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

Diagram showing a series of components (C1, C2, C3, C4, C5, f6, f7, f8) connected with arrows indicating the flow of information, along with weights (W1, W2, W3, W4, W5, W6, W7, W8) pointing into each component. The output is labeled "bike."
Confounding factors

- Fonts
- Distortions
- Colors
- Blur
- Shadows
- Borders
- Textures
- Sizes
- …
Visual search

Fast retrieval, learn concepts on the fly
Single shot (feed forward) detector

Figure: Object detection with high confidence in identifying a cat.
Pose recognition

Dense part and keypoint labelling
Neural art

Real-time visual style transfer
Demos
Big Data + GPU Compute + Optimisation

A few million labelled images

A few hundred teraflops of compute capability

A few dozen grad students
A composition of parametric linear and non-linear operators

Convolutional neural networks

Tensor data

height × width × channels
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?

How deep is deep enough?

How deep is deep enough?


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

The need for gradients

The error function is optimised using (stochastic) gradient descent

We require the error function derivatives
Efficient computation of the gradient

\[ \text{error}(w_1, \ldots, w_8) \]
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)
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)

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_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)**

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

**backward (backprop)**

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

forward (eval)

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

backward (backprop)

\[ dzdx = \text{vl\_nnconv}(x, W, b, dzdy) \]
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)
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);
Network abstraction: AutoNN

Defining and evaluating a deep network

% 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

- 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

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
Parallel Computing Toolbox (GPU)

Platform (Win, macOS, Linux)

NVIDIA CUDA (GPU)
NVIDIA CuDNN (Deep Learning Primitives; optional)
MatConvNet vs Neural Network Toolbox

An increasingly powerful alternative

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_nncnv, 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
- Multi-GPUs in parallel
- Visual features using activations

**Train**
- LSTM (time series, text)
- DAG Networks
- Create new layers

**Deploy / Share**
- Validation
- Training plots
- Hyper-parameter optimization

**New Product**
- **GPU Coder**: convert MATLAB models to NVIDIA CUDA code

---

- Tensorflow-Keras importer
- GoogLeNet model
- Label for semantic segmentation
- Resize & augment images
MatConvNet: Check it out

http://vlfeat.org/matconvnet/

https://github.com/vlfeat/matconvnet

Karel Lenc  
Sam Albanie  
Joao Henriques