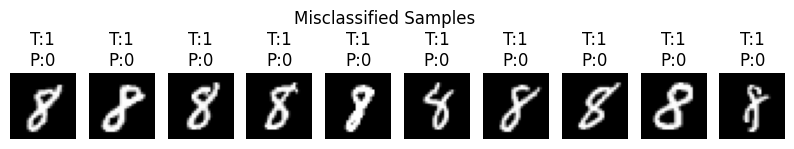
**Put Your Code Results Here:**

**Test Accuracy (is 8 or not): 0.9269**



**Autodiff Trace Table (sample features):**

**Variable Primal (v) Forward Tangent (ẋ) Reverse Adjoint (v̄)**

**0 v1 0.500000 1.000000 2.877583**

**1 v2 2.000000 0.000000 0.500000**

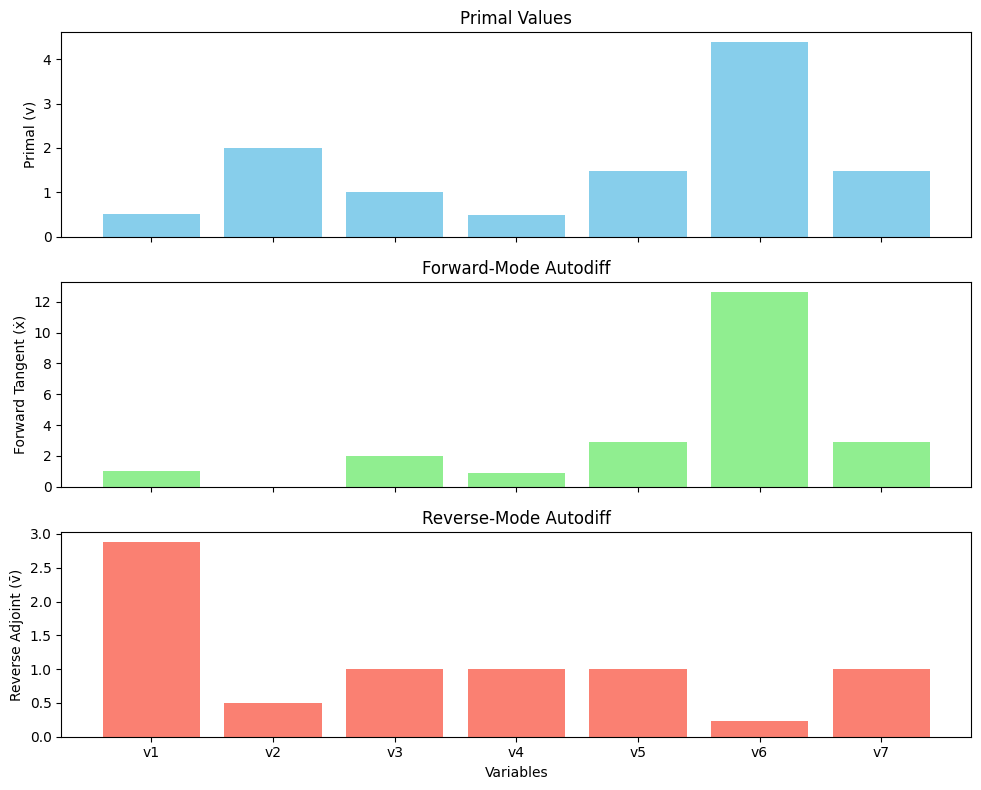
**2 v3 1.000000 2.000000 1.000000**

**3 v4 0.479426 0.877583 1.000000**

**4 v5 1.479426 2.877583 1.000000**

**5 v6 4.390423 12.633804 0.227768**

**6 v7 1.479426 2.877583 1.000000**



**What do these graphs mean?**

These three graphs show how numbers change when we do some math steps, from the input all the way to the final output.

* The first blue graph shows the normal values we get from each step. For example, we multiply, take sine, add, etc., and get values like v1 to v7.
* The second green graph shows da how much the output will change if we slightly change the input x1. This is called forward-mode autodiff.
* The third graph red one. Which shows how much each value (v1 to v7) affects the final output. This is called reverse-mode autodiff, and it’s how deep learning learns by going backward.

**Why do the forward-mode and reverse-mode values look like that?**

The forward-mode values start from x1 and move forward through the steps. So, if one step has a big function like exp() or log(), it can make the numbers grow bigger.

The reverse-mode values go backward. So, the last number (v7) gives its effect to the steps before it. If one variable is used in many steps (like v1), it gets more "blame" or "credit", so its number gets bigger.

**What did you learn from this assignment?**

I learned how computers figure out how much each number matters when making a prediction. It’s like when we do math step-by-step and also check how changes affect the result.

I think I understand how backpropagation works — it’s like going backward through the math to find what caused the result. But I still have to read teacher PPT again when I back to home.

Also, I learned the difference between forward type (goes from input to output) and reverse type (goes from output back to input).