Representation Learning

A Literature Survey

Aydın Göze POLAT
Computer Engineering Department
Middle East Technical University (METU)
Ankara, TURKEY
goze.polat@metu.edu.tr

Abstract— In this study, an analysis of the literature on representation learning and some of the precursory research studies on relatively unexplored areas in the literature are given. It appears that representation learning techniques as the driving force behind many deep learning approaches are starting to acquire theoretical foundations and empirical support.

Keywords— *deep learning; unsupervised learning; representation learning; feature learning; invariants; bio-inspired models; artificial neural networks*

I. Introduction

Machine learning models try to learn generalizations or make predictions in a data-driven way. However, how to achieve these tasks in the most efficient way possible is an open question. Minsky defines the fundamental credit assignment problem as the problem of determining the modifiable components of a learning system that are responsible from its success or failure and changing them to improve performance [1]. From the perspective of machine learning, these components mainly correspond to representation, evaluation and optimization.

In the last decade, the research on *deep learning* revealed that achieving a good internal *representation* of the data within a model is a critical problem. According to Bengio et al., more than anything, success of a machine learning algorithm depends on the question whether it can create a good representation of the data and take advantage of it or not [2]. In this study, I give an overview of the research that is relevant to deep learning and representation learning.

A. Deep Learning

Shmidhuber distinguishes deep and shallow architectures according to the depth of their credit assignment paths [3]. Credit assignment path is a chain of possibly learnable causal links between actions and effects. The smallest depth for any credit assignment path corresponds to the problem's depth.

A deep architecture has multiple levels of nonlinear operations (i.e. depth > 2). Shallow architectures can be used as universal approximators, (i.e. they can approximate any function), however, in practice, for some problems deep architectures are necessary, because decreasing the depth can result in an exponential increase in the size of a model (e.g instead of O(n) nodes necessary for depth d, O(2^n) for depth d-1) [4].

Even though increasing the depth can reduce the size of a model, training a deep architecture using traditional methods does not work well. For example, the backpropagation algorithm with random initialization of weights (that is commonly used for shallow feed forward networks) has a poor performance in deep architectures, because increased depth creates sharp nonlinearities and this results in the exploding or vanishing gradient problem described in [5].

Deep learning comprises machine learning techniques that take advantage of learning multiple levels of abstraction in a hierarchical manner to efficiently train deep architectures [6].

B. Representation Learning

Feature engineering was traditionally used as a way to compensate for the lack of automated mechanisms to transform the data into a good form and increase the efficiency of machine learning algorithms. Now, however, efficiently learning good data representations without any feature engineering is possible, especially for the types of problems where an abundance of unlabeled data can be easily collected or generated [2].

Representation learning consists of techniques that allow learning transformations of the data to extract useful features more efficiently [2]. It is useful in the application areas such as vision, audio, speech, natural language processing, robotics and neuroscience for various purposes such as classification, segmentation, compression and denoising. Moreover, the application areas continue to grow,, since representation learning can be applied to any area where there is a large amount of unlabeled data.

Learning features can be achieved in an unsupervised, supervised or semi-supervised way. However, since unlabeled data is easily available, unsupervised learning is commonly used for representation learning, thus representation learning is also often referred to as “unsupervised feature learning” [2].

1) Deep representations

Representation learning is closely connected to deep learning, because learning a deep representation where abstract features emerge at higher levels of a feature hierarchy is often desired. This is due to the fact that abstract features are often invariant to low level changes and robust against noise.

2) Shallow representations

Although deep learning models have state-of-the-art performance in many areas, shallow models can be competitive too. Coates et al. give an analysis for the changes to receptive field size, number of hidden nodes (features), step size (stride) between features and the effect of whitening [7]. They demonstrate that by increasing the number of features extracted, and increasing the number of hidden nodes, it is possible to achieve state-of-the-art performance even with simple algorithms like k-means. Moreover, they believe that reducing the step size while increasing the number of features can monotonically improve the accuracy, even though increasing the receptive field size can have negative effects since a larger receptive field can require more parameters to learn and more data to train. This study agrees with the fact that shallow architectures can perform well if their size is increased.

The problem with shallow representations is that it is already theoretically well known that for some problems, the increase in the size would be exponentional compared to a deep architecture [4]. Thus, training shallow architectures are not always practical and it might be a better idea to represent the features in a more compact way. In this study, I will give an overview of the representation learning research relevant to the deep architectures.

II. Motivation

The most apparent benefit of representation learning is that instead of feature engineering, it allows extracting features directly from the data. Since there is an exponential increase in the size of unlabeled data every year [8], the task of extracting features from unlabeled samples becomes more and more critical. Extracting large numbers of useful features from high dimensional data is an important task because the higher the number of useful extracted features is, the higher accuracy a model will achieve, independent of the algorithm [7]. Moreover, Banko and Brill demonstrate that accuracy of the learners continue to improve as the training set gets larger [9].

Another big advantage of representation learning is that by transforming the input representation into a more useful form (e.g. changing the dimensionality, disentangling the latent variables [2]), it is possible to avoid the curse of dimensionality and directly work with a very high dimensional input. This means that, it is possible to directly take advantage of the big data. In fact, Banko and Brill claim that both supervised and unsupervised learners can benefit from additional training data [9]. However, they also show that the size of the learned representations can increase exponentially for Winnow algorithm and memory based algorithms[1]. Their work clearly points out that, at least for certain types of problems, instead of the learning algorithm, size of the training set is the decisive factor for improvements that are observed. Hence, to be able to directly use large amounts of data is an important advantage of representation learning.

Learning various types of representations can directly affect AI. For instance, multimodal representations [10], embeddings [11], multiview features [12] and distributed representations [13] might allow inference that is more efficient and human like. Moreover, compositionality of a deep representation (i.e. learned features are hierarchical constituents of other more abstract features) is similar to how humans intuitively represent things and therefore is also valuable for inference.

III. Literature

Representation learning ideas come from various research areas and from various points in the history of learning models[2]. Below, a

[1] Note that both types of algorithms construct shallow representations, unlike deep learners which can create hierarchical and more compact representations

[2] For instance, the main focus of this research, learning of deep representations, directly requires ideas from relevant deep learning research.

comprehensive literature survey consisting of a brief historical overview and a detailed analysis in a categorical way is given.

A. Historical Overview

Built on many ideas coming from earlier learning models, representation learning literature can be analyzed in terms of early fundamental research and relevant examples throughout the early history of learning models.

1) Early research

The artificial neuron model suggested by McCulloch and Pitts in 1943 [14] and Rosenblatt's perceptron learning algorithm in 1958 [15], sparked researchers' interest in learning models. Backpropagation algorithm in 1963 [16], Neocognitron in 1980 [17], Hopfield networks [18], principal component analysis (PCA) [19], and self organizing maps (SOM) [20] in 1982, Boltzmann machines in 1985 [21], then in 1986 the restricted Boltzmann machines (RBM) [22] and multilayer perceptrons (MLP) [23], after that in 1988 radial basis function (RBF) networks [24], autoencoders in 1989 [25], nonlinear generalization of PCA by Kramer in 1991 [26], sigmoid belief networks in 1992 [27], support vector machines (SVM) [28] in 1995 and sparse coding in 1996 [29] can be considered among some of the earlier research. Aside from being the first deep learning example, Neocognitron also introduced convolutional neural networks (CNN) in 1980 [17], and in 2003 Benkhe introduced a generalization of the CNNs [30].

a) Neocognitron

Fukushima proposed the very first deep learning architecture in the history in 1979 [31]. And a year later he introduced Neocognitron, which is designed for visual pattern recognition [17]. His model was *bioinspired* from the visual system and the hierarchy model of it suggested by Hubel and Wiesel [32]. In his study, several precursory research ideas were applied. For instance, he introduced convolutional neural networks the first time. He used unsupervised learning to train a deep architecture in which abstract features that are robust against noise emerged. Moreover, these features displayed translation invariance. He described the unsupervised training of the network to acquire translation invariance:

> The network is self-organized by "learning without a teacher", and acquires an ability to recognize stimulus patterns based on the geometrical similarity (Gestalt) of their shapes without affected by their positions [31].

b) Backpropagation applied to MLP

In 1989, LeCun et al. demonstrated a model that can recognize digits by using the backpropagation algorithm on an MLP that can directly learn from normalized digit images instead of feature engineered input [33]. They constrained the network weights and architecture by hand, according to the digit recognition domain, to achieve more efficient training.

c) An improved CNN model

A later relevant example was given by LeCun et al. in 1998 [34]. After a review and comparison of relevant research on hand written character recognition, they used their CNN model, LeNet, to demonstrate that it could achieve state-of-the-art performance, giving better results than all the other reviewed models. Unlike the traditional shallow models which used a classifier on top of a single feature extractor layer, LeNet consisted of several subsequent convolution and subsampling layers, effectively creating a deep architecture. Their research contained important conclusions about feature learning. For instance, they presented the idea that, instead of early segmentation of objects in images using features found by experts (i.e. feature extraction as a fixed transform), a heuristic algorithm (i.e. Heuristic Over-Segmentation) can be used in a data-driven way for each layer (i.e. representation learning instead of feature engineering).

2) The deep learning reboot

The idea of taking advantage of large amounts of data with unsupervised training to learn representations and to construct a deep initial representation was present in the early research. For instance, the very first example of deep learning, Neocognitron, was around for decades [31]. However, the effective beginning of deep learning research was after the suggestion of practical techniques that performed particularly well. In 2006, Hinton et al.'s greedy algorithm of layerwise pretraining described a way to achieve better and better layerwise representations of the input for Deep Belief Networks (DBN) [35]. Shortly after Hinton, Ranzato et al. came up with an energy based model that can learn sparse and overcomplete representations via unsupervised learning [36]. One year later, Bengio et al. pointed out that Hinton et al.'s strategy of layerwise transformation of the input to achieve better internal representations can be applied beyond DBNs [37]. These developments and their clear practical success in the benchmarks resulted in a high amount of interest and new research in the representation and deep learning areas.

B. Categorical Overview

I divided the literature into four main categories. After I focus on theoretical research and relevant models, I will give a brief overview of empirical studies and applications.

1) Theoretical research

a) We already do representation learning

Hinton discusses why there must be a basic learning algorithm for feature extraction that can be applied to “richly structured high-dimensional sensory data” [38]. He believes that basic, uniform architecture in the cortex and its high adaptability and plasticity to early damage supports the idea that there is such an algorithm. He then talks about strategies that can be combined to create algorithms that are suitable for deep architectures with millions of connections similar to cortex.

b) Why learn deep representations?

Erhan et al. believe that unsupervised pretraining plays a role of optimization and regularization [39]. Their reasoning is that in the beginning of training with stochastic gradient descent (SGD), it is more probable for the weight changes to be larger and more dramatic compared to the later phases, in which the trajectory of SGD will follow a path trapped in a much smaller region and eventually towards a basin of attraction. Even though this late behavior is desired for finding a minimum, this introduces an imbalance regarding the influence of each example in the training set. That is, the main trajectory of SGD and the final basin of attraction *heavily depend on initial parameters and early training examples*. As they show, dramatic effects early on can cause over-fitting and the problem can persist even with large amounts of data given later in the training. Thus the early training often has a powerful and disproportionate effect on the trajectory of SGD. They believe that unsupervised pretraining is effective because, in the above setting, it can find a good starting point for the SGD and allow the trajectory to move more easily towards a basin of attraction that corresponds to the parameters that proved useful in the unsupervised phase. As a future research direction, they propose an investigation of algorithms that can reduce the effect of early training examples.

For another perspective on why learning deep representations work, an analysis of what features unsupervised deep learning captures and how such representations emerge given by Paul and Venkatasubramanian can be a relevant read [40]. They make use of group theory[3] to create a framework to analyze layer-wise pre-training and explain why invariant features emerge within the higher layers. However, their framework can not be used for the supervised case. Therefore, finding a single theory that allows analyzing both supervised and unsupervised case is currently a new research direction in the representation learning theory.

c) Useful intrinsic assumptions

In his overview of unsupervised feature learning techniques, Bengio gives an analysis on priors for representation [2]. For example, *multiple explanatory factors* (or distributed representations) allow the representation to achieve high expressiveness, *hierarchical organization of explanatory factors* allows both learning invariant features and disentangling latent variables, *shared factors across tasks* allow learned parameters to be reusable, *natural clustering* or well separated P(X|Y=i) for different i allows achieving easier classification, *manifolds* allow high dimensional natural data (that in general has a much lower variation than what is possible) to be mapped into a much lower dimensionality and *sparsity* allows variable sized representations as well as reducing the amount of relevant factors for a given sample. Note that, unlike other priors, priors observed in deep representations seem to be often desirable. For instance, fundamental assumptions intrinsic to many shallow models can become limiting factors. Supporting this idea, Bengio and LeCun provide mathematical and empirical evidence that nonparametric learners such as kernel methods can not always efficiently learn high dimensional functions [41]. This, they argue, is due to the fact that kernel machines are shallow architectures because there is one large layer of simple template matchers. Having only a single layer of trainable coefficients and shallow architectures can be very inefficient due to the number of computational elements and examples required, also discussed in Håstad's paper about depth-breadth tradeoff in circuits design [4]. Empirical results in their study, where kernel methods are compared with deep architectures, show that deep architectures have the ability to capture features beyond immediate neighbors; therefore, it is easier to capture abstract features with deep architectures.

In another study, Delalleau and Bengio define *deep architectures* as function families that are equivalent to deep circuits [42]. Deep architectures are hard to train and require good

[3] They use group theory to build a framework based on “orbit-stabilizer interplay in group actions”.

priors to be able to successfully used for representation learning. They point out that when used in combination with priors such as independence, sparsity, grouping (encouraging all of the units in a group to be off together) and slowness (forcing temporal coherency between some units), disentangling factors of variation in the representation is easier.

There are also priors for sparse coding, such as Laplacian that can be used for improving sparsity of the transformed input. Bradley and Bagnell propose smoother priors to achieve differentiable[4] sparse coding [43].

d)Deep architectures can approximate anything

When exponentially deep, even if limited by the width of the input layer, sigmoid belief networks, RBMs and DBNs can approximate any distribution over binary vectors to arbitrary accuracy [44]. The required number of parameters to approximate any distributions in RBMs is equal to that of DBNs.

In another relevant study, Roux and Bengio show that increasing the number of units in a restricted Boltzmann machine (RBM) always improves the modeling power and an RBM can approximate any discrete distribution if we allow an increase in the number of hidden units [45]. Adding more layers to a deep belief network (DBN) can generalize better by having a more compact representation. However, the overall performance is always dependent on the previous layers' ability to stay faithful to the data distribution (i.e. ability to map the data distribution to something close to itself).

e)Representation learning is not a bag of tricks

Rooyen and Williamson suggest a theoretical framework for representation learning [46]. Their theory is about when it is possible to learn a feature in an unsupervised way. They give a new method for representation learning that is not sensitive to loss. They believe that learning generic features boil down to a manifold assumption, that is, it is possible to concentrate high dimensional data into a lower dimensional space, thanks to the low variability of data. Moreover, they suggest using rate-distortion theory (a lossy data compression scheme) and its generalizations to evaluate the quality of features.

f) Representation of transformation itself

Linear transformations can be incorporated into the representation learning algorithms to achieve transformation invariance. For instance, the transformation invariant version of a restricted Boltzmann machine proposed by Lee and Sohn can represent data both by its weights and by its transformations [47]. This representation is used for achieving transformation invariance (e.g. scaling, rotation and translation) via probabilistic max pooling. Moreover, they show that this can also be used in a similar manner in other well known alternative methods such as autoencoders and sparse coding [48].

g) I-theory: how sensory cortex learns representations

Anselmi et al. define *selectivity* in representation as the property that two points have the same representation only if they can be transformed to each other [49]. While talking about invariance and selectivity, they also give supporting mathematical evidence for the claims given in their previous study, which describes a theory about representation learning in sensory cortex, namely *i-theory* [50]. I-theory tries to bring an explanation to how invariant and selective features can be learned in the ventral stream of visual cortex. They believe that learning invariant representations reduces the need for large number of labeled samples for classification tasks and that this is the link between invariant representations and small sample complexity for recognition. Their theory describes how invariant and selective representations can be learned by modules (i.e. HW module) that consist of simple and complex cells to come up with good representations that reduce the sample complexity for supervised learning tasks in the later stages.

h) Alternative to greedy layerwise training

Ngiam et al. describe a pretraining method alternative to greedy layerwise training [51]. They formalize a probabilistic methodology for training deep architectures. They propose a deep MLP that can model the energy landscapes of probabilistic models and using it they jointly train all layers of their learning model at the same time. Their test results indicate that this technique can achieve better performance compared to greedy layer-wise training and stacking.

i) Is it really hard to train deep MLPs?

Glorot and Bengio believe that the answer to above question is that it is easier to train deep MLPs with alternative ways of initialization and more suitable activation functions [52]. Compared to greedy layerwise pretraining algorithm that was proposed in 2006 [35],

[4] The advantage of smoother priors is that they make the sparse code differentiable, adding stability to estimation of maximum a-posteriori (MAP).

directly applying gradient descent algorithms to a randomly initialized feedforward neural network often gives poor results[5]. To better understand the reason for that, Glorot and Bengio analyze the standard gradient descent on randomly initialized deep neural networks [52]. Their first observation is that logistic sigmoid function is not a good activation function for deep networks with random initialization. It turns out that, with logistic sigmoid function, the mean value of a random initialization will saturate the top hidden layers (i.e. vanishing gradient problem [5]). They claim that introducing another type of non-linearity that won't saturate easily can improve the performance of gradient descent algorithm on a randomly initialized deep neural network. The second observation is that training can be harder when each layer's singular Jacobian values diverge from 1. Therefore, a constraint that keeps the Jacobians close to 1 can work. Their work shows that, *even without unsupervised pretraining*, using a more appropriate type of nonlinearity and ways of initialization to prevent the exploding/vanishing gradient problem, deep feedforward networks can be trained directly in a supervised way.

2) Models

CNNs, DBNs, DBMs, deep autoencoders, ICA, manifold learning and hybrid models are discussed. Greedy layer-wise pretraining is still fairly common in many models.

a) How to construct deep recurrent neural networks

Even though the concept of depth might not be as clear as feedforward neural networks, in RNNs, it is possible to benefit from depth. According to Pascanu et al. an RNN can be made deeper at various points such as input-to-hidden function, hidden to hidden transition and hidden to output function [54]. They propose ways to construct deep RNNs, and they show that they empirically perform better than their shallow counterparts.

b)Independent component analysis (ICA)

ICA is a general purpose statistical technique that tries to linearly transform the input into components or latent variables that have distributions that don't reflect the average case that can be achieved via random sampling and maximize both “interestingness” and independence from other components [55] [56]. The estimation of latent variables can be achieved using various methods such as maximum likelihood estimation, minimization of mutual information between components or simply optimizing the objective functions that checks maximum nongaussianity. Oja and Hyvarinen's faster version of ICA (FastICA) is an efficient way for independent component analysis that is parallel, distributed, better suited for nonchanging environments and not gradient based [56]. ICA can be used in various areas such as audio and image signal processing, econometrics and telecommunications.

A weakness of ordinary ICA is that it needs orthonormality constraint for features. Another weakness is, although it is heavily focusing on creating a good representation via generic techniques, for certain cases, further preprocessing that is application dependent might be necessary to significantly increase the success of ICA; for instance, band-pass filtering for an application with time series would increase the performance of ICA [56]. Moreover, ICA is sensitive to whitening (i.e. a linear transformation of vector x into x' so that the covariance matrix of x' equals identity matrix; making variance 1 for all variables in x' and making them uncorrelated).

All of the above disadvantages seem to make ICA unsuitable for high dimensional data. However, Le et al. get around many of the above problems to come up with a technique that allows ICA to learn overcomplete sparse representations [57]. Instead of the orthonormality constraint necessary for the original ICA, they use a soft orthogonalization using reconstruction error minimization (RICA). RICA can learn overcomplete features using a tiled CNN without requiring whitening. They also draw formal connections between sparse autoencoders and ICA.

c)Probabilistic models

Hinton et al. suggest a fast algorithm for training DBNs and introduce the idea of greedy layer-wise training and stacking to successfully train deep networks [35]. In the pretraining phase, they first start with training an RBM and use the hidden layer which learned a set of features from the input as the visible layer of another RBM and so on. After that they stack each trained hidden layer to construct a deep network. If the backward directed weights in the final network are not removed, the resulting network is a DBM and it can be used as a classifier. Training the DBM with the label units results in learning a joint density for labels and inputs. In their model, DBN, they remove the backward directed weights aside from the final hidden layer. A DBN can be used for reconstructing the input together with a label. To achieve that, label units can be added to the layer

[5] Giryes et al. give an analysis on the properties of deep learning when initialized with random wieghts [53].

before the final hidden layer (i.e. add the label units to the visible layer of the top RBM).

In another study, Hinton and Salakhutdinov describe an efficient way to train *deep* autoencoders [58]. They first point out that gradient descent performs well only if there is already a good solution close to the initialized weights. Then they described pretraining, a way to initialize weights in a more sensible way than random assignment. In pretraining, a stack of restricted Boltzmann machines (i.e. each with single layer of feature detectors) are trained. The output of one trained RBM is used as an input for the next RBM for training again. After all the RBMs are trained, they are *unrolled* to construct a deep autoencoder with the same weights. Finally, the stochastic activities are converted to real probabilistic values and fine-tuned with backpropagation algorithm. This procedure effectively creates a deep autoencoder. This method of dimensionality reduction outperformed PCA [59].

The above models have proved to have superior performance in classification tasks compared to nongenerative models and have drawn researchers' interest in the deep learning and representation learning areas. However, using RBMs in the pretraining has a drawback: the visible units in the first layer will be interpreted as probabilistic values coming from the input and representing the input as such is not be suitable for some problems. For instance, pixels from natural images can not be interpreted as probabilistic values, unless the image is already preprocessed (e.g. segmented).

Markov Random Field:

To represent statistically related variables Markov random fields (MRF) can be used [60]. A Markov random field is an undirected graph that can be cyclic, unlike a Bayesian network where the graph is directed and acyclic. Learning dependencies between a set of random variables is possible using a MRF. MRF parameters jointly learned with deep features have the advantage of blending the inference with classification. Using non-linear functions by jointly training multilayer perceptrons with pretrained unary classifiers and simultaneously learning pairwise features, Chen et al.'s proposed algorithm is claimed to be able to efficiently learn deep structural models where the dependencies between the output models are captured via long range connections instead of neighboring chain connections [61].

DBMs are closely connected to MRFs. Representing the DBM's energy as a function of centered states is called “centering trick”. It makes learning more stable and the trained models obtain better generative and discriminative properties. Although this strategy works well with mid-scale models where there are a few hundred hidden units, for the centering trick to work in larger scale models, Montavon and Müller recommend the usage of a regularizer as a way to limit the effective dimensionality [62].

d) Models based on sparse coding

Sparse and overcomplete representations are biologically plausible:

Olshausen and Field connect the biological description of “localized, oriented and bandpass” spatial receptive fields for simple cells in the mammalian visual system with basis functions of wavelet transforms and sparse coding [63]. They also talk about neurobiological implications of sparse coding with an overcomplete basis set. Since in an overcomplete code the basis functions are neither orthogonal nor linearly independent, the subset used from the overcomplete set for a given input will introduce nonlinearities to the input-output function that represents the given input. They believe that this might be the explanation for the weak form of nonlinearities observed in the simple cell responses. Their proposed theory is based on the assumption that the visual system can create an efficient representation of natural input by extracting statistically independent structure in images. They reason that, since sparse coding can reduce the number of elements with statistical dependencies in the representation, it can create an efficient representation. Moreover, the receptive fields that emerge from their algorithm are fairly similar to the ones observed in the primary visual cortex. Finally, they predict that their ideas can be used hierarchically to construct more elaborate models (that can potentially explain higher regions of visual system).

Sparse feature learning for deep belief network:

In general, reconstruction of input representation with certain constraints allows the newly learned representation to have desirable traits such as sparsity or low dimension. Ranzato et al. describe a way to efficiently learn sparse representations using such constraints [64]. Strategies for unsupervised training can involve minimizing a contrastive term in the loss function (e.g. RBMs) or introducing a representation constraint that ensures that training samples can be better reconstructed than random points in the input space (e.g. symmetric sparse encoding). Symmetric sparse encoding has the advantage of efficient training, fast inference and it can work without whitening or mean removal from the input

Efficient learning of sparse representation with an energy based model:

The strategy of learning a set of sparse and overcomplete features with an energy based model to initialize a CNN gave Ranzato et al.'s model state-of-the-art performance on MNIST dataset in 2006 [36].Their main idea is adding a nonlinearity between the encoder and decoder that can sparsify the representation. They believe that, compared to the previous study, where there is a sparsity term in the loss function, which would often require a normalization procedure [65], directly sparsifying the code after the encoder is an improvement in their model. However, they did not try to extend their model to a deeper encoder/decoder architecture or stacking of multiple autoencoders..

e) Convolutional neural networks (CNN)

A CNN is a bioinspired design of MLP in which neurons have overlapping receptive fields. There are variants of CNN models.

Ranzato et al. propose a feature extractor which consists of convolution filters, sigmoid nonlinearity and finally, a feature max-pooling layer [66]. Using two feature extractors in a sequence by giving the features obtained from the first extractor to the second as an input, they combine a layer-wise unsupervised training with convolutional neural network ideas. They claim that this method is better against the over-parameterization problems that are common to purely supervised training methods. Their method is suited for problems with very few labeled training samples.

Hierarchical generative models can be used for constructing deep representations. However, models like deep belief networks (DBN) are not easy to scale and therefore they can not work with full-sized images. Lee et al., suggest a model that improves DBNs using ideas from convolutional neural networks (CNN) [67]. This new type of convolutional deep belief network (CDBN), achieves translation invariance similar to a CNN and it allows inference in a probabilistic manner similar to a DBN. Therefore, CBDN combines the advantages of CNNs and DBNs. Their model learns sparse overcomplete representations. To achieve that they use a regularization method which encourage hidden unit groups to have a low average activation value [68]. Moreover, in their model, they present a novel technique called *probabilistic max-pooling*. Similar to the original *max-pooling* in CNNs, this technique compresses the data in higher layers (“in a probabilistically sound way”). Their approach is suitable for learning deep hierarchical representations in a scalable way, allowing directly training from large images (e.g. 200x200) [69].

Ng and Manning propose and investigate a variety of new deep learning models that can work with unlabeled data [70]. Tiled convolution networks (TCNN) is a modified version of CNNs that relax the constraint regarding the adjacent nodes in a hidden layer having tied connections [71]. By doing so it can learn complex abstract features more easily. However, this comes with a drawback: unlike ordinary CNN, TCNNs can not be trained by an algorithm like CBDN for representation learning. Moreover, the relaxation in the neighboring hidden nodes causes the model to be more free and susceptible to overfitting. To overcome these problems, Ng et al. use an unsupervised training algorithm that is based on independent component analysis (ICA). Since TCNN can learn representations using this algorithm without supervised training, it can easily take advantage of unlabeled data to overcome overfitting problem.

f) Autoencoders

An autoencoder is a MLP that is trained to reconstruct its own input. It has encoder and decoder parts. Encoders transform the input and decoder decodes the transformed representation back to the input. By introducing constraints in the loss function or nonlinearities right after the encoding, an encoder can learn the representation with desired property using backpropagation. However, backpropagation does not work well for deep autoencoders.

Bengio et al. give an empirical analysis of Hinton et al.'s greedy layerwise algorithm for DBNs, give an extension for continuous input values and suggest alternative deep architectures that can be constructed using the same algorithm (i.e. instead of RBMs, they train each layer using autoencoders) [37]. They also confirm that greedy layerwise pretraining can initialize the weights near a better local minimum than random initialization. They believe that this is due to internally achieving a good distributed representation with the pretraining.

Denoising autoencoders:

Denoising autoencoders can be used for extracting and composing robust features [72]. Bengio et al. demonstrate that a generalization of autoencoders that is robust against arbitrary data corruption and reconstruction loss can handle continuous and discrete variables better [73]. Vincent et al. propose a model that can be constructed by stacking denoising autoencoders [74]. The advantage of using denoising autoencoders is that unlike the regularized autoencoders, they can learn Gabor-like filters naturally. Moreover, their final model has higher performance than the ones constructed with regularized autoencoders. A disadvantage of their model is that the decision of selecting the type and level of noise in the input is left to the designer. However, they claim that even without any guidance by prior domain knowledge, their experiments with simple and generic noise types and little tuning displayed high performance. One gap in their study is that they only used shallow denoising autoencoders, and they suspect that an investigation of how to corrupt or add noise to the input best and experiments with stacking deep denoising autoencoders instead of shallow ones might further increase the overall performance of their model. A final critical suggestion they make is learning the types of the corruption that fit to the domain the best from the data itself, rather than hand-engineering.

g) Dimensionality reduction

Wong et al. suggest adapting supervised kernel dimension reduction to a new framework for unsupervised dimensionality reduction [75]. In their framework they derive a low-dimensional representation Z that captures as much information as possible about the original representation X. Autoencoders that try to do that have to learn an encoding function Z=f(X) and a decoding function X=g(Z).

A disadvantage of autoencoders is that number of neurons and the reconstruction scheme needs to be given apriori; for instance, if an ordinary autoencoder is used then the number of nodes in the hidden layer must be smaller than the number of input nodes and the output must be as close to the input as possible. (If it is a denoising autoencoder, then the number of nodes in the hidden layer must be large and the output must be close to the original signal, instead of the input itself that was formed by corrupting the original signal). Therefore, in general, encoder and decoder functions f and g and the number of layers and hidden nodes are all apriori design decisions. In comparison, kernel dimension reduction does not require apriori f and g functions.

Manifold learning:

A manifold is a topological space. It has the property of having neighborhood homeomorphic to an Euclidean space, that is around every point, it is like an Euclidean space [76]. Latent factors of variation in images such as pose, facial expression and morphology are abstract features that can be important to disentangle. Reed et al. propose to disentangle such factors of variations by learning their manifold coordinates and modeling their joint interaction [77]. Their proposed model is a higher order Boltzmann machine where hidden unit groups have multiplicative interactions and each group learns to encode a different latent factor. Their model is claimed to have state-of-the-art results on Toronto Face Database for face and emotion recognition.

In another study, Salakhutdinov and Hinton show a way to transform the input space into low dimensional feature space suitable for K-nearest neighbor classification. Before fine tuning, their method has a pretraining phase that can make use of the unlabeled data to come up with a better non-linear transformation, therefore, it performs well even if the labeled data is limited [78].

Multi-view feature learning works with multi-observational data (e.g. binocular, spatiotemporal etc.) which is often used for the purpose of encoding the relationship between several images. Memisevic's analysis on multi-view feature learning, shows that latent variables can encode transformations using shared rotation angles in the joint eigenspaces of multiple image warps; as a byproduct of learning about transformation specific features, transformation invariant features can emerge [12].

3)Empirical studies

Here, I discuss some of the evaluation results and comparison of various techniques and empirical results about what might be the components that can significantly change the performance of deep architectures. Other empirical studies include [79] [80] [81] [82] and [83].

a)A comparison of deep and shallow

Larochell et al. evaluate deep architectures for problems with many factors of variation and conclude that they outperform SVMs and single hidden-layer feed-forward neural networks [84]. Their empirical analysis shows that this is true until the data distribution becomes too complex and issues such as computational constraints arise.

b) Activation function matters

Rectifiers are the commonly used activation functions for the state of the art neural networks. Zhang et al. propose a generalized version of rectifiers namely Parametric Rectified Linear Units (PReLU) [85]. They also propose an initialization method that is derived particularly for rectifier nonlinearities, which allows them to train deep rectified models directly from scratch. They claim that their new model is the first to surpass human level performance on Imagenet classification dataset. However, their algorithm still makes mistakes in certain types of images that are easy for humans (e.g. context understanding, high level knowledge).

c) Quality of training samples matters

Bengio et al., empirically test whether the most useful features in the representation are the most commonly shared ones [86]. To test this idea they synthetically create *out of distribution* examples. Deep architectures trained with these samples perform better than ones trained with random samples.

d) Quality of coding scheme matters

Yang et al., points to the intrinsic advantages of Hadamard code (i.e. error correction in target coding) for deep representation learning [87]. Target coding is a way to represent the elements of a set with codewords of constant length within a matrix where each row represents a distinct element. Hadamard code is an error correcting code that can be obtained from Hadamard matrix. However, other coding principles are not investigated and the performance of target coding is not tested for shallow classifiers. Therefore, the research on target coding may require more empirical effort. Nonetheless, as in the case of denoising autoencoders [74], the error correcting properties of Hadamard code can be a good advantage in their scheme.

4)Applications

Representation learning and deep learning application areas include vision, audio, speech, robotics, information retrieval, natural language processing, physics and neuroscience. Below, examples from various areas are given.

e)Video classification

Recently, unlike previous architectures designed for short snippet of videos (lasting only seconds), Ng et al. proposed various deep neural network architectures that can work with long videos (minutes) either “by aggregating frame-level CNN outputs into video-level predictions” or by using a long short term memory (LSTM) as a sequence processor [88]. Their first proposed method consists of adapting a CNN to handle full length videos and design choices regarding convolutional temporal pooling architectures. Their second method has a recurrent neural network with long short term memory (LSTM) units and it interprets the video as an ordered sequence of frames.

f) Semantic hashing

Hinton and Salakhutdinov demonstrate a way to learn deep graphical word-count vector models of large set of documents, that can be used for the purpose of information retrieval [89]. They claim that the deepest layer in their network can represent the document better than latent semantic analysis does. Their semantic hashing can be used for filtering the documents given to TF-IDF to achieve better accuracy than giving the raw documents instead.

g)Cross-language semantic representations

Xiao and Guo, demonstrate a bridging technique between representations of different languages [90]. They point out that this is not a trivial problem because, words that exist in different languages can have different disjoint feature spaces. They use their method on discriminative sentiment analysis tasks on Amazon product reviews (that can consist of more than one language) to show that their model performs well.

h)Brain decoding

Vural et. al., propose a deep temporal convolutional neural network that is suitable for problems where there is not enough labeled sample in the training data [91].

i) High energy physics

Sadowski et al., apply deep learning to particle collider data to discover the decay events for Higgs boson to tau leptons (quantum particles) [92].

IV. Future Directions

The research and application areas for representation learning continue to grow and new studies that target the gaps and shortcomings of the existing representation learning techniques comprise some of the literature that might be precursory for future directions. After the shortcomings and gaps in the literature, I give some of the research that addresses biological plausibility.

A. Shortcomings and Gaps

1) Representation is application dependent

According to Guyon et al., evaluation of clustering algorithms can not be considered as an application-independent problem, therefore, usefulness of a cluster[6] representation depends on the application context [94]. They propose building a taxonomy of clustering problems, to differentiate the similar contexts from others and evaluate them according to the context. Therefore, from this perspective, representation

[6] Berkhin's survey on clustering techniques gives detailed information [93].

learning can not be considered an application-independent problem either. *Representation learning is a search for explanatory factors* that are abstract; however, learning some abstractions might not be necessary for certain problems and some of the good abstract features can be more easily learned if application dependent knowledge can be incorporated into the representation. Some semi-supervised models can do that, as in [95] [96] [97], however, the literature can benefit from more models that can be simultaneously trained with labeled and unlabeled data.

2) Can representation learning also generalize into trunsductive models?

Transduction is inference from specific training cases to specific test cases. Due to a lack of generalization step in the middle, (i.e. in induction), sometimes a transductive model can make predictions where an inductive model can not (e.g. when labeled data is too few) [98]. Since a transductive model does not build a predictive model, adding new cases to the dataset requires the transduction process to be done from scratch. In general, representation learning does not work this way, but still it is used where there is not enough labeled data. In the extreme case, where the size of the representation is data-driven, (e.g. stacking new layers on top of each other according to a stopping criteria acquired from the data) it is possible to learn sample specific examples and make inference from them. The problem is that representation learning is not necessarily application dependent, therefore, directly representing samples instead of making generalizations is not desired. However, for some problems, the opposite might be necessary.

3) Biological plausibility and human like inference

An analysis of deep neural networks suggests that they carry off the semantic information to the higher levels as a space transformation but the units do not acquire semantic meaning individually [99]. Moreover, slight changes in a test case, which will be imperceptible to human visual system, can have big effects on a deep architecture, indicating that fairly discontinuous input-output mappings can be learned by them. In fact, it is claimed that such slight changes with dramatic effects can be found for virtually every test case. Another related observation made by Nguyen et al. is that the reverse is also true. That is, the images unrecognizable to humans (e.g. white noise) can be “recognized” (e.g. lion) by the models with very high confidence [100]. Thus, currently it appears that, even if deep neural networks and human visual system have certain similarities, success of current models should not be taken as a clear indication of human-like generalization capabilities.

4) Building blocks of models can be changed

By combining inspiration from biology with inspiration from digital computers, different types of neural networks can be designed. For instance, when coupled with an external memory, neural networks obtain a Von Neumann like architecture. The modified neural networks can act like a Turing Machine that maintains a property of conventional neural networks, that is being differentiable end-to-end so that they can be trained with a gradient descent algorithm [101]. They are called Neural Turing Machines and they can learn various fundamental operations or algorithms such as copying, sorting or they can act like an associative memory.

5) How to evaluate a good representation?

Evaluation of a learned representation is a key concept that needs to be addressed by more researchers. Aside from the indirect way of evaluating learned representations via training several classifiers on top of them to make a performance comparison, the suggestions on how to evaluate the efficiency of a representation are limited in the literature. Since invariance is a desired property in a learned feature, methods that can measure invariance are proposed. For instance, Goodfellow et al. demonstrate two methods to evaluate learned representations for their level of invariance to input transformations [102]. Their evaluation of convolutional deep belief networks and stacked autoencoders shows that convolutional networks learn many more invariant features than stacked autoencoders. For a measure of invariance, they have considered hidden unit responses above a certain threshold (firing) as an indication of a feature[7]. They adjust the threshold for each unit so that all units will “fire” at the same rate when the input is random. Then they measure the robustness of each unit by first finding an input that makes them fire, and then by measuring their firing rate when they transform the input in various ways (i.e. translation, 2D rotation, 3D rotation). The overall score is found by dividing the average score of all the hidden units by the average score achieved by the deepest layer. They also suggest another metric for evaluation which is identity invariance, for which none of their test models achieved ga ood score. They claim that this is an indication of a lack of sophistication in the current models. Even though Goodfellow et al. [102] and [46] as mentioned in subsection III.B.1.e, propose ways

[7] This is in conflict with another study that claims that features do not necessarily correspond to hidden nodes [100]

to evaluate features, the research in this area is still limited.

B. Biological Plausibility

1) Neurons have different receptive field size

Techniques in the representation learning and deep learning allowed researchers to successfully train very large deep architectures that can learn a high number of features. Although the possible advantages of learning a high number of features [7] and training with a large dataset [9] are well known, training a large network brings some practical problems. For instance, a high number of feature extractors brings a quadratic increase to the number of parameters required to be learned. In other words, there is a quadratic increase in the number of parameters to be learned with the width of the network. For this reason, researchers often decrease the connectivity between layers by hand in order to train a deep architecture faster. The traditional, bioinspired way to train deep architectures is to manually limit the receptive fields of higher level nodes or features (as in [103]). However, it is often not obvious how to limit the receptive fields best. To automate this process, Coates and Ng suggest a data-driven way to select the receptive fields [104]. After the features are pretrained, their algorithm first groups features according to a similarity metric[8] that allows a useful grouping according to feature responses. Then it puts the features in the same group into the same receptive field. Therefore, their algorithm finds a useful grouping or clustering of features in each layer that can be used as receptive fields for one or more feature extractors in the next layer.

2) Neurons have different activation functions

In ANN models, it is common to have a single and unchanging non-linear activation function. However, the capability to have independent activation functions for each neurons can have expressive advantages that can improve the performance of a model. Agostinelli et al., propose a piecewise linear activation function that can be independently learned with gradient descent for each neuron. Due to the adaptive nature of this new activation function, their model is claimed to perform better then deep neural networks with fixed rectified linear units [105].

3) Target propagation algorithm

Bengio et al. explore new methods for deep learning and representation learning, that can be more biologically plausible [106]. Firstly, they believe that the biologically plausible update rule for synaptic weights, namely spike timing dependent plasticity (STDP), is a form of gradient descent on an objective function. The objective function, they claim, can be reward driven, supervised or unsupervised so long as weights can be updated to adjust firing rates of the neurons in the direction that can improve the objective function. Moreover, they believe that this can be interpreted as a variational expectation maximization (EM) algorithm i.e. working with posteriors that are inexact and approximated by neural dynamics.

Target propagation is a new way to train deep networks [107]. Layers of reconstruction-based autoencoders can be exploited to perform credit assignment without relying on backpropagation and derivatives. The advantage of such a scheme is that it is not affected by several layers of strong nonlinearities, unlike backpropagation algorithm which would perform poorly. By providing both the target and the input in the input layer, stacked autoencoders can propagate targets via the reconstructions they compute. This is called “target propagation” or “targetprop”. As an alternative that can also avoid some of the pitfalls of backpropagation, targetprop also inspired another recent study regarding biological plausibility in deep learning.

Bengio et al. claim that the gradients that will be used for updating the hidden states of EM algorithm can be approximated with pairwise interactions between layers, which they believe eventually learn to form a denoising auto-encoder using the aforementioned targetprop algorithm [106]. They further claim that exploring biologically plausible alternatives to backpropagation can also be useful for machine learning. For instance, intrinsic reliance to smoothness and derivatives in backpropagation is not required in targetprop.

4) Models that can mimic prenatal activity

Researchers show that prenatal retina already develops similar *receptive field mosaics* to adults; that is, receptive field center distribution and overlap distribution of ganglion cells are already fairly developed before birth [108]. One explanation is that in the prenatal retina, there are spontaneous patterned waves of activity. Such activity may effectively refine connections in an orderly way to represent maps of sensory space [109]. It was already observed decades ago that prenatal retinal ganglion cells activate (almost once per minute) in a periodical manner [110]. It turns out, the activation of ganglion cell can propagate from one cell to the next like a wave. This means that genetically controlled mechanisms might have an important role on the initialization of the topology and connection weights of the neural network that

[8] Similarity metric can be arbitrarily specified by the researcher

constitutes at least some portion of the visual system [111]. Therefore, I believe that, new representation learning techniques can arise from bioinspired mechanisms for initialization.

V. Conclusion And Future Work

An analysis of the literature on representation learning was given. Then some of the possible precursory research studies on relatively unexplored areas were mentioned.

Representation learning, as the driving force behind many deep learning approaches, is starting to acquire a theoretical foundation and empirical support, as mentioned in subsections III.B.1, III.B.2 and III.B.3. Moreover, there are already precursory studies (as discussed in section IV) that might address the shortcomings in the area such as the evaluation problem, application dependent representations, exploration of model expressiveness and biological plausibility.

As a future study, I plan to implement a model to investigate the direction I discussed in subsection IV.B.4.

Acknowledgment

Many thanks to Ferda Nur Alpaslan and Uğur Halıcı for their suggestions and guidance. Warm thanks to Joanna Aileen Thompson for her proofreading skills.

References

[1] M. Minsky, “Steps Toward Artificial Intelligence,” in *Computers and Thought*, E. Feigenbaum and J. Feldman, Eds. McGraw-Hill, New York, 1963, pp. 406–450.

[2] Y. Bengio, A. Courville, and P. Vincent, “Representation Learning: A Review and New Perspectives,” *Pattern Anal. Mach. Intell. IEEE Trans. On*, vol. 35, no. 8, pp. 1798–1828, 2013.

[3] J. Schmidhuber, “Deep Learning in Neural Networks: An Overview,” The Swiss AI Lab IDSIA, IDSIA-03-14 / arXiv:1404.7828v1 [cs.NE], 2014.

[4] J. H\a astad, *Computational limitations of small-depth circuits*. MIT press, 1987.

[5] S. Hochreiter, “Untersuchungen zu dynamischen neuronalen Netzen,” *Masters Thesis Inst. Inform. Tech. Univ. Munchen*, 1991.

[6] Y. Bengio, *Learning Deep Architectures for AI. Foundations and Trends in Machine Learning, V2(1)*. Now Publishers, 2009.

[7] A. Coates, H. Lee, and A. Ng, “An Analysis of Single-Layer Networks in Unsupervised Feature Learning,” in *Advances in Neural Information Processing Systems*, 2010.

[8] J. Gantz and D. Reinsel, “The digital universe in 2020: Big data, bigger digital shadows, and biggest growth in the far east,” *IDC IView IDC Anal. Future*, vol. 2007, pp. 1–16, 2012.

[9] M. Banko and E. Brill, “Scaling to very very large corpora for natural language disambiguation,” in *Proceedings of the 39th Annual Meeting on Association for Computational Linguistics*, 2001, pp. 26–33.

[10] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Y. Ng, “Multimodal deep learning,” in *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, 2011, pp. 689–696.

[11] J. Weston, S. Bengio, and N. Usunier, “Large scale image annotation: learning to rank with joint word-image embeddings,” *Mach. Learn.*, vol. 81, no. 1, pp. 21–35, 2010.

[12] R. Memisevic, “On multi-view feature learning,” in *Proceedings of the 29th International Conference on Machine Learning (ICML-12)*, 2012, pp. 161–168.

[13] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, “Distributed representations of words and phrases and their compositionality,” in *Advances in Neural Information Processing Systems*, 2013, pp. 3111–3119.

[14] W. McCulloch and W. Pitts, “A logical calculus of the ideas immanent in nervous activity,” *Bull. Math. Biophys.*, vol. 7, pp. 115–133, 1943.

[15] F. Rosenblatt, “The perceptron: a probabilistic model for information storage and organization in the brain.,” *Psychol. Rev.*, vol. 65, no. 6, p. 386, 1958.

[16] A. E. Bryson, W. F. Denham, and S. E. Dreyfus, “Optimal programming problems with inequality constraints,” *AIAA J.*, vol. 1, no. 11, pp. 2544–2550, 1963.

[17] K. Fukushima, “Neocognitron: A self-organizing neural network for a mechanism of pattern recognition unaffected by shift in position,” *Biol. Cybern.*, vol. 36, no. 4, pp. 193–202, 1980.

[18] J. J. Hopfield, “Neural Networks and physical systems with emergent collective computational abilities,” *Proc Natl. Acad. Sci.*, vol. 79, pp. 2554–2558, 1982.

[19] E. Oja, “Simplified neuron model as a principal component analyzer,” *J. Math. Biol.*, vol. 15, no. 3, pp. 267–273, 1982.

[20] T. Kohonen, “Self-Organized Formation of Topologically Correct Feature Maps,” *Biol. Cybern.*, vol. 43, no. 1, pp. 59–69, 1982.

[21] D. H. Ackley, G. E. Hinton, and T. J. Sejnowski, “A learning algorithm for boltzmann machines*,” *Cogn. Sci.*, vol. 9, no. 1, pp. 147–169, 1985.

[22] P. Smolensky, “Information processing in dynamical systems: foundations of harmony theory,” in *Parallel distributed processing: explorations in the microstructure of cognition, vol. 1*, 1986, pp. 194–281.

[23] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning Internal Representations by Error Propagation,” in *Parallel Distributed Processing*, vol. 1, D. E. Rumelhart and J. L. McClelland, Eds. MIT Press, 1986, pp. 318–362.

[24] D. S. Broomhead and D. Lowe, “Radial basis functions, multi-variable functional interpolation and adaptive networks,” DTIC Document, 1988.

[25] P. Baldi and K. Hornik, “Neural Networks and Principal Component Analysis: Learning from Examples Without Local Minima,” *Neural Netw.*, vol. 2, pp. 53–58, 1989.

[26] M. Kramer, “Nonlinear Principal Component Analysis Using Autoassociative Neural Networks,” *AIChE J.*, vol. 37, pp. 233–243, 1991.

[27] R. M. Neal, “Connectionist learning of belief networks,” *Artif. Intell.*, vol. 56, no. 1, pp. 71–113, 1992.

[28] C. Cortes and V. Vapnik, “Support-vector networks,” *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, 1995.

[29] B. A. Olshausen and D. J. Field, “Emergence of simple-cell receptive field properties by learning a sparse code for natural images,” *Nature*, vol. 381, no. 6583, pp. 607–609, 1996.

[30] S. Behnke, “Discovering Hierarchical Speech Features using Convolutional Non-negative Matrix Factorization,” in *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, 2003, vol. 4, pp. 2758–2763.

[31] K. Fukushima, “Neural network model for a mechanism of pattern recognition unaffected by shift in position - Neocognitron,” *Trans IECE*, vol. J62-A(10), pp. 658–665, 1979.

[32] D. H. Hubel and T. N. Wiesel, “Receptive Fields, Binocular Interaction, and Functional Architecture in the Cat’s Visual Cortex,” *J. Physiol. Lond.*, vol. 160, pp. 106–154, 1962.

[33] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, “Backpropagation applied to handwritten zip code recognition,” *Neural Comput.*, vol. 1, no. 4, pp. 541–551, 1989.

[34] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-Based Learning Applied to Document Recognition,” *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[35] G. E. Hinton, S. Osindero, and Y.-W. Teh, “A Fast Learning Algorithm for Deep Belief Nets,” *Neural Comput.*, vol. 18, no. 7, pp. 1527–1554, May 2006.

[36] M. Ranzato, C. Poultney, S. Chopra, and Y. Lecun, “Efficient Learning of Sparse Representations with an Energy-Based Model,” in *Advances in Neural Information Processing Systems (NIPS 2006)*, 2006, pp. 1137–1144.

[37] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle, “Greedy layer-wise training of deep networks,” in *Neural Information Processing Systems (NIPS)*, 2007.

[38] G. E. Hinton, “To recognize shapes, first learn to generate images,” *Prog. Brain Res.*, vol. 165, pp. 535–547, 2007.

[39] D. Erhan, Y. Bengio, A. Courville, P.-A. Manzagol, P. Vincent, and S. Bengio, “Why Does Unsupervised Pre-training Help Deep Learning?,” *J Mach Learn Res*, vol. 11, pp. 625–660, Mar. 2010.

[40] A. Paul and S. Venkatasubramanian, “Why does Deep Learning work?-A perspective from Group Theory,” *ArXiv Prepr. ArXiv14126621*, 2014.

[41] Y. Bengio and Y. LeCun, “Scaling learning algorithms towards AI,” in *Large-Scale Kernel Machines*, 2007.

[42] Y. Bengio and O. Delalleau, “On the expressive power of deep architectures,” in *Algorithmic Learning Theory*, 2011, pp. 18–36.

[43] D. M. Bradley and J. A. Bagnell, “Differential sparse coding,” 2008.

[44] N. Le Roux and Y. Bengio, “Deep belief networks are compact universal approximators,” *Neural Comput.*, vol. 22, no. 8, pp. 2192–2207, 2010.

[45] N. Le Roux and Y. Bengio, “Representational power of restricted Boltzmann machines and deep belief

networks,” *Neural Comput.*, vol. 20, no. 6, pp. 1631–1649, 2008.

[46] B. van Rooyen and R. C. Williamson, “A Theory of Feature Learning,” *ArXiv Prepr. ArXiv150400083*, 2015.

[47] K. Sohn and H. Lee, “Learning invariant representations with local transformations,” *ArXiv Prepr. ArXiv12066418*, 2012.

[48] I. J. Goodfellow, A. Courville, and Y. Bengio, “Spike-and-Slab Sparse Coding for Unsupervised Feature Discovery,” in *NIPS Workshop on Challenges in Learning Hierarchical Models*, 2011.

[49] F. Anselmi, L. Rosasco, and T. Poggio, “On invariance and selectivity in representation learning,” *ArXiv Prepr. ArXiv150305938*, 2015.

[50] F. Anselmi and T. A. Poggio, “Representation Learning in Sensory Cortex: a theory,” 2014.

[51] J. Ngiam, Z. Chen, P. W. Koh, and A. Y. Ng, “Learning deep energy models,” in *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, 2011, pp. 1105–1112.

[52] X. Glorot and Y. Bengio, “Understanding the difficulty of training deep feedforward neural networks,” in *International conference on artificial intelligence and statistics*, 2010, pp. 249–256.

[53] R. Giryes, G. Sapiro, and A. M. Bronstein, “On the Stability of Deep Networks,” *ArXiv Prepr. ArXiv14125896*, 2014.

[54] R. Pascanu, C. Gulcehre, K. Cho, and Y. Bengio, “How to Construct Deep Recurrent Neural Networks,” *ArXiv Prepr. ArXiv13126026*, 2013.

[55] P. Comon, “Independent component analysis – a new concept?,” *Signal Process.*, vol. 36, no. 3, pp. 287–314, 1994.

[56] A. Hyvärinen and E. Oja, “Independent component analysis: algorithms and applications,” *Neural Netw.*, vol. 13, no. 4, pp. 411–430, 2000.

[57] Q. V. Le, A. Karpenko, J. Ngiam, and A. Y. Ng, “ICA with reconstruction cost for efficient overcomplete feature learning,” in *Advances in Neural Information Processing Systems*, 2011, pp. 1017–1025.

[58] G. E. Hinton and R. R. Salakhutdinov, “Reducing the dimensionality of data with neural networks,” *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[59] M. E. Tipping and C. M. Bishop, “Probabilistic principal component analysis,” *J. R. Stat. Soc. Ser. B Stat. Methodol.*, vol. 61, no. 3, pp. 611–622, 1999.

[60] R. Kindermann, J. L. Snell, and others, *Markov random fields and their applications*, vol. 1. American Mathematical Society Providence, RI, 1980.

[61] L.-C. Chen, A. G. Schwing, A. L. Yuille, and R. Urtasun, “Learning Deep Structured Models,” *ArXiv Prepr. ArXiv14072538*, 2014.

[62] G. Montavon and K.-R. Müller, “Deep Boltzmann machines and the centering trick,” in *Neural Networks: Tricks of the Trade*, Springer, 2012, pp. 621–637.

[63] B. A. Olshausen and D. J. Field, “Sparse coding with an overcomplete basis set: A strategy employed by V1?,” *Vision Res.*, vol. 37, no. 23, pp. 3311–3325, Dec. 1997.

[64] Y. Boureau, Y. L. Cun, and others, “Sparse feature learning for deep belief networks,” in *Advances in neural information processing systems*, 2008, pp. 1185–1192.

[65] B. A. Olshausen, “Sparse codes and spikes,” *Probabilistic Models Brain Percept. Neural Funct.*, pp. 257–272, 2002.

[66] M. A. Ranzato, F. J. Huang, Y.-L. Boureau, and Y. LeCun, “Unsupervised Learning of Invariant Feature Hierarchies with Applications to Object Recognition,” in *Computer Vision and Pattern Recognition, 2007. CVPR ’07. IEEE Conference on*, 2007, pp. 1–8.

[67] 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 International Conference on Machine Learning (ICML)*, 2009, pp. 609–616.

[68] H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng, “Unsupervised learning of hierarchical representations with convolutional deep belief networks,” *Commun. ACM*, vol. 54, no. 10, pp. 95–103, 2011.

[69] H. Lee, *Unsupervised feature learning via sparse hierarchical representations*. Stanford University, 2010.

[70] A. Y. Ng and C. D. Manning, “Discovery of Deep Structure from Unlabeled Data,” DTIC Document, 2014.

[71] J. Ngiam, Z. Chen, D. Chia, P. W. Koh, Q. V. Le, and A. Y. Ng, “Tiled convolutional neural networks,” in *Advances in Neural Information Processing Systems*, 2010, pp. 1279–1287.

[72] P. Vincent, L. Hugo, Y. Bengio, and P.-A. Manzagol, “Extracting and composing robust features with denoising autoencoders,” in *Proceedings of the 25th international conference on Machine learning*, New York, NY, USA, 2008, pp. 1096–1103.

[73] Y. Bengio, L. Yao, G. Alain, and P. Vincent, “Generalized denoising auto-encoders as generative models,” in *Advances in Neural Information Processing Systems*, 2013, pp. 899–907.

[74] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, “Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion,” *J. Mach. Learn. Res.*, vol. 11, pp. 3371–3408, 2010.

[75] M. Wang, F. Sha, and M. I. Jordan, “Unsupervised kernel dimension reduction,” in *Advances in Neural Information Processing Systems*, 2010, pp. 2379–2387.

[76] M. H. Freedman and F. Quinn, *Topology of 4-manifolds*, vol. 39. Princeton University Press Princeton, 1990.

[77] S. Reed, K. Sohn, Y. Zhang, and H. Lee, “Learning to disentangle factors of variation with manifold interaction,” in *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, 2014, pp. 1431–1439.

[78] R. Salakhutdinov and G. Hinton, “Learning a Nonlinear Embedding by Preserving Class Neighbourhood Structure,” in *Proceedings of the International Conference on Artificial Intelligence and Statistics*, 2007, vol. 11.

[79] B. Huval, T. Wang, S. Tandon, J. Kiske, W. Song, J. Pazhayampallil, M. Andriluka, R. Cheng-Yue, F. Mujica, A. Coates, and others, “An Empirical Evaluation of Deep Learning on Highway Driving,” *ArXiv Prepr. ArXiv150401716*, 2015.

[80] J. Jang, Y. Park, and I. H. Suh, “Empirical evaluation on deep learning of depth feature for human activity recognition,” in *Neural Information Processing*, 2013, pp. 576–583.

[81] J. Ngiam, A. Coates, A. Lahiri, B. Prochnow, Q. V. Le, and A. Y. Ng, “On optimization methods for deep learning,” in *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, 2011, pp. 265–272.

[82] G. Hu, Y. Yang, D. Yi, J. Kittler, W. Christmas, S. Z. Li, and T. Hospedales, “When Face Recognition Meets with Deep Learning: an Evaluation of Convolutional Neural Networks for Face Recognition,” *ArXiv Prepr. ArXiv150402351*, 2015.

[83] S. E. Fahlman, “An Empirical Study of Learning Speed in Back-Propagation Networks,” Carnegie-Mellon Univ., CMU-CS-88-162, 1988.

[84] 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.

[85] K. He, X. Zhang, S. Ren, and J. Sun, “Delving deep into rectifiers: Surpassing human-level performance on imagenet classification,” *ArXiv Prepr. ArXiv150201852*, 2015.

[86] Y. Bengio, F. Bastien, A. Bergeron, N. Boulanger-Lewandowski, T. M. Breuel, Y. Chherawala, M. Cisse, M. Côté, D. Erhan, J. Eustache, and others, “Deep learners benefit more from out-of-distribution examples,” in *International Conference on Artificial Intelligence and Statistics*, 2011, pp. 164–172.

[87] S. Yang, P. Luo, C. C. Loy, K. W. Shum, and X. Tang, “Deep Representation Learning with Target Coding,” 2015.

[88] J. Y.-H. Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici, “Beyond Short Snippets: Deep Networks for Video Classification,” *ArXiv Prepr. ArXiv150308909*, 2015.

[89] R. Salakhutdinov and G. Hinton, “Semantic hashing,” *RBM*, vol. 500, no. 3, p. 500, 2007.

[90] M. Xiao and Y. Guo, “A novel two-step method for cross language representation learning,” in *Advances in Neural Information Processing Systems*, 2013, pp. 1259–1267.

[91] O. Firat, E. Aksan, I. Oztekin, and F. T. Y. Vural, “Learning Deep Temporal Representations for Brain Decoding,” *ArXiv Prepr. ArXiv14127522*, 2014.

[92] P. J. Sadowski, D. Whiteson, and P. Baldi, “Searching for Higgs Boson Decay Modes with Deep Learning,” in *Advances in Neural Information Processing Systems*, 2014, pp. 2393–2401.

[93] P. Berkhin, “A survey of clustering data mining techniques,” in *Grouping multidimensional data*, Springer, 2006, pp. 25–71.

[94] I. Guyon, U. Von Luxburg, and R. C. Williamson, “Clustering: Science or art,” in *NIPS 2009 Workshop on Clustering Theory*, 2009.

[95] X. Song, Z. Liu, X. Yang, J. Yang, and Y. Qi, “Extended Semi-supervised Fuzzy Learning Method for Nonlinear Outliers via Pattern Discovery,” *Appl. Soft Comput.*, 2015.

[96] G. Papandreou, L.-C. Chen, K. Murphy, and A. L. Yuille, “Weakly-and Semi-Supervised Learning of a DCNN for Semantic Image Segmentation,” *ArXiv Prepr. ArXiv150202734*, 2015.

[97] D.-H. Lee, “Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks,” in *Workshop on Challenges in Representation Learning, ICML*, 2013.

[98] V. N. Vapnik and V. Vapnik, *Statistical learning theory*, vol. 1. Wiley New York, 1998.

[99] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, “Intriguing properties of neural networks,” *ArXiv Prepr. ArXiv13126199*, 2013.

[100] A. Nguyen, J. Yosinski, and J. Clune, “Deep Neural Networks are Easily Fooled: High Confidence Predictions for Unrecognizable Images,” *ArXiv Prepr. ArXiv14121897*, 2014.

[101] A. Graves, G. Wayne, and I. Danihelka, “Neural Turing Machines,” *ArXiv Prepr. ArXiv14105401*, 2014.

[102] I. Goodfellow, H. Lee, Q. V. Le, A. Saxe, and A. Y. Ng, “Measuring invariances in deep networks,” in *Advances in neural information processing systems*, 2009, pp. 646–654.

[103] D. C. Ciresan, U. Meier, J. Masci, L. M. Gambardella, and J. Schmidhuber, “High-Performance Neural Networks for Visual Object Classification,” IDSIA, Feb. 2011.

[104] A. Coates and A. Y. Ng, “Selecting receptive fields in deep networks,” in *Advances in Neural Information Processing Systems*, 2011, pp. 2528–2536.

[105] F. Agostinelli, M. Hoffman, P. Sadowski, and P. Baldi, “Learning Activation Functions to Improve Deep Neural Networks,” *ArXiv Prepr. ArXiv14126830*, 2014.

[106] Y. Bengio, D.-H. Lee, J. Bornschein, and Z. Lin, “Towards Biologically Plausible Deep Learning,” *ArXiv Prepr. ArXiv150204156*, 2015.

[107] Y. Bengio, “How auto-encoders could provide credit assignment in deep networks via target propagation,” *ArXiv Prepr. ArXiv14077906*, 2014.

[108] A. Anishchenko, M. Greschner, J. Elstrott, A. Sher, A. M. Litke, M. B. Feller, and E. Chichilnisky, “Receptive field mosaics of retinal ganglion cells are established without visual experience,” *J. Neurophysiol.*, vol. 103, no. 4, pp. 1856–1864, 2010.

[109] C. L. Torborg and M. B. Feller, “Spontaneous patterned retinal activity and the refinement of retinal projections,” *Prog. Neurobiol.*, vol. 76, no. 4, pp. 213–235, 2005.

[110] L. Galli and L. Maffei, “Spontaneous impulse activity of rat retinal ganglion cells in prenatal life,” *Science*, vol. 242, no. 4875, pp. 90–91, 1988.

[111] A. G. Polat, “Modeling neurons that can self organize into building blocks and hiearchies,” Middle East Technical University, 2012.