Lesson Plan

Lesson 4.2 Debugging

Warm-up   (10 mins)
Main Activity   (20 mins)
Wrap up   (10 mins)
Extras   (10 mins)

Prerequisites: In our previous chapters we discussed how models retrain them and update their parameters by automating the process of creating the rules to reach the desired output.

Lesson Overview: Students will understand how to debug AI problems and why debugging is the requirement of every successful software application.

Lesson objective:

   To make students familiar with debugging their code and AI model's output.

Getting started:

1. Gather students together for this activity and announce that they will learn problem solving techniques, also known as debugging today.
2. Remind students about how they learned about automated code.

Debugging:

Debugging in programming is finding problems and designing solutions that give the desired results. There are various methods of debugging code. We'll discuss the basic pipeline of debugging a code which goes under practice.

#1. *Problem definition*:
We undergo debugging when we find that the code we've developed is not giving the desired output. So, the first step in debugging is realizing that we have a problem.

#2. *Gather information*:
This is where you gather data about the expected output for the given problem.

#3. *Select the block of code*:
If the code is large in size, select the code or module where you found the problem.

#4. *Solve the problem*:
Finally, solve the problem and re-check if the problem occurs again. If it does, find another solution until the problem is solved.

Extended Learning:

Show them a mathematical equation on a board as given below:

$4 + \mathbf{5} = 10$

Now, the expected output for this problem is 10. Ask the students to modify the digit in the bold letters to get the desired output.

The correct input to the equation above should be 4 + 6 = 10.

The simple solution to the above equation is:

$$4 + X = 10$$

$$X = 10 - 4$$

$$X = 6$$

This is how we debug the mathematical equation. In the same way when we get an error or we don't get the desired output, it's good practice for the developers to check the underlying code and figure out the problem (bug) and solve it (debug) to get the desired output.