A Shallow Introduction into the Deep Machine Learning

Jan Čech
What is the “Deep Learning”?

- Deep learning
  = both the classifiers and the features are learned automatically

- Typically not feasible, due to high dimensionality

- Suboptimal, requires expert knowledge, works in specific domain only

Deep neural network

(e.g. SIFT, SURF, HOG, or MFCC in audio)

- Features
  (e.g. SIFT, SURF, HOG, or MFCC in audio)
Deep learning successes

- Deep learning methods have been extremely successful recently
  - Consistently beating state-of-the-art results in many fields, winning many challenges by a significant margin

Computer vision:
- Hand writing recognition, Action/activity recognition, Face recognition
- Large-scale image category recognition (ILSVRC’ 2012 challenge)

<table>
<thead>
<tr>
<th>Institution</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>INRIA/Xerox</td>
<td>33%</td>
</tr>
<tr>
<td>Uni Amsterdam</td>
<td>30%</td>
</tr>
<tr>
<td>Uni Oxford</td>
<td>27%</td>
</tr>
<tr>
<td>Uni Tokyo</td>
<td>26%</td>
</tr>
<tr>
<td><strong>Uni Toronto</strong></td>
<td><strong>16% (deep neural network)</strong> [Krizhevsky-NIPS-2012]</td>
</tr>
</tbody>
</table>

Automatic speech recognition:
- TIMIT Phoneme recognition, speaker recognition, lip reading

Natural Language Processing, Text Analysis:
- IBM Watson, Google translate
Learning the representation – Sparse coding

- Natural image statistics
  - Luckily, there is a redundancy in natural images
  - Pixel intensities are not i.i.d. (but highly correlated)

- Sparse coding [Olshausen-1996, Ng-NIPS-2006]

Input images: $x^{(1)}, x^{(2)}, \ldots, x^{(m)}; (x^{(i)} \in \mathbb{R}^{n \times n})$

Learn dictionary of basis functions $\phi_1, \phi_2, \ldots, \phi_k; (\phi_j \in \mathbb{R}^{n \times n})$ that

$$x \approx \sum_{j=1}^{k} a_j \phi_j \quad \text{s.t.} \quad a_j \quad \text{are mostly zero, \text{``sparse''}}$$

$$\min_{a, \phi} \sum_{i=1}^{m} \left( \left\| x^{(i)} - \sum_{j=1}^{k} a_j^{(i)} \phi_j \right\|^2 + \lambda \sum_{j=1}^{k} |a_j^{(i)}| \right)$$
Sparse coding

Natural Images

Test example

Learned bases ($\phi_1, \ldots, \phi_{64}$): “Edges”

$\approx 0.8 \ast \phi_{36} + 0.3 \ast \phi_{42} + 0.5 \ast \phi_{63}$

$[0, 0, \ldots, 0, 0.8, 0, \ldots, 0, 0.3, 0, \ldots, 0, 0.5, \ldots]$

$= [a_1, \ldots, a_{64}]$ (feature representation)

Compact & easily interpretable
Unsupervised Learning Hierarchies of features

- Many approaches to unsupervised learning of feature hierarchies
  - Sparse Auto-encoders [Bengio-2007]
  - Restricted Boltzmann Machines [Hinton-2006]
- These models can be stacked: lower hidden layer is used as the input for subsequent layers.
- The hidden layers are trained to capture higher-order data correlations.
- Learning the hierarchies and classification can be implemented by a (Deep) Neural Network

[Lee-ICML-2009]
Resemblance to sensory processing in the brain

- Needless to say that the brain is a neural network

- Primary visual cortex V1
  - Neurophysiological evidences that primary visual cells are sensitive to the orientation and frequency (Gabor filter like impulse responses)
  - [Hubel-Wiesel-1959] (Nobel Price winners)
    - Experiments on cats with electrodes in the brain

- A single learning algorithm hypothesis?
  - “Rewiring” the brain experiment [Sharma-Nature-2000]
    - Connecting optical nerve into A1 cortex (a subject was able to solve visual tasks by using the processing in A1)
(Artificial) Neural Networks

- Neural networks are here for more than 50 years
  - Rosenblatt-1956 (perceptron)
  - Minsky-1969 (xor issue, => skepticism)
Neural Networks

Rumelhart and McClelland – 1986:
- Multi-layer perceptron,
- Back-propagation (supervised training)
  - Differentiable activation function
  - Stochastic gradient descent

Empirical risk

\[ Q(w) = \sum_{i=1}^{n} Q_i(w), \]

Update weights:

\[ w := w - \alpha \nabla Q_i(w). \]

What happens if a network is deep?
(it has many layers)
What was wrong with back propagation?

- Local optimization only (needs a good initialization, or re-initialization)
- Prone to over-fitting
  - too many parameters to estimate
  - too few labeled examples
- Computationally intensive

=> Skepticism: A deep network often performed worse than a shallow one
What was wrong with backpropagation?

However nowadays:

- Weights can be initialized better (Use of unlabeled data)
- Large collections of labeled data available
  - ImageNet (14M images, 21k classes, hand-labeled)
- Reducing the number of parameters by weight sharing
  - Convolutional layers – [LeCun-1989]
- Novel tricks to prevent overfitting of deep nets
- Fast enough computers (parallel hardware, GPU)

=> Optimism: It works!
Deep convolutional neural networks

- An example for Large Scale Classification Problem:
    - Recognizes 1000 categories from ImageNet
    - Outperforms state-of-the-art by significant margin (ILSVRC 2012)

- 5 convolutional layers, 3 fully connected layers
- 60M parameters, trained on 1.2M images (~1000 examples for each category)
Deep convolutional neural networks

- Additional tricks: “Devil is in the details”
  - Rectified linear units instead of standard sigmoid
    => Mitigate vanishing gradient problem
  - Convolutional layers followed by max-pooling
    • Local maxima selection in overlapping windows (subsampling)
    => dimensionality reduction, shift insensitivity
  - Dropout
    • Averaging results of many independent models (similar idea as in Random forests)
    • 50% of hidden units are randomly omitted during the training, but weights are shared in testing time
    => Probably very significant to reduce overfitting
  - Data augmentation
    • Images are artificially shifted and mirrored (10 times more images)
    => transformation invariance, reduce overfitting
Deep convolutional neural networks

- No unsupervised pre-initialization!
  - The training is supervised by standard back-propagation
  - Enough labeled data: 1.2M labeled training images for 1k categories
  - Learned filters in the first layer
    - Resemble cells in primary visual cortex

- Training time:
  - 5 days on NVIDIA GTX 580, 3GB memory (Krizhevsky, today faster)
  - 90 cycles through the training set

- Test time (forward step) on GPU
  - 5 ms/image in a batch mode
  - (My experience: 100 ms/image in Matlab, including image decompression and normalization)
Preliminary experiments 1: Category recognition

  - Network pre-trained for 1000 categories provided
- Which categories are pre-trained?
  - 1000 “most popular” (probably mostly populated)
  - Typically very fine categories (dog breeds, plants, vehicles…)
  - Category “person” (or derived) is missing
  - Recognition subjectively surprisingly good…

![Diagram showing category recognition results](http://cmp.felk.cvut.cz/~cechj/tmp/Tristan.jpg)
Sensitivity to image rotation
Sensitivity to image blur
It is not a texture only...
Preliminary experiments 2: Category retrieval

- 50k randomly selected images from Profimedia dataset
- Category: Ocean liner
Preliminary experiments 2: Category retrieval

- Category: Restaurant (results out of 50k-random-Profiset)
Preliminary experiments 2: Category retrieval

- Category: stethoscope (results out of 50k-random-Profiset)
Preliminary experiments 3: Similarity search

- Indications in the literature that the last hidden layer carry semantics
  - Last hidden layer (4096-dim vector), final layer category responses (1000-dim vector)
  - New (unseen) categories can be learned by training (a linear) classifier on top of the last hidden layer
    - Oquab, Bottou, Laptev, Sivic, CVPR, 2014
    - Girshick, Dphanue, Darell, Malik, CVPR, 2014
  - Responses of the last hidden layer can be used as a compact global image descriptor
    - Semantically similar images should have small Euclidean distance
Preliminary experiments 3: Similarity search

- Qualitative comparison: (20 most similar images to a query image)
    - Nearest neighbour search in 20M images of Profimedia
    - Standard global image statistics (e.g. color histograms, gradient histograms, etc.)
  2. Caffe NN (last hidden layer response + Euclidean distance),
    - Nearest neighbour search in 50k images of Profimedia

MUFIN results
Preliminary experiments 3: Similarity search

Caffe NN results

1: 0
2: 3614.19
3: 4036.38
4: 4120.14
5: 4645.88
6: 4768.97
7: 4861.46
8: 5014.75
9: 5024.26
10: 5156.86
11: 5186.45
12: 5228.55
13: 5300.27
14: 5339.76
15: 5343.16
16: 5391.45
17: 5430.03
18: 5463.58
19: 5472.23
20: 5480.98
Preliminary experiments 3: Similarity search

MUFIN results
Preliminary experiments 3: Similarity search

Caffe NN results
Preliminary experiments 3: Similarity search

MUFIN results
Preliminary experiments 3: Similarity search

Caffe NN results
Preliminary experiments 3: Similarity search

MUFIN results
Preliminary experiments 3: Similarity search

Caffe NN results

1: 0
2: 2184.64
3: 2542.78
4: 2621.33
5: 2671.85
6: 2768.1
7: 2861.94
8: 2880.53
9: 3021.58
10: 3071.51
11: 3075.16
12: 3083.7
13: 3137.8
14: 3191.61
15: 3208.09
16: 3211.58
17: 3212.02
18: 3216.26
19: 3217.13
20: 3221.93
Preliminary experiments 3: Similarity search

MUFIN results
Preliminary experiments 3: Similarity search

Caffe NN results

1: 0
2: 2812.02
3: 2968.18
4: 3189.3
5: 3284.86
6: 3286.28
7: 3304.93
8: 3402.86
9: 3433.69
10: 3473.81
11: 3495.67
12: 3528.47
13: 3549.56
14: 3559.5
15: 3562.74
16: 3574.01
17: 3576.81
18: 3597.88
19: 3599.39
20: 3662.85
Preliminary experiments 3: Similarity search

MUFIN results
Preliminary experiments 3: Similarity search

Caffe NN results
Preliminary experiments 3: Similarity search

MUFIN results
Preliminary experiments 3: Similarity search

Caffe NN results
Preliminary experiments 3: Similarity search

MUFIN results
Preliminary experiments 3: Similarity search

Caffe NN results
Multiple object classes
Multiple object classes
Object detection: Deep Nets and Sliding Windows

- An image of a scene contains multiple objects
- Exhaustive sliding window detector is prohibitively slow

=> Category independent region proposals:
  - Objectness [Alexe-TPAMI-2012]
  - Selective search [Uijlings-IJCV-2013]
  - Edgeboxes [Zitnick-ECCV-2014]

More recent approaches, see next lecture
Novel tricks

- Network initialization
  - Mishkin, Matas. *All you need is a good init.* ICLR 2016
  - Weights initialization: zero mean, unit variance, orthogonality

- Batch normalization
  - Zero mean and unit variance weights are “supported” during training to avoid vanishing gradient

⇒ Small sensitivity to learning rate setting (can be higher, faster training – 10 times fewer epochs needed)
⇒ Regularizer (dropout can be excluded/smaller) (better optimum found)
Novel tricks II.

- Exponential Linear Units (ELU) [Clevert et al., ICLR 2016]
  - Self normalizing properties, batch normalization unnecessary
  - Faster training reported

- ADAM optimizer [Kingma and Ba, ICLR 2015]
  = (ADAptive moments)
  - Often improves over SGD (with momentum),
  - Low sensitivity on learning rate setting
Novel architectures

- ImageNet Large Scale Visual Recognition Challenge (ILSVRC)

HoG + DPM  CNN

AlexNet  VGG Net  GoogLeNet  ResNet

classification error %


28.2  25.8  16.4  11.7  7.3  6.7  3.57  5.1

number of layers


0  50  100  150

=> “Go deeper”
CNN architectures

- **AlexNet**
  - [Krizhevsky et al., NIPS 2012]

```
11x11 conv, 96, /4, pool/2
  ↓
5x5 conv, 256, pool/2
  ↓
3x3 conv, 256, pool/2
  ↓
3x3 conv, 384
  ↓
3x3 conv, 384
  ↓
3x3 conv, 256, pool/2
  ↓
fc, 4096
  ↓
fc, 4096
  ↓
fc, 1000
```
CNN architectures

- VGG Net: VGG-16, VGG-19
  - [Simonyan and Zisserman, ICLR 2015]
  - Deeper than AlexNet
  - Smaller filters (3x3 convolutions), more layers
    => Same effective receptive field,
    but more “non-linearity
CNN architectures

- GoogLeNet
  - [Szegedy et al., CVPR 2015]
  - 22 layers, No Fully-Connected layers
  - Accurate, much less parameters
  - “Inception” module (Net in net)
CNN architectures

- **ResNet**
  - [He et al., CVPR 2016]
    
  => Plain deeper models are not better (vanishing gradient)

- Residual modules, 152 layers

\[
H(x) = F(x) + x
\]
CNN models (comparison)

Face interpretation tasks

- Face recognition, face verification
  - Architecture similar to AlexNet - very deep CNN (softmax at the last layer)

[Taigman-ECVV-2014] DeepFace: Closing the Gap to Human-Level Performance in Face Verification (authors from Facebook)
[Parkhi-BMVC-2015] Deep Face recognition (authors from Oxford Uni)
  - 2.6M images of 2.6k celebrities, trained net available

- Face represented by penultimate layer response, similarity search, large scale indexing
Face interpretation tasks

- Facial landmarks, Age / Gender estimation
  - Multitask network
    - Shared representation
    - Combination of both classification and regression problems
Age estimation – How good the network is?

- Our survey
  ~20 human subjects, ~100 images of 2 datasets

**MORPH dataset**
- True: 22, MAE: 18.8
- True: 36, MAE: 17.8
- True: 33, MAE: 16.3
- True: 22, MAE: 16.1
- True: 25, MAE: 16.0

**IMDB dataset**
- True: 25, MAE: 0.5
- True: 66, MAE: 1.0
- True: 29, MAE: 1.0
- True: 19, MAE: 1.0
- True: 43, MAE: 1.0
Age estimation – How good the network is?

- Better than average human…

```
Average human : 6.8  48.6   24.1
Human crowd   : 4.7  65.1   19.0
Machine       : 3.2  82.6   26.0
```

```
Average human : 8.2  41.7   31.5
Human crowd   : 5.7  59.0   21.0
Machine       : 5.1  62.5   42.7
```

- [Franc-Cech-BwildW-2017]
- Network runs real-time on CPU
Predicting Decision Uncertainty from Faces

- [Jahoda, Vobecky, Cech, Matas. *Detecting Decision Ambiguity from Facial Images*. In Face and Gestures, 2018]

- Can we train a classifier to detect uncertainty?

=> YES, we can…

- CNN 25% error rate, while human volunteers 45%

Training set: 1,628 sequences
Test set: 90 sequences
Sexual Orientation from Face Images

- Better accuracy than human in (gay vs. heterosexual)
  - 81% accuracy (for men), average human accuracy (61%)
  - 71% accuracy (for women) average human accuracy (54%)
  - Accuracy further improved if 5 images provided (91%, 83%)
General recipe to use deep neural networks

**Recipe to use deep neural network to “solve any problem” (G. Hinton 2013)**

- Have a deep net
- If you do not have enough labeled data, pre-train it by unlabeled data; otherwise do not bother with pre-initialization
- Use rectified linear units instead of standard neurons (sigmoid)
- Use dropout to regularize it (you can have many more parameters than training data)
- If there is a spatial structure in your data, use convolutional layers

**Novel:**

- Use Batch Normalization [Ioffe-Szegedy-NIPS-2015]
- ReLU => ELU
- Adaptive Optimizers (ADAM)
- Various architectures (AlexNet, VGG, GoogLeNet, ResNet)

**Experience:**

- Data matters (the more data the better), data augmentation helps
Conclusions

- CNNs efficiently learn the abstract representation (shared among classes)
- Low computational demands for running, Training needs GPU
- Many “deep” toolboxes: Caffe (Berkeley), MatconvNet (Oxford), TensorFlow (Google), Theano (Montreal), Torch, …
- NNs are (again) in the “Golden Age” (or witnessing a bubble), as many practical problems seem solvable in near future
- Explosion of interest of DNN in literature, graduates get incredible offers, start-ups appear all the time

- Do we understand enough what is going on?
  http://www.youtube.com/watch?v=LVLoc6FrLi0

Acknowledgement: I borrowed some images from slides of G. Hinton, A. Ng, Y. Le Cun, Fei-Fei Li, K. He.
Further Resources

- Deep Learning Textbook
  - Ian Goodfellow and Yoshua Bengio and Aaron Courville, Deep Learning, MIT Press, 2016
  - Available on-line for free.

- Lectures / video-lectures
  - Stanford University course on Deep Learning (cs231n)
  - MIT lectures on Introduction in Deep Learning (MIT 6.S191)

- Various blogs and on-line journals
  - Andrej Karpathy (blog)
  - Distill (distill.pub)