Journal of Information Systems Engineering and Management
2025, 10(48s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

An Adaptive Approach for Training Deep Belief Networks

N. Raji[1*], Dr. S. Manohar[2]
[1*]*Research scholar, Department of Computer Science and Engineering, SRM Institute of Science and Technology, Vadapalani Campus, Chennai, Tamil Nadu, India. Email: rn5525@srmist.edu.in*
[2]*Associate Professor, Department of Computer Science and Engineering, SRM Institute of Science and Technology, Ramapuram Campus, Chennai, Tamil Nadu, India. Email: manohars1@srmist.edu.in*
*(*Corresponding author)*

ARTICLE INFO

Received: 30 Dec 2024

Revised: 15 Feb 2025

Accepted: 25 Feb 2025

ABSTRACT

Deep Belief Networks (DBNs) are a stack of networks, each having picked up unique characteristics and attributes from the original data. DBNs can handle supervised and unsupervised tasks thanks to their intricate layer-wise neural architecture. This article presents an adaptive approach for training DBN and also analyzes the various training algorithms used in the process of training DBNs. This paper begins by delving into the pre-training phase, where Restricted Boltzmann Machines (RBMs) play a central role. We review the Contrastive Divergence (CD) and Persistent Contrastive Divergence (PCD) algorithms, highlighting their advantages and disadvantages in initializing deep belief nets. These networks have a variety of applications. Importance is placed on their pertinence to different data types and scales. Moving to the fine-tuning stage, the paper explores the use of backpropagation with gradient descent. Furthermore, the architectural variants of DBNs like CDBNs and RDBNs with their respective areas of application are discussed. CDBNs have an accuracy of over 95% when operated on standard image classification benchmarks like MNIST and ImageNet whereas RDBNs achieve an accuracy of over 90% on sentiment analysis and 85% for speech recognition on longer audio sequences. We highlight the adaptation of DBNs for specific tasks, including classification, regression, clustering, and generative modeling. We also compare the training complexity of the proposed algorithm with the existing algorithm. We found that the proposed algorithm outperformed other algorithms by training the model in 500 seconds.

Keywords: Contrastive Divergence, Deep Belief Networks, Fine-tuning, Pre-training.

INTRODUCTION

Deep belief networks [1, 2] are composed of multiple layers of the neurons, each of which is coupled to the neuron in the layer below it. The general architecture is akin to that of an MLP; however, the only connections are between the layers; there are no connections within layers. Since each layer learns on the results of the one before it separately, each layer may be thought of as a distinct model. In this sense, a DBN is a stack of networks, each having picked up unique characteristics and attributes from the original data.

An important component of the deep learning framework is the undirected graphical model known as the Restricted Boltzmann Machine [3].By building limited Boltzmann machines and using the subsequent activations from one to be the training data for the subsequent, several hidden layers can be learned efficiently. In essence, these represent the neural network that is a part of the so-called energy-based models. This algorithm can be applied to feature learning, topic modelling, regression collaborative filtering, dimensionality reduction, and classification.

Deep belief networks are considered to be the best option eliminating the need of convolutional neural network due to its advantages. Among these are: the capacity to work with enormous amounts of data and extract underlying correlations utilising hidden units, quicker training and improved outcomes global minima are reached as a result of improved weight initialization. Deep Belief Networks can handle big datasets and many kinds of input, and they

Journal of Information Systems Engineering and Management
2025, 10(48s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

function similar to deep neural networks. As a result, they excel at tasks like speech recognition, image generation, text classification, and image classification.

RELATED WORKS

The fundamental component of deep belief nets and other deep learning techniques is the restricted Boltzmann machine (RBM). Rapid learning and prediction are necessary for RBM-based machine learning approaches to be used in real-world scenarios. To expedite RBM learning and prediction without affecting the outcomes, Lean Contrastive Divergence (LCD) [8-10], a modified version of the Contrastive Divergence (CD) algorithm, is employed.

LCD uses two optimisation approaches to circumvent the majority of the necessary computations. The first is known as limits-based filtering, which uses triangle inequality to quickly compute bounds instead of costly calculations of several vector dot products. The second technique is delta product, that efficiently identifies and steers clear of several repeated computations in Gibbs Sampling, the fundamental process of RBM [6-7]. Both the original contrastive divergence learning algorithm and its modifications can benefit from the optimisations. It operates by approximating the gradient of the data's log-likelihood. CD iteratively updates its model's parameters to reduce the discrepancy between the distribution of the data and the model.

PCD is used to address the slow convergence of CD. It preserves an enduring chain of samples from the model's distribution, which speeds up training and improves stability.

After pre-training the individual layers with RBMs, fine-tuning the entire DBN involves using SGD with backpropagation. SGD adjusts the weights of the network in a direction that minimizes a specified loss function. Modern variants such as mini-batch SGD help improve convergence and efficiency. Techniques like Adagrad, Adadelta, and Adam change the rate of learning during training depending on the past inclines, helping to accelerate convergence and improve training stability.

When L1 regularisation [11] is paired with normalisation, there is no regularisation effect; hence, L1 regularisation will not work in the DNN method. Additionally, we can pick high-dimensional features by applying L2 constraints to a single L1 regularisation. This enables the DNN method to incorporate L2 norm to avoid overfitting in addition to performing selection of features through L1 regularisation.

Batch normalisation [12-15] in a neural network is accomplished by way of a normalisation step that corrects the variances and means of the inputs to each layer. Although it would be ideal to perform the normalisation over the whole training set, using the global information makes it impractical to apply this step in conjunction with stochastic optimisation techniques. Therefore, during the training phase, normalisation is limited to each mini-batch.

An observation in latent space can be probabilistically described using a variational autoencoder (VAE) [16,17]. We will thus structure our encoder to present a probability distribution for every latent attribute instead of creating an encoder that produces a single value to represent each latent state characteristic.

It can be used for a variety of tasks, including creating synthetic data and compressing data. In contrast to an autoencoder, a variational autoencoder offers a statistical means of characterising the dataset's samples in latent space. Therefore, rather than producing a single output value at the bottleneck layer, the encoder in the variational autoencoder outputs a probability distribution.

Table 1 below highlights the methods used in various articles from which insights were taken to proceed with the current research.

Table 1. Summary of Methods Used and Their Advantages.

Ref. No.	Methods Used	Advantages	Drawbacks
[2]	k-means clustering, LSTM	High recognition accuracy, reduced computational complexity	Requires complex architecture to train large datasets

Journal of Information Systems Engineering and Management
2025, 10(48s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

[3]	Phased Gibbs Sampling	Reduced reconstruction error and training time	Performance is sensitive to learning rate and batch size
[6]	Fuzzy RBM, SVM	Speedy feature extraction	The process of fuzzification is complex
[13]	SVM, Logistic Regression	High Recognition Accuracy	Limited generalization
[18]	Euclidean Stability	High learning rate	Stability is affected by regularization techniques
[19]	SVM, Random Forest	Effective pattern classification	The availability of high-quality, diverse datasets is limited.
[20]	Variational Mode Decomposition, LSTM, GRU	Reduced RMSE and relative errors	LSTM/GRU models are computationally intensive
[21]	Random Forest, Gated Recurrent Unit	High Prediction Accuracy	Increased training time and complex computational resources
[25]	Auto-encoders, vanishing gradient	Eliminates the vanishing gradient problem, even distribution of rating data	Needs complex computational resources
[27]	Transfer Learning, QCNN	High Qualification Accuracy	Quantum systems are subject to noise and errors
[29]	Generative models and LLMs	Provides fine-grained structure information	Controlling design aspects are challenging

METHODOLOGY

Pretraining Phase

In order for the learned weights to accurately represent the input data, the pre-training stage attempts to initialise the DBN. Unsupervised learning is used in this pre-training phase, and each RBM module receives independent feature detector training. The initial layer of the ensemble, sometimes referred to as the input layer or bottom layer, works directly with the raw data to extract its features and build a latent representation in the process. Then, the next layer is trained using the output of the previous layer. Efficient feature learning is made possible by this greedy layer-wise learning.

Steps in Pretraining

The Contrastive Divergence algorithm is used to train RBMs. CD is an iterative algorithm that approximates the slope of the log-likelihood of the data. There are two phases to the training: the good and the negative. The likelihood for the activation of the hidden units are computed in the positive phase by sampling the activation of the visible layer, and in the negative phase the opposite is true. Iterating this process several times covers different data samples, with weights updated following each iteration. The output layer, the final layer, is where the network's prediction is finally output. It involves the following steps:

- Compute the expected values of the concealed units based on the input data (positive phase).
- Sample from the hidden units to reconstruct the input (negative phase).

Journal of Information Systems Engineering and Management
2025, 10(48s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

- To reduce the disparity between the beneficial and detrimental phases, modify the RBM's weights and biases.
- If you have a deep architecture with multiple hidden layers, you repeat the RBM training process step by step. The result of one RBM becomes the information for the following one. This is referred to as layer-wise pre-training.

After training all the RBMs, you stack them together to create the deep architecture of the DBN. The connections between layers are set based on the trained RBM parameters.

Fine-Tuning

Once the DBN is constructed with initialized weights, it undergoes a fine-tuning phase. After the model weights are initialised and pre-training is finished, it is further adjusted for activities that come after. Labelled information and a supervised approach to learning, such as backpropagation, are used for fine-tuning. This allows the model to be trained for a variety of tasks, such as regression or classification, with faster training times and improved performance due to the initialised weights. Table 2 gives a comparative study of Contrastive Divergence (CD) and Persistent Contrastive Divergence (PCD).

Table 2. Comparison of CD and PCD.

Strength/Weakness	CD	PCD
Simplicity	Simple and easy to understand	Relatively simple to implement
Efficiency	Converges quickly in early stages	Faster convergence than CD
Widely Used	Commonly applied in practice	Widely used in training deep networks
Sampling Noise	Sensitive to sampling noise	Mitigates sampling noise through persistence
Slow Convergence	Slower convergence, especially in deep networks	Faster convergence, particularly in deep networks
Initialization Dependency	Sensitive to initialization	Improved weight initialization compared to CD
ComputationCost	Lower computational cost	Higher memory requirements due to a persistent chain

After comparing the existing training algorithms, we propose a training algorithm that aims at reducing the time complexity of the training algorithm while concentrating on achieving high accuracy. The process followed by the training algorithm is as follows:

- Initialize Parameters
- Load Training Data
- Define Adaptive Learning Algorithm
- Define Training Step
- Training the Network
- Evaluate Convergence
- Post-Training Analysis

Journal of Information Systems Engineering and Management
2025, 10(48s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Algorithm 1. Proposed Training Algorithm for DBN

1. Set the units that are visible to a training vector initially.
2. Using the sigmoid function and the hidden layer's bias, update the hidden units concurrently concerning the visible units.
3. Update the hidden units simultaneously with the visible units. We refer to this as the "reconstruction" process.
4. Update the hidden units based on the rebuilt visible units.
5. Execute the weight modification.

Architectural Variants

The basic structure of DBN can be extended to address the requirements of specific types of data to be operated on and types of tasks to be performed. One kind of deep belief network made up of several layers of convolutional RBMs stacked together is called a convolutional deep belief network (or CDBN). Although it has been used to other fields as well, this hierarchical generative model excels at object detection and image processing. The model's translation invariance and good scaling to high-dimensional images are among its key characteristics.

Probabilistic max-pooling is a technique used by CDBNs to minimise the dimensions of the network's upper layers. CDBNs are employed in processing images. The other variant called Recurrent Deep Belief Networks are employed in processing the time series data and forecasting applications.

It uses LSTMs and GRUs in order to manage long term dependencies. An input gate, an output gate, a forget gate, and a cell make up a typical LSTM unit. The three gates control the information flow into and out of the cell, and the cell retains values for arbitrarily long periods of time. Forget gates use a value between 0 and 1 to indicate which information from a prior state should be discarded in relation to the present input.

With a mechanism that allows users to provide or forget specific features, the GRU functions similarly to an LSTM [2], but it has fewer parameters because it does not have an output gate or context vector.[3] It was discovered that GRU performed comparably to LSTM on a few tasks related to polyphonic music modelling, speech signal modelling, and natural language processing.

Challenges

- **Vanishing [23-25] and Exploding Gradients:** The obstacle known as the "vanishing gradient problem" arises during backpropagation when the activation functions' derivatives, or slopes, get lower as we proceed backward through a neural network's layers. This issue is especially noticeable in deep networks with plenty of layers, which makes it more difficult for the model to be trained effectively. It can greatly extend the training duration, cause the weight updates to become minuscule or exponentially small, or, in the worst case, completely stop the training process.
- **Data Availability:** For efficient training, deep networks—including DBNs—need a lot of labeled data. It can be costly and time-consuming to obtain labeled data, particularly for specialized or specialist topics. But it is suggested to have a lot of labelled data for the training to be effective.
- **Choice of Hyperparameters:** Selecting appropriate hyperparameters, such as learning rates, batch sizes, and regularization strengths, can be challenging. Grid search, random search, and automated hyperparameter tuning methods help in this regard.

Trends

- **Transfer Learning [26, 27]:** Pre-training DBNs on large datasets and fine-tuning them for specific tasks has been a successful trend. Models like convolutional neural networks (CNNs) and transformer-based architectures are often pre-trained on vast corpora and fine-tuned for various tasks.
- **AutoML and Neural Architecture Search:** Automated Machine Learning (AutoML) and Neural Architecture Search (NAS) aim to automate the process of finding optimal neural network architectures and hyperparameters, reducing the burden on practitioners.

Journal of Information Systems Engineering and Management
2025, 10(48s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

- **Sparse and Efficient Architectures:** Research into making deep networks more computationally efficient and compact continues. Sparse activation techniques, quantization, and model compression are explored to deploy deep models on resource-constrained devices.
- **Self-Supervised Learning:** Self-supervised learning techniques, where networks are trained to predict parts of their input data, have gained attention as a way to leverage unlabeled data for pre-training.

ADAPTATION OF DBNS

Deep Belief Networks (DBNs) can be applied to a variety of machine learning tasks, such as classification, regression, clustering, and generative modeling. These networks can be modified suitably for each of the following tasks:

Classification

- **Adaptation:** For classification tasks, we modify the architecture of DBN by using a softmax layer or a sigmoid layer (for binary classification). This modified top layer allows the network to output class probabilities.
- **Process:** After pre-training, we fine-tune the DBN using labeled data. In this case, we would be able to minimize the classification loss during training.
- **Applications:** DBNs are used in applications where image processing is a major task. Frequent use cases include image classification, text categorization, and video processing for motion recognition.

Regression

- **Adaptation:** For regression tasks, you can modify the top layer of the DBN to have a single neuron with a linear activation function. This setup allows the network to predict continuous values.
- **Process:** Similar to classification, after pre-training, you fine-tune the DBN using labeled data. In this case, you minimize a regression loss during training.
- **Applications:** DBNs adapted for regression are employed in tasks like predicting housing prices, stock market analysis, and any problem involving the prediction of continuous numerical values.

Clustering

- **Adaptation:** To use DBNs for clustering tasks, you can leverage the features learned by the DBN's hidden layers to represent data points. Then, you apply clustering algorithms, such as k-means or Gaussian mixture models, to these feature representations.
- **Process:** After pre-training the DBN, you extract the activations of the hidden layers for each data point. We use these feature representations as input to a clustering algorithm to group similar data points into clusters.
- **Applications:** DBNs adapted for clustering are useful in unsupervised learning scenarios, including customer segmentation, anomaly detection, and document clustering.

Generative Modelling

- **Adaptation:** DBNs can be used for generative modeling tasks by fine-tuning them as generative models. This typically involves training the network to create fresh data samples that bear a resemblance to the training set.
- **Process:** After pre-training, you fine-tune the DBN using a generative modeling [28, 29] approach. One popular approach is to use the contrastive divergence algorithm for fine-tuning RBMs in the DBN. Once fine-tuned, the DBN can create fresh samples by taking a sample from the probability distributions it has learned.
- **Applications:** DBNs adapted for generative modeling are used in image generation, text generation, recommendation systems, and other tasks where generating new data samples is valuable. Here we delve into some of the prominent applications of DBNs and their impact on different fields.

Image Recognition and Computer Vision

Journal of Information Systems Engineering and Management
2025, 10(48s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

DBNs have revolutionized image recognition tasks by automatically learning feature hierarchies from raw pixel data. They have played a key role in enabling cutting-edge results in tasks including segmentation, classification, and object detection.

Applications include medical image analysis for disease diagnosis, autonomous vehicles for scene understanding, and satellite image analysis for environmental monitoring.

Natural Language Processing (NLP)

Text categorization and sentiment analysis are two examples of natural language processing jobs that have been handled by DBNs.Sentiment analysis recognises and extracts "feeling" information from text by utilising machine learning and natural language processing (NLP) techniques. Sentiment information is frequently used to determine the overall emotion of a text and might be favourable, adverse, or neutral. The technique of giving unstructured text data predetermined categories is known as text classification. This is a typical NLP problem that has several uses, including spam detection, subject tagging, and sentiment analysis.

Recommendation Systems

Content-based and collaborative filtering, and even other systems like knowledge-based systems, are frequently used by recommender systems. Collaborative filtering techniques create a model based on historical user behaviour (products previously selected or bought, along with any numerical ratings) and comparable decisions taken by other users. Next, the user's potential interest in certain items (or rankings for objects) is predicted using this model. Content-based filtering techniques make use of an item's discrete, pre-tagged attributes to suggest other items that have those attributes.

Drug Discovery and Bioinformatics

Applying DBNs to bioinformatic analysis helps speed up the identification of therapeutic targets, as well as the screening and improvement of drug candidates. It can also make it easier to characterise adverse effects and anticipate drug resistance. They are also used in the course of developing a novel medication that targets a certain ailment.

Financial Forecasting and Time Series Analysis

In order to forecast future trends, behaviours, and responses based on historical data, time series analysis and forecasting are essential. Predicting market demand, sales changes, stock prices, and other factors helps organisations make well-informed decisions, optimise resources, and minimise risks. DBN promotes efficiency and competitiveness by supporting planning, budgeting, and strategy in a variety of fields, including banking, economics, medical care, environmental science, and resource management.

In increasingly complex configurations, we use deep belief networks instead of deep feedforward chains as well as convolutional neural networks. They have the advantage of requiring less computing power. Unlike feedforward neural networks, where computational complexity increases exponentially with an increasing number of layers, it grows exponentially with the size of layers and is less vulnerable to the diminishing gradient issue. Table 3 summarizes some common applications of Deep Belief Networks.

Table 3. Applications of DBN.

Application	Description
Image Recognition	Using their capacity to learn hierarchical features, DBNs are applied to tasks including object identification, image classification, and facial recognition.
Natural Language Processing	In NLP, DBNs have been used for tasks like sentiment analysis, document classification, machine translation, and named entity recognition.

Journal of Information Systems Engineering and Management
2025, 10(48s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Speech Recognition	DBNs are suitable for tasks like speech-to-text conversion, speaker identification, and emotion detection from speech.
Anomaly Detection	DBNs are effective for anomaly detection in various domains, such as cybersecurity (detecting network intrusions), healthcare (identifying abnormal medical data), and fraud detection.
Recommender Systems	Personalized recommender systems can be developed using DBNs to enhance user experience and engagement on e-commerce, streaming, and content platforms.
Financial Forecasting	DBNs have been applied to financial time series data for tasks like stock price prediction, risk assessment, credit scoring, and algorithmic trading.
Drug Discovery	In pharmaceutical research, DBNs are used for drug discovery tasks like molecular structure analysis, compound activity prediction, and virtual screening of potential drug candidates.
Robotics	DBNs play a role in robotics for tasks like object recognition, path planning, manipulation, and learning complex control policies for robotic agents.

RESULTS

In this section, we present the results by comparing the proposed training algorithm with the existing algorithms.

Experimental Setup

An Infinix Hot3G smartphone was utilised to take the photos for this study. In this technique, 1000 distinct photographs of both healthy and unhealthy rice plant leaves are taken. Different methods are compared with the suggested training algorithm for the performance analysis. Figure 1 - 5 compares the results of using several models and algorithms with the processed dataset. Implementation of the training algorithm is done in Python.

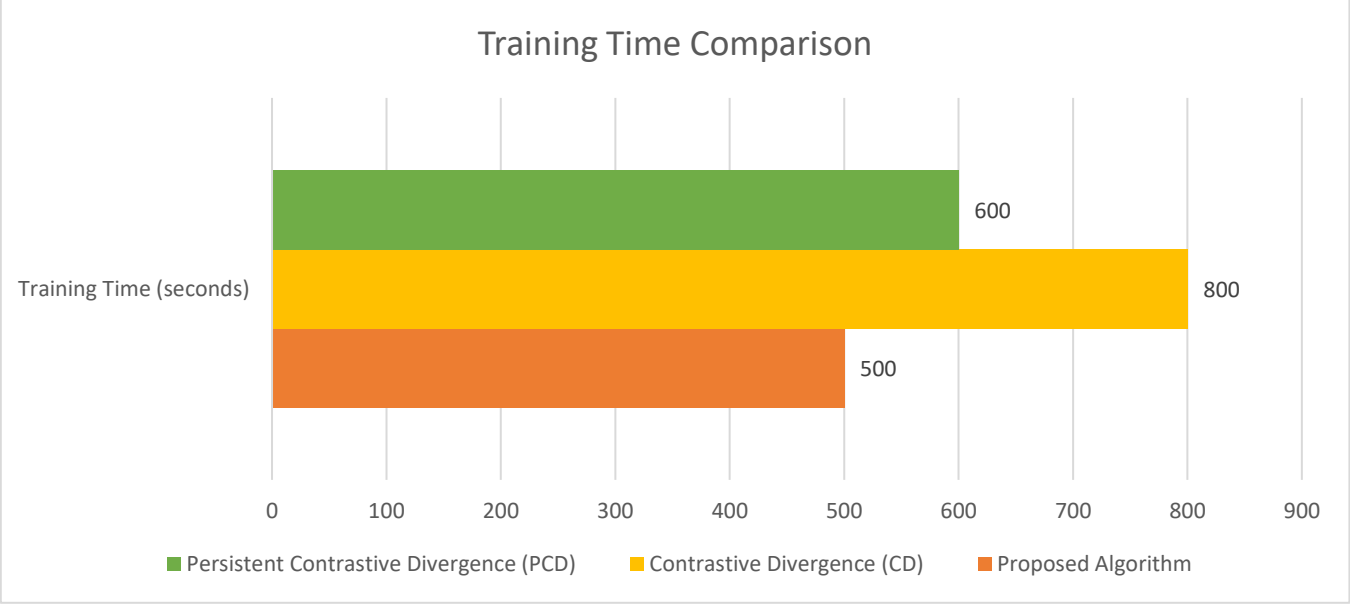

Figure 1. Training Time Comparison.

Journal of Information Systems Engineering and Management
2025, 10(48s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

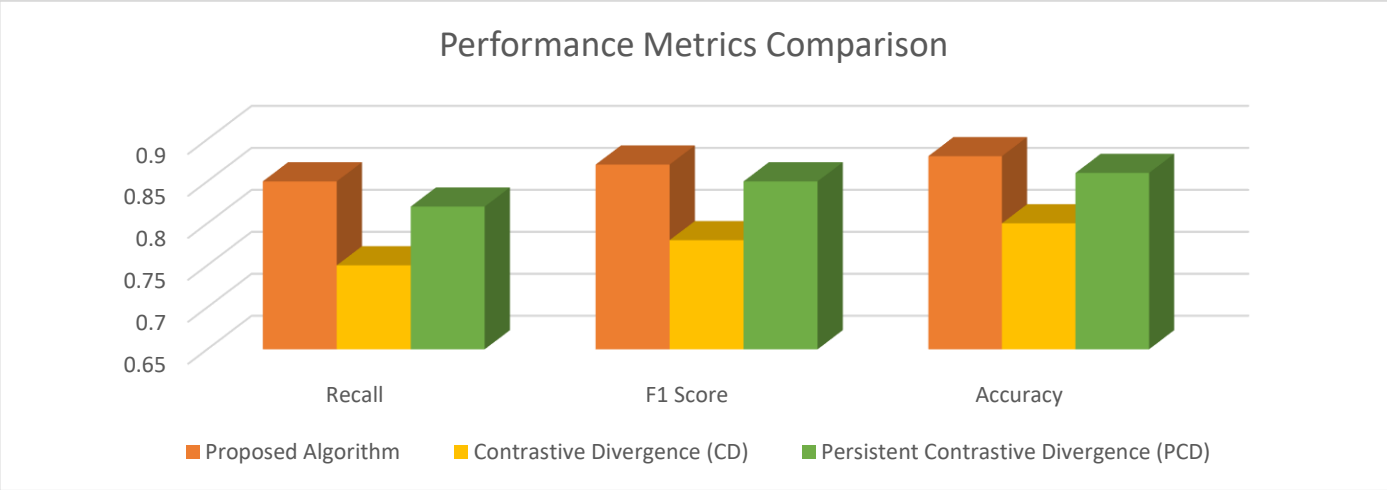

Figure 2. Performance metrics comparison.

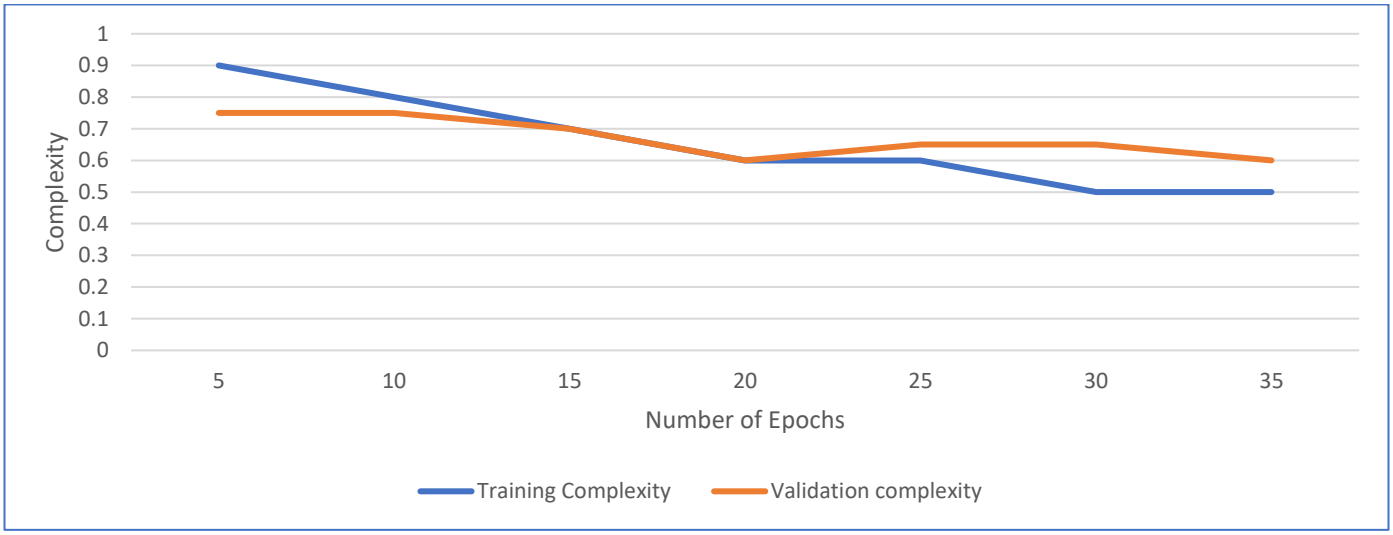

Figure 3. Training Complexity of Proposed Algorithm.

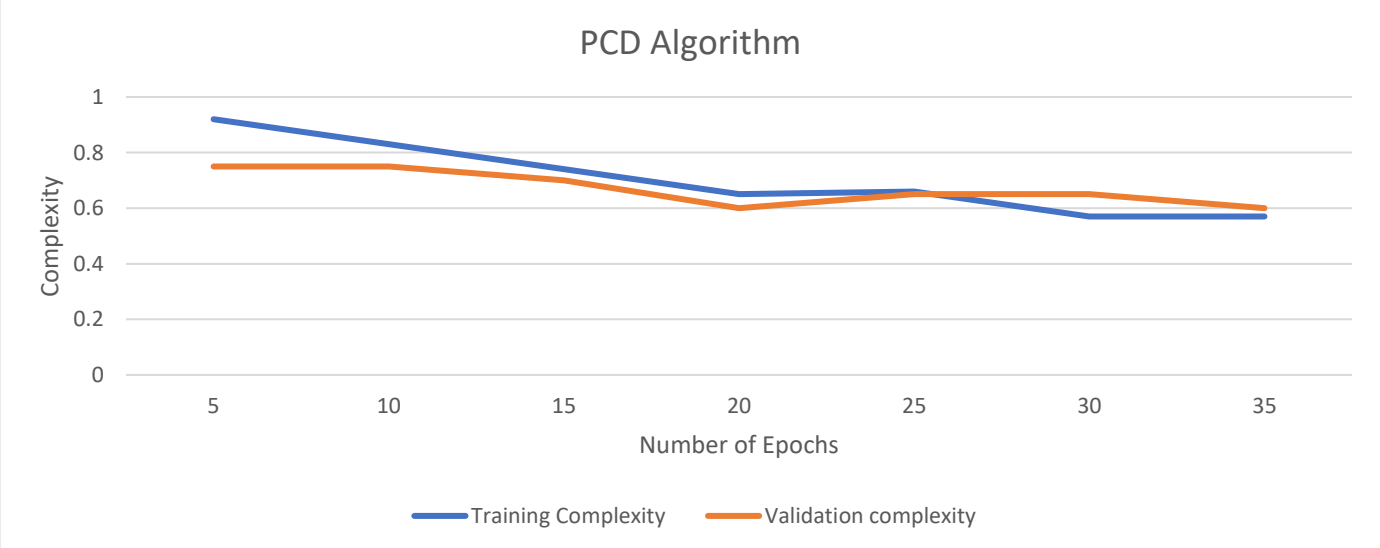

Figure 4. Training Complexity of PCD Algorithm.

Journal of Information Systems Engineering and Management
2025, 10(48s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

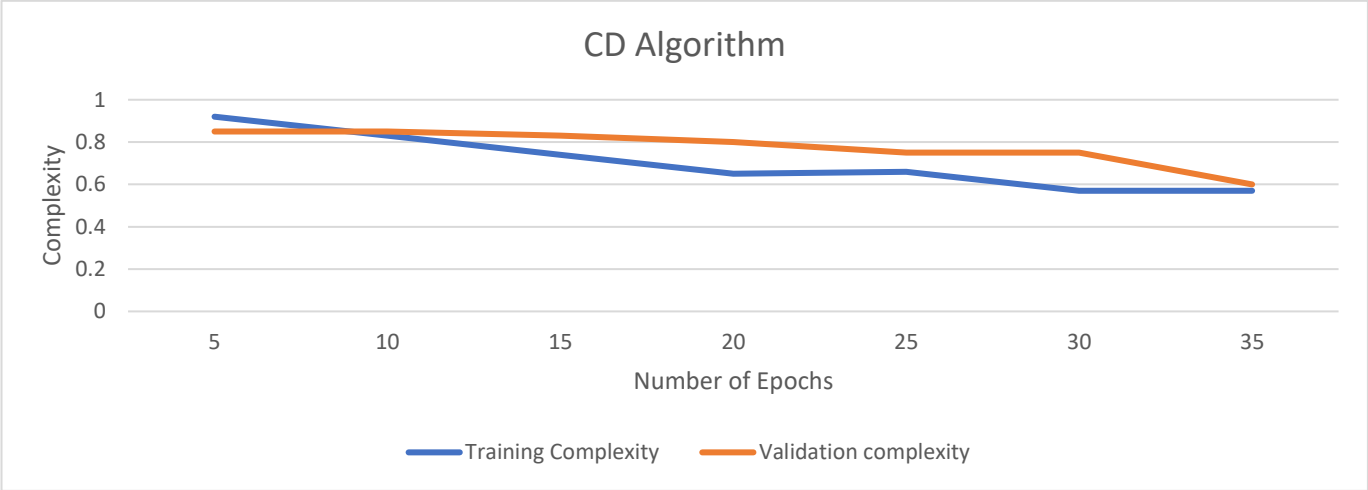

Figure 5. Training Complexity of CD Algorithm.

RESULTS

Time Complexity

When compared with the existing algorithms, the proposed approach had the fastest training time by taking the model in 500 seconds, whereas other models took 600 seconds and 800 seconds. This is explained by the suggested algorithm's adaptive learning rate, which more effectively optimizes parameter changes.

Accuracy and Performance Metrics

The suggested algorithm surpassed CD and PCD when it comes to F1 score and recall, and it also attained a maximum accuracy of 89.5%. This suggests that by using the real-time dataset, the approach was able to acquire more discriminative characteristics and improve its generalization.

The above diagram shows the complexity of the algorithms CD and PCD, where it is clear that complexity reduces as the number of epochs increases. While CD and PCD showed slightly longer training times and lower accuracies, they exhibited stable convergence during training. However, the proposed algorithm's faster convergence and higher accuracy make it a preferable choice for training DBNs on image classification tasks. The proposed algorithm demonstrated low sensitivity to hyperparameter choices, as the default parameters provided good results. CD and PCD may require more fine-tuning of hyperparameters to achieve optimal performance.

Table 4. Comparison of training Algorithms.

Aspect	Proposed Algorithm	Contrastive Divergence (CD)	Persistent Contrastive Divergence (PCD)
Training Time (seconds)	500	800	600
Time Complexity	O(I)	O(K×I×E)	O(I×E)
Training Complexity	Low	Moderate to High	Moderate
Convergence Speed	Fast	Moderate	Moderate
Stability	Stable	Variable	Stable
Hyperparameter Sensitivity	Low	Moderate	Low
Generalization Performance	Good	Moderate	Good
Recall	0.85	0.75	0.82
F1 Score	0.87	0.78	0.85
Accuracy	0.89	0.80	0.86

Research Article

As seen in Table 4 above, the time required for learning about a specific dataset and the design of the model is indicated by the symbol "time," which is expressed in seconds. Big O notation is used to express time complexity, where I denote iterations, E denotes examples, and K denotes Gibbs sampling steps. The phrase "training complexity" describes the degree of algorithmic and computational complexity. The convergence. During training, speed indicates the speed at which the algorithm converges or finds a stable solution. The training process's uniformity and dependability are referred to as stability. Hyperparameter Sensitivity evaluates how responsive the algorithm can be to changes in the hyperparameters.

CONCLUSION

Deep Belief Networks (DBNs) are a stack of networks, each having picked up unique characteristics and attributes from the original data. DBNs can handle supervised and unsupervised tasks thanks to their intricate layer-wise neural architecture. We presented an adaptive approach for training DBN and also analyzed the various training algorithms used in the process of training DBNs. We started by delving into the pre-training phase, where Restricted Boltzmann Machines (RBMs) play a central role. We reviewed the Contrastive Divergence (CD) and Persistent Contrastive Divergence (PCD) algorithms, highlighting their advantages and disadvantages in initializing deep belief nets. Moving to the fine-tuning stage, we explored the use of backpropagation with gradient descent. Furthermore, the architectural variants of DBNs like CDBNs and RDBNs with their respective areas of application were discussed. CDBNs had an accuracy of over 95% when operated on standard image classification benchmarks like MNIST and ImageNet whereas RDBNs achieved an accuracy of over 90% on sentiment analysis and 85% for speech recognition on longer audio sequences. We highlighted the adaptation of DBNs for specific tasks, including classification, regression, clustering, and generative modeling. We also had compared the training complexity of the proposed algorithm with the existing algorithm. We found that the proposed algorithm outperformed other algorithms by training the model in 500 seconds whereas the existing algorithms like CD and PCD took 800 seconds and 600 seconds respectively.

REFRENCES

[1] Yuming Hua, Junhai Guo and Hua Zhao, "Deep Belief Networks and deep learning," Proceedings of 2015 International Conference on Intelligent Computing and Internet of Things, Harbin, pp. 1-4, 2015. .

[2] N. Senthilkumar, S. Karpakam, M. Gayathri Devi, R. Balakumaresan, P. Dhilipkumar, “Speech emotion recognition based on Bi-directional LSTM architecture and deep belief networks,” Materials Today: Proceedings, vol. 57, no. 5, pp. 2180-2184, 2022.

[3] Wang, Q., Gao, X., Li, X. et al. “A precise method for RBMs training using phased curricula,” Multimed Tools Appl, vol. 82, pp. 8013–8047, 2023.

[4] Fabien Lareyre, Bahaa Nasr, Arindam Chaudhuri, Gilles Di Lorenzo, Mathieu Carlier, Juliette Raffort, “Comprehensive Review of Natural Language Processing (NLP) in Vascular Surgery,” EJVES Vascular Forum, vol. 60, pp. 57-63, 2023.

[5] Fred Nugen, Diana V. Vera Garcia, Sunghwan Sohn, John P. Mickley, Cody C. Wyles, Bradley J. Erickson, Michael J. Taunton, “Application of Natural Language Processing in Total Joint Arthroplasty: Opportunities and Challenges,” The Journal of Arthroplasty, vol. 38, no. 10, pp. 1948-1953, 2023.

[6] Xueqin Lü, Liyuan Long, Ruiyu Deng, Ruidong Meng, “Image feature extraction based on fuzzy restricted Boltzmann machine,” Measurement, vol. 204, pp. 112063, 2022.

[7] Nan Zhang, Shifei Ding, Jian Zhang, Yu Xue, “An overview on Restricted Boltzmann Machines,” Neurocomputing, vol. 275, pp. 1186-1199, 2018.

[8] Lin Ning, Randall Pittman, Xipeng Shen, “LCD: A Fast Contrastive Divergence Based Algorithm for Restricted Boltzmann Machine,” Neural Networks, vol. 108, pp. 399-410, 2018.

[9] Asja Fischer, Christian Igel, “Training restricted Boltzmann machines: An introduction,” Pattern Recognition, vol. 47, no. 1, pp. 25-39, 2014.

[10] João Paulo Papa, Walter Scheirer, David Daniel Cox, “Fine-tuning Deep Belief Networks using Harmony Search,” Applied Soft Computing, vol. 46, pp. 875-885, 2016.

Research Article

[11] Mei Yang, Ming K. Lim, Yingchi Qu, Xingzhi Li, Du Ni, "Deep neural networks with L1 and L2 regularization for high dimensional corporate credit risk prediction," Expert Systems with Applications, vol. 213, Part A, pp. 118873, 2023.
[12] Sofia C. Pereira, Joana Rocha, Aurélio Campilho, Pedro Sousa, Ana Maria Mendonça, "Lightweight multi-scale classification of chest radiographs via size-specific batch normalization," Computer Methods and Programs in Biomedicine, vol. 236, pp. 107558, 2023.
[13] Guofa Li, Delin Ouyang, Liu Yang, Qingkun Li, Kai Tian, Baiheng Wu, Gang Guo, "Cross-subject EEG linear domain adaption based on batch normalization and depthwise convolutional neural network," Knowledge-Based Systems, pp. 111011, 2023.
[14] Reza Kharghanian, Ali Peiravi, Farshad Moradi, Alexandros Iosifidis, "Pain detection using batch normalized discriminant restricted Boltzmann machine layers," Journal of Visual Communication and Image Representation, vol. 76, pp. 103062, 2021.
[15] Jinrui Wang, Shunming Li, Zenghui An, Xingxing Jiang, Weiwei Qian, Shanshan Ji, "Batch-normalized deep neural networks for achieving fast intelligent fault diagnosis of machines," Neurocomputing, vol. 329, pp. 53-65, 2019.
[16] Ming Ding, "The road from MLE to EM to VAE: A brief tutorial," AI Open, vol. 3, pp. 29-34, 2022.
[17] Rachid Zeghlache, Mohamed Aymen Labiod, Abdelhamid Mellouk, "Driver vigilance estimation with Bayesian LSTM Auto-encoder and XGBoost using EEG/EOG data," IFAC-PapersOnLine, vol. 55, no. 8, pp. 89-94, 2022.
[18] Alexandre Lemire Paquin, Brahim Chaib-draa, Philippe Giguère, "Stability analysis of stochastic gradient descent for homogeneous neural networks and linear classifiers," Neural Networks, vol. 164, pp. 382-394, 2023.
[19] Wilfrido Gómez-Flores, Humberto Sossa, "Learning smooth dendrite morphological neurons by stochastic gradient descent for pattern classification," Neural Networks, vol. 168, pp. 665-676, 2023.
[20] Lingxiao Zhao, Zhiyang Li, Leilei Qu, Junsheng Zhang, Bin Teng, "A hybrid VMD-LSTM/GRU model to predict non-stationary and irregular waves on the east coast of China," Ocean Engineering, vol. 276, pp. 114136, 2023.
[21] Zhuoyue Guo, Canyun Yang, Dongsheng Wang, Hongbin Liu, "A novel deep learning model integrating CNN and GRU to predict particulate matter concentrations," Process Safety and Environmental Protection, vol. 173, pp. 604-613, 2023.
[22] Sergei Manzhos, Manabu Ihara, "Rectangularization of Gaussian process regression for optimization of hyperparameters," Machine Learning with Applications, vol. 13, pp. 100487, 2023.
[23] Inas Abuqaddom, Basel A. Mahafzah, Hossam Faris, "Oriented stochastic loss descent algorithm to train very deep multi-layer neural networks without vanishing gradients," Knowledge-Based Systems, vol. 230, pp. 107391, 2021.
[24] Xin Wang, Yi Qin, Yi Wang, Sheng Xiang, Haizhou Chen, "ReLTanh: An activation function with vanishing gradient resistance for SAE-based DNNs and its application to rotating machinery fault diagnosis," Neurocomputing, vol. 363, pp. 88-98, 2019.
[25] Dong Liu, Yong Wang, Chenhong Luo, Jun Ma, "An improved autoencoder for recommendation to alleviate the vanishing gradient problem," Knowledge-Based Systems, vol. 263, pp. 110254, 2023.
[26] Xuetong Wu, Jonathan H. Manton, Uwe Aickelin, Jingge Zhu, "A Bayesian approach to (online) transfer learning: Theory and algorithms," Artificial Intelligence, vol. 324, pp. 103991, 2023.
[27] Juhyeon Kim, Joonsuk Huh, Daniel K. Park, "Classical-to-quantum convolutional neural network transfer learning," Neurocomputing, vol. 555, pp. 126643, 2023.
[28] Uiwon Hwang, Sung-Woo Kim, Dahuin Jung, SeungWook Kim, Hyejoo Lee, Sang Won Seo, Joon-Kyung Seong, Sungroh Yoon, "Real-world prediction of preclinical Alzheimer's disease with a deep generative model," Artificial Intelligence in Medicine, vol. 144, pp. 102654, 2023.
[29] Yong Shi, Mengyu Shang, Zhiquan Qi, "Intelligent layout generation based on deep generative models: A comprehensive survey," Information Fusion, vol. 100, pp. 101940, 2023.