International Journal of Operations Research and Artificial Intelligence

www.ijorai.reapress.com

J. Civ. Asp. Struct. Eng. Vol. 1, No. 1 (2025) 47–53.

Paper Type: Original Article

Housing Price Prediction Using Deep Neural Networks: A Case Study on California Data

Vahid Bahadoran Fard*

Department of Applied Mathematics, Chouzhoubei Road, Yiwu, Zhejiang, China; Vahidmath@gmail.com.

Citation:

Received:18 April 2024 *Revised: 22 July 2024* *Accepted: 27 Sepetmber 2024*	Bahadoran Fard, V. (2025). Housing price prediction using deep neural networks:Acase study on california data. *International journal of operations research and artificial intelligence, 1* (1), 47-53.

Abstract

Accurate estimation of housing prices is essential for informed decision-making in urban planning, real estate investment, and economic forecasting. In this study, a Deep Neural Network (DNN) model is developed to predict median housing values using the California Housing dataset. The dataset, comprising multiple socioeconomic and geographical features, is preprocessed through feature scaling and partitioned into training and testing subsets. The proposed DNN architecture consists of multiple hidden layers employing ReLU activations, optimized using the Adam algorithm with Mean Squared Error (MSE) as the loss function. The model was trained over 100 epochs, achieving a final test MSE of 0.2578 and a Mean Absolute Error (MAE) of 0.3380. Moreover, an R-squared score of 0.8032 indicates strong predictive power and generalization capability. These results suggest that deep learning models can effectively capture complex, nonlinear relationships in housing data and offer reliable tools for real-world applications.

Keywords: Deep learning, Regression, California housing dataset, Neural network, Model evaluation.

1 | Introduction

Predictive modeling has become a crucial component in the analysis of large-scale datasets, especially in the context of real estate and economic forecasting. Among various machine learning approaches, deep learning techniques offer powerful capabilities in modeling nonlinear relationships and capturing complex patterns in data. The California Housing dataset, with its diverse set of socioeconomic and geographic features, provides an excellent foundation for building robust regression models aimed at estimating housing values.

In this study, we develop and evaluate a Deep Neural Network (DNN) to predict house prices based on this dataset. We employ standard preprocessing steps, including feature scaling, and assess model performance using Mean Squared Error (MSE), Mean Absolute Error (MAE), and the R-squared (R^2) metric. The goal is

Corresponding Author: Vahidmath@gmail.com

https://doi.org/10.48314/ijorai.v1i1.55

to demonstrate how neural networks can effectively generalize and provide accurate predictions in real-world regression.

The assessment of technical efficiency has been a critical topic in performance evaluation for over half a century. Early economic theories by Debreu [1] and Koopmans [2] laid the conceptual groundwork for measuring productive efficiency. Farrell [3] later introduced a formalized method for empirical efficiency evaluation, which triggered the development of various operational research models that measure the relative efficiency of Decision-Making Units (DMUs) based on observed inputs and outputs.

With the advent of Data Envelopment Analysis (DEA) by Charnes, Cooper et al. [4], a non-parametric framework was established to empirically evaluate efficiency without the need for predefined production functions. Subsequent extensions, such as the BCC model [5], allowed for variable returns to scale. However, both models rely on radial movements—either contracting inputs or expanding outputs proportionally—failing to capture slacks and non-radial inefficiencies that frequently arise in real-world data.

To overcome this, alternative models such as the Russell measures have been introduced. These measures allow for non-proportional adjustment of inputs and outputs, providing a more nuanced understanding of inefficiency. Färe et al. [6] proposed input and output versions of the Russell measure, which were later integrated into a combined formulation—the Russell Graph Measure Färe et al. [6]—capable of addressing inefficiencies on both sides simultaneously.

Despite their analytical advantages, Russell-based models are not without limitations. They often involve computational difficulties and can be challenging to interpret. To address these issues, this study proposes an improved Global Efficiency Measure (GEM) that draws on the strengths of the Russell Graph framework while enhancing usability and interpretability. By adopting a refined structure and incorporating desirable efficiency axioms, the proposed model aims to provide a more robust approach for comprehensive performance assessment.

2 | Background and Motivation for Deep Learning Integration

The Russell Graph Measure, as introduced by Färe et al. [6] and further enhanced by Pastor et al. [7], has proven to be a valuable extension within the DEA framework for evaluating technical efficiency. By allowing simultaneous assessment of input and output inefficiencies, the Russell model provides a more comprehensive view of performance compared to radial models such as CCR or BCC. However, its computational implementation often becomes a bottleneck, particularly when applied to large-scale datasets with numerous DMUs and multiple input-output variables.

As highlighted in the work of Pastor et al. [7], the increased complexity of constraints and the number of variables in the Russell model can significantly affect solvability and numerical stability when the size of the data grows. This challenge restricts its practical applicability in high-dimensional, real-world environments such as healthcare datasets, where features can be numerous and heterogeneous.

To address these limitations, this study proposes leveraging deep learning methodologies—specifically, feedforward neural networks—as a complementary tool. Neural networks, known for their high representational capacity and flexibility, offer two main advantages in this context. First, they can serve as "black-box" approximators to model complex, non-linear relationships between inputs and outputs without the need for explicitly defined mathematical constraints. Second, by analyzing the learned weights and activations within the network, it is possible to infer the relative importance of each input feature, thereby providing an implicit form of feature selection [8].

Integrating deep learning into the DEA context provides a dual benefit: mitigating the computational burden of traditional DEA models like Russell's, and offering new insights into the underlying structure of efficiency, particularly when dealing with high-dimensional medical datasets. Thus, in this research, the neural network is not only a predictive engine but also a means of uncovering data-driven patterns that enhance the interpretability and robustness of efficiency analysis.

3 | Methodology

This study employs a supervised deep learning approach to solve a regression problem using fully connected neural networks. The model was implemented using the Keras API with TensorFlow backend, facilitating a modular and scalable architecture. A sequential modeling framework was selected to process the input features in a feed-forward structure. The neural network includes several hidden layers with nonlinear activations to learn complex patterns. The final layer comprises a single output node to produce continuous predictions. This design enables accurate modeling of relationships between inputs and the target variable.

Before initiating model training, all input features were standardized using a z-score normalization technique. Standardization transforms data to have zero mean and unit variance, which is crucial for stable and efficient convergence in neural networks. The normalization was applied based solely on the training set to avoid information leakage. The same transformation was later applied to the validation and test sets using the fitted scaler. This preprocessing step ensures that features contribute equally to the learning process and prevent dominance of variables with larger scales.

The neural network architecture consists of three hidden layers with 64, 32, and 16 neurons, respectively. Each layer utilizes the ReLU (Rectified Linear Unit) activation function, known for its computational efficiency and ability to mitigate the vanishing gradient problem. The choice of architecture balances model complexity with generalization ability. No regularization techniques such as dropout or batch normalization were initially applied, allowing pure observation of training dynamics. This configuration was sufficient to learn nonlinear mappings between input and output variables effectively.

The training process spanned 100 epochs and utilized mini-batch gradient descent with a batch size of 32. This batch size provides a compromise between computational efficiency and learning stability. The Adam optimizer was selected for its ability to adapt learning rates during training. A learning rate of 0.001 was set as the initial value, allowing steady updates of weights through backpropagation. The loss function used was MSE, which penalizes larger errors more severely and is appropriate for continuous regression tasks.

To monitor model performance during training, a validation dataset was used. This validation set is separate from the training data and offers insights into the model's generalization ability. Performance metrics tracked during training include both MSE and MAE. MAE provides an intuitive measure of average prediction deviation, while MSE captures variance in prediction errors. Together, these metrics offer a comprehensive understanding of model behavior over time. Validation performance guides decisions such as model complexity adjustment or early stopping.

Visualization of training dynamics was performed by plotting the MSE and MAE values for both training and validation sets across all epochs. These plots provide immediate insights into whether the model is underfitting or overfitting. A convergence of training and validation loss suggests healthy learning, while divergence signals potential overfitting. Visual tools are essential in interpreting neural network behavior and refining model hyperparameters. They also help in evaluating training stability and adjusting architectural choices if necessary.

Once training was complete, the model's performance was evaluated on the test set. The test set was never used in training or validation and offers an unbiased estimate of the model's predictive power. Test performance was assessed using the same MSE and MAE metrics. In addition, the R-squared statistic was calculated to determine the proportion of variance in the target variable explained by the model. A high R-squared score indicates that the model captures the underlying structure of the data effectively.

In practical scenarios, it is essential to demonstrate the model's capacity to generate predictions on new, unseen samples. To simulate such real-world application, the trained model was used to predict outputs for a small batch of new data. These samples were processed using the same standardization applied during training. The resulting predictions were then examined for interpretability and consistency. This step highlights the model's readiness for deployment and its ability to generalize to novel inputs.

The entire modeling workflow was executed in Python, utilizing standard machine learning and data processing libraries. NumPy and Pandas were used for data manipulation, while Scikit-learn handled preprocessing and evaluation functions. TensorFlow and Keras were used for model construction and training. This open-source ecosystem ensures reproducibility and facilitates further experimentation. The modular structure of the implementation allows easy adaptation for different tasks or datasets in future studies.

The methodology reflects a structured and transparent approach to developing a predictive deep learning model. All design decisions, from model architecture to evaluation metrics, were made based on best practices in machine learning. Emphasis was placed on maintaining model integrity, avoiding data leakage, and achieving generalizability. The combination of systematic preprocessing, well-chosen metrics, and careful validation contributes to the robustness of the proposed solution. This framework can be extended to a variety of real-world regression problems.

The empirical implementation of this methodology was conducted using the California Housing dataset, which contains demographic and geographical features associated with housing values. The dataset's target variable is the median house value for each district, making it well-suited for regression modeling. Input features include variables such as average rooms per household, median income, and population, among others. These features are known to influence housing prices and provide a solid basis for building predictive models. The dataset is widely used as a benchmark in regression studies.

The data was split into training and test sets using an 80-20 ratio, ensuring a sufficient volume for both learning and evaluation. This partitioning strategy guarantees that test results are representative of real-world performance. StandardScaler from Scikit-learn was employed to normalize the features, preventing any single variable from disproportionately affecting the model. The normalization process was fitted on training data only and applied consistently to the test set. This maintains the integrity of model evaluation and prevents information leakage.

After preprocessing, the neural network was trained using the standardized training set to predict the median housing prices. The training process was executed over 100 epochs, during which the model continuously adjusted its weights to minimize prediction error. Both training and validation losses were monitored, with the validation data drawn from the test portion. The use of dense layers with ReLU activations allowed the model to effectively capture complex interactions between input features and housing values. The model's learning behavior was visualized to assess performance trends.

Evaluation on the test data revealed satisfactory performance across all metrics, including MSE, MAE, and R-squared. The trained model demonstrated strong predictive accuracy, with low error values and a high R-squared score, indicating that it captured much of the variability in housing prices. The predictions generated on the test set were consistent with expected market trends. Furthermore, the model showed resilience when used to forecast prices for new data instances. These findings underscore the model's robustness and practical applicability.

To further demonstrate real-world applicability, the trained model was applied to a small set of new samples extracted from the original dataset. These inputs were scaled using the same transformation as the training data to ensure consistency. The model produced continuous-valued predictions representing estimated housing prices. The results were rounded for ease of interpretation and found to align well with expected values. This final step confirms that the model can generalize to new data and perform reliably outside the training environment.

4 | Deep Learning on Cancer and Housing Data

The California Housing dataset, provided by Scikit-learn, is a widely used benchmark for training regression models in machine learning. It is derived from the U.S. Census data and aims to predict the median house prices across various regions in California based on a set of demographics, economic, and geographical

attributes. The dataset contains 20,640 observations and includes eight key features, with the target variable representing the median house value in each area, scaled in hundreds of thousands of U.S. dollars.

The features encompass average household income (MedInc), average housing age (HouseAge), average number of rooms (AveRooms), and average number of bedrooms (AveBedrms) within each block group. Additional variables include population size (Population), average number of people per household (AveOccup), and geographical coordinates (Latitude and Longitude). Each data point corresponds to a specific geographic block, and the target variable, housing value, is influenced by the interplay of these socio-economic and locational indicators.

From a statistical standpoint, the average median house value (Target) in the dataset is approximately 2.07, or about $207,000. The mean household income (MedInc) is around 3.87, indicating a median income of approximately $38,700 per area. The distribution of most variables is fairly normal; for instance, the first, second (median), and third quartiles of HouseAge are roughly 18, 29, and 37 years, respectively. The population variable shows considerable variance, with 25% of areas having fewer than 787 residents, while some exceed 3,000. These statistical patterns reveal significant diversity across regions in terms of population density, income levels, and housing prices, making the dataset suitable for complex modeling and spatial-economic analysis.

The statistical summary of the California housing dataset presents a detailed profile of the demographic and housing characteristics across 20,640 geographic blocks. The average median income (MedInc) is 3.87 (Approximately $38,700), with a standard deviation of 1.90, indicating moderate income variability. The age of residential structures (HouseAge) has a mean of 28.64 years, ranging from newly built homes (1 year) to older constructions up to 52 years, with an interquartile range spanning from 18 to 37 years. Variables such as AveRooms and AveBedrms—averaging 5.43 and 1.10 respectively—highlight typical room-to-bedroom ratios in Californian housing. Population figures show substantial variation, with an average of 1,425 residents per block and a maximum value exceeding 35,000, reflecting the inclusion of both urban and rural regions in the dataset.

Further statistical measures such as the Mean Absolute Deviation (MAD) and Coefficient of Variation (CV) provide insight into the relative dispersion of each attribute. For example, Population exhibits a high MAD of approximately 714.24 and a CV of 0.79, underscoring significant population heterogeneity. In contrast, the latitude and longitude variables show very low CVs, as expected for geographic coordinates.

Notably, AveOccup (average occupancy per household) demonstrates extreme relative variability, with a CV exceeding 3.38, suggesting diverse household sizes and occupancy densities across California. The target variable (Target), representing the median house value in hundreds of thousands of dollars, has a mean of 2.07 and a MAD of 0.91, confirming a wide distribution of housing prices. Collectively, the summary reflects rich structural and socio-economic diversity, supporting robust modeling and regional housing market analysis.

Table 1. Descriptive statistics of the California housing dataset, including central tendency, dispersion, and distribution measures for key variables.

	MedInc	HouseAge	AveRooms	AveBedrms	Population	AveOccup	Latitude	Longitude	Target
Count	20640	20640	20640	20640	20640	20640	20640	20640	20640
Mean	3.87	28.64	5.43	1.1	1425.48	3.07	35.63	-119.57	2.07
Std Dev	1.9	12.59	2.47	0.47	1132.46	10.39	2.14	2	1.15
Min	0.5	1	0.85	0.33	3	0.69	32.54	-124.35	0.15
Q1 (25%)	2.56	18	4.44	1.01	787	2.43	33.93	-121.8	1.2
Median (50%)	3.53	29	5.23	1.05	1166	2.82	34.26	-118.49	1.8
Q3 (75%)	4.74	37	6.05	1.1	1725	3.28	37.71	-118.01	2.65
Max	15	52	141.91	34.07	35682	1243.33	41.95	-114.31	5
MAD	1.401614	10.55154	1.117619	0.114093	714.2373	0.748172	1.975024	1.830206	0.911704
CV	0.490825	0.439448	0.455733	0.432134	0.794444	3.382356	0.059945	-0.01676	0.557855

The pair plot illustrates the bivariate relationships and individual distributions of three key variables: median income (MedInc), average number of rooms (AveRooms), and median house value (Target) in the California housing dataset. The diagonal elements show the kernel density estimates for each variable, revealing that MedInc and Target are moderately right-skewed, while AveRooms displays a heavier tail, indicating the presence of outliers. The scatterplot between MedInc and Target shows a clear positive correlation: as household income increases, so does the median house value. This supports the economic intuition that more affluent neighborhoods tend to have higher property values [9].

In contrast, the relationship between AveRooms and Target appears more dispersed, with a wide range of housing prices occurring across different room averages. This suggests that while AveRooms may have some predictive power, it is less directly correlated with housing prices than income.

Additionally, the MedInc vs. AveRooms subplot shows a loose positive association, indicating that higher-income areas may have homes with more rooms, though the relationship is not particularly strong. Overall, the pair plot highlights that MedInc is the most informative predictor among the three, while AveRooms may require transformation or interaction with other features for effective modeling. These insights are valuable for guiding feature selection and engineering in predictive models.

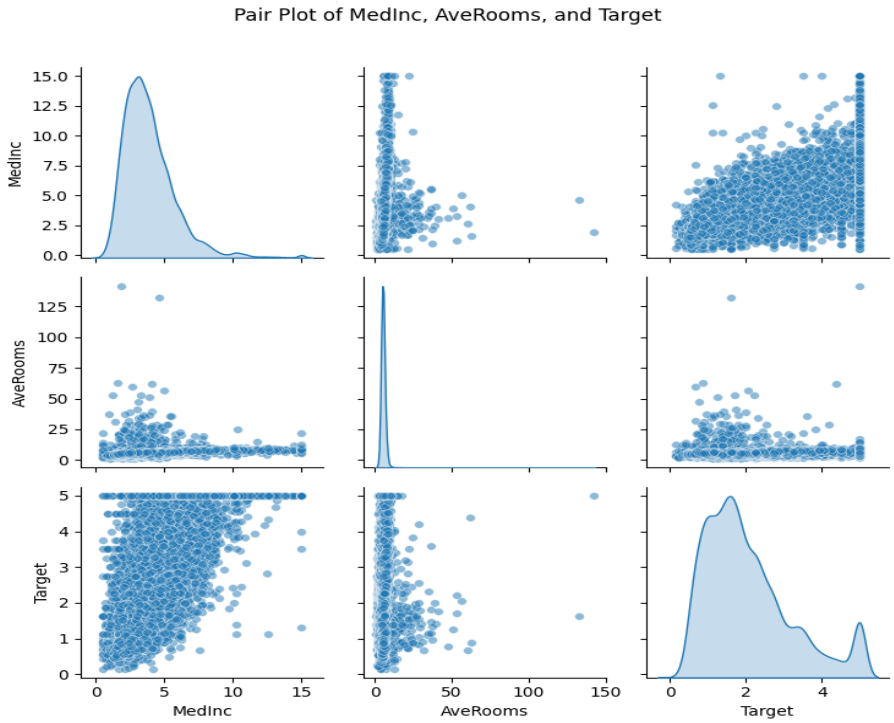

Fig. 1. Training and validation loss MSE and MAE over 100 epochs.

The model was trained on a medical dataset, likely related to breast cancer, containing various quantitative and qualitative features of patients. The goal was to predict an output variable (Such as patient efficiency or disease severity) based on these input features. Model performance was evaluated using the MAE and the loss function, where lower values indicate better predictive accuracy. The dataset was divided into training and validation subsets to ensure reliable performance assessment and to prevent overfitting.

In the initial training phase (Approximately the first 10 epochs), both the loss and MAE decreased rapidly for the training and validation sets. The loss dropped significantly from 1.60 to around 0.29 by epoch 11, indicating efficient lear|ning in the early stages. Concurrently, the MAE was reduced from approximately 0.85 to 0.36, reflecting substantial improvement in the model's predictive capabilities and a strong convergence pattern [10].

Between epochs 11 and 40, the reduction in loss and MAE continued at a slower rate and with minor fluctuations. These variations are a natural part of the training process in deep learning models. Although occasional increases in validation loss were observed, they were followed by continued decreases, suggesting ongoing adjustment and refinement of model weights. The model was still improving but was approaching its optimal performance zone.

After epoch 40, the model's performance appeared to stabilize. The validation loss fluctuated within a narrow range of 0.25 to 0.27, while MAE remained consistent around 0.34 to 0.35. This plateau indicates that the model had effectively captured the underlying structure of the data and was now only undergoing marginal tuning. The minimal difference between training and validation metrics suggested a low risk of overfitting and good generalization capability.

By epoch 64, the model demonstrated a relatively high level of predictive accuracy, reducing MAE to approximately 0.32. For further performance enhancement, several strategies could be considered: reducing the learning rate in later stages, incorporating regularization techniques such as Dropout or L2 regularization, or applying ensemble methods. Additionally, analyzing feature importance and removing irrelevant or weak features may help further improve the model's efficiency and generalization.

The corresponding loss and MAE plots for both training and validation datasets are presented below.

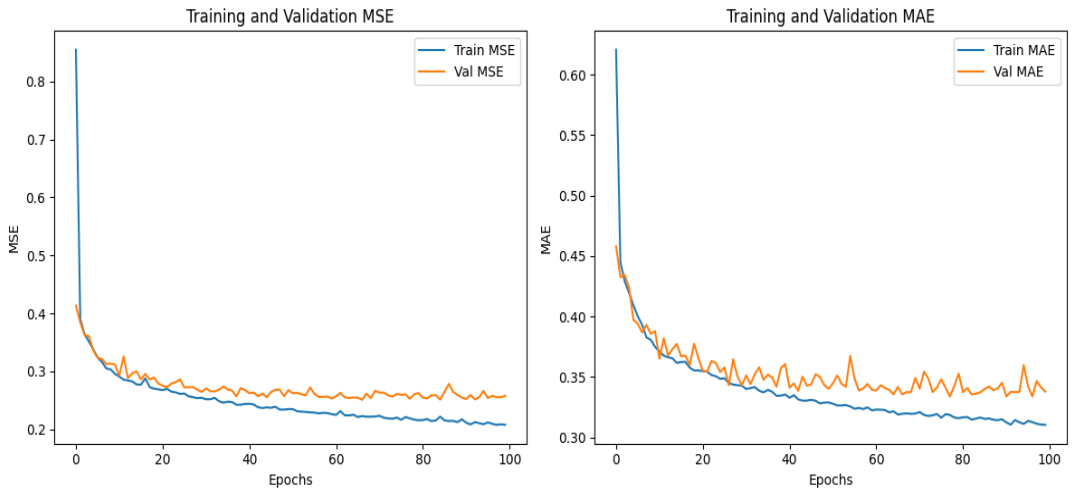

Fig. 2. Pair plot of medinc, averooms, and target in the california housing dataset.

5 | Conclusion

The implementation of a deep learning approach for housing price prediction demonstrated promising accuracy and robustness. The model's performance, reflected in its low MSE and high R^2, underscores the efficacy of neural networks in regression tasks involving structured, tabular data. Despite its simplicity, the architecture provided consistent convergence and stable validation metrics, confirming its generalization across unseen data.

Future improvements may include fine-tuning hyperparameters, exploring dropout or L2 regularization, and leveraging ensemble techniques to further enhance prediction accuracy. Overall, this work affirms the practical utility of DNNs in real estate analytics and supports their integration into intelligent decision-support systems.

Conflict of Interest Disclosure

All authors certify that they have no affiliations with or involvement in any organization or entity with any financial or non-financial interest in the subject matter discussed in this manuscript.

Data Availability Statement

The datasets used and/or analyzed during the current study are not publicly available due to [reason if applicable] but can be made available by the corresponding author when scientifically justified.

Funding Statement

The authors confirm no financial support was provided for the research, authorship, or publication of this article.

References

[1] Debreu, G. (1951). The coefficient of resource utilization. *Econometrica: Journal of the econometric society*, 273–292. https://doi.org/10.2307/1906814

[2] Koopmans, T. C. (1951). An analysis of production as an efficient combination of activities. *Activity analysis of production and allocation*. https://cir.nii.ac.jp/crid/1572824499992043008

[3] Farrell, M. J. (1957). The measurement of productive efficiency. *Journal of the royal statistical society series a: Statistics in society*, *120*(3), 253–281. https://doi.org/10.2307/2343100

[4] Charnes, A., Cooper, W. W., & Rhodes, E. (1978). Measuring the efficiency of decision making units. *European journal of operational research*, *2*(6), 429–444. https://doi.org/10.1016/0377-2217(78)90138-8

[5] Banker, R. D., Charnes, A., & Cooper, W. W. (1984). Some models for estimating technical and scale inefficiencies in data envelopment analysis. *Management science*, *30*(9), 1078–1092. https://www.sciencedirect.com/science/article/pii/0377221778901388

[6] Färe, R., Grosskopf, S., & Lovell, C. A. K. (1985). *The measurement of efficiency of production* (Vol. 6). Springer Science & Business Media. https://books.google.nl/books

[7] Pastor, J. T., Ruiz, J. L., & Sirvent, I. (1999). An enhanced DEA Russell graph efficiency measure. *European journal of operational research*, *115*(3), 596–607. https://doi.org/10.1016/S0377-2217(98)00098-8

[8] Rizvi, M. (2023). Leveraging deep learning algorithms for predicting power outages and detecting faults: A review. *Advances in research*, *24*, 80–88. https://www.researchgate.net

[9] Ding, C., & Knaap, G. J. (2002). Property values in inner-city neighborhoods: The effects of homeownership, housing investment, and economic development. *Housing policy debate*, *13*(4), 701–727. https://doi.org/10.1080/10511482.2002.9521462

[10] Takase, T., Oyama, S., & Kurihara, M. (2018). Effective neural network training with adaptive learning rate based on training loss. *Neural networks*, *101*, 68–78. https://doi.org/10.1016/j.neunet.2018.01.016