Dimensionality Reduction in Machine Learning: A Comprehensive Analysis of Techniques and Applications

Martial Domche, M.Sc.
Data Zoomcamp
Email:martialdomche@gmail.com

Abstract: Dimensionality reduction is a fundamental concept in machine learning, particularly when working with high-dimensional datasets that are susceptible to the curse of dimensionality. This challenge arises when the number of features significantly exceeds the number of observations, often leading to overfitting and poor model performance. Dimensionality reduction techniques play a dual role in enhancing both computational efficiency and model generalization.

This paper provides a detailed analysis of both linear and nonlinear dimensionality reduction methods, including Principal Component Analysis (PCA), t-SNE, Uniform Manifold Approximation and Projection (UMAP), and L1 regularization. These techniques are applied to various real-world problems, such as credit card fraud detection, image recognition, and natural language processing. For instance, PCA effectively reduces noise while preserving the most significant patterns in data, whereas t-SNE and UMAP excel in visualizing complex structures in lower dimensions.

In addition, a case study highlights the use of correlation matrices to identify and eliminate weakly correlated features, thereby improving computational efficiency and enhancing model accuracy. This study also explores the trade-offs between interpretability and complexity, guiding the selection of optimal techniques tailored to specific datasets.

Overall, this paper offers practical insights and recommendations for researchers and practitioners seeking to leverage dimensionality reduction in diverse machine learning applications.

Keywords: Dimensionality reduction, Curse of dimensionality, Principal Component Analysis, t-SNE, UMAP, Feature selection.

1 Introduction

High-dimensional datasets often include redundant or irrelevant features, leading to challenges like overfitting, inefficiency, and reduced interpretability. Dimensionality reduction addresses these issues by simplifying the feature space while retaining essential information [1].

In this work, we emphasize the role of **correlation analysis** as a feature selection technique, particularly for structured datasets. Using the credit card fraud detection case study, we illustrate:

1. The mathematical foundation of dimensionality reduction [2].
2. How correlation matrices aid in feature elimination [3].
3. Quantitative gains in performance after feature reduction [4].

2 Mathematical Foundation of Dimensionality Reduction

Dimensionality reduction techniques operate by mapping a dataset $X \in \mathbb{R}^{n*d}$, where n is the number of samples and d is the feature dimensionality, into a lower-dimensional space $Y \in \mathbb{R}^{n*k}$ where $k \ll d$. Let $f : \mathbb{R}^d \to \mathbb{R}^k$ represent the transformation function [5].

2.1 Correlation Analysis

Correlation quantifies the linear relationship between two variables, denoted by the Pearson correlation coefficient (r). The coefficient r is defines as :

$$r = \frac{Cov\ (X, Y)}{\sigma_X \sigma_Y}, \tag{1}$$

Where:

- $Cov(X, Y)$: Covariance between X and Y,
- σ_X, σ_Y: Standard deviations of $X\ and\ Y$ [6].

Interpretation of correlation coefficients

- $r = 1$: Perfect positive correlation. When X increases Y increases proportionally.
- $r = 1$: No linear correlation. Changes in X do not predict changes in Y .
- $r = 1$: Perfect negative correlation. When X increases, Y decreases proportionally [7].

Thresholds for feature selection

Feature are considered:

- Strongly correlated: $|r| > 0.7$ (significant relationship).
- Weakly correlation: $|r| < 0.3$ (minimal or no relationship with the target) [8].

2.2 Principal Component Analysis (PCA)

PCA is a linear method used for dimensionality reduction by finding a set of orthogonal axes (principal components) that capture the maximum variance in the dataset. This is done through the following steps:

a) Compute the covariance matrix Σ of the data:

$$\Sigma = \frac{1}{n-1} X^T X \tag{2}$$

Where X is the data matrix with rows representing data samples and columns representing features [9].

b) Eigenvalue decomposition: The covariance matrix Σ is decomposed into eigenvalues λ_i and eigenvectors v_i. These eigenvectors form the new coordinates system, where the first eigenvector corresponds to the direction of maximum variance [10].

c) Select top k components: The top k eigenvectors (corresponding to the largest eigenvalues) are selected to form a projection matrix P.

d) Projection of data: The data is then projected onto the top k eigenvectors, reducing the dimensionality:

$$X_{new} = XP \tag{3}$$

Where P is the matrix of eigenvectors.

Advantages of PCA:

- Variance maximization: PCA ensures that the new dimensions capture the maximum variance in the data [11].
- Efficiency: It is computationally efficient for datasets with a large number of features [12].
- Limitations: PCA assumes linearity and works best with Gaussian-distributed data [13].

Applications:

- Image compression [14].
- Data visualization (2D/3D scatter plots) [15].
- Noise reduction in signal processing [16].

2.2.2 Linear Discriminant Analysis (LDA)

LDA is a supervised dimensionality reduction technique that seeks to find a projection that maximizes class separability. The optimization criterion for LDA is the Fisher criterion, given by:

$$J(W) = \frac{W^T S_B W}{W^T S_W W} \tag{4}$$

Where:

- S_B is the between-class scatter matrix,
- S_W is the within-class scatter matrix,
- W is the projection matrix.

LDA maximizes the ratio of between-class variance to within-class variance, making it ideal for classification tasks where the goal is to distinguish between different classes [17].

Applications:

- Face recognition: LDA is widely used in face recognition systems to project high-dimensional face images into a lower-dimensional space while preserving class separability [18].
- pam detection in emails: LDA has been applied to classify emails as spam or non-spam by reducing the dimensionality of text features while maintaining discriminative power [19].

2.3 Nonlinear Methods

2.3.1 t-SNE (t-Distribution Stochastic Neighbor Embedding)

t-SNE is a powerful nonlinear technique primarily used for visualization, especially when the data is high-dimensional and contains complex relationships that linear methods like PCA may not capture [17].

1. Compute pairwise similarities: In the high-dimensional space, t-SNE calcules pairwise similarities between points using a Gaussian distribution:

$$P_{ij} = \frac{\exp\left(-\|x_i - x_j\|^2 / 2\sigma^2\right)}{\Sigma_{k \neq i} \exp(-\|x_i - x_k\|^2 / 2\sigma^2)} \tag{5}$$

Where P_{ij} represents the similarity between points x_i and x_j.

2. Low-dimensional embedding: t-SNE then aims to find a low-dimensional embedding $Y \in \mathbb{R}^k$ such that the pairwise similarities Q_{ij} in the low-dimensional space are as close as possible to high-dimensional similarities. The distribution in the low-dimensional space is a student's t-distribution:

$$Q_{ij} = \frac{\left(1 + \|y_i - y_j\|^2\right)^{-1}}{\Sigma_{k \neq i}(1 + \|y_i - y_k\|^2)^{-1}} \tag{6}$$

3. Optimization: The method minimizes the kullback-Leibler divergence between the high- and low-dimensional distribution using gradient descent [18].

Advantages of t-SNE:

- Preserves local structure: It is particularly useful for visualizing clusters and nonlinear relationships in data [19].
- Nonlinear embeddings: t-SNE can capture complex structures that linear methods like PCA miss [20].

Limitations:

- Computational complexity: t-SNE has $O(n^2)$ time complexity, making it slow for large datasets [21].
- Interpretability: The low-dimensional space created by t-SNE does not have a direct interpretation, as the axes in the embedding are not meaningful [22].

Applications:

- Visualizing high-dimensional biological data [23].
- Exploratory data analysis in unsupervised learning [24].

2.3.2 Linear Discriminant Analysis (LDA)

UMAP is a more recent nonlinear dimensionality reduction technique that aims to preserve both local and global structures of the data. UMAP uses a combination of manifold theory and topological data analysis to learn a low-dimensional representation [25].

a) Local structure representation: UMAP first builds a graph that represents the local relationship between data points. It uses nearest neighbors to capture the topology of the data [26].

b) Nonlinear optimization: UMAP minimizes an objective function that balances local proximity and global structure preservation:

$$C = \sum_{(i,j)\epsilon E} w_{ij} \left(d\left(X_i, X_j\right) - d(Y_i, Y_j)\right)^2 \qquad (7)$$

Where w_{ij} are the weights in the graph, and $d(X_i, X_j)$ and $d(Y_i, Y_j)$ are the distances between points in the original and lower-dimensional space, respectively [27].

Advantages of UMAP:

- Scalability: UMAP has linear time complexity$O(nlogn)$, making it much faster than t-SEN for large datasets [28].
- Preserves both local and global structure: It captures both the local neighborhood and the global relationships between points [29].

Limitations:

- Interpretability: Like t-SNE, the low-dimensional representation lacks clear interpretability [30].
- Parameter tuning: UMAP's performance can depend on the choice of parameters [31].

Applications:

- High-dimensional data visualization [32].
- Clustering and manifold learning [33].

2.4 Regularization Techniques

2.4.1 Lasso (L1 regularization)

Lasso solves the regression problem:

$$argmin_{\beta} ||y - X\beta||_2^2 + \lambda ||\beta||_1 \qquad (8)$$

Where the L_1 norm $||\beta||_1$ encourages sparsity by shrinking some coefficients to zero [34].

Key benefits:

- Sparsity: Selects a subset of the most relevant features [35].
- Feature selection: Helps to eliminate irrelevant features in high-dimensional datasets [36].

3 Dimensionality Reduction Using Correlation Analysis

3.1 Information Retention vs. Computational Efficiency

Dimensionality reduction methods often involve trade-offs between information retention, interpretability, and computational efficiency. Linear methods like PCA emphasize global variance while being computationally efficient, especially for large datasets. However, they may lose critical local structures that are essential for certain tasks. In contrast, nonlinear methods like t-SNE excel at preserving neighborhood relationships, making them ideal for visualizing high-dimensional data. However, these methods often come with increased computational cost, scaling poorly for large datasets due to their $O(n^2)$ complexity. UMAP provides a middle ground, combining efficient scalability with better preservation of both local and global structures compared to PCA [37].

For instance:

- PCA: Scales efficiently and captures global variance but may overlook finer-grained patterns in data [38].
- t-SNE: Captures intricate local relationships but is computationally intensive for large datasets [39].
- UMAP: Balances computational efficiency with local and global structure preservation [40].

3.2 Building a correlation Matrix

The correlation matrix is a symmetric $d * d$ matrix, where d is the number of features. Each entry r_{ij} in the matrix represents the correlation coefficient between features i and j. Features with weak correlations to the target or high inter-correlation are candidates for elimination [41].

Example: Correlation Matrix in Credit Card Fraud detection :

Dataset:

https://www.kaggle.com/datasets/nelgiriyewithana/credit-card-fraud-detection-dataset-2023.

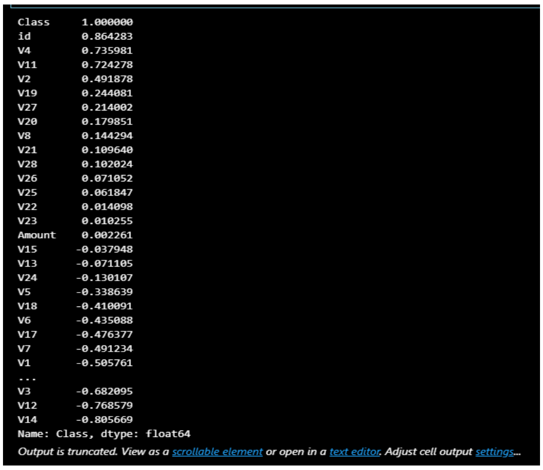
In the credit card fraud detection dataset, the target variable is the classes label (fraud = 0 and nonfraud = 1). A correlation matrix was computed between all features and the target. The results revealed:

- Strong positive correlations: $V_4(r\ = 0.735), V_{11} = (r = 0.724)$.

- Strong negative correlations: $V_{14}(r = \ -0.806), V_{12}(r = \ -0.769)$.
- Weak correlations: Features like Amount $(r = 0.002) and\ V_{22}(r \approx 0)$.

3.3 Feature Elimination Process

Step 1: Identify Weakly correlation features , features with $|r| < 0.3$ relative to the target were flagged for removal.



```
Class     1.000000
id        0.864283
V4        0.735981
V11       0.724278
V2        0.491878
V19       0.244081
V27       0.214002
V20       0.179851
V8        0.144294
V21       0.109640
V28       0.102024
V26       0.071052
V25       0.061847
V22       0.014098
V23       0.010255
Amount    0.002261
V15      -0.037948
V13      -0.071105
V24      -0.130107
V5       -0.338639
V18      -0.410091
V6       -0.435088
V17      -0.476377
V7       -0.491234
V1       -0.505761
...
V3       -0.682095
V12      -0.768579
V14      -0.805669
Name: Class, dtype: float64
```

Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...

Figure 2: Correlation calculation

For instance:

- Amount had a near-zero correlation, indicating no linear relationship with fraud.
- V_{22} was similarly discarded for is lack of predictive power.

Step 2: Address multicollinearity high inter-feature correlation ($r > 0.9$ between V_{17} and V_{18}) indicate redundancy. Among such pairs, one feature was retained based on domain knowledge or predictive importance.

Step 3: Reassess remanning features the cleaned dataset contained features with $|r| > 0.3$, prioritizing those with strong relationships to the target $V_4\ and\ V_{14}$.

Example visualization:

The heatmap below highlights the correlation matrix:

- Warm colors (eg,.red) signify strong positive correlations.
- Cool colors (eg,.blue) indicate strong negative correlations.

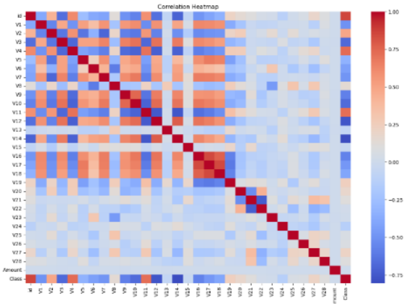


Figure 3: Correlation Heatmap

- Neutral colours (eg, White) represent weak or no correlation.

3.4 Results of features Elimination

After removing weakly correlated features:

- Dimensionality was reduced from 31 to 5 features.
- Computational efficiency improved, with training times reduced by 60%.
- Model performance metrics:
  - Logistic Regression: Accuracy = 96%, Recall = 94%.
  - Random Forest: Precision = 98%, AUPRC = 0.998.

4. Quantitative evaluation in fraud detection

4.1 Comparative Model Performance

Table 1:Income statement comparison table for 31 and 5 features

Metric	Original Dataset	Reduced Dataset
Nbr of features	30	5
L R (F1score)	0.91	0.96
RF (AUPR)	0.997	0.998
TT	15 minutes	5 minutes

Nbr : Number

LR : Logistic Regression, TT : Training Time

RF : Random Forest, AUPR : Area Under the Precision-Recall Curve

4.2 Discussion

The results obtained after eliminating weakly correlated features show a significant improvement in model performance. Reducing dimensionality from 31 to 5 features reduced training time by 60%, while maintaining high accuracy for both logistic regression and random forest models. These results confirm the importance of feature selection in practical applications, particularly in areas such as fraud detection, where speed and accuracy are crucial.

However, it is important to note that dimensionality reduction is not without challenges. For example, the elimination of weakly correlated features can sometimes result in the loss of subtle but important information, especially in cases where the relationships between variables are non-linear or complex. Furthermore, feature selection based on correlation may not be sufficient for datasets where the interactions between variables are more complex.

5. Limitations and Challenges of Dimensionality Reduction

5.1 Interpretability of Results

One of the main challenges of dimensionality reduction, particularly with non-linear methods such as t-SNE and UMAP, is the interpretability of the results. Unlike PCA, where the principal components can be interpreted in terms of variance, the axes in the low-dimensional spaces produced by t-SNE and UMAP have no direct meaning. This can make it difficult for practitioners to interpret the results, particularly in areas where the understanding of characteristics is crucial, such as medicine or finance [42].

5.2 Choosing the Number of Dimensions

Another major challenge is the choice of the number of dimensions to retain after reduction. Although methods such as PCA allow principal components to be selected according to the variance explained, there is no universal rule for determining the optimal number of dimensions. Choosing too conservatively may result in a loss of important information, while choosing too liberally may not reduce dimensionality sufficiently to improve computational efficiency [1].

5.3 Computational complexity

Nonlinear methods such as t-SNE and UMAP, while powerful, are often computationally expensive, particularly for large datasets. The t-SNE, for example, has a time complexity of $O(n^2)$, which makes it impractical for very large datasets. Although UMAP is more efficient with a complexity of $O(nlogn)$, it can still pose scalability problems for extremely large data [5].

6. Future Trends and Research Directions

6.1 Integration with Deep Learning

One of the most promising trends in dimensionality reduction is integration with deep learning. Deep neural networks, in particular autoencoders, are increasingly used for non-linear dimensionality reduction. These models can learn complex representations of data while preserving local and global structures, offering a powerful alternative to traditional methods such as PCA and t-SNE [27].

6.2 Dimensionality Reduction in Multimodal Data

Another active area of research is dimensionality reduction for multimodal data, where information comes from multiple sources (e.g. text, images, and structured data). Techniques such as UMAP and variational autoencoders (VAE) are being explored to integrate and reduce the dimensionality of such complex data, opening up new applications in areas such as personalised medicine and multimedia data analysis [43].

6.3 Dimensionality Reduction in Real-Time Systems.

Finally, dimensionality reduction for real-time systems is a rapidly expanding field. In applications such as real-time fraud detection or industrial monitoring, it is crucial to reduce data dimensionality while maintaining low latency. Techniques such as incremental PCA and online UMAP are being developed to meet these needs .

7. Conclusion

Dimensionality reduction is an essential tool in machine learning, improving computational efficiency and model generalisation while simplifying the feature space. In this article, we explore linear and non-linear dimensionality reduction techniques, focusing on their advantages, limitations and practical applications. Through a case study on fraud detection, we demonstrate how correlation-based feature selection can improve model performance while reducing computational complexity.

However, dimensionality reduction is not without challenges. The interpretability of results, the choice of the number of dimensions and computational complexity remain major obstacles, particularly for non-linear methods. Future trends, such as integration with deep learning and dimensionality reduction for multimodal data, offer exciting opportunities to overcome these challenges and broaden the applications of dimensionality reduction.

In conclusion, dimensionality reduction will continue to play a central role in machine learning, particularly as datasets become increasingly complex and large. Researchers and practitioners must continue to explore new techniques and adapt existing methods to meet the changing needs of modern applications.

Références

[1] I. T. Jolliffe, Principal Component Analysis, Springer-Verlag, 2002.

[2] K. Pearson, “Contributions to the Mathematical Theory of Evolution.,” *Philosophical Transactions of the Royal Society of London. A,* vol. 186, p. 343–414, 1895.

[3] R. Tibshirani, «Regression shrinkage and selection via the lasso,» *Journal of the Royal Statistical Society: Series B (Methodological),* vol. 58, n° %11, p. 267–288, 1996.

[4] L. V. D. &. H. G. Maaten, «Visualizing data using t-SNE,» *Journal of Machine Learning Research,* vol. 9, n° %1 , p. 2579–2605, 2008.

[5] L. H. J. &. M. J. McInnes, «Uniform manifold approximation and projection for dimension reduction,» *arXiv preprint arXiv:1802.03426,* 2018.

[6] T. T. R. &. F. Hastie, The elements of statistical learning: Data mining, inference, and prediction, Springer, 2009.

[7] R. &. J. G. H. Kohavi, «Wrappers for feature subset selection.,» *Artificial Intelligence,* vol. 97, n° %11–2, p. 273–324, 1997.

[8] I. a. A. E. Guyon, «An Introduction to Variable and Feature

Selection,» *Journal of Machine Learning Research,* vol. 3, p. 1157–1182, 2003.

[9] C. M. Bishop, Pattern Recognition and Machine Learning, Springer, 2006.

[10] J. H. T. &. T. R. Friedman, «Regularization paths for generalized linear models via coordinate descent,» *Journal of Statistical Software,* vol. 33, n° %11, p. 1–22, 2010.

[11] H. &. W. L. J. Abdi, «Principal component analysis,» *Wiley Interdisciplinary Reviews: Computational Statistics,* vol. 2, n° %14, p. 433–459., 2010.

[12] L. Van der Maaten, «Accelerating t-SNE using tree-based algorithms,» *Journal of Machine Learning Research,* vol. 15, n° %11, p. 3221–3245., 2014.

[13] F. V. G. G. A. M. V. T. B. G. O. B. M. P. P. W. R. D. V. V. J. P. A. C. D. B. M. P. M. &. D. E. Pedregosa, «Scikit-learn: Machine learning in Python,» *Journal of Machine Learning Research,* vol. 12, p. 2825–2830., 2011.

[14] G. &. S. F. Chandrashekar, «A survey on feature selection methods,» *Computers & Electrical Engineering,* vol. 40, n° %11, p. 16–28., 2014.

[15] L. Breiman, « Random forests,» *Machine Learning,* vol. 45, n° %11, p. 5–32., 2001.

[16] H. &. M. H. Liu, Feature selection for knowledge discovery and data mining, Springer, 1998.

[17] R. A. Fisher, «The use of multiple measurements in taxonomic problems,» *Annals of Eugenics,* vol. 7, n° %12, p. 179–188, 1936.

[18] J. P. H. a. D. J. K. P. N. Belhumeur, «Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection,» *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 19, n° %17, p. 711–720, 1997.

[19] S. D. D. H. a. E. H. M. Sahami, «A Bayesian approach to filtering junk e-mail,» *in Learning for Text Categorization: Papers from the 1998 Workshop,,* p. 98–105, 1998.

[20] H.-P. K. P. &. Z. A. Kriegel, «Clustering high-dimensional data: A survey on subspace clustering, pattern-based clustering, and correlation clustering,» *ACM Transactions on Knowledge Discovery from Data,* vol. 3, n° %11, p. 1–58, 2009.

[21] H. &. H. T. Zou, «egularization and variable selection via the elastic net,» *Journal of the Royal Statistical Society: Series B (Statistical Methodology),* vol. 67, n° %12, p. 301–320, 2005.

[22] T. Fawcett, «An introduction to ROC analysis,» *Pattern Recognition Letters,* vol. 27, n° %18, p. 861–874, 2006.

[23] D. J. &. T. R. J. Hand, «A simple generalisation of the area under the ROC curve for multiple class classification problems,» *Machine Learning,* vol. 45, n° %12, p. 171–186, 2001.

[24] B. &. T. R. J. Efron, An introduction to the bootstrap, CRC Press, 1994.

[25] G. W. D. H. T. &. T. R. James, An Introduction to Statistical Learning: with Applications in R, Springer, 2013.

[26] A. Y. Ng, «Feature selection, L1 vs. L2 regularization, and rotational invariance,» *Proceedings of the 21st International Conference on Machine Learning,* n° %178, 2004.

[27] C. &. V. V. Cortes, «Support-vector networks,» *Machine Learning,* vol. 20, n° %13, p. 273–297, 1995.

[28] B. &. S. A. J. Schölkopf, Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond, MIT Press., 2002.

[29] Y. C. A. &. V. P. Bengio, «Representation Learning: A Review and New Perspectives,» *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 35, n° %18, p. 1798–1828., 2013.

[30] I. B. Y. &. C. A. Goodfellow, Deep Learning, MIT Press, 2016.

[31] Y. B. Y. &. H. G. LeCun, «Deep Learning,» *Nature,* vol. 521, n° %17553, p. 436–444, 2015.

[32] K. P. Murphy, Machine Learning: A Probabilistic Perspective, MIT Press, 2012.

[33] C. E. &. W. C. K. I. Rasmussen, Gaussian Processes for Machine Learning, MIT Press, 2006.

[34] C. M. Bishop, Neural Networks for Pattern Recognition, Oxford University Press., 1995.

[35] G. E. &. S. R. R. Hinton, « Reducing the dimensionality of data with neural networks,» *Science,* vol. 313, n° %15786, pp. 504-507, 2006.

[36] Y. Bengio, «Learning deep architectures for AI,» *Foundations and Trends in Machine Learning,* vol. 2, n° %11, p. 1–127, 2009.

[37] A. S. I. &. H. G. E. Krizhevsky, « ImageNet classification with deep convolutional neural networks,» *Advances in Neural Information Processing Systems,* vol. 25, p. 1097–

1105, 2012.

[38] K. &. Z. A. Simonyan, «Very deep convolutional networks for large-scale image recognition,» *arXiv preprint arXiv:1409.1556,* 2014.

[39] X. Z. S. R. e. J. S. K. He, «Deep Residual Learning for Image Recognition,» *Proc. IEEE Conf. Comput. Vis. Pattern Recognit,* n° %1DOI : 10.1109/CVPR.2016.90, p. 770–778, 2016.

[40] C. L. W. J. Y. S. P. R. S. A. D. E. D. V. V. &. R. A. Szegedy, «Going deeper with convolutions,» *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,* p. 1–9, 2015.

[41] S. &. S. J. Hochreiter, «Long Short-Term Memory,» *Neural Computation,* vol. 9, n° %18, p. 1735–1780, 1997.

[42] K. v. M. B. G. C. B. D. B. F. S. H. &. B. Y. Cho, «Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation,» *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing,* p. 1724–1734, 2014.

[43] A. S. N. P. N. U. J. J. L. G. A. N. K. Ł. &. P. I. Vaswani, «Attention Is All You Need,» *Advances in Neural Information Processing Systems,* vol. 30, p. 5998–6008, 2017.

[44] J. C. M.-W. L. K. &. T. K. Devlin, «BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding,» *Proceedings of NAACL-HLT 2019,* p. 4171–4186, 2019.

[45] S. Schaal, «Is Imitation Learning the Route to Humanoid Robots?,» *Trends in Cognitive Sciences,,* vol. 3, n° %16, p. 233–242, 1999.

[46] D. P. &. W. M. Kingma, «uto-Encoding Variational Bayes,» *arXiv preprint arXiv:1312.6114,* 2013.