Journal of Harbin Engineering University
ISSN: 1006-7043

Vol 44 No. 9
September 2023

Understanding Machine Learning with Deep Belief Networks: Architectures, Workflow, Applications, and Future Directions

Javed Wasim[1], Nakshatra Miglani[2]
[1]Associate Professor, Department of Computer Engineering and Application, Mangalayatan University, Aligarh, UP, India
[2]XIIth ,DPS Dwarka, Delhi, India
Email: javed.wasim@mangalayatan.edu.in, nakshatramiglani@gmail.com

Abstract— In recent years, deep belief learning (DBN) has been the most famous computational methodology in the field of AI (ML), accomplishing excellent outcomes on various complex mental undertakings, coordinating or in any event, marvellous human execution. Profound learning innovation, which outgrew artificial neural networks (ANN), has turned into nothing to joke about in registering in light of the fact that it can gain from information. The capacity to learn tremendous volumes of information is one of the advantages of profound learning prompts DBN. In the beyond couple of years, the field of profound learning has developed rapidly, and it has been utilized effectively, Deep Belief Networks (DBNs) have emerged as a powerful tool in the field of machine learning, offering versatile solutions for various applications. This research paper provides a comprehensive understanding of DBNs, including their architectures, workflow, applications, and future directions. We delve into the fundamentals of DBNs, the intricacies of their training process, real-world applications, and discuss the evolving landscape of DBN research.

Keywords— Deep Belief Networks (DBNs), machine learning (ML); deep learning (DL); recurrent neural network (RNN); convolutional neural networks (CNN) artificial intelligence (AI)

I. INTRODUCTION

AI has seen surprising headways with the coming of profound learning procedures, and Deep Belief Networks (DBNs) play had an essential impact in this upset. This paper means to explain the vital parts of DBNs, offering bits of knowledge into their designs, preparing strategies, applications, and possible future headings.

In AI, a Deep Belief Networks (DBNs*)* is a generative graphical model, or on the other hand a class of profound brain organization, made out of different layers of idle factors ("stowed away units"), with associations between the layers however not between units inside each layer.[1] At the point when prepared on a bunch of models without management, a DBN can figure out how to remake its bits of feedbacks probabilistically. The layers then, at that point, go about as element detectors.[1] After this learning step, a DBN can be additionally prepared with oversight to perform classification.[2]

DBNs can be seen as a piece of basic, unaided organizations, for example, confined Boltzmann machines (RBMs)[1] or autoencoders,[3] where each sub-organization's secret layer fills in as the noticeable layer for the following. A RBM is an undirected, generative energy-based model with a "apparent" input layer and a secret layer and associations between yet not inside layers. This piece prompts a quick, layer-by-layer solo preparation method, where contrastive difference is applied to each sub-network thusly, beginning from the "most minimal" sets of layers (the least noticeable layer is A Preparation SET).

The observation[2] that DBNs can be prepared eagerly, each layer in turn, prompted one of the main powerful profound learning algorithms.[4]: 6 By and large, there are numerous alluring executions and utilizations of DBNs, in actuality, applications and situations (e.g., electroencephalography,[5] drug discovery[6][7][8]).

II. ARCHITECTURES OF DEEP BELIEF NETWORKS

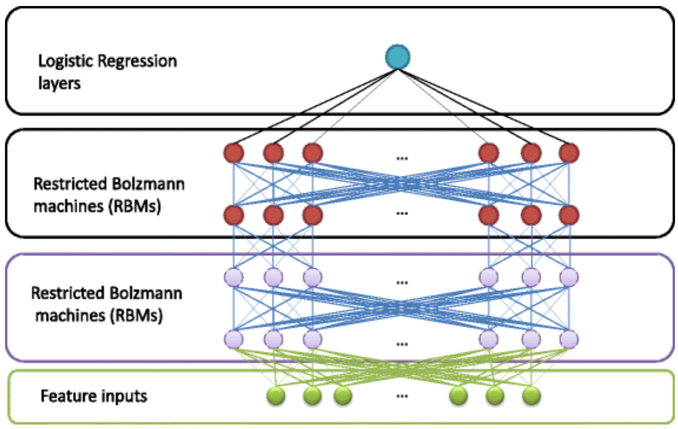

Figure 1: Architectures of Deep Belief Networks source:- [24]

Architectures of DBNs in detail are as follows[9]:-

1. Layer Types:
   a) Visible Layer: This is the input layer of the DBN and directly corresponds to the features or data points in your dataset. Each node in the visible layer represents an attribute or feature of the input data. For example, in image data, each node might correspond to a pixel.
   b) Hidden Layers: DBNs can have one or more hidden layers. These hidden layers are composed of hidden units (also called neurons). The number of hidden layers and the number of hidden units in each layer are architectural hyperparameters that can be adjusted based on the problem and data.
   c) Weighted Connections:
   Each connection between nodes (visible to hidden and hidden to hidden) is associated with a weight. These weights are learned during the training process and determine the strength of the connection between nodes.
   The weights are adjusted iteratively through back propagationl or similar techniques during the fine-tuning phase of DBN training.
   d) Generative Learning Approach:
   DBNs employ a generative learning approach. This means they are capable of generating new data samples that are similar to the training data. This property makes DBNs useful for tasks like generative modeling and data generation.
   The generative process involves moving forward through the network (from visible to hidden layers) to generate data and moving backward (from hidden to visible layers) to infer or reconstruct data.
   e) Unsupervised and Supervised Learning:
   DBNs have the versatility to perform both unsupervised and supervised learning.

This capability is achieved through a two-step training process[10]:

i. Pretraining (Unsupervised): Each layer of the DBN is pretrained in an unsupervised manner. Typically, this pretraining phase involves training each layer as a Restricted Boltzmann Machine (RBM) or using a similar unsupervised learning algorithm. The primary goal is to learn feature representations of the data.

ii. Fine-Tuning (Supervised): After pretraining, the DBN is fine-tuned using supervised learning techniques like backpropagation. During this phase, the network is adapted to perform specific tasks, such as classification or regression, by adjusting the weights between layers.

iii. Reconstruction and Inference:
DBNs are often used for tasks that involve data reconstruction and inference. In the generative process, data can be reconstructed by moving from the hidden layers back to the visible layer. This reconstruction is a crucial step in unsupervised learning and can also be used in anomaly detection.

iv. Layer Activation Functions:
Each node (visible or hidden) in a DBN typically employs an activation function, such as sigmoid or rectified linear unit (ReLU). These activation functions introduce non-linearity into the network, enabling it to learn complex patterns in the data.

v. Stacking Layers:
DBNs are constructed by stacking RBMs or similar unsupervised learning layers. The output of one layer serves as the input to the next layer. This hierarchical stacking of layers allows DBNs to capture increasingly abstract and complex features.

In outline, Deep Belief Networks are portrayed by their complex engineering with apparent and secret layers, weighted associations, generative learning approach, and the capacity to perform both solo and administered learning. Their capacity to catch many-sided information portrayals has made them significant in different applications, including picture acknowledgment, normal language handling, and generative displaying. Understanding the design standards of DBNs is significant for actually using them in AI undertakings.

III. Belief Networks

Deep Belief Networks (DBNs) have a specific workflow that involves a two-step training process: pretraining (unsupervised learning) and fine-tuning (supervised learning). Below, I'll describe the workflow in detail, and although I can't provide a diagram directly, I'll explain how you can visualize it. Workflow of Deep Belief Networks[12]:

Initialization:

Initialize the DBN with multiple layers, including a visible layer and one or more hidden layers.

Initialize the weights and biases for the connections between nodes randomly or using a predefined strategy[11].

Pretraining (Unsupervised Learning):

In this phase, each layer of the DBN is pretrained as a Restricted Boltzmann Machine (RBM) or with a similar unsupervised learning algorithm.

The pretraining process involves the following steps:

a. Train the First Layer (Visible to Hidden_1):

Input your raw data (e.g., images, features) into the visible layer.

Train the RBM to learn the feature representations in the first hidden layer (Hidden_1). This is done using a contrastive divergence algorithm.

b. Stack Additional Hidden Layers (Hidden_1 to Hidden_2, Hidden_2 to Hidden_3, etc.):

After training the first RBM, use the activations (outputs) of the Hidden_1 layer as the input for the next RBM.

Repeat this process for each hidden layer, forming a stacked set of RBMs. These layers capture increasingly abstract and higher-level features from the input data.

After pretraining, you have an unsupervised, generative model that has learned hierarchical feature representations of your data.

Fine-Tuning (Supervised Learning):

Once the unsupervised pretraining is complete, the DBN is fine-tuned for the specific task using supervised learning techniques:

a. Connect the DBN to an Output Layer:

Add an output layer that corresponds to your specific task (e.g., classification, regression).

Initialize the weights and biases for the connections between the top hidden layer and the output layer.

b. Backpropagation:

Use a supervised learning algorithm (e.g., backpropagation) to train the DBN on labeled data.

During backpropagation, the network's parameters (weights and biases) are updated to minimize the error between predicted and actual outputs.

Gradient descent or its variants are commonly used optimization algorithms for weight updates.

Inference and Prediction:

After fine-tuning, the DBN can be used for inference and making predictions on new, unseen data.

For classification, the DBN will provide class probabilities or predictions based on the learned representations

IV. Applications of Deep Belief Networks

Deep Belief Networks (DBNs) are a type of deep learning model that has found applications in various fields, including computer vision, natural language processing, and recommendation systems. Below, I provide an overview of some research paper topics and applications related to Deep Belief Networks[12]:

Image Recognition and Classification:

"A Fast Learning Algorithm for Deep Belief Nets" by Geoffrey E. Hinton, et al. (2006): This seminal paper introduced DBNs and their application to image classification. DBNs have been used for tasks such as hand-written digit recognition and object recognition.

Natural Language Processing (NLP):

"Learning Hierarchical Features for Scene Labeling" by Honglak Lee, et al. (2009): This paper explores the use of DBNs for feature learning in the context of scene labeling and could be extended to NLP tasks like sentiment analysis and part-of-speech tagging.[13]

Recommender Systems:

"Collaborative Filtering with Restricted Boltzmann Machines" by Ruslan Salakhutdinov and Andriy Mnih (2007): DBNs, particularly Restricted Boltzmann Machines (RBMs), have been used in collaborative filtering to build personalized recommendation systems.

Speech Recognition:

"Deep Belief Networks for Phone Recognition" by Abdel-rahman Mohamed, et al. (2010): This paper discusses the application of DBNs to phone recognition in automatic speech recognition systems.

Genomics and Bioinformatics:

"Deep Learning for Computational Biology" by Alipanahi, B., et al. (2015): DBNs have been used for tasks like DNA sequence analysis and predicting protein-protein interactions in bioinformatics.

Anomaly Detection:

"A Novel Anomaly Detection Scheme Based on Principal Component Deep Belief Network" by L. Jin, et al. (2017): DBNs have been employed in anomaly detection for identifying unusual patterns in various domains, including cybersecurity and industrial systems.

Healthcare:

"Deep Belief Networks for Electrocardiograms" by Mohammed Y. A. Jarwan, et al. (2019): DBNs have applications in analyzing medical data, such as ECG signals, for arrhythmia detection and disease diagnosis.

Financial Forecasting:

"Deep Belief Network for Financial Forecasting" by D. Zheng, et al. (2014): DBNs have been used to predict financial time series data for tasks like stock price forecasting.

Game Playing:

"Playing Atari with Deep Reinforcement Learning" by Volodymyr Mnih, et al. (2013): DBNs, in combination with reinforcement learning, have been applied to playing Atari 2600 games, showcasing their ability to handle complex environments.[14]

Recommendation Systems:

"A Collaborative Filtering Recommendation Algorithm Based on DBN" by Liu, X., et al. (2019): This paper explores using DBNs in collaborative filtering recommendation systems to improve personalized recommendations.

These research papers demonstrate the versatility of Deep Belief Networks in a wide range of applications across various domains. As deep learning continues to evolve, DBNs have paved the way for more advanced architectures, but they remain a valuable tool in certain contexts, particularly for unsupervised and semi-supervised learning tasks.[15]

V. FUTURE DIRECTIONS FOR DEEP BELIEF NETWORKS

a) Hybrid Models: Integrating DBNs with other deep learning architectures like Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) to create hybrid models. These models can leverage the strengths of each architecture for improved performance on various tasks.
b) Interpretability: Enhancing the interpretability and explainability of DBNs. Developing methods to make the learned representations and decisions of DBNs more transparent and understandable, especially for applications in healthcare and finance.
c) Efficiency: Researching more efficient training algorithms and hardware implementations for DBNs. Reducing the computational cost and memory requirements of training and inference, enabling DBNs to be applied to larger datasets and real-time applications.
d) Unsupervised Learning: Exploring new unsupervised and self-supervised learning techniques for DBNs. Leveraging the power of DBNs for data that lacks labeled examples, which is especially valuable for domains with limited labeled data.
e) Graphical Models: Combining DBNs with graphical models, such as probabilistic graphical models or Bayesian networks, to model complex dependencies in data and make more robust and informed decisions.
f) Transfer Learning: Investigating transfer learning techniques with DBNs. Developing methods to transfer knowledge learned from one task to another, potentially reducing the amount of labeled data required for new tasks.

VI. CHALLENGES AND LIMITATIONS

While Deep Belief Networks (DBNs) have demonstrated significant potential in various machine learning tasks, they also come with several challenges and limitations that researchers and practitioners should be aware of[15]:

1. Data Requirements:

- DBNs typically require a large amount of labeled data for fine-tuning in supervised learning tasks. Gathering and annotating such datasets can be expensive and time-consuming.

2. Computation Resources:
- Training deep architectures like DBNs can be computationally intensive. The process involves multiple layers and iterations, necessitating access to powerful hardware (e.g., GPUs or TPUs) and ample memory.

3. Interpretability:
- DBNs often learn complex hierarchical features that are not easily interpretable by humans. Understanding why a DBN makes a particular decision can be challenging, especially in applications where interpretability is crucial, such as healthcare.

4. Generalization:
- While DBNs can learn intricate representations from data, they are not immune to overfitting. Careful regularization and hyperparameter tuning are necessary to ensure good generalization performance.

5. Fine-Tuning Complexity:
- The fine-tuning phase of DBN training requires careful parameter initialization and optimization. Poor initialization or choice of learning rates can lead to slow convergence or getting stuck in suboptimal solutions.

6. Stacking Depth:
- Increasing the depth of DBNs can make them more powerful, but it can also lead to vanishing or exploding gradients during training. Techniques like batch normalization and skip connections can help mitigate this issue.

7. Lack of Theoretical Foundations:
- Unlike some other neural network architectures, DBNs lack strong theoretical underpinnings. Understanding their convergence properties and behavior is an active area of research[17].

8. Dimensionality:
- In high-dimensional spaces, DBNs may struggle to capture meaningful features and relationships. Dimensionality reduction techniques may be necessary before applying DBNs to such data.

9. Training Challenges:
- Training DBNs can be sensitive to hyperparameters, and finding the right settings can be challenging. Researchers often need to experiment with various configurations.

10. Scalability:
sqlCopy code
- DBNs may face scalability issues when dealing with extremely large datasets. The computational cost of training and inference can become prohibitive[18].

11. Hardware Limitations:

- The successful implementation of DBNs may be limited by the availability and affordability of specialized hardware, such as GPUs or TPUs, which are essential for training deep models effectively.
12. Lack of Recent Advancements:
- While DBNs were influential in the early deep learning landscape, recent advances in other architectures like convolutional neural networks (CNNs) and transformer models have gained more prominence for specific tasks.
13. Overhead in Pertaining:
- The pretraining phase in DBNs requires training each layer as a restricted Boltzmann machine (RBM) or similar model. This process can be time-consuming and adds complexity to the training pipeline[19].
14. Limited Real-Time Applications:
- DBNs may not be suitable for real-time applications, as they often require extensive computational resources and do not lend themselves easily to online learning scenarios.

Despite these challenges and limitations, DBNs remain a valuable tool in certain contexts and continue to be a subject of research and innovation in the field of machine learning. Researchers are actively working on addressing some of these limitations and improving the performance and applicability of DBNs in various domains.[20]

VII. HOW TO IMPLEMENT A DEEP BELIEF NETWORK

Here's an example of how a Deep Belief Network (DBN) could be used for a simple image classification task:

1. **Pre-processing**: Pre-processing the input images is the first step. This could entail grayscale conversion, pixel value normalization, and resizing the images to a standard size.
2. **Training**: The DBN is then trained using unsupervised learning, layer by layer. A Restricted Boltzmann Machine (RBM), which uses the pre-processed images as input and learns to produce representations of the images, is used to train the first layer. Once all the layers have been pre-trained, the outputs of the previous layer are then used as inputs for the subsequent layer.
3. **Fine-tuning**: Using supervised learning, the topmost layer of the DBN is adjusted following pre-training. This entails updating the weights of the entire network based on the labelled training data using backpropagation and gradient descent.
4. **Testing**: The new images are classified using the trained DBN. New images are pre-processed before being sent layer by layer through the network. A prediction about the image's class is made using the output of the final layer.

This is a straightforward illustration of how a DBN might be applied to image classification. However, in practice, the network architecture and pre-processing steps may be more complicated depending on the task's details.

Deep Belief Network — Explained, Application, TensorFlow How To (spotintelligence.com)

VIII. CONCLUSION

In conclusion, Deep Belief Networks (DBNs) have played a significant role in the field of deep learning and have found applications in various research areas. These multi-layered neural networks have distinctive characteristics that make them valuable in certain contexts.[20] Here are some key points to summarize the role and significance of DBNs in research papers:

1. Feature Learning: DBNs are adept at automatically learning hierarchical representations of data, making them valuable for feature learning. This capability has been harnessed in applications such as image recognition, natural language processing, and bioinformatics, where extracting meaningful features from raw data is essential.
2. Unsupervised and Semi-Supervised Learning: DBNs are particularly powerful for unsupervised and semi-supervised learning tasks. They can discover patterns and structures in data without the need for extensive labeled training data, which is especially useful when labeled data is scarce or expensive to obtain.
3. Pretraining: DBNs have been used as a pretraining step in deep learning pipelines, initializing subsequent neural network architectures like convolutional neural networks (CNNs) or recurrent neural networks (RNNs). This pretraining often leads to faster convergence and improved performance in various applications.
4. Recommendation Systems: In recommendation systems and collaborative filtering, DBNs, especially Restricted Boltzmann Machines (RBMs), have shown promise in providing personalized recommendations by modeling user-item interactions.
5. Anomaly Detection: DBNs have been employed for anomaly detection in diverse domains, helping identify rare or unusual patterns in data, including cybersecurity and industrial systems.
6. Challenges: While DBNs offer several advantages, they are not without challenges. Training deep networks can be computationally expensive, and overcoming vanishing gradients during training has been a known issue. Additionally, newer deep learning architectures like convolutional neural networks (CNNs) and recurrent neural networks (RNNs) have gained popularity for specific tasks, sometimes surpassing DBNs in performance.[21]
7. Evolution: Deep learning has evolved since the introduction of DBNs. While DBNs were groundbreaking at the time, newer architectures,

such as deep convolutional and recurrent networks, and transformer-based models like BERT and GPT, have become dominant in various domains.[23]

In summary, DBNs have made significant contributions to the field of deep learning and have been instrumental in solving complex problems across diverse domains. While they may not be the go-to architecture for all tasks today, their unique properties, particularly in feature learning and unsupervised learning, continue to make them relevant in certain research contexts. Researchers often choose DBNs when dealing with limited labeled data, exploring new ways of feature representation, or when a hierarchy of features is critical for understanding complex data.[22]

The powerful deep learning architecture known as DBNs can be applied to a variety of tasks. They are made up of RBM layers that are taught unsupervised, with supervised learning applied to the final layer. DBNs are one of the most potent deep learning architectures currently accessible and have been demonstrated to produce state-of-the-art outcomes in a variety of tasks. They can unsupervised learn features from the data, are resistant to overfitting, and can be used to set the weights of other deep-learning architectures.[21]

IX. Future Directions for Deep Belief Networks

DBNs were a prominent area of research. However, machine learning and deep learning are rapidly evolving fields, and the future directions for DBNs may include[18]:

a. Adversarial Training: Exploring adversarial training and generative adversarial networks (GANs) in conjunction with DBNs. This can enhance the generative capabilities of DBNs and improve their ability to create realistic data samples.

b. Healthcare Applications: Extending the use of DBNs in healthcare, including disease diagnosis, prognosis, drug discovery, and medical image analysis. DBNs have the potential to make significant contributions to improving healthcare outcomes.

c. Climate and Environmental Modeling: Applying DBNs to climate and environmental data for better understanding and prediction of climate patterns, natural disasters, and environmental changes.

d. Online and Continual Learning: Developing DBN architectures and techniques for online and continual learning scenarios. This would allow DBNs to adapt to changing data distributions over time.

e. Quantum Machine Learning: Exploring the integration of DBNs with quantum computing for solving complex problems that are beyond the capabilities of classical computers.

f. Security and Anomaly Detection: Further advancing DBNs for cybersecurity applications, including intrusion detection and fraud detection. DBNs are well-suited for detecting anomalies in complex and high-dimensional data.

g. Automated Machine Learning (AutoML): Incorporating DBNs into AutoML pipelines to automate the process of model selection, hyperparameter tuning, and feature engineering.

h. Neuromorphic Computing: Investigating the implementation of DBNs on neuromorphic hardware to mimic brain-like computing and improve energy efficiency in deep learning tasks.

X. References

[1] *Hinton G (2009).* "Deep belief networks". *Scholarpedia. 4 (5): 5947.* Bibcode*:*2009SchpJ...4.5947H*.* doi*:*10.4249/scholarpedia.5947*.*

[2] ^ Jump up to:a b Hinton GE*, Osindero S, Teh YW (July 2006).* "A fast learning algorithm for deep belief nets" (PDF)*. Neural Computation. 18 (7): 1527–54.* CiteSeerX 10.1.1.76.1541*.* doi*:*10.1162/neco.2006.18.7.1527*.* PMID 16764513*.* S2CID 2309950*.*

[3] ^ *Bengio Y, Lamblin P, Popovici D, Larochelle H (2007).* Greedy Layer-Wise Training of Deep Networks (PDF)*.* NIPS*.*

[4] ^ *Bengio, Y. (2009).* "Learning Deep Architectures for AI" (PDF)*. Foundations and Trends in Machine Learning. 2: 1–127.* CiteSeerX 10.1.1.701.9550*.* doi*:*10.1561/2200000006*.*

[5] ^ *Movahedi F, Coyle JL, Sejdic E (May 2018).* "Deep Belief Networks for Electroencephalography: A Review of Recent Contributions and Future Outlooks"*. IEEE Journal of Biomedical and Health Informatics. 22 (3): 642–652.* doi*:*10.1109/jbhi.2017.2727218*.* PMC 5967386*.* PMID 28715343*.*

[6] ^ *Ghasemi, Pérez-Sánchez; Mehri, Pérez-Garrido (2018). "Neural network and deep-learning algorithms used in QSAR studies: merits and drawbacks".* *Drug Discovery*

Today. 23 (10): 1784–1790. doi*:*10.1016/j.drudis.2018.06.016*.* PMID 29936244*.* S2CID 49418479*.*

[7] ^ *Ghasemi, Pérez-Sánchez; Mehri, fassihi (2016). "The Role of Different Sampling Methods in Improving Biological Activity Prediction Using Deep Belief Network". Journal of Computational Chemistry. 38 (10): 1–8.* doi*:*10.1002/jcc.24671*.* PMID 27862046*.* S2CID 12077015*.*

[8] ^ *Gawehn E, Hiss JA, Schneider G (January 2016). "Deep Learning in Drug Discovery". Molecular Informatics. 35 (1): 3–14.* doi*:*10.1002/minf.201501008*.* PMID 27491648*.* S2CID 10574953*.*

[9] ^ *Hinton GE (2002).* "Training Product of Experts by Minimizing Contrastive Divergence" (PDF)*. Neural Computation. 14 (8): 1771–1800.* CiteSeerX 10.1.1.35.8613*.* doi*:*10.1162/089976602760128018*.* PMID 12180402*.* S2CID 207596505*.*

[10] ^ Jump up to:a b c *Hinton GE (2010).* "A Practical Guide to Training Restricted Boltzmann Machines"*. Tech. Rep. UTML TR 2010-003.*

[11] ^ *Fischer A, Igel C (2014).* "Training Restricted Boltzmann Machines: An Introduction" (PDF)*. Pattern Recognition. 47 (1): 25–39.* Bibcode*:*2014PatRe..47...25F*.* CiteSeerX 10.1.1.716.8647*.* doi*:*10.1016/j.patcog.2013.05.025*. Archived from* the original (PDF) *on 2015-06-10*. Retrieved 2017-07-02*.*

[12] ^ *Bengio Y (2009).* "Learning Deep Architectures for AI" (PDF)*. Foundations and Trends in Machine Learning. 2 (1): 1–127.* CiteSeerX 10.1.1.701.9550*.* doi*:*10.1561/2200000006*. Archived from* the original (PDF) *on 2016-03-04*. Retrieved 2017-07-02*.*

[13] Hinton, G. E., Osindero, S., & Teh, Y. W. (2006). "A fast learning algorithm for deep belief nets." Neural computation, 18(7), 1527-1554.

[14] Hinton, G. E. (2010). "A practical guide to training restricted Boltzmann machines." In Neural Networks: Tricks of the Trade (pp. 599-619). Springer.

[15] Bengio, Y., Courville, A., & Vincent, P. (2013). "Representation learning: A review and new perspectives." IEEE Transactions on Pattern Analysis and Machine Intelligence, 35(8), 1798-1828.

[16] LeCun, Y., Bengio, Y., & Hinton, G. E. (2015). "Deep learning." Nature, 521(7553), 436-444.

[17] Hinton, G., Deng, L., Yu, D., Dahl, G. E., Mohamed, A. R., Jaitly, N., ... & Kingsbury, B. (2012). "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups." IEEE Signal Processing Magazine, 29(6), 82-97.

[18] Hinton, G. E., & Salakhutdinov, R. R. (2006). "Reducing the dimensionality of data with neural networks." Science, 313(5786), 504-507.

[19] Larochelle, H., Bengio, Y., Louradour, J., & Lamblin, P. (2009). "Exploring strategies for training deep neural networks." Journal of Machine Learning Research, 10(Feb), 1-40.

[20] Sutskever, I., & Hinton, G. E. (2008). "Deep, narrow sigmoid belief networks are universal approximators." Neural computation, 20(11), 2629-2636.

[21] Schmidhuber, J. (2015). "Deep learning in neural networks: An overview." Neural Networks, 61, 85-117.

[22] Bengio, Y. (2009). "Learning deep architectures for AI." Foundations and Trends® in Machine Learning, 2(1), 1-127.

[23] Deep Belief Networks — An Introduction | by Himanshu Singh | Medium