Machine Learning Techniques Exploration

Polynomial Interpolation, Neural Networks, CNNs, Adversarial ML, and GANs

Jamil Gafur

Introducing US-RSE (US Research Software Engineer Association)

- The **US-RSE** is an organization focused on building a strong community of research software engineers across the United States. Our mission is to provide a collaborative space for individuals who use their software engineering expertise to advance research and innovation. Here's a brief overview:
- **Community Building**:
  - The US-RSE strives to create a supportive network for research software engineers (RSEs), providing opportunities to connect, communicate, and share resources.
  - We host monthly community calls, various working groups, and online discussions on platforms like Slack to foster this collaborative spirit.
- **Advocacy**:
  - We advocate for RSEs, helping to raise awareness of the crucial role they play in scientific advancements.
  - We work to improve the recognition and support of RSEs, both within academia and industry.
- **Resources and Training**:
  - The association offers resources to enhance the career growth of RSEs, including training programs, white papers, and workshops.
  - We support various working groups, including **Education and Training**, **Website Development**, and **Diversity, Equity, and Inclusion (DEI)** initiatives.
- **Organizational Growth**:
  - The association has been growing steadily since its inception in 2019 and continues to expand across the United States and internationally.
  - We recently became a member of the **Open Collective Foundation**, allowing us to accept donations and transparently manage resources for the community.

Agenda

- **Polynomial Interpolation**
  - **Definition:** A method of estimating values between known data points using polynomials.
  - **Applications:** Used in curve fitting, signal processing, and numerical analysis.
- **Neural Networks**
  - **Definition:** A collection of algorithms designed to recognize patterns, inspired by the human brain.
  - **Applications:** Image recognition, language processing, and autonomous systems.
- **Convolutional Neural Networks (CNNs)**
  - **Definition:** A specialized type of neural network for processing structured grid data (e.g., images).
  - **Applications:** Computer vision, image classification, medical imaging, and video analysis.
- **Adversarial Machine Learning**
  - **Definition:** The study of malicious attacks on machine learning models and how to defend against them.
  - **Applications:** Enhancing model robustness, developing security for AI systems, and testing model vulnerability.
- **Generative Adversarial Networks (GANs)**
  - **Definition:** A framework for training models that generate new data by pitting two networks (generator and discriminator) against each other.
  - **Applications:** Image generation, data augmentation, and deepfake creation.

Polynomial Interpolation

- **Definition:** Polynomial interpolation involves fitting a polynomial function to a set of data points to estimate values between them.
- **Use Case:**
  - **Application:** Fits smooth curves to noisy data, making it useful for approximating complex relationships and filling in missing data points.
  - **Example:** In signal processing or curve fitting for experimental data, where smooth transitions between noisy observations are needed.
- **Visual:**
  - **Graph:** A plot showing noisy data points with a smooth polynomial curve fit overlaid, demonstrating how the polynomial captures the underlying pattern of the data.

Polynomial Interpolation Code Overview

- **Data Generation**: Create synthetic quadratic data with noise.
- **Polynomial Feature Expansion**: Transform data for fitting.
- **Least Squares Estimation**: Find best-fitting polynomial.
- **Visualization**: Plot noisy data and fitted polynomial.

Neural Networks Overview

- **Definition**: A network of layers transforming inputs into outputs.
- **Use Case**: Approximate complex relationships in data.

Neural Network Architecture

- **Architecture**: Multi-layer perceptron (MLP).
- **Training**: Backpropagation and Adam optimizer.
- **Loss Function**: Mean Squared Error (MSE).
- **Visualization**: Neural network structure and data flow.

Neural Network Code Overview

- Neural network with ReLU activations.
- **Training**: Adjust weights using Adam optimizer and Mean Squared Error (MSE) loss function.
- **Metrics**:
  - **MSE (Mean Squared Error)**: Measures the average squared difference between predicted and actual values. Lower MSE indicates better model performance.
  - **MAE (Mean Absolute Error)**: Measures the average absolute difference between predicted and actual values. It is less sensitive to outliers than MSE.
  - **R^2 (R-squared)**: Represents the proportion of variance in the dependent variable that is predictable from the independent variables. A higher R^2 value indicates a better fit.
- **Visualization**: Loss and accuracy plots to track model performance during training.

Convolutional Neural Networks (CNNs) Overview

- **Definition**: A type of deep learning model for image classification.
- **Key Layers**: Convolutional layers, pooling layers, fully connected layers.
- **Activation**: ReLU.
- **Use Case**: Image classification, especially for visual data like MNIST.

CNN Architecture

- **Layers**: Convolutional layers to extract features, pooling to reduce dimensionality.
- **Model**: Simple CNN with two convolutional layers.
- **Training**: Train with MNIST dataset.
- **Visualization**: Diagram showing CNN layers.

CNN Code Overview

- **Data**: Use MNIST dataset of handwritten digits.
- **Model**: CNN with two convolutional layers, followed by fully connected layers.
- **Training**: Train using cross-entropy loss and accuracy metrics.
- **Visualization**: Accuracy vs. Epochs and sample predictions.

IOWA

Adversarial Machine Learning (AML) Overview

- **Concept**: Adversarial attacks introduce small perturbations to input data, misleading the model.
- **Key Attack**: Fast Gradient Sign Method (FGSM).
- **Use Case**: Demonstrates vulnerabilities in machine learning models.

Adversarial ML Code Overview

- **Definition:** A Convolutional Neural Network (CNN) trained on the MNIST dataset, which contains images of handwritten digits.
- **Objective:** To classify images of digits from 0-9 using CNN's convolutional layers to extract features.
- **Adversarial Attack: FGSM Perturbations Added to Input**
  - **Definition:** The Fast Gradient Sign Method (FGSM) is a simple and effective adversarial attack where small perturbations are added to the input image based on the gradient of the loss function with respect to the input.
  - **Purpose:** To intentionally mislead the model by creating adversarial examples that are visually similar to the original but classified incorrectly.
- **Testing: Evaluate Model's Accuracy on Perturbed Images**
  - **Goal:** Assess how well the CNN performs when presented with adversarial images, which are modified inputs designed to fool the model.
- **Visualization:**
  - **Comparison:** Display side-by-side images of the original MNIST digits and their adversarial counterparts, showing how the perturbations alter the image while still being visually recognizable.

GANs Overview

- **Definition**: A generative model that creates fake data.
- **Components**: Generator and Discriminator.
- **Min-Max Game**: Generator tries to fool the discriminator, discriminator tries to differentiate real from fake data.
- **Use Case**: Generate new data that resembles the training data (e.g., synthetic images).

GANs Architecture

- **Generator**: Creates fake data (e.g., images).
- **Discriminator**: Distinguishes between real and fake data.
- **Training**: Both networks are trained simultaneously.
- **Visualization**: Diagram of GANs showing the generator and discriminator.

Practical Applications

- **Polynomial Interpolation**: Data smoothing, curve fitting.
- **Neural Networks**: Function approximation, pattern recognition.
- **CNNs**: Image classification, object detection.
- **Adversarial ML**: Model robustness testing, security applications.
- **GANs**: Data generation, content creation.

Challenges in Machine Learning

- **Data Quality**: Noisy, incomplete data affects performance.
- **Model Robustness**: Adversarial attacks and overfitting.
- **Training Complexity**: Large datasets and long training times.
- **Interpretability**: Understanding how complex models make decisions.