Large-Scale Deep Learning

IPAM SUMMER SCHOOL

Marc'Aurelio Ranzato  -  Google
ranzato@google.com
www.cs.toronto.edu/~ranzato

UCLA, 24 July 2012
Two Approaches to Deep Learning

**Deep Neural Nets:**
- usually more efficient (at training & test time)
- typically more unconstrained (partition function has to be replaced by other constraints, e.g. sparsity).
- more flexible
- ideal for end-to-end learning of complex systems

**Deep Probabilistic Models:**
- typically intractable
- easier to compose
- easier to interpret (e.g., you can generate samples from them)
- better deal with uncertainty
Example: Auto-Encoder

Neural Net:

\[ Z = \sigma \left( W_e^T X + b_e \right) \]

reconstruction \[ \hat{X} = W_d Z + b_d \]

Probabilistic Model (Gaussian RBM):

\[ E[Z|X] = \sigma \left( W^T X + b_e \right) \]

\[ E[X|Z] = W Z + b_d \]

Vincent “A connection between score matching and denoising autoencoders” Neural Comp. 2011
Swersky et al. “On autoencoders and score matching for EBM” ICML 2011
gated MRF

\[ p(h_k^c = 1|v) = f(P_k(C'v)^2) \]

- relation to simple-complex cell model
gated MRF & Subspace ICA

Subspace ICA:

\[ p(h^c_k = 1 | v) = f(P_k(C'v)^2) \]

Hyvarinen et al. “Emergence of...features by...independent feature subspaces” Neural Comp 2000

Probabilistic Model (gated MRF):

\[ p(h^c_k = 1 | v) = \sigma(-P_k(C'v)^2) \]

Welling et al. “Learning sparse topographic representations with PoT” NIPS 2002
Ranzato et al. “Factored 3-way RBMs for modeling natural images” AISTATS 2010
gated MRF & IPSD

IPSD:

\[ \text{feature} = \sqrt{(P_k(C'v)^2)} \]

Kavukcuoglu et al. “Learning invariant features through topographic filter maps” CVPR 2009

Probabilistic Model (gated MRF):

\[ p(h^c_k = 1|v) = \sigma (-P_k(C'v)^2) \]

Welling et al. “Learning sparse topographic representations with PoT” NIPS 2002
Ranzato et al. “Factored 3-way RBMs for modeling natural images” AISTATS 2010
Probabilities: yes/no?

**Deep Neural Nets:**
- usually more efficient (at training & test time)
- typically more unconstrained (partition function has to be replaced by other constraints, e.g. sparsity).
- more flexible
- ideal for end-to-end learning of complex systems

**Deep Probabilistic Models:**
- typically intractable
- easier to compose
- easier to interpret (e.g., you can generate samples from them)
- better deal with uncertainty
Today we are going to focus on Deep Neural Nets since they are more easily scalable
By “pooling” (e.g., max or average) filter responses at different locations we gain robustness to the exact spatial location of features.
POOLING

Over the years, some new modules have proven to be very effective when plugged into conv-nets:

- **L2 Pooling**

\[
h_{i+1,x,y} = \sqrt{\sum_{(j,k) \in N(x,y)} h_{i,j,k}^2}
\]

*Jarrett et al. “What is the best multi-stage architecture for object recognition?” ICCV 2009*
POOLING

Over the years, some new modules have proven to be very effective when plugged into conv-nets:

- \textbf{L2 Pooling}

\[ h_{i+1, x, y} = \sqrt{\sum_{(j, k) \in N(x, y)} h_{i, j, k}^2} \]

\[ p(h_k^c = 1 | v) = f(P_k(C'v)^2) \]

\textit{Like a gated MRF!}

\textit{Jarrett et al. “What is the best multi-stage architecture for object recognition?” ICCV 2009}
POOLING & LCN

Over the years, some new modules have proven to be very effective when plugged into conv-nets:

- **L2 Pooling**

\[
h_{i+1, x, y} = \sqrt{\sum_{(j, k) \in N(x, y)} h_{i, j, k}^2}
\]

- **Local Contrast Normalization**

\[
h_{i+1, x, y} = \frac{h_{i, x, y} - m_{i, N(x, y)}}{\sigma_{i, N(x, y)}}
\]

*Jarrett et al. “What is the best multi-stage architecture for object recognition?” ICCV 2009*
L2 POOLING

\[
\sqrt{\sum_{i=1}^{5} (\cdot)_i^2}
\]

Kavukguoglu et al. “Learning invariant features …” CVPR 2009
L2 POOLING

\[
\sqrt{\sum_{i=1}^{5} (\cdot)^2_i}
\]

Kavukguoglu et al. “Learning invariant features …” CVPR 2009
L2 Pooling helps learning representations more robust to local distortions!
LOCAL CONTRAST NORMALIZATION

\[ h_{i+1,x,y} = \frac{h_{i,x,y} - m_{i,N}(x,y)}{\sigma_{i,N}(x,y)} \]
LOCAL CONTRAST NORMALIZATION

\[ h_{i+1, x, y} = \frac{h_{i, x, y} - m_i, N(x, y)}{\sigma_{i, N(x, y)}} \]

![Diagram of LCN](image)
L2 Pooling & Local Contrast Normalization help learning more invariant representations!
CONV NETS: TYPICAL ARCHITECTURE

One stage (zoom)

Filtering → Pooling → LCN

Whole system

Input Image → 1st stage → 2nd stage → 3rd stage → Linear Layer → Class Labels
Since convolutions and sub-sampling are differentiable, we can use standard back-propagation.

Algorithm:
Given a small mini-batch
- FPROP
- BPROP
- PARAMETER UPDATE
CONV NETS: EXAMPLES

- **Object category recognition**
  Boureau et al. “Ask the locals: multi-way local pooling for image recognition” ICCV 2011

- **Segmentation**
  Turaga et al. “Maximin learning of image segmentation” NIPS 2009

- **OCR**
  Ciresan et al. “MCDNN for Image Classification” CVPR 2012

- **Pedestrian detection**
  Kavukcuoglu et al. “Learning convolutional feature hierarchies for visual recognition” NIPS 2010

- **Robotics**
  Sermanet et al. “Mapping and planning..with long range perception” IROS 2008
LIMITATIONS & SOLUTIONS

- requires lots of labeled data to train
+ unsupervised learning

- difficult optimization
+ layer-wise training

- scalability
+ distributed training
LIMITATIONS & SOLUTIONS

- requires lots of labeled data to train
+ unsupervised learning

- difficult optimization
+ layer-wise training

- scalability
+ distributed training
Observation #1: more features always improve performance unless data is scarce.

Observation #2: deep learning methods have higher capacity and have the potential to model data better.

Q #1: Given lots of data and lots of machines, can we scale up deep learning methods?

Q #2: Will deep learning methods perform much better?
The Challenge

A Large Scale problem has:
- lots of training samples (>10M)
- lots of classes (>10K) and
- lots of input dimensions (>10K).

- best optimizer in practice is on-line SGD which is naturally sequential, hard to parallelize.
- layers cannot be trained independently and in parallel, hard to distribute
- model can have lots of parameters that may clog the network, hard to distribute across machines
Our Solution

Le et al. “Building high-level features using large-scale unsupervised learning” ICML 2012
Our Solution

1\textsuperscript{st} machine

2\textsuperscript{nd} machine

3\textsuperscript{rd} machine

1\textsuperscript{st} layer

2\textsuperscript{nd} layer

input

Ranzato
Our Solution

1\textsuperscript{st} machine

2\textsuperscript{nd} machine

3\textsuperscript{rd} machine

MODEL PARALLELISM
Distributed Deep Nets

MODEL PARALLELISM

Le et al. “Building high-level features using large-scale unsupervised learning” ICML 2012
Distributed Deep Nets

MODEL PARALLELISM + DATA PARALLELISM

Le et al. “Building high-level features using large-scale unsupervised learning” ICML 2012
Asynchronous SGD

PARAMETER SERVER

1st replica

2nd replica

3rd replica
Asynchronous SGD

\[ \frac{\partial L}{\partial \theta_1} \]

PARAMETER SERVER

1\textsuperscript{st} replica

2\textsuperscript{nd} replica

3\textsuperscript{rd} replica
Asynchronous SGD

PARAMETER SERVER

\[ \theta_1 \]

1st replica

2nd replica

3rd replica
Asynchronous SGD

PARAMETER SERVER
(update parameters)
Asynchronous SGD

\[ \frac{\partial L}{\partial \theta_2} \]

PARAMETER SERVER

1\textsuperscript{st} replica

2\textsuperscript{nd} replica

3\textsuperscript{rd} replica

Ranzato
Asynchronous SGD

PARAMETER SERVER

$\theta_2$

1\textsuperscript{st} replica

2\textsuperscript{nd} replica

3\textsuperscript{rd} replica
Asynchronous SGD

PARAMETER SERVER
(update parameters)
Highly Distributed Asynchronous SGD

SHARDED PARAMETER SERVER

1\textsuperscript{st} replica

2\textsuperscript{nd} replica

3\textsuperscript{rd} replica
Problem: each parameter needs its own learning rate!
Highly Distributed Asynchronous SGD

SHARDED PARAMETER SERVER

1st replica

2nd replica

3rd replica

Ranzato
Adagrad

\[ \theta_t^i \leftarrow \theta_{t-1}^i - \eta_t^i \frac{\partial L}{\partial \theta_t^i} \]

\[ \eta_t^i = \frac{\bar{\eta}}{\sqrt{\sum_{k=0}^{t} \left( \frac{\partial L}{\partial \theta_k^i} \right)^2}} \]

- similar to diagonal approx. of Hessian
- takes care of different scaling of learning rates
- handles annealing automatically

Duchi et al. “Adaptive subgradient methods for online learning and stochastic optimization”
JMLR 2011
from Quiroga et al. “Invariant visual representation by single neurons in the human brain” Nature 2005

“Here we report on a remarkable subset of MTL neurons that are selectively activated by strikingly different pictures of given individuals, landmarks or objects and in some cases even by letter strings with their names.”

Halle Berry neuron
Unsupervised Learning With 1B Parameters

DATA: 10M youtube (unlabeled) frames of size 200x200.

Le et al. “Building high-level features using large-scale unsupervised learning” ICML 2012
Unsupervised Learning With 1B Parameters

Deep Net:
- 3 stages
- each stage consists of local filtering, L2 pooling, LCN
  - 18x18 filters
  - 8 filters at each location
  - L2 pooling and LCN over 5x5 neighborhoods
- training jointly the three layers by:
  - reconstructing the input of each layer
  - sparsity on the code

Le et al. “Building high-level features using large-scale unsupervised learning” ICML 2012
Unsupervised Learning With 1B Parameters

Deep Net:
- 3 stages
- each stage consists of local filtering, L2 pooling, LCN
  - 18x18 filters
  - 8 filters at each location
  - L2 pooling and LCN over 5x5 neighborhoods
- training jointly the three layers by:
  - reconstructing the input of each layer
  - sparsity on the code

1B parameters!!!
Unsupervised Learning With 1B Parameters

One stage (zoom)

Filtering → L2 Pooling → LCN

Whole system

Input Image → 1st stage → 2nd stage → 3rd stage
Unsupervised Learning With 1B Parameters

\[ L = \text{ReconstructionErrorLayer}(1) + \]
\[ \quad \text{ReconstructionErrorLayer}(2) + \]
\[ \quad \text{ReconstructionErrorLayer}(3) + \]
\[ \quad \lambda \text{SparsityLayer}(3) \]
Validating Unsupervised Learning

The network has seen lots of objects during training, but without any label.

Q.: how can we validate unsupervised learning?

Q.: Did the network form any high-level representation? E.g., does it have any neuron responding for faces?

- build validation set with 50% faces, 50% random images
- study properties of neurons
Validating Unsupervised Learning

1st stage → 2nd stage → 3rd stage

neuron responses
Top Images For Best Face Neuron
Best Input For Face Neuron
Face / No face
Invariance Properties

feature response vs. horizontal offset

feature response vs. vertical offset

best feature
threshold
Invariance Properties

- In the left graph, the feature response is plotted against the scale factor. The graph shows a peak at a certain scale factor, indicating a strong response, with a threshold line for comparison.

- In the right graph, the feature response is plotted against the rotate degree. The graph shows a gradual decrease in response as the rotate degree increases, with a threshold line for comparison.
## Comparison: Face Neuron

<table>
<thead>
<tr>
<th>Method</th>
<th>Accuracy %</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random guess</td>
<td>65</td>
</tr>
<tr>
<td>Best training sample</td>
<td>74</td>
</tr>
<tr>
<td>1 layer net</td>
<td>71</td>
</tr>
<tr>
<td>Deep net before training</td>
<td>67</td>
</tr>
<tr>
<td><strong>Deep net after training</strong></td>
<td><strong>81</strong></td>
</tr>
<tr>
<td>Deep net after training without LCN</td>
<td>78</td>
</tr>
</tbody>
</table>
## Comparison: Face Neuron

<table>
<thead>
<tr>
<th>Method</th>
<th>Accuracy %</th>
</tr>
</thead>
<tbody>
<tr>
<td>K-Means on 40x40, k=30000</td>
<td>72</td>
</tr>
<tr>
<td>3-layer Autoencoder</td>
<td>72</td>
</tr>
<tr>
<td>Deep net after training</td>
<td>81</td>
</tr>
</tbody>
</table>
Pedestrian Neuron
Top Images for Pedestrian Neuron
Pedestrian Neuron
Cat Neuron
Top Images for Cat Neuron
Cat Neuron
Unsupervised + Supervised (ImageNet)
Object Recognition on ImageNet

**IMAGENET v.2011 (16M images, 20K categories)**

<table>
<thead>
<tr>
<th>METHOD</th>
<th>ACCURACY %</th>
</tr>
</thead>
<tbody>
<tr>
<td>Weston &amp; Bengio 2011</td>
<td>9.3</td>
</tr>
<tr>
<td>Linear Classifier on deep features</td>
<td>13.1</td>
</tr>
<tr>
<td>Deep Net (from random)</td>
<td>13.6</td>
</tr>
<tr>
<td>Deep Net (from unsup.)</td>
<td><strong>15.8</strong></td>
</tr>
</tbody>
</table>
Top Inputs After Supervision
Top Inputs After Supervision
References on ConvNets & alike

Tutorials & Background Material


Convolutional Nets

Unsupervised Learning

- ICA with Reconstruction Cost for Efficient Overcomplete Feature Learning. Le, Karpenko, Ngiam, Ng. In NIPS*2011

- Rifai, Vincent, Muller, Glorot, Bengio, Contracting Auto-Encoders: Explicit invariance during feature extraction, in: Proceedings of the Twenty-eight International Conference on Machine Learning (ICML'11), 2011


Multi-modal Learning

Locally Connected Nets

- Gregor, LeCun “Emergence of complex-like cells in a temporal product network with local receptive fields” Arxiv. 2009
- Ranzato, Mnih, Hinton “Generating more realistic images using gated MRF's” NIPS 2010
- Le, Ngiam, Chen, Chia, Koh, Ng “Tiled convolutional neural networks” NIPS 2010

Distributed Learning


Papers on Scene Parsing

- Socher, Lin, Ng, Manning, “Parsing Natural Scenes and Natural Language with Recursive Neural Networks”. International Conference of Machine Learning (ICML 2011) 2011.
Papers on Object Recognition

- Sermanet, LeCun: Traffic Sign Recognition with Multi-Scale Convolutional Networks, Proceedings of International Joint Conference on Neural Networks (IJCNN'11)
- Ciresan, Meier, Gambardella, Schmidhuber. Convolutional Neural Network Committees For Handwritten Character Classification. 11th International Conference on Document Analysis and Recognition (ICDAR 2011), Beijing, China.

Papers on Action Recognition

- Learning hierarchical spatio-temporal features for action recognition with independent subspace analysis, Le, Zou, Yeung, Ng. In Computer Vision and Pattern Recognition (CVPR), 2011

Papers on Segmentation

Papers on Vision for Robotics


Deep Convex Nets & Deconv-Nets

- Zeiler, Taylor, Fergus "Adaptive Deconvolutional Networks for Mid and High Level Feature Learning." ICCV. 2011

Papers on Biological Inspired Vision

- Pinto, Doukhan, DiCarlo, Cox "A high-throughput screening approach to discovering good forms of biologically inspired visual representation." {PLoS} Computational Biology. 2009
Papers on Embedded ConvNets for Real-Time Vision Applications

Papers on Image Denoising Using Neural Nets
- Burger, Schuler, Harmeling: Image Denoising: Can Plain Neural Networks Compete with BM3D?, Computer Vision and Pattern Recognition, CVPR 2012,

ConvNets & Invariance from a Mathematical Perspective
- Bruna, Mallat: Invariance Scattering Convolutional Network, PAMI 2012,

ConvNets to Learn Embeddings
- Hadsell, Chopra, LeCun: Dimensionality Reduction by Learning an Invariant Mapping, CVPR 2006,
Software & Links

Deep Learning website
- http://deeplearning.net/

C++ code for ConvNets
- http://eblearn.sourceforge.net/

Matlab code for R-ICA unsupervised algorithm

Python-based learning library
- http://deeplearning.net/software/theano/

Lush learning library which includes ConvNets
- http://lush.sourceforge.net/

Torch7: learning library that supports neural net training
http://www.torch.ch
Acknowledgements

Quoc Le, Andrew Ng

Jeff Dean, Kai Chen, Greg Corrado, Matthieu Devin, Mark Mao, Rajat Monga, Paul Tucker, Samy Bengio

Yann LeCun, Pierre Sermanet, Clement Farabet