Structured deep learning: Pose and gestures

Christian Wolf
Université de Lyon, INSA-Lyon
LIRIS UMR CNRS 5205

April 30th, 2015
LIRIS: scientific activities

- Simulation, virtualité et sciences computationnelles
- Vision intelligente et reconnaissance visuelle
- M2DisCo
- GeoMod
- Géométrie et modélisation
- SAARA
- Beagle
- Imagine
- R3AM
- DRIM
- SILEX
- GRAMA
- DM2L
- Data Science
- Interactions et cognition
- Services, Systèmes distribués et Sécurité
The «IMAGINE» team

- 6 Full professors
- 13 Assistant professors (MCF)
- 1 Engineer
- 2 Post-doctoral students
- 23 PhD students
Computer vision: recognition of «X»

Faces | Emotions | Objects | Scene labeling

Gestures | Pose | Behavior | Text
Pose and gestures

Deep Learning: overview

Pose estimation: hand

Direct deep gesture recognition (without pose estimation)
Objectives : prediction!

- Being able to make predictions on unseen data
- Learn a prediction model from training data
- Generalize!
Learning to predict

We would like to predict a value $t$ from an observed input $x$

$$ t = y(x, w) $$

if $t$ is continuous: regression
If $t$ is discrete: classification

Parameters $w$ are learned from training data.
Two classical deep models

Deep Belief Networks

Convolutional Neural Networks
Neural networks

« Perceptron »

\[ y(x, w) = \sum_{i=0}^{D} w_i x_i \]
Neural networks

« Logistic regression »

\[ y_k(x, w) = h \left( \sum_{i=0}^{D} w^{(1)}_{ji} x_i \right). \]
Neural networks

« Multi-layer perceptron » (MLP)

\[ y_k(x, w) = \sigma \left( \sum_{j=0}^{M} w_{kj}^{(2)} h \left( \sum_{i=0}^{D} w_{ji}^{(1)} x_i \right) \right) \]

(Biases have been integrated)
An example

Figure 5.4 Example of the solution of a simple two-class classification problem involving synthetic data using a neural network having two inputs, two hidden units with \( \tanh \) activation functions, and a single output having a logistic sigmoid activation function. The dashed blue lines show the \( z = 0 \) contours for each of the hidden units, and the red line shows the \( y = 0 \) decision surface for the network. For comparison, the green line denotes the optimal decision boundary computed from the distributions used to generate the data.

Symmetries, and thus any given weight vector will be one of a set \( 2^M \) equivalent weight vectors. Similarly, imagine that we interchange the values of all of the weights (and the bias) leading both into and out of a particular hidden unit with the corresponding values of the weights (and bias) associated with a different hidden unit. Again, this clearly leaves the network input–output mapping function unchanged, but it corresponds to a different choice of weight vector. For \( M \) hidden units, any given weight vector will belong to a set of \( M! \) equivalent weight vectors associated with this interchange symmetry, corresponding to the \( M! \) different orderings of the hidden units. The network will therefore have an overall weight-space symmetry factor of \( M!2^M \).

For networks with more than two layers of weights, the total level of symmetry will be given by the product of such factors, one for each layer of hidden units.

It turns out that these factors account for all of the symmetries in weight space (except for possible accidental symmetries due to specific choices for the weight values). Furthermore, the existence of these symmetries is not a particular property of the \( \tanh \) function but applies to a wide range of activation functions (Kurková and Kainen, 1994). In many cases, these symmetries in weight space are of little practical consequence, although in Section 5.7 we shall encounter a situation in which we need to take them into account.

5.2. Network Training

So far, we have viewed neural networks as a general class of parametric nonlinear functions from a vector \( x \) of input variables to a vector \( y \) of output variables. A simple approach to the problem of determining the network parameters is to make an analogy with the discussion of polynomial curve fitting in Section 1.1, and therefore to minimize a sum-of-squares error function.

Given a training set comprising a set of input vectors \( \{x_n \} \), where \( n = 1, \ldots, N \), together with a corresponding set of output vectors \( \{y_n \} \), the goal is to find a network that minimizes the error function:

\[
\text{Error} = \sum_{n=1}^{N} (y_n - \hat{y}_n)^2
\]

where \( \hat{y}_n \) is the network's predicted output for input \( x_n \). This is typically achieved through an iterative process called backpropagation, which adjusts the weights to minimize the error function. The process involves computing the gradient of the error function with respect to the weights and using this gradient to update the weights in the direction that reduces the error.
Learning by gradient descent

Given a loss function, e.g. cross entropy error:

$$E(w) = -\sum_{n=1}^{N} \sum_{k=1}^{K} t_{kn} \ln y_k(x_n, w)$$

Iterative minimisation through gradient descent:

$$w^{(\tau+1)} = w^{(\tau)} - \eta \nabla E(w^{(\tau)})$$

Learning rate

Can be blocked in a local minimum

[C. Bishop, Pattern recognition and Machine learning, 2006]
Neural network
Back propagation

Nature

1986

- Solve general learning problems
- Tied with biological system

[Slide: X. Wang, Tutorial ICIP 2014]
Neural network
Back propagation

1986
2006

- SVM
- Boosting
- Decision tree
- KNN
- ...

- Flat structures
- Loose tie with biological systems
- Specific methods for specific tasks
  - Hand crafted features (GMM-HMM, SIFT, LBP, HOG)

Kruger et al. TPAMI’13

[Slide: X. Wang, Tutorial ICIP 2014]
Handcrafted vs. learned features

\[ X_i^m \rightarrow \text{SIFT, HoG, BoW, ...} \rightarrow \text{Handcrafted feature extraction} \rightarrow Z_i^m \rightarrow g(Z_i^m|\theta_g) \rightarrow l_i \]

\[ X_i^m \rightarrow f(X_i^m|\theta_f) \rightarrow Z_i^m \rightarrow g(Z_i^m|\theta_g) \rightarrow l_i \]

Parameters learned from training data
Convolutional Neural Networks

Introduced by Fukushima in 1980
Refined by LeCun, Bottou, Bengio, Haffner in 1998

- Parameter sharing in convolutional layers
- End to end training of the full set of parameters
Example: convolutional face finder

Learn function $F$:

$$F(\text{Image}) \rightarrow 1 \quad \text{if face}$$

$$F(\text{image}) \rightarrow -1 \quad \text{else}$$
Δ = (sortie - classe) face? [-1,1]  
Sortie > 0  → visage!

[Slide: Christophe Garcia, LIRIS]
- Unsupervised & Layer-wised pre-training
- Better designs for modeling and training (normalization, nonlinearity, dropout)
- New development of computer architectures
  - GPU
  - Multi-core computer systems
- Large scale databases

Big Data!
Imagenet / ILSVRC

- 1,461,406 annotated images in ILSVRC 2010
- 1000 classes, from word-net
- Manual annotation and careful verification

Crowd-sourcing with Amazon Mechanical Turk
### ImageNet / ILSVRC

#### Image classification

<table>
<thead>
<tr>
<th>Ground truth</th>
<th>Accuracy: 1</th>
<th>Accuracy: 1</th>
<th>Accuracy: 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Steel drum</td>
<td>Steel drum</td>
<td>Scale T-shirt</td>
<td>Scale T-shirt</td>
</tr>
<tr>
<td></td>
<td>Folding chair</td>
<td>Drumstick</td>
<td>Giant panda</td>
</tr>
<tr>
<td></td>
<td>Loudspeaker</td>
<td>Mud turtle</td>
<td>Drumstick</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Mud turtle</td>
</tr>
</tbody>
</table>

#### Single-object localization

<table>
<thead>
<tr>
<th>Ground truth</th>
<th>Accuracy: 1</th>
<th>Accuracy: 0</th>
<th>Accuracy: 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Steel drum</td>
<td>Persian cat</td>
<td>Steel drum</td>
<td>Persian cat</td>
</tr>
<tr>
<td></td>
<td>Folding chair</td>
<td>Folding chair</td>
<td>Loudspeaker</td>
</tr>
<tr>
<td></td>
<td>Microphone</td>
<td>Steel drum</td>
<td>Steel drum</td>
</tr>
<tr>
<td></td>
<td>Person</td>
<td>Folding chair</td>
<td>Folding chair</td>
</tr>
<tr>
<td></td>
<td>Microphone</td>
<td>Steel drum</td>
<td>Steel drum</td>
</tr>
<tr>
<td></td>
<td>Person</td>
<td>Folding chair</td>
<td>Folding chair</td>
</tr>
<tr>
<td></td>
<td>Microphone</td>
<td>Steel drum</td>
<td>Steel drum</td>
</tr>
<tr>
<td></td>
<td>Person</td>
<td>Folding chair</td>
<td>Folding chair</td>
</tr>
</tbody>
</table>

#### Object detection

<table>
<thead>
<tr>
<th>Ground truth</th>
<th>AP: 1.0 1.0 1.0 1.0</th>
<th>AP: 0.0 0.5 1.0 0.3</th>
<th>AP: 1.0 0.7 0.5 0.9</th>
</tr>
</thead>
<tbody>
<tr>
<td>Steel drum</td>
<td>Person</td>
<td>Folding chair</td>
<td>Microphone</td>
</tr>
<tr>
<td></td>
<td>Folding chair</td>
<td>Microphone</td>
<td>Steel drum</td>
</tr>
<tr>
<td></td>
<td>Microphone</td>
<td>Steel drum</td>
<td>Steel drum</td>
</tr>
<tr>
<td></td>
<td>Person</td>
<td>Folding chair</td>
<td>Folding chair</td>
</tr>
<tr>
<td></td>
<td>Microphone</td>
<td>Steel drum</td>
<td>Steel drum</td>
</tr>
<tr>
<td></td>
<td>Person</td>
<td>Folding chair</td>
<td>Folding chair</td>
</tr>
<tr>
<td></td>
<td>Microphone</td>
<td>Steel drum</td>
<td>Steel drum</td>
</tr>
<tr>
<td></td>
<td>Person</td>
<td>Folding chair</td>
<td>Folding chair</td>
</tr>
</tbody>
</table>

[ Russakovsky et al., 2015 – ILSVRC 2015 ]
Fig. 1: The diversity of data in the ILSVRC image classification and single-object localization tasks. For each of the eight dimensions, we show example object categories along the range of that property. Object scale, number of instances and image clutter for each object category are computed using the metrics defined in Section 3.2.2 and in Appendix B. The other properties were computed by asking human subjects to annotate each of the 1000 object categories (Russakovsky et al., 2013).

[Russakovsky et al., 2015 – ILSCVR 2015]
Neural network Back propagation

Deep belief net

Speech

1986 2006 2011 2012

• ImageNet 2013 – image classification challenge

<table>
<thead>
<tr>
<th>Rank</th>
<th>Name</th>
<th>Error rate</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>NYU</td>
<td>0.11197</td>
<td>Deep learning</td>
</tr>
<tr>
<td>2</td>
<td>NUS</td>
<td>0.12535</td>
<td>Deep learning</td>
</tr>
<tr>
<td>3</td>
<td>Oxford</td>
<td>0.13555</td>
<td>Deep learning</td>
</tr>
</tbody>
</table>

MSRA, IBM, Adobe, NEC, Clarifai, Berkley, U. Tokyo, UCLA, UIUC, Toronto .... Top 20 groups all used deep learning

• ImageNet 2013 – object detection challenge

<table>
<thead>
<tr>
<th>Rank</th>
<th>Name</th>
<th>Mean Average Precision</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>UvA-Euvision</td>
<td>0.22581</td>
<td>Hand-crafted features</td>
</tr>
<tr>
<td>2</td>
<td>NEC-MU</td>
<td>0.20895</td>
<td>Hand-crafted features</td>
</tr>
<tr>
<td>3</td>
<td>NYU</td>
<td>0.19400</td>
<td>Deep learning</td>
</tr>
</tbody>
</table>

[Slide: X. Wang, Tutorial ICIP 2014]
• ImageNet 2014 – Image classification challenge

<table>
<thead>
<tr>
<th>Rank</th>
<th>Name</th>
<th>Error rate</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Google</td>
<td>0.06656</td>
<td>Deep learning</td>
</tr>
<tr>
<td>2</td>
<td>Oxford</td>
<td>0.07325</td>
<td>Deep learning</td>
</tr>
<tr>
<td>3</td>
<td>MSRA</td>
<td>0.08062</td>
<td>Deep learning</td>
</tr>
</tbody>
</table>

• ImageNet 2014 – object detection challenge

<table>
<thead>
<tr>
<th>Rank</th>
<th>Name</th>
<th>Mean Average Precision</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Google</td>
<td>0.43933</td>
<td>Deep learning</td>
</tr>
<tr>
<td>2</td>
<td>CUHK</td>
<td>0.40656</td>
<td>Deep learning</td>
</tr>
<tr>
<td>3</td>
<td>DeepInsight</td>
<td>0.40452</td>
<td>Deep learning</td>
</tr>
<tr>
<td>4</td>
<td>UvA-Euvison</td>
<td>0.35421</td>
<td>Deep learning</td>
</tr>
<tr>
<td>5</td>
<td>Berkeley Vision</td>
<td>0.34521</td>
<td>Deep learning</td>
</tr>
</tbody>
</table>

[Slide: X. Wang, Tutorial ICIP 2014]
<table>
<thead>
<tr>
<th><strong>Deep Learning</strong></th>
<th><strong>Temporary Social Media</strong></th>
<th><strong>Prenatal DNA Sequencing</strong></th>
<th><strong>Additive Manufacturing</strong></th>
<th><strong>Baxter: The Blue-Collar Robot</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>With massive amounts of computational power, machines can now recognize objects and translate speech in real time. Artificial intelligence is finally getting smart.</td>
<td>Mocoaqua that quickly self-destruct could enhance the privacy of online communications and make people freer to be spontaneous.</td>
<td>Reading the DNA of fetuses will be the next frontier of the genomic revolution. But do you really want to know about the genetic problems or musical aptitude of your unborn child?</td>
<td>Skeptical about 3D printing? GE, the world's largest manufacturer, is on the verge of using the technology to make jet parts.</td>
<td>Rodney Brooks's newest creation is easy to interact with, but the complex innovations behind the robot show just how hard it is to get along with people.</td>
</tr>
<tr>
<td><strong>Memory Implants</strong></td>
<td><strong>Smart Watches</strong></td>
<td><strong>Ultra-Efficient Solar Power</strong></td>
<td><strong>Big Data from Cheap Phones</strong></td>
<td><strong>Supergrids</strong></td>
</tr>
<tr>
<td>A maverick neuroscientist believes he has deciphered the code by which the brain forms long-term memories. Next: testing a prosthetic implant for people suffering from long-term conditions.</td>
<td>The designers of the Pebble watch realized that a mobile phone is more useful if you don't have to take it out of your pocket.</td>
<td>Doubling the efficiency of a solar cell would completely change the economics of renewable energy. Nanotechnology just might make it possible.</td>
<td>Collecting and analyzing information from simple cell phones can provide surprising insights into how people move about and behave—and even help us understand the spread of diseases.</td>
<td>A new high-power circuit breaker could finally make highly efficient DC power grids practical.</td>
</tr>
<tr>
<td>Date</td>
<td>Event</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>--------</td>
<td>-----------------------------------------------------------------------</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10/2012</td>
<td>G. Hinton’s group wins ImageNet / ILSVRC</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>03/2013</td>
<td>G. Hinton joins Google</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>09/2013</td>
<td>Clarifai wins ImageNet / ILSVRC</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12/2013</td>
<td>Y. LeCun directs Facebook’s new AI lab</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>01/2014</td>
<td>Google acquires DeepMind for 400M$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>05/2014</td>
<td>A. Ng directs Baidu’s new AI lab</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>09/2014</td>
<td>Google wins ImageNet / ILSVRC</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Hinton NIPS paper mentions internal google dataset with 100 000 000 labelled images and 18 000 classes</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>04/2015</td>
<td>Clarifai acquires 10M$ venture capital</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Why does it work so well?

- Parameter sharing (convolutional layers)

![Diagram showing convolution and subsampling]

- Deep models
- Massive amounts of training data
  - Pascal VOC: 20,000 annotated images
  - ImageNet: 1,500,000 annotated images
- Computational power for training (GPUs!)
- New regularization techniques (dropout etc.)
« AlexNet » (wins 2012)

- Alex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton
- 8 Layers: 5 convolutional, 3 fully connected
- 60 million learned parameters
- Learned on 2 different (communicating) GPUs
- Dropout!

[Krizhevsky et al, ECCV 2012]
« GoogLeNet » (wins 2014)

- 20 layers
- Supervision at multiple layers
- 6.6% error
Image classification

Easiest classes

- red fox (100)
- hen-of-the-woods (100)
- ibex (100)
- goldfinch (100)
- flat-coated retriever (100)
- tiger (100)
- hamster (100)
- porcupine (100)
- stingray (100)
- Blenheim spaniel (100)

Hardest classes

- muzzle (71)
- hatchet (68)
- water bottle (68)
- velvet (68)
- loupe (66)
- hook (66)
- spotlight (66)
- ladle (65)
- restaurant (64)
- letter opener (59)
Sharing of features / detectors

[Honglak Lee, NIPS 2010]
Visualization of CNN features

- Select the strongest activations in the feature map
- Backproject them into the lower layers using a deconvolutional network (using stored locations of max responses)

[Zeiler and Fergus, ECCV 2014]
Visualization of CNN features

Fig. 2. Visualization of features in a fully trained model. For layers 2-5 we show the top activation sites and maps to images that cause high activations in a given feature map, projected down to pixel space using our deconvolutional network approach. Our reconstructions are not samples from the model: they are reconstructed patterns from the validation set that cause high activations in a given feature map. For each feature map we also show the corresponding image patches. Note: (i) the strong grouping within each feature map, (ii) greater invariance at higher layers and (iii) exaggeration of discriminative parts of the image, e.g. eyes and noses of dogs (layer 4, row 1, cols 1). Best viewed in electronic form. The compression artifacts are a consequence of the 30Mb submission limit, not the reconstruction algorithm itself.

[Zeiler and Fergus, ECCV 2014]
Visualization of CNN features

[Zeiler and Fergus, ECCV 2014]
Feature evolution during training

A single Strongest activation in the feature map, downprojected
Epochs 1, 2, 5, 10, 20, 30, 40, 64

[Zeiler and Fergus, ECCV 2014]
Which layers are important?

All of them ..... depth is important!

<table>
<thead>
<tr>
<th>Error %</th>
<th>Train Top-1</th>
<th>Val Top-1</th>
<th>Val Top-5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Our replication of Krizhevsky et al. [18], 1 convnet</td>
<td>35.1</td>
<td>40.5</td>
<td>18.1</td>
</tr>
<tr>
<td>Removed layers 3,4</td>
<td>41.8</td>
<td>45.4</td>
<td>22.1</td>
</tr>
<tr>
<td>Removed layer 7</td>
<td>27.4</td>
<td>40.0</td>
<td>18.4</td>
</tr>
<tr>
<td>Removed layers 6,7</td>
<td>27.4</td>
<td>44.8</td>
<td>22.4</td>
</tr>
<tr>
<td>Removed layer 3,4,6,7</td>
<td>71.1</td>
<td>71.3</td>
<td>50.1</td>
</tr>
</tbody>
</table>

[Zeiler and Fergus, ECCV 2014]
Dropout

- Introduced in 2012 for Imagenet (« AlexNet »)

- During training, for each training sample, 50% of the units are disabled.
- Punishes co-adaptation of units
- Large performance gains
Data augmentation

- Add cropped and mirrored versions of each training image
- Increases invariance
Beyond classification: other applications
Object detection: region CNN (« R-CNN »)

Objective: detect and recognize objects in arbitrary position and bounding boxes of arbitrary aspect ratio

1. Input image
2. Extract region proposals (~2k)
3. Compute CNN features
4. Classify regions

From oversegmented image

[GIrshick et al., CVPR 2014]
Object detection: bounding box regression

A CNN predicts multiple bounding boxes per image: 1 for each object instance

Training requires solving a matching problem between network outputs and ground truth object instances

[Erhan et al., CVPR 2014]
Imagenet trained CNNs ... as features!

Discriminatively trained CNNs on ImageNet can be used as features for other datasets or even other applications.

<table>
<thead>
<tr>
<th># Train</th>
<th>Acc % 15/class</th>
<th>Acc % 30/class</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bo et al. [3]</td>
<td>–</td>
<td>81.4 ± 0.33</td>
</tr>
<tr>
<td>Yang et al. [17]</td>
<td>73.2</td>
<td>84.3</td>
</tr>
<tr>
<td>Non-pretrained convnet</td>
<td>22.8 ± 1.5</td>
<td>46.5 ± 1.7</td>
</tr>
<tr>
<td>ImageNet-pretrained convnet</td>
<td><strong>83.8 ± 0.5</strong></td>
<td><strong>86.5 ± 0.5</strong></td>
</tr>
</tbody>
</table>

Results on Caltech-101 (101 categories, 9144 images)

[Zeiler and Fergus, ECCV 2014]
Human pose estimation

Figure 6: Results on the LSP and FLIC datasets. We show the part localization results along with the graph skeleton we used in the model. The first two rows are examples from the LSP dataset, and the 3rd row is from the FLIC dataset. The last row shows some failure cases, which are typically due to large foreshortening, occlusions and distractions from clothing or overlapping people.

6 Conclusion
We have presented a graphical model for human pose which exploits the fact the local image measurements can be used both to detect parts (or joints) and also to predict the spatial relationships between them (Image Dependent Pairwise Relations). These spatial relationships are represented by a mixture model over types of spatial relationships. We use DCNNs to learn conditional probabilities for the presence of parts and their spatial relationships within image patches. Hence our model combines the representational flexibility of graphical models with the efficiency and statistical power of DCNNs. Our method outperforms the state of the art methods on the LSP and FLIC datasets and also performs very well on the Buffy dataset without any training.

7 Acknowledgements
This research has been supported by the Office of Naval Research ONR MURI N000014-10-1-0933 and ONR N00014-12-1-0883.

References

[Chen and Yuille, NIPS 2014]
Graphical Model with Image Dependent Pairwise Relations learned by CNN

A deep network learns:
- **Part detectors** for different joints
- **Pairwise part compatibility** functions

Final inference through graphical model (dynamic programming and generalized distance transform)
Open problems

- Vidéo : temporal and spatio-temporal dependencies

- Complex problems, complex interactions
- Deformable models
- Theoretical results on deep models

Figure: Karpathy et al, CVPR 2014
Conclusion

- Deep learning is a powerful tool for feature and representation learning.
- Works best on large amount of training data (big data!)
- **Data augmentation** is often important to increase invariance properties
- Increasingly used for cross data (pre)-training
- Feed-forward models: no inherent search / combinatorial problems
- Including structure is a goal and a challenge