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

Image of a bicycle with a diagram showing:
- C1 to C8
- f6, f7, f8
- W1 to W8
- Bike label
Confounding factors

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

Single shot (feed forward) detector
Pose recognition

Dense part and keypoint labelling
Real-time visual style transfer

Neural art
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
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?

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

How deep is deep enough?

How deep is deep enough?

16 convolutional layers

50 convolutional layers

152 convolutional layers


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

Backpropagation

$$\frac{d}{dw_1} \text{error}(w_1, \ldots, w_8)$$
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_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

forward (eval)

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

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

forward (eval)

\[ z(z) \in \mathbb{R} \]

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

### Forward (Eval)

```
x \rightarrow vl_nnconv(w, b) \rightarrow y
```

\[ y = vl_nnconv(x, W, b) \]

### Backward (Backprop)

```
\frac{dz}{dx} \rightarrow vl_nnconv(w, b) \rightarrow \frac{dz}{dy}
```

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

**forward (eval)**

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

**backward (backprop)**

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

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_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
- **Validation**
- **Training plots**
- Hyper-parameter optimization

### 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