Advanced Institute for Artificial Intelligence

No-Code Neural Networks

https://advancedinstitute.ai

No-Code Neural Networks

Building AI Models Without Programming

References and Resources

Tools and Platforms

- Google Teachable Machine
- ml5.js
- TensorFlow.js
- TensorFlow Playground
- Runway ML

Introduction

Why No-Code AI Tools?

The AI Accessibility Gap

The Challenge

- □ **Traditional AI Development:**
  - Requires programming skills (Python, etc.)
  - Understanding of ML algorithms
  - Experience with frameworks (PyTorch, TensorFlow)
  - Computing resources and environment setup
- □ **The Problem:**
  - Many creative ideas come from **non-programmers**
  - Artists, designers, educators, domain experts
  - Long learning curve discourages experimentation
- □ **The Solution: No-Code AI Tools**
  - Democratize AI development
  - Enable rapid prototyping
  - Focus on ideas, not implementation

What are No-Code AI Tools?

Overview

- **Definition:** Platforms that allow building ML models through graphical interfaces
- **Key features:**
  - Visual, drag-and-drop interfaces
  - Pre-built model architectures
  - Automated training process
  - Easy data collection and labeling
  - Direct deployment/export options
- **Use cases:**
  - Rapid prototyping
  - Educational demonstrations
  - Quick proof-of-concepts
  - Non-technical stakeholder presentations
- **Not a replacement for code!** But a complementary tool

When to Use No-Code vs Code

Making the Right Choice

- □ **Use No-Code Tools when:**
  - Quick prototyping and experimentation
  - Teaching/learning ML concepts
  - Limited technical resources
  - Standard use cases (image/audio classification)
  - Demonstrating ideas to stakeholders
- □ **Use Code when:**
  - Custom architectures needed
  - Production-scale deployment
  - Complex data pipelines
  - Performance optimization critical
  - Integration with existing systems
- □ **Best approach:** Start no-code, migrate to code when needed

Tool 1

Google Teachable Machine

Google Teachable Machine

Overview

- □ **What is it?**
  - Free web-based tool by Google
  - Train models directly in your browser
  - No installation or sign-up required
- □ **Supported tasks:**
  - **Image classification** (using webcam or files)
  - **Audio classification** (sounds, speech, music)
  - **Pose classification** (body positions)
- □ **Key features:**
  - Real-time training and testing
  - Export to TensorFlow, TensorFlow.js, TensorFlow Lite
  - Shareable hosted models
- □ **Website:** teachablemachine.withgoogle.com

How Teachable Machine Works

The Process

- **Step 1: Choose project type**
  - Image, Audio, or Pose
- **Step 2: Gather data**
  - Use webcam for live capture
  - Upload files from computer
  - Need examples for each class
- **Step 3: Train**
  - Click ”Train Model”
  - Training happens in browser (uses TensorFlow.js)
  - Takes seconds to minutes

How Teachable Machine Works

The Process

- **Step 4: Test**
  - Real-time prediction
  - Adjust data and retrain if needed
- **Step 5: Export**
  - Download model or get hosted link

Teachable Machine - Image Project

Example Use Cases

- **Classification examples:**
  - Rock-paper-scissors detector
  - Product quality control (good/defect)
  - Recycling sorter (plastic/paper/metal)
  - Emotion detector (happy/sad/neutral)
  - Hand gesture controls

Teachable Machine - Image Project

Example Use Cases

- □ **Technical details:**
  - Uses MobileNet architecture (transfer learning)
  - Trains only final layers on your data
  - Requires 20-50+ images per class
  - Works best with clear, consistent backgrounds
- □ **Tips for better results:**
  - Diverse examples (different angles, lighting)
  - Balance classes (similar number of samples)
  - Add "None" class for background

Teachable Machine - Audio Project

Sound Classification

- **Classification examples:**
  - Voice command recognition (yes/no/stop)
  - Musical instrument detector
  - Environmental sound classifier
  - Baby cry vs. laughter
  - Machine fault detection (by sound)
- **Technical details:**
  - Records 1-second audio samples
  - Converts to spectrograms
  - Uses CNN on spectrograms
- **Tips:**
  - Record in consistent environment
  - Include background noise class
  - Record at least 30+ samples per class

Teachable Machine - Pose Project

Body Position Classification

- **Classification examples:**
  - Exercise form checker (correct/incorrect squat)
  - Yoga pose detector
  - Sign language recognition
  - Dance move classifier
  - Posture monitor
- **Technical details:**
  - Uses PoseNet for keypoint detection
  - Trains classifier on keypoint positions
  - Works with single person in frame
- **Tips:**
  - Good lighting and camera angle
  - Full body in frame
  - Varied examples (different people, clothes)

Exporting from Teachable Machine

Deployment Options

- **Option 1: Upload model (cloud hosted)**
  - Shareable link
  - Use in web apps with JavaScript
  - Free hosting by Google
- **Option 2: Download model**
  - **TensorFlow:** For Python applications
  - **TensorFlow.js:** For web/Node.js
  - **TensorFlow Lite:** For mobile (Android/iOS)
- **Integration:**
  - Simple API for predictions
  - Works in p5.js, ml5.js projects
  - Can embed in websites
- **Limitation:** Model architecture is fixed (can't customize)

Live Demo: Teachable Machine

What We'll Build

- **Demo project: Hand Gesture Classifier**
  - Class 1: Thumbs up
  - Class 2: Thumbs down
  - Class 3: Peace sign
  - Class 4: None (background)
- **Steps we'll follow:**
  - 1. Create new image project
  - 2. Capture training data (webcam)
  - 3. Train the model
  - 4. Test in real-time
  - 5. Export model
- **Expected time: 5-10 minutes**
- **Participants: Try it yourself!**

Tool 2

TensorFlow Playground

TensorFlow Playground

Understanding Neural Networks Visually

- □ **What is it?**
  - Interactive visualization of neural networks
  - See how networks learn in real-time
  - Runs entirely in browser
- □ **Purpose:**
  - **Educational tool**, not for production
  - Understand network behavior, and experiment with architectures
- □ **Features:**
  - Adjust layers, neurons, activation functions
  - See decision boundaries update live
  - Various toy datasets (spiral, circle, XOR)
  - Visualize neuron activations
- □ **Website:** playground.tensorflow.org

TensorFlow Playground - Features

What You Can Experiment With

- **Dataset selection:**
  - Classification problems (circle, XOR, spiral, etc.)
  - Regression problems
- **Network architecture:**
  - Number of hidden layers (0-6)
  - Neurons per layer (1-8)
  - Add/remove neurons visually

TensorFlow Playground - Features

What You Can Experiment With

- **Hyperparameters:**
  - Learning rate
  - Activation function (ReLU, Tanh, Sigmoid, Linear)
  - Regularization (L1, L2)
  - Batch size
- **Input features:**
  - X1, X2 (coordinates)
  - $X1^2$, $X2^2$ (squares)
  - $X1 \times X2$, sin(X1), sin(X2)

Learning with TensorFlow Playground

Key Insights

- **Visualizations show:**
  - How decision boundaries form
  - Effect of different activation functions
  - Impact of network depth vs. width
  - Why some problems need deep networks
- **Common experiments:**
  - Try solving XOR with no hidden layer (impossible!)
  - Solve spiral with different architectures
  - Compare ReLU vs. Tanh activations
  - See overfitting with too many neurons

Learning with TensorFlow Playground

Key Insights

□ **Best for:**

- Teaching neural network concepts
- Understanding why deep learning works
- Intuition about hyperparameters

Tool 3

ml5.js

ml5.js

Machine Learning for Web Creators

- □ **What is it?**
  - JavaScript library for ML in the browser
  - Built on TensorFlow.js
  - Simplified API for common tasks
- □ **Philosophy:**
  - Make ML accessible to artists, creative coders
  - Works with p5.js (creative coding library)
  - Low-code (not no-code, but very simple)

ml5.js

Machine Learning for Web Creators

- □ **Pre-trained models:**
  - Image classification (MobileNet)
  - Object detection (COCO-SSD, YOLO)
  - Pose detection (PoseNet, BodyPix)
  - Hand pose, face detection
  - Text generation, sentiment analysis
- □ **Website:** ml5js.org

ml5.js - Why It's Almost No-Code

Simplified Machine Learning

- **Example: Image classification in 5 lines**

```javascript
let classifier = ml5.imageClassifier('MobileNet', modelLoaded);
classifier.classify(image, gotResult);

function gotResult(error, results) {
  console.log(results);  // [{label: 'dog', confidence: 0.95}]
}
```

- **Why it's beginner-friendly:**
  - No model architecture to define
  - No training loop to write
  - Pre-trained models ready to use
  - Simple callback-based API
- **Great for:** Creative projects, interactive art, web demos

ml5.js - Transfer Learning

Custom Models with Feature Extraction

- **Feature extractor approach:**
  - Use pre-trained model as feature extractor
  - Train small classifier on top
  - Much faster than full training
- **Example workflow:**

ml5.js - Transfer Learning

Custom Models with Feature Extraction

```javascript
const featureExtractor = ml5.featureExtractor('MobileNet');
const classifier = featureExtractor.classification(video);
// Add training data
classifier.addImage('cat');
classifier.addImage('dog');
// Train
classifier.train(function(loss) {
  console.log('Loss:', loss);
});
// Predict
classifier.classify(function(err, result) {
  console.log(result);
});
```

Tool 5

Runway ML

Runway ML

AI for Creatives

- **What is it?**
  - Platform for creative applications of AI
  - Web-based and desktop app
  - Focus on generative models
- **Capabilities:**
  - Image generation (Stable Diffusion, etc.)
  - Video editing (background removal, style transfer)
  - 3D modeling
  - Motion tracking
  - Custom model training

Runway ML

AI for Creatives

- □ **Target audience:**
  - ■ Artists, designers, filmmakers
  - ■ Content creators
- □ **Pricing:** Free tier + paid plans
- □ **Website:** runwayml.com

Runway ML - Features

Beyond Classification

- **Pre-trained models:**
  - 100+ models available
  - Image generation, style transfer
  - Object detection and segmentation
  - Pose estimation
  - Video effects

Runway ML - Features

Beyond Classification

- □ **Custom training:**
  - Train GANs, style transfer models
  - Object detection models
  - No code required
- □ **Integration:**
  - Export to various formats
  - API for real-time use
  - Plugins for creative tools (Photoshop, etc.)
- □ **More advanced than Teachable Machine**

Comparison

Choosing the Right Tool

Tool Comparison

At a Glance

Tool	Type	Ease	Cost	Best For
Teachable Machine	Web		Free	Quick prototypes
TensorFlow Playground	Web		Free	Learning
ml5.js	Web/Code		Free	Creative coding
Runway ML	Web/Desktop		Free/Paid	Creative AI

- □ **Easiest:** Teachable Machine
- □ **Most educational:** TensorFlow Playground
- □ **Most creative:** Runway ML

Which Tool Should You Use?

Decision Guide

- □ **Use Teachable Machine if:**
  - Complete beginner, first ML project
  - Need quick demo or proof-of-concept
  - Image/audio/pose classification
  - Want to share model easily
- □ **Use TensorFlow Playground if:**
  - Learning neural network concepts
  - Teaching ML to others
  - Understanding hyperparameters

Which Tool Should You Use? (cont.)

Decision Guide

- □ **Use ml5.js if:**
  - Building web application
  - Creative coding project
  - Need real-time browser inference
  - Working with p5.js
- □ **Use Runway ML if:**
  - Creative/artistic applications
  - Generative AI projects
  - Video/image editing with AI
  - Have budget for paid features
- □ **Remember:** Can combine tools!

Limitations

Understanding the Trade-offs

Limitations of No-Code Tools

What You Give Up

- **Limited customization:**
  - Fixed architectures
  - Can’t modify training process
  - Limited hyperparameter control
- **Performance:**
  - May not match custom models
  - Optimized for ease, not efficiency
  - Larger model sizes
- **Scalability:**
  - Not for large-scale production
  - Limited batch processing
  - May have usage limits
- **Data:**
  - Limited data preprocessing options

Limitations of No-Code Tools (cont.)

What You Give Up

- **Task variety:**
  - Focused on common tasks
  - Classification, detection, segmentation
  - Limited support for custom problems
- **Integration:**
  - May be difficult to integrate with existing systems
  - API limitations
  - Dependency on external services
- **Learning:**
  - Black box - don't learn what's inside
  - May not understand underlying concepts
  - Can be crutch for avoiding learning
- **When limitations matter, move to code!**

Best Practices with No-Code Tools

Getting the Most Out of Them

- **Data quality matters:**
  - Collect diverse, representative examples
  - Balance classes (similar samples)
  - Include edge cases
  - Add ”none” class for background

Best Practices with No-Code Tools

Getting the Most Out of Them

□ **Iterate:**

- Start simple, add complexity
- Test frequently
- Collect more data where model fails

□ **Understand limitations:**

- Know when to move to code
- Test in realistic conditions
- Have backup plans

□ **Use as learning tool:**

- Understand concepts before diving into code
- Prototype ideas quickly
- Then implement properly in code

Hands-On Session

Let’s Build Something!

Hands-On Activity

Build Your Own Model

- □ **Task: Create a custom classifier**
- □ **Options:**
  - 1. Hand gesture recognition (rock/paper/scissors)
  - 2. Facial expression classifier (happy/sad/neutral)
  - 3. Sound classifier (clap/snap/whistle)
  - 4. Your own idea!
- □ **Steps:**
  - 1. Go to teachablemachine.withgoogle.com
  - 2. Choose project type
  - 3. Collect training data (minimum 30 samples per class)
  - 4. Train model
  - 5. Test and iterate
  - 6. Share your result!

Activity Guidelines

Tips for Success

- □ **Tips:**
  - Work in pairs or small groups
  - Start simple (2-3 classes)
  - Collect at least 30 examples per class
  - Vary your examples (angles, lighting, backgrounds)
  - Test with new examples, not training data
- □ **Common issues:**
  - Not enough data → collect more
  - Confusion between classes → make them more distinct
  - Works in training but not testing → overfitting, add variety
- □ **Have fun and experiment!**
- □ **Share your results with the class**

Key Takeaways

Summary

- □ **No-code AI tools democratize machine learning**
  - Anyone can build AI models - Focus on ideas, not implementation
- □ **Great for:**
  - Quick prototyping
  - Learning concepts
  - Demos and proof-of-concepts
  - Teaching and education
- □ **Tools covered today:**
  - Teachable Machine - easiest, most versatile
  - TensorFlow Playground - educational
  - ml5.js - creative web projects
  - Runway ML - creative AI applications
- □ **Start no-code, transition to code when needed**

Questions?