

Sprint Review: Neural Network Playground

Team 8 | SE101 Fall 2025

Team Members: Ario Barin Ostovary, David Estrine, Kevin Yan, Sicheng Ouyang, Yi Xing

Status: Completed and Delivered (Dec 2, 2025)

Introduction

Machine learning is one of the most transformative technologies of our time, yet the barrier to entry remains high for beginners. Traditional approaches require extensive coding knowledge, mathematical background, and familiarity with complex frameworks like TensorFlow or PyTorch before users can even train their first model.

Neural Network Playground was created to solve this problem. Our application provides a visual, no-code interface where users can design, train, and test neural networks through intuitive drag-and-drop interactions. By abstracting away the complexity of code, we enable students, educators, and curious learners to focus on understanding *how* neural networks work rather than struggling with syntax and implementation details.

Our Mission: Lower the barrier to entry for machine learning education by providing a hands-on, visual learning experience that makes neural network concepts accessible to everyone.

1. Executive Summary

The Neural Network Playground was delivered on time, meeting all "Must-Have" requirements. The application enables beginners to visually design and train neural networks without code.

Key Achievements:

- **Functional:** End-to-end workflow (Build, Train, Test) complete.
 - **Quality:** 83% backend test coverage (Target: 70%), 0 critical bugs.
 - **Performance:** Training <5 min/dataset on CPU.
-

2. Increments & Velocity

Total Velocity: 110 Story Points (Avg 4.8/day) | **Total Commits:** 79

Sprint 1: Foundation (Nov 2-8)

- **Delivered:** Backend API, Dataset Registry, 5 Curated Datasets.
- **Velocity:** 2.9 pts/day (Ramp-up phase).

Sprint 2: Core Engine (Nov 9-15)

- **Delivered:** Dynamic PyTorch Model Builder, Training Engine (Threaded), API Endpoints.
- **Velocity:** 5.0 pts/day.

Sprint 3: Frontend (Nov 16-22)

- **Delivered:** React Flow Visual Builder, Training Dashboard, Testing Interface.
- **Velocity:** 5.7 pts/day.

Sprint 4: Polish & Integration (Nov 23-Dec 2)

- **Delivered:** UI Polish (Dark mode, Visualizations), Bug Fixes (Scroll, Wine Glass), Final Docs.
 - **Velocity:** 7.5 pts/day (Final push).
-

3. Stakeholder Feedback Summary

Stakeholders: Course Instructor, TAs, End Users (Students).

- **Educational Value:** Validated. The "No-Code" visual approach aligns with the Charter's goal of lowering the barrier to entry for ML beginners.
 - **Technical Execution:** Approved. Architecture (FastAPI + React) and Dataset Registry pattern met technical requirements.
 - **Usability:** Positive. User testing confirmed workflow completion in <15 minutes.
 - **Scope:** Managed. "Nice-to-have" features (Persistence, Smart Error Guidance) were correctly deferred to prioritize core stability.
-

4. Retrospective

What Went Well:

- **Architecture:** Dynamic model construction proved simpler and more robust than code generation.
- **Quality:** High test coverage (83%) prevented regression bugs during integration.
- **UX:** Dataset-specific visualizations (Wine Glass, Flowers) significantly enhanced engagement.

Challenges:

- **Time Pressure:** 23-day timeline required strict MVP scoping.
- **Frontend Complexity:** React Flow integration required significant learning curve.

Next Steps (Future Work):

- Model Persistence (Save/Load).
 - WebSocket integration for real-time updates.
-

Conclusion

Over the course of 23 days and 4 sprints, Team 8 successfully delivered a fully functional Neural Network Playground that meets all project requirements. Our application demonstrates that complex machine learning concepts can be made accessible through thoughtful design and visual interaction.

What We Accomplished:

- **Complete End-to-End Workflow:** Users can select datasets, visually build neural network architectures, train models with real-time feedback, and test predictions—all without writing a single line of code.
- **5 Curated Datasets:** Including Iris (classification), MNIST (image recognition), California Housing (regression), Wine Quality, and Synthetic data for experimentation.
- **Dynamic Model Builder:** A flexible PyTorch-based engine that constructs neural networks on-the-fly from visual layer configurations.
- **Rich Visualizations:** Dataset-specific testing interfaces with interactive elements (wine glass animations, California map, iris flower diagrams) that make learning engaging.
- **Production-Quality Code:** 83% test coverage, clean architecture, and comprehensive documentation.

Impact: Our project validates that visual, interactive tools can effectively teach neural network fundamentals. The positive stakeholder feedback and successful user testing confirm that Neural Network Playground achieves its educational mission.

We are proud to deliver a tool that will help the next generation of learners take their first steps into the world of machine learning.

Thank you for your attention. Questions?