Generative Grading
CS398
Checkout this assignment on code.org
The Code.org Dataset

- Students learning nested loops
- 50k students with 1.5 million submissions to a curriculum of 8 exercises.
- 800 human labels across 2 of the exercises.
Lesson 10: Artist: For Loops

Instructions:
How would you modify what you’ve learned to draw these squares? They start at 15 pixels long, the largest is 300 pixels long, and each square is 15 pixels larger than the last.

Blocks:
for counter from 15 to 300 count by 15

Run
Can you write an algorithm to give auto-feedback for this assignment?
Edit distance is meaningless, Code is Zifian

Exponential combination of decisions.
Super fat tailed. Everything looks unique
Hard Problem

Brute force solution?

Code Zipf Plot

\[
f(k) = \frac{1}{\sum_n^N (1/n^\alpha)}
\]

1 million unique solutions to programming Linear Regression

WWW 2014
Hard Problem

Brute force solution?

1 million unique solutions to programming Linear Regression
WWW 2014
Instructions

- If there are many moves, focus on the first one.
- Random code strategy is for when the student seems to be trying things randomly.
- Lookout for students who don't get nesting or pre/post conditions. Often extra blocks in a body is an indication they don't get that the post of the loop has to match the precondition.

Question

when run
move forward by 100 pixels
repeat 5 times
do turn right by 90 degrees
move forward by 10 pixels

Label Console

- Strategy:
  - Beeper Boundary (most people do this)
  - Triangle Strategy
  - Recursive Strategy

- Looping:
  - Correct use of looping
  - Doesn't use a while
  - Doesn't have correct stop condition
  - Body is missing statements
  - Body has extra statements
  - Body order is incorrect
  - Sets up initial precondition
  - Does not get nesting
  - Loop post condition doesn't match precondition

- Cleanup:

Record label
public class MySoln {
    public void run() {
        move(50);
        for(int i=0; i<4; i++){
            if(frontIsClear()) {
                turnLeft(90);
            }
            for(int j=0; j<i; i++){
                move(i * 20);
                turnRight(120);
                move(10);
            }
        }
    }
}
Label student code

Traditional Deep Learning Doesn’t Work

Last Problem (P8)

Feedback F1 Score

Old Gaurd

Humans
Inaccurate, Uninterpretable, and Data Hungry

Label student code

Last Problem (P8)

Old Gaurd
Deep Learning
Humans

Feedback F1 Score

0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0

Piech et al, ICML 2014
Label student code

Last Problem (P8)

Feedback F1 Score

Old
Gaurd

Deep
Learning

Humans
Label student code

Data Hungry

Last Problem (P8)

Feedback F1 Score

0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9

<table>
<thead>
<tr>
<th></th>
<th>Old Gaurd</th>
<th>Deep Learning</th>
<th>Humans</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

We need one shot learning

We need verifiability
We need a strategy! How are we going to solve this problem???
[Suspense]
Taste of the future of AI
Humans Don’t Need Much Data

Single training example:

Test set:
Bayesian Program Learning
Lake et al. Human-level concept learning through probabilistic program induction
Imagine Students

- Struggle with double for loops
- Confuses logic for deleting bricks
Imagine Students

• Struggle with double for loops
• Confuses logic for deleting bricks

This is easy and exponential

A student’s “ability”

θ \sim \text{pythonSample}

A student’s “choices”

c \sim \text{pythonSample}|\theta

The resulting code

\Pi \sim \text{pythonSample}|c

This is hard and linear

\Pr(\theta, c|\Pi)

Infer ability and choices from code
Generative Understanding

Last Problem (P8)

- Struggle with double for loops
- Confuses logic for deleting bricks
Generative Understanding

Label student code

Last Problem (P8)

Feedback F1 Score

- Old Guard
- Deep Learning
- Zero Shot Learning
- Humans

- Struggle with double for loops
- Confuses logic for deleting bricks
In simple terms: generate a ton of our own labelled examples. It’s an unreasonably good way to start to hit high performance
How many decisions with 3 options to get to 10K labelled programs?
How do I write said grammars??
ideaToText
Teachers Articulate N misconceptions

1. This is code for a single decision point

2. Give a name to the choice that the student is making

3. How do those choices translate into grades?

4. What does the code look like? Often evokes other decision points
Starter Code in Blocks

```plaintext
For(???, ???, ???) {
}
```

Starter Code in PsuedoCode

```plaintext
For(???, ???, ???) {
}
```

Solution in Blocks

```plaintext
for counter from ??? to ??? count by ???
```

Solution in PsuedoCode

```plaintext
For(15, 300, 15) {
    Repeat(4) {
        Move(Counter)
        TurnLeft(90)
    }
}
```
First challenge: no solution or yes solution.

Second challenge: draw square using repeat or unrolled
Pro Tips
Keep your grammar clean!

One idea per class
Decisions are not independent
Be clever in your use of outcomes

Template Choices

class MoveLeftCounter(Decision):
def registerChoices(self):
    options = {
        'ML':100 * self.getChoice('ability'),
        'LM':10,
        'M':10,
        'L':10,
        'MLM':10,
        'MLX':10
    }

# Other code...