Fuzzy Logic

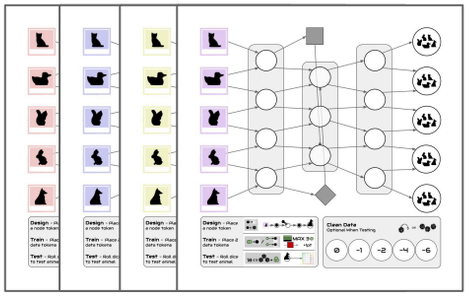
Build a cardboard machine learning model

Designer	Artist	Core Playtesters
Michael Novack	Flaticon.com >> Freepick & Roundicon Premium	Feifei Novack Brian Kemp Mike Krantz Karen Kemp

Introduction

You are a data scientist part of a startup creating a service for identifying cute animals in photos. This is becoming a hot market so you will need to compete against the other startups in this space. Whoever gets to market first will have an advantage, but at the end of the day the best model will win.

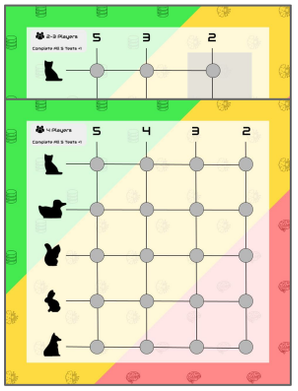
Components



4 player boards

20 animal figures

1 Score Board



100 data tokens

44 node tokens

15 overitting tokens

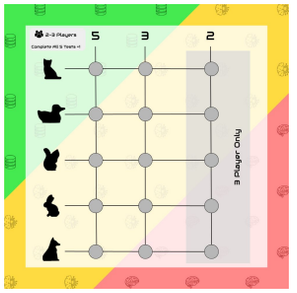
3 dice

1st player marker

4 data cleaning tokens

Set Up

Center of Table Set Up



x44

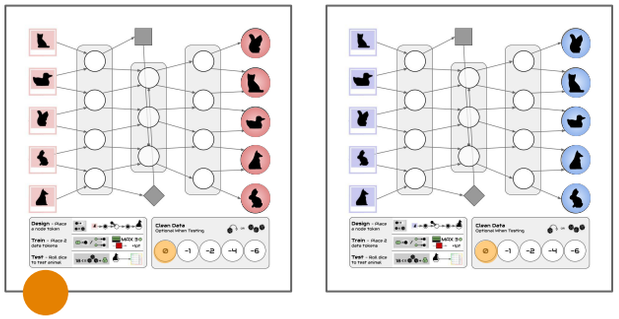
x100

x15

x3

1. Place the scoreboard in the middle of the table. Place the side face up that matches the number of players
2. Place the dice, node tokens, data tokens and overfitting tokens within arms reach of all players.

Player Set Up



1. Each player receives a player board, and animal tokens of a single color.
2. Each player places a data cleaning token at the 0 place on the tracker.
3. Give a player the 1st player marker with a method of your choice.
4. The first player places all of their animal tokens on the their board on the right most column in any order.
5. All players must set up their player boards to match the first players.

Gameplay Overview

Play starts with the first player and continues in clockwise order.

On a players first turn they will take 3 actions, and then all future turns they must take 1 and only 1 action.

Actions a player can take:

- **Design Model** - Allows you to attempt a test for an animal
- **Train Model** - Increases your probability of a successful test
- **Test Model** - Test a specific animal to score points

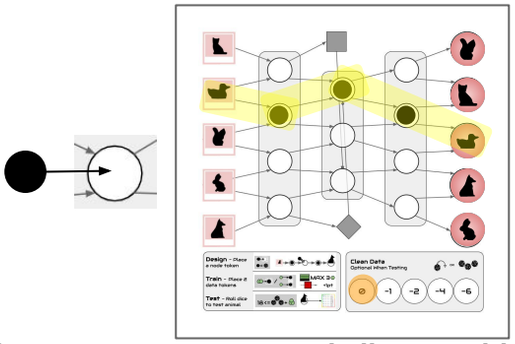
Player triggers the end game by either:

1. Successfully passing tests for all 5 animal tokens.
2. If all players have no possible way to perform any tests. This is only possible if players do too much overfitting which is explained later.

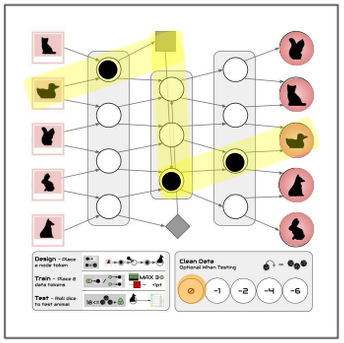
When the end game is triggered play continues clockwise until all players have had an even number of turns.

Design Model

Add a node token on any empty node on your player board. A node token cannot be removed or transferred to another node once it is placed.



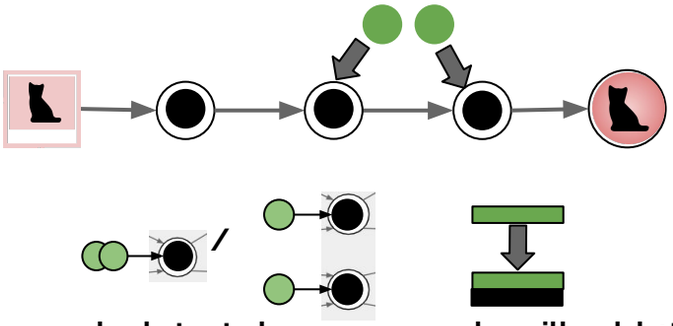
The grey square and diamond in the middle of the board are not nodes. They allow you to move from the bottom of the board to the board and visa versa.



Train Model

Train a model by adding data to a model, so you increase your odds of a successful test. Add 2 data tokens on node(s) on your player board.

- The node must have a node token on it to receive any data tokens
- Data tokens cannot be removed or transferred after they are placed.
- Both data tokens may be placed on the same node or on 2 different nodes.

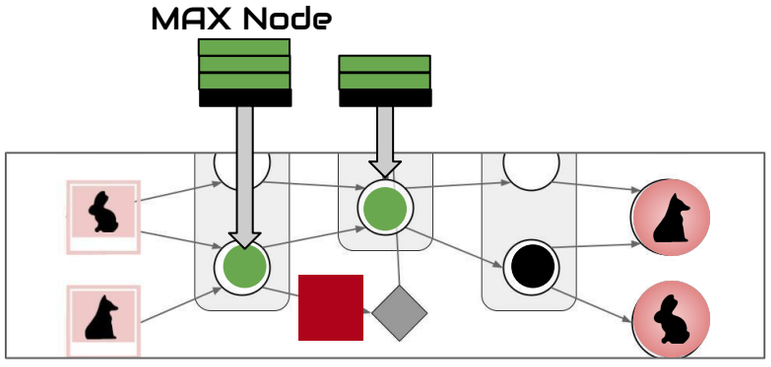


For each data token on a node will add +1 when calculating a successful test.

Training Model - Continue

Max 3 data tokens per node. This means you have a highly trained node, but there might be an overfitting problem. Overfitting is when your model is too complex or data is too heavily biased to give accurate responses.

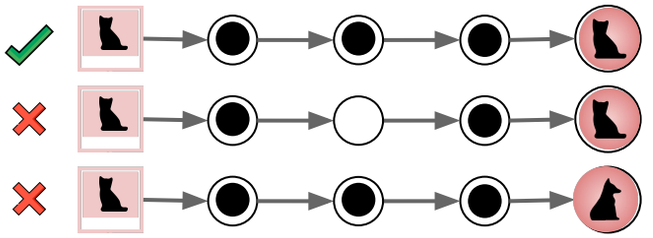
- You get +1 point for each maxed out node for having a highly trained model
- Place an overfitted token on one of the connections of your choice leading to the next node. This makes it so this connection path cannot be used for testing for the rest of the game.
- If a path is no longer needed because you have already successfully tested a specific animal there is no downside of blocking off the path.
- It is possible to block off all paths that can lead to testing a specific animal, so be careful when maxing out nodes.



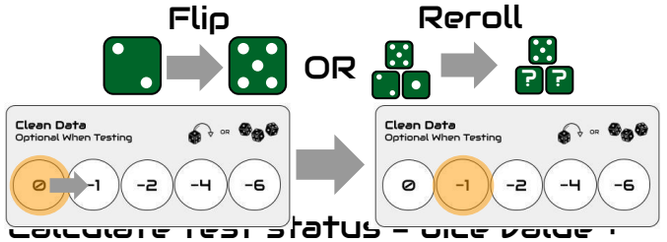
The path to the grey diamond cannot be used for the rest of the game.

Test Model

1. Identify a path to test, which are 3 nodes with node tokens that connect an animal on the left of the player board with the matching animal token.



2. Roll the 3 dice.
3. You may use the clean data action to modify the value of your dice:
   a. Either reroll any number of dice or flip a single die (1->6, 2 ->5, 3->4).
   b. Then slide the clean data token over 1 space on your player board.
   c. You can use the action at most 4 times per game.
   d. Each time you use the clean data action it creates negative points at the end of the game, as you should have used clean data before testing.



4. Calculate test status = dice value + # of data tokens on the testing path

5. Evaluate if test passed. Details are on next page.

Test Model – Continue

Failed test = test status 17 or less : The players' turn is over and play continues clockwise.

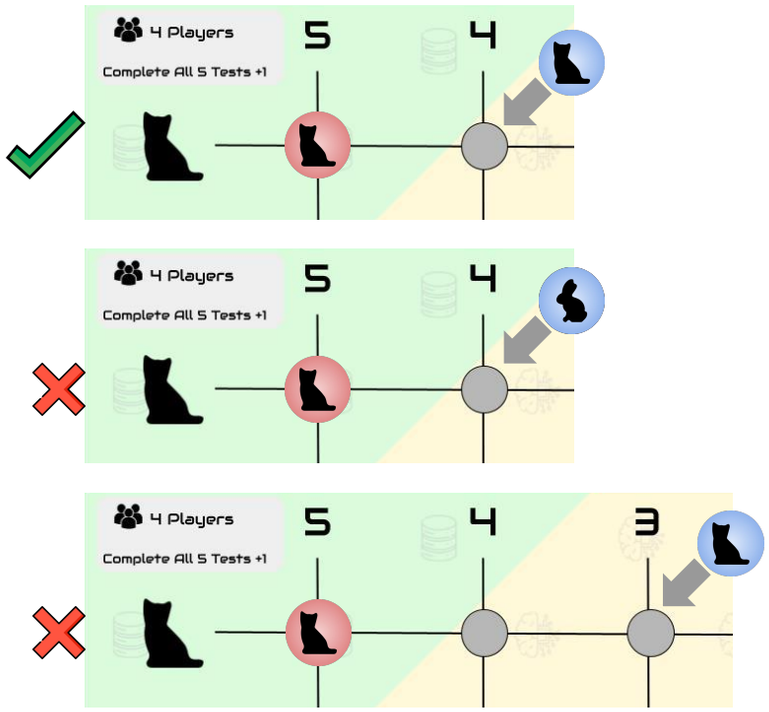
Successful test = test status 18+

15 + 4 = 19 ✓

14 + 4 = 18 ✓

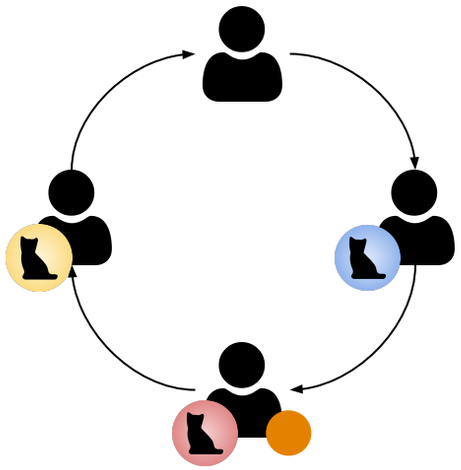
14 + 3 = 17 ✗

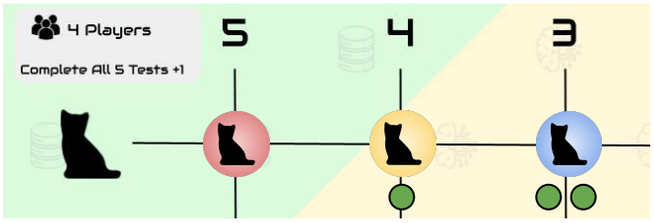
Place the animal token on the score board that was tested on the highest available point value on the score board for that animal.



All players that pass a given animal in a round receive the same number of points.

- A round starts with the first player.
- Add data tokens next to player(s) animal figures so that all players have the same point value.
- Data tokens on the scoreboard count as 1 point at the end of the game.



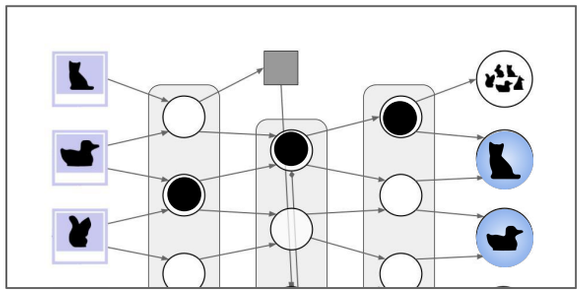


As 3 of the players that successful tested the cat in this round each get 5 points.

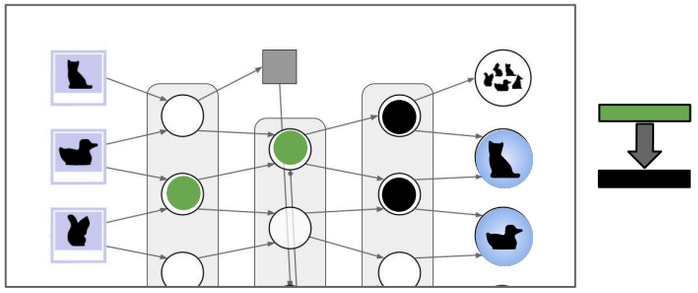
Game Play Example

Alex (blue player) and Veronica (yellow player are playing a 2 player game. Alex wants to test the duck photo. He will spend 3 turns doing this.

Turn 1 (Design Model) - He needs to spend a turn completing the connection path to the duck.



Turn 2 (Train Model) - He could do a test now as there is a valid path for the duck photo, but the chances of success is low. He decides to train his model to prepare, by increasing two nodes by one.

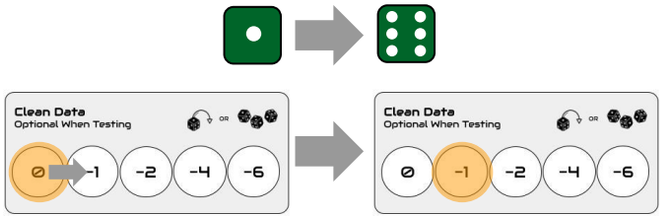


Turn 3 (Test Model) - Rolls the 3 die to test for the duck photo.

13 + 2 = 15

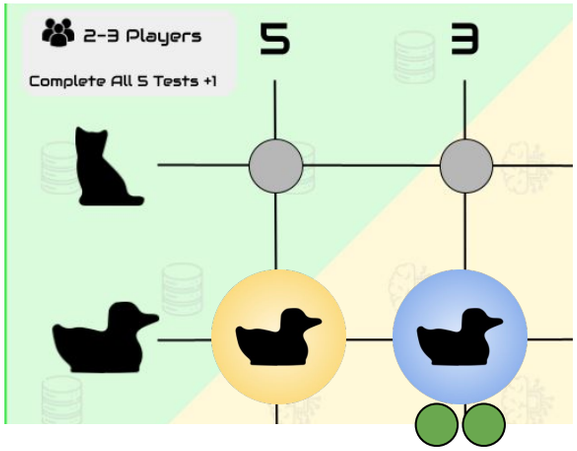
If Alex does not do anything the test would fail as the test status is below 18

He uses the data clean action to flip a die. He must slide the clean data token over 1 space. Alex will receive a penalty at the end of the game for using the clean data action.



17 + 2 = 19

Now the test status is 18+, so he has successfully passed the test. Veronica is the first player and also successfully passed the duck photo test this round. This means they will both receive the same number of points. Veronica (5pts) is in first place on the scoreboard. Alex is in second place (3pts) and places 2 data tokens on the score board next to his animal token to indicate 2 additional points at the end of the game to give him a total of 5 points.



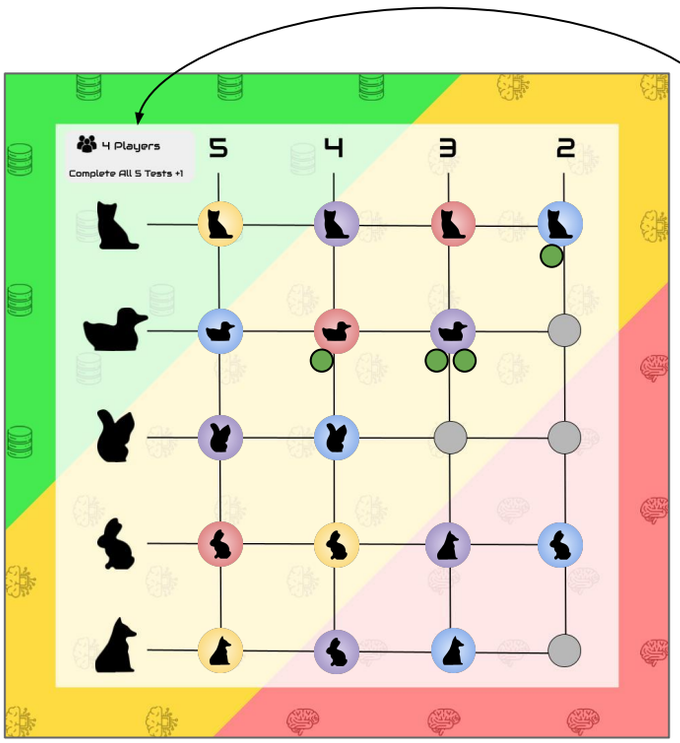
End Game Scoring

The highest score wins, below are the scoring conditions:

1. Add the points from the scoreboard, including any data tokens.
2. Any players that have completed all 5 tests receive +1 point.
3. Each maxed node is +1 point.
4. Subtract the points the clean data token is over.

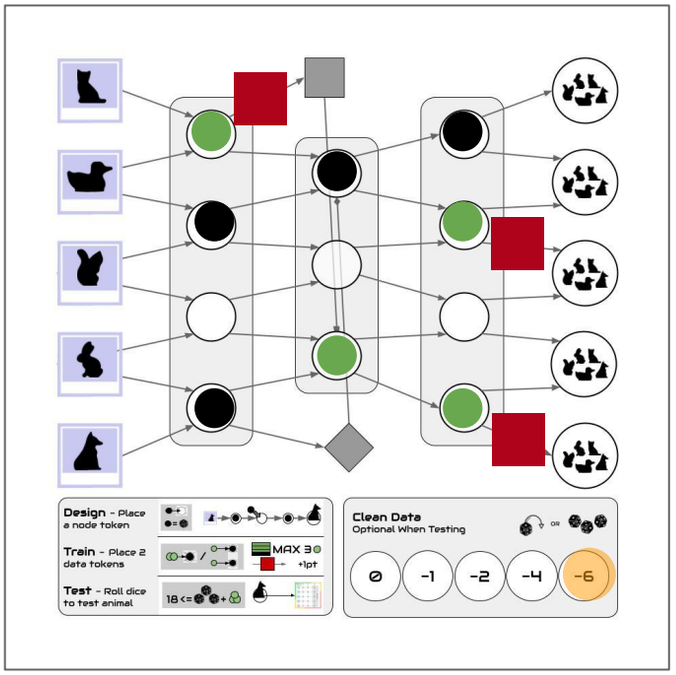
If there is a tie they start with the first scoring condition and see if there is a winner. Continue down the list until a winner is found. If there is still a tie after evaluating each scoring condition separately then share the win.

Example scoring we will use the blue players set up. The player total score is 15 points.



+1
+3
+5
+4
+2
+3

18



+1
+1
+1
-6

-3

Introduction

This section is not part of the rules. It is additional information that explains more about how the game relates to the real world of AI development.

Let's start with some definitions:

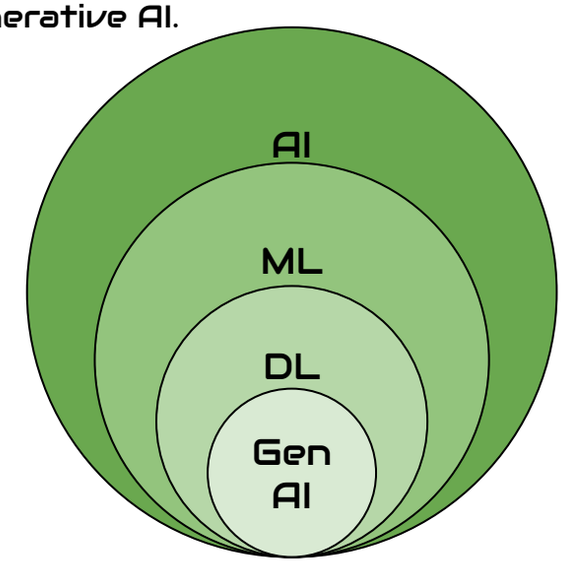
AI is broadly defined as the ability of machines to mimic human behavior. It can be rule-based, statistical, or involve machine learning algorithms.

Machines learning , is a subset of AI, and uses data instead of explicit programming to create solutions. Machine learning algorithms leverage statistical techniques to automatically detect patterns and make predictions or decisions based on historical data that they are trained on. A major drawback of ML is that humans need to manually figure out relevant features for the data based on business knowledge and some statistical analysis.

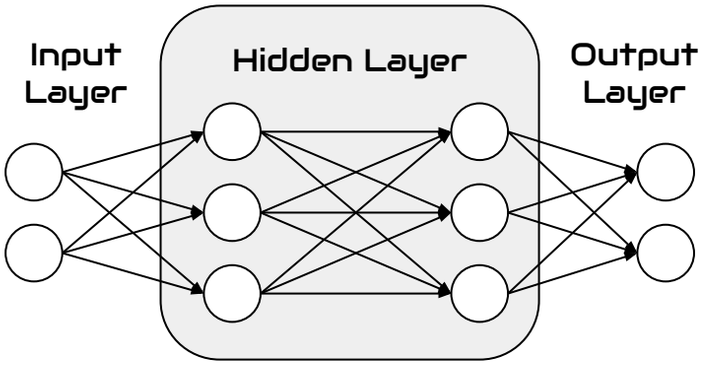
Deep learning is defined as a machine learning technique that teaches the computer to learn from the data that is inspired by humans. DL utilizes deep neural networks with multiple layers to learn hierarchical representations of data. It automatically extracts relevant features and eliminates manual feature engineering.

Generative AI , a branch of artificial intelligence and a subset of Deep Learning, focuses on creating models capable of generating new content that resemble existing data.

People are often interchangeably use these terms, but it is better to think of these as subsets of each other AI >> Machine learning >> Deep learning >> Generative AI.



In Fuzzy Logic you are creating a neural network, which is a type of deep learning model. Below a depiction of a neural network and looks very similar to the player board.



Keep in mind that machine learning models give probability not deterministic responses, meaning it will never be 100% accurate. The accuracy of responses depends on the amount and quality of data it is provided.