

Learning Deep Representations

Yoshua Bengio

September 25th, 2008

DARPA Deep Learning Workshop

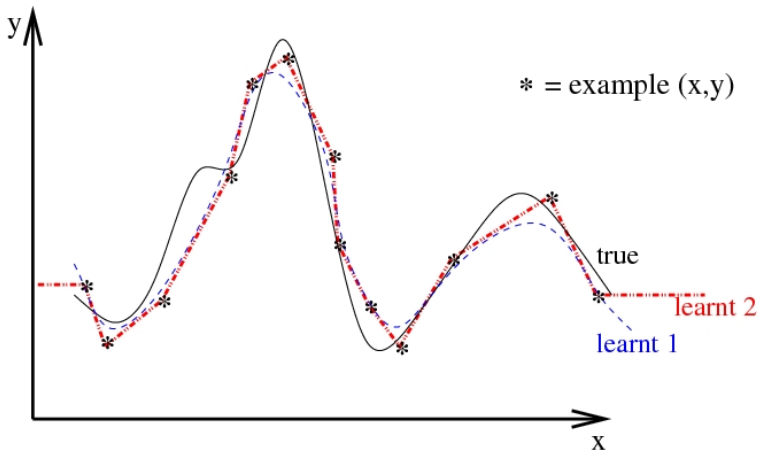
Thanks to : James Bergstra, Olivier Breuleux, Aaron Courville, Olivier Delalleau, Dumitru Erhan, Pascal Lamblin, Hugo Larochelle, Jerome Louradour, Nicolas Le Roux, Pierre-Antoine Manzagol, Dan Popovici, François Rivest, Joseph Turian, Pascal Vincent
Check review paper “Learning Deep Architectures for AI” on my web page

- Why deep learning?
Theoretical results : *to efficiently represent highly-varying functions*
- Why is it hard? : non-convexity
- Why our current algorithms working?
- Going Forward : Research program
 - Focus on optimization, large scale, sequential aspect
 - Exploit un-/semi-supervised, multi-task, multi-modal learning
 - Curriculum
 - Parallel search for solutions
 - Synthetically generated + real data of increasing complexity

1D Generalization

Why Deep Learning? Let us go back to basics.

Easy 1D generalization if the target function is smooth (few variations).



Curse of Dimensionality

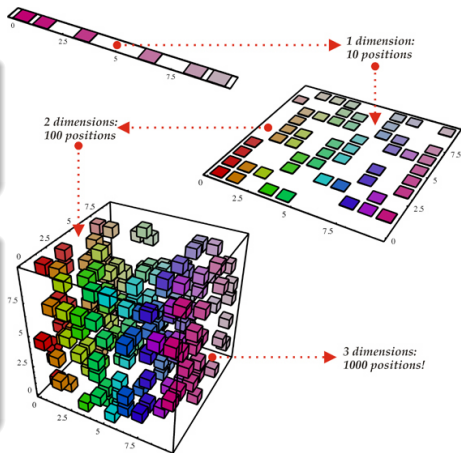
Local generalization : local kernel SVMs, GP, decision trees, LLE, Isomap, etc.

Theorem Sketch

Local learning algorithms cannot generalize to variations not covered by the training set.

Informal Corollary

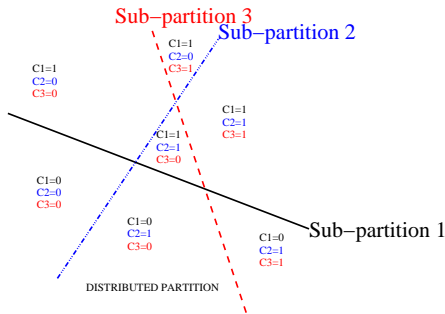
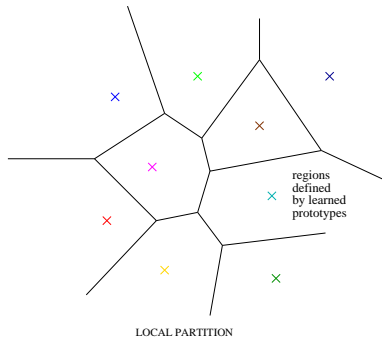
Local learning algorithms can require a number of training examples exponential in the input dimension to obtain a given generalization error.



Local learning ok in high dimension if target function is smooth

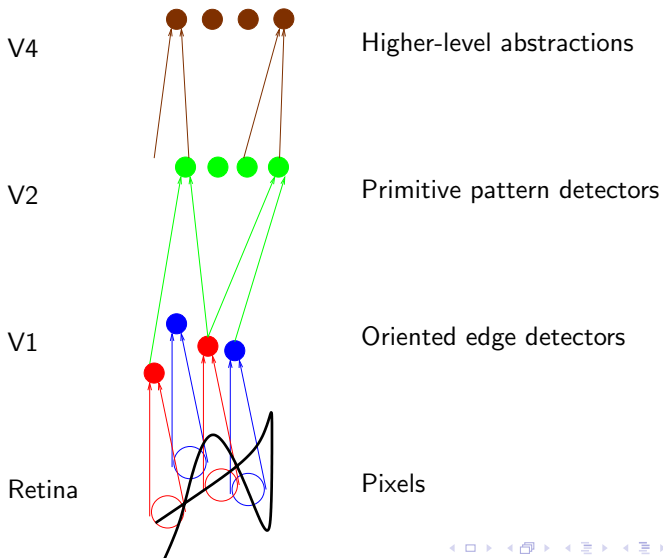
Strategy : Distributed Representations

- **Distributed representation** : input \Rightarrow combination of many features
- Parametrisation : **Exponential advantage** : distr. vs local
- Missing in most learning algorithms



Exploiting Multiple Levels of Representation

Distributed not enough : need non-linear + depth of composition



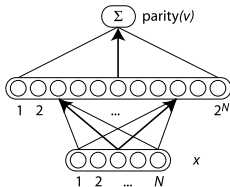
Architecture Depth

Most current learning algorithms have depth 1, 2 or 3 : **shallow**

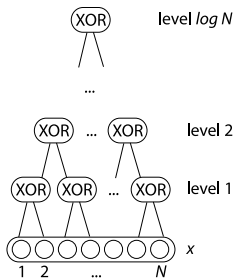
Theorem Sketch

When a function can be compactly represented by a deep architecture, representing it with a shallow architecture can require a number of elements exponential in the input dimension.

SHALLOW



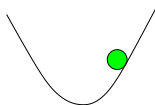
DEEP



Fat architecture \Rightarrow too rich space \Rightarrow poor generalization

Training Deep Architectures : the Challenge

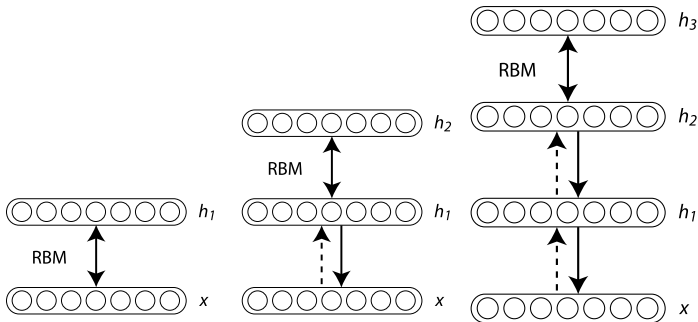
- Two levels suffice to represent any function
- Shallow & local learning works for simpler problems : insufficient for AI-type tasks
- Up to 2006, failure of attempts to train deep architectures (except Yann Le Cun's convolutional nets !)
- Why ? **Non-convex optimisation** and **stochastic** !
- Focus NIPS 1995-2005 : convex learning algorithms



⇒ Let us face the challenge !

2006 : Breakthrough !

- **FIRST** : successful training of **deep architectures** !
Hinton et al (UofT) Neural Comp. 2006, followed by Bengio et al (U.Montreal), and Ranzato et al (NYU) at NIPS'2006
- One **trains one layer after the other** of a deep MLP
- **Unsupervised** learning in each layer of initial representation
- Continue training an ordinary but deep MLP **near a better minimum**



Deep Belief Network (DBN)

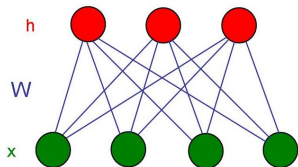
Individual Layer : RBMs and auto-encoders

State-of-the-art 'layer components' : variants of RBMs and Auto-Encoders

Deep connections between the two...

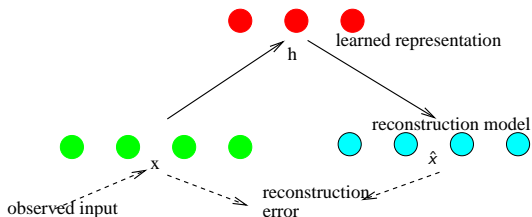
Restricted Boltzmann Machine

Efficient inference of factors h



Auto-encoder :

Find compact representation :
encode x into $h(x)$,
decode into $\hat{x}(h(x))$.



Denoising Auto-Encoders

More flexible alternative to RBMs/DBNs, while competitive in accuracy

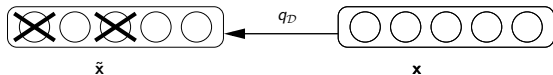


x

- Clean input $\mathbf{x} \in [0, 1]^d$ is partially destroyed, yielding corrupted input : $\tilde{\mathbf{x}} \sim q_{\mathcal{D}}(\tilde{\mathbf{x}}|\mathbf{x})$.
- $\tilde{\mathbf{x}}$ is mapped to hidden representation $\mathbf{y} = f_{\theta}(\tilde{\mathbf{x}})$.
- From \mathbf{y} we reconstruct a $\mathbf{z} = g_{\theta'}(\mathbf{y})$.
- Train parameters to minimize the cross-entropy “reconstruction error”
- Corresponds to maximizing variational bound on likelihood of a generative model
- Naturally handles missing values / occlusion / multi-modality

Denoising Auto-Encoders

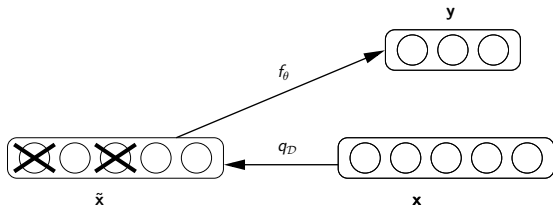
More flexible alternative to RBMs/DBNs, while competitive in accuracy



- Clean input $\mathbf{x} \in [0, 1]^d$ is **partially destroyed**, yielding **corrupted input** : $\tilde{\mathbf{x}} \sim q_D(\tilde{\mathbf{x}}|\mathbf{x})$.
- $\tilde{\mathbf{x}}$ is mapped to **hidden representation** $\mathbf{y} = f_\theta(\tilde{\mathbf{x}})$.
- From \mathbf{y} we **reconstruct** a $\mathbf{z} = g_{\theta'}(\mathbf{y})$.
- Train parameters to minimize the **cross-entropy** “reconstruction error”
- Corresponds to maximizing variational bound on likelihood of a generative model
- Naturally handles missing values / occlusion / multi-modality

Denoising Auto-Encoders

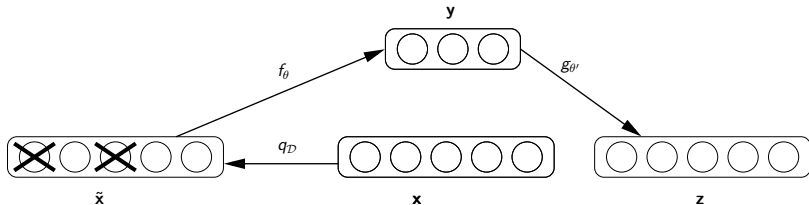
More flexible alternative to RBMs/DBNs, while competitive in accuracy



- Clean input $\mathbf{x} \in [0, 1]^d$ is **partially destroyed**, yielding **corrupted input** : $\tilde{\mathbf{x}} \sim q_D(\tilde{\mathbf{x}}|\mathbf{x})$.
- $\tilde{\mathbf{x}}$ is mapped to **hidden representation** $\mathbf{y} = f_\theta(\tilde{\mathbf{x}})$.
- From \mathbf{y} we **reconstruct** a $\mathbf{z} = g_{\theta'}(\mathbf{y})$.
- Train parameters to minimize the **cross-entropy** “reconstruction error”
- Corresponds to maximizing variational bound on likelihood of a generative model
- Naturally handles missing values / occlusion / multi-modality

Denoising Auto-Encoders

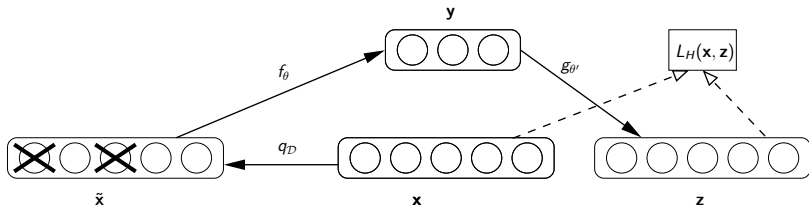
More flexible alternative to RBMs/DBNs, while competitive in accuracy



- Clean input $\mathbf{x} \in [0, 1]^d$ is **partially destroyed**, yielding **corrupted input** : $\tilde{\mathbf{x}} \sim q_D(\tilde{\mathbf{x}}|\mathbf{x})$.
- $\tilde{\mathbf{x}}$ is mapped to **hidden representation** $\mathbf{y} = f_\theta(\tilde{\mathbf{x}})$.
- From \mathbf{y} we **reconstruct** a $\mathbf{z} = g_{\theta'}(\mathbf{y})$.
- Train parameters to minimize the **cross-entropy** “reconstruction error”
- Corresponds to maximizing variational bound on likelihood of a generative model
- Naturally handles missing values / occlusion / multi-modality

Denoising Auto-Encoders

More flexible alternative to RBMs/DBNs, while competitive in accuracy



- Clean input $\mathbf{x} \in [0, 1]^d$ is **partially destroyed**, yielding **corrupted input** : $\tilde{\mathbf{x}} \sim q_{\mathcal{D}}(\tilde{\mathbf{x}}|\mathbf{x})$.
- $\tilde{\mathbf{x}}$ is mapped to **hidden representation** $\mathbf{y} = f_{\theta}(\tilde{\mathbf{x}})$.
- From \mathbf{y} we **reconstruct** a $\mathbf{z} = g_{\theta'}(\mathbf{y})$.
- Train parameters to minimize the **cross-entropy** “**reconstruction error**”
- Corresponds to maximizing variational bound on likelihood of a generative model
- Naturally handles missing values / occlusion / multi-modality

Recent benchmark problems

Variations on MNIST digit classification

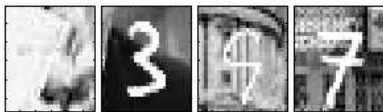
basic : subset of original MNIST digits : 10 000 training samples, 2 000 validation samples, 50 000 test samples.



(a) **rot** : applied random rotation (angle between 0 and 2π radians)



(b) **bg-rand** : background made of random pixels (value in $0 \dots 255$)



(c) **bg-img** : background is random patch from one of 20 images

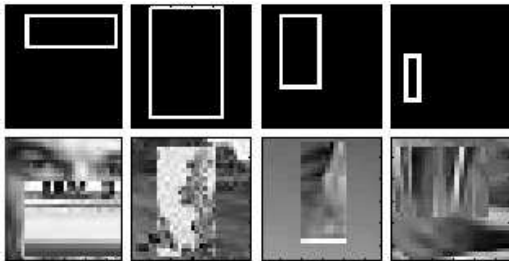


(d) **rot-bg-img** : combination of rotation and background image

Benchmark problems

Shape discrimination

- **rect** : discriminate between tall and wide rectangles on black background.



- **rect-img** : borderless rectangle filled with random image patch. Background is a different image patch.
- **convex** : discriminate between convex and non-convex shapes.



Performance comparison

Results

Dataset	SVM _{rbf}	DBN-3	SAA-3	SdA-3 (ν)
basic	3.03 \pm 0.15	3.11 \pm 0.15	3.46 \pm 0.16	2.80 \pm 0.14 (10%)
rot	11.11 \pm 0.28	10.30 \pm 0.27	10.30 \pm 0.27	10.29 \pm 0.27 (10%)
bg-rand	14.58 \pm 0.31	6.73 \pm 0.22	11.28 \pm 0.28	10.38 \pm 0.27 (40%)
bg-img	22.61 \pm 0.37	16.31 \pm 0.32	23.00 \pm 0.37	16.68 \pm 0.33 (25%)
rot-bg-img	55.18 \pm 0.44	47.39 \pm 0.44	51.93 \pm 0.44	44.49 \pm 0.44 (25%)
rect	2.15 \pm 0.13	2.60 \pm 0.14	2.41 \pm 0.13	1.99 \pm 0.12 (10%)
rect-img	24.04 \pm 0.37	22.50 \pm 0.37	24.05 \pm 0.37	21.59 \pm 0.36 (25%)
convex	19.13 \pm 0.34	18.63 \pm 0.34	18.41 \pm 0.34	19.06 \pm 0.34 (10%)

Strategy : multi-task semi-supervised learning

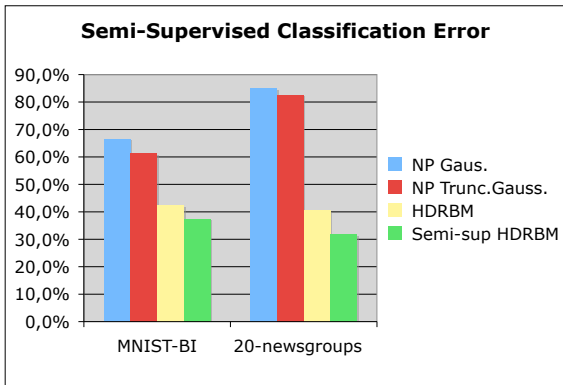
- Most available examples not semantically labeled
- Each example informs many tasks
- Representations shared among tasks
- semi-supervised + multi-task \Rightarrow Self-Taught Learning (Raina et al)
- Generalize even with 0 examples on new task !



Semi-Supervised Discriminant RBM

RBM's and auto-encoders easily extend to **semi-supervised and multi-task settings!**

Larochelle & Bengio, ICML'2008, Hybrid Discriminant RBM :
Comparisons against the current state-of-the-art in semi-supervised learning : local **Non-Parametric** semi-supervised algorithms based on neighborhood graph ; using only 1000 labeled examples.



Understanding The Challenge

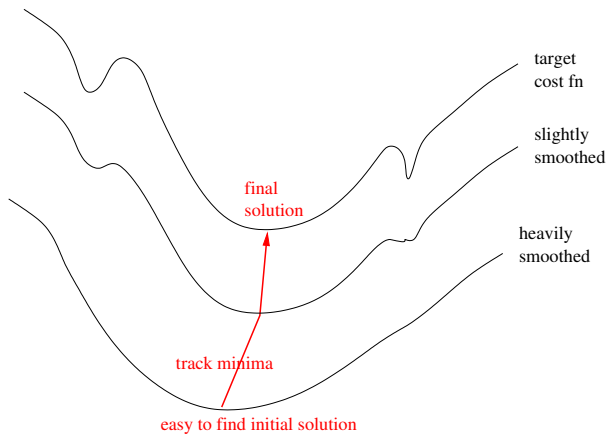
Hypothesis : under constraint of compact deep architecture, *main challenge is difficulty of optimization.*

Clues :

- Ordinary training of deep architectures (random initialization) : much more sensitive to initialization seed \Rightarrow local minima
- Comparative experiments (Bengio et al, NIPS'2006) show that the main difficulty is getting the lower layers to do something useful.
- Current learning algorithms for deep nets appear to be guiding the optimization to a “good basin of attraction”

Understanding Why it Works

Hypothesis : current solutions similar to **continuation methods**



Several Strategies are Continuations

- Older : stochastic gradient from small parameters
- Breakthrough : greedy layer-wise construction
- New : gradually bring in more difficult examples



Curriculum Strategy

Start with simpler, easier examples, and gradually introduce more of the more complicated ones as the learner is ready to learn them.



Design the sequence of tasks / datasets to guide learning/optimization.



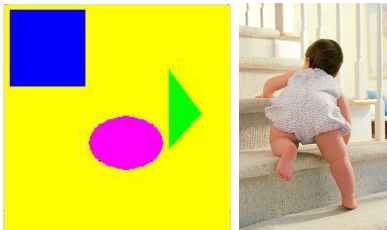
Strategy : Society = Parallel Optimisation

- Each agent = potential solution
- Better solutions spread through learned language
- Similar to genetic evolution : parallel search + recombination
- R. Dawkins' *Memes*
- Simulations support this hypothesis



Baby AI Project

Combine many strategies, to obtain a baby AI that masters the semantics of a simple visual + linguistic universe



There is a small triangle. What color is it? **Green**

Current work : generating synthetic videos, exploit hints in synthetically generated data (knowing semantic ground truth)

The Research Program

- **Motivation : need deep architectures for AI !**
- **Focus :**
 - Optimization issue, avoiding poor local minima
 - Large datasets
 - Sequential aspect / learning context
- **Exploit :**
 - unsupervised / semi-supervised learning
 - multiple tasks
 - mutual dependencies in several modalities (image - language)
 - Curriculum : human-guided training, self-guided (active) learning
 - Parallel search
 - mixture of synthetically generated and natural data, of gradually increasing complexity.

The U.Montreal Machine Learning Lab :

- Created in 1993, now 2 chairs, 20 researchers including Yoshua Bengio, Douglas Eck, Pascal Vincent, Aaron Courville, Joseph Turian
- A NIPS presence since 1988 ; **YB Program Co-Chair NIPS'2008**
- Major contribution to understanding recurrent neural networks and context learning, since 1994
- Major contribution to distributed representations in language modeling, since 2000-2003
- Three groups initiated renewal in **deep architectures** in 2006 : UofT, NYU, U.Montreal
- Organized **Deep Learning Workshop at NIPS'2008**