International Journal of Engineering Research in Africa
ISSN: 1663-4144, Vol. 24, pp 124-136
doi:10.4028/www.scientific.net/JERA.24.124
© 2016 Trans Tech Publications, Switzerland
Submitted: 2016-03-03
Revised: 2016-04-20
Accepted: 2016-04-20
Online: 2016-06-01

A Review of Deep Machine Learning

Ben-Bright Benuwa[1,2, a], Yongzhao Zhan [1,b], Benjamin Ghansah[1,2,c*], Dickson Keddy Wornyo[1,d] and Frank Banaseka Kataka[1,e]

[1] School of Computer Science and Telecommunication Engineering, Jiangsu University, Xuefu Road 301 Jingkou District Zhenjiang province Jiangsu City, Zhenjiang 212013, China

[2] School Of Computer Science, Data Link Institute P. O Box 2481 Tema Ghana, West Africa.

[a] benuwa778@gmail.com, [b] yzzhan@ujs.edu.cn, [c] ben@datalink.edu.gh, [d]macdicksons@yahoo.com, [e] frankbanaseka@yahoo.com

Keywords: Deep learning, Deep belief networks, feature learning, unsupervised learning, Boltzmann Machine, neural nets

Abstract. The rapid increase of information and accessibility in recent years has activated a paradigm shift in algorithm design for artificial intelligence. Recently, Deep Learning (a surrogate of Machine Learning) have won several contests in pattern recognition and machine learning. This review comprehensively summarises relevant studies, much of it from prior state-of-the-art techniques. This paper also discusses the motivations and principles regarding learning algorithms for deep architectures.

1. Introduction

Deep learning (deep structured learning, hierarchical learning or deep machine learning) is a branch of machine learning based on a set of algorithms that attempt to model high-level abstractions in data by using multiple processing layers with complex structures, or otherwise composed of multiple non-linear transformations [1-6]. Deep learning is part of a broader family of machine learning methods based on learning representations of data. An observation (e.g., an image) can be represented in many ways such as a vector of intensity values per pixel, or in a more abstract way as a set of edges, regions of particular shape, etc. Some representations make it easier to learn tasks (e.g., face recognition or facial expression recognition [7]) from examples. One of the potentials of deep learning is replacing handcrafted features with efficient algorithms for unsupervised or semi-supervised feature learning and hierarchical feature extraction [8, 9].

Studies in this area attempts to make better representations and create models to learn these representations from large-scale unlabeled data. Some of the representations are inspired by advances in neuroscience and are loosely based on interpretation of information processing and communication patterns in a nervous system, such as neural coding which attempts to define a relationship between various stimuli and associated neuronal responses in the brain [10].

Various deep learning architectures such as deep neural networks, convolutional deep neural networks, deep belief networks and recurrent neural networks have been applied to fields like computer vision, automatic speech recognition, natural language processing, audio recognition and bioinformatics where they have been shown to produce state-of-the-art results on various tasks. Alternatively, *deep learning* has been characterized as a buzzword, or a rebranding of neural networks [11, 12]. Deep learning could be characterized as a class of machine learning algorithms that Use a cascade of many layers of nonlinear processing units for feature extraction and transformation. Each successive layer uses the output from the previous layer as input. The algorithms may be supervised or unsupervised and applications include pattern analysis (unsupervised) and classification (supervised).

Are based on the (unsupervised) learning of multiple levels of features or representations of the data. Higher level features are derived from lower level features to form a hierarchical representation. Are part of the broader machine learning field of learning representations of data.

Learn multiple levels of representations that correspond to different levels of abstraction; the levels form a hierarchy of concepts [1].

All rights reserved. No part of contents of this paper may be reproduced or transmitted in any form or by any means without the written permission of Trans Tech Publications, www.ttp.net. (#69048248, Purdue University Libraries, West Lafayette, USA-15/08/16,14:17:35)

These characterizations have in common (1) multiple layers of nonlinear processing units and (2) the supervised or unsupervised learning of feature representations in each layer, with the layers forming a hierarchy from low-level to high-level features [1]. The composition of a layer of nonlinear processing units used in a deep learning algorithm depends on the problem to be solved. Layers that have been used in deep learning include hidden layers of an artificial neural network and sets of complicated propositional formulas [2]. They may also include latent variables organized layer-wise in deep generative models such as the nodes in Deep Belief Networks [13] and Deep Boltzmann Machines [14, 15].

Neural networks get their representations from using layers of learning. Primate brains do a similar thing in the visual cortex, so the hope was that using more layers in a neural network could allow it to learn better models. Nonetheless, studies have shown that the internal representations between these models could not work, but however successful models were realised to be build with a shallow network, one with only a single layer of data representation [16]. Learning in a deep neural network, one with more than one layer of data representation, just wasn't working out. In reality, deep learning has been around for as long as neural networks have existed but were not just good at its implementation as depicted in figures 1 and 2

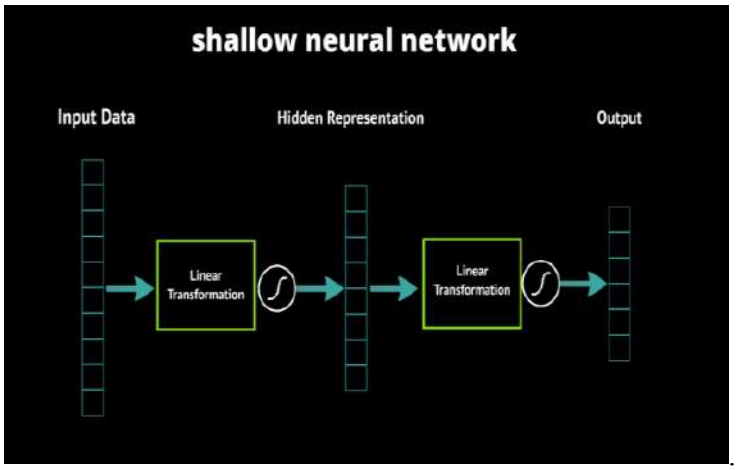

Figure 1: Single Layered Neural Network [17]

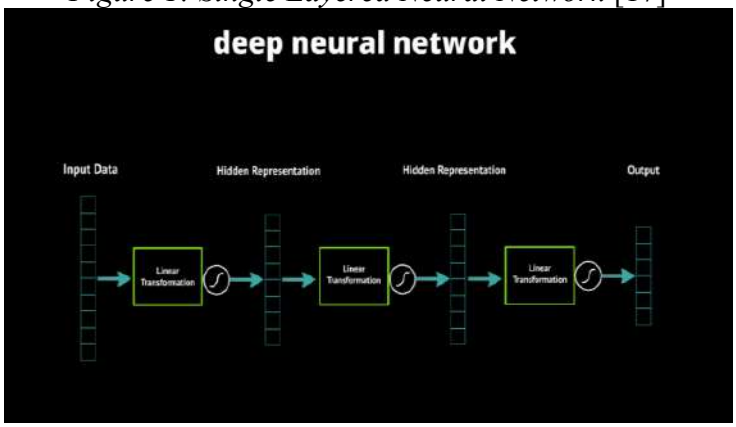

Figure 2 Deep Neural Network [17]

Deep learning algorithms are contrasted with shallow learning algorithms by the number of parameterized transformations a signal encounters as it propagates from the input layer to the output layer, where a parameterized transformation is a processing unit that has trainable parameters, such as weights and thresholds [4]. A chain of transformations from input to output is a *credit assignment path* (CAP). CAPs describe potentially causal connections between input and output and may vary in length. For a feedforward neural network, the depth of the CAPs, and thus the depth of the network, is the number of hidden layers plus one (the output layer is also parameterized). For recurrent neural networks, in which a signal may propagate through a layer more than once, the CAP is potentially unlimited in length. There is no universally agreed upon

threshold of depth dividing shallow learning from deep learning, but most researchers in the field agree that deep learning has multiple nonlinear layers (CAP > 2) and [4]considers CAP > 10 to be very deep learning.

So, what is deep learning?

It's a term that covers a particular approach to building and training neural networks. Neural networks have been around since the 1950s, and like nuclear fusion, they've been an incredibly promising laboratory idea whose practical deployment has been beset by constant delays. They take an array of numbers (that can represent pixels, audio waveforms, or words), run a series of functions on that array, and output one or more numbers as outputs. The outputs are usually a prediction of some properties you're trying to guess from the input, for example whether or not an image is a picture of a cat.

The functions that are run inside the black box are controlled by the memory of the neural network, arrays of numbers known as weights that define how the inputs are combined and recombined to produce the results. Dealing with real-world problems like cat-detection requires very complex functions, which mean these arrays are very large, containing around 60 million numbers in the case of one of the recent computer vision networks. The biggest obstacle to using neural networks has been figuring out how to set all these massive arrays to values that will do a good job transforming the input signals into output predictions.

Training

One of the theoretical properties of neural networks that has kept researchers working on them is that they should be teachable. It's pretty simple to show on a small scale how you can supply a series of example inputs and expected outputs, and go through a mechanical process to take the weights from initial random values to progressively better numbers that produce more accurate predictions (I'll give a practical demonstration of that later). The problem has always been how to do the same thing on much more complex problems like speech recognition or computer vision with far larger numbers of weights.

There was a real breakthrough in the 2012 which was published by an Imagenet paper [18] sparking the current renaissance in neural networks. Alex Krizhevsky, Ilya Sutskever, and Geoff Hinton brought together a whole bunch of different ways of accelerating the learning process, including convolutional networks, clever use of GPUs, and some novel mathematical tricks like ReLUand dropout, and showed that in a few weeks they could train a very complex network to a level that it could outperform conventional approaches to computer vision [1, 18].

This isn't an aberration; similar approaches to have been used very successfully in natural language processing and speech recognition. This is the heart of deep learning — the new techniques that have been discovered that allow us to build and train neural networks to handle previously unsolved problems. With most machine learning, the hard part is identifying the features in the raw input data, for example SIFT or SURF in images [19]. Deep learning removes that manual step, instead relying on the training process to discover the most useful patterns across the input samples. You still have to make choices about the internal layout of the networks before you start training, but the automatic feature discovery makes life a lot easier. In other ways, too, neural networks are more general than most other machine-learning techniques. The same underlying techniques for architecting and training networks are useful across all kinds of natural data, from audio to seismic sensors or natural language. No other approach is nearly as flexible. Deep learning works really well, and if you ever deal with messy data from the real world, it's going to be an essential element in your toolbox over the next few years. Until recently, it's been an obscure and daunting area to learn about, but its success has brought a lot of great resources and projects that make it easier than ever to get started.

The rest of the paper is organized into the following sections; Section 2 presents the fundamental concepts of DL, Section 3 discusses the architectures of DL, Section 4 talks about the various application areas of DL. Finally Section 5 outlines the main conclusions and recommendations.

2. Fundamental Concepts

Deep learning algorithms are based on distributed representations. The underlying assumption behind distributed representations is that observed data is generated by the interactions of factors organized in layers. Deep learning adds the assumption that these layers of factors correspond to levels of abstraction or composition. Varying numbers of layers and layer sizes can be used to provide different amounts of abstraction [3]. Deep learning exploits this idea of hierarchical explanatory factors where higher level, more abstract concepts are learned from the lower level ones. These architectures are often constructed with a greedy layer-by-layer method. Deep learning helps to disentangle these abstractions and pick out which features are useful for learning [3].
For supervised learning tasks, deep learning methods obviate feature engineering, by translating the data into compact intermediate representations akin to principal components, and derive layered structures which remove redundancy in representation [1]. Many deep learning algorithms are applied to unsupervised learning tasks. This is an important benefit because unlabeled data is usually more abundant than labeled data. An example of a deep structure that can be trained in an unsupervised manner is a deep belief network [3]. Deep neural networks are generally interpreted in terms of: Universal approximation theorem [20-24] or Probabilistic inference [1-4, 13, 25].

The Universal approximation theorem concerns the capacity of feedforward neural networks with a single hidden layer of finite size to approximate continuous functions [20-24]. In 1989, the first proof was published by Cybenko [21] for sigmoid activation functions and was generalized to feed-forward multi-layer architectures in 1991 by Hornik [22]. The approximation could be represented mathematically as follows:
Let $\varphi(.)$ be a non-constant, bounded, and monotonically-increasing continuous function. Let Im denote the m-dimensional unit hypercube [0,1]. The space of continuous functions on Im is denoted by C(Im) Then, given any function $f \in C(I_m)\ and\ \varepsilon > 0$, there exists an integer N and real constants $v_i, b_i \in R,$, where i=1,…, N such that we may define:

$$F(x) = \sum_{i=1}^{N} v_i\ \varphi\left(W_i^T x + b_i\right) \tag{1}$$

is an approximate realization of the function f where f is independent of φ; that is,

$F(x) - f(x) < \varepsilon$
for all $x \in I_m$. In other words, functions of the form $F(x)$ are dense in $C(I_m)$.

Here N represent the number of units or sample, W is the weight symmetric matrix interaction term, b is a bias term and T is the target or transpositional vector that represents the column matrix, when I_m is replaced with any compact subset of R^m. The Probabilistic Interpretation [25], which led to the introduction of dropout as regularizer in neural networks[26] is derives from the field of machine learning. It features inference, as well as the optimization concepts of training and testing, related to fitting and generalization respectively. More specifically, the Probabilistic Interpretation considers the activation nonlinearity as a cumulative distribution function [25] as indicated in equation (2).

$$P(H|E) = \frac{P(H|E).P(H)}{P(E)} \tag{2}$$

The probabilistic interpretation was introduced and popularized by luminaries such as Geoff Hinton, Yoshua Bengio, Yann LeCun and Juergen Schmidhuber [27].

In 2006, three separate groups developed ways of overcoming the difficulties that many in the machine learning world encountered while trying to train deep neural networks. The leaders of these three groups are the fathers of the age of deep learning. This is not at all hyperbole; these persons ushered in a new epoch. Their work breathed new life into neural networks when many had given up on their utilities [28]. A few years after, Geoff Hinton is offered a job by Google; Yann LeCun becomes director of AI Research at Facebook, while Yoshua Bengio takes up a position as

research chair for Artificial Intelligence at University of Montreal, funded in part by the video game company Ubisoft [29]. Their trajectories show that their work is serious business. Before their work, the earliest layers in a deep network simply weren't learning useful representations of the data. In many cases they weren't learning anything at all. Instead they were staying close to their random initialization because of the nature of the training algorithm for neural networks. Using different techniques, each of these three groups was able to get these early layers to learn useful representations, which led to much more powerful neural networks.

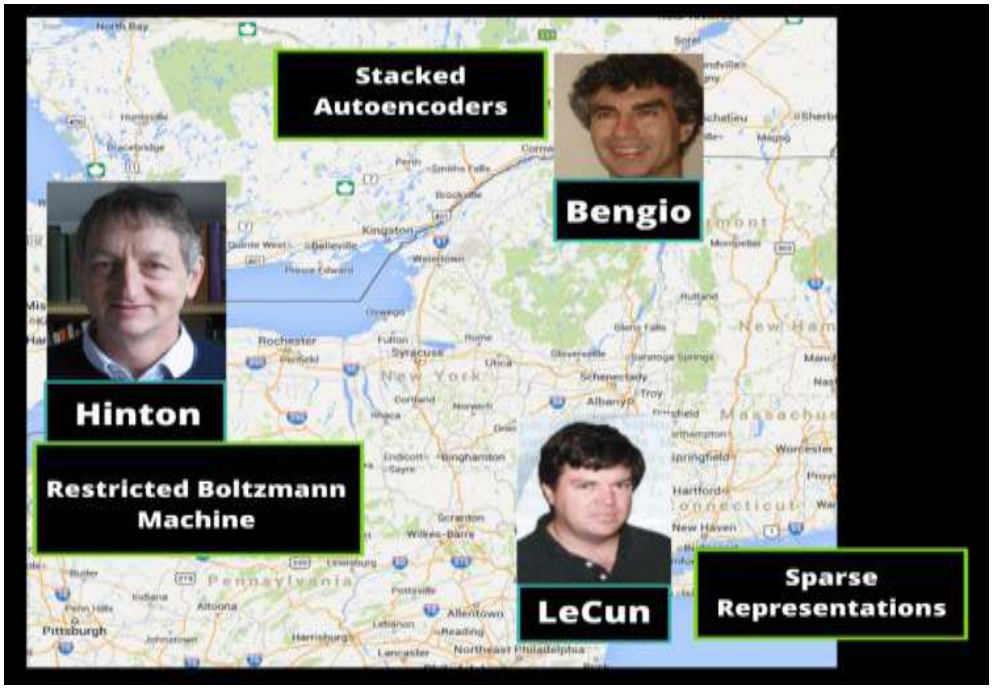

Figure 3 Fathers of Deep Learning [17]

3. Architectures

There are huge numbers of variants of deep architectures. Most of them are branches of some original parent architecture. It is not always possible to compare the performance of multiple architectures all together, because they are not all evaluated on the bases of same data sets. Deep learning is a fast-growing field, and new architectures, variants, or algorithms appear every few weeks.

Deep Neural networks

A Deep Neural Network (DNN) is an artificial neural network (ANN) with multiple hidden layers of units between the input and output layers [1, 4]. Similar to shallow ANNs, DNNs can model complex non-linear relationships. DNN architectures, examples, in object detection and parsing, generate compositional models where the object is expressed as a layered composition of image primitives [30]. The extra layers enable composition of features from lower layers, thus giving the potential for modeling complex data with fewer units than a similarly performing shallow network [2]. DNNs are typically designed as feedforward networks, but recent research has successfully applied the deep learning architecture to recurrent neural networks for applications such as language modeling[31]. Convolutional Deep Neural Networks (CNNs) are used in computer vision where their success is well documented [32]. More recently, CNNs have been applied to acoustic modeling for Automatic Speech Recognition (ASR), where they have shown success over previous models [33].

A DNN can be discriminatively trained with the standard back-propagation algorithm. The weight updates can be done via stochastic gradient descent using the equation (3):

$$w_{ij}\,(t+1) = w_{ij}\,(t) + \mu \frac{\partial C}{\partial w_{ij}} \tag{3}$$

Here, μ is the learning rate, and C is the cost function. The choice of the cost function depends on factors such as the learning type (supervised, unsupervised, reinforcement, etc.) and the activation function. For example, when performing supervised learning on a multiclass classification problem, common choices for the activation function and cost function are the softmax function and cross entropy function, respectively. The softmax function is defined as

$$P_j = \frac{exp\ (x_j)}{\sum_k \exp(x_k)} \tag{4}$$

where P_j represents the class probability (output of the unit j) and x_j and x_k represent the total input to units j and k respectively of the same level. Cross entropy (cost fuction in supervised leaerning on multiclass classification problems) is defined as

$$C_r = \sum_j d_j \ \log(P_j) \tag{5}$$

where d_j represents the target probability for output unit j and P_j is the probability output for j after applying the activation function[34].

These can be used to output object bounding boxes in form of a binary mask. They are also used for multi-scale regression to increase localization precision. DNN-based regression can learn features that capture geometric information in addition to being a good classifier. They remove the limitation of designing a model which will capture parts and their relations explicitly. This helps to learn a wide variety of objects. The model consists of multiple layers, each of which has a rectified linear unit for non-linear transformation. Some layers are convolutional, while others are fully connected. Every convolutional layer has an additional max pooling. The network is trained to minimize L2 error for predicting the mask ranging over the entire training set containing bounding boxes represented as masks [30].

Problems with Deep Neural Networks

As with ANNs, many issues can arise with DNNs if they are naively trained. Two common issues are overfitting and computation time. DNNs are prone to overfitting because of the added layers of abstraction, which allow them to model rare dependencies in the training data. Regularization methods such as weight decay (l_2-regularization) or sparsity (l_1-regularization) can be applied during training to help combat overfitting [35]. A more recent regularization method applied to DNNs is dropout regularization. In dropout, some numbers of units are randomly omitted from the hidden layers during training. This helps to break the rare dependencies that can occur in the training data [36].

The dominant method for training these structures has been error-correction training (such as back-propagation with gradient descent) due to its ease of implementation and its tendency to converge to better local optima than other training methods. However, these methods can be computationally expensive, especially for DNNs. There are many training parameters to be considered with a DNN, such as the size (number of layers and number of units per layer), the learning rate and initial weights. Sweeping through the parameter space for optimal parameters may not be feasible due to the cost in time and computational resources. Various 'tricks' such as using mini-batching (computing the gradient on several training examples at once rather than individual examples) [37] have been shown to speed up computation. The large processing throughput of Graphics Processing Units (GPUs) has produced significant speedups in training, due to the matrix and vector computations required being well suited for GPUs [4]. Radical alternatives to back-propagation such as Extreme Learning Machines [38], "No-prop" networks [39], training recurrent networks without backtracking [40], and Weightlessneural networks [41] are gaining attention.

Deep Belief Networks

A deep belief network (DBN) is a probabilistic, generative model made up of multiple layers of hidden units. It can be considered a composition of simple learning modules that make up

each layer [42]. A DBN can be used to generatively pre-train a DNN by using the learned DBN weights as the initial DNN weights. Back-propagation or other discriminative algorithms can then be applied for fine-tuning of these weights. This is particularly helpful when limited training data is available, because poorly initialized weights can significantly hinder the learned model's performance. These pre-trained weights are in a region of the weight space that is closer to the optimal weights than are randomly chosen initial weights. This allows for both improved modeling and faster convergence of the fine-tuning phase [43].

A DBN can be efficiently trained in an unsupervised, layer-by-layer manner, where the layers are typically made of Restricted Boltzmann machines (RBM). A RBM is an undirected, generative energy-based model with a "visible" input layer and a hidden layer, with connections between the layers but not within the layers. The training method for RBMs proposed by Geoffrey Hinton for use with training "Product of Expert" models is called Contrastive Divergence (CD) [44]. CD provides an approximation to the maximum likelihood method that would ideally be applied for learning the weights of the RBM [37, 45]. In training a single RBM, weight updates are performed with gradient ascent via the following equation:

$$\Delta w_{ij}\,(t+1) = w_{ij}\,(t) + \mu \frac{\partial \log(P(v))}{\partial w_{ij}} \tag{6}$$

.

Here, $P(v)$ is the probability of a visible vector, which is given by

$$\mathrm{P(v)} = \frac{1}{Z}\sum_h e^{-E(v,h)} \tag{7}$$

.

is the partition function [46], which describes the expected number of occupied microstates of a system under a giving thermodynamic condition (used for normalizing constants) and $E(v,\ h)$ is the energy function assigned to the state of the network. A lower energy indicates the network is in a more "desirable" configuration. The gradient

$\frac{\partial \log(P(v))}{\partial w_{ij}}$ has the simple form $\langle v_i\ ,\ h_j\rangle data - \langle v_i\ ,\ h_j\rangle\ model$ where $\langle \ldots . \rangle$ represent averages with respect to distribution p. The issue arises in sampling the $\langle v_i\ ,\ h_j\rangle\ model$ because this requires running alternating Gibbs sampling for a long time. CD replaces this step by running alternating Gibbs sampling for n steps (values of *n=1* have empirically been shown to perform well). After n steps, the data is sampled and that sample is used in place of $\langle v_i\ ,\ h_j\rangle\ model$. The CD procedure works as follows [37]: Initialize the visible units to a training vector. Update the hidden units in parallel given the visible units:

$$\mathrm{P}\left(h_j = 1 \middle| \mathrm{V}\right) = \sigma(b_j + \textstyle\sum_i v_i w_{ij}) \tag{8}$$

.

Here σ is the sigmoid function, V is the set of visible units and b_j is the bias of h_j. The update of the visible units in parallel given the hidden units is given as:

$$\mathrm{P}(v_i = 1|\mathrm{H}) = \sigma(a_i + \textstyle\sum_j h_j w_{ij}) \tag{9}$$

where a_i is the bias of v_i and H is the set of hidden units. This is called the "reconstruction" step.

Re-update the hidden units in parallel given the reconstructed visible units using the same equation as in step 2.

Perform the weight update:

$$\Delta w_{ij} \propto \langle v_i h_j\rangle_{data} - \langle v_i h_j\rangle_{reconstruction}$$

Once an RBM is trained, another RBM is "stacked" atop it, taking its input from the final already-trained layer. The new visible layer is initialized to a training vector, and values for the units in the already-trained layers are assigned using the current weights and biases. The new RBM is then trained with the procedure above. This whole process is repeated until some desired stopping criterion is met [2]. Although the approximation of CD to maximum likelihood is very crude (CD has been shown to not follow the gradient of any function), it has been empirically shown to be effective in training deep architectures [37].

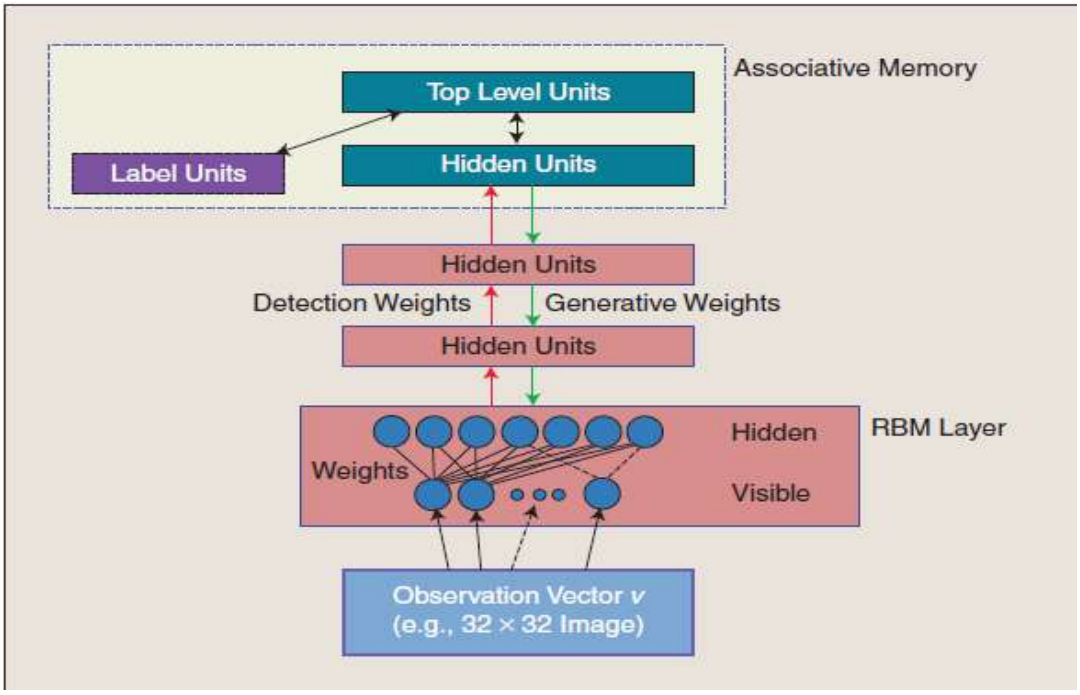

Figure 4 Deep Belief Network [6]

Convolutional Neural Networks (CNN)

A CNN is composed of one or more convolutional layers with fully connected layers (matching those in typical artificial neural networks) on top. It also uses tied weights and pooling layers. This architecture allows CNNs to take advantage of the 2D structure of input data. In comparison with other deep architectures, convolutional neural networks are starting to show superior results in both image and speech applications [47]. They can also be trained with standard back-propagation. CNNs are easier to train than other regular, deep, feed-forward neural networks and have many fewer parameters to estimate, making them a highly attractive architecture to use [48]. Examples of applications in Computer Vision include DeepDream [49].

Convolutional Deep Belief Networks

A recent achievement in deep learning is the use of Convolutional Deep Belief Networks (CDBN). CDBNs have structure very similar to a convolutional neural networks and are trained similar to deep belief networks. Therefore, they exploit the 2D structure of images, like CNNs do, and make use of pre-training like deep belief networks. They provide a generic structure which can be used in many image and signal processing tasks. Recently, many benchmark results on standard image datasets like CIFAR [50] have been obtained using CDBNs [51].

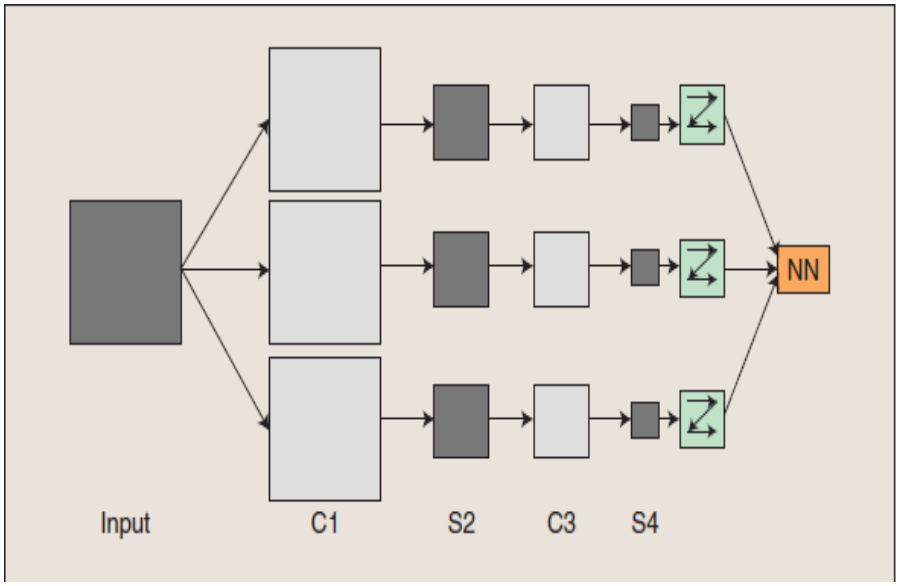

Figure 5: Convolutional Neural network [6]

4. Deep Learning Applications

There have been several studies demonstrating the effectiveness of deep learning methods in a variety of application domains. In addition to the Mixed National Institute of Standards and Technology (MNIST) handwriting challenge [52], there are applications in face detection [53, 54], speech recognition and detection [55], general object recognition [56], natural language processing [57], and robotics.

The reality of data proliferation and abundance of multimodal sensory information is admittedly a challenge and a recurring theme in many military as well as civilian applications, such as sophisticated surveillance systems. Consequently, interest in deep machine learning has not been limited to academic research. Recently, the Defense Advanced Research Projects Agency (DARPA) announced a research program exclusively focused on deep learning [6, 58]. Several private organizations have focused their attention on commercializing deep learning technologies with applications to broad domains.

Lenz et al [59] recently presented a system for detecting robotic grasps from RGB-D data using a deep learning approach which has several advantages over current state-of-the-art methods. Their approach firstly proved that using deep learning allows you to avoid using hand-engineering features, but learning them instead. Secondly, their results showed that deep learning methods significantly outperformed even well designed hand-engineered features from previous work. Hence deep learning system with group regularization is capable of robustly detecting grasps for a wide range of objects,

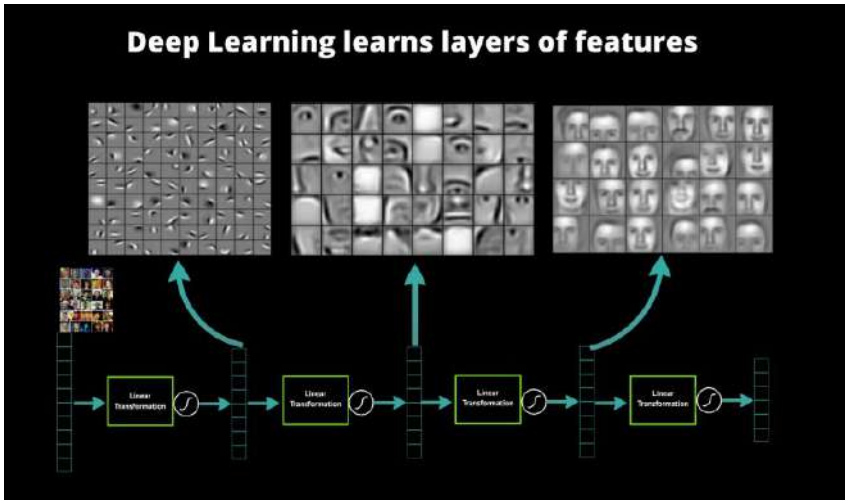

Figure 6 Deep Learning Architecture [17]

5. Discussions and Conclusions

Given the extensive strides of artificial intelligence in recent times, coupled with the recognition that deep learning is evolving as one of its most powerful techniques, the subject is explicably attracting both criticism and comment, and in some cases from outside the field of computer science itself. Though this paper has tried to present a comprehensive review on prior work conducted in deep learning, there still remains a great deal of work to be done in improving the learning process. For example where current focus is on lending fertile ideas from other areas of machine learning, such as context of dimensionality reduction, there is still much work needed to be done [6]. One example includes recent work on sparse coding [60] where the inherent high dimensionality of data is reduced through the use of compressed sensing theory, allowing accurate representation of signals with very small numbers of basis vectors [61]. Another example is semi-supervised manifold learning [62] where the dimensionality of data is reduced by measuring the similarity between training data samples, and then projecting these similarity measurements to lower-dimensional spaces. In addition, further inspiration and techniques may be found from evolutionary programming approaches [63, 64] where conceptually adaptive learning and core architectural changes can be learned with minimal engineering efforts. While deep learning has been successfully applied to challenging pattern inference tasks, the goal of the field is far beyond task-specific applications. This scope may make the comparison of various methodologies increasingly complex and will likely necessitate a collaborative effort by the research community to address. It should also be noted that, despite the great prospect offered by deep learning technologies, some domain-specific tasks may not be directly improved by such schemes. An example is identifying and reading the routing numbers at the bottom of bank checks.Though these digits are human readable, they are comprised of restricted character sets which specialized readers can recognize flawlessly at very high data rates [65]. Similarly, iris recognition is not a task that humans generally perform; indeed, without training, one iris looks very similar to another to the untrained eye, yet engineered systems can produce matches between candidate iris images and an image database with high precision and accuracy to serve as a unique identifier [66]. Finally, recent developments in facial recognition [54] show equivalent performance relative to humans in their ability to match query images against large numbers of candidates, potentially matching far more than most humans can recall [67]. Nevertheless, these remain highly specific cases and are the result of lengthy feature engineering optimization processes (as well as years of research) that do not map to other, more general applications. Furthermore, deep learning platforms can also benefit from engineered features while learning more complex representations which engineered systems typically lack. Despite the myriad of open research issues and the fact that the field is still in its infancy, it is abundantly clear that advancements made with respect to developing deep machine learning systems will undoubtedly shape the future of machine learning.

References

[1] D. Li and D. Yu, "Deep Learning: Methods and Applications," *Foundations and Trends in Signal Processing, Now Publishers,* 2014.

[2] Y. Bengio, "Learning deep architectures for AI," *Foundations and trends® in Machine Learning,* vol. 2, pp. 1-127, 2009.

[3] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* vol. 35, pp. 1798-1828, 2013.

[4] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Networks,* vol. 61, pp. 85-117, 2015.

[5] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature,* vol. 521, pp. 436-444, 2015.

[6] I. Arel, D. C. Rose, and T. P. Karnowski, "Deep machine learning-a new frontier in artificial intelligence research [research frontier]," *Computational Intelligence Magazine, IEEE,* vol. 5, pp. 13-18, 2010.

[7] P. O. Glauner, "Deep Convolutional Neural Networks for Smile Recognition," *arXiv preprint arXiv:1508.06535,* 2015.

[8] H. A. Song and S.-Y. Lee, "Hierarchical Representation Using NMF," in *Neural Information Processing*, 2013, pp. 466-473.

[9] M. Xiong, J. Chen, Z. Wang, C. Liang, Q. Zheng, Z. Han*, et al.*, "Deep Feature Representation via Multiple Stack Auto-Encoders," in *Advances in Multimedia Information Processing--PCM 2015*, ed: Springer, 2015, pp. 275-284.

[10] B. A. Olshausen, "Emergence of simple-cell receptive field properties by learning a sparse code for natural images," *Nature,* vol. 381, pp. 607-609, 1996.

[11] R. Collobert, "Deep learning for efficient discriminative parsing," in *International Conference on Artificial Intelligence and Statistics*, 2011.

[12] L. Gomes, "Machine-learning maestro michael jordan on the delusions of big data and other huge engineering efforts," *IEEE Spectrum, Oct,* vol. 20, 2014.

[13] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural computation,* vol. 18, pp. 1527-1554, 2006.

[14] R. Salakhutdinov and G. Hinton, "An efficient learning procedure for deep Boltzmann machines," *Neural computation,* vol. 24, pp. 1967-2006, 2012.

[15] D. H. Staelin and C. H. Staelin, "Models for Neural Spike Computation and Cognition," ed: CreateSpace.[Links], 2011.

[16] N. Kriegeskorte, "Deep Neural Networks: A New Framework for Modeling Biological Vision and Brain Information Processing," *Annual Review of Vision Science,* vol. 1, pp. 417-446, 2015.

[17] S. Bengio, L. Deng, H. Larochelle, H. Lee, and R. Salakhutdinov, "Guest Editors' Introduction: Special Section on Learning Deep Architectures," *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* vol. 35, pp. 1795-1797, 2013.

[18] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097-1105.

[19] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: an efficient alternative to SIFT or SURF," in *Computer Vision (ICCV), 2011 IEEE International Conference on*, 2011, pp. 2564-2571.

[20] B. C. Csáji, "Approximation with artificial neural networks," *Faculty of Sciences, Etvs Lornd University, Hungary,* vol. 24, 2001.

[21] G. Cybenko, "Approximation by superpositions of a sigmoidal function," *Mathematics of control, signals and systems,* vol. 2, pp. 303-314, 1989.

[22] K. Hornik, "Approximation capabilities of multilayer feedforward networks," *Neural networks,* vol. 4, pp. 251-257, 1991.

[23] S. Haykin and R. Lippmann, "Neural Networks, A Comprehensive Foundation," *International Journal of Neural Systems,* vol. 5, pp. 363-364, 1994.
[24] M. H. Hassoun, *Fundamentals of artificial neural networks*: MIT press, 1995.
[25] K. P. Murphy, *Machine learning: a probabilistic perspective*: MIT press, 2012.
[26] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," *arXiv preprint arXiv:1207.0580,* 2012.
[27] Y. Bengio, "Deep learning of representations: Looking forward," in *Statistical language and speech processing*, ed: Springer, 2013, pp. 1-37.
[28] A. Testolin, I. Stoianov, M. De Filippo De Grazia, and M. Zorzi, "Deep unsupervised learning on a desktop PC: a primer for cognitive scientists," *Front. Psychol,* vol. 4, p. 10.3389, 2013.
[29] A. Weapons, "an Open Letter from AI & Robotics Researchers, 2015," ed.
[30] C. Szegedy, A. Toshev, and D. Erhan, "Deep neural networks for object detection," in *Advances in Neural Information Processing Systems*, 2013, pp. 2553-2561.
[31] S. Kombrink, T. Mikolov, M. Karafiát, and L. Burget, "Recurrent Neural Network Based Language Modeling in Meeting Recognition," in *INTERSPEECH*, 2011, pp. 2877-2880.
[32] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE,* vol. 86, pp. 2278-2324, 1998.
[33] T. N. Sainath, A.-r. Mohamed, B. Kingsbury, and B. Ramabhadran, "Deep convolutional neural networks for LVCSR," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, 2013, pp. 8614-8618.
[34] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly*, et al.*, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *Signal Processing Magazine, IEEE,* vol. 29, pp. 82-97, 2012.
[35] Y. Bengio, N. Boulanger-Lewandowski, and R. Pascanu, "Advances in optimizing recurrent networks," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, 2013, pp. 8624-8628.
[36] G. E. Dahl, T. N. Sainath, and G. E. Hinton, "Improving deep neural networks for LVCSR using rectified linear units and dropout," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, 2013, pp. 8609-8613.
[37] G. Hinton, "A practical guide to training restricted Boltzmann machines," *Momentum,* vol. 9, p. 926, 2010.
[38] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: theory and applications," *Neurocomputing,* vol. 70, pp. 489-501, 2006.
[39] B. Widrow, A. Greenblatt, Y. Kim, and D. Park, "The no-prop algorithm: A new learning algorithm for multilayer neural networks," *Neural Networks,* vol. 37, pp. 182-188, 2013.
[40] Y. Ollivier and G. Charpiat, "Training recurrent networks online without backtracking," *arXiv preprint arXiv:1507.07680,* 2015.
[41] I. Aleksander, M. De Gregorio, F. M. G. França, P. M. V. Lima, and H. Morton, "A brief introduction to Weightless Neural Systems," in *ESANN*, 2009.
[42] G. E. Hinton, "Deep belief networks," *Scholarpedia,* vol. 4, p. 5947, 2009.
[43] H. Larochelle, D. Erhan, A. Courville, J. Bergstra, and Y. Bengio, "An empirical evaluation of deep architectures on problems with many factors of variation," in *Proceedings of the 24th international conference on Machine learning*, 2007, pp. 473-480.
[44] G. E. Hinton, "Training products of experts by minimizing contrastive divergence," *Neural computation,* vol. 14, pp. 1771-1800, 2002.
[45] A. Fischer and C. Igel, "Training restricted Boltzmann machines: an introduction," *Pattern Recognition,* vol. 47, pp. 25-39, 2014.
[46] G. Barnich, H. A. Gonzalez, A. Maloney, and B. Oblak, "One loop partition function of three-dimensional flat gravity," *Journal of High Energy Physics,* vol. 2015, pp. 1-8, 2015.
[47] T. Sercu, C. Puhrsch, B. Kingsbury, and Y. LeCun, "Very deep multilingual convolutional neural networks for LVCSR," *arXiv preprint arXiv:1509.08967,* 2015.

[48] C. Yan, F. Coenen, and B. Zhang, "Driving posture recognition by convolutional neural networks," *IET Computer Vision,* 2015.
[49] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov*, et al.*, "Going deeper with convolutions," *arXiv preprint arXiv:1409.4842,* 2014.
[50] A. Krizhevsky and G. Hinton, "Convolutional deep belief networks on cifar-10," *Unpublished manuscript,* 2010.
[51] H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng, "Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations," in *Proceedings of the 26th Annual International Conference on Machine Learning*, 2009, pp. 609-616.
[52] Y. LeCun, "The MNIST database of handwritten digits.{Online}," ed, 2007.
[53] B. Kwolek, "Face detection using convolutional neural networks and Gabor filters," in *Artificial Neural Networks: Biological Inspirations–ICANN 2005*, ed: Springer, 2005, pp. 551-556.
[54] M. Osadchy, Y. L. Cun, and M. L. Miller, "Synergistic face detection and pose estimation with energy-based models," *The Journal of Machine Learning Research,* vol. 8, pp. 1197-1215, 2007.
[55] S. Sukittanon, A. C. Surendran, J. C. Platt, and C. J. Burges, "Convolutional networks for speech detection," in *Interspeech*, 2004.
[56] F. J. Huang and Y. LeCun, "Large-scale learning with svm and convolutional for generic object categorization," in *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, 2006, pp. 284-291.
[57] H. Lee, P. Pham, Y. Largman, and A. Y. Ng, "Unsupervised feature learning for audio classification using convolutional deep belief networks," in *Advances in neural information processing systems*, 2009, pp. 1096-1104.
[58] T. Khorshed, "Research Problem Definition Part."
[59] I. Lenz, H. Lee, and A. Saxena, "Deep learning for detecting robotic grasps," *The International Journal of Robotics Research,* vol. 34, pp. 705-724, 2015.
[60] K. Kavukcuoglu, M. A. Ranzato, R. Fergus, and Y. Le-Cun, "Learning invariant features through topographic filter maps," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, 2009, pp. 1605-1612.
[61] B. B. Benuwa, B. Ghansah, D. K. Wornyo, and S. A. Adabunu, "A Comprehensive Review of Particle Swarm Optimization," in *International Journal of Engineering Research in Africa*, 2016, pp. 141-161.
[62] J. Weston, F. Ratle, H. Mobahi, and R. Collobert, "Deep learning via semi-supervised embedding," in *Neural Networks: Tricks of the Trade*, ed: Springer, 2012, pp. 639-655.
[63] K. A. De Jong, "Evolving intelligent agents: A 50 year quest," *IEEE Computational Intelligence Magazine,* vol. 1, pp. 12-17, 2008.
[64] M. M. Islam and X. Yao, "Evolving artificial neural network ensembles," in *Computational intelligence: a compendium*, ed: Springer, 2008, pp. 851-880.
[65] S. V. Rice, F. R. Jenkins, and T. A. Nartker, *The fifth annual test of OCR accuracy*: Information Science Research Institute, 1996.
[66] E. M. Newton and P. J. Phillips, "Meta-analysis of third-party evaluations of iris recognition," *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on,* vol. 39, pp. 4-11, 2009.
[67] A. Adler and M. E. Schuckers, "Comparing human and automatic face recognition performance," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on,* vol. 37, pp. 1248-1255, 2007.

International Journal of Engineering Research in Africa Vol. 24
10.4028/www.scientific.net/JERA.24

A Review of Deep Machine Learning
10.4028/www.scientific.net/JERA.24.124

DOI References
[1] D. Li and D. Yu, Deep Learning: Methods and Applications, Foundations and Trends in Signal Processing, Now Publishers, (2014).
10.1561/2000000039
[2] Y. Bengio, Learning deep architectures for AI, Foundations and trends® in Machine Learning, vol. 2, pp.1-127, (2009).
10.1561/2200000006
[3] Y. Bengio, A. Courville, and P. Vincent, Representation learning: A review and new perspectives, Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 35, pp.1798-1828, (2013).
10.1109/tpami.2013.50
[4] J. Schmidhuber, Deep learning in neural networks: An overview, Neural Networks, vol. 61, pp.85-117, (2015).
10.1016/j.neunet.2014.09.003
[5] Y. LeCun, Y. Bengio, and G. Hinton, Deep learning, Nature, vol. 521, pp.436-444, (2015).
10.1038/nature14539
[6] I. Arel, D. C. Rose, and T. P. Karnowski, Deep machine learning-a new frontier in artificial intelligence research [research frontier], Computational Intelligence Magazine, IEEE, vol. 5, pp.13-18, (2010).
10.1109/mci.2010.938364
[8] H. A. Song and S. -Y. Lee, Hierarchical Representation Using NMF, in Neural Information Processing, 2013, pp.466-473.
10.1007/978-3-642-42054-2_58
[9] M. Xiong, J. Chen, Z. Wang, C. Liang, Q. Zheng, Z. Han, et al., Deep Feature Representation via Multiple Stack Auto-Encoders, in Advances in Multimedia Information Processing-PCM 2015, ed: Springer, 2015, pp.275-284.
10.1007/978-3-319-24075-6_27
[10] B. A. Olshausen, Emergence of simple-cell receptive field properties by learning a sparse code for natural images, Nature, vol. 381, pp.607-609, (1996).
10.1038/381607a0
[13] G. E. Hinton, S. Osindero, and Y. -W. Teh, A fast learning algorithm for deep belief nets, Neural computation, vol. 18, pp.1527-1554, (2006).
10.1162/neco.2006.18.7.1527
[14] R. Salakhutdinov and G. Hinton, An efficient learning procedure for deep Boltzmann machines, Neural computation, vol. 24, pp.1967-2006, (2012).
10.1162/neco_a_00311
[16] N. Kriegeskorte, Deep Neural Networks: A New Framework for Modeling Biological Vision and Brain Information Processing, Annual Review of Vision Science, vol. 1, pp.417-446, (2015).
10.1146/annurev-vision-082114-035447
[17] S. Bengio, L. Deng, H. Larochelle, H. Lee, and R. Salakhutdinov, Guest Editors' Introduction: Special Section on Learning Deep Architectures, Pattern Analysis and Machine Intelligence, IEEE Transactions on,

vol. 35, pp.1795-1797, (2013).
10.1109/tpami.2013.118
[19] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, ORB: an efficient alternative to SIFT or SURF, in Computer Vision (ICCV), 2011 IEEE International Conference on, 2011, p.25642571.
10.1109/iccv.2011.6126544
[21] G. Cybenko, Approximation by superpositions of a sigmoidal function, Mathematics of control, signals and systems, vol. 2, pp.303-314, (1989).
10.1007/bf02551274
[22] K. Hornik, Approximation capabilities of multilayer feedforward networks, Neural networks, vol. 4, pp.251-257, (1991).
10.1016/0893-6080(91)90009-t
[23] S. Haykin and R. Lippmann, Neural Networks, A Comprehensive Foundation, International Journal of Neural Systems, vol. 5, pp.363-364, (1994).
10.1142/s0129065794000372
[24] M. H. Hassoun, Fundamentals of artificial neural networks: MIT press, (1995).
10.1145/272874.1067696
[27] Y. Bengio, Deep learning of representations: Looking forward, in Statistical language and speech processing, ed: Springer, 2013, pp.1-37.
10.1007/978-3-642-39593-2_1
[28] A. Testolin, I. Stoianov, M. De Filippo De Grazia, and M. Zorzi, Deep unsupervised learning on a desktop PC: a primer for cognitive scientists, Front. Psychol, vol. 4, p.10. 3389, (2013).
10.3389/fpsyg.2013.00251
[30] C. Szegedy, A. Toshev, and D. Erhan, Deep neural networks for object detection, in Advances in Neural Information Processing Systems, 2013, pp.2553-2561.
10.1109/cvpr.2014.276
[32] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, Gradient-based learning applied to document recognition, Proceedings of the IEEE, vol. 86, pp.2278-2324, (1998).
10.1109/5.726791
[33] T. N. Sainath, A. -r. Mohamed, B. Kingsbury, and B. Ramabhadran, Deep convolutional neural networks for LVCSR, in Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on, 2013, pp.8614-8618.
10.1109/icassp.2013.6639347
[34] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A. -r. Mohamed, N. Jaitly, et al., Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups, Signal Processing Magazine, IEEE, vol. 29, pp.82-97, (2012).
10.1109/msp.2012.2205597
[35] Y. Bengio, N. Boulanger-Lewandowski, and R. Pascanu, Advances in optimizing recurrent networks, in Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on, 2013, pp.8624-8628.
10.1109/icassp.2013.6639349
[36] G. E. Dahl, T. N. Sainath, and G. E. Hinton, Improving deep neural networks for LVCSR using rectified linear units and dropout, in Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on, 2013, pp.8609-8613.
10.1109/icassp.2013.6639346
[37] G. Hinton, A practical guide to training restricted Boltzmann machines, Momentum, vol. 9, p.926, (2010).

10.1007/978-3-642-35289-8_32
[38] G. -B. Huang, Q. -Y. Zhu, and C. -K. Siew, Extreme learning machine: theory and applications, Neurocomputing, vol. 70, pp.489-501, (2006).
10.1016/j.neucom.2005.12.126
[39] B. Widrow, A. Greenblatt, Y. Kim, and D. Park, The no-prop algorithm: A new learning algorithm for multilayer neural networks, Neural Networks, vol. 37, pp.182-188, (2013).
10.1016/j.neunet.2012.09.020
[42] G. E. Hinton, Deep belief networks, Scholarpedia, vol. 4, p.5947, (2009).
10.4249/scholarpedia.5947
[43] H. Larochelle, D. Erhan, A. Courville, J. Bergstra, and Y. Bengio, An empirical evaluation of deep architectures on problems with many factors of variation, in Proceedings of the 24th international conference on Machine learning, 2007, pp.473-480.
10.1145/1273496.1273556
[44] G. E. Hinton, Training products of experts by minimizing contrastive divergence, Neural computation, vol. 14, pp.1771-1800, (2002).
10.1162/089976602760128018
[45] A. Fischer and C. Igel, Training restricted Boltzmann machines: an introduction, Pattern Recognition, vol. 47, pp.25-39, (2014).
10.1016/j.patcog.2013.05.025
[46] G. Barnich, H. A. Gonzalez, A. Maloney, and B. Oblak, One loop partition function of threedimensional flat gravity, Journal of High Energy Physics, vol. 2015, pp.1-8, (2015).
10.1007/jhep04(2015)178
[48] C. Yan, F. Coenen, and B. Zhang, Driving posture recognition by convolutional neural networks, IET Computer Vision, (2015).
10.1049/iet-cvi.2015.0175
[51] H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng, Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations, in Proceedings of the 26th Annual International Conference on Machine Learning, 2009, pp.609-616.
10.1145/1553374.1553453
[53] B. Kwolek, Face detection using convolutional neural networks and Gabor filters, in Artificial Neural Networks: Biological Inspirations-ICANN 2005, ed: Springer, 2005, pp.551-556.
10.1007/11550822_86
[54] M. Osadchy, Y. L. Cun, and M. L. Miller, Synergistic face detection and pose estimation with energy-based models, The Journal of Machine Learning Research, vol. 8, pp.1197-1215, (2007).
10.1007/11957959_10
[56] F. J. Huang and Y. LeCun, Large-scale learning with svm and convolutional for generic object categorization, in Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on, 2006, pp.284-291.
10.1109/cvpr.2006.164
[59] I. Lenz, H. Lee, and A. Saxena, Deep learning for detecting robotic grasps, The International Journal of Robotics Research, vol. 34, pp.705-724, (2015).
10.1177/0278364914549607
[60] K. Kavukcuoglu, M. A. Ranzato, R. Fergus, and Y. Le-Cun, Learning invariant features through topographic filter maps, in Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on, 2009, pp.1605-1612.
10.1109/cvpr.2009.5206545

[61] B. B. Benuwa, B. Ghansah, D. K. Wornyo, and S. A. Adabunu, A Comprehensive Review of Particle Swarm Optimization, in International Journal of Engineering Research in Africa, 2016, pp.141-161. 10.4028/www.scientific.net/jera.23.141
[62] J. Weston, F. Ratle, H. Mobahi, and R. Collobert, Deep learning via semi-supervised embedding, in Neural Networks: Tricks of the Trade, ed: Springer, 2012, pp.639-655. 10.1007/978-3-642-35289-8_34
[63] K. A. De Jong, Evolving intelligent agents: A 50 year quest, IEEE Computational Intelligence Magazine, vol. 1, pp.12-17, (2008). 10.1109/mci.2007.913370
[64] M. M. Islam and X. Yao, Evolving artificial neural network ensembles, in Computational intelligence: a compendium, ed: Springer, 2008, pp.851-880. 10.1007/978-3-540-78293-3_20
[66] E. M. Newton and P. J. Phillips, Meta-analysis of third-party evaluations of iris recognition, Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on, vol. 39, pp.4-11, (2009). 10.1109/tsmca.2008.2008210
[67] A. Adler and M. E. Schuckers, Comparing human and automatic face recognition performance, Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on, vol. 37, pp.1248-1255, (2007). 10.1109/tsmcb.2007.907036