Intro to Deep Convolutional Networks

Vít Listík (tivvit@seznam.cz)
Recent usages

- Self Driving Cars [https://youtu.be/tiwVMrTLUWg?t=7m39s](https://youtu.be/tiwVMrTLUWg?t=7m39s)
- Games [https://youtu.be/qv6UVOQ0F44?t=1m5s](https://youtu.be/qv6UVOQ0F44?t=1m5s)
- AlphaGo
- ImageNet, MNIST
- SyntaxNet
- Word embeddings
- Image Colorization
- Medical applications
- Spoken language understanding
- Generative (adversarial) networks
- Q&A
- Image Analogies
ILSVRC top-5 error on ImageNet

“Deep Image: Scaling up Image Recognition”
   — Baidu: 5.98%, Jan. 13, 2015

“Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification”
   — Microsoft: 4.94%, Feb. 6, 2015

   — Google: 4.82%, Feb. 11, 2015
History

● 1943 - Computational model
● 1948 - Turing B-type machines (models) - Hebbian Learning
● 1957 - HW machine - image recognition
● 1969 - a famous book entitled *Perceptrons* by Marvin Minsky and Seymour Papert showed that it was impossible for these classes of network to learn an XOR function.
● 1975 - XOR demystified + backpropagation
● 2009 - RNN won several contests
Perceptron (Adaline)
Perceptron

- Binary classifier (dichotomy)
- Linearly separable problems
- Dot product (GPU)

\[ y = x_1 w_1 + x_2 w_2 \]

\[ \sum_{i=0}^{m} w_i x_i \]

\[ f(x) = \begin{cases} 
1 & \text{if } w \cdot x + b > 0 \\ 
0 & \text{otherwise} 
\end{cases} \]
Neural networks

input layer

hidden layer 1  hidden layer 2

output layer
NN

- Biologically inspired
- Multi layer perceptron
- Feedforward (DAG) x recurrent
- Hyperparameters
  - Structure
  - Activation function
NN vs SVM (discussion)

- Global optimum
- SVM maximizes the margin
- Perceptron minimizes the error
- SVM < 10K samples
- NN > 10K samples
- Online learning
- Overfitting
Learning

Cost function (loss function)

\[ f^* \in F - \text{optimal solution} \]

\[ f^*, C(f^*) \leq C(f) \forall f \in F. \]

- Validation set
- Batches
- Backpropagation
Backpropagation

- Propagation of errors
  - finding differences between target and actual value (delta)
- Updating the weights - minimizing the error
- Stochastic gradient descent
- Chain rule - derivative of functions (squared error)
- Vanishing gradient problem (exponential decay)
# Activation functions

<table>
<thead>
<tr>
<th>Name</th>
<th>Plot</th>
<th>Equation</th>
<th>Derivative</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Identity</td>
<td></td>
<td>( f(x) = x )</td>
<td>( f'(x) = 1 )</td>
<td>((-\infty, \infty))</td>
</tr>
<tr>
<td>Binary step</td>
<td></td>
<td>( f(x) = \begin{cases} 0 &amp; \text{for } x &lt; 0 \ 1 &amp; \text{for } x \geq 0 \end{cases} )</td>
<td>( f'(x) = \begin{cases} 0 &amp; \text{for } x \neq 0 \ ? &amp; \text{for } x = 0 \end{cases} )</td>
<td>([0, 1])</td>
</tr>
<tr>
<td>Logistic (a.k.a Soft step)</td>
<td></td>
<td>( f(x) = \frac{1}{1 + e^{-x}} )</td>
<td></td>
<td>((0, 1))</td>
</tr>
<tr>
<td>TanH</td>
<td></td>
<td>( f(x) = \tanh(x) = \frac{2}{1 + e^{-2x}} - 1 )</td>
<td>( f'(x) = 1 - f(x)^2 )</td>
<td>((-1, 1))</td>
</tr>
<tr>
<td>ArcTan</td>
<td></td>
<td>( f(x) = \tan^{-1}(x) )</td>
<td>( f'(x) = \frac{1}{x^2 + 1} )</td>
<td>((-\frac{\pi}{2}, \frac{\pi}{2}))</td>
</tr>
<tr>
<td>Softsign [7]</td>
<td></td>
<td>( f(x) = \frac{x}{1 +</td>
<td>x</td>
<td>} )</td>
</tr>
<tr>
<td>Rectified Linear Unit (ReLU) [8]</td>
<td></td>
<td>( f(x) = \begin{cases} 0 &amp; \text{for } x &lt; 0 \ x &amp; \text{for } x \geq 0 \end{cases} )</td>
<td>( f'(x) = \begin{cases} 0 &amp; \text{for } x &lt; 0 \ 1 &amp; \text{for } x \geq 0 \end{cases} )</td>
<td>([0, \infty))</td>
</tr>
</tbody>
</table>
Example

http://playground.tensorflow.org/
Deep neural nets

- Replacing handcrafted features
- Capable of learning complex tasks
- Tradeoff: a lot of examples, computationally expensive
- from low-level to high-level features
- Deep?
  - 1971 - 8 layers
  - 2015 - thousands of layers
- Overfitting $\rightarrow$ regularization
  - Weight decay
  - Dropout
Dropout

(a) Standard Neural Net

(b) After applying dropout.
Recurrent - RNN

- Language
- Speech

LSTM
Convolutional (CNN, ConvNet)

- Overlapping regions
- Computational effectiveness
- Raw pixels

- Kernels (filters)
- Local connectivity
- (Max) pooling
Kernels
Max pooling
Fine tuning

- Training pre-trained networks
- Smaller amount of samples
- Adjusted learning rate

https://github.com/BVLC/caffe/tree/master/examples/finetune_flickr_style
Frameworks

- Caffee
- Theano
- Torch
- TensorFlow
- Keras
- ...

25
Example

https://github.com/tivvit/ML-Prague-2016-email-workshop/blob/master/keras-example.ipynb
Sources

- https://docs.google.com/presentation/d/11fNaOAd1cUiTlxWVDlpEEFB4bgILW6qVLwFYT_RNUJQ/edit
- http://colah.github.io/posts/2015-08-Understanding-LSTMs/
- https://www.youtube.com/watch?v=GlcnxUIrtek
- http://ufldl.stanford.edu/tutorial/supervised/ConvolutionalNeuralNetwork/