Advisor Meeting 2 — Talking Points Guide

Ryan MacKellar – HNR 499 Senior Project

Theme: Lifting the Hood — From Abstract Math to Real-World Machine Learning

---

1. Begin With the Big Picture

> "The first part of this project was about proving that machine learning is just math — linear algebra and calculus repeatedly applied to data. The second part is about showing that I *own* that math: I'm breaking down the black box of machine learning into its core components, and rebuilding it piece by piece. This approach not only deepens my understanding, it also lets me transition smoothly into a real-world application — identifying motorcycle parts through image classification."

- ML models may look like "magic," but they're really a series of matrix operations and optimization steps.
- My project now focuses on understanding and manually replicating those steps.
- The final stage connects that understanding to a practical, physical-world use case: identifying components during a motorcycle rebuild using computer vision.

---

2. Peeling Back the Abstraction Layers

Stage 1 — High-Level Model (Keras CNN)

- **Goal:** Prove the concept works.
- Achieves ~98–99% accuracy on MNIST with minimal code.
- Uses high-level layers (Conv2D, Dense, etc.), but hides the inner math.

Talking Point:

> "This is the 'black box' stage — it shows that the system works, but not why. It's the baseline I'll progressively deconstruct."

Stage 2 — Manual Training Loop

- Replaced model.fit() with a custom training routine.
- Now I explicitly control:

    - Loss computation
    - Gradient calculation
    - Weight updates via optimizer
- tf.GradientTape() shows the calculus of learning — backpropagation in action.

Talking Point:

> “Here, I’m no longer letting TensorFlow handle the details. I’m running the show — every update is visible, calculated, and intentional.”

Stage 3 — TensorFlow Low-Level Ops

- Replaced layers with raw operations (tf.nn.conv2d, tf.Variable, etc.).
- Manually manage weights, forward passes, activations, and pooling.
- Convolution, softmax, and feature extraction are all expressed directly in code.

Talking Point:

> “This step is where the math becomes visible. It’s no longer a layer — it’s matrix multiplication, convolution kernels, and ReLU activations written out by hand.”

Stage 4 — Pure NumPy Models

- Implemented **softmax regression** and a **2-layer MLP** entirely from scratch.
- Every gradient is calculated manually.
- The model still learns — proving that *the math alone* is enough to power machine learning.

Talking Point:

> “This is the proof of mastery stage. If TensorFlow vanished tomorrow, I could still train a model because I understand exactly what it’s doing.”

---

3. Why This Matters

- Most projects focus on *using* machine learning tools. Mine is about *understanding* them deeply.
- By progressively stripping away abstractions, I build a bridge between theory and practice.
- That understanding gives me flexibility — I can now modify, optimize, or reimplement any component to suit a specific application.

Talking Point:

"The deeper I go, the more control I have. At this point, the libraries are transparent — and that gives me the power to innovate."

---

4. Real-World Application: Motorcycle Part Identification

"For the application phase, I'm tying this project directly to my own mechanical work — using image classification to identify motorcycle parts. It's a perfect way to show how mathematical theory, coding practice, and real-world utility come together."

- **Motivation:** During a motorcycle rebuild, identifying and cataloging hundreds of components is time-consuming and error-prone.
- **Objective:** Build a model that can analyze an image of a part (e.g., piston, valve, gasket, clip) and classify it correctly.
- **Approach:**
    - Use a convolutional neural network (CNN) architecture.
    - Fine-tune it on a dataset of labeled motorcycle parts (photos I can easily collect).
    - Demonstrate how the same techniques used for MNIST can be adapted for a real-world classification problem.
- **Outcome:** A working prototype that can identify parts from photos — potentially extendable to real-time workshop tools.

Talking Point:

"It's a natural extension: the same convolutional math that identifies handwritten digits can also distinguish between a valve spring and a piston ring. This shows how the project isn't just theoretical — it's directly applicable to real engineering tasks."

Closing Statement

"The story of this project is: machine learning is just math — and now, I understand that math deeply enough to rebuild every piece from the ground up. With that foundation, I'm applying those principles to a real-world challenge: teaching a neural network to recognize motorcycle parts. It's a perfect demonstration of the journey from abstract mathematics to hands-on, practical intelligence."