

Our senior design product is about creating an end-to-end visual application for manipulating neural networks. This will involve both designing, training, and testing them. Explainable AI is a growing field, and a graphical interface for a commonly used framework will be a welcome contribution to that space. We will use the PyTorch backend for our project. It will involve lots of coding, both in the front and back end. This project will cover a wide range of disciplines from UI development to neural network analysis.

I believe that my cumulative education and career experiences have made me into a strong candidate for this project. I've taken a couple of electives that are not standard curriculum but have taught me a lot about machine learning. Through AI Principles and Applications (CS4033), I learned optimization strategies and Boolean logic. In Math Logic (CS5121), I learned Boolean algebra and how to solve complicated logical problems. Currently, I am taking Probability and Random Processes (EECE6019), which will help me understand more about probability. All of these courses have come together to make me a skilled developer.

Most of my co-op experience has been centered around machine learning and artificial intelligence. I have spent four semesters as a Software Engineering Co-op at Siemens, working on many different projects. For my first project, I created a boilerplate for federated learning with differential privacy, a way to protect user identity while improving a large model. I also worked heavily with computer vision to detect electronic components on circuit boards. Eventually, I moved on to constructive solid geometry, and I spent my last semester leading a team to create an interface between generative AI models and 3D modeling software. These experiences were integral in my professional development. I became a lot better at PyTorch, Python, and AI development, and these skills will be perfect for our project.

As stated before, this project will have an end-to-end design. This means that all of the model creation, training, and evaluation will be able to be controlled through the application. We will start with a basic neural network, with just a few layers, and create a very basic interface to add layers. One extremely helpful and unique feature would be to include data validation – not all layers are compatible with each other, and the dimensions of data can conflict. This is usually discovered at runtime, but it can be inferred in the design stage using linear algebra. Including this validation will be helpful to users.

I'm excited to participate in this project because I think we can create a tool that is both educational and productive. My end goal is to walk away with a functional, natively hosted application that interfaces seamlessly with the PyTorch API. I think we can evaluate this well by comparing results of models we create directly with code to models we create with our interface. If the results match up, then we've succeeded. Now, the PyTorch API is extremely complicated, and there are many functions that we will have to implement. We don't need to implement every function, but at least a basic set would be necessary. As long as we document our reasonings behind what we choose and provide ample examples, we will have a sufficient product.

Overall, this is a project that shows a lot of promise, and I'm excited to work on it.