Deep Learning III

Programs: Matlab, Swift, Python
Data
MacOS, Clouds: eGPU, GPU array
Software: Metal, OpenGL, OpenCL
Apps

Index of /matconvnet/models

Name	Last modified	Size	Description
Parent Directory		-	
beta10/	2015-04-13 07:52	-	
beta13/	2015-09-19 12:10	-	
beta16/	2015-11-21 11:30	-	
beta18/	2016-02-21 02:32	-	
beta22/	2016-05-02 02:44	-	
fast-rcnn-caffenet-p..>	2016-09-30 00:37	203M	
fast-rcnn-caffenet-p..>	2016-09-30 09:50	67K	
imagenet-matconvnet-..>	2016-09-30 00:33	366M	
imagenet-matconvnet-..>	2016-09-30 09:50	56K	
imagenet-matconvnet-..>	2016-09-30 00:34	366M	
imagenet-matconvnet-..>	2016-09-30 09:50	56K	
imagenet-matconvnet-..>	2016-09-30 00:34	492M	
imagenet-matconvnet-..>	2016-09-30 09:50	110K	
imagenet-resnet-101-..>	2016-09-30 00:33	159M	
imagenet-resnet-101-..>	2016-09-30 09:50	1.3M	
imagenet-resnet-152-..>	2016-09-30 00:33	215M	
imagenet-resnet-152-..>	2016-09-30 09:50	1.9M	
imagenet-resnet-50-d..>	2016-09-30 00:33	92M	
imagenet-resnet-50-d..>	2016-09-30 09:50	664K	
imagenet-vgg-f.mat	2016-09-30 00:35	217M	
imagenet-vgg-f.svg	2016-09-30 09:50	60K	
imagenet-vgg-m-1024.mat	2016-09-30 00:35	310M	
imagenet-vgg-m-1024.svg	2016-09-30 09:50	60K	

VLFeat.org

Google Custom Search

Home | Download | Tutorials | Applications | Documentation

Home

The **VLFeat** open source library implements popular computer vision algorithms specializing in image understanding and local features extraction and matching. Algorithms include Fisher Vector, VLAD, SIFT, MSER, k-means, hierarchical k-means, agglomerative information bottleneck, SLIC superpixels, quick shift superpixels, large scale SVM training, and many others. It is written in C for efficiency and compatibility, with interfaces in MATLAB for ease of use, and detailed documentation throughout. It supports Windows, Mac OS X, and Linux. The latest version of VLFeat is `0.9.21`.

ACM OpenSource Award

Download

- **VLFeat 0.9.21** (Windows, Mac, Linux)
- Source code and installation
- git repository, bug tracking.

Documentation

- MATLAB commands
- C API with algorithm descriptions
- Command line tools

Tutorials

- Features: Covariant detectors, HOG, SIFT, MSER, Quick shift, SLIC
- Clustering: IKM, HIKM, AIB
- Matching: Randomized kd-trees
- All tutorials

Example applications

- Caltech-101 classification
- SIFT matching for auto-stitching
- All example applications

Citing

```
@misc{vedaldi08vlfeat,
 Author = {A. Vedaldi and B. Fulkerson},
 Title = {{VLFeat}: An Open and Portable Library
          of Computer Vision Algorithms},
 Year  = {2008},
 Howpublished = {\url{http://www.vlfeat.org/}}
}
```

Acknowledgments

PASCAL2 Pattern Analysis, Statistical Modelling and Computational Learning Yandex UCLA Vision Lab Oxford VGG.

News

8/1/2018 **VLFeat 0.9.21 released**

Maintenance release. Fixed **`vl_argparse`** to be compatible with MatConvNet. Fixed the binaries for recent versions of macOS.

14/1/2015 **VLFeat 0.9.20 released**

Maintenance release. Bugfixes.

12/9/2014 ***MatConvNet***

IMAGENET

14,197,122 images, 21841 synsets indexed

Explore Download Challenges Publications Updates About

Logged in as jmwu. My Account | Logout

ImageNet is an image database organized according to the WordNet hierarchy (currently only the nouns), in which each node of the hierarchy is depicted by hundreds and thousands of images. Currently we have an average of over five hundred images per node. We hope ImageNet will become a useful resource for researchers, educators, students and all of you who share our passion for pictures.

Click here to learn more about ImageNet, Click here to join the ImageNet mailing list.

What do these images have in common? *Find out!*

Research updates on improving ImageNet data

kaggle

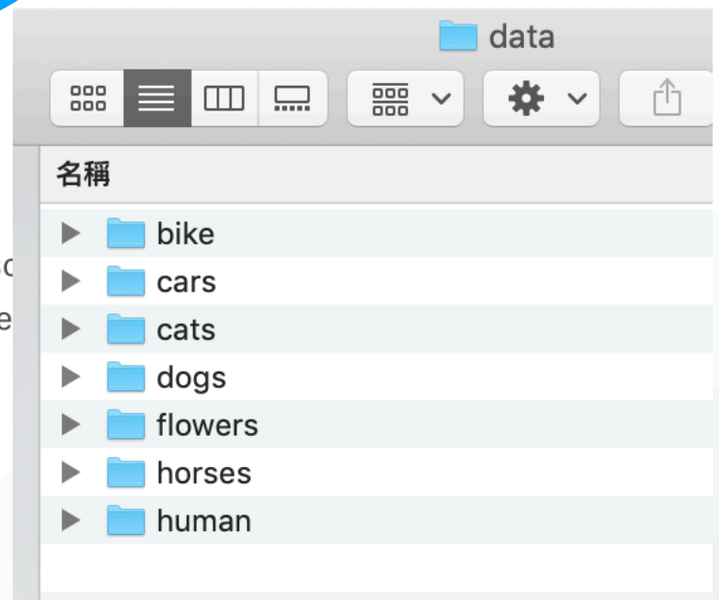
Simple Tutorial - Image Recognition -CNN

Python notebook using data from Images Dataset · 3,078 views · 4mo ago

deep learning, tutorial, cnn

40 | Copy and Edit 55 | ...

Seven categories for testing



Version 73

73 commits

Notebook

Tutorial - Image Recognition

Data

Comments

Tutorial - Image Recognition

This kernel covers the Image Recognition using CNN, Keras ,Tensc
implementation using image dataset . The following topics will be

- **Loading and preprocessing own dataset**
- **Designing and training a CNN model in Keras**
- **Plotting the Loss and Accuracy curve**
- **Evaluating the model & Predicting the output class of a test image**
- **Visualizing the intermediate layer output of CNN**
- **Plotting the confusion matrix for your result**

Loading and preprocessing own dataset

The dataset that I am using for this kernel is my own accumulated dataset of 7 types of classes namely 'flowers', 'cars', 'cats', 'horses', 'human', 'b
image samples.

Anna from Kaggle

eGPU

- External GPU
- MacOS 10.13.4
-

Data Parallel Problems

- Plenty of problems fall into this category (luckily ☺)
  - Graphics, image & video processing, physics, scientific computing, …
- This type of parallelism is called *data parallelism*
- And GPUs are the perfect solution for them!
  - In fact the more the data, the more efficient GPUs become at these algorithms
  - Bonus: You can relatively easily add more processing cores to a GPU and increase the throughput

Parallelism in CPUs v. GPUs

- CPUs use *task parallelism*
  - Multiple tasks map to multiple threads
  - Tasks run different instructions
  - 10s of relatively heavyweight threads run on 10s of cores
  - Each thread managed and scheduled explicitly
  - Each thread has to be individually programmed
- GPUs use *data parallelism*
  - SIMD model (Single Instruction Multiple Data)
  - Same instruction on different data
  - 10,000s of lightweight threads on 100s of cores
  - Threads are managed and scheduled by hardware
  - Programming done for batches of threads (e.g. one pixel shader per group of pixels, or draw call)

Stream Processing

- What we just described:
  - Given a (typically large) set of data ("stream")
  - Run the same series of operations ("kernel" or "shader") on all of the data (SIMD)

- GPUs use various optimizations to improve throughput:
  - Some on-chip memory and local caches to reduce bandwidth to external memory
  - Batch groups of threads to minimize incoherent memory access
    - Bad access patterns will lead to higher latency and/or thread stalls.
  - Eliminate unnecessary operations by exiting or killing threads
    - Example: Z-Culling and Early-Z to kill pixels which will not be displayed

Blackmagic eGPU for MacBook Pro available exclusively from Apple

將外接式繪圖處理器搭配 Mac 使用

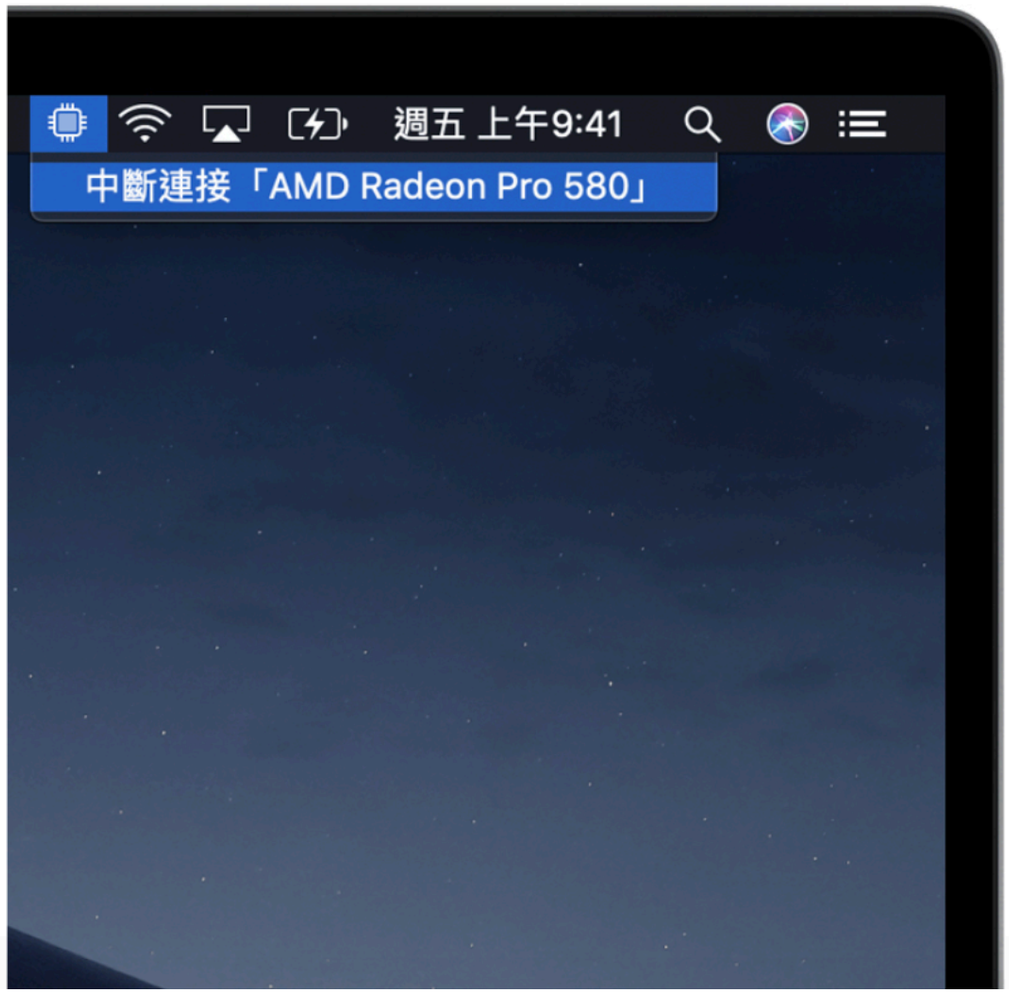
配備 Thunderbolt 3 並執行 macOS High Sierra 10.13.4 或以上版本的 Mac 可以連接至外接式繪圖處理器 (亦稱為 eGPU)，取得額外的繪圖效能。

eGPU 可提供 Mac 額外的繪圖效能，以執行專業 app、3D 遊戲、VR 內容創作等各種用途。

任何執行 macOS High Sierra 10.13.4 或以上版本並配備 Thunderbolt 3 的 Mac[1]，皆支援 eGPU。瞭解如何更新 Mac 上的軟體。

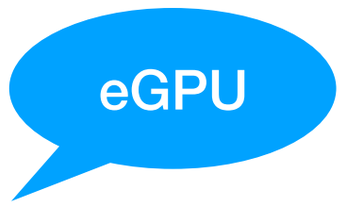
您可以透過 eGPU 在 Mac 上執行以下操作：

- 加速用到 Metal、OpenGL 和 OpenCL 的 app
- 連接更多外接監視器和顯示器
- 使用插入 eGPU 的虛擬實境頭戴顯示器
- 在使用 eGPU 的同時為 MacBook Pro 充電
- 在闔上 MacBook Pro 的內建顯示器時，使用 eGPU 搭配 MacBook Pro
- 在使用者登入時連接 eGPU
- 使用 Mac 上的多個 Thunderbolt 3 (USB-C) 連接埠連接超過一個以上的 eGPU[2]
- 使用選單列項目 安全地中斷連接 eGPU



[教學][TB3][eGPU] macOS 全版本 eGPU 外接式顯卡 NVIDIA / AMD 顯卡安裝教學

POSTED ON 2019-01-13



#Thunderbolt3 擴充計畫

2018 年 3 月， Apple 釋出 macOS 10.13.4 更新，正式支援 eGPU ，正式進入 Thunderbolt 3 大 eGPU 時代。只要你多課金買了 Thunderbolt 3 的 eGPU Box，加上不錯的顯卡，你就可以大大改善 macOS 裝置顯卡都很低規的問題
來源：https://support.apple.com/zh-tw/HT208544

> *macOS High Sierra 10.13.4 和以上版本中的 eGPU 支援，是要藉由強大的 eGPU 效能來提升 Metal、OpenGL 和 OpenCL app 的速度。並非所有 app 都支援 eGPU 加速*
> *– 新增外接繪圖處理器（eGPU）支援。*

任何執行 macOS High Sierra 10.13.4 或以上版本並配備 Thunderbolt 3 的

Metal by Example

High-performance graphics and data-parallel programming for iOS and macOS

Swift

Using Basis Universal Texture Compression with Metal

July 11, 2019 by Warren Moore

In this short article, we'll take a look at a relatively new compressed texture format called Basis. Basis is developed by Binomial, LLC, a company founded by Rich Geldreich (of crunch fame) and Stephanie Hurlburt.

Basis is unique among compression formats in that it emphasizes efficient transcoding between compressed formats. This means that a single .basis file can be transformed, at runtime, into a format that's optimal for the target platform, without decompressing it in memory, saving space and bandwidth.

Sample Code

Training a Neural Network with Metal Performance Shaders

Use an MPS neural network graph to train a simple neural network digit classifier.

Download

SDKs

macOS 10.15+

Xcode 11.0+

Framework

Metal Performance Shaders

Overview

The sample code describes how to write a neural network using `MPSNNGraph` and how to train the network to recognize a digit in an image. The sample trains a network for 300 iterations on a batch size of 40 images. You'll see how to set up training of weights and biases using data sources, including how to initialize and update weights. You'll also see how to validate the network using a test dataset.

Note

This sample code project is associated with WWDC 2019 session 614: Metal for Machine Learning.

On This Page

Forge: neural network toolkit for Metal

24 APRIL 2017 10 minutes

Today I am happy to announce Forge, an open source library that makes it a bit easier to build neural networks with MPSCNN.

Convolutional neural networks on the iPhone with VGGNet

30 AUGUST 2016 34 minutes

I implemented the VGGNet architecture for image recognition on the iPhone, using the new convolutional neural network API from the Metal Performance Shaders framework.

In this post I explain how CNNs work and specifically how to get VGGNet running on your iPhone using Metal.

The demo app sends the video feed from the iPhone's camera through the neural network to get the top-5 classification scores for whatever you're looking at:

Quick Start

cd matconvnet-1.0-beta25
run matlab/vl_compilenn ;

Step 1

```
urlwrite(...
  'http://www.vlfeat.org/matconvnet/models/imagenet-vgg-f.mat', ...
  'imagenet-vgg-f.mat') ;

% Setup MatConvNet.
run matlab/vl_setupnn ;

% Load a model and upgrade it to MatConvNet current version.
net = load('imagenet-vgg-f.mat') ;
net = vl_simplenn_tidy(net) ;

% Obtain and preprocess an image.
im = imread('peppers.png') ;
im_ = single(im) ; % note: 255 range
im_ = imresize(im_, net.meta.normalization.imageSize(1:2)) ;
im_ = im_ - net.meta.normalization.averageImage ;

% Run the CNN.
res = vl_simplenn(net, im_) ;

% Show the classification result.
scores = squeeze(gather(res(end).x)) ;
[bestScore, best] = max(scores) ;
figure(1) ; clf ; imagesc(im) ;
title(sprintf('%s (%d), score %.3f',...
  net.meta.classes.description{best}, best, bestScore)) ;
```

Step 2

Save as
demo_imagenet.m

```
% Setup MatConvNet.
run matlab/vl_setupnn ;

% Load a model and upgrade it to MatConvNet current version.
net = load('imagenet-vgg-f.mat') ;
net = vl_simplenn_tidy(net) ;

% Obtain and preprocess an image.
im = imread('peppers.png') ;
im_ = single(im) ; % note: 255 range
im_ = imresize(im_, net.meta.normalization.imageSize(1:2)) ;
im_ = im_ - net.meta.normalization.averageImage ;

% Run the CNN.
res = vl_simplenn(net, im_) ;

% Show the classification result.
scores = squeeze(gather(res(end).x)) ;
[bestScore, best] = max(scores) ;
figure(1) ; clf ; imagesc(im) ;
title(sprintf('%s (%d), score %.3f',...
  net.meta.classes.description{best}, best, bestScore)) ;
```

demo_imagenet.m

/ ▸ Users ▸ apple ▸ Desktop ▸ Jiann-Ming Wu ▸ code2019 ▸ MatConvNet ▸ matconvnet-1.0-beta25 ▸

Current Folder

Name

- data
- doc
- examples
- matconvnet.xcodeproj
- matlab
- utils
- apples.png
- cats.png
- CONTRIBUTING.md
- COPYING
- demo_imagenet.m
- imagenet-vgg-f.mat
- Makefile
- matconvnet.sln
- matconvnet.vcxproj
- matconvnet.vcxproj.filters
- README.md

Details

Select a file to view details

Imagenet-vgg-f.mat

Editor - /Users/apple/Desktop/Jiann-Ming Wu/code2019/MatConvNet/matconvnet-1.0-beta25/demo_imagenet.m

+17 | cnn_imagenet_camdemo.m | ToolboxDemo.m | demo_imagenet.m | vl_simplenn.m | +

```
12    % Run the CNN.
13    res = vl_simplenn(net, im_) ;
14
15    % Show the classification result.
16    scores = squeeze(gather(res(end).x)) ;
17    [bestScore, best] = max(scores) ;
18    figure(1) ; clf ; imagesc(im) ;
19    title(sprintf('%s (%d), score %.3f',...
20       net.meta.classes.description{best}, best, bestScore)
```

Command Window

```
ans =

  1×1000 cell array

  Columns 1 through 2

    {'tench, Tinca ti…'}    {'goldfish, Caras…'}

  Columns 3 through 4
```

MATLAB R2019a - academic use

HOME | PLOTS | APPS

Search Documentation

New Script | New Live Script | New | Open | Find Files | Compare | Import Data | Save Workspace | New Variable | Open Variable | Clear Workspace | Favorites | Analyze Code | Run and Time | Clear Commands | Simulink | Layout | Preferences | Set Path | Parallel | Add-Ons | Help | Community | Request Suppor | Learn MATLAB

FILE | VARIABLE | CODE | SIMULINK | ENVIRONMENT | RESOURCES

/ ▸ Users ▸ apple ▸ Desktop ▸ Jiann-Ming Wu ▸ code2019 ▸ MatConvNet ▸ matconvnet-1.0-beta25 ▸

Current Folder

Name ▲

- doc
- examples
- matconvnet.xcodeproj
- matlab
- utils
- CONTRIBUTING.md
- COPYING
- imagenet-vgg-f.mat
- Makefile
- matconvnet.sln
- matconvnet.vcxproj
- matconvnet.vcxproj.filters
- README.md

Details

Select a file to view details

Editor - /Users/apple

+5 script_render_fitt

```
1  func
2  %CNN
3  %
```

Command Window

```
im_ = imre
im_ = im_

% Run the
 = vl_s

 how the
scores = s
[bestScore
figure(1)
title(spri
   net.meta.classes.description{best}, best, bestScore)) ;
```

>>

EX-I

Figure 1

File Edit View Insert Tools Desktop Window Help

getBatch
for trai
menting s

ize(1:2)) ;

Adaline and FPGA & Filter

MATLAB R2019a - academic use

HOME | PLOTS | APPS | EDITOR | PUBLISH | VIEW

Search Documentation

New | Open | Save | Find Files | Compare | Print — FILE
Go To | Find — NAVIGATE
Insert | Comment | Indent — EDIT
Breakpoints — BREAKPOINTS
Run | Run and Advance | Run Section | Advance | Run and Time — RUN

/ ▸ Users ▸ apple ▸ Desktop ▸ Jiann-Ming Wu ▸ code2019 ▸ MatConvNet ▸ matconvnet-1.0-beta25 ▸

Current Fol...

Name

- data
- doc
- examples
- matconvnet....
- matlab
- utils
- apples.png
- cats.png
- CONTRIBUTI...
- COPYING
- demo_image...
- imagenet-v...
- Makefile
- matconvnet....
- matconvnet....
- matconvnet....
- README.md

Details

Select a file to view

Editor - /Users/apple/Desktop/Jiann-Ming Wu/code2019/MatConvNet/matconvnet-1.0-beta25/demo_imagenet.m

+16 | interpolating_3Dface_elstic_nen.m | cnn_imagenet_camdemo.m | ToolboxDemo.m | demo_imagenet.m | vl_simplenn.m

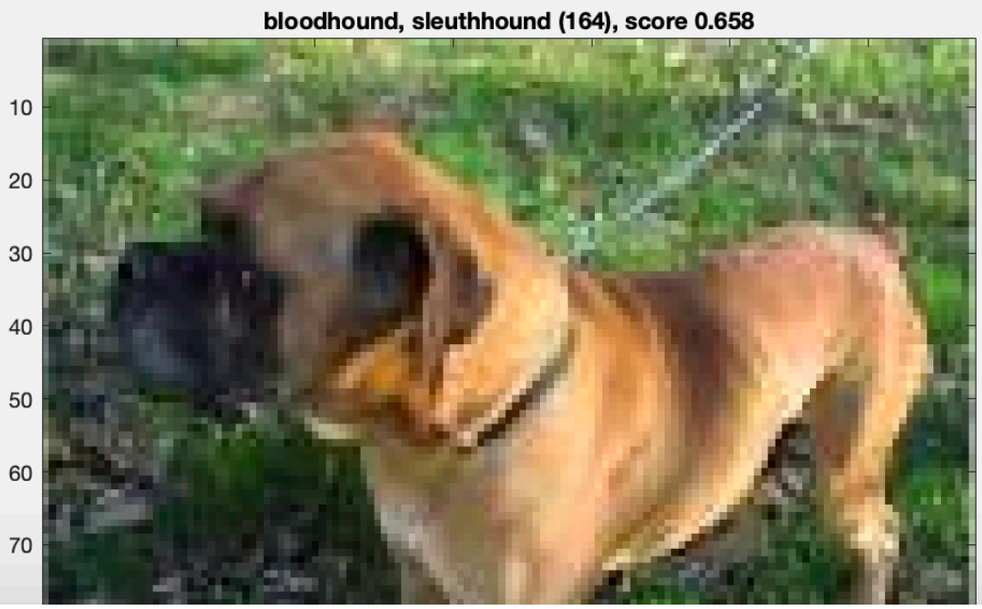
```
2    % Load a model and upgrade it to MatConvNet current version.
3    net = load('imagenet-vgg-f.mat') ;
4    net = vl_simplenn_tidy(net) ;
5
6    % Obtain and preprocess an image.
7    % im = imread('peppers.png') ;
8    im = imread('../../image_recognition_CNN/data/dogs/dog.11.jpg');
9    im_ = single(im) ; % note: 255 range
10   im_ = imresize(im_,
11   im_ = im_ - net.met
```

Command Window

```
>> demo_imagenet
>>
```

Figure 1

File Edit View Insert Tools Desktop Window Help



HOME | PLOTS | APPS | EDITOR | PUBLISH | VIEW

Search Documentation

New | Open | Save | Find Files | Compare | Print — FILE
Go To | Find — NAVIGATE
Insert | Comment | Indent — EDIT
Breakpoints — BREAKPOINTS
Run | Run and Advance | Run Section | Advance | Run and Time — RUN

/ ▸ Users ▸ apple ▸ Desktop ▸ Jiann-Ming Wu ▸ code2019 ▸ MatConvNet ▸ matconvnet-1.0-beta25 ▸

Current Fol...

Name

- data
- doc
- examples
- matconvnet....
- matlab
- utils
- apples.png
- cats.png
- CONTRIBUTI...
- COPYING
- demo_image...
- demo_image...
- imagenet-v...
- Makefile
- matconvnet....
- matconvnet....
- matconvnet....
- README.md

Details

Select a file to view

Editor - /Users/apple/Desktop/Jiann-Ming Wu/code2019/MatConvNet/matconvnet-1.0-beta25/demo_imagenet.m

+16 | interpolating_3Dface_elstic_nen.m | cnn_imagenet_camdemo.m | ToolboxDemo.m | demo_imagenet.m | vl_simplenn.m | +

```
1 - run matlab/vl_setupnn ;
2   % Load a model and upgrade it to MatConvNet current version.
3 - net = load('imagenet-vgg-f.mat') ;
4 - net = vl_simplenn_tidy(net) ;
5
6 - % C
7 - % j
8 - im                                    ata/cars/mycar.png');
9 - im_
10 - im_                                  imageSize(1:2)) ;
```

Figure 1

File Edit View Insert Tools Desktop Window Help

Command Window

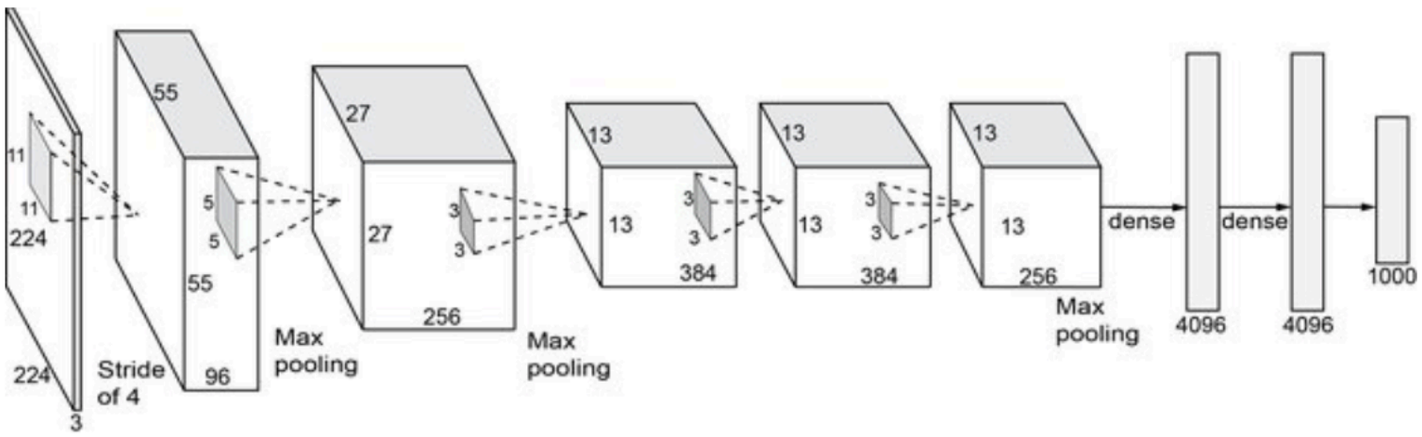
```
>> demo
fx >>
```

VERY DEEP CONVOLUTIONAL NETWORKS FOR LARGE-SCALE IMAGE RECOGNITION

Karen Simonyan* **& Andrew Zisserman+**
Visual Geometry Group, Department of Engineering Science, University of Oxford
{karen,az}@robots.ox.ac.uk

ABSTRACT

In this work we investigate the effect of the convolutional network depth on its accuracy in the large-scale image recognition setting. Our main contribution is a thorough evaluation of networks of increasing depth using an architecture with very small (3×3) convolution filters, which shows that a significant improvement on the prior-art configurations can be achieved by pushing the depth to 16–19 weight layers. These findings were the basis of our ImageNet Challenge 2014 submission, where our team secured the first and the second places in the localisation and classification tracks respectively. We also show that our representations generalise well to other datasets, where they achieve state-of-the-art results. We have made our two best-performing ConvNet models publicly available to facilitate further research on the use of deep visual representations in computer vision.



Figure

Caption

Fig. 1: Architecture of VGG-F.

```
>> net.layers{1}

ans =

  struct with fields:

        name: 'conv1'
        type: 'conv'
     weights: {[11×11×3×64 single]  [64×1 single]}
        size: [11 11 3 64]
         pad: [0 0 0 0]
      stride: [4 4]
    precious: 0
      dilate: 1
        opts: {}
```

MATLAB R2019a - academic use

HOME | PLOTS | APPS | EDITOR | PUBLISH | VIEW

Search Documentation | Sign In

New | Open | Save | Find Files | Compare | Print — FILE

Go To | Find — NAVIGATE

Insert | Comment | Indent — EDIT

Breakpoints — BREAKPOINTS

Continue | Step | Step In | Step Out | Run to Cursor | Function Call Stack: demo_imagenet | Quit Debugging — DEBUG

/ ▸ Users ▸ apple ▸ Desktop ▸ Jiann-Ming Wu ▸ code2019 ▸ MatConvNet ▸ matconvnet-1.0-beta25 ▸

Current Folder

Name

- data
- doc
- examples
- matconvnet.xcodeproj
- matlab
- utils
- apples.png
- cats.png
- CONTRIBUTING.md
- COPYING
- demo_imagenet.m
- imagenet-vgg-f.mat
- Makefile
- matconvnet.sln
- matconvnet.vcxproj
- matconvnet.vcxproj.filters
- README.md

Details

Select a file to view details

Editor - /Users/apple/Desktop/Jiann-Ming Wu/code2019/MatConvNet/matconvnet-1.0-...

+18 | ToolboxDemo.m | demo_imagenet.m | vl_simplenn.m | cnn_cifar.m | +

```
 9 -    im_ = single(im) ; % note: 255 range
10 -    im_ = imresize(im_, net.meta.normalization
11 -    im_ = im_ - net.meta.normalization.average
12
13      % Run the CNN.
14      res = vl_simplenn(net, im_) ;
15
16      % Show the classification result.
17 -    scores = squeeze(gather(res(end).x)) ;
18 -    [bestScore, best] = max(scores) ;
19 -    figure(1) ; clf ; imagesc(im) ;
20 -    title(sprintf('%s (%d), score %.3f',...
21        net.meta.classes.description{best}, bes
```

Command Window

```
>> demo_imagenet
14   res = vl_simplenn(net, im_) ;
K>>
```

Workspace - dem...

Name	Value
im	306x4
im_	224x2
net	1x1 st

2 usages of "res" found | script | Ln 14 Col 1

MATLAB R2019a - academic use

HOME | PLOTS | APPS | EDITOR | PUBLISH | VIEW

Search Documentation | Sign In

New | Open | Save | Find Files | Compare | Print — FILE
Go To | Find — NAVIGATE
Insert | Comment | Indent — EDIT
Breakpoints — BREAKPOINTS
Continue | Step | Step In | Step Out | Run to Cursor | Function Call Stack: vl_simplenn | Quit Debugging — DEBUG

/ ▸ Users ▸ apple ▸ Desktop ▸ Jiann-Ming Wu ▸ code2019 ▸ MatConvNet ▸ matconvnet-1.0-beta25 ▸

Current Folder

Name
- data
- doc
- examples
- matconvnet.xcodeproj
- matlab
- utils
- apples.png
- cats.png
- CONTRIBUTING.md
- COPYING
- demo_imagenet.m
- imagenet-vgg-f.mat
- Makefile
- matconvnet.sln
- matconvnet.vcxproj
- matconvnet.vcxproj.filters
- README.md

Details

Select a file to view details

Editor - /Users/apple/Desktop/Jiann-Ming Wu/code2019/MatConvNet/matconvnet-1.0-...

+18 | ToolboxDemo.m | demo_imagenet.m | vl_simplenn.m | cnn_cifar.m

```
289
290   %-------------------------------------------------------------------------
291   %-------------------------------------------------------------------------
292
293
294 - for i=1:n
295 -   if opts.skipForward, break; end;
296 -   l = net.layers{i};
297 -   res(i).time = tic;
298 -   switch l.type
299 -     case 'conv'
300 -       res(i+1).x = vl_nnconv(res(i).x, l.
301           'pad', l.pad, ...
```

break

opts.skipForward: logical =

0

Command Window

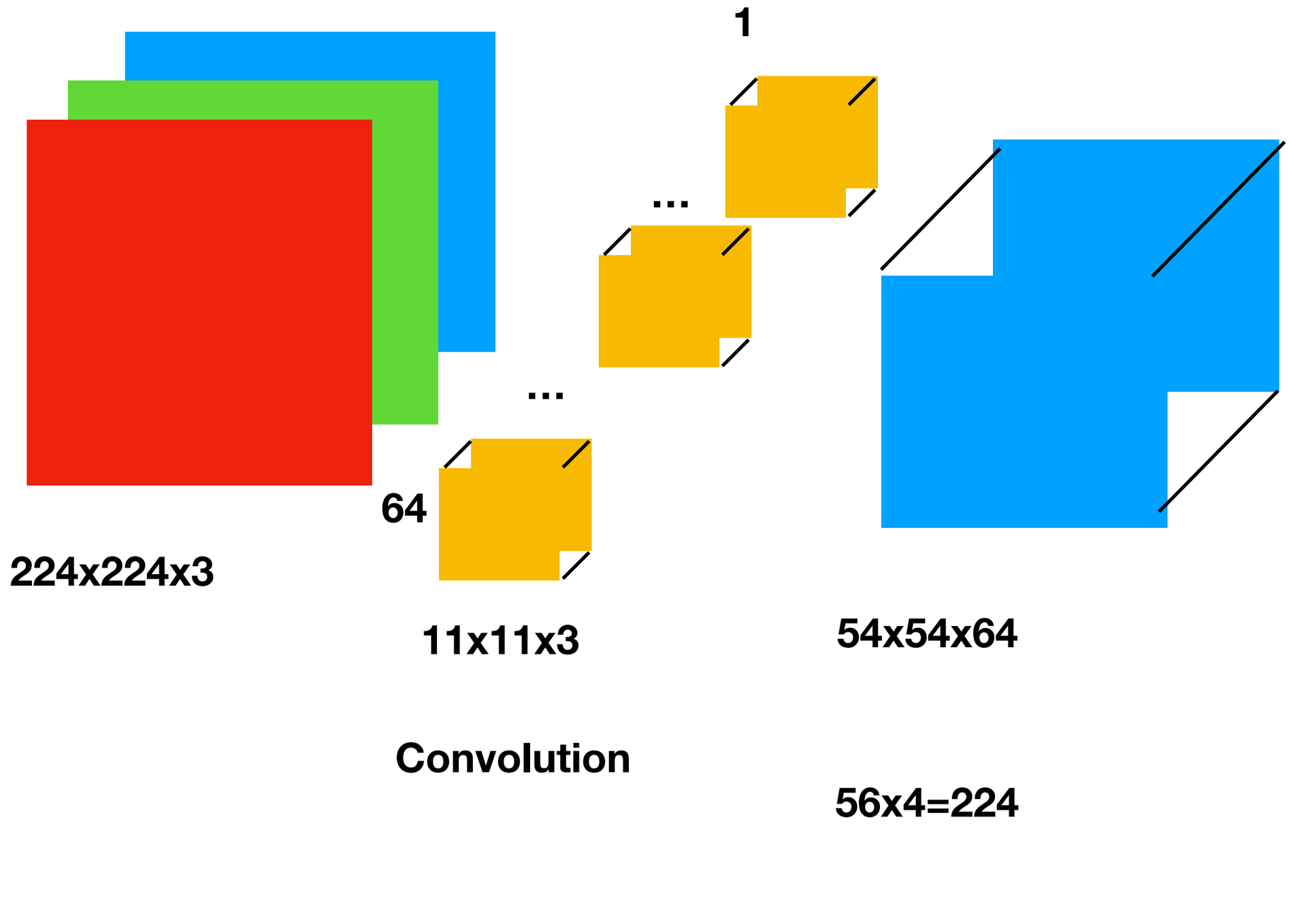
```
>> demo_imagenet
14   res = vl_simplenn(net, im_) ;
K>>
```

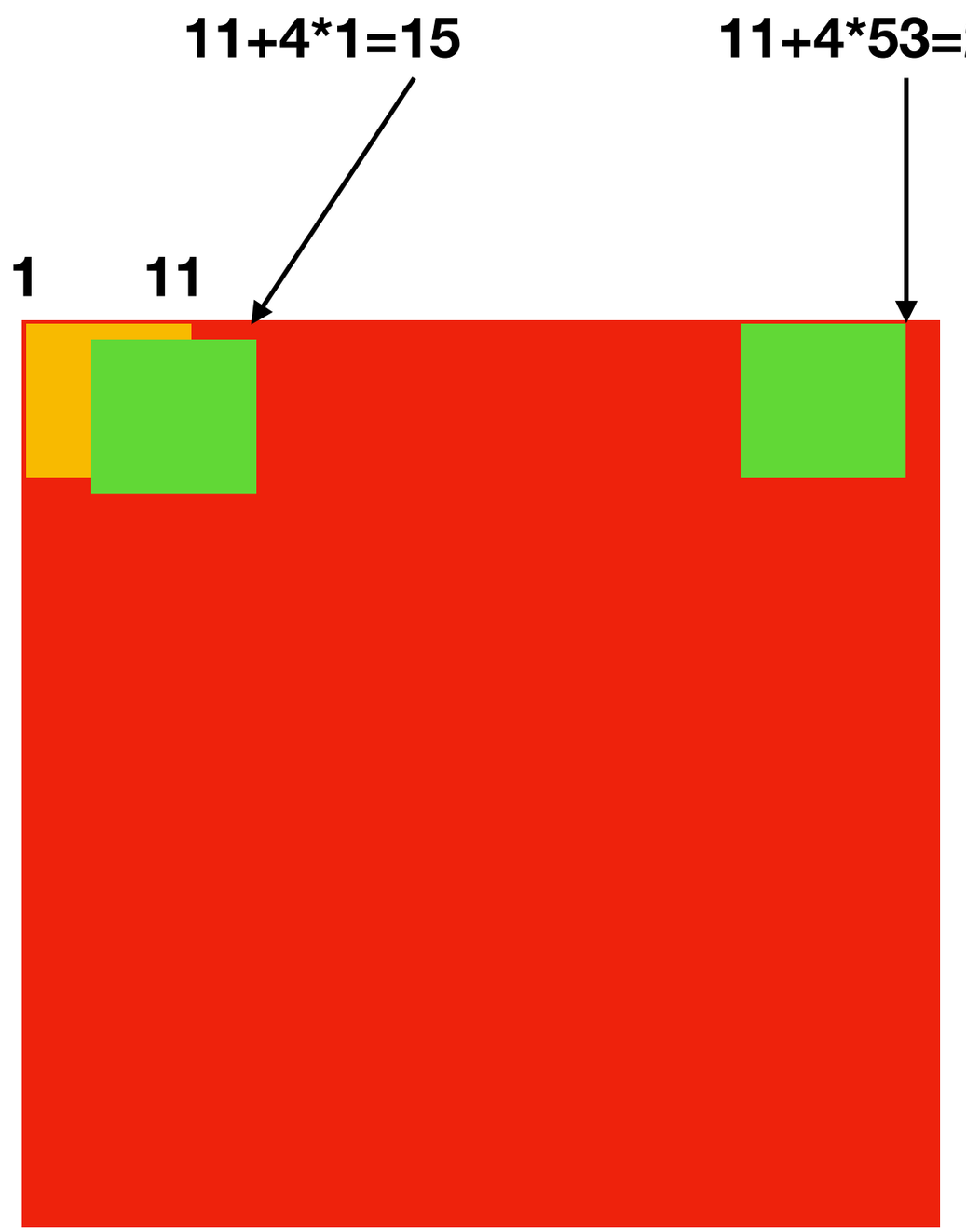
Workspace - vl_si...

Name	Value
backPropLim	1
bnormCudnn	1x1 c
cudnn	1x1 c
doder	0
gpuMode	0
n	21
net	1x1 s
opts	1x1 s
res	1x22
testMode	0
varargin	0x0 c
x	224x2

```
>> net.layers{1}

ans =

  struct with fields:

        name: 'conv1'
        type: 'conv'
     weights: {[11×11×3×64 single]  [64×1 single]}
        size: [11 11 3 64]
         pad: [0 0 0 0]
      stride: [4 4]
    precious: 0
      dilate: 1
        opts: {}
```

stride





Convolution with 4x4 stride

MATLAB R2019a - academic use

HOME | PLOTS | APPS | EDITOR | PUBLISH | VIEW

Search Documentation | Sign In

New | Open | Save | Find Files | Compare | Print | Go To | Find | Insert | Comment | Indent | Breakpoints | Continue | Step | Step In | Step Out | Run to Cursor | Function Call Stack: vl_simplenn | Quit Debugging

FILE | NAVIGATE | EDIT | BREAKPOINTS | DEBUG

/ ▸ Users ▸ apple ▸ Desktop ▸ Jiann-Ming Wu ▸ code2019 ▸ MatConvNet ▸ matconvnet-1.0-beta25 ▸

Current Folder

Name

- data
- doc
- examples
- matconvnet.xcodeproj
- matlab
- utils
- apples.png
- cats.png
- CONTRIBUTING.md
- COPYING
- demo_imagenet.m
- imagenet-vgg-f.mat
- Makefile
- matconvnet.sln
- matconvnet.vcxproj
- matconvnet.vcxproj.filters
- README.md

Details

Select a file to view details

Editor - /Users/apple/Desktop/Jiann-Ming Wu/code2019/MatConvNet/matconvnet-1.0-...

+18 | ToolboxDemo.m | demo_imagenet.m | vl_simplenn.m | cnn_cifar.m

```
378 -     end
379
380
381 -     % optionally forget intermediate result
382 -     needsBProp = doder && i >= backPropLim;
      -     forget = opts.conserveMemory && ~needsB
      -     if i > 1
```

Command Window

```
14   res = vl_simplenn(net, im ) ;
K>> size(res(i+1).x )

ans =

    54    54    64

K>> size(res(i).x )

ans =

   224   224     3
```

Output size of 1st hidden layer

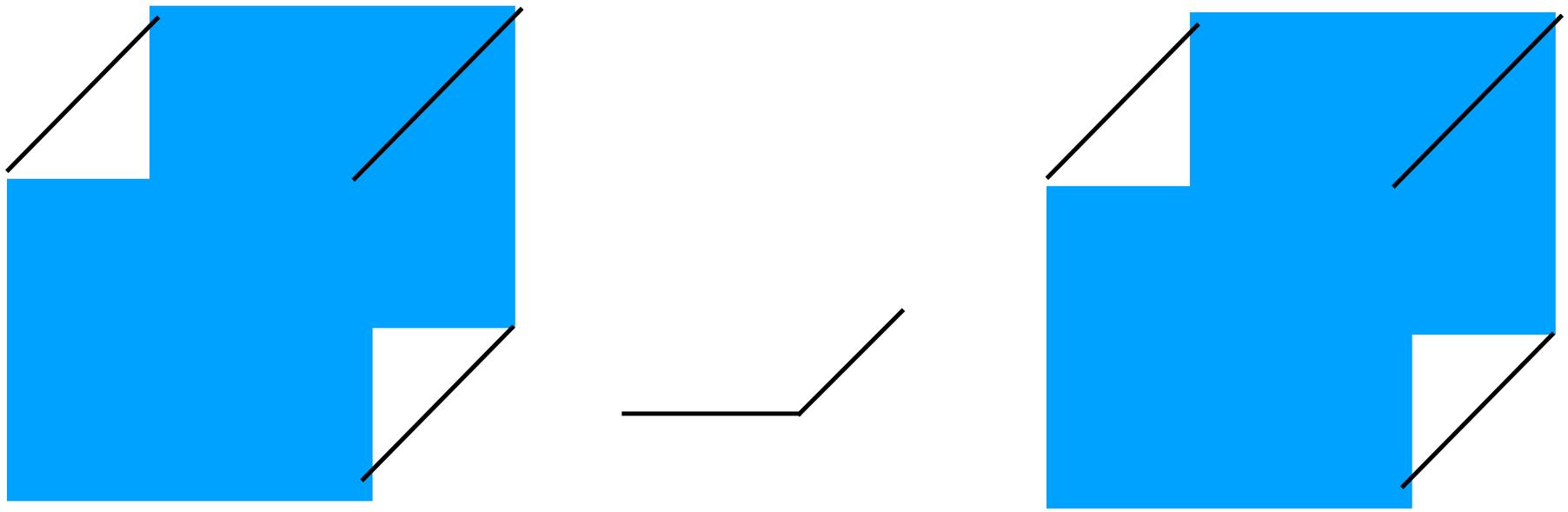
Input size of 1st hidden layer

Workspace - vl_si...

Name	Value
ans	[224,2
backPropLim	1
bnormCudnn	1x1 ce
cudnn	1x1 ce
doder	0
gpuMode	0
i	1
l	1x1 st
n	21
net	1x1 st
opts	1x1 st
res	1x22 s
testMode	0
varargin	0x0 ce
x	224x2

Paused in debugger

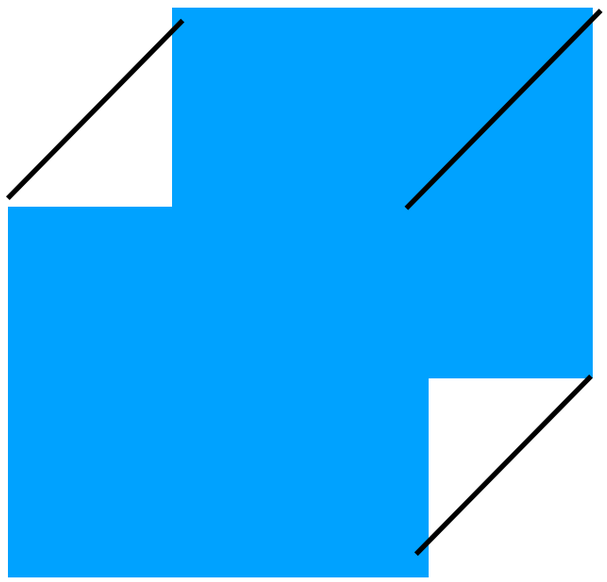
ReLu



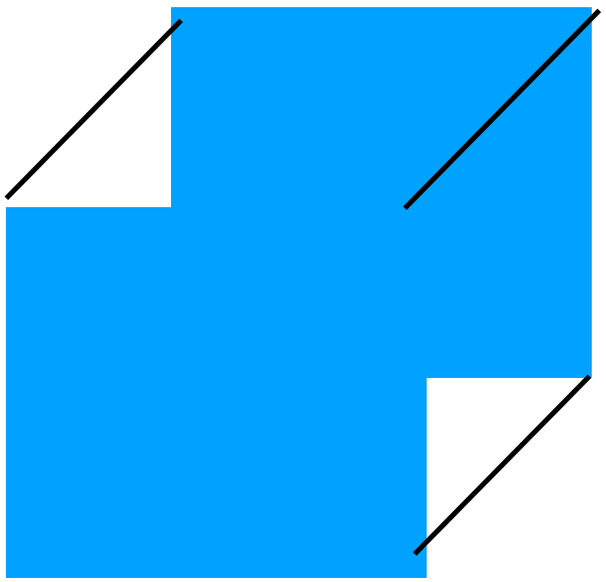
54x54x64

54x54x64

Norm



Normalization



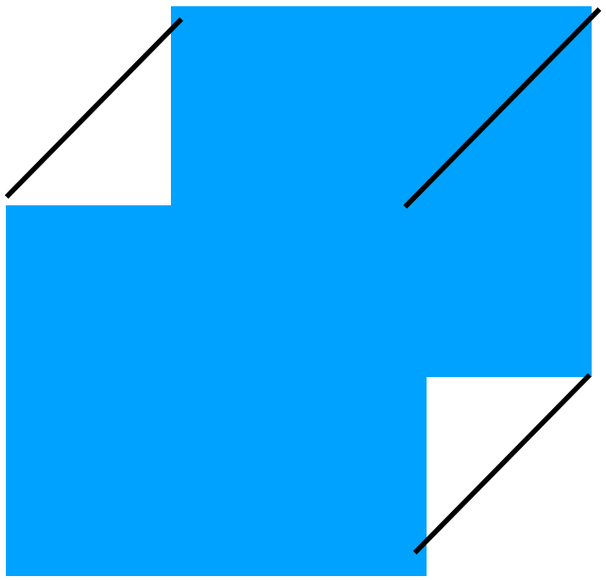
54x54x64

54x54x64

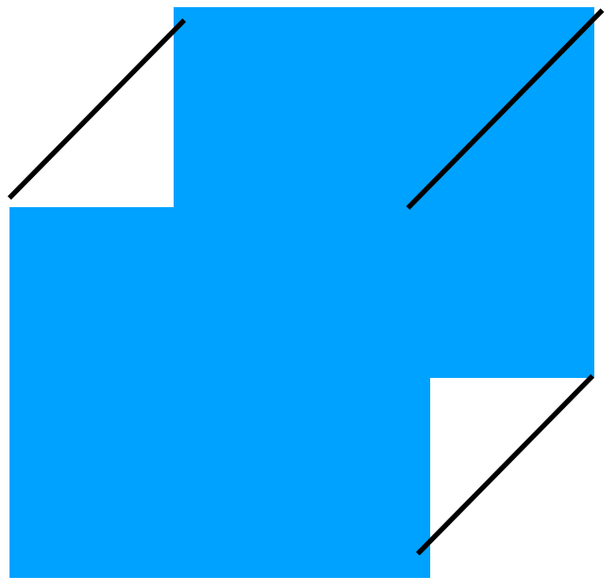
struct with fields:

name: 'pool1'
type: 'pool'
method: 'max'
pool: [3 3]
stride: [2 2]
pad: [0 1 0 1]
weights: {}
precious: 0
opts: {}

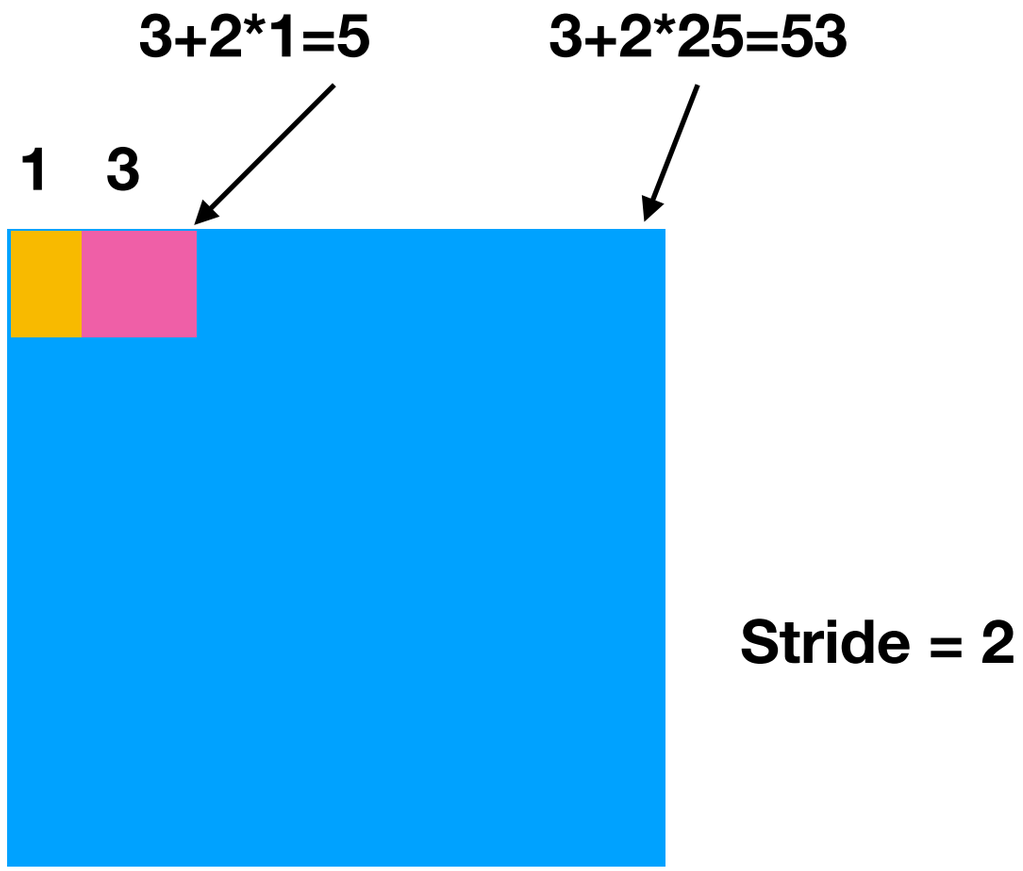
Max pooling



54x54x64



27x27x64


3+2*1=5
3+2*25=53
1
3
Stride = 2
54x54

MATLAB R2019a - academic use

HOME | PLOTS | APPS | EDITOR | PUBLISH | VIEW

New | Open | Save | Find Files | Compare | Print | Go To | Find | Insert | Comment | Indent | Breakpoints | Continue | Step | Step In | Step Out | Run to Cursor | Function Call Stack: vl_simplenn | Quit Debugging

FILE | NAVIGATE | EDIT | BREAKPOINTS | DEBUG

/ ▸ Users ▸ apple ▸ Desktop ▸ Jiann-Ming Wu ▸ code2019 ▸ MatConvNet ▸ matconvnet-1.0-beta25 ▸

Current Folder

Name

- data
- doc
- examples
- matconvnet.xcodeproj
- matlab
- utils
- apples.png
- cats.png
- CONTRIBUTING.md
- COPYING
- demo_imagenet.m
- imagenet-vgg-f.mat
- Makefile
- matconvnet.sln
- matconvnet.vcxproj
- matconvnet.vcxproj.filt...
- README.md
- test1.png

Details

Select a file to view details

Editor - /Users/apple/Desktop/Jiann-Ming Wu/code2019/MatConvNet/matconvnet-1.0-beta25/matlab/simplenn/vl_simplenn.m

+17 | cnn_imagenet_camdemo.m | ToolboxDemo.m | demo_imagenet.m | vl_simplenn.m | cnn_cifar.m

```
379
380       % optionally forget intermediate results
381       needsBProp = doder && i >= backPropLim;
382       forget = opts.conserveMemory && ~needsBProp ;
383       if i > 1
```

Command Window

```
K>> size(res(i+1).x)

ans =

    27    27    64

K>> size(res(i).x)

ans =

    54    54    64

K>>
```

3 usages of "needsBProp" found

MatConvNet

< Back to Alex Krizhevsky's home page

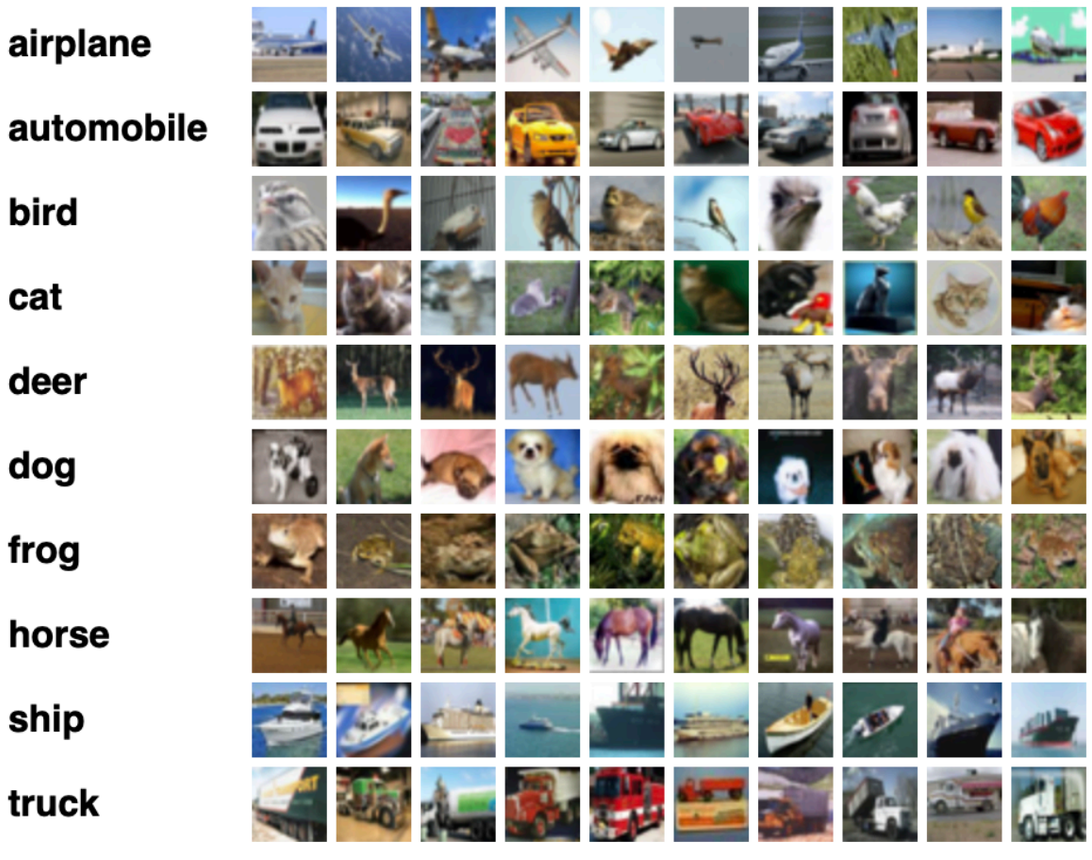
The CIFAR-10 and CIFAR-100 are labeled subsets of the 80 million tiny images dataset. They were collected by Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton.

The CIFAR-10 dataset

The CIFAR-10 dataset consists of 60000 32x32 colour images in 10 classes, with 6000 images per class. There are 50000 training images and 10000 test images.

The dataset is divided into five training batches and one test batch, each with 10000 images. The test batch contains exactly 1000 randomly-selected images from each class. The training batches contain the remaining images in random order, but some training batches may contain more images from one class than another. Between them, the training batches contain exactly 5000 images from each class.

Here are the classes in the dataset, as well as 10 random images from each:



The classes are completely mutually exclusive. There is no overlap between automobiles and trucks. "Automobile" includes sedans, SUVs, things of that sort. "Truck" includes only big trucks. Neither includes pickup trucks.

MATLAB R2019a - academic use

HOME | PLOTS | APPS | EDITOR | PUBLISH | VIEW

Search Documentation

New | Open | Save | Find Files | Compare | Print — FILE
Go To | Find — NAVIGATE
Insert | Comment | Indent — EDIT
Breakpoints — BREAKPOINTS
Pause | Run and Advance | Run Section | Advance | Run and Time — RUN

/ ▸ Users ▸ apple ▸ Desktop ▸ Jiann-Ming Wu ▸ code2019 ▸ MatConvNet ▸ matconvnet-1.0-beta25 ▸ examples ▸ cifar

Current Folder

Name

- cnn_cifar.m
- cnn_cifar_init.m
- cnn_cifar_init_nin.m

cnn_cifar.m (Function)

Demonstrates MatConvNet on CIFAR-10

- cnn_cifar(varargin)
- getBatch(opts)
- getSimpleNNBatch(imdb, batch)
- getDagNNBatch(opts, imdb, bat...
- getCifarImdb(opts)

Editor - /Users/apple/Desktop/Jiann-Ming Wu/code2019/MatConvNet/matconvnet-1.0-beta25_2019_10/exampl...

+4 | script_load_scan_lattice_structured.m | script_render_fittings.m | script_load_scan.m | cnn_cifar.m

```
1  function [net, info] = cnn_cifar(varargin)
2  % CNN_CIFAR   Demonstrates MatConvNet on CIFAR-10
3  %    The demo includes two standard model: LeNet and
4  %    Network (NIN). Use the 'modelType' option to cho
5
6  run(fullfile(fileparts(mfilename('fullpath')), ...
7    '..', '..', 'matlab', 'vl_setupnn.m')) ;
8
9  opts.modelType = 'lenet' ;
10 [opts, varargin] = vl_argparse(opts, varargin) ;
```

Command Window

```
117/500: 617.5 (643.0) Hz objective: 0.808 top1err: 0.279 t
118/500: 617.7 (643.0) Hz objective: 0.809 top1err: 0.278 t
119/500: 617.8 (630.6) Hz objective: 0.809 top1err: 0.278 t
120/500: 617.8 (611.6) Hz objective: 0.807 top1err: 0.278 t
121/500: 617.7 (610.3) Hz objective: 0.807 top1err: 0.278 t
122/500: 617.9 (635.9) Hz objective: 0.808 top1err: 0.278 t
123/500:
```

MATLAB R2019a - academic use

HOME | PLOTS | APPS | EDITOR | PUBLISH | VIEW

Search Documentation | Sign In

New | Open | Save | Find Files | Compare | Print | Go To | Find | Insert | Comment | Indent | Breakpoints | Pause | Run and Advance | Run Section | Advance | Run and Time

FILE | NAVIGATE | EDIT | BREAKPOINTS | RUN

/ ▸ Users ▸ apple ▸ Desktop ▸ Jiann-Ming Wu ▸ code2019 ▸ MatConvNet ▸ matconvnet-1.0-beta25 ▸ examples ▸ cifar

Current ...

Name

cnn_cifar.m
cnn_cifar...
cnn_cifar...

Editor - /Users/apple/Desktop/Jiann-Ming Wu/code2019/MatConvNet/matconvnet-1.0-beta25_2019_10/examples/cifar/cnn_cifa...

+4 | script_load_scan_lattice_structured.m | script_render_fittings.m | script_load_scan.m | cnn_cifar.m | +

```
22 -  opts = vl_argparse(opts, varargin) ;
23 -  if ~isfield(opts.train, 'gpus'), opts.train.gpus = []; end;
24
25 -  % --------------------------------------------------------------------
26 -  %                                                          Prepare r
27 -  % --------------------------------------------------------------------
28
29 -  switch opts.modelType
30 -    case 'lenet'
31 -      net = cnn_cifar_init('networkType', opts.networkType) ;
32 -    case 'nin'
```

cnn_cifa...

Demonstrates MatConvNet on CIFAR-10

cnn_cifar(...
getBatch(o...
getSimple...
getDagNN...
getCifarIm...

Command Window

```
train: epoch 44:  50/500: 642.1 (639.2) Hz objective: 0.168 top1err:
train: epoch 44:  51/500: 642.5 (662.4) Hz objective: 0.169 top1err:
train: epoch 44:  52/500: 642.4 (637.2) Hz objective: 0.168 top1err:
train: epoch 44:  53/500: 642.6 (652.2) Hz objective: 0.169 top1err:
train: epoch 44:  54/500: 642.9 (659.1) Hz objective: 0.168 top1err:
train: epoch 44:  55/500:
```

Wo...

Name

ans
best
bestScor
im
im_
net
res
scores

Busy | cnn_cifar | Ln 7 | Col 7

MATLAB R2019a - academic use

HOME | PLOTS | APPS | EDITOR | PUBLISH | VIEW

New | Open | Save | Find Files | Compare | Print — FILE
Go To | Find — NAVIGATE
Insert | Comment | Indent — EDIT
Breakpoints — BREAKPOINTS
Run | Run and Advance | Run Section | Advance | Run and Time — RUN

/ ▸ Users ▸ apple ▸ Desktop ▸ Jiann-Ming Wu ▸ code2019 ▸ MatConvNet ▸ matconvnet-1.0-beta25 ▸ examples ▸ cifar

Current Fo...

Name

- cnn_cifar.m
- cnn_cifar_i...
- cnn_cifar_i...

Editor - /Users/apple/Desktop/Jiann-Ming Wu/code2019/MatConvNet/matconvnet-1.0-beta25_2019_10/examples/cifar

+3 | demo_taylor_ex2_jmwu.m | script_load_scan_lattice_structured.m | script_render_fittings.m | script_load_scan.m

```
22 -    opts = vl_argparse(opts, varargin) ;
23 -    if ~isfield(opts.train, 'gpus'), opts.train.gpus = []
24
25      % ----------------
```

Command Window

```
val: epoch 45:  97/10
val: epoch 45:  98/10
val: epoch 45:  99/10
val: epoch 45: 100/10

ans =

  struct with fields:

    layers: {1×13 cel
      meta: [1×1 stru

>>
```

cnn_cifar....

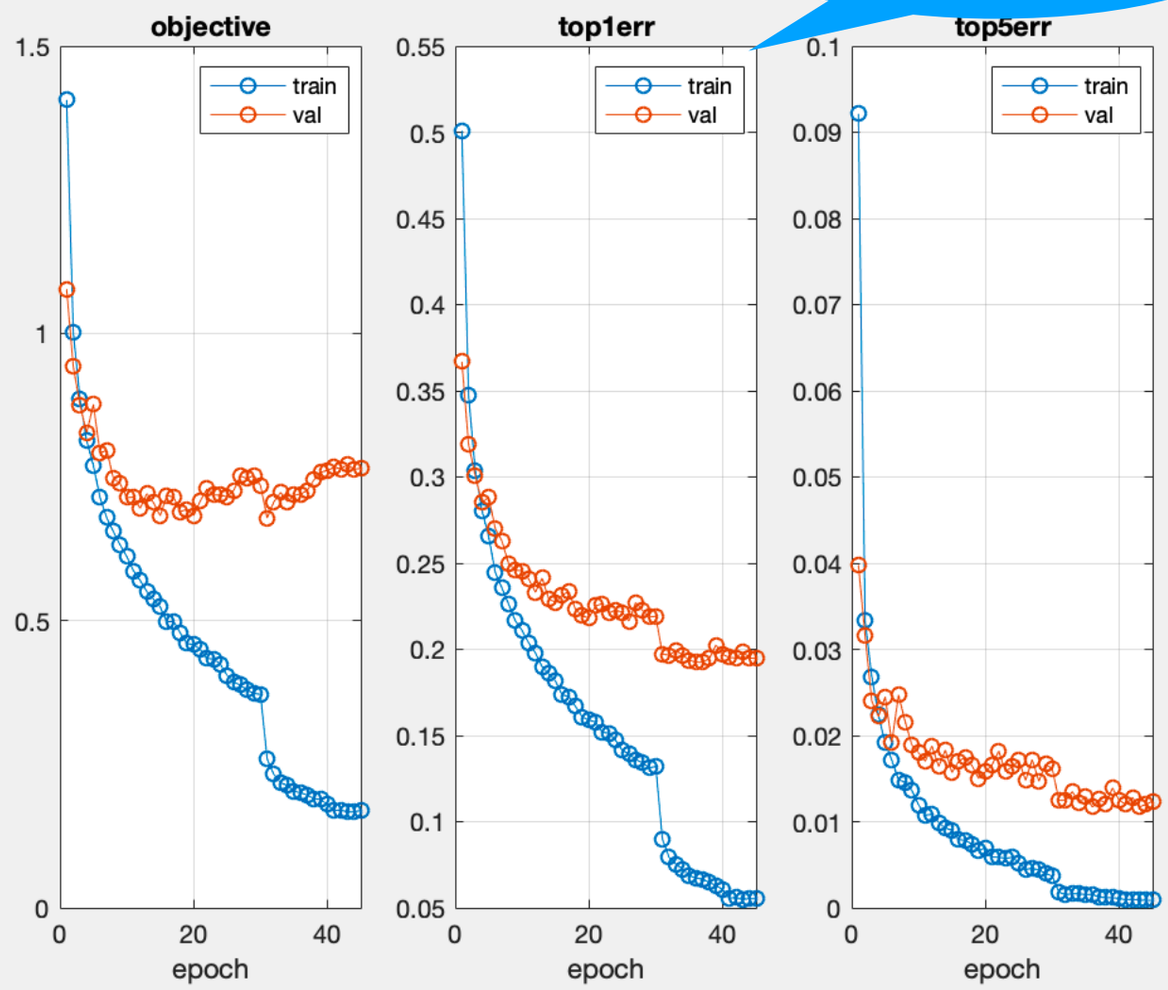
Demonstrates MatConvNet on CIFAR-10

- cnn_cifar(va...
- getBatch(opts)
- getSimpleN...
- getDagNNBa...
- getCifarImd...

Figure 1

File Edit View Insert Tools Desktop Window Help

Ex-II



/Users/apple/Desktop/Jiann-Ming Wu/code2019/MatConvNet/matconvnet-1.0-beta25/examples/cifar/cnn_cifar_init.m

PUBLISH VIEW

Find Files, Compare, Print, Save — FILE
Go To, Find — NAVIGATE
Insert, Comment, Indent — EDIT
Breakpoints — BREAKPOINTS
Run, Run and Advance, Run Section, Advance, Run and Time — RUN

Deep Convolution neural network

```
% Define network CIFAR10-quick
net.layers = {} ;

% Block 1
net.layers{end+1} = struct('type', 'conv', ...
                           'weights', {{0.01*randn(5,5,3,32, 'single'), zeros(1, 32, 'single')}
                           'learningRate', lr, ...
                           'stride', 1, ...
                           'pad', 2) ;
net.layers{end+1} = struct('type', 'pool', ...
                           'method', 'max', ...
                           'pool', [3 3], ...
                           'stride', 2, ...
                           'pad', [0 1 0 1]) ;
net.layers{end+1} = struct('type', 'relu') ;

% Block 2
net.layers{end+1} = struct('type', 'conv', ...
                           'weights', {{0.05*randn(5,5,32,32, 'single'), zeros(1,32,'single')}
                           'learningRate', lr, ...
                           'stride', 1, ...
```

Training set

Testing set

HOME | PLOTS | APPS | EDITOR | PUBLISH | VIEW

Search Documentation

New | Open | Save | Find Files | Compare | Print — FILE

Go To | Find — NAVIGATE

Insert | Comment | Indent — EDIT

Breakpoints — BREAKPOINTS

Run | Run and Advance | Run Section | Advance | Run and Time — RUN

Users ▸ apple ▸ Desktop ▸ Jiann-Ming Wu ▸ code2019 ▸ MatConvNet ▸ matconvnet-1.0-beta25 ▸ data ▸ cifar ▸ cifar-10-batches-mat

Current Folder

Name

- batches.meta.mat
- data_batch_1.mat
- data_batch_2.mat
- data_batch_3.mat
- data_batch_4.mat
- data_batch_5.mat
- readme.html
- test_batch.mat

Details

Editor - /Users/apple/Desktop/Jiann-Ming Wu/code2019/MatConvNet/matconvnet-1.0-beta25/examples/cifar/cnn_cifar_init.m

+4 | script_load_scan_lattice_structured.m | script_render_fittings.m | script_load_scan.m | cnn_cifar.m | cnn_cifar_init.m

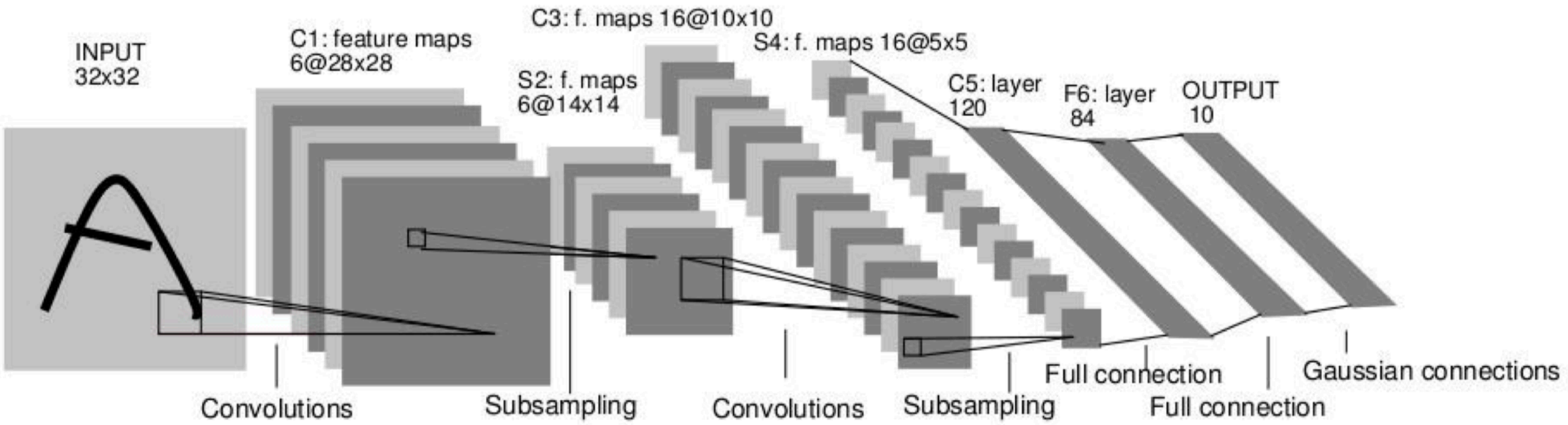
```
47                                  'pad', [0 1 0 1]) ; % Emulate caffe
48
49      % Block 4
50      net.layers{end+1} = struct('type', 'conv', ...
51                                 'weights', {{0.05*randn(4,4,64,64, 'si
52                                 'learningRate', lr, ...
53                                 'stride', 1, ...
54                                 'pad', 0) ;
55      net.layers{end+1} = struct('type', 'relu') ;
56
57      % Block 5
58      net.layers{end+1} = struct('type', 'conv', ...
59                                 'weights', {{0.05*randn(1,1,64,10, 'si
60                                 'learningRate', .1*lr, ...
61                                 'stride', 1, ...
```

Learning Multiple Layers of Features from Tiny Images

Alex Krizhevsky

April 8, 2009

LeNet 5



Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner,
Gradient-based learning applied to document recognition, Proc. IEEE 86(11): 2278–2324, 1998.

```
% Define network CIFAR10-quick
net.layers = {} ;

% Block 1
net.layers{end+1} = struct('type', 'conv', ...
                           'weights', {{0.01*randn(5,5,3,32, 'single'), zeros(1, 32, 'single')}}, ...
                           'learningRate', lr, ...
                           'stride', 1, ...
                           'pad', 2) ;
net.layers{end+1} = struct('type', 'pool', ...
                           'method', 'max', ...
                           'pool', [3 3], ...
                           'stride', 2, ...
                           'pad', [0 1 0 1]) ;
net.layers{end+1} = struct('type', 'relu') ;

% Block 2
net.layers{end+1} = struct('type', 'conv', ...
                           'weights', {{0.05*randn(5,5,32,32, 'single'), zeros(1,32,'single')}}, ...
                           'learningRate', lr, ...
                           'stride', 1, ...
                           'pad', 2) ;
net.layers{end+1} = struct('type', 'relu') ;
net.layers{end+1} = struct('type', 'pool', ...
                           'method', 'avg', ...
                           'pool', [3 3], ...
                           'stride', 2, ...
                           'pad', [0 1 0 1]) ; % Emulate caffe

% Block 3
net.layers{end+1} = struct('type', 'conv', ...
                           'weights', {{0.05*randn(5,5,32,64, 'single'), zeros(1,64,'single')}}, ...
                           'learningRate', lr, ...
                           'stride', 1, ...
                           'pad', 2) ;
net.layers{end+1} = struct('type', 'relu') ;
net.layers{end+1} = struct('type', 'pool', ...
                           'method', 'avg', ...
                           'pool', [3 3], ...
                           'stride', 2, ...
                           'pad', [0 1 0 1]) ; % Emulate caffe

% Block 4
net.layers{end+1} = struct('type', 'conv', ...
                           'weights', {{0.05*randn(4,4,64,64, 'single'), zeros(1,64,'single')}}, ...
                           'learningRate', lr, ...
                           'stride', 1, ...
                           'pad', 0) ;
net.layers{end+1} = struct('type', 'relu') ;

% Block 5
net.layers{end+1} = struct('type', 'conv', ...
                           'weights', {{0.05*randn(1,1,64,10, 'single'), zeros(1,10,'single')}}, ...
                           'learningRate', .1*lr, ...
                           'stride', 1, ...
                           'pad', 0) ;

% Loss layer
net.layers{end+1} = struct('type', 'softmaxloss') ;
```

Alex's CIFAR-10 tutorial, Caffe style

Alex Krizhevsky's cuda-convnet details the model definitions, parameters, and training procedure for good performance on CIFAR-10. This example reproduces his results in Caffe.

We will assume that you have Caffe successfully compiled. If not, please refer to the Installation page. In this tutorial, we will assume that your caffe installation is located at `CAFFE_ROOT`.

We thank @chyojn for the pull request that defined the model schemas and solver configurations.

This example is a work-in-progress. It would be nice to further explain details of the network and training choices and benchmark the full training.

Prepare the Dataset

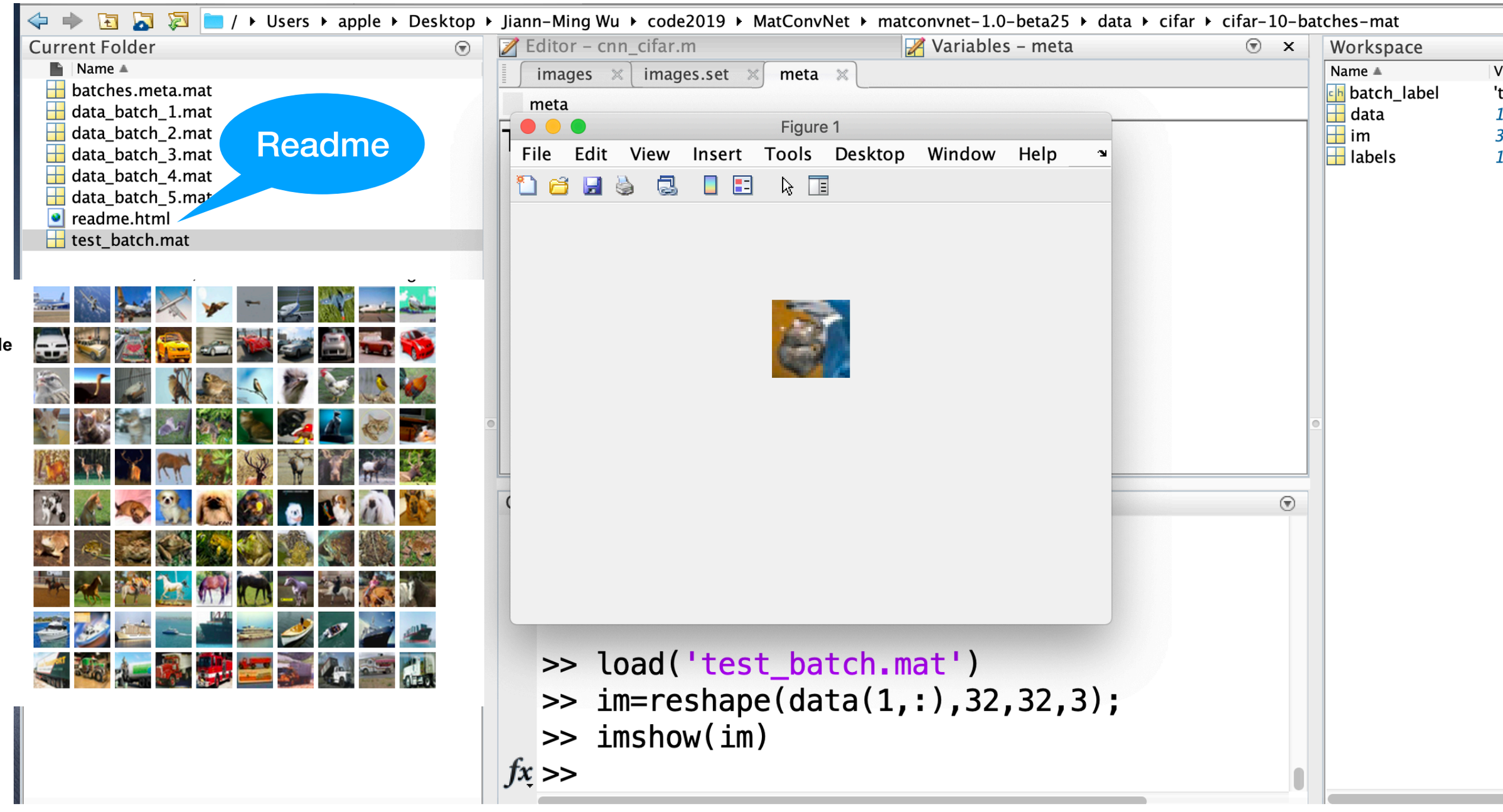
You will first need to download and convert the data format from the CIFAR-10 website. To do this, simply run the following commands:

```
cd $CAFFE_ROOT
./data/cifar10/get_cifar10.sh
./examples/cifar10/create_cifar10.sh
```

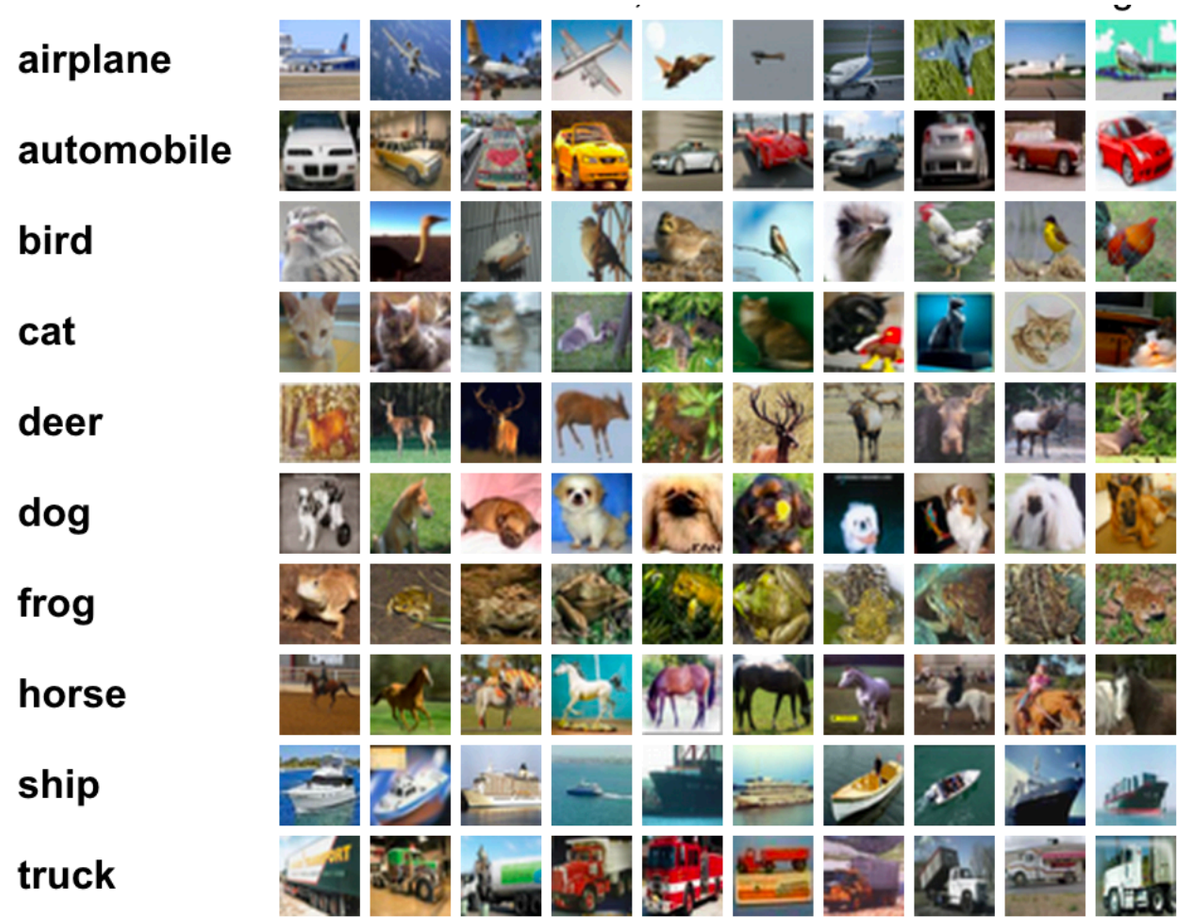
If it complains that `wget` or `gunzip` are not installed, you need to install them respectively. After running the script there should be the dataset, `./cifar10-leveldb`, and the data set image mean `./mean.binaryproto`.

Architecture and Deep Learning

Example : network CIFAR10-quick


/ ▸ Users ▸ apple ▸ Desktop ▸ Jiann-Ming Wu ▸ code2019 ▸ MatConvNet ▸ matconvnet-1.0-beta25 ▸ data ▸ cifar ▸ cifar-10-batches-mat
Current Folder
Name
batches.meta.mat
data_batch_1.mat
data_batch_2.mat
data_batch_3.mat
data_batch_4.mat
data_batch_5.mat
readme.html
test_batch.mat
Readme
airplane
automobile
bird
cat
deer
dog
frog
horse
ship
truck
Editor - cnn_cifar.m
images
images.set
meta
meta
Figure 1
File Edit View Insert Tools Desktop Window Help
Variables - meta
Workspace
Name
batch_label
data
im
labels
>> load('test_batch.mat')
>> im=reshape(data(1,:),32,32,3);
>> imshow(im)
>>

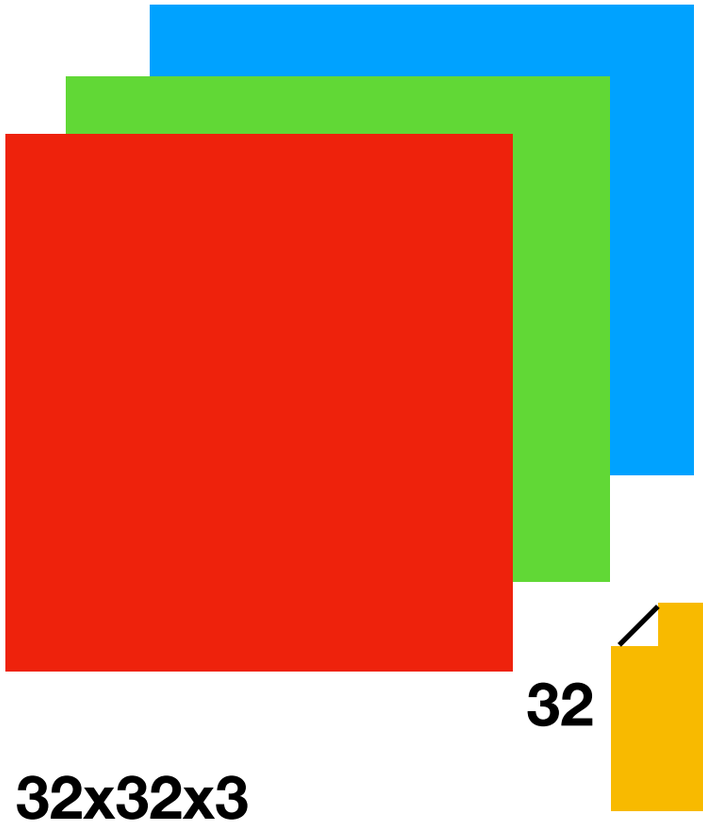
```
>> load('test_batch.mat')
>> im=reshape(data(1,:),32,32,3);
>> imshow(im)
>> labels(1)

ans =

    uint8

     3
```



Mid-term project

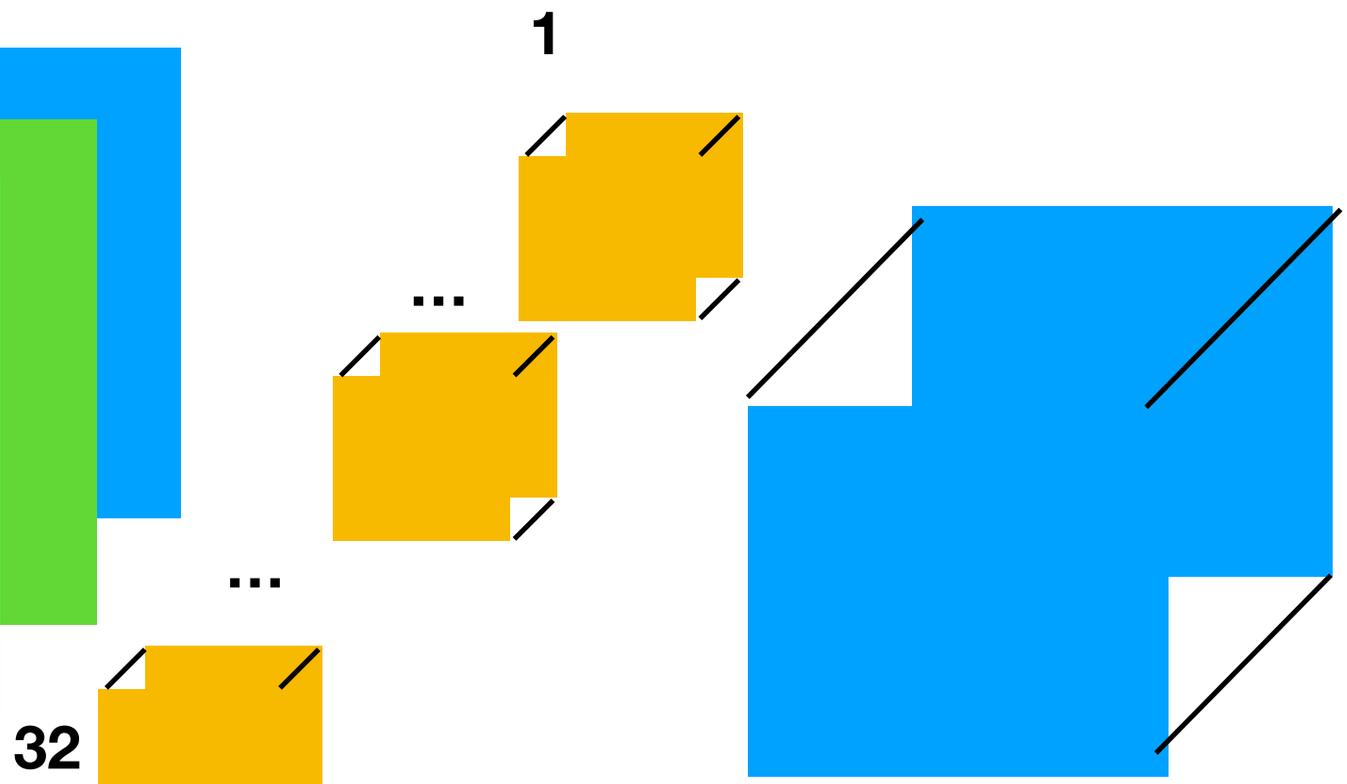
Exercise 1
Execute cnn_cifar.m one epoch and trace vl_simplenn.m.
State the input size and output size of each layer and sketch the architecture

100 images per batch



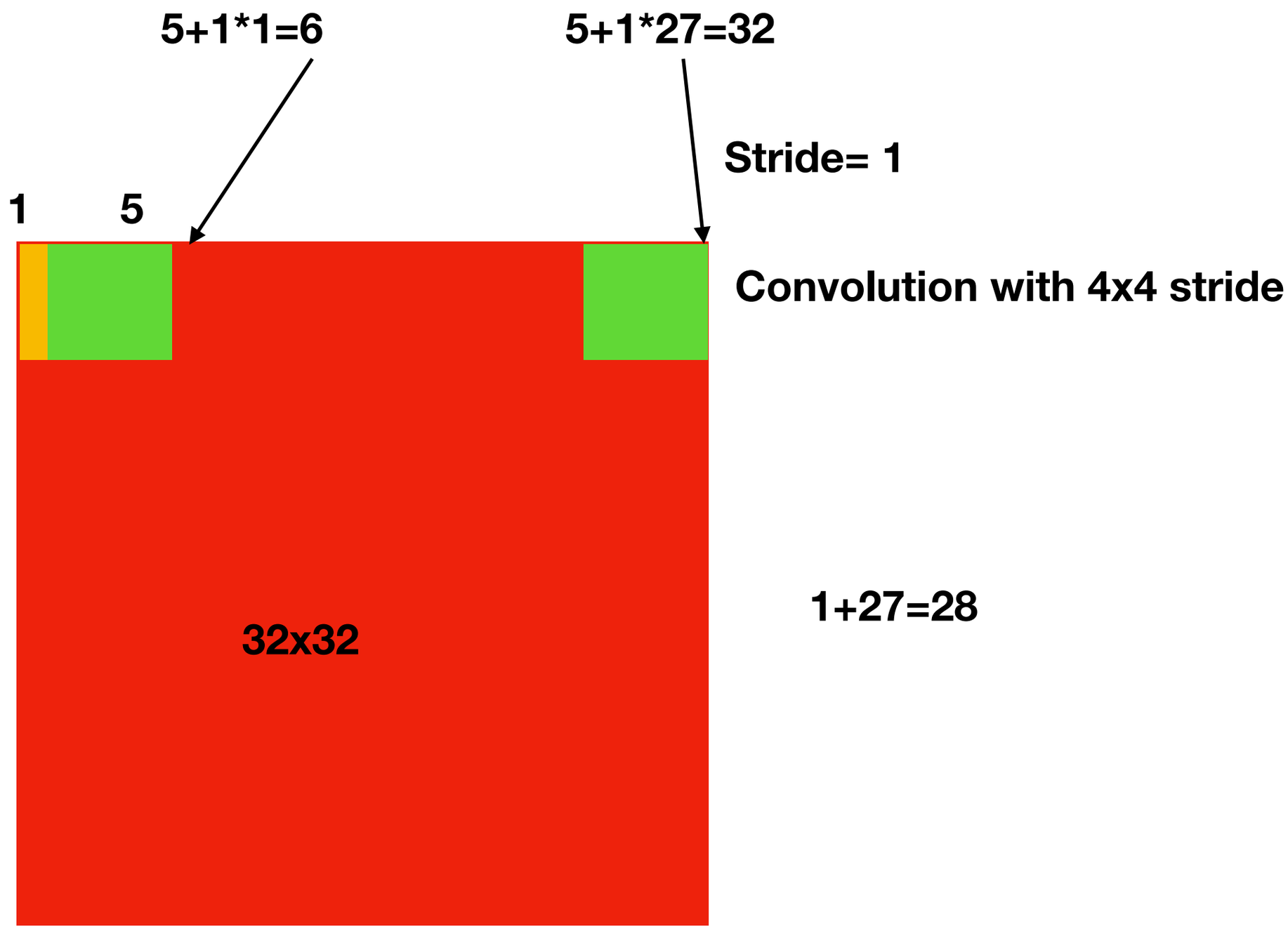
32x32x3

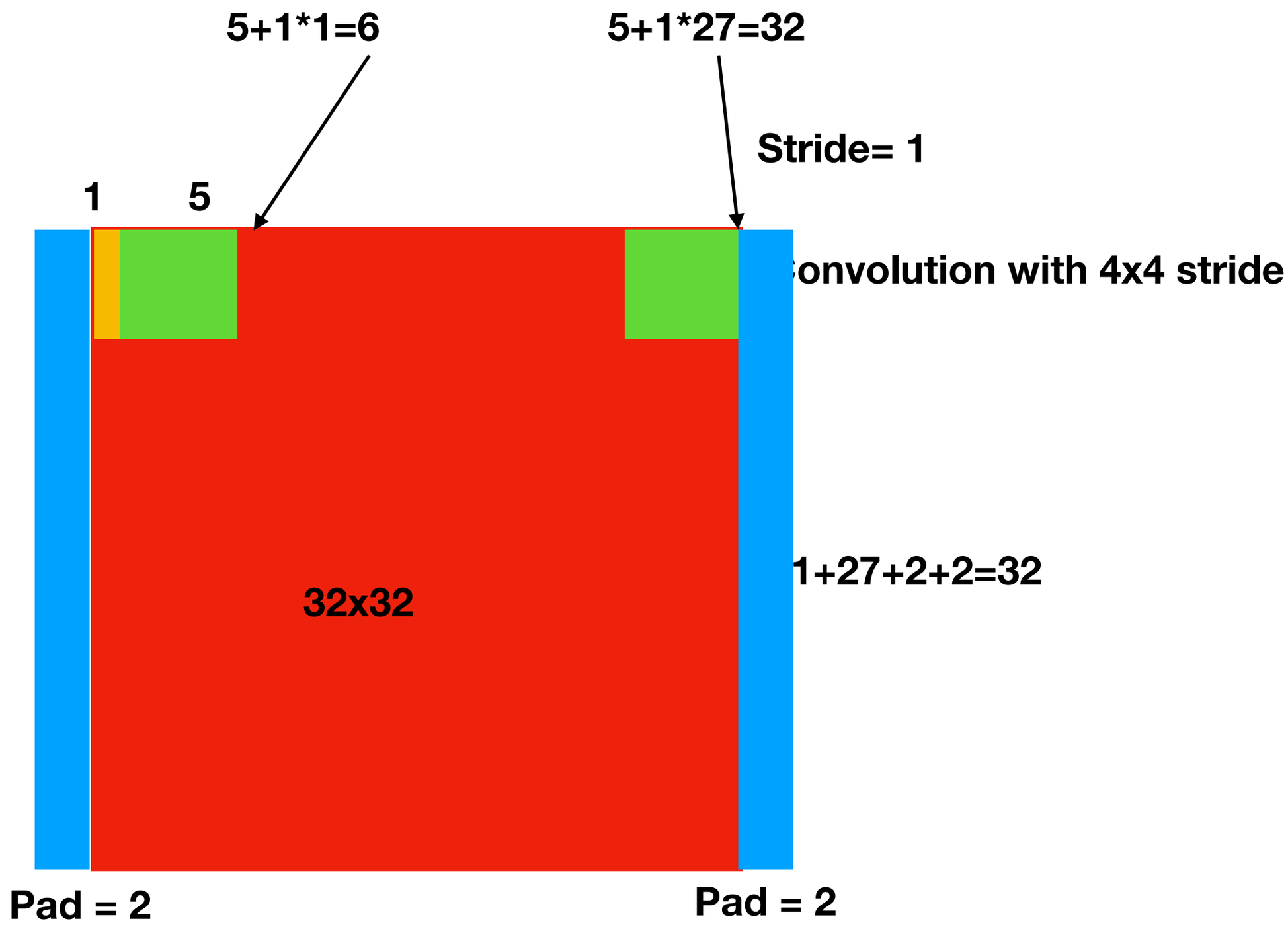
Convolution



5x5x3

32x32x32


5+1*1=6
5+1*27=32
Stride= 1
1
5
Convolution with 4x4 stride
32x32
1+27=28


5+1*1=6
5+1*27=32
Stride= 1
1
5
onvolution with 4x4 stride
32x32
1+27+2+2=32
Pad = 2
Pad = 2

FILE | NAVIGATE | EDIT | BREAKPOINTS | DEBUG

/ ▸ Users ▸ apple ▸ Desktop ▸ Jiann-Ming Wu ▸ code2019 ▸ MatConvNet ▸ matconvnet-1.0-beta25 ▸ data ▸ cifar-lenet

Current Folder

Name

- net-epoch-25.mat
- net-epoch-26.mat
- net-epoch-27.mat
- net-epoch-28.mat
- net-epoch-29.mat
- net-epoch-30.mat
- net-epoch-31.mat
- net-epoch-32.mat
- net-epoch-33.mat
- net-epoch-34.mat
- net-epoch-35.mat
- net-epoch-36.mat
- net-epoch-37.mat
- net-epoch-38.mat
- net-epoch-39.mat
- net-epoch-40.mat
- net-epoch-41.mat
- net-epoch-42.mat
- net-epoch-43.mat
- net-epoch-44.mat
- net-train.pdf

Details

Select a file to

delete net-epoch-44.mat

Editor - /Users/apple/Desktop/Jiann-Ming Wu/code2019/MatConvNet/matconvnet-1.0-...

+18 | ToolboxDemo.m | demo_imagenet.m | vl_simplenn.m | cnn_cifar.m | +

```
299 -        case 'conv'
300 -          res(i+1).x = vl_nnconv(res(i).x, l.
301              'pad', l.pad, ...
302              'stride', l.stride, ...
303              'dilate', l.dilate, ...
304              l.opts{:}, ...
305              cudnn{:}) ;
306
307 -        case 'convt'
```

Command Window

```
  data mem|   1MB|   12MB|     3MB|    3MB|    3MB|
 param mem|   n/a|   10KB|     0B|     0B| 100KB|

parameter memory|569KB (1.5e+05 parameters)|
      data memory| 31MB (for batch size 100)|

...n: resuming by loading epoch 44
...: epoch 45:   1/500:294 for i=1:n
K>>
```

MATLAB R2019a - academic use

HOME | PLOTS | APPS | EDITOR | PUBLISH | VIEW

Search Documentation | Sign In

New | Open | Save | Find Files | Compare | Print | Go To | Find | Insert | Comment | Indent | Breakpoints | Continue | Step | Step In | Step Out | Run to Cursor | Function Call Stack: vl_simplenn | Quit Debugging

FILE | NAVIGATE | EDIT | BREAKPOINTS | DEBUG

/ ▸ Users ▸ apple ▸ Desktop ▸ Jiann-Ming Wu ▸ code2019 ▸ MatConvNet ▸ matconvnet-1.0-beta25 ▸ examples ▸ cifar

Current Folder

Name

cnn_cifar.m
cnn_cifar_init.m
cnn_cifar_init_nin.m

Details

Select a file to view details

Editor - /Users/apple/Desktop/Jiann-Ming Wu/code2019/MatConvNet/matconvnet-1.0-...

+18 | ToolboxDemo.m | demo_imagenet.m | vl_simplenn.m | cnn_cifar.m

```
290      % -------------------------------------------------------------------------
291      %
292      % -------------------------------------------------------------------------
293
294      for i=1:n
295        if opts.skipForward, break; end;
296        l = net.layers{i} ;
297        res(i).time = tic ;
298        switch l.type
```

Command Window

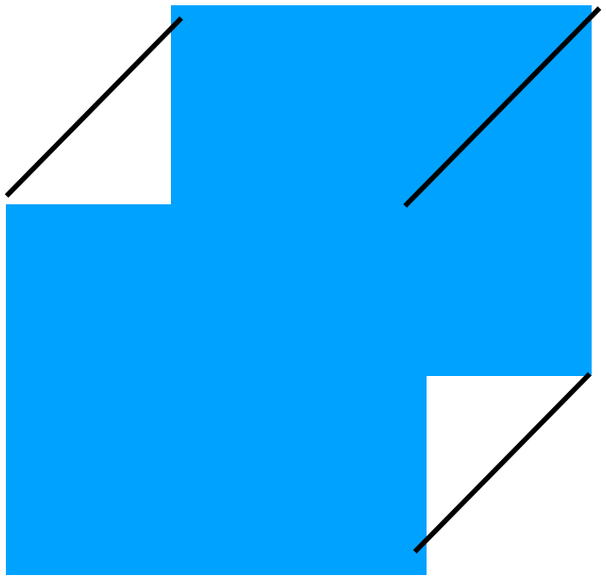
```
  data mem|   1MB|   12MB|     3MB|    3MB|
 param mem|   n/a|   10KB|      0B|

parameter memory|569KB (1.5e+05 para
     data memory| 31MB (for batch size

cnn_train: resuming by loading epoch 44
train: epoch 45:   1/500:294 for i=1:n
K>>
```

Workspace - vl_si...

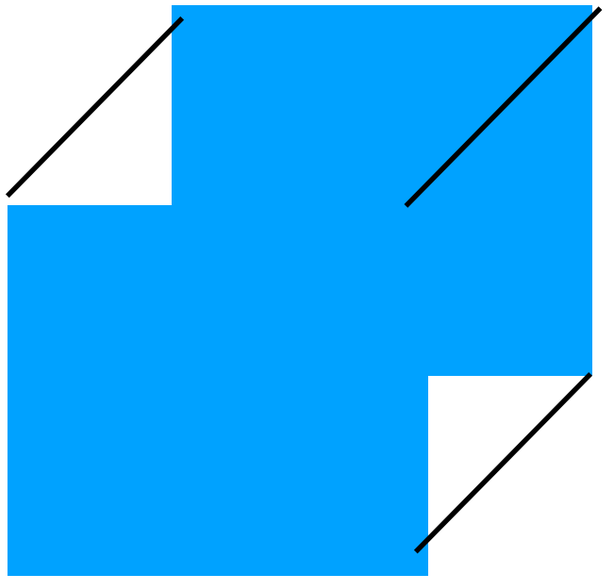
Name	Value
backPropLim	1
bnormCudnn	1x1 ce
cudnn	1x1 ce
doder	1
dzdy	1
gpuMode	0
n	13
net	1x1 st
opts	1x1 st
res	1x14
testMode	0
varargin	1x16
x	4-D s

Execute cnn_cifar.m one epoch

Max pooling



??x??x??



??x??x??

Exercise II

1. Use deep network designer App to design cifar10-quick
2. Train and test your deep network

Exercise III

1. Use deep network designer App to design your deep network for improving cifar10-quick
2. Train and test your deep network

Deep Network Designer

DESIGNER

New Import | Duplicate Cut Copy Paste | Fit to View Zoom In Zoom Out | Auto Arrange | Analyze | Export

FILE BUILD NAVIGATE LAYOUT ANALYSIS EXPORT

LAYER LIBRARY

Filter layers...

INPUT

- imageInputLayer
- image3dInputLayer
- sequenceInputLayer
- roiInputLayer

CONVOLUTION AND FULLY CONNECTED

- convolution2dLayer
- convolution3dLayer
- groupedConvolution2dLayer
- transposedConv2dLayer
- transposedConv3dLayer
- fullyConnectedLayer

SEQUENCE

- lstmLayer
- bilstmLayer

imageinput
imageInputLayer

conv
convolution2dL...

PROPERTIES

Number of layers	2
Number of connections	0
Input type	Image
Output type	None