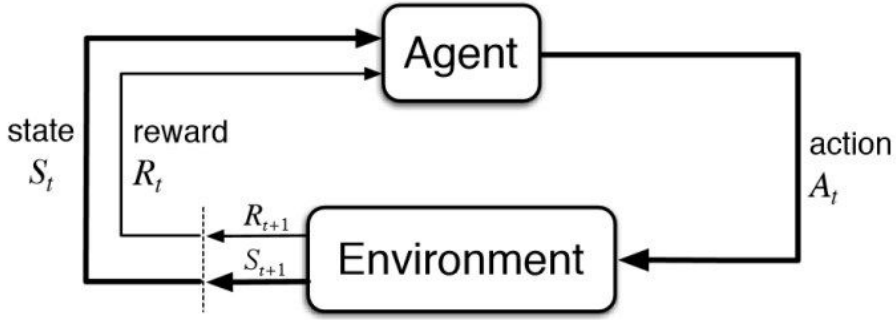
Reinforcement Learning Bot

Reinforcement Learning

Reinforcement learning is the 3rd branch of machine learning.

It involves teaching an *Agent* to choose the best *Policy* for picking an *Action* given its *Environment*, based on the *Positive* and *Negative Rewards* it receives.



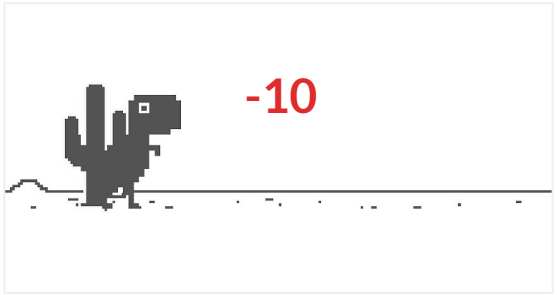


Reinforcement Learning Bot

Using Deep Q-Learning model

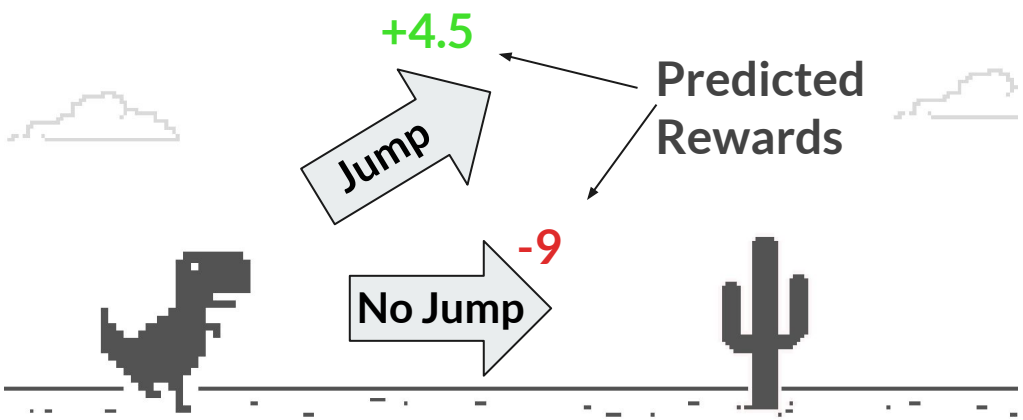
Environment Rewards:

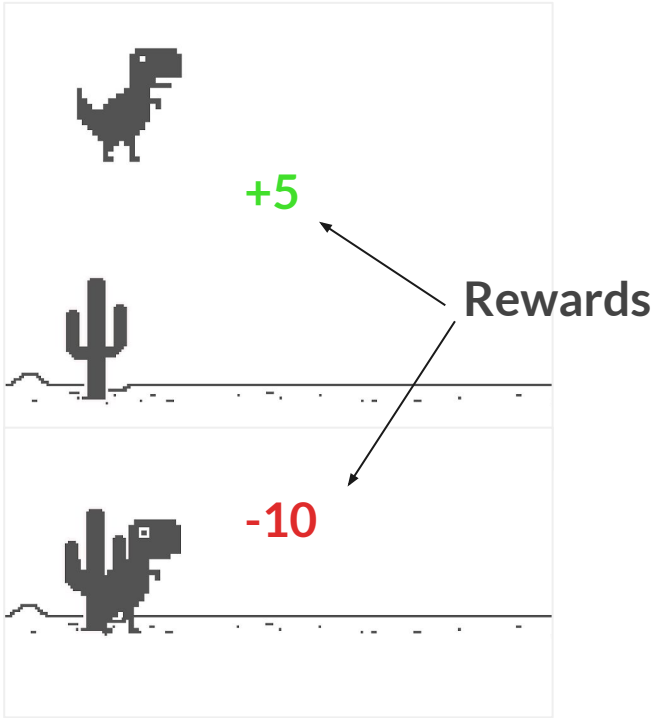
- ***Passing an Obstacle*: +5**
- ***Crashing*: -10**



RL Bot: Future Rewards

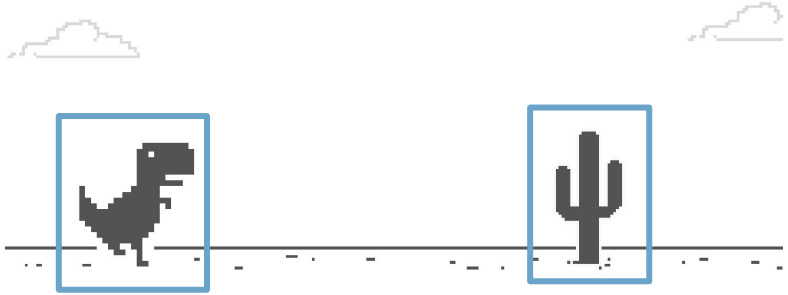
Discount factor of 0.9 for future rewards

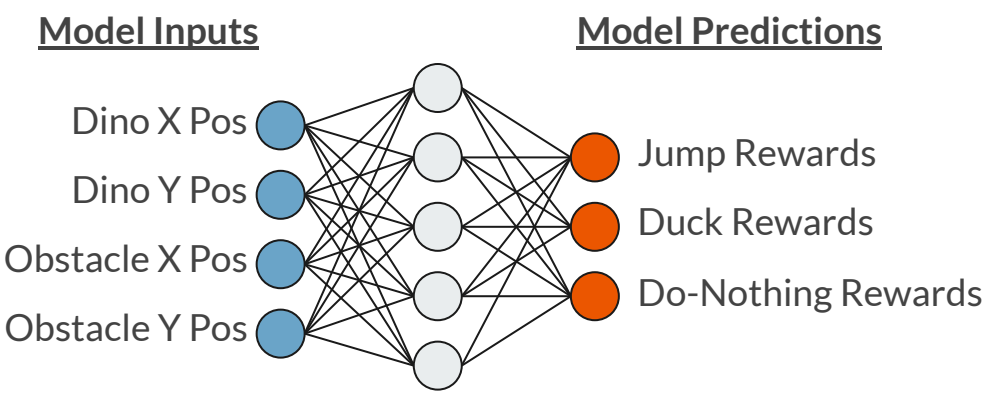




RL Bot: Model Structure

RL model sees the current environment and tries to predict the reward for each possible action.





RL Bot: Actions

Move in the direction of maximum predicted reward

RL Bot: Results

RL Bot was only able to get a high score of about 1,300

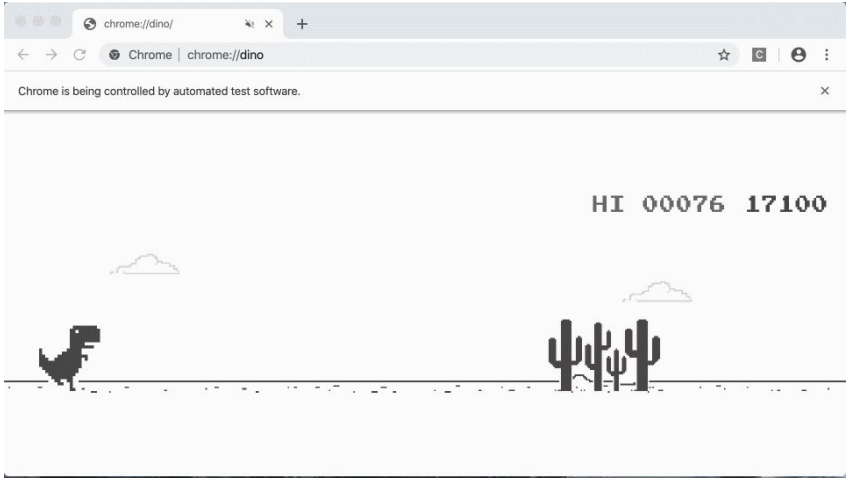
But with a lot of variance in the scores.

RL Bot had a hard time differentiating between cacti and pterodactyls.

Conclusions

Conclusions

- The Heuristic Bot did a good job of getting far with a simple set of rules
- The RL Bot wasn't able to do as well in the environment
- Game is mostly deterministic and not enough nuance to make the DeepQ model better suited
- DeepQ approach would likely require much more training (hard to run in parallel due to the time and memory overhead of playing the game with Selenium)



Thank you

Austin Poor

 /a-poor

 /austinpoor

 /@apoor