Predicting Automobile Fuel Efficiency with Support Vector Regression and Artificial Neural Networks

Prepared for: Dr. Chaity Banerjee Mukherjee

Prepared by: Azaria A. Reed, Student, CS 430: Survey of Artificial Intelligence – 01

Date: December 8, 2025

1 Introduction

Automobile fuel efficiency impacts the environment, fuel cost, and even vehicle design. With rising fuel prices and tightening regulations, manufacturers and consumers alike need tools that reliably estimate miles per gallon (MPG) from a car's technical characteristics before production or purchase, respectively. The UCI Auto MPG dataset contains historical vehicle specifications and corresponding fuel efficiencies, notably serving as a benchmark for regression models predicting MPG from car features.

This project uses Support Vector Regressor (SVR) and a feed-forward Artificial Neural Network (ANN), two machine learning approaches, to model city-cycle fuel consumption in miles per gallon. Both methods resourcefully use existing libraries and code, namely scikit-learn and pandas, per the technical instructions. Plainly, the project's goal is to build baseline SVR and ANN models on the UCI Auto MPG dataset, compare their predictive performance with mean squared error (MSE), and analyze the ANN model's training behavior with a convergence plot of its training and validation errors across epochs.

SVR can model nonlinear relationships using kernel functions and often performs well on small to medium tabular datasets. ANNs can estimate complex nonlinear mappings, given appropriate regularization and training. Relevantly, a dataset description, related work surveys on regression and model diagnostics, an explanation of the two approaches and the Adam optimizer for the ANN, an experimental results report, and a discussion of the relative performance of SVR and ANN on the Auto MPG task will ensue.

2 Related Work

Regression and model-evaluation research frequently use the UCI Auto MPG dataset because it contains continuous and discrete attributes, missing values, and a moderate sample size [1]. Even when not focusing directly on this dataset, many papers exploring regression diagnostics, ensemble methods, and complex models use similar tabular data.

Wright and König [2] analyze how random forests should handle categorical features and propose splitting strategies that account for both the category structure and performance. Their work shows that, unless the algorithm handles categorical variables carefully, models that include many decision trees can struggle with mixed feature types. Although the UCI Auto MPG dataset does not focus on random forests, it does include discrete variables. Wright and König's work applicably motivates careful preprocessing when working with more complex models.

Ojha and others [3] introduce type-2 hierarchical fuzzy inference trees and multiobjective programming to optimize them. Importantly, their method addresses nonlinear regression problems where interpretability and uncertainty modeling matter. Both ANNs and fuzzy inference trees capture nonlinear relationships between inputs and outputs. However, fuzzy inference trees emphasize rule-based structure rather than hidden layers. This difference indicates that, to outperform simple linear regression, many researchers explore rich nonlinear models.

Li and Zhu [4] propose a specification test that uses local and global smoothing techniques for regression models. Their method examines whether a fitted regression function matches the underlying data-generating process. Their work emphasizes that, by itself, a low training error does not guarantee a correct functional form. Despite this project's reliance on convergence plots rather than specification tests, Li and Zhu's work confirms the need to compare models and monitor for overfitting.

Wager and Hastie [5] study random forest confidence intervals with iterative resampling and sensitivity-based variance estimation. By treating random forests as statistical estimators, they estimate prediction uncertainty. This perspective on multi-model systems parallels how this project treats SVR and ANN: as competing estimators, comparable through error metrics and diagnostic plots, despite no formal confidence interval computation.

Finally, Freedman [6] uses visual backpropagation with input sensitivity maps to visualize which parts of an image influence a neural network's prediction. Although there, the focus is on convolutional networks and vision, it illustrates that interpreting neural networks alongside measuring their accuracy is a research goal. Accordingly, this project examines the ANN's training and validation errors over epochs to understand its learning dynamics and overfitting behavior.

Unitedly, these papers demonstrate that handling mixed feature types, exploring nonlinear models beyond simple linear regression, and using diagnostics and uncertainty estimation to understand model behavior are modern concerns in regression research. Using SVR and ANN, the project applies these ideas at a smaller scale to the Auto MPG prediction problem.

3 Dataset Description

The UCI Auto MPG dataset is from the UCI Machine Learning Repository and originally from the StatLib library [1]. The data describes city-cycle fuel consumption for passenger cars from the 1970s and early 1980s. Each record corresponds to a car model, including its technical characteristics and fuel efficiency for typical city driving.

Specifically, the dataset contains **398 records** and **nine attributes**:

- **Displacement**, a continuous feature describing engine displacement in cubic inches
- **Cylinders**, a continuous feature describing the number of cylinders
- **Horsepower**, a continuous feature with missing values describing engine horsepower
- **Weight**, a continuous feature describing vehicle weight in pounds
- **Acceleration**, a continuous feature describing the time to accelerate from 0 to 60 mph in seconds
- **Model year**, a discrete feature describing the model year from 1970 to 1982
- **Origin**, a discrete feature describing the region of origin as 1 (USA), 2 (Europe), or 3 (Asia)
- **Car name**, a string identifier that encodes make and model
- **MPG**, the continuous target variable describing fuel efficiency in miles per gallon

For this dataset, the goal is to predict MPG using regression. The combination of features enables models to understand how trade-offs between technology and design affect fuel efficiency. Nevertheless, the dataset poses a challenge: the horsepower feature contains missing values, represented by non-numeric "?" values, and the discrete features can introduce sharp nonlinearities in the response surface.

The project focuses on the seven numeric features and treats mpg as the target. Notably, car_name is a string identifier, not a predictive feature, because the project centers on SVR and ANN for numeric tabular data, not text encoding.

4 Methodology

4.1 Data Preprocessing

This project's source code uses pandas and scikit-learn to implement data loading and preprocessing. First, it reads the auto-mpg.data file with whitespace as the delimiter and assigns the standard UCI column names. With some pandas configuration, it then treats the non-numeric "?" values in the horsepower column as missing values. Then, it drops all rows with missing values, removing a small number of incomplete records and producing a clean, suitable dataset for both algorithms.

After cleaning, the program selects the seven numeric predictor columns and keeps MPG as the target. As in the homework, it then splits the data into training (80%) and testing (20%) sets using train_test_split with a constant random state to ensure reproducibility [7]. By mirroring the course's homework assignments, this project uses the taught strategies.

4.2 Support Vector Regression

Support Vector Regression extends support vector machines to continuous outputs. Instead of drawing a line that separates classes, SVR tries to find a curve or function that stays close to the data, or, in alternate phrasing, within an ε-insensitive tube around the data, while keeping the model simple. SVR can model nonlinear mappings between inputs and targets using a radial basis function (RBF) kernel.

The source code contains an SVR implementation using a **scikit-learn Pipeline**:

- **StandardScalar** to standardize each numeric feature to zero mean and unit variance
- **SVR(kernel= "rbf")** as the regression model

The pipeline ensures consistent scaling across both the training and testing sets. Aside from the kernel choice, default hyperparameters are present. During training, the pipeline received the training features and targets, and, during evaluation, it produced test-set predictions for MPG.

4.3 Artificial Neural Networks

For the ANN, the program uses scikit-learn's MLPRegressor to implement a feed-forward multilayer neural network. To elaborate, the network architecture consists of two hidden layers with 64 and 32 neurons, respectively, and Rectified Linear Unit (ReLU) activation. 1000 is the maximum number of iterations to allow sufficient optimization steps, and Adaptive Moment Estimation (Adam) is the optimizer.

Training the ANN and tracking convergence involved two related models:

1. A **curve-tracking ANN** for computing the mean squared error across epochs on an inner training and validation subset scaled with StandardScalar to produce a training-vs-validation MSE plot
2. A **final ANN pipeline** with StandardScalar fit once on the complete 80% training set to compute the final test-set MSE

The described structure allowed treating the final ANN similarly to SVR, as a standard pipeline model, while still generating a convergence plot depicting how training and validation errors evolve across epochs.

4.4 Evaluation Metrics and Plots

As in the course homework, MSE is the evaluation metric for both SVR and ANN [7]. For interpretation, a lower MSE indicates a better predictive performance.

For **diagnostic plots**, there is:

- A **predicted-vs-actual MPG scatter plot** for the SVR model, serving as a comparable diagnostic for a method that does not train in epochs
- A **convergence plot** for the ANN with training and validation MSE on the same graph across 100 epochs

5 Experimental Results

Adam performs stochastic gradient descent with adaptive learning rates for each parameter. It maintains running estimates of the mean and uncentered variance of past gradients. During training, Adam updates each weight using such, which combines the benefits of momentum and Root Mean Square Propagation (RMSProp). This design improves the optimizer's performance, even on noisy or poorly scaled problems, and works well out of the box for many neural network architectures.

The final test-set results appear below:

- **Support Vector Regressor (SVR):**
    - Test MSE ≈ **9.27383**
- **Artificial Neural Networks (ANN):**
    - Test MSE ≈ **7.38223**

Both models achieve reasonable accuracy on the UCI Auto MPG dataset, but the ANN yields a noticeably lower MSE than SVR, resulting in more accurate MPG predictions for unseen cars.

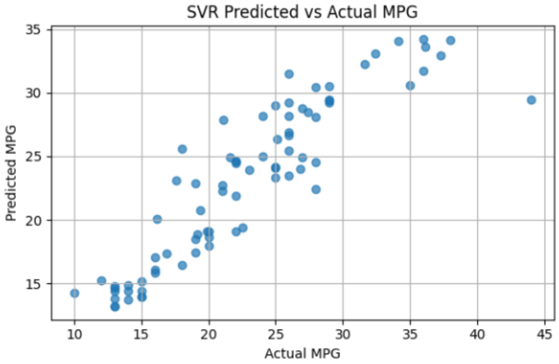

Figure 1. SVR Predicted-Vs-Actual MPG Scatter Plot

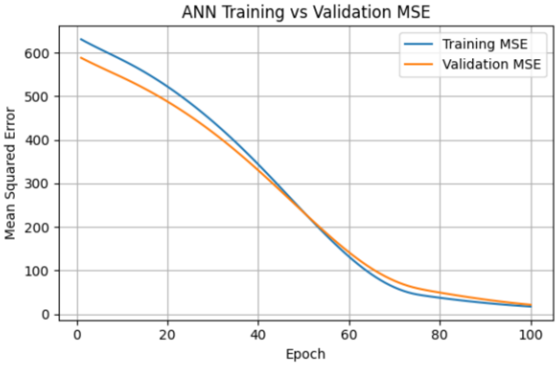

Figure 2. ANN Convergence Plot

In the SVR predicted-vs-actual scatter plot, the point distribution is roughly along the diagonal line, where predicted and actual MPG are equal. Although the model overestimates MPG for a few lower-efficiency cars and underestimates for some higher-efficiency cars, most predictions fall near this line. This pattern indicates that SVR captures the general relationship between features and MPG, even though it does not match the ANN's best accuracy.

The ANN convergence plot shows high training and validation MSE values at early epochs, mainly before 60. After the first 60-70 epochs, both curves drop sharply. After that interval, the curves slowly decline and flatten. Intriguingly, the training and validation curves remain close across epochs, suggesting improvement on both sets without a large generalization gap.

6 Discussion

The SVR and ANN comparison on the UCI Auto MPG dataset illustrates that both models benefit from simple preprocessing and feature scaling, but differ in their ability to capture complex relationships.

While the SVR model performs respectably with a test MSE of approximately 9.27, the ANN still outperforms it. Several factors may explain the performance gap. First, the SVR model employed default hyperparameters, namely regularization and kernel width. Without tuning these values, the model may be too simple, conducive to underfitting. Second, SVR with an RBF kernel does not automatically learn feature combinations in the same way a multilayer network does, even though it can capture nonlinear structure. Third, in this specific setting, the dataset, comprising only a few hundred records, and feature scaling may favor ANN's architecture and optimization dynamics.

The ANN model achieves the lowest test MSE of approximately 7.38, outperforms the SVR, and, on average, predicts automobile fuel efficiency more accurately. In explanation, the network's two hidden layers and ReLU activation give it flexibility to model nonlinear interactions among features. The convergence plot shows stable, monotonic improvement in both training and validation errors, and the closely spaced curves indicate that the network does not severely overfit the training data, suggesting that the Adam optimizer trains the network efficiently.

Regarding overfitting, the ANN convergence plot is essential to interpretation. If the training error dropped while the validation error rose, the model would overfit. During experimentation, the errors decreased simultaneously, with the gap between them remaining modest. This behavior suggests that the chosen architecture and early-stopping behavior implicit in the fixed epoch count control overfitting reasonably well. While the SVR model lacks an epoch-based convergence view, the predicted-vs-actual plot shows moderate variance and no extreme deviations, suggesting mild underfitting rather than overfitting.

Ultimately, the project exemplifies that simple pipelines with existing libraries can reproduce classic results on benchmark datasets and provide clear comparisons between distinct regression algorithms. In this setting, the ANN's flexibility and the Adam optimizer's adaptive learning allow the network to outperform an RBF-kernel SVR for predicting MPG.

7 References

[1] UCI Machine Learning Repository, "Auto MPG," University of California, Irvine. Available: Auto MPG dataset page.

[2] M. N. Wright and I. König, "Splitting on categorical predictors in random forests," *PeerJ*, 2019.

[3] V. Ojha, V. Snášel, and A. Abraham, "Multiobjective programming for type-2 hierarchical fuzzy inference trees," *IEEE Transactions on Fuzzy Systems*, 2017.

[4] L. Li and L. Zhu, "Specification testing for regressions: an approach bridging between local smoothing and global smoothing methods," *Annals of Statistics*, 2017.

[5] S. Wager and T. Hastie, "Confidence intervals for random forests: the jackknife and the infinitesimal jackknife," *Journal of Machine Learning Research*, 2014.

[6] R. Freedman, "Visual Backpropagation," *arXiv preprint*, 2019.

[7] A. Reed and W. Laughner, "Predicting automobile fuel efficiency using SVR and ANN," CS 430 Term Project Presentation, University of Alabama in Huntsville, 2025.