# Learning Deep Architectures (part I)

Enrique Romero

Advanced Topics in Artificial Intelligence
Master in Artificial Intelligence
Soft Computing Group
Departament de Llenguatges i Sistemes Informàtics
Universitat Politècnica de Catalunya, Barcelona, Spain

2017-2018
2012-2013 (first course)

# Outline

# Outline

# Outline

# NVIDIA, GPUs and Deep Learning (2014)

(03-2014) NVIDIA uses Deep Learning results (instead of video games, as they usually do) in the presentation of a GPU of 8 teraflops

`https://www.youtube.com/watch?v=37Yt41ouaNM&list=UUHuiy8bXnmK5nisYHUd1J5g`

# Outline

# A Couple of Examples...

- (06-2015) Deep Learning Machine Beats Humans in IQ Test: https://www.technologyreview.com/s/538431/deep-learning-machine-beats-humans-in-iq-test/
- (06-2015) Inceptionism: Engineers from Google extract information of the different layers of a deep neural network to obtain new images: http://googleresearch.blogspot.com.es/2015/06/inceptionism-going-deeper-into-neural.html
- (01-2016) Playing "go": http://www.nature.com/nature/journal/v529/n7587/full/nature16961.html
- (02-2016) Traning drones to autonomously navigate on previously-unseen trails in a densely wooded forest: https://news.developer.nvidia.com/autonomous-search-and-rescue-drones-outperform-humans-at-navigatin

# Outline

# Deep Learning in Big Companies

- (03-2013) Google acquires DNNresearch Inc (for an undisclosed sum of money), and hires Geoffrey Hinton, Alex Krizhevsky and Ilya Sutskever

  `http://www.wired.com/wiredenterprise/2013/03/google_hinton/`
- (12-2013) Facebook hires Yann LeCun to head its new AI lab

  `http://techcrunch.com/2013/12/09/facebook-artificial-intelligence-lab-lecun`
- (01-2014) Google acquires AI company DeepMind

  `http://www.forbes.com/sites/amitchowdhry/2014/01/27/google-to-acquire-artificial-intelligence-company-deepmind/`
- (05-2014) Baidu (aka China's Google) hires Andrew Ng

  `http://www.technologyreview.com/news/527301/chinese-search-giant-baidu-hires-man-behind-the-google-brain`
- Other important companies: Microsoft, Apple, IBM, Amazon

# Outline

# Deep Learning in the Media

- (03-2014) Facebook uses Deep Learning to recognize faces
  `http://www.technologyreview.com/news/525586/facebook-creates-software-that-matches-faces-almost-as-well-as-you`
- (02-2015) Google Develops Computer Program Capable of Learning Tasks Independently
  `http://www.theguardian.com/technology/2015/feb/25/google-develops-computer-program-capable-of-learning-tasks-indepen`
- (12-2016) The Great A.I. Awakening (about Google Translate)
  `https://www.nytimes.com/2016/12/14/magazine/the-great-ai-awakening.html`
- And many more...

# Outline

# Outline

# References

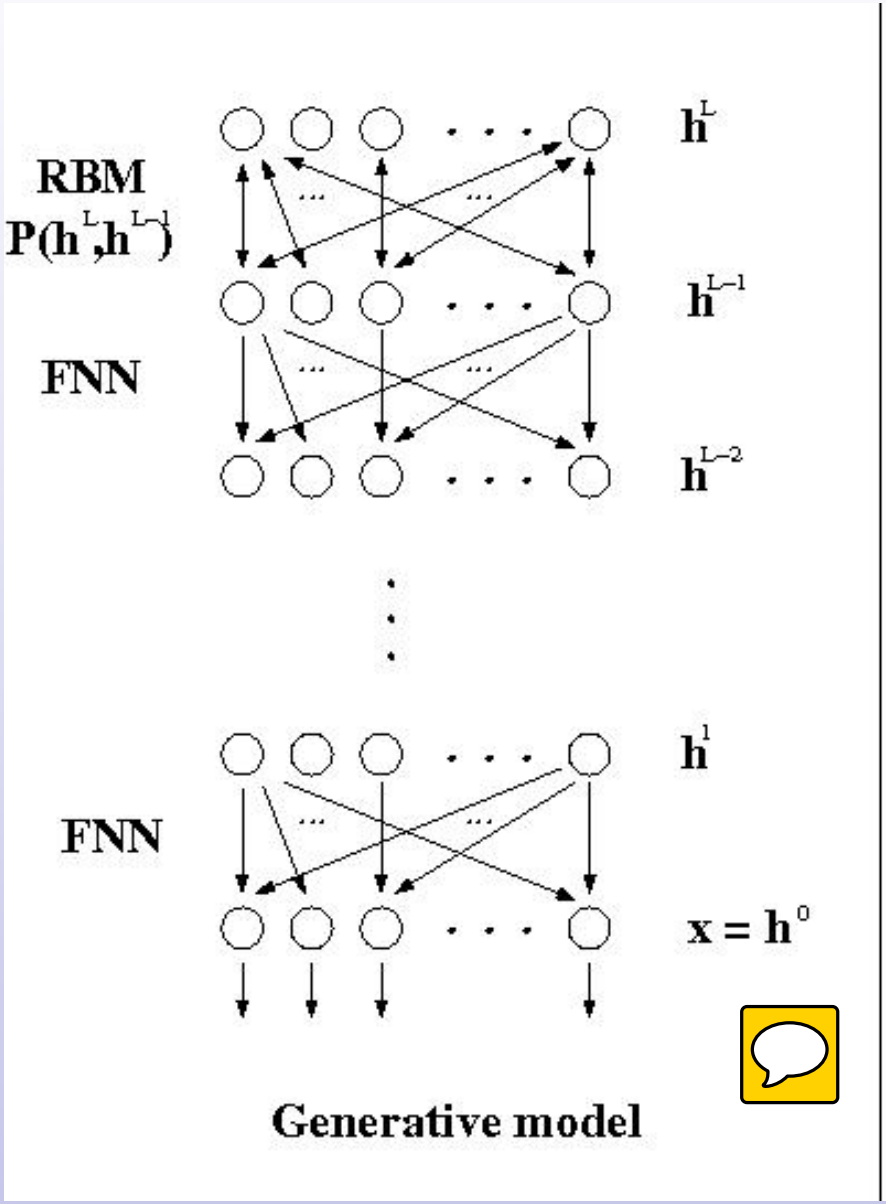
[Bengio, 2009]: Bengio, Y. (2009). Learning Deep Architectures for AI. *Foundations and Trends in Machine Learning*, 2(1):1–127.
www.iro.umontreal.ca/~bengioy/papers/ftml_book.pdf

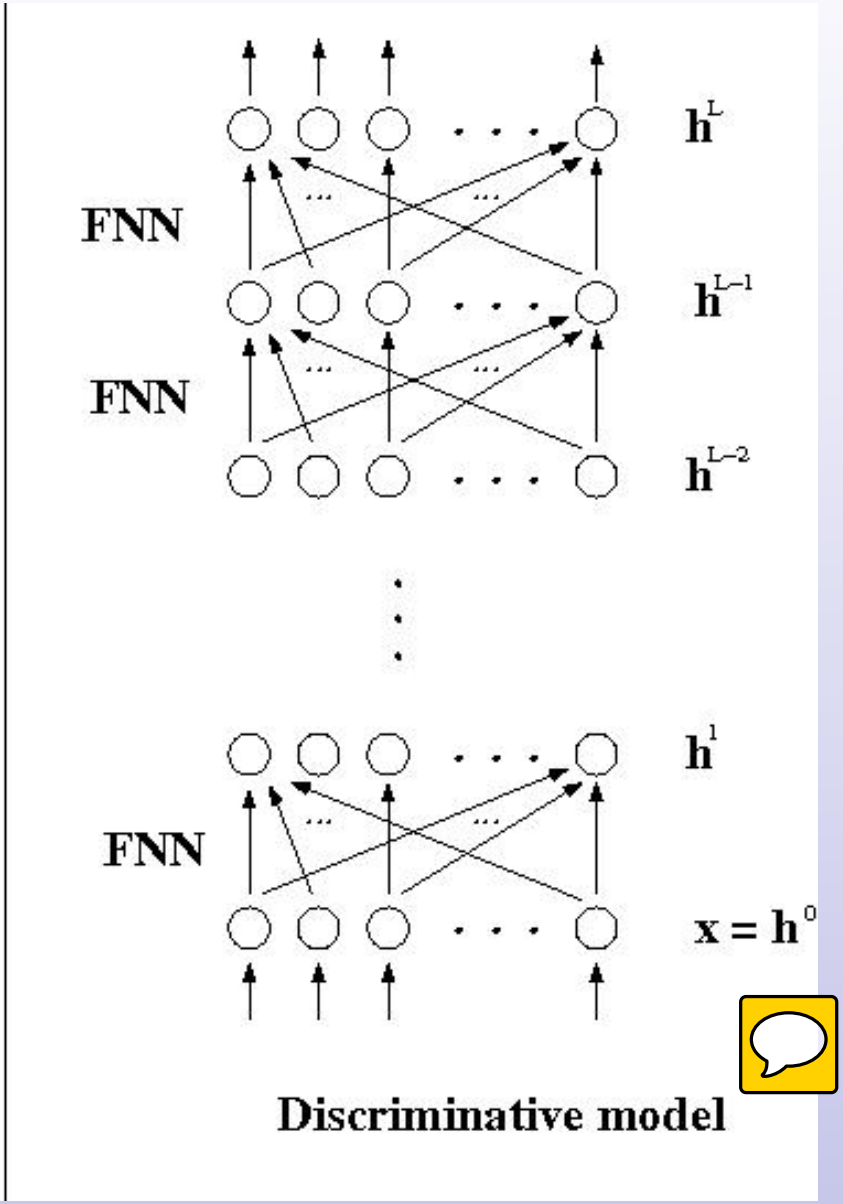
Other interesting references:

- A recent (2016) Deep Learning book
  http://www.deeplearningbook.org/
- Deep Learning talk (2015) by Geoffrey E. Hinton
  https://www.youtube.com/watch?v=IcOMKXAw5VA
- Deep Learning in a Nutshell: History and Training
  https://devblogs.nvidia.com/parallelforall/deep-learning-nutshell-history-training/
- Neural Network videolectures of Hugo Larochelle http://www.youtube.com/playlist?list=PL6Xpj9I5qXYEcOhn7TqghAJ6NAPrNmUBH

# Outline

# Some Examples of Deep Architectures



# Some Examples of Deep Architectures

Remarks:

- Both architectures have $L$ hidden layers of units and weights
- Architecture in the right side
  - Is discriminative
  - Is a classical MLP: The computation starts in the lowest layer ($\mathbf{x}$), and propagates upwards
  - The output is $\mathbf{h}^L$
- Architecture in the left side
  - Is generative
  - The computation starts in the deepest layer (a Restricted Boltzmann Machine) that computes $P(\mathbf{h}^L, \mathbf{h}^{L-1})$ and can sample $\mathbf{h}^L$ (**SYMMETRIC CONNECTIONS**)
  - The rest of the layers propagate the activations downwards as standard MLPs
  - The output is $\mathbf{x}$

# Outline

# Definition of Deep Architectures

Deep architectures are composed of **MULTIPLE LEVELS OF NON-LINEAR OPERATIONS**, such as neural networks with many hidden layers or complicated propositional formulae re-using many sub-formulae

We will focus on **NEURAL NETWORKS WITH MANY HIDDEN LAYERS** (that **MAY WORK IN A "NON-STANDARD" WAY**)

**RECALL THAT MOST CURRENT MACHINE LEARNING ALGORITHMS CORRESPOND TO SHALLOW ARCHITECTURES (1, 2 OR 3 LEVELS)**: k-Nearest Neighbors, LDA, Decision Trees, Gaussian Processes, Kernel Machines,...

# Definition of Deep Architectures

The **depth of the architecture** refers to the number of levels of composition of non-linear operations in the function learned

From now on, we will use the terms *DEEP ARCHITECTURE* and *DEEP NEURAL NETWORK* **without distinction**, using *NEURAL NETWORK* as a synonymous of *MULTI-LAYER PERCEPTRONS* (*RADIAL BASIS FUNCTION NETWORKS* ARE NOT CONSIDERED)

# Outline

# The Challenge of Artificial Intelligence

Although much progress has been done in machine learning, the challenge of Artificial Intelligence remains. Several examples:

- Do we have algorithms that can discover visual concepts that seem necessary to interpret images (in the web, for example)?
- Do we have algorithms that can infer enough semantic concepts (from an image, for example) to be able to interact with humans?
- Do we have algorithms that can understand images/scenes and describe them in natural language?

The answer to these questions is "No" (in a general setting)

# Outline

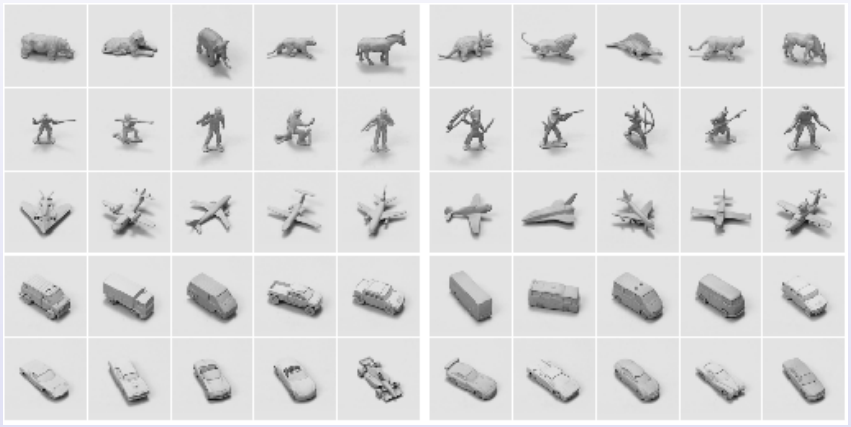
# Reality is Complex

Reality is complex (see figure 1): **we do not have an analytical understanding (formalized prior knowledge) of the world to explain the huge variety of different images of the same object** by varying position, orientation, lighting focus,...

For example, an apparently simple abstraction such as MAN

- Corresponds to a very large set of possible images,...
- ... which are very different from each other for most measures (Euclidean distance, for example),...
- ... and can be very close to other abstractions (see figure 2)

# Reality is Complex



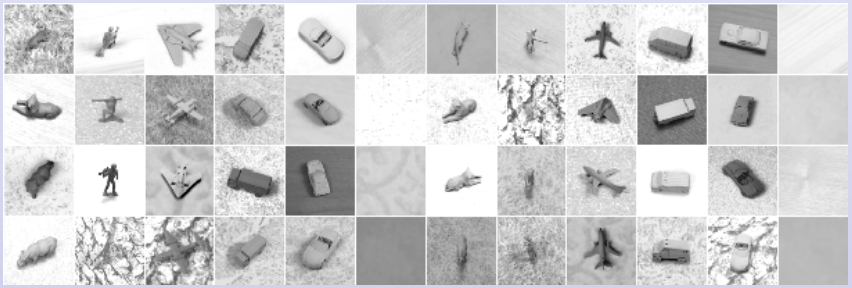


Figure 1 : Pixel intensities change (even in BW images) depending of many factors of variation/variance; Images from `http://www.cs.nyu.edu/~yann/research/norb/`

# Reality is Complex

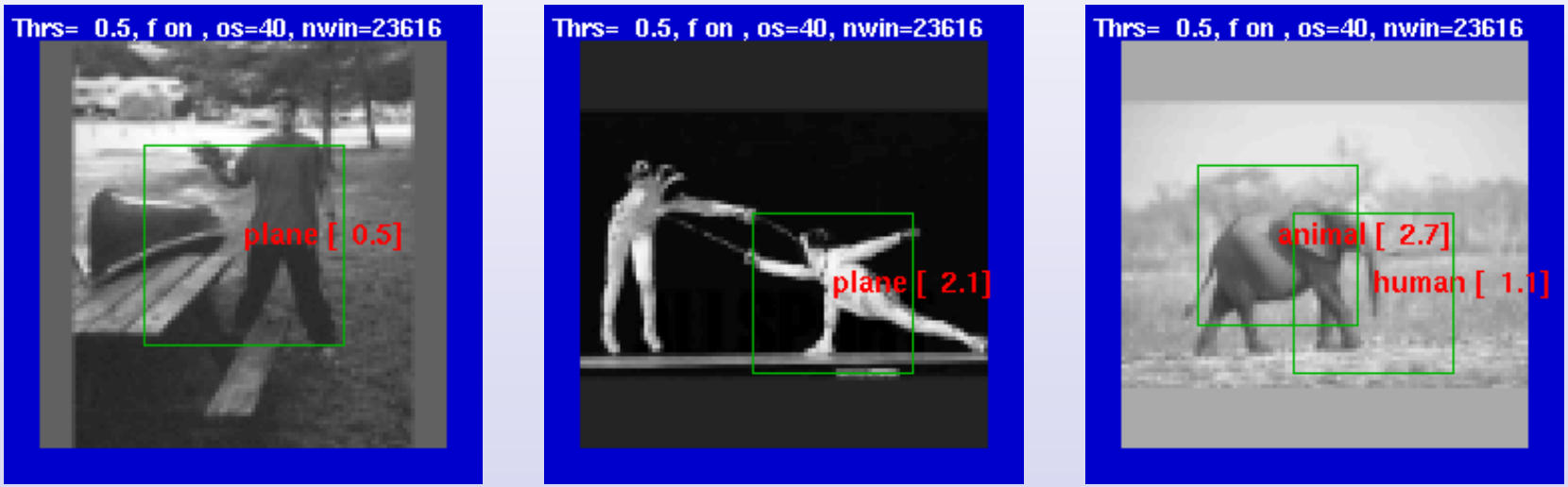



Figure 2 : Similar images of different abstractions; Images from `http://www.cs.nyu.edu/~yann/research/norb/`

# Reality is Complex

More examples of complex reality:

- If there are more than one object in the image, the problem is even more difficult
- Video data: for example, trying to detect an object moving in a scene
- Speech recognition
- Natural Language processing: machine translation, information retrieval,...
- Many, many more

**The human brain performs these tasks in a natural way**

# Outline

# Desiderata for Learning in AI

**A central concept behind learning in AI is the understanding that USEFUL REPRESENTATIONS of data such as image, video and audio signals are usually ABSTRACT**:

- Abstract features represent the semantic content of the data, divorced from the low-level features of the raw data (e.g., pixels, voxels, or waveforms)

In general, abstract representations are **INVARIANT to most LOCAL CHANGES of the input** (for example, the semantic content of an image changes little if we move everything in the scene 10 pixels to the left, while the pixel-level representation may change dramatically)

# Desiderata for Learning in AI

**Therefore, a desiderata of machine learning is to learn a representation that preserves the task-relevant factors of variation (features) of the data while being invariant to irrelevant factors of variation**

Hypothesis: These seemingly conflicting goals forces the representations that capture these concepts generally to be **highly non-linear functions** of the raw input

Deep Learning achieves this kind of complex transformation of the data by **implementing multiple layers of representation**, where each layer is constructed as a simple (but nonlinear) transformation of the previous representation: **abstract representations are constructed in terms of less abstract ones**

# Desiderata for Learning in AI

**To make learning practical** we need

- Learn from a **very large set of examples** (otherwise, we will only will be able to construct toy problems, since there may be many factors of variation in real data)
- (Maybe) Learn from **unlabeled data** (unlabeled data is much easier to obtain than labeled data)
- Learn with **little human input**

**The state-of-the-art of learning with deep architectures (reasonably) satisfy all these requirements, and their success obtained in several difficult tasks suggest that they are good candidates to tackle these kind of problems**

# Outline

# A Brief History of Deep Learning

Inspired by the architectural depth of the brain, researchers had wanted for decades (extensively since 1985) to train deep multi-layer neural networks

However, no successful results were obtained before 2006 (except for Convolutional Networks, a very special architecture designed for vision tasks):

- Results were good with one or two hidden layers, but...
- Training deeper networks (starting from random weights with classical activation functions and classical optimization methods) consistently yielded worse results

# A Brief History of Deep Learning

In 2006, Geoffrey E. Hinton et al. introduced **Deep Belief Networks** [Hinton et al., 2006] for generative learning

- The learning algorithm **trains one layer at a time in a greedy manner**: the output of an already trained layer is the input of the next layer
- Every layer is trained based on an unsupervised training algorithm for **Restricted Boltzmann Machines** (RBMs), a particular case of **Boltzmann Machines** [Ackley et al., 1985]
- The whole training is divided into two steps:
  - A greedy layer-wise pre-training (training of every layer)
  - A fine-tuning of the whole network after the pre-training

This model could be thought as an (improved) alternative to Sigmoid Belief Networks y and the Wake-Sleep algorithm [Neal, 1992, Hinton et al., 1995]

# A Brief History of Deep Learning

More or less at the same time, [Hinton and Salakhutdinov, 2006] used stacked RBMs with pre-training to construct **Deep Discriminative Neural Networks** and **Deep Auto-Encoders**:

- For deep discriminative neural networks:
  - A greedy layer-wise pre-training was performed with a deep architecture, based on RBMs
  - After the pre-training, a new layer is added, randomly initialized, and trained with back-propagation in a standard supervised way
- For deep Auto-Encoders:
  - The data is first encoded with multiple layers, and pre-trained with the algorithm for RBMs in every layer
  - After the pre-training, the model is "unfolded" to obtain a decoder that initially uses the same weights
  - The whole Auto-Encoder is then fine-tuned with back-propagation to minimize the reconstruction error

# A Brief History of Deep Learning

Shortly after, other algorithms exploiting the same principle (*train intermediate levels of representation with unsupervised learning*) were proposed, based on **Auto-Encoders**:
[Bengio et al., 2007b, Ranzato et al., 2007]

Subsequently, other algorithms not based neither on RBMs nor Auto-Encoders but exploiting the same ideas were proposed:
[Weston et al., 2008, Mobahi et al., 2009]

# A Brief History of Deep Learning

The "official" version of what was wrong with Back-propagation (a plausible story, but false in Hinton's words):

- It requires labeled training data, and there are not enough labeled data (almost all data are unlabeled)
- The learning time does not scale well (it is very slow in networks with multiple hidden layers)
- It can get stuck in poor local minima (often quite good for shallow networks, but far from optimal for deep networks)

What has happened since 1985? Hinton's version:

- Labeled data sets are much larger, and **back-propagation outstands whih a large number of labeled examples**
- **Computers are much faster** (learning times of deep networks have been reduced)
- There have been found **good ways to train the weights of large deep networks** (avoiding poor local minima)

# Outline

# The Main Ideas of Deep Learning

The main idea of deep learning methods is **LEARNING FEATURE HIERARCHIES USEFUL FOR THE PROBLEM AT HAND**:

- Features from higher levels in the hierarchy are constructed on the basis of features in lower levels (**DEEP**)
- Features are learned, without human interaction (**LEARNING**)

In the ideal case:

- Different levels of features are related with **different levels of abstraction**, related to human tasks that we do not know how to describe formally
- A deep hierarchy of features allow the system to **learn very complex functions**

# Different Levels of Abstraction

Deep architectures are **able to LEARN DISTRIBUTED REPRESENTATIONS**

The objective of Deep Learning is **to LET THE LEARNING ALGORITHM DISCOVER THE FEATURES that compose those distributed representations**

Therefore, training deep architectures can be seen as **LEARNING TO TRANSFORM ONE REPRESENTATION (the output of the previous stage) INTO ANOTHER**

**HIGHER LEVELS ARE SUPPOSED TO BE MORE ABSTRACT and represent more complex features**

# Outline

# Outline

# Results on the MNIST Data Set

The MNIST data set:

- Database of handwritten digits
- An example is an image (28x28 grey levels) and a label (0-9)
- Training/Test data: 60,000/10,000 examples
- Task: Classification

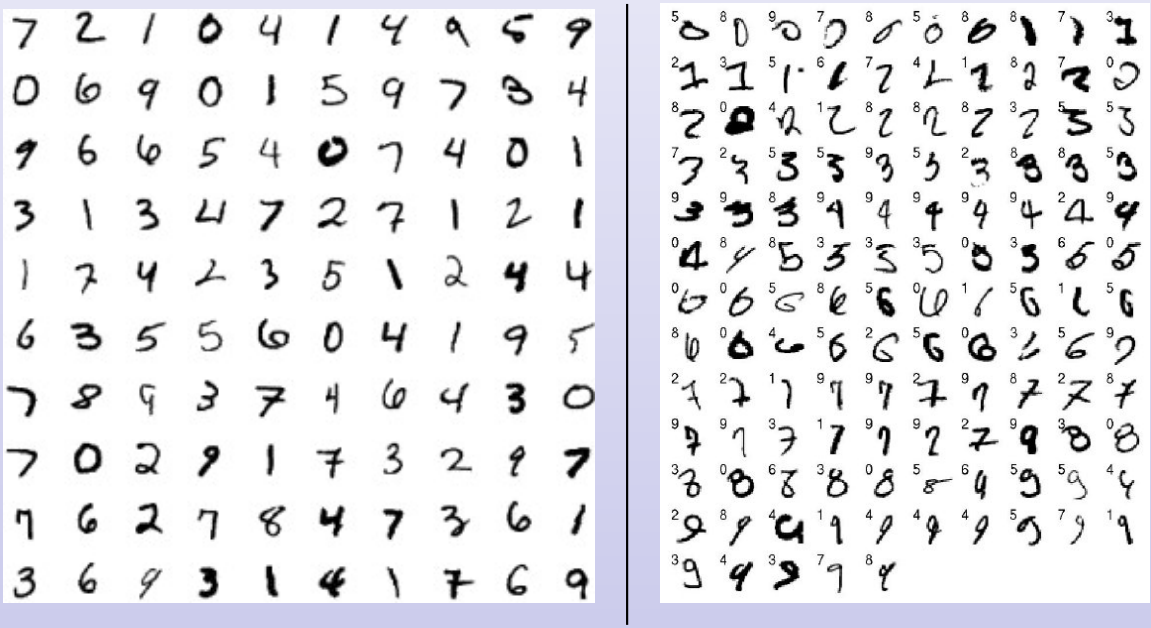


Figure 3 : MNIST examples: Random/difficult (left/right) examples

# Results on the MNIST Data Set

In 2006, the best results (test error) on the MNIST data set (permutation-invariant task: no preprocessing or enhancement) were

- 2.8% for Nearest Neighbor classifiers
- 1.4% for Support Vector Machines

**The generative Deep Belief Networks proposed in [Hinton et al., 2006] (a 784-500-500-2000 unsupervised architecture plus 10 unsupervised-supervised units) obtained 1.25% error rate**

**The Deep Discriminative Neural Networks proposed in [Hinton and Salakhutdinov, 2006] (a 784-500-500-2000-10 architecture) obtained 1.20% error rate**

Nowadays, these results are outperformed (by other deep models)

# Outline

# Results on the Reuters Data Set

The Reuters data set:

- Database of documents (news)
- Every example is a document and a label (a topic)
- Documents were preprocessed to obtain their 2,000 more common words, that were the input to the system (a vector of frequencies)
- Number of labels: 103
- Training data: 402,207 examples (100,000 for validation)
- Test data: 402,207 examples
- Task: Document Retrieval (find similar documents)

A well-known document retrieval method is Latent Semantic Analysis (LSA), a method based on PCA

# Results on the Reuters Data Set

An Auto-Encoder based on Restricted Boltzmann Machines was constructed, with architecture 2000-500-250-125-10 [Hinton and Salakhutdinov, 2006]:

- The multiclass cross-entropy error function $-\sum_i p_i log\ \widetilde{p}_i$ was used for the fine-tuning
- The 10 code units were linear and the remaining hidden units were logistic

**When the cosine of the angle between two codes was used to measure similarity, the auto-encoder clearly outperformed LSA**

# Results on the Reuters Data Set

The codes produced by the two-dimensional LSA and the **2000-500-250-125-2 Deep Auto-Encoder proposed in [Hinton and Salakhutdinov, 2006]** were

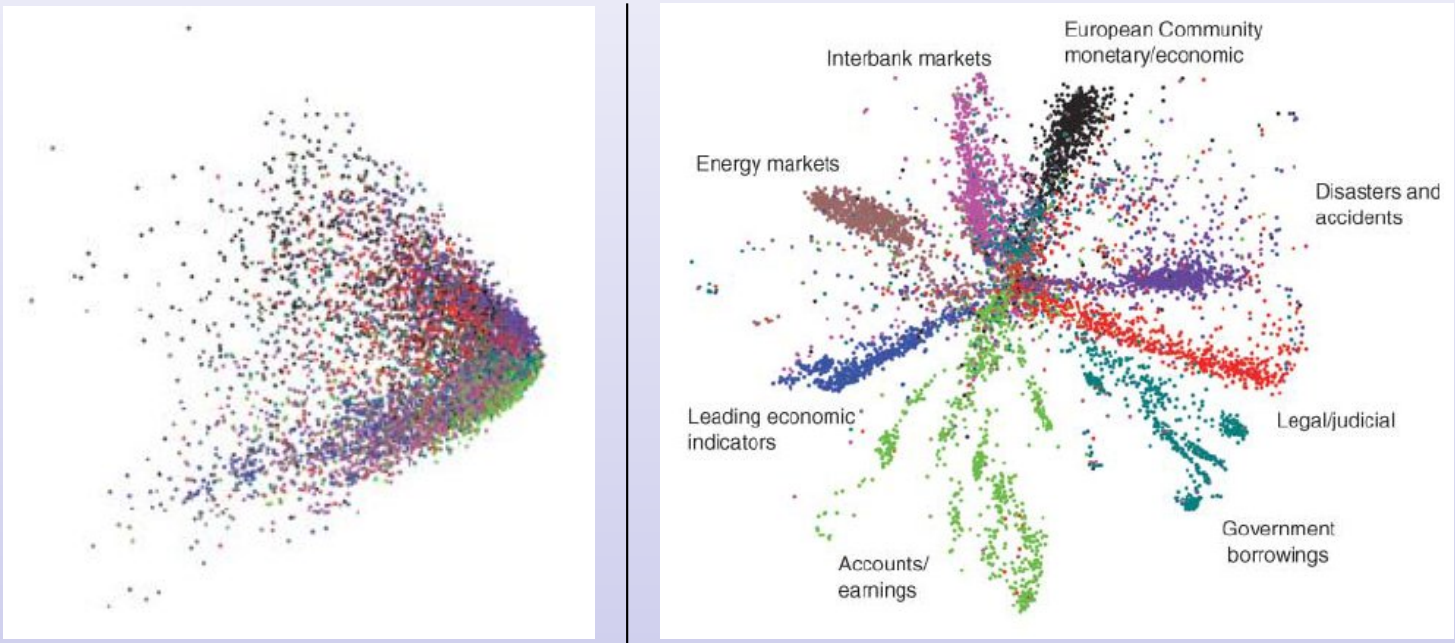



Figure 4 : Codes produced by LSA (left) and the auto-encoder proposed in [Hinton and Salakhutdinov, 2006] (right) for the Reuters database

# Outline

# Results on the TIMIT Data Set

TIMIT: Texas Instruments (TI) and Massachusetts Institute of Technology (MIT) is the database most widely used by the Phone Recognition research community, mainly because it is totally and manually annotated at the phone level

A **phone** is an atomic sound of a language: it is the smallest identifiable unit found in a stream of speech that is able to be transcribed with an IPA symbol

The task of Phone Recognition is related to automatic speech recognition, keyword detection, language recognition, speaker identification, etc

# Results on the TIMIT Data Set

The TIMIT data set:

- Database of recordings of English speech
- 6300 sentences: 10 sentences from 630 speakers of different sexes from 8 major dialect regions of the US (5.4 hours)
- All sentences were manually segmented at the phone level (time-aligned orthographic, phonetic and word transcriptions)
- Training data: 4620 sentences, but usually only 3696 sentences are used (462 speakers)
- Test data: 1344 sentences from 168 speakers
- Core test data: 192 sentences, 8 from each of 24 speakers (2 males and 1 female from each dialect region)
- Task: Phone Recognition

# Results on the TIMIT Data Set

In 2012, the best results (test error) on the TIMIT data set were

- 24.4%, for Heterogeneous Classifiers (1998)
- 22.7%, for Triphone HMMs discriminatively trained (2009)

**The model proposed by Mohamed, Dahl and Hinton [Mohamed et al., 2012] obtained an error rate of 20.70% (a quite large difference in this data set)**

Li Deng, Principal Researcher at Microsoft, realized that it could be a new way to tackle the speech recognition problem (see BBC News: `http://www.bbc.co.uk/news/technology-20266427`)

**Similar outstanding results have been obtained in other speech recognition data sets [Hinton et al., 2012a]**

# Outline

# Unsupervised and Transfer Learning Challenge 2011

The goals of this competition (`http://www.causality.inf.ethz.ch/unsupervised-learning.php`) are

- The assessment of data representations produced by unsupervised learning procedures, for use in supervised learning tasks
- The evaluation of transfer learning methods capable of producing data representations useful across many similar supervised learning tasks, after training from only one of them

# Unsupervised and Transfer Learning Challenge 2011

Large datasets from various application domains:

- Handwriting recognition (AVICENNA dataset):
  - Training: 150,205 (develop.) and 50,000 (transfer) examples
  - Task: Handwriting recognition
- Video processing (HARRY dataset):
  - Training: 69,652 (develop.) and 20,000 (transfer) examples
  - Task: Recognize human actions like hand clapping, picking up a phone, walking, running, driving a car, etc
- Image recognition (RITA dataset):
  - Training: 111,808 (develop.) and 24,000 (transfer) examples
  - Task: Object recognition
- Ecology (SYLVESTER dataset):
  - Training: 572,820 (develop.) and 100,000 (transfer) examples
  - Task: Classify forest cover types
- Text processing (TERRY dataset):
  - Training: 217,034 (develop.) and 40,000 (transfer) examples
  - Task: Text Categorization

# Unsupervised and Transfer Learning Challenge 2011

In a first phase of the challenge (unsupervised learning), the competitors were given only unlabeled data to learn their data representation

In a second (final) phase of the challenge (transfer learning), the competitors were also provided with a limited amount of labeled data from source tasks, distinct from the target tasks

**The team LISA ranked first in the second (final) phase and fourth in the first phase** [Guyon et al., 2011]

**The leader of the team LISA, composed of 18 members, was Yoshua Bengio (University of Montreal)**, and based their solution on Deep Learning techniques specific for unsupervised learning of representations, exploiting unsupervised learning of single-layer models as building blocks to construct deeper models

# Outline

# Large Scale Visual Recognition Challenge 2012

The goal of this competition (`http://www.image-net.org/challenges/LSVRC/2012/`) is to estimate the content of photographs using a subset of the ImageNet dataset, a larger hand-labeled dataset

The 2012 ImageNet challenge dataset:

- Database of images, from flickr and other search engines
- An example is an image (32x32 RGB)
- Number of labels: 1,000
- Training data: 1,200,000 images and their labels
- Validation data: 50,000 images and their labels
- Test data: 150,000 images
- Tasks: Classification, Classification with localization, Fine-grained classification on 100+ dog categories

# IMAGENET Challenge

**The team Supervision was the winner in two out of three tasks (they did not compete in the third one) with a very large difference**:
`http://www.image-net.org/challenges/LSVRC/2012/results.html`

**The team Supervision was composed of Alex Krizhevsky, Ilya Sutskever and Geoffrey Hinton (University of Toronto)**

The paper can be found at [Krizhevsky et al., 2012]

# IMAGENET Challenge

The description of the model was (in authors' words):

- Our model is a large, **deep convolutional neural network** trained on raw RGB pixel values
- The neural network, which has 60 million parameters and 650,000 neurons, consists of five convolutional layers, some of which are followed by max-pooling layers, and three globally-connected layers with a final 1000-way softmax
- To make training faster, we used **Rectified Linear units** and a **very efficient GPU implementation** of convolutional nets
- To reduce overfitting in the globally-connected layers we employed hidden-unit **dropout**, a recently-developed regularization method that proved to be very effective
- It was trained on two NVIDIA GPUs for about a week

# IMAGENET Challenge

Dropout [Hinton et al., 2012b, Srivastava et al., 2014] is a **regularization technique** based on:

- For every training case, **hidden units are randomly omitted from the network with probability $p$ (typically 0.5)**
- At test time, the "mean network" that contains all of the hidden units is used but **with their outgoing weights halved** to compensate for the fact that twice as many of them are active
- An interpretation of the dropout procedure is as a very efficient way of performing **model averaging with neural networks**: There is almost certainly a different network for each presentation of each training case but all of these networks share the same weights for the hidden units that are present

# Outline

# Google and Stanford Experiments

In 2012, Google (Ranzato, Monga, Devin, Chen and Corrado) and Standford (Le and Ng) researchers trained a deep neural network to learn class-specific feature detectors from only unlabeled data [Le et al., 2012]

The abstract says: "We consider the problem of building high-level, class-specific feature detectors from only unlabeled data. For example, is it possible to learn a face detector using only unlabeled images?"

# Google and Stanford Experiments

Training data: 10 million 200x200 pixel images downloaded from the Internet

Model: 9-layered locally connected sparse auto-encoder with pooling and local contrast normalization (the model has 1 billion connections)

Trained using parallelism and asynchronous stochastic gradient descent on a cluster with 1,000 machines (16,000 cores) for three days

# Google and Stanford Experiments

After training, the test set was used to measure the performance of each neuron in classifying faces against distractors:

- For each neuron, the maximum and minimum activation values were found, then picked 20 equally spaced thresholds in between
- The reported accuracy is the best classification accuracy among 20 thresholds

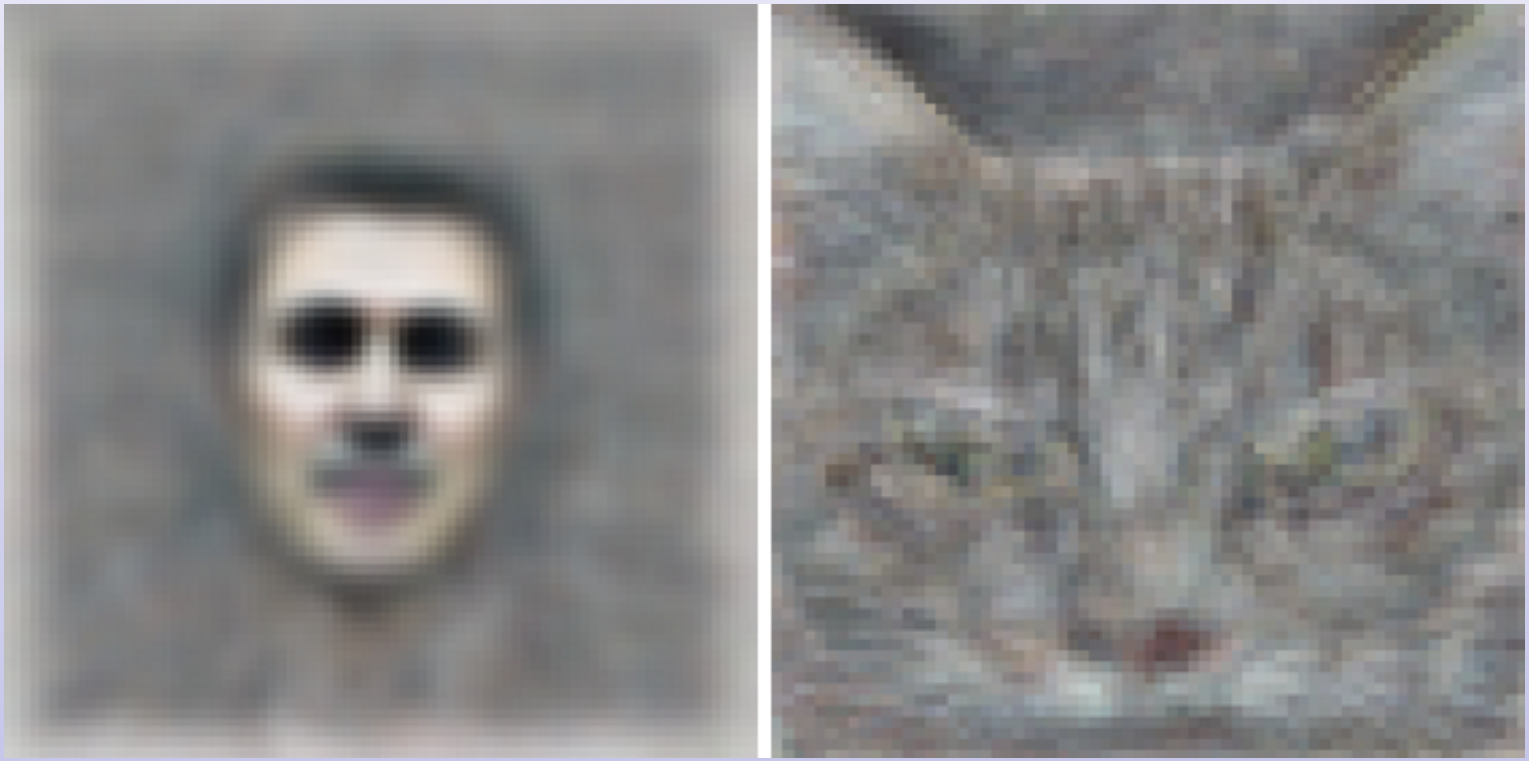
Surprisingly, the best neuron in the network performs very well in recognizing faces, even trained in a non-supervised way:

- The best neuron in the network achieves 81.7% accuracy in detecting faces
- The best neuron in a one-layered network achieves 71% accuracy
- The best linear filter, selected among 100,000 filters sampled randomly from the training set, achieves 74%

# Google and Stanford Experiments

The same network is sensitive to other high-level concepts such as cat faces and human bodies

The optimal stimulus (numerical constraint optimization) of the best neuron at recognizing human and cats faces were



# Outline

# Netflix Prize

Netflix Prize (`http://www.netflixprize.com`):

- Online DVD-rental service of Netflix
- The Netflix Prize was an open competition for the best algorithm to predict user ratings for films, based on previous ratings without any other information about the users or films
- Training data:
  - 100,480,507 ratings
  - 480,189 users
  - 17,770 movies
  - A training example is a quadruplet (user, movie, date, grade)
- Test data: 2,817,131 triplets of the form (user, movie, date)
- Task: Collaborative Filtering

# Netflix Prize

On 21 September 2009, the grand prize of US $1,000,000 was given to the BellKor's Pragmatic Chaos team, which improved by 10.06% Netflix's own algorithm (Cinematch) for predicting ratings

**The model of the winner team used (among other things) Restricted Boltzmann Machines, based on a paper of Salakhutdinov, Mnih and Hinton [Salakhutdinov et al., 2007]**

# Outline

# And Many More...

- Kaggle Competition: Merck Molecular Activity (`https://www.kaggle.com/c/MerckActivity`): **The team gggg, composed of Dahl, Salakhutdinov, Jaitly, Jordan-Squire and Hinton** was the winner of the competition (US $22,000) using **deep networks with Rectified Linear units trained with dropout (on a GPU)**
- Kaggle Competition: Job Salary Prediction (`https://www.kaggle.com/c/job-salary-prediction`): **The team lazylearner, composed of Volodymyr Mnih, a PhD student of Geoffrey Hinton**, was the winner of the competition (US $3,000) **using deep networks with Rectified Linear units trained with dropout (on a GPU)**
- Other tasks, such as modeling textures, modeling motion, object segmentation, information retrieval, robotics, natural language processing, etc

# Outline

# Outline

# Biological Motivation

- Humans often describe concepts in hierarchical ways, with multiple levels of abstraction
- The brain appears to process information through multiple stages of transformation and representation, as in the primate visual system, with a sequence of processing stages:
  - Detection of edges
  - Primitive shapes: lines, corners,...
  - ... moving up to gradually mode complex visual shapes

# Outline

# Theoretical Motivation

We say that the expression of a function is **compact** when it has few degrees of freedom (parameters) that need to be tuned by learning (we expect that more compact representations of a function yield better generalization)

There exist functions that cannot be compactly represented (in terms of number of elements) by architectures that are too shallow: **functions that can be compactly ($O(N)$) represented by an architecture of depth $k$ might require an exponential number of parameters ($O(2^N)$) to be represented by an architecture of depth $k-1$**

# Theoretical Motivation

For example, it can be proved that the number of examples needed by kernel machines with a Gaussian kernel for the **parity function** grows exponentially with the input dimension [Bengio et al., 2007a]

When adding a level to the architecture ($N$ input units / $M$ output units):

- The number of parameters grows quadratically as $N \cdot M$
- The number of elements that can be represented grows exponentially as $2^M$

# Outline

# Distributed Representations

Regarding the representation of the information, we have

- Purely local representations (at one extreme):
  - Every piece of information is localized in a subset of features
  - Features are many times mutually exclusive
  - Example: representation of integers $n < N$ with a vector of $N$ bits, with a 1 in position $n$ and 0 elsewhere
- Dense distributed representations (at the other extreme):
  - The features are not mutually exclusive (features may even be statistically independent)
  - The information is not localized in a particular feature but distributed across many
  - Example: binary representation of integers $n < N$ with a vector of $\log_2 N$ positions
- Sparse distributed representations (in the middle): the representation is distributed, but many of the elements are 0

**It is believed that the success of deep architectures is closely related to the fact that they learn distributed representations**

# Distributed Representations

**Purely local representations scale polynomially with the number of features** (example: local representation of integers)

**In contrast, the total number of examples that can be distinguished using a distributed representation scales, usually, exponentially with the number of features** (example: binary representation of integers)

An interpretation: **Each feature in a discrete distributed representation can be seen as a partition of the space in $M$ regions** ($M$ is the number of possible values of the feature): Different partitions (related to different features) are combined to give rise to a potentially exponential number of possible intersection regions of the input space

# Outline

# Outline

# Which Kind of Data Are More Suitable?

According to Geoffrey Hinton, the data more suitable for deep architectures should have these properties:

- High-dimensional data (hundreds, thousands, or even more)
- Noise is not the main problem in the data
- The data is structured, but this structure is difficult to represent in a simple model
- We have a VERY large amount of available data (to capture the many factors of variation in real data)

With this kind of data, **the main problem is to imagine how can we represent this complex structure so that it can be learned**

**A Deep Learning model will construct a representation of the complex structure of the data with a lot of hidden layers**

# Which Kind of Data Are More Suitable?

When comparing to SVMs, for example, it should be noted that

- In the final model of SVMs ($f(x) = b + \sum_i \alpha_i K(\mathbf{x}, \mathbf{x}_i)$), every support vector $\mathbf{x}_i$ can be seen as a "feature" to be compared with in order to obtain a "global match": **Although the support vectors are found in a very clever way, they are not learned in the training process: they are fixed a priori**
- The same happens if we look at the transformation of the input data in the Feature Space (with the kernel trick): **Even if the dimension of the Feature Space is infinite, the transformation is fixed**
- There is **only one layer of adaptive weights**

**We are not saying that SVMs are not a good and useful model, simply that they are probably not the most suitable model for certain problems**

# Outline

# What About the Resources Needed?

When you work with deep learning architectures to process VERY large amounts of data, you may need

- A good knowledge of the underlying concepts of the models
- Very efficient implementations of the software
- High performance resources (GPUs, supercomputers)
- In some extreme cases (Google Translate), in high quantities (2000 GPUs)
- In some extreme cases (Google Translate), even construct specific hardware (TPUs)
- (Probably) A good team of scientists and engineers
- Many hours (weeks, months,...) of trial-and-error runs

Maybe your laptop is not enough...

That's it!

# Bibliography


- Ackley, D. H., Hinton, G. E., and Sejnowski, T. J. (1985). A Learning Algorithm for Boltzmann Machines. *Cognitive Science*, 9:147–169.
- Bengio, Y. (2009). Learning Deep Architectures for AI. *Foundations and Trends in Machine Learning*, 2(1):1–127.
- Bengio, Y., Delalleau, O., and Le Roux, N. (2007a). The Curse of Highly Variable Functions for Local Kernel Machines. In *Advances in Neural Information Processing Systems*, volume 18, pages 107–114. MIT Press.
- Bengio, Y., Lamblin, P., Popovici, D., and Larochelle, H. (2007b). Greedy Layer-wise Training of Deep Networks. In *Advances in Neural Information Processing Systems*, volume 19, pages 153–160. MIT Press.
- Guyon, I., Dror, G., Lemaire, V., Taylor, G., and Aha, D. (2011). Unsupervised and Transfer Learning Challenge. In *International Joint Conference on Neural Networks*.
- Hinton, G. E., Dayan, P., Frey, B. J., and Neal, R. (1995). The Wake-Sleep Algorithm for Unsupervised Neural Networks. *Science*, 268:1158–1161.
- Hinton, G. E., Deng, L., Yu, D., Dahl, G., Mohamed, A. R., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T. N., and Kingsbury, B. (2012a). Deep Neural Networks for Acoustic Modeling in Speech Recognition: The Shared Views of Four Research Groups . *IEEE Signal Processing Magazine*, 29(6):82–97.
- Hinton, G. E., Osindero, S., and Teh, Y. (2006). A Fast Learning Algorithm for Deep Belief Nets. *Neural Computation*, 18(7):1527–1554.
- Hinton, G. E. and Salakhutdinov, R. R. (2006). Reducing the Dimensionality of Data with Neural Networks. *Science*, 313(5786):504–507.
- Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2012b). Improving Neural Networks by Preventing Co-adaptation of Feature Detectors. Technical Report abs/1207.0580, Computing Research Repository (CoRR).

# Bibliography


- Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). ImageNet Classification with Deep Convolutional Neural Networks. In *Advances in Neural Information Processing Systems*, volume 24, pages 1106–1114. MIT Press.
- Le, Q. V., Ranzato, M. A., Monga, R., Devin, M., Chen, K., Corrado, G. S., and Ng, A. Y. (2012). Building High-level Features Using Large Scale Unsupervised Learning. In *29th International Conference on Machine Learning*.
- Mobahi, H., Collobert, R., and Weston, J. (2009). Deep learning from Temporal Coherence in Video. In *26th International Conference on Machine Learning*, pages 737–744.
- Mohamed, A. R., Dahl, G., and Hinton, G. E. (2012). Acoustic Modeling Using Deep Belief Networks. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(1):14–22.
- Neal, R. M. (1992). Connectionist Learning of Belief Networks. *Artificial Intelligence*, 56(1):11–113.
- Ranzato, M., Poultney, C., Chopra, S., and LeCun, Y. (2007). Efficient Learning of Sparse Representations with an Energy-based Model. In *Advances in Neural Information Processing Systems*, volume 19, pages 1137–1144. MIT Press.
- Salakhutdinov, R. R., Mnih, A., and Hinton, G. E. (2007). Restricted Boltzmann Machines for Collaborative Filtering. In *24th International Conference on Machine Learning*, pages 791–798.
- Srivastava, N., Hinton, G. E., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*, 15:1929–1958.
- Weston, J., Ratle, F., and Collobert, R. (2008). Deep Learning via Semi-supervised Embedding. In *25th International Conference on Machine Learning*, pages 1168–1175.