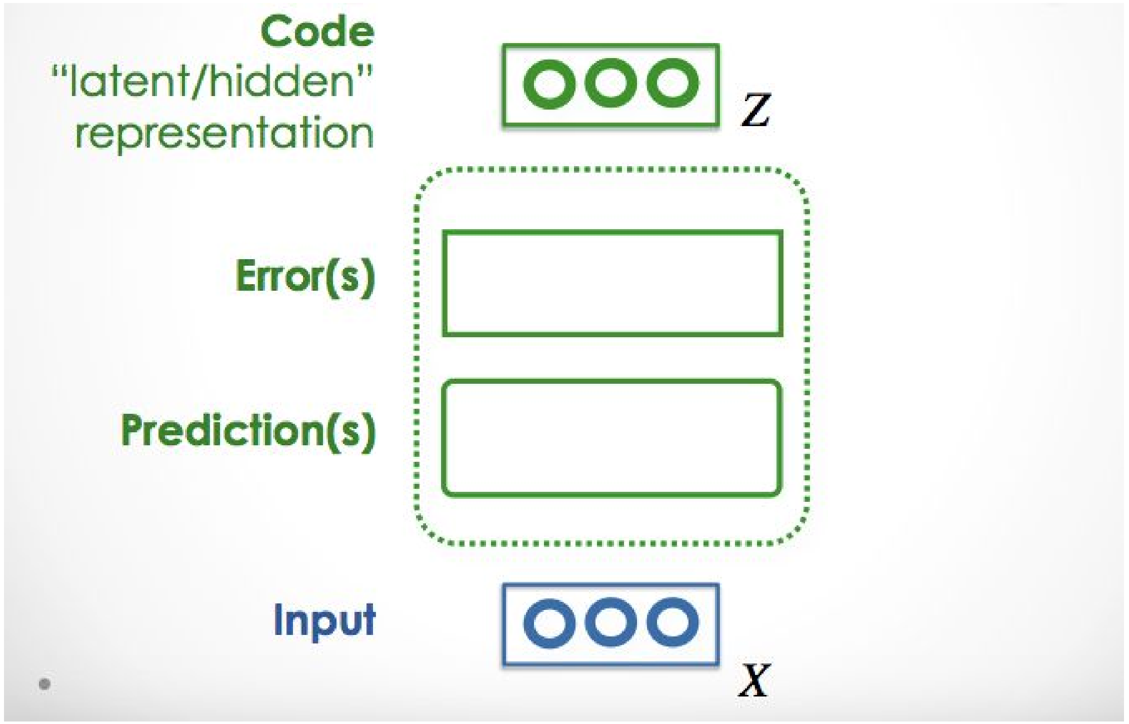
Auto Encoders

Alex Litzenberger

What they are.

- A method for unsupervised learning
- Creates and encoding of the input with higher level representations

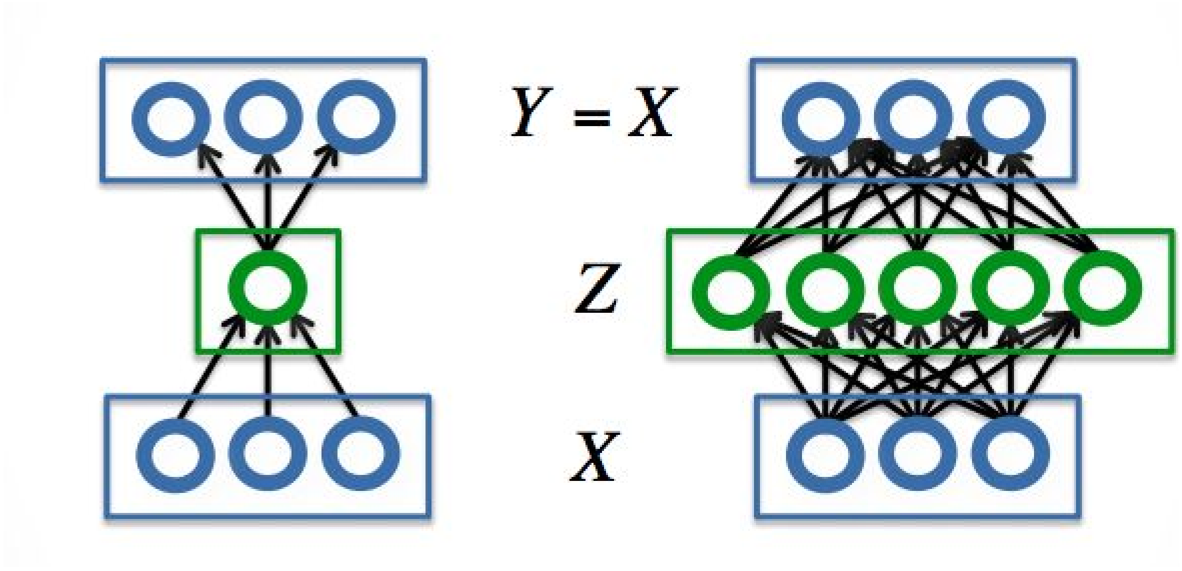


Deep autoencoders

Utilise deep neural networks to autoencode

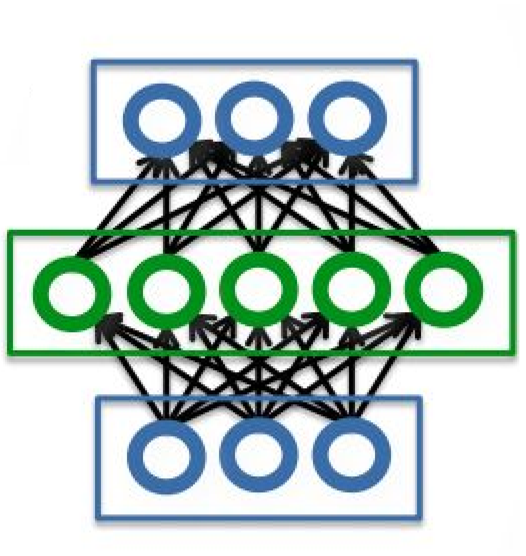
Sparsity and Density

Since the goal is to get an encoding of the input of the information that is informative it is desirable for the encoder to learn features of the input data. If sparsity or density were not enforced on the machine it would simply learn the identity function so auto encoders tend to either be sparse or dense.



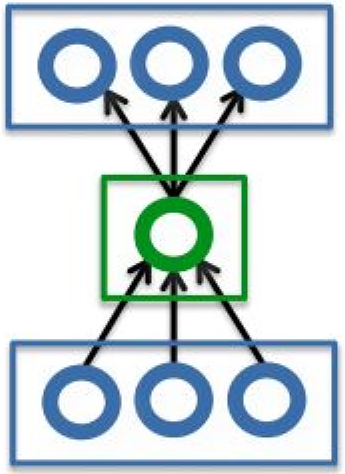
Sparsity

Entails the cost function taking account of the total value of the output weights.

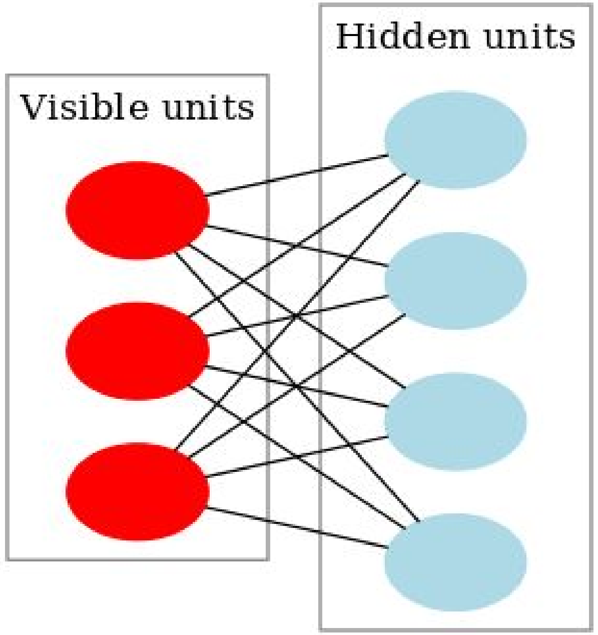


Density

Entails having few parameters in the encoding.



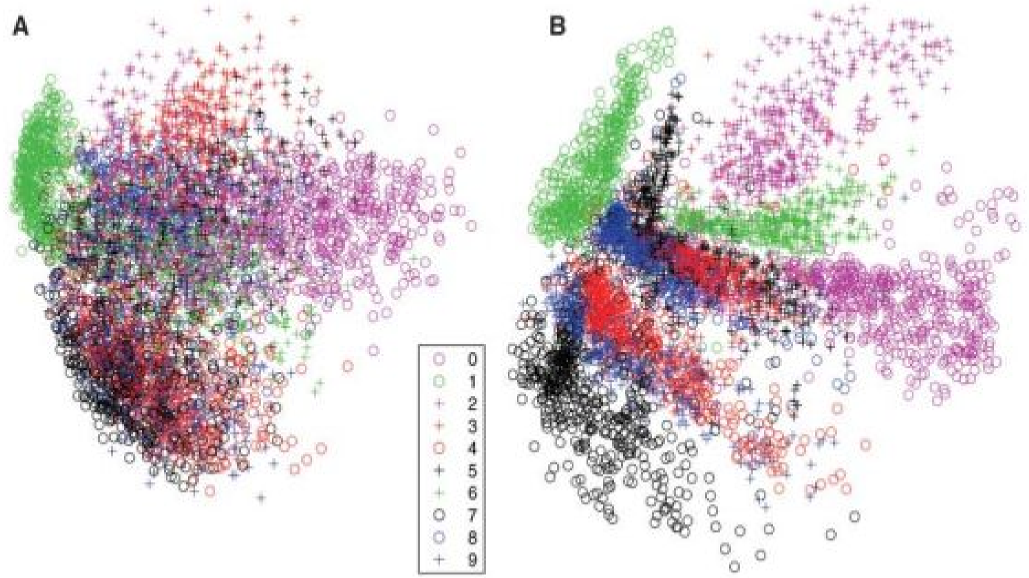
Restricted boltzmann machines (RBMs)

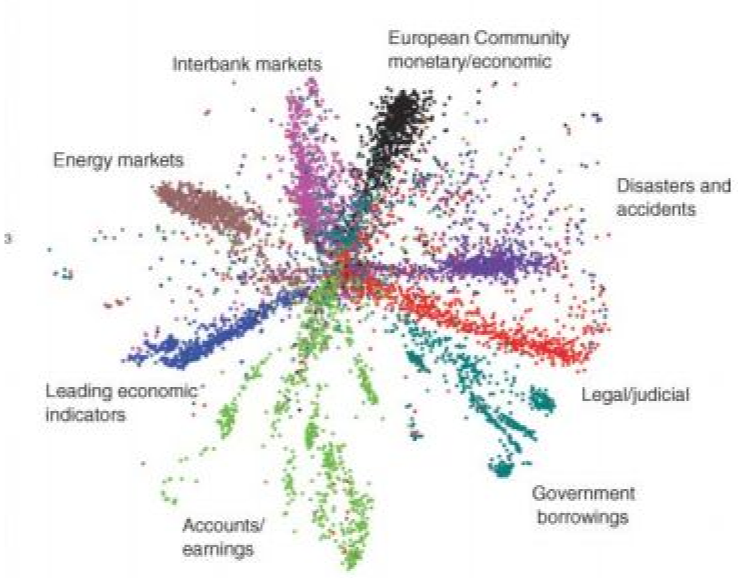


- Two layer neural networks
- Generative
- Learn features of the visible layer
- Have sparse or dense representations
- Cost function is a function of reconstruction accuracy.
- Input is the goal output

A bit of History

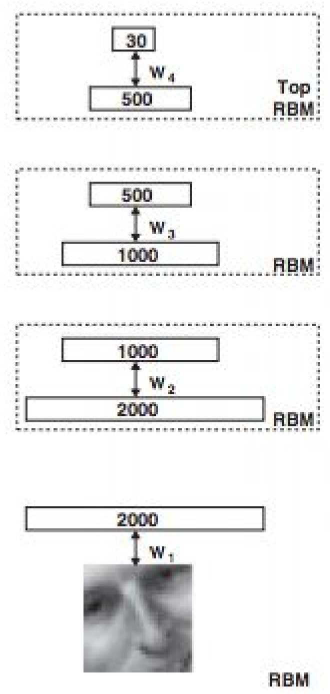
The creation of DNN autoencoders is usually credited to Hinton & Salakhutdinov and their 2006 paper *Reducing the Dimensionality of Data with Neural Networks*





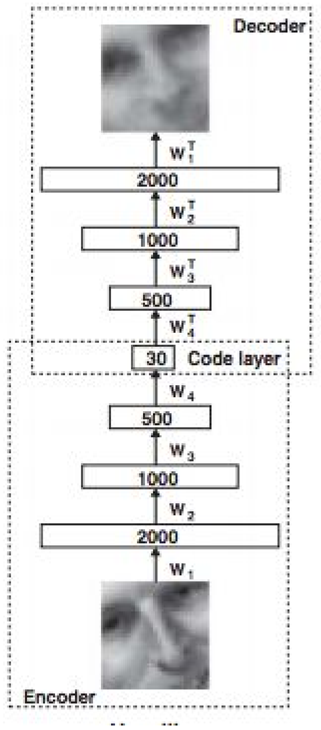
Stacked RBMs

- The paper showed that RBMs could be effectively stacked
- The outputs to a given RBM were used to train the next RBM
- This allows feature encoding multiple times over.
- The layers usually get progressively more sparse or dense.



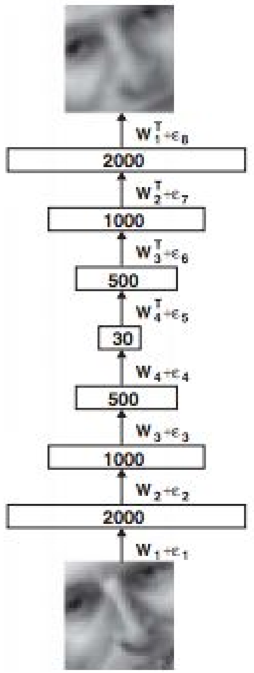
Unrolling

- Once the RBMs are trained the encoder can be “unrolled”
- The encoder simply allows the output from one level to be used as input for the next directly rather than for training.
- The decoder inverts the activation function
- Feeds forward through the layers in reverse order
- Initialised with the same weights as the encoder



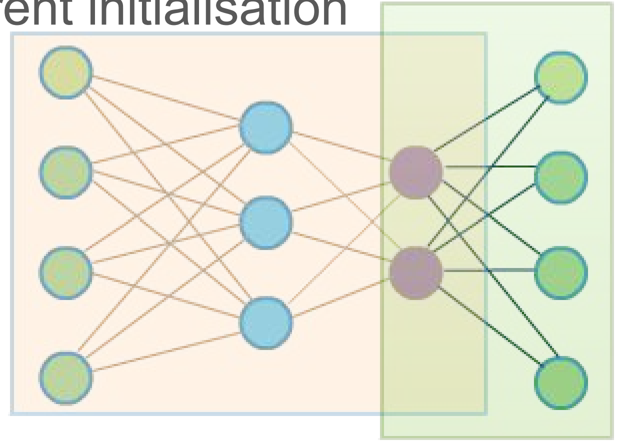
Training an autoencoder

- Referred to as fine tuning
- The unrolled autoencoder is trained on reproducing input.
- Uses standard backpropagation and stochastic gradient descent.
- Takes account i the multi-layer interactions to produce better reproductions and representations.



Symmetric vs. Asymmetric

- So far only symmetric auto-encoders have been covered.
- There are also Asymmetric auto-encoders.
- Rather than unrolling the network a distinct decoder is used.
- This decoder could be a neural network with different initialisation parameters
- It could also be an of an entirely different form.
- Autoencoders are not limited to deep neural nets!



Advantages of Autoencoders

- Produce higher level feature encodings.
- Can be trained for specific desirable encodings.
- Search, and encoding are fast after learning is done.
- Can leverage large amounts of unlabeled data.
- Allow more transparency than most neural networks into their inner workings.
- Encodings can produce independant incite into the underlying structure of the data.

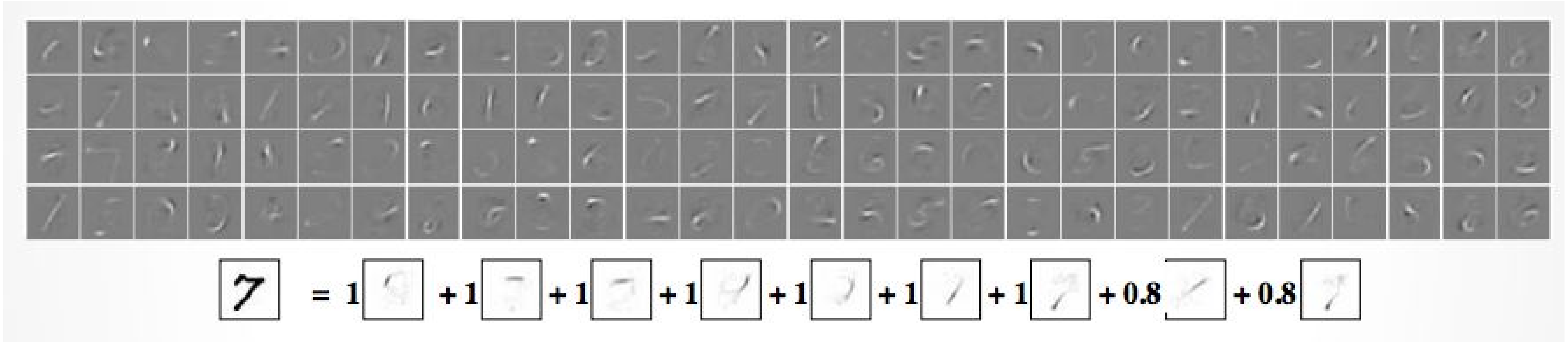
Use cases

Semantic Hashing

- Determining the subject of a document.
- Determining the relation two documents have.
- Creates many dimensional feature vectors for each document
- Allows document similarity to be assessed by computing the cosine of the arch between the feature vectors.
- A set of documents similar to a document can be produced in constant time (in regard to the number of documents) simply by determining the feature vector of the document.

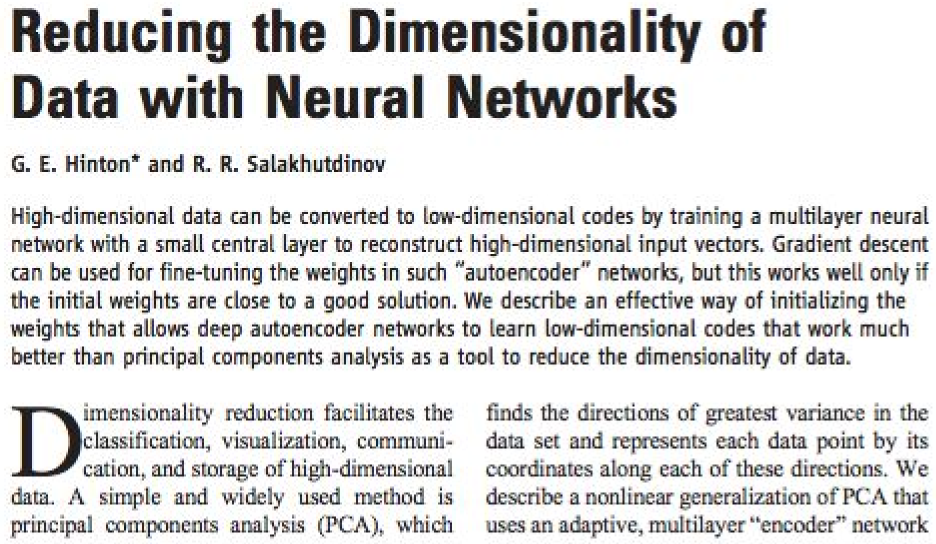
Image autoencoding

- MNIST dataset
- Learned feature representations
- High quality “natural” reproduction



Important Papers

Reducing the Dimensionality of Data with Neural Networks



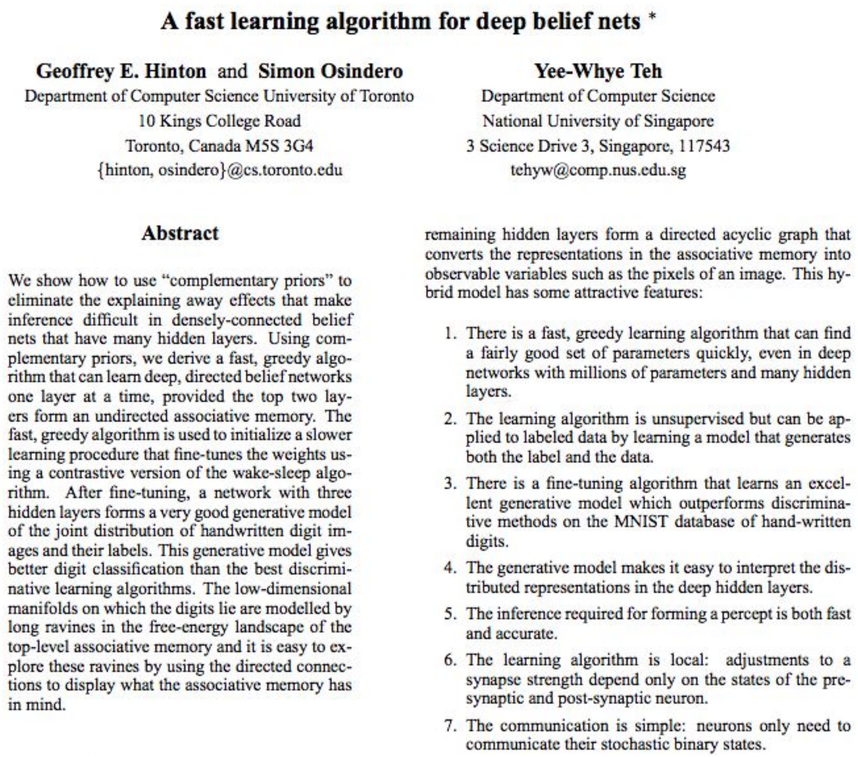
Reducing the Dimensionality of Data with Neural Networks

G. E. Hinton* and R. R. Salakhutdinov

High-dimensional data can be converted to low-dimensional codes by training a multilayer neural network with a small central layer to reconstruct high-dimensional input vectors. Gradient descent can be used for fine-tuning the weights in such "autoencoder" networks, but this works well only if the initial weights are close to a good solution. We describe an effective way of initializing the weights that allows deep autoencoder networks to learn low-dimensional codes that work much better than principal components analysis as a tool to reduce the dimensionality of data.

Dimensionality reduction facilitates the classification, visualization, communication, and storage of high-dimensional data. A simple and widely used method is principal components analysis (PCA), which finds the directions of greatest variance in the data set and represents each data point by its coordinates along each of these directions. We describe a nonlinear generalization of PCA that uses an adaptive, multilayer "encoder" network

A fast learning algorithm for deep belief nets



A fast learning algorithm for deep belief nets *

Geoffrey E. Hinton and **Simon Osindero**
Department of Computer Science University of Toronto
10 Kings College Road
Toronto, Canada M5S 3G4
{hinton, osindero}@cs.toronto.edu

Yee-Whye Teh
Department of Computer Science
National University of Singapore
3 Science Drive 3, Singapore, 117543
tehyw@comp.nus.edu.sg

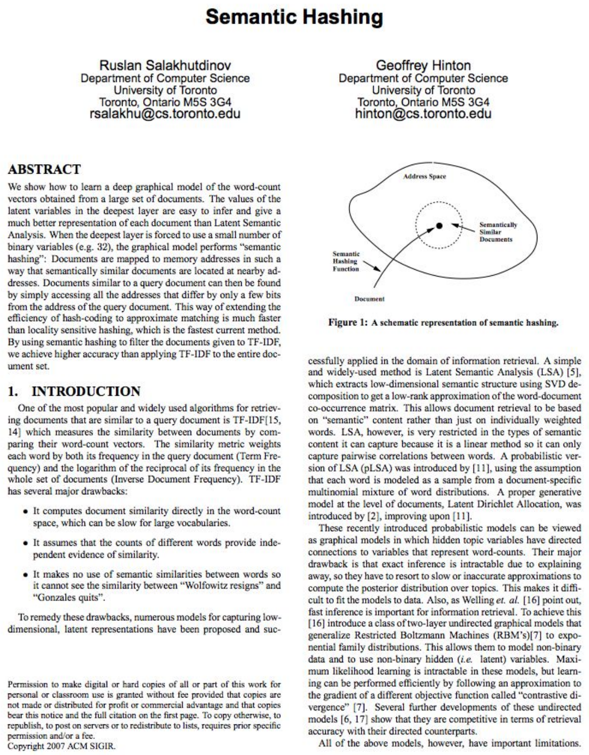
Abstract

We show how to use "complementary priors" to eliminate the explaining away effects that make inference difficult in densely-connected belief nets that have many hidden layers. Using complementary priors, we derive a fast, greedy algorithm that can learn deep, directed belief networks one layer at a time, provided the top two layers form an undirected associative memory. The fast, greedy algorithm is used to initialize a slower learning procedure that fine-tunes the weights using a contrastive version of the wake-sleep algorithm. After fine-tuning, a network with three hidden layers forms a very good generative model of the joint distribution of handwritten digit images and their labels. This generative model gives better digit classification than the best discriminative learning algorithms. The low-dimensional manifolds on which the digits lie are modelled by long ravines in the free-energy landscape of the top-level associative memory and it is easy to explore these ravines by using the directed connections to display what the associative memory has in mind.

remaining hidden layers form a directed acyclic graph that converts the representations in the associative memory into observable variables such as the pixels of an image. This hybrid model has some attractive features:

1. There is a fast, greedy learning algorithm that can find a fairly good set of parameters quickly, even in deep networks with millions of parameters and many hidden layers.
2. The learning algorithm is unsupervised but can be applied to labeled data by learning a model that generates both the label and the data.
3. There is a fine-tuning algorithm that learns an excellent generative model which outperforms discriminative methods on the MNIST database of hand-written digits.
4. The generative model makes it easy to interpret the distributed representations in the deep hidden layers.
5. The inference required for forming a percept is both fast and accurate.
6. The learning algorithm is local: adjustments to a synapse strength depend only on the states of the pre-synaptic and post-synaptic neuron.
7. The communication is simple: neurons only need to communicate their stochastic binary states.

Semantic hashing



Semantic Hashing

Ruslan Salakhutdinov
Department of Computer Science
University of Toronto
Toronto, Ontario M5S 3G4
rsalakhu@cs.toronto.edu

Geoffrey Hinton
Department of Computer Science
University of Toronto
Toronto, Ontario M5S 3G4
hinton@cs.toronto.edu

ABSTRACT

We show how to learn a deep graphical model of the word-count vectors obtained from a large set of documents. The values of the latent variables in the deepest layer are easy to infer and give a much better representation of each document than Latent Semantic Analysis. When the deepest layer is forced to use a small number of binary variables (e.g. 32), the graphical model performs "semantic hashing": Documents are mapped to memory addresses in such a way that semantically similar documents are located at nearby addresses. Documents similar to a query document can then be found by simply accessing all the addresses that differ by only a few bits from the address of the query document. This way of extending the efficiency of hash-coding to approximate matching is much faster than locality sensitive hashing, which is the fastest current method. By using semantic hashing to filter the documents given to TF-IDF, we achieve higher accuracy than applying TF-IDF to the entire document set.

1. INTRODUCTION

One of the most popular and widely used algorithms for retrieving documents that are similar to a query document is TF-IDF[15, 14] which measures the similarity between documents by comparing their word-count vectors. The similarity metric weights each word by both its frequency in the query document (Term Frequency) and the logarithm of the reciprocal of its frequency in the whole set of documents (Inverse Document Frequency). TF-IDF has several major drawbacks:

- It computes document similarity directly in the word-count space, which can be slow for large vocabularies.
- It assumes that the counts of different words provide independent evidence of similarity.
- It makes no use of semantic similarities between words so it cannot see the similarity between "Wolfowitz resigns" and "Gonzales quits".

To remedy these drawbacks, numerous models for capturing low-dimensional, latent representations have been proposed and successfully applied in the domain of information retrieval. A simple and widely-used method is Latent Semantic Analysis (LSA) [5], which extracts low-dimensional semantic structure using SVD decomposition to get a low-rank approximation of the word-document co-occurrence matrix. This allows document retrieval to be based on "semantic" content rather than just on individually weighted words. LSA, however, is very restricted in the types of semantic content it can capture because it is a linear method so it can only capture pairwise correlations between words. A probabilistic version of LSA (pLSA) was introduced by [11], using the assumption that each word is modeled as a sample from a document-specific multinomial mixture of word distributions. A proper generative model at the level of documents, Latent Dirichlet Allocation, was introduced by [2], improving upon [11].

These recently introduced probabilistic models can be viewed as graphical models in which hidden topic variables have directed connections to variables that represent word-counts. Their major drawback is that exact inference is intractable due to explaining away, so they have to resort to slow or inaccurate approximations to compute the posterior distribution over topics. This makes it difficult to fit the models to data. Also, as Welling *et. al.* [16] point out, fast inference is important for information retrieval. To achieve this [16] introduce a class of two-layer undirected graphical models that generalize Restricted Boltzmann Machines (RBM's)[7] to exponential family distributions. This allows them to model non-binary data and to use non-binary hidden (*i.e.* latent) variables. Maximum likelihood learning is intractable in these models, but learning can be performed efficiently by following an approximation to the gradient of a different objective function called "contrastive divergence" [7]. Several further developments of these undirected models [6, 17] show that they are competitive in terms of retrieval accuracy with their directed counterparts.

All of the above models, however, have important limitations.

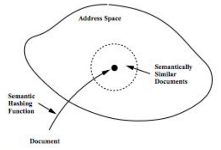
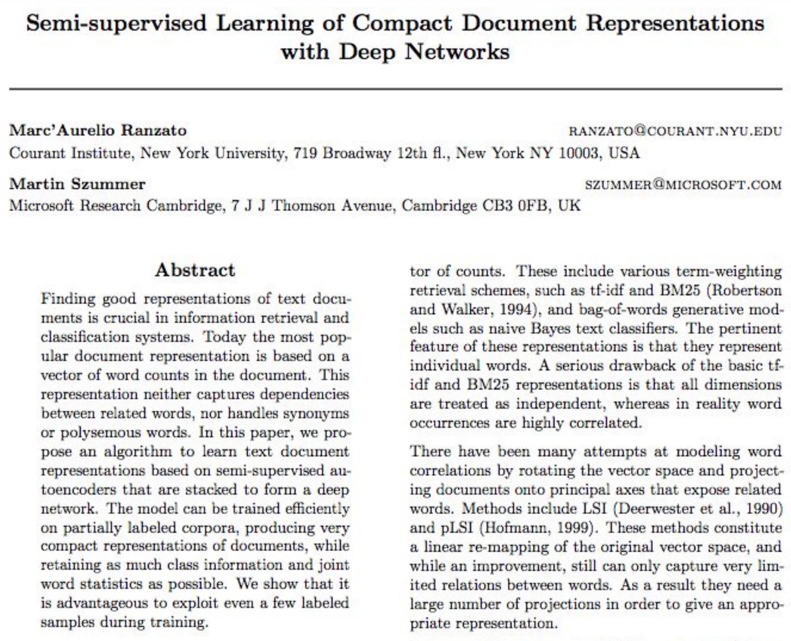


Figure 1: A schematic representation of semantic hashing.

Semi-supervised Learning of Compact Document Representations with deep Networks



Semi-supervised Learning of Compact Document Representations with Deep Networks

Marc'Aurelio Ranzato RANZATO@COURANT.NYU.EDU
Courant Institute, New York University, 719 Broadway 12th fl., New York NY 10003, USA

Martin Szummer SZUMMER@MICROSOFT.COM
Microsoft Research Cambridge, 7 J J Thomson Avenue, Cambridge CB3 0FB, UK

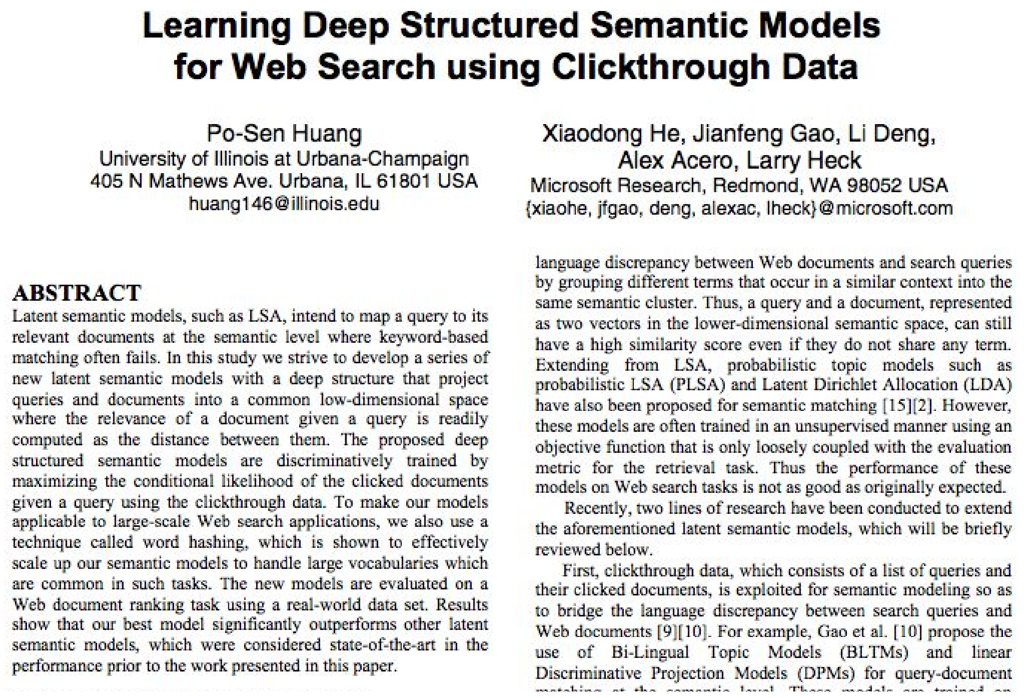
Abstract

Finding good representations of text documents is crucial in information retrieval and classification systems. Today the most popular document representation is based on a vector of word counts in the document. This representation neither captures dependencies between related words, nor handles synonyms or polysemous words. In this paper, we propose an algorithm to learn text document representations based on semi-supervised autoencoders that are stacked to form a deep network. The model can be trained efficiently on partially labeled corpora, producing very compact representations of documents, while retaining as much class information and joint word statistics as possible. We show that it is advantageous to exploit even a few labeled samples during training.

tor of counts. These include various term-weighting retrieval schemes, such as tf-idf and BM25 (Robertson and Walker, 1994), and bag-of-words generative models such as naive Bayes text classifiers. The pertinent feature of these representations is that they represent individual words. A serious drawback of the basic tf-idf and BM25 representations is that all dimensions are treated as independent, whereas in reality word occurrences are highly correlated.

There have been many attempts at modeling word correlations by rotating the vector space and projecting documents onto principal axes that expose related words. Methods include LSI (Deerwester et al., 1990) and pLSI (Hofmann, 1999). These methods constitute a linear re-mapping of the original vector space, and while an improvement, still can only capture very limited relations between words. As a result they need a large number of projections in order to give an appropriate representation.

Learning Deep Structured Semantic Models for Web Search using Clickthrough Data



Learning Deep Structured Semantic Models for Web Search using Clickthrough Data

Po-Sen Huang
University of Illinois at Urbana-Champaign
405 N Mathews Ave. Urbana, IL 61801 USA
huang146@illinois.edu

Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, Larry Heck
Microsoft Research, Redmond, WA 98052 USA
{xiaohe, jfgao, deng, alexac, lheck}@microsoft.com

ABSTRACT

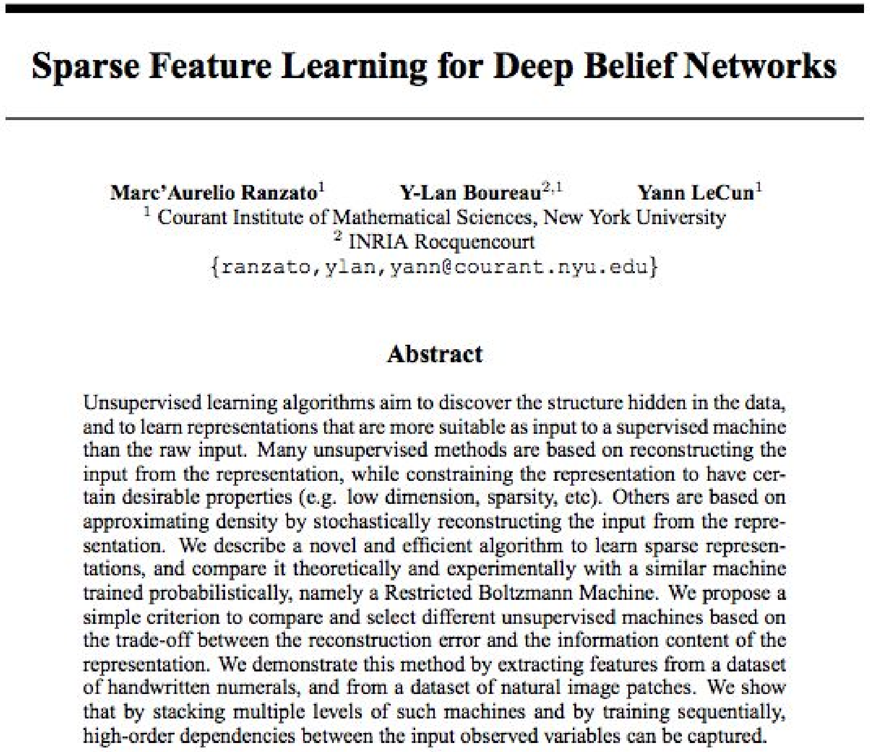
Latent semantic models, such as LSA, intend to map a query to its relevant documents at the semantic level where keyword-based matching often fails. In this study we strive to develop a series of new latent semantic models with a deep structure that project queries and documents into a common low-dimensional space where the relevance of a document given a query is readily computed as the distance between them. The proposed deep structured semantic models are discriminatively trained by maximizing the conditional likelihood of the clicked documents given a query using the clickthrough data. To make our models applicable to large-scale Web search applications, we also use a technique called word hashing, which is shown to effectively scale up our semantic models to handle large vocabularies which are common in such tasks. The new models are evaluated on a Web document ranking task using a real-world data set. Results show that our best model significantly outperforms other latent semantic models, which were considered state-of-the-art in the performance prior to the work presented in this paper.

language discrepancy between Web documents and search queries by grouping different terms that occur in a similar context into the same semantic cluster. Thus, a query and a document, represented as two vectors in the lower-dimensional semantic space, can still have a high similarity score even if they do not share any term. Extending from LSA, probabilistic topic models such as probabilistic LSA (PLSA) and Latent Dirichlet Allocation (LDA) have also been proposed for semantic matching [15][2]. However, these models are often trained in an unsupervised manner using an objective function that is only loosely coupled with the evaluation metric for the retrieval task. Thus the performance of these models on Web search tasks is not as good as originally expected.

Recently, two lines of research have been conducted to extend the aforementioned latent semantic models, which will be briefly reviewed below.

First, clickthrough data, which consists of a list of queries and their clicked documents, is exploited for semantic modeling so as to bridge the language discrepancy between search queries and Web documents [9][10]. For example, Gao et al. [10] propose the use of Bi-Lingual Topic Models (BLTMs) and linear Discriminative Projection Models (DPMs) for query-document

Sparse Feature Learning for Deep Belief Networks



Sparse Feature Learning for Deep Belief Networks

Marc'Aurelio Ranzato[1] **Y-Lan Boureau**[2,1] **Yann LeCun**[1]
[1] Courant Institute of Mathematical Sciences, New York University
[2] INRIA Rocquencourt
{ranzato,ylan,yann@courant.nyu.edu}

Abstract

Unsupervised learning algorithms aim to discover the structure hidden in the data, and to learn representations that are more suitable as input to a supervised machine than the raw input. Many unsupervised methods are based on reconstructing the input from the representation, while constraining the representation to have certain desirable properties (e.g. low dimension, sparsity, etc). Others are based on approximating density by stochastically reconstructing the input from the representation. We describe a novel and efficient algorithm to learn sparse representations, and compare it theoretically and experimentally with a similar machine trained probabilistically, namely a Restricted Boltzmann Machine. We propose a simple criterion to compare and select different unsupervised machines based on the trade-off between the reconstruction error and the information content of the representation. We demonstrate this method by extracting features from a dataset of handwritten numerals, and from a dataset of natural image patches. We show that by stacking multiple levels of such machines and by training sequentially, high-order dependencies between the input observed variables can be captured.

Sources

- Totorial on auto encoders: https://piotrmirowski.wordpress.com/2014/03/27/tutorial-on-auto-encoders/
- All the papers in the previous slides
- Deeplearning4j.org
- http://www.jmlr.org/papers/volume11/vincent10a/vincent10a.pdf
- int8.io