MatConvNet
Deep learning research in MATLAB

Dr Andrea Vedaldi
University of Oxford

MATLAB Expo, October 2017
Deep learning: a magic box

Pixels & labels in, model parameters out

\[ \text{bike} \]
Confounding factors

- Fonts
- Distortions
- Colors
- Blur
- Shadows
- Borders
- Textures
- Sizes
...
Fast retrieval, learn concepts on the fly

BBC News Search

Search results page 1 of 100 (5,000 results)

Ranked in 1.81s
Object detection

Single shot (feed forward) detector
Dense part and keypoint labelling
Demos
Big Data + GPU Compute + Optimisation

A few million labelled images

A few hundred teraflops of compute capability

A few dozen grad students

Dark magic
Convolutional neural networks

A composition of parametric linear and non-linear operators

$\text{Tensor data} \quad = \quad \text{RGB} \quad = \quad \text{Tensor} \quad = \quad \text{height} \times \text{width} \times \text{channels}$
Operators (aka layers)

Linear convolution

- Filter bank
  - several filters
  - each generating an output channel

- Tensor input-output
  - big filters
  - multi-dimensional

Non-linear activation

- Simple non-linear functions
  - \( \max\{0, x\} \)
How deep is deep enough?

AlexNet (2012)

5 convolutional layers

3 fully-connected layers
How deep is deep enough?

How deep is deep enough?

- AlexNet (2012)
- VGG-M (2013)
- VGG-VD-16 (2014)
- GoogLeNet (2014)
How deep is deep enough?

- AlexNet (2012)
- VGG-M (2013)
- VGG-VD-16 (2014)
- GoogLeNet (2014)
How deep is deep enough?


The need for gradients

Learning = optimise the parameters $w$ to minimise a fitting error

The error function is optimised using (stochastic) gradient descent

We require the error function derivatives
Efficient computation of the gradient

Forward pass:

\[ x \rightarrow c_1 \rightarrow c_2 \rightarrow c_3 \rightarrow c_4 \rightarrow c_5 \rightarrow f_6 \rightarrow f_7 \rightarrow f_8 \rightarrow \text{loss} \]

Backward pass:

\[ d_{\text{error}} \frac{d}{d w_1} \rightarrow d_{\text{error}} \frac{d}{d w_2} \rightarrow d_{\text{error}} \frac{d}{d w_3} \rightarrow d_{\text{error}} \frac{d}{d w_4} \rightarrow d_{\text{error}} \frac{d}{d w_5} \rightarrow d_{\text{error}} \frac{d}{d w_6} \rightarrow d_{\text{error}} \frac{d}{d w_7} \rightarrow d_{\text{error}} \frac{d}{d w_8} \]
Deep learning software

Requirements

Flexible and usable API
- Concise & powerful
- Automatic differentiation

Extensible
- Keep up with research
- Test new ideas

Efficient
- GPU
- Optimised compute graph

MATLAB
- Simple yet powerful language
- Historically, widely adopted in computer vision and robotics
- Great GPU support
- Great documentation
- Recently, native support for deep nets…
MatConvNet

The first modern deep learning toolbox in MATLAB

Why?
- Fully MATLAB-hackable
- As efficient as other tools (Caffe, TensorFlow, Torch, …)

Real-world state-of-the-art applications
- See demos
- Many more

Cutting-edge research
- 900+ citations in academic papers

Education
- Several international courses use it

Pedigree
- Spawn of VLFeat (Mark Everingham Award)
- Has been around since the “beginning” (~2012)
Deep learning sandwich

MatConvNet

Applications

MatConvNet SimpleNN
Very basic network abstraction

MatConvNet DagNN
Explicit compute graph abstraction

MatConvNet AutoNN
Implicit compute graph

MatConvNet Primitives
vl_nnconv, vl_nnpool, … (MEX/M files)

MatConvNet Kernel
GPU/CPU implementation of low-level ops

MATLAB
Parallel Computing Toolbox (GPU)

Platform (Win, macOS, Linux)

NVIDIA CUDA (GPU)  NVIDIA CuDNN (Deep Learning Primitives; optional)

Pure MATLAB code
What can you do with it

**Use a pre-trained model**
- VGG-VD, ResNet, ResNext, SSD, R-CNN, …

**Learn a new model**
- Arbitrary compute graphs
- SGD on multi GPUs

**Create new layer types**
- Native MATLAB (gpuArrays)

**Hack the compute graph**
- Visualisation, debugging, optimisations

**Hack autodiff**
- Define a new API

**Hack everything**
- Everything is open
MatConvNet

Deep learning sandwich

Applications

MatConvNet SimpleNN
Very basic network abstraction

MatConvNet DagNN
Explicit compute graph abstraction

MatConvNet AutoNN
Implicit compute graph

MatConvNet Primitives
vl_nnconv, vl_nnpool, ... (MEX/M files)

MatConvNet Kernel
GPU/CPU implementation of low-level ops

MATLAB

Parallel Computing Toolbox (GPU)

Platform (Win, macOS, Linux)

NVIDIA CUDA (GPU)

NVIDIA CuDNN (Deep Learning Primitives; optional)
MatConvNet

Deep learning sandwich

Applications

MatConvNet SimpleNN
Very basic network abstraction

MatConvNet DagNN
Explicit compute graph abstraction

MatConvNet AutoNN
Implicit compute graph

MatConvNet Primitives
vl_nncnv, vl_nnpool, … (MEX/M files)

MatConvNet Kernel
GPU/CPU implementation of low-level ops

MATLAB
Parallel Computing Toolbox (GPU)

Platform (Win, macOS, Linux)

NVIDIA CUDA (GPU)
NVIDIA CuDNN (Deep Learning Primitives; optional)
Primitive: convolution

\[
y = \text{vl\_nnconv}(x, W, b)
\]
Primitive: convolution

Forward (eval)

\[
y = \text{vl\_nnconv}(x, W, b)
\]
**Primitive: convolution**

**forward (eval)**

\[ y = \text{vl\_nnconv}(x, W, b) \]

**backward (backprop)**

\[ dzdx = \text{vl\_nnconv}(x, W, b, dzdy) \]
Primitive: convolution

**forward (eval)**

\[
y = \text{vl}\_\text{nnconv}(x, W, b)
\]

**backward (backprop)**

\[
dzdx = \text{vl}\_\text{nnconv}(x, W, b, dzdy)
\]
MatConvNet

Deep learning sandwich

Applications

MatConvNet SimpleNN
Very basic network abstraction

MatConvNet DagNN
Explicit compute graph abstraction

MatConvNet AutoNN
Implicit compute graph

MatConvNet Primitives
vl_nnconv, vl_nnpool, … (MEX/M files)

MatConvNet Kernel
GPU/CPU implementation of low-level ops

MATLAB
Parallel Computing Toolbox (GPU)

Platform (Win, macOS, Linux)

NVIDIA CUDA (GPU)

NVIDIA CuDNN (Deep Learning Primitives; optional)
MatConvNet

Deep learning sandwich

Applications

MatConvNet SimpleNN
Very basic network abstraction

MatConvNet DagNN
Explicit compute graph abstraction

MatConvNet AutoNN
Implicit compute graph

MatConvNet Primitives
vl_nnconv, vl_nnpool, … (MEX/M files)

MatConvNet Kernel
GPU/CPU implementation of low-level ops

MATLAB
Parallel Computing Toolbox (GPU)

Platform (Win, macOS, Linux)

NVIDIA CUDA (GPU)
NVIDIA CuDNN (Deep Learning Primitives; optional)
% Define network & loss
x0 = Input() ;
y = Input();
x1 = vl_nnconv(x0, 'size', [5, 5, 1, 20]) ;
x2 = vl_nnpool(x1, 2, 'stride', 2) ;
x3 = vl_nnconv(x2, 'size', [5, 5, 20, 10]) ;
loss = vl_nnloss(x3, y);
% Define compute graph
a = Input() ;
b = Input();
c = sqrt(max(a, 0) + a.*b/2) ;
**Autodiff vs symbolic differentiation**

**Why this instead of Maple / Symbolic Toolbox**

Autodiff **is not** symbolic differentiation

Autodiff
- computes derivatives numerically
- as efficiently as possible

Under the hood
- Autodiff appends a backward extension to the graph
- executing the graph computes both function and its derivative
MatConvNet vs Neural Network Toolbox

An increasingly powerful alternative

MatConvNet pre-trained models
Examples, demos, tutorials

Applications

MatConvNet SimpleNN
Very basic network abstraction

MatConvNet DagNN
Explicit compute graph abstraction

MatConvNet AutoNN
Implicit compute graph

MatConvNet Primitives
vl_nnconv, vl_nnpool, ... (MEX/M files)

MatConvNet Kernel
GPU/CPU implementation of low-level ops

MATLAB
Parallel Computing Toolbox (GPU)

Platform (Win, macOS, Linux)

NVIDIA CUDA (GPU)
NVIDIA CuDNN (Deep Learning Primitives; optional)
An increasingly powerful alternative

MatConvNet vs Neural Network Toolbox

MatConvNet pre-trained models
Examples, demos, tutorials

MatConvNet SimpleNN
Very basic network abstraction

MatConvNet DagNN
Explicit compute graph abstraction

MatConvNet AutoNN
Implicit compute graph

MatConvNet Primitives
vl_nnconv, vl_nnpool, … (MEX/M files)

MatConvNet Kernel
GPU/CPU implementation of low-level ops

Applications

MATLAB Neural Network Toolbox

MATLAB

Parallel Computing Toolbox (GPU)

Platform (Win, macOS, Linux)

NVIDIA CUDA (GPU)
NVIDIA CuDNN (Deep Learning Primitives; optional)
## New in the Neural Network Toolbox

### Data Access
- App for Ground Truth labeling
- Alexnet, VGG-16, VGG-19
- Caffe model importer

### Networks
- CNN Regression
- **Object detection** using Fast R-CNN and R-CNN
- Object detector evaluation

### Train
- Multi-GPUs in parallel
- Visual features using activations

### Deploy / Share
- Tensorflow-Keras importer
- GoogLeNet model
- Label for semantic segmentation
- Resize & augment images

<table>
<thead>
<tr>
<th>R2017a</th>
<th>R2017b</th>
</tr>
</thead>
<tbody>
<tr>
<td>App for Ground Truth labeling</td>
<td>Tensorflow-Keras importer</td>
</tr>
<tr>
<td>Alexnet, VGG-16, VGG-19</td>
<td>GoogLeNet model</td>
</tr>
<tr>
<td>Caffe model importer</td>
<td>Label for semantic segmentation</td>
</tr>
<tr>
<td>CNN Regression</td>
<td>Resize &amp; augment images</td>
</tr>
<tr>
<td><strong>Object detection</strong> using Fast R-CNN and R-CNN</td>
<td>LSTM (time series, text)</td>
</tr>
<tr>
<td>Object detector evaluation</td>
<td>DAG Networks</td>
</tr>
<tr>
<td>Multi-GPUs in parallel</td>
<td>Create new layers</td>
</tr>
<tr>
<td>Visual features using activations</td>
<td>Validation</td>
</tr>
<tr>
<td></td>
<td>Training plots</td>
</tr>
<tr>
<td></td>
<td>Hyper-parameter optimization</td>
</tr>
</tbody>
</table>

**New Product**

- GPU Coder: convert MATLAB models to NVIDIA CUDA code
MatConvNet: Check it out

http://vlfeat.org/matconvnet/

https://github.com/vlfeat/matconvnet

Karel Lenc  Sam Albanie  Joao Henriques