Deep Learning for Visual Perception

Sven Behnke

University of Bonn
Computer Science Institute VI
Autonomous Intelligent Systems
Much Interest in Deep Learning

Support vector machine

[Google Trends]

Deep learning

Sven Behnke: Deep Learning for Visual Perception
Industry Acquisitions and Hirings

- Google
  - DNNresearch (Geoffrey Hinton)
  - DeepMind (Demis Hassabis)
- Baidu
  - Andrew Ng
- Facebook
  - Yann LeCun
- Microsoft
  - Li Deng

Google Hires Brains that Helped Supercharge Machine Learning

BY ROBERT MCMILLAN 03.13.13 6:30 AM

Geoffrey Hinton (right) Alex Krizhevsky, and Ilya Sutskever (left) will do machine learning work at Google. Photo: U of T
Special Issues and Meetings

- Special issues of many journals (PAMI, NN)
- Specialized workshops at major machine learning conferences (NIPS, ICLM)
- Representation Learning Conference (ICLR)
- Deep Learning Summits (RE.WORK, NVidia)
Deep Learning Definition

- Deep learning is a set of algorithms in machine learning that attempt to learn layered models of inputs, commonly neural networks.
- The layers in such models correspond to distinct levels of concepts, where
  - higher-level concepts are defined from lower-level ones, and
  - the same lower-level concepts can help to define many higher-level concepts.

[Bengio 2009]
Layered Representations

[Schulz and Behnke, KI 2012]
Performance of the Human Visual System
Psychophysics

- Gestalt principles
- Heuristics
- Context
- Attention

Sven Behnke: Deep Learning for Visual Perception
Visual Illusions

Kanizsa Figures

Müller-Lyer horizontal/vertical

Ebbinghaus-Titchener

Munker-White
Observations

In the world around us it mostly holds that:

- Neighboring things have something to do with each other
  - Spatially
  - Temporally
- There is hierarchical structure
  - Objects consist of parts
  - Parts are composed of components, ...
Spatial Arrangement of Facial Parts
Face Perception
Horizontal and Vertical Dependencies

Constellation Model:
Fully connected shape model

Weber, Welling, Perona ’00
Fergus, Zisserman, Perona ’03

Implicit Shape Model:
Star-Model w.r.t. Reference Point

Leibe, Schiele ’03
Leibe, Leonardis, Schiele ’04
Multi-Scale Representation

- Image pyramids are not expressive enough
Increasing Number of Features with Decreasing Resolution

- Rich representations also in the higher layers
Modeling Horizontal Dependencies

- 1D: HMM, Kalman Filter, Particle Filter
- 2D: Markov Random Fields
- Decision for level of description problematic
- Ignores vertical dependencies, flat models do not scale
Modeling Vertical Dependencies

- Structure graphs, etc.
- Ignores horizontal dependencies
Problem: Cycles make exact inference impossible
Idea: Use approximate inference
Human Visual System

Ventral (temporal) path

LGN

V1

Ocular dominance stripes

Orientation column

Blob

V2

What?

Where?

Dorsal (parietal) path

A1T

C1T

P1T

V4

MT

MST

LIP

VIP

[Kandel et al. 2000]
Visual Processing Hierarchy

- Increasing complexity
- Increasing invariance
- All connections bidirectional
- More feedback than feed forward
- Lateral connections important

[Krüger et al., TPAMI 2013]

Sven Behnke: Deep Learning for Visual Perception
Feed-Forward Models

- Neocognitron: Fukushima 1980

- Supervised training of convolutional networks: LeCun 1989
Feed-forward Models Cannot Explain Human Performance

- Performance increases with observation time

![Graph showing performance changes with observation time for different SOA and ISI conditions.](image)

*no mask condition*

80 ms SOA (ISI=60 ms) model **HMAX**

50 ms SOA (ISI=30 ms)

20 ms SOA (ISI=0 ms)

(Serre, Oliva and Poggio, PNAS, 2007)
Neural Abstraction Pyramid

[Behnke, LNCS 2766, 2003]

- Data-driven
- Analysis
- Feature extraction

- Model-driven
- Synthesis
- Feature expansion

Signals

Grouping  Competition  Completion

Abstract features
Iterative Interpretation

- Interpret most obvious parts first

- Use partial interpretation as context to resolve local ambiguities

[Behnke, LNCS 2766, 2003]
Local Recurrent Connectivity

Processor element

[Behnke, LNCS 2766, 2003]
Biological vs. Artificial Neurons

Biological (Pyramidal cell)

Artificial (Sigma Unit)

\[
y = \Phi \left( \sum_{j} w_j x_j \right)
\]
Separation of Input Patterns

- Dot product $\mathbf{w} \cdot \mathbf{x}$ separates the input space into two regions: one with value $\geq 0$ and one with value $< 0$
- Separation is a line, defined by the weights and bias $\mathcal{J}$

$$w_1 x_1 + w_2 x_2 - \mathcal{J} = 0$$

$$x_2 = \frac{\mathcal{J}}{w_2} - \frac{w_1}{w_2} x_1$$

- $\mathcal{J} = 0$
- $\mathcal{J} > 0$
Generalization

x_2

x_1

Probably bad
Generalization

Probably good
XOR Problem

- Boolean XOR function is not linearly separable

- If we could use two hyper planes, we could separate one class from both sides

- This can be accomplished by a Multi-Layer Perceptron

- Problem: How to train multiple layers?
Backpropagation of Error

- Forward propagation of activity
- Backward propagation of error gradient
- Weight update by gradient descent

\[ o_{k}^{\text{out}} = \Phi(a_{k}^{\text{out}}) \]
\[ a_{k}^{\text{out}} = \sum_{j=1}^{m} w_{jk}^{\text{out}} o_{j}^{\text{hid}} \]
\[ o_{j}^{\text{hid}} = \Phi(a_{j}^{\text{hid}}) \]
\[ a_{j}^{\text{hid}} = \sum_{i=1}^{l} w_{ij}^{\text{hid}} o_{i}^{\text{in}} \]
\[ o_{i}^{\text{in}} = x_{i} \]

\[ \delta_{k}^{\text{out}} = \Phi'(a_{k}^{\text{out}})(o_{k}^{\text{out}} - t_{k}) \]
\[ \delta_{j}^{\text{hid}} = \Phi'(a_{j}^{\text{hid}})\sum_{k=1}^{n} w_{jk}^{\text{out}} \delta_{k}^{\text{out}} \]
\[ \Delta w_{jk}^{\text{out}} = -\eta a_{j} \delta_{k}^{\text{out}} ; \]
\[ \Delta w_{ij}^{\text{hid}} = -\eta o_{i} \delta_{j}^{\text{hid}} ; \]
Flat vs. Deep Networks

- A neural network with a single hidden layer that is wide enough can compute any function (Cybenko, 1989)
  - Certain functions, like parity, may require exponentially many hidden units (in the number of inputs)
- Deep networks (with multiple hidden layers) may be exponentially more efficient
  - Parity example: Compute carry bit sequentially
Learning a Feature Hierarchy

[Behnke, IJCNN’99]

32x32 x 4
16x16 x 8
8x8 x 16
4x4 x 32
2x2 x 64
1x1 x 128

Step edges
Lines
Curves
Parts
Digits
Digit Reconstruction

[Behnke, IJCAI’01]

Degradation

Sven Behnke: Deep Learning for Visual Perception
Digit Reconstruction

[Behnke, IJCAI’01]

Degradation

Input

Output

Target

Sven Behnke: Deep Learning for Visual Perception
Binarization of Matrix Codes

Original → Degraded → Target

[Behnke, ICANN 2003]

Output

Hidden
Face Localization

- BioID data set:
  - 1521 images
  - 23 persons

- Encode eye positions with blobs

[Behnke, KES’03]
Face Localization

[Behnke, KES’03]
Auto-Encoder

- Try to push input through a bottleneck

- Activities of hidden units form an efficient code
  - There is no space for redundancy in the bottleneck

- Extracts frequently independent features (factorial code)

Desired Output = Input

Input vector

Code

Output vector

Sven Behnke: Deep Learning for Visual Perception
Deep Autoencoders
(Hinton & Salakhutdinov, 2006)

- Multi-layer autoencoders for non-linear dimensionality reduction
- Difficult to optimize deep autoencoders using backpropagation
- Greedy, layer wise training
- Unrolling
- Supervised fine-tuning

Sven Behnke:
Deep Learning for Visual Perception
MNIST Digits
entirely unsupervised except for the colors
GPU Implementations (CUDA)

- Affordable parallel computers
- General-purpose programming
- Convolutional  
  [Scherer & Behnke, 2009]

- Local connectivity  
  [Uetz & Behnke, 2009]
Image Categorization: NORB

- 10 categories, jittered-cluttered

- **Max-Pooling**, cross-entropy training

- Test error: 5.6% (LeNet7: 7.8%)

[Scherer, Müller, Behnke, ICANN’10]
Image Categorization: LabelMe

- 50,000 color images (256x256)
- 12 classes + clutter (50%)

Error TRN: 3.77%; TST: 16.27%
Recall: 1,356 images/s

[Uetz, Behnke, ICIS2009]
Multi-Column Deep Convolutional Networks

- Different preprocessings
- Trained with distortions
- Bagging deep networks

- MNIST: 0.23%
- NORB: 2.7%
- CIFAR10: 11.2%
- Traffic signs: 0.54% test error

[Ciresan et al. CVPR 2012]
ImageNet Challenge

- 1.2 million images
- 1000 categories, no overlap
- Subset of 11 million images from 15,000+ categories
- Hierarchical category structure (WordNet)

Task: recognize object category
- Low penalty for extra detections
- Hierarchical error computation

Golf cart (motor vehicle, self-propelled vehicle, wheeled vehicle, ...)
Egyptian cat (domestic cat, domestic animal, animal)
Large Unsupervised Feature Learning

- 9 layer model
- Locally connected
- Sparse auto-encoder
- L2 pooling
- Local contrast normalization
- 1 billion connections
- Trained on 10 million images
- Unsupervised learned detectors

Supervised ImageNet 2011 results (14M images, 22K categories): 15.8%

[Le et al. 2012]
Large Convolutional Network

- Rectifying transfer functions
- 650,000 neurons
- 60,000,000 parameters
- 630,000,000 connections
- Trained using dropout and data augmentation
- Testing 10 sub-images
- ILSVRC-2012: top-5 error 15.3%

[Image of a diagram showing the architecture of the network with labels for layers and connections, and a grid of learned low-level filters]

[Krizhevsky et al. NIPS 2012]
Validation Classification

Krizhevsky et al. NIPS 2012

Sven Behnke: Deep Learning for Visual Perception
Surpassing Human Performance

[He et al. 2015]
Sven Behnke: Deep Learning for Visual Perception
Object-class Segmentation

- Class annotation per pixel
- Multi-scale input channels
- Evaluated on MSRC-9/21 and INRIA Graz-02 data sets
Object Detection in Images

- Bounding box annotation
- Structured loss that directly maximizes overlap of the prediction with ground truth bounding boxes
- Evaluated on two of the Pascal VOC 2007 classes

[Schulz, Behnke, ICANN 2014]
RGB-D Object-Class Segmentation

- Kinect-like sensors provide dense depth
- Scale input according to depth, compute pixel height

NYU Depth V2

<table>
<thead>
<tr>
<th>Method</th>
<th>floor</th>
<th>struct</th>
<th>furnit</th>
<th>prop</th>
<th>Class Avg.</th>
<th>Pixel Acc.</th>
</tr>
</thead>
<tbody>
<tr>
<td>CW</td>
<td>84.6</td>
<td>70.3</td>
<td>58.7</td>
<td>52.9</td>
<td>66.6</td>
<td>65.4</td>
</tr>
<tr>
<td>CW+DN</td>
<td>87.7</td>
<td>70.8</td>
<td>57.0</td>
<td>53.6</td>
<td>67.3</td>
<td>65.5</td>
</tr>
<tr>
<td>CW+H</td>
<td>78.4</td>
<td>74.5</td>
<td>55.6</td>
<td>62.7</td>
<td>67.8</td>
<td>66.5</td>
</tr>
<tr>
<td>CW+DN+H</td>
<td>93.7</td>
<td>72.5</td>
<td>61.7</td>
<td>55.5</td>
<td>70.9</td>
<td>70.5</td>
</tr>
<tr>
<td>CW+DN+H+SP</td>
<td>91.8</td>
<td>74.1</td>
<td>59.4</td>
<td>63.4</td>
<td>72.2</td>
<td>71.9</td>
</tr>
<tr>
<td>CW+DN+H+CRF</td>
<td>93.5</td>
<td>80.2</td>
<td>66.4</td>
<td>54.9</td>
<td>73.7</td>
<td>73.4</td>
</tr>
<tr>
<td>Müller et al.[8]</td>
<td>94.9</td>
<td>78.9</td>
<td>71.1</td>
<td>42.7</td>
<td>71.9</td>
<td>72.3</td>
</tr>
<tr>
<td>Random Forest [8]</td>
<td>90.8</td>
<td>81.6</td>
<td>67.9</td>
<td>19.9</td>
<td>65.1</td>
<td>68.3</td>
</tr>
<tr>
<td>Couprie et al.[9]</td>
<td>87.3</td>
<td>86.1</td>
<td>45.3</td>
<td>35.5</td>
<td>63.6</td>
<td>64.5</td>
</tr>
<tr>
<td>Höft et al.[10]</td>
<td>77.9</td>
<td>65.4</td>
<td>55.9</td>
<td>49.9</td>
<td>62.3</td>
<td>62.0</td>
</tr>
<tr>
<td>Silberman [12]</td>
<td>68</td>
<td>59</td>
<td>70</td>
<td>42</td>
<td>59.7</td>
<td>58.6</td>
</tr>
</tbody>
</table>

CW is covering windows, H is height above ground, DN is depth normalized patch sizes. SP is averaged within superpixels and SVM-reweighted. CRF is a conditional random field over superpixels [8]. Structure class numbers are optimized for class accuracy.

[Schulz, Höft, Behnke, ESANN 2015]
Neural Abstraction Pyramid for RGB-D Video Object-class Segmentation

- NYU Depth V2 contains RGB-D video sequences
- Recursive computation is efficient for temporal integration

NYU Depth V2 contains RGB-D video sequences

Recursive computation is efficient for temporal integration

NYU Depth V2 contains RGB-D video sequences

Recursive computation is efficient for temporal integration

RGB Depth Output Truth

[Pavel, Schulz, Behnke, IJCNN 2015]
Geometric and Semantic Features for RGB-D Object-class Segmentation

- **New geometric feature**: distance from wall
- **Semantic features** pretrained from ImageNet
- Both help significantly

[Husain et al. under review]
Semantic Segmentation Priors for Object Discovery

- Combine bottom-up object discovery and semantic priors
- Semantic segmentation used to classify color and depth superpixels
- Higher recall, more precise object borders

[Garcia et al. under review]
RGB-D Object Recognition and Pose Estimation

- Use pretrained features from ImageNet

[Schwarz, Schulz, Behnke, ICRA2015]
Canonical View, Colorization

- Objects viewed from different elevation
- Render canonical view
- Colorization based on distance from center vertical

[Schwarz, Schulz, Behnke, ICRA2015]
Features Disentangle Data

- t-SNE embedding

[Schwarz, Schulz, Behnke ICRA2015]
Recognition Accuracy

- Improved both category and instance recognition

<table>
<thead>
<tr>
<th>Method</th>
<th>Category Accuracy (%)</th>
<th>Instance Accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>RGB</td>
<td>RGB-D</td>
</tr>
<tr>
<td>Lai et al. [1]</td>
<td>74.3 ± 3.3</td>
<td>81.9 ± 2.8</td>
</tr>
<tr>
<td>Bo et al. [2]</td>
<td>82.4 ± 3.1</td>
<td>87.5 ± 2.9</td>
</tr>
<tr>
<td>PHOW[3]</td>
<td>80.2 ± 1.8</td>
<td>—</td>
</tr>
<tr>
<td>Ours</td>
<td><strong>83.1 ± 2.0</strong></td>
<td><strong>88.3 ± 1.5</strong></td>
</tr>
<tr>
<td>Ours</td>
<td><strong>83.1 ± 2.0</strong></td>
<td><strong>89.4 ± 1.3</strong></td>
</tr>
</tbody>
</table>

Confusion

1: pitcher / coffe mug
2: peach / sponge

[Schwarz, Schulz, Behnke, ICRA2015]
Generating Image Captions

- Multimodal recurrent neural network generative model

[Karpathy, Fei-Fei 2015]
Generating Image Captions

A group of people shopping at an outdoor market. There are many vegetables at the fruit stand.

A skateboarder does a trick on a ramp.

A dog is jumping to catch a frisbee.

A little girl in a pink hat is blowing bubbles.

A refrigerator filled with lots of food and drinks.

A herd of elephants walking across a dry grass field.

Two hockey players are fighting over the puck.

A close up of a cat laying on a couch.

A red motorcycle parked on the side of the road.

A yellow school bus parked in a parking lot.

[Vinyals et al. 2015]
Dreaming Deep Networks

[Mordvintsev et al 2015]
Painting Style Transfer

Turner

van Gogh

Munch

Original

Turner

van Gogh

Munch

[Gatys et al. 2015]
Conclusion

- Flat models do not suffice
- Jump from signal to symbols too large
- Deep learning helps here:
  - **Hierarchical, locally connected** models
  - **Non-linear** feature extraction
- **Structure** of learning machine does matter
- Proposed architectures map well to **GPUs**
- **Iterative interpretation** uses partial results as context to resolve ambiguities
- Many questions open
  - Graphical models vs. neural networks
  - Structured vs. unstructured modelling
  - Stability of recurrent networks
Presentation 1
Gregoire Montavon (TU Berlin): Deep Learning of Molecular Properties in the Chemical Compound Space

- Use deep neural networks as a non-linear function approximator in chemistry
- Targets computed by slow conventional method
- Can compute molecular properties of similar molecules quickly
- Application: Search compounds by property
Material recognition is instance of image categorization

Supervised training of deep convolutional networks

Reaches human performance

Seems to work different than human visual system