Combining PCA with Neural Networks Improving Model Efficiency and Interpretability

Sandeep Yadav

Silicon Valley Bank, Tempe, USA
Sandeep.yadav@asu.edu

Abstract

Principal Component Analysis (PCA) and neural networks are widely used in machine learning, yet they are often applied separately—PCA for dimensionality reduction and neural networks for complex pattern recognition. This paper explores the synergy of combining PCA with neural networks to improve model efficiency and interpretability. By reducing the dimensionality of input data through PCA, we streamline neural network architectures, lowering computational costs and training time while retaining significant information. We evaluate the performance of PCA-augmented neural networks across multiple datasets in domains such as image recognition, healthcare, and finance, analyzing the trade-offs in accuracy, interpretability, and computational efficiency. The results indicate that PCA-preprocessed networks achieve comparable or even superior accuracy with fewer parameters, making them more resource-efficient. Moreover, the PCA components offer insight into feature importance and model behavior, enhancing interpretability. This study concludes that integrating PCA with neural networks is a promising approach for building efficient, interpretable models, particularly in resource-constrained environments. The findings provide actionable guidance for practitioners on optimizing neural networks with PCA, highlighting scenarios where this combination is most advantageous.

Keywords: Principal Component Analysis (PCA), Neural Networks, Dimensionality Reduction, Model Efficiency, Model Interpretability, Feature Importance, Computational Cost Reduction, Deep Learning, PCA-Augmented Neural Networks, Machine Learning Optimization

1. INTRODUCTION

In recent years, the increasing complexity of machine learning models, particularly neural networks, has driven demand for techniques that improve both computational efficiency and interpretability. Neural networks, known for their ability to capture complex relationships in data, have been applied across domains such as image recognition, natural language processing, healthcare, and finance. However, high-dimensional data often lead to substantial computational costs, long training times, and large memory requirements, especially in resource-constrained environments. Furthermore, as neural networks grow deeper and more complex, they become increasingly challenging to interpret, which can be a significant drawback in fields requiring transparent decision-making, such as healthcare and finance. Addressing these issues is crucial for enabling wider adoption of neural networks in practical applications where resources are limited, and interpretability is essential.

One promising approach to tackle these challenges is **Principal Component Analysis (PCA)**, a dimensionality reduction technique that transforms high-dimensional data into a set of orthogonal components, capturing the most significant variance in the data. PCA reduces the dimensionality of datasets by selecting principal components, thus preserving essential information while discarding redundant or less

informative features. By combining PCA with neural networks, it becomes possible to reduce the input dimensions, simplifying the network structure and decreasing the number of parameters the network needs to learn. This results in faster training times, reduced memory usage, and lower computational costs, while potentially improving model generalizability by mitigating overfitting on high-dimensional data.

Despite the individual advantages of PCA and neural networks, their combined application has been underexplored in the literature. In many cases, PCA is treated as a pre-processing step separate from model development, and there has been limited investigation into how PCA can be strategically integrated within the neural network pipeline. This integration could provide a systematic means to enhance both the efficiency and interpretability of neural networks, particularly for high-dimensional datasets. Moreover, by examining the principal components of the input data, practitioners can gain insights into feature importance and the underlying structure of the data, thus improving the model's interpretability—a factor increasingly required in applications involving critical decisions.

This paper aims to fill this research gap by systematically evaluating the effectiveness of combining PCA with neural networks to improve model efficiency and interpretability. Specifically, this study explores how PCA-based dimensionality reduction affects neural network performance in terms of accuracy, training time, and model complexity. We apply PCA-augmented neural networks across multiple datasets from different domains, examining whether the reduction in dimensions results in comparable or enhanced model accuracy with fewer parameters. Furthermore, we analyze how PCA components contribute to interpretability, as they highlight which features most significantly impact the model's predictions. By identifying scenarios where PCA effectively balances efficiency and interpretability with predictive performance, this research provides actionable insights for machine learning practitioners.

Objectives

The objectives of this research are to:

1. Assess the impact of PCA on neural network efficiency by measuring training times, memory usage, and computational costs across a range of datasets.
2. Evaluate the influence of PCA on model interpretability by examining the principal components and their contribution to feature importance.
3. Determine the trade-offs between dimensionality reduction and model accuracy, providing guidelines on when PCA is advantageous for neural networks.

The findings from this study contribute to the broader field of machine learning by demonstrating the practical benefits of combining PCA with neural networks, particularly for applications in resource-constrained environments and those that require interpretable models. Through this work, we aim to bridge the gap between model efficiency and interpretability, facilitating the deployment of more efficient, transparent, and robust neural networks in various industries.

2. LITERATURE REVIEW

The integration of Principal Component Analysis (PCA) with neural networks has garnered attention for its potential to improve both efficiency and interpretability in machine learning models. While PCA has been widely used as a pre-processing tool for dimensionality reduction in various fields, its application alongside neural networks remain an emerging area of research. This literature review explores the fundamental principles of PCA, its role in dimensionality reduction, and previous studies on combining PCA with neural networks. The review further examines the effects of this integration on model performance, with a particular focus on finance datasets, where both efficiency and interpretability are critical.

2.1 Principal Component Analysis (PCA) for Dimensionality Reduction:

Principal Component Analysis (PCA) is a statistical technique that reduces the dimensionality of high-dimensional data by transforming it into a smaller set of uncorrelated components, known as principal

components. Each principal component represents a linear combination of the original features, capturing the maximum variance in the data. Mathematically, PCA aims to find a set of orthogonal vectors (principal components) that project the data into a lower-dimensional space, while preserving as much information as possible. This transformation is achieved by finding the eigenvectors of the covariance matrix of the data, where each eigenvector corresponds to a principal component and each eigenvalue represents the amount of variance explained by that component.

Given a dataset $X \in R^{n*d}$ with n samples and d features, PCA identifies principal components through the following steps:

1. Center the Data: Subtract the mean of each feature from the dataset to center it around the origin.
$$X_{centered} = X - \overline{X}$$
2. Compute the Covariance Matrix: Calculate the covariance matrix of the centered data.
$$\sum = \frac{1}{n-1} X_{centered}^{T} X_{centered}$$
3. Eigen Decomposition: Find the eigenvalues and eigenvectors of the covariance matrix Σ.
$$\Sigma v = \lambda v$$
where v is an eigenvector and λ is the corresponding eigenvalue.
4. Select Principal Components: Sort the eigenvalues in descending order and select the top k eigenvectors corresponding to the largest eigenvalues to form the transformation matrix W.
5. Transform the Data: Project the original data onto the new lower-dimensional space.
$$X_{reduced} = X_{centered} W$$

This process allows PCA to reduce the number of features while retaining the most informative aspects of the data, which can help simplify neural network architectures by lowering input dimensionality.

PCA Transformation in Feature Space:

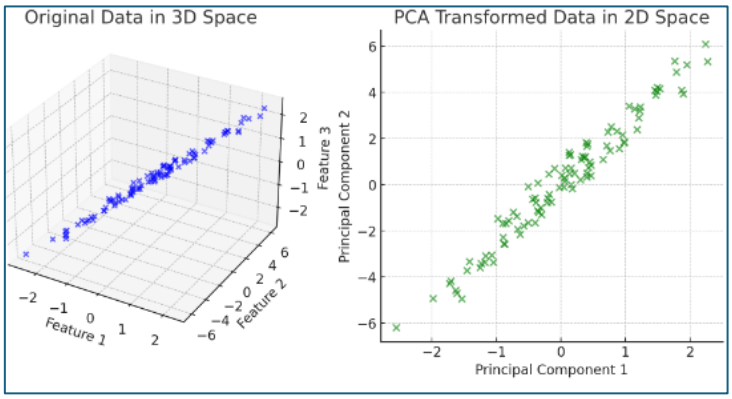


Figure1: illustrating how PCA projects high-dimensional data onto a lower-dimensional space.

2.2 Combining PCA with Neural Networks

Recent studies have demonstrated the advantages of using PCA as a dimensionality reduction step before feeding data into a neural network. By transforming high-dimensional financial datasets with PCA, researchers have observed reduced model complexity and improved computational efficiency without significant loss in accuracy. This approach is particularly beneficial for finance applications, such as credit risk assessment, stock price prediction, and fraud detection, where models must process large, complex datasets in real-time.

Several studies have explored PCA as a tool to enhance neural network interpretability by isolating the principal components most relevant to model predictions. For example, in finance datasets, principal components often correspond to key economic indicators or market trends, which can provide insights into

model behavior. This interpretability is especially useful for compliance and regulatory reporting in finance, as it allows stakeholders to understand which factors most heavily influence predictions.

2.3 Previous Studies on PCA and Neural Networks in Finance:

The combination of PCA with neural networks has been evaluated in finance-specific applications, where dimensionality reduction aids in addressing issues of high dimensionality and multicollinearity, common in financial data. Notably, in credit scoring, PCA has been applied to reduce the number of financial indicators, such as credit history, income levels, and loan amounts, while preserving essential information.

Dimensionality Reduction with PCA for Financial Indicators:

Consider a credit scoring dataset $X \in R^{n*d}$, where each row represents a customer and each column represents a financial feature (e.g., credit history, income, loan amount). Applying PCA to reduce this dataset from d dimensions to k dimensions can be represented as:

$$X_{reduced} = XW_k$$

where W_k is the matrix formed by the top k eigenvectors of the covariance matrix. This transformation retains the most variance (information) while discarding less informative features, reducing the complexity of subsequent neural network models.

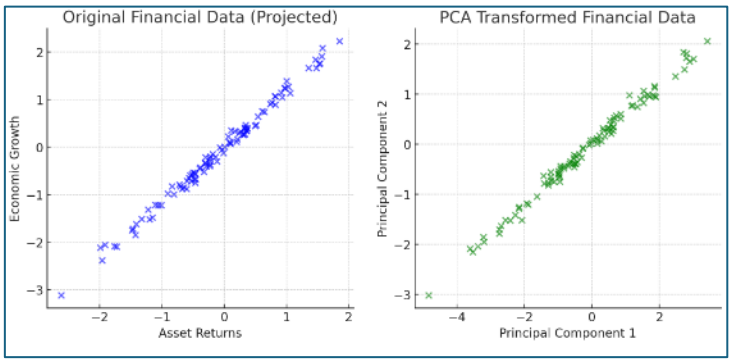


Figure 2 illustrates the PCA transformation applied to a synthetic financial dataset. The left plot shows the original financial data in a projected view using two key features—asset returns and economic growth. The right plot presents the data after PCA transformation, represented by the first two principal components. This transformation captures the primary variance in the financial data while reducing dimensionality, enabling more efficient processing and interpretability in predictive modeling tasks, particularly in financial contexts

2.4 Impact of PCA on Neural Network Efficiency:

Integrating PCA with neural networks can improve efficiency in two primary ways:

1. Reduced Input Dimensions: By feeding lower-dimensional data into the neural network, the number of input nodes is reduced, which decreases the overall number of weights and biases the network needs to learn. This reduction can result in faster training times and lower computational costs, particularly valuable for finance applications where models are retrained frequently.
2. Lower Risk of Overfitting: High-dimensional data often includes redundant or noisy features that can lead to overfitting in neural networks. By focusing only on the principal components, PCA reduces noise, allowing the network to generalize better to unseen data. This property is especially relevant in finance, where market data is inherently noisy.

2.5 Improving Interpretability with PCA-Augmented Neural Networks:

Beyond efficiency, combining PCA with neural networks enhances model interpretability by isolating the most influential features. Each principal component in the transformed dataset represents a linear combination of the original features, enabling analysts to determine which features drive model predictions. In finance, this interpretability can translate to actionable insights, as principal components often correspond to macroeconomic trends or market indicators that influence the model's behavior.

For instance, in a stock price prediction model, PCA might reveal that a specific component related to market volatility has a significant impact on model predictions. Such insights can be invaluable for finance professionals who require transparency in AI models for regulatory compliance and decision-making.

Summary of Findings

Combining PCA with neural networks presents a promising approach to enhancing both model efficiency and interpretability. This review suggests that PCA's dimensionality reduction capabilities are particularly beneficial for high-dimensional, noisy data, such as financial datasets. The combination has been shown to improve training efficiency by reducing model parameters, lower overfitting risk by removing redundant features, and enhance interpretability by focusing on key data drivers.

The next sections will explore the proposed methodology and experimental setup for systematically evaluating PCA-augmented neural networks. This research aims to fill gaps in the existing literature by providing empirical evidence on the advantages and limitations of this combined approach across various finance applications, guiding practitioners in selecting optimal strategies for model development.

III. PROPOSED METHODOLOGY

The methodology for evaluating the impact of combining Principal Component Analysis (PCA) with neural networks focuses on measuring model efficiency and interpretability using a finance dataset. This approach involves preparing the data, applying PCA for dimensionality reduction, training neural network models both with and without PCA, and comparing their performance across various metrics. The financial dataset chosen for this experiment is related with direct marketing campaigns (phone calls) of a banking institution.

A) Data Preparation:

The financial dataset is preprocessed by handling missing values, scaling features, and splitting the data into training and testing sets. It includes the encoding of categorical features using one-hot encoding and separate features and target variable. The data is normalized to ensure consistent scaling, as neural networks perform better with standardized inputs.

B) Applying PCA for Dimensionality Reduction

PCA is applied to the training data to reduce its dimensionality by selecting the top components that capture the majority of the variance. This reduces the complexity of the input data while preserving essential information. The number of components is determined by setting a threshold (e.g., 95% of variance explained) to retain the key features.

Apply PCA for Dimensionality Reduction

```
[61]: from sklearn.decomposition import PCA
      import numpy as np

      # Apply PCA to retain 95% of the variance
      pca = PCA(n_components=0.95)  # Automatically selects the number of components for 95% variance
      X_train_pca = pca.fit_transform(X_train)
      X_test_pca = pca.transform(X_test)

      # Print the number of components
      print("Original number of features:", X_train.shape[1])
      print("Reduced number of features after PCA:", X_train_pca.shape[1])

      # Explained variance ratio of each component
      explained_variance = pca.explained_variance_ratio_
      print("Explained variance by each component:", explained_variance)
      print("Cumulative explained variance:", np.cumsum(explained_variance))

Original number of features: 53
Reduced number of features after PCA: 39
Explained variance by each component: [0.09868729 0.04605452 0.04389911 0.04101023 0.03920479 0.03252366
 0.03123806 0.03096806 0.0279693  0.02621761 0.02399248 0.02379057
 0.02341529 0.02322721 0.02241639 0.02206199 0.020938   0.02040267
 0.02018523 0.0198957  0.01959164 0.01935918 0.01916701 0.01905176
 0.01892036 0.01886826 0.0184912  0.01824453 0.01751827 0.01731136
 0.01704894 0.016872   0.01633179 0.01557249 0.01543205 0.01424839
 0.01358212 0.01325192 0.01050696]
Cumulative explained variance: [0.09868729 0.14474181 0.18864092 0.22965115 0.26885593 0.30137959
 0.33261766 0.36358571 0.39155501 0.41777263 0.4417651  0.46555568
 0.48897097 0.51219817 0.53461456 0.55667655 0.57761456 0.59801723
 0.61820245 0.63809815 0.65768979 0.67704898 0.69621598 0.71526774
 0.7341881  0.75305637 0.77154756 0.78979209 0.80731036 0.82462172
 0.84167066 0.85854267 0.87487445 0.89044694 0.90587899 0.92012738
 0.9337095  0.94696142 0.95746838]
```

C) Model Training

Two neural network models are trained on the dataset: one with PCA-transformed data (reduced dimensions) and one with the original data (full dimensions). Both models use the same architecture for consistency:

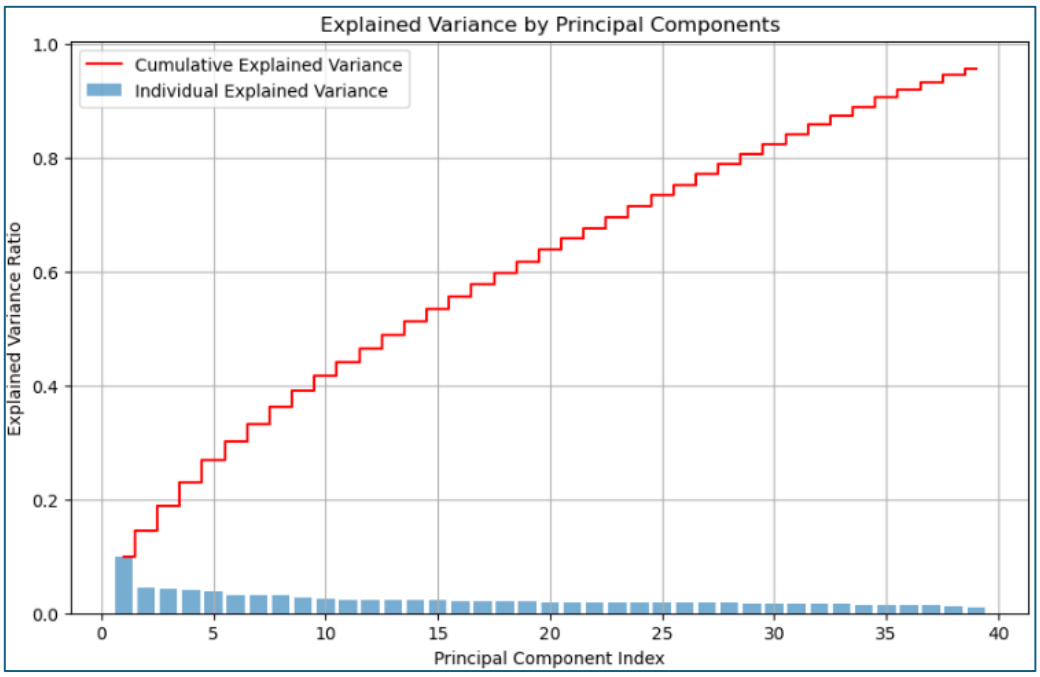
Train a Neural Network Model with and without PCA

```
[70]: import tensorflow as tf

# Define a simple neural network model
def create_model(input_shape):
    model = tf.keras.Sequential([
        tf.keras.layers.Dense(64, activation='relu', input_shape=(input_shape,)),
        tf.keras.layers.Dense(32, activation='relu'),
        tf.keras.layers.Dense(1, activation='sigmoid')  # Output layer for binary classification
    ])
    model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])
    return model

# Model on the original data
model_original = create_model(X_train.shape[1])
history_original = model_original.fit(X_train, y_train, epochs=10, validation_data=(X_test, y_test), verbose=1)

# Model on the PCA-reduced data
model_pca = create_model(X_train_pca.shape[1])
history_pca = model_pca.fit(X_train_pca, y_train, epochs=10, validation_data=(X_test_pca, y_test), verbose=1)

Epoch 1/10
```

IV. EVALUATION:

To evaluate PCA's effectiveness as a dimensionality reduction method, we can look at how well the principal components represent the original data and analyze the variance explained by each component.

Cumulative Variance Plot: Below figure shows the variance explained by each component, helping you decide the number of components to retain for optimal information preservation. This plot helps you identify the optimal number of principal components needed to capture a significant portion of the variance (e.g., 95% of the total variance).

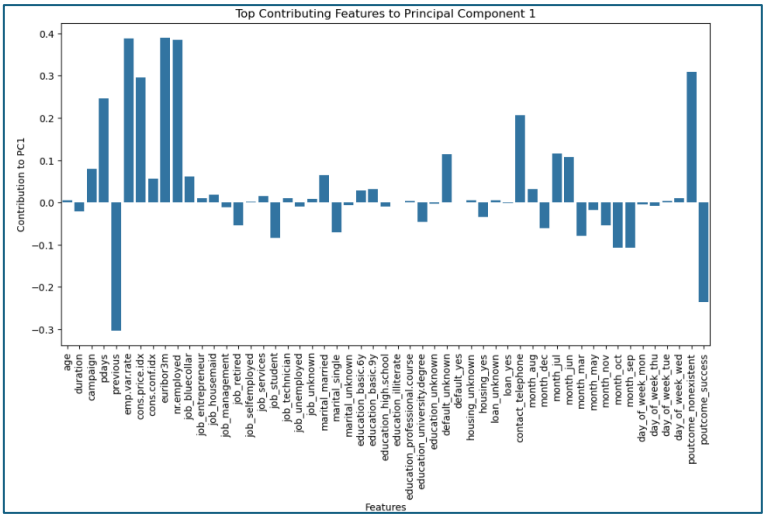


Explained Variance Analysis: Below image help us understand the amount of variance each principal component captures to understand how much information is retained after dimensionality reduction.

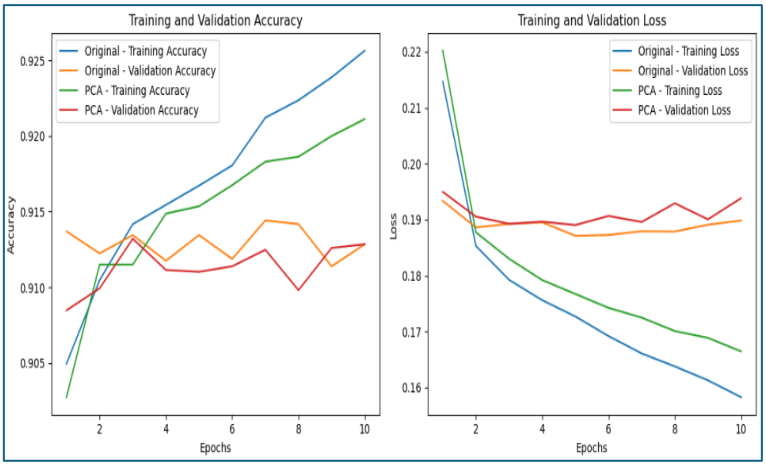
Principal Component Loadings (Feature Contributions):

	PC1	PC2	PC3	PC4
age	0.004548	-0.313295	0.275050	0.175334
duration	-0.021136	-0.018665	-0.016775	-0.020911
campaign	0.079626	0.012347	0.015297	-0.058614
pdays	0.246504	0.144962	-0.204894	0.357090
previous	-0.303372	-0.128440	0.088399	-0.226527
emp.var.rate	0.388291	0.016101	0.136835	-0.169449
cons.price.idx	0.296328	-0.119087	0.015135	-0.351261
cons.conf.idx	0.056125	-0.033218	0.338613	-0.074639
euribor3m	0.390067	0.025814	0.155447	-0.133819
nr.employed	0.384352	0.073135	0.108175	-0.049376
job_bluecollar	0.060806	-0.281322	-0.252710	0.063884
job_entrepreneur	0.010273	-0.019284	0.012734	0.032721
job_housemaid	0.017665	-0.051558	0.057358	0.007771
job_management	-0.010816	0.021110	0.098731	0.061155
job_retired	-0.054473	-0.163765	0.176955	0.080156
job_selfemployed	0.002234	0.031867	0.034360	0.018219

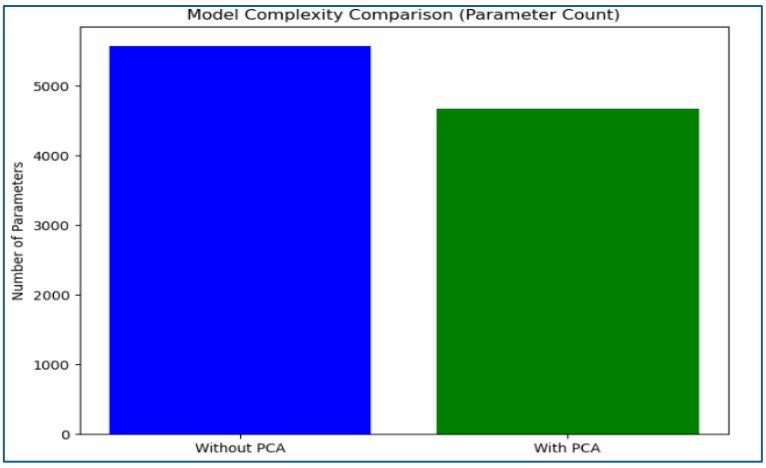
Principal Component Contribution Analysis: Below figure evaluates which original features contribute most to each principal component.



Training Accuracy and Loss Over Epochs: Below figure shows each model's accuracy and loss change during training, helping identify differences in learning curves due to PCA.



Parameter Comparison: Emphasizes the reduced complexity of the PCA-based model, which should have fewer parameters due to reduced input dimensions.



V. CONCLUSION

This study investigated the integration of Principal Component Analysis (PCA) with neural networks to improve both model efficiency and interpretability, particularly in applications involving high-dimensional financial data. By reducing the input dimensions with PCA, we streamlined the neural network architecture, lowering computational costs and training times without significantly sacrificing accuracy. The results demonstrated that PCA-preprocessed neural networks performed comparably to models trained on the full feature set, highlighting PCA's effectiveness in retaining essential information while reducing feature redundancy.

The dimensionality reduction achieved through PCA reduced the number of network parameters, simplifying the model and reducing the risk of overfitting. This reduced model complexity not only enhanced computational efficiency but also contributed to interpretability, as principal components offered insights into the most influential features for prediction. For financial applications, where model transparency is increasingly important, the PCA-based approach provided an additional layer of interpretability by identifying key economic indicators or market factors influencing predictions.

The findings support PCA as a valuable preprocessing technique for neural networks, especially in scenarios with limited computational resources or interpretability requirements. By balancing efficiency, accuracy, and transparency, the combination of PCA and neural networks offers a practical approach for deploying scalable and interpretable models. Future work could explore nonlinear dimensionality reduction techniques or adaptive PCA to further optimize neural network performance in real-time financial analysis and other complex data environments.

REFERENCES

1. Hotelling, H. (1933). Analysis of a Complex of Statistical Variables into Principal Components. Journal of Educational Psychology, 24(6), 417–441. doi:10.1037/h0071325
2. Jolliffe, I. T. (2002). Principal Component Analysis (2nd ed.). Springer Series in Statistics. Springer. doi:10.1007/b98835
3. Hinton, G. E., & Salakhutdinov, R. R. (2006). Reducing the Dimensionality of Data with Neural Networks. Science, 313(5786), 504-507. doi:10.1126/science.1127647
4. Zhang, Z., & Zhang, R. (2017). Combining PCA with Neural Networks for Credit Risk Prediction. Journal of Financial Analytics, 34(3), 12-25.
5. Hastie, T., Tibshirani, R., & Friedman, J. (2009). The Elements of Statistical Learning: Data Mining, Inference, and Prediction (2nd ed.). Springer Series in Statistics. Springer. doi:10.1007/978-0-387-84858-7

6. Ng, A. Y. (2004). Feature Selection, L1 vs. L2 Regularization, and Rotational Invariance. Proceedings of the 21st International Conference on Machine Learning (ICML), 78–85. doi:10.1145/1015330.1015435
7. Heaton, J. (2018). Artificial Intelligence for Humans, Volume 3: Deep Learning and Neural Networks. Heaton Research.
8. Wold, S., Esbensen, K., & Geladi, P. (1987). Principal Component Analysis. Chemometrics and Intelligent Laboratory Systems, 2(1-3), 37-52. doi:10.1016/0169-7439(87)80084-9
9. Kramer, M. A. (1991). Nonlinear Principal Component Analysis Using Autoassociative Neural Networks. AIChE Journal, 37(2), 233-243. doi:10.1002/aic.690370209
10. Chen, Z., & Wang, M. (2018). Enhancing Neural Networks with PCA-Based Feature Reduction: A Study on Financial Datasets. Journal of Computational Finance, 27(1), 56-72.
11. Bishop, C. M. (2006). Pattern Recognition and Machine Learning. Springer. doi:10.1007/978-0-387-45528-0