Tutorial 3. Deep Learning Toolbox

Xingyu ZENG(xyzeng@ee.cuhk.edu.hk)

DeepLearnToolbox

- A open-source Matlab toolbox for Deep Learning

- You can download in
  https://github.com/rasmusbergpalm/DeepLearnToolbox

- If you use this toolbox in your research please cite
  @MASTERSTHESIS\{IMM2012-06284, author = "R. B. Palm", title = "Prediction as a candidate for learning deep hierarchical models of data", year = "2012", }

DeepLearnToolbox

- Advantage
  - Matlab, easy to use
  - Open-source
- Disadvantage
  - Only CPU version, slow

Install Steps

1. Download the toolbox,
2. Addpath(genpath('DeepLearnToolbox'))

DeepLearnToolbox

- A Matlab toolbox for Deep Learning
  - NN/ - A library for Feedforward Backpropagation Neural Networks
  - CNN/ - A library for Convolutional Neural Networks
  - DBN/ - A library for Deep Belief Networks
  - SAE/ - A library for Stacked Auto-Encoders
  - CAE/ - A library for Convolutional Auto-Encoders
  - util/ - Utility functions used by the libraries
  - data/ - Data used by the examples
  - tests/ - unit tests to verify toolbox is working

DeepLearnToolbox

Feedforward Backpropagation Neural Networks

- Common Function
  - nnsetup.m
    - To setup one network
  - nntrain.m
    - To train one network
  - nnpredict.m
    - To test samples with one network

DeepLearnToolbox

- nnsetup.m

Usage example:

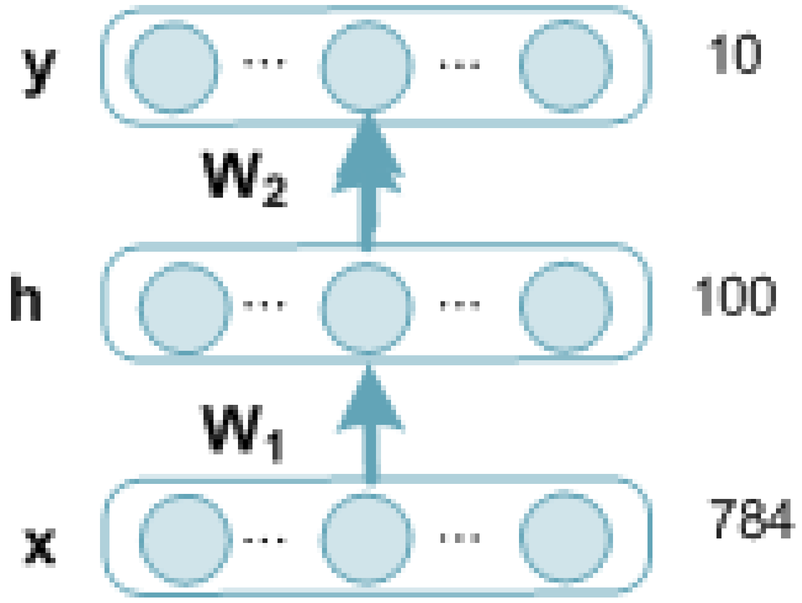
nn = nnsetup([784 100 10]); % to build up one three layers network

nn.activation_function = 'sigm';

nn.output= 'softmax';

h=sigmoid(W_1*x);

y=softmax(W_2*h);



DeepLearnToolbox

- nntrain.m

Usage example:

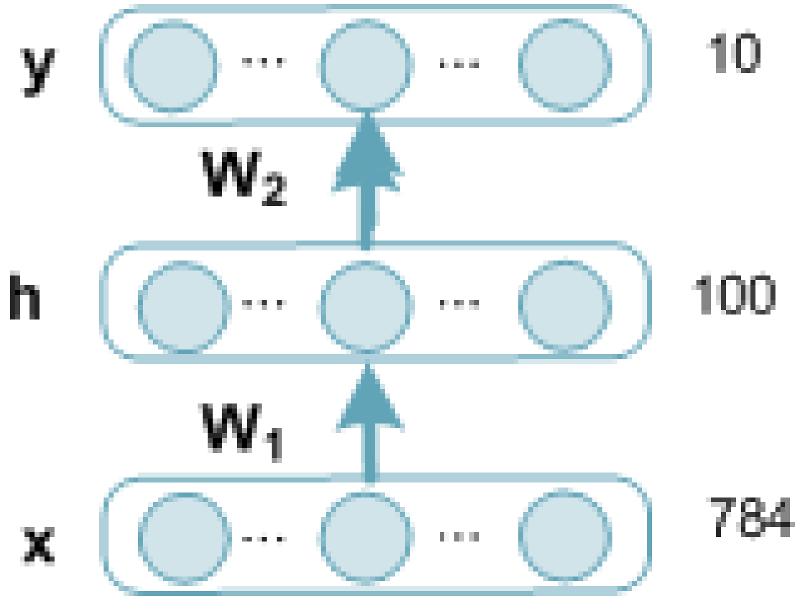
```
nn.learningRate = 0.1;
opts.numepochs =  1; %Number of full sweeps through data
opts.batchsize = 100;  %Take a mean gradient step over this many
                        %samples

nn = nntrain(nn, train_x, train_y, opts);
```

DeepLearnToolbox

- nnpredict.m

Usage example:

```
labels = nnpredict(nn, test_x);
```



Notes:

1. labels, the classes predicted by nn
2. nn.a{end}, the values of the output layer

```
[~,labels]=max(nn.a{end},[],2);
```

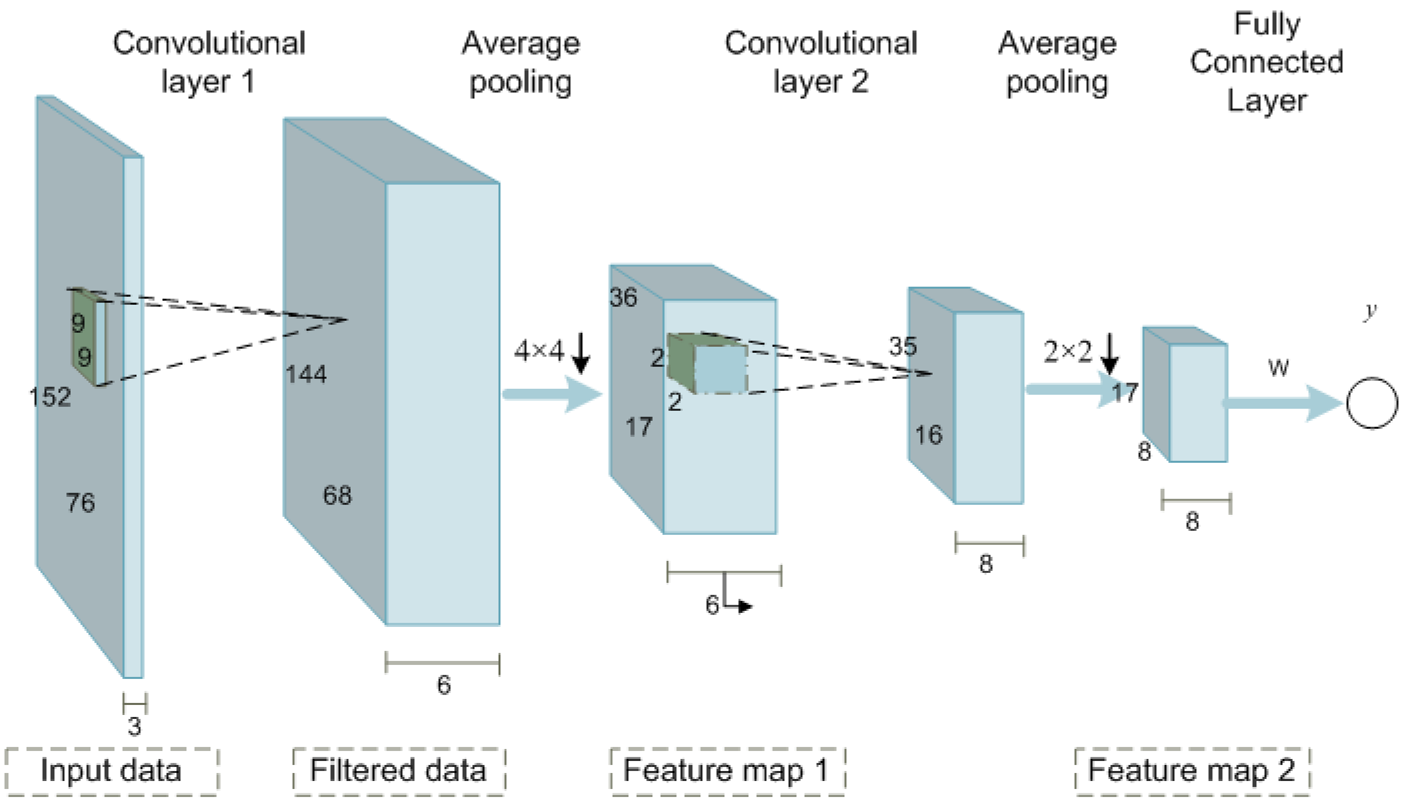
DeepLearnToolbox

Convolutional Neural Networks

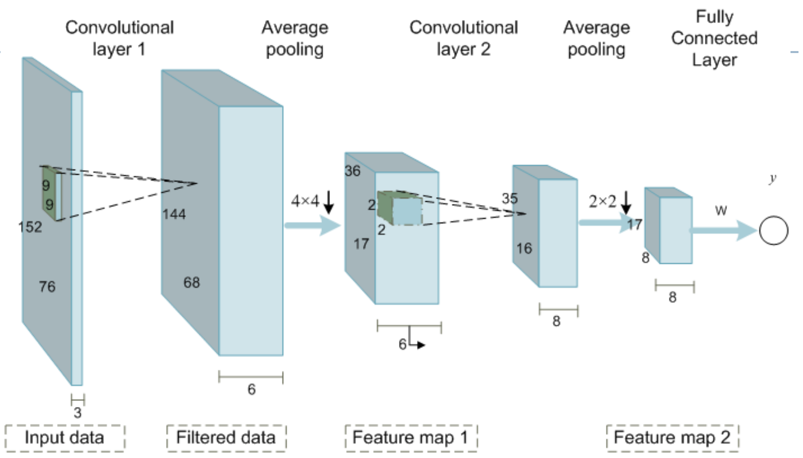
- Common Function
  - cnnsetup.m
    - To setup one convolutional network
  - cnntrain.m
    - To train one convolutional network
  - cnnff.m
    - Forward step with one convolutional network
  - cnntest.m
    - To test samples with one convlutional network

DeepLearnToolbox

- Suppose input size: 152*76*3



DeepLearnToolbox



- cnnsetup.m

Usage example:

```
cnn.layers = {
    struct('type', 'i') %input layer
    struct('type', 'c', 'outputmaps', 6, 'kernelsize', 9) %convolution layer
    struct('type', 's', 'scale', 4) %sub sampling layer
    struct('type', 'c', 'outputmaps', 8, 'kernelsize', 2) %convolution layer
    struct('type', 's', 'scale', 2) %subsampling layer
};

% the size of {train_x, train_y} is useful for setup cnn
cnn = cnnsetup(cnn, train_x, train_y);
```

DeepLearnToolbox

- cnntrain.m

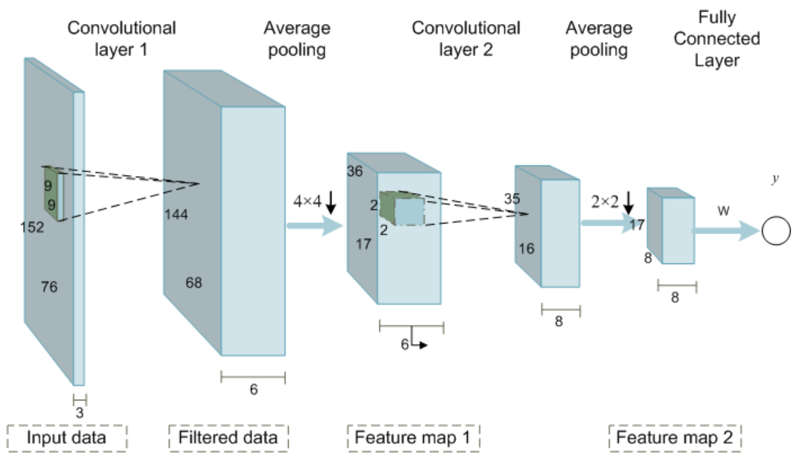
Usage example:

opts.alpha = 1; % learning rate

opts.batchsize = 50;

opts.numepochs = 1;

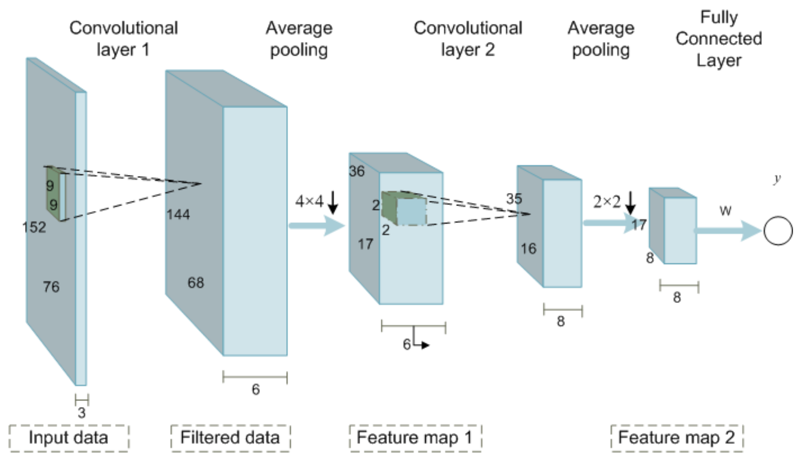
cnn = cnntrain(cnn, train_x, train_y, opts);



DeepLearnToolbox

- cnnff.m

Usage example:



```
cnn = cnnff(cnn, x);
```

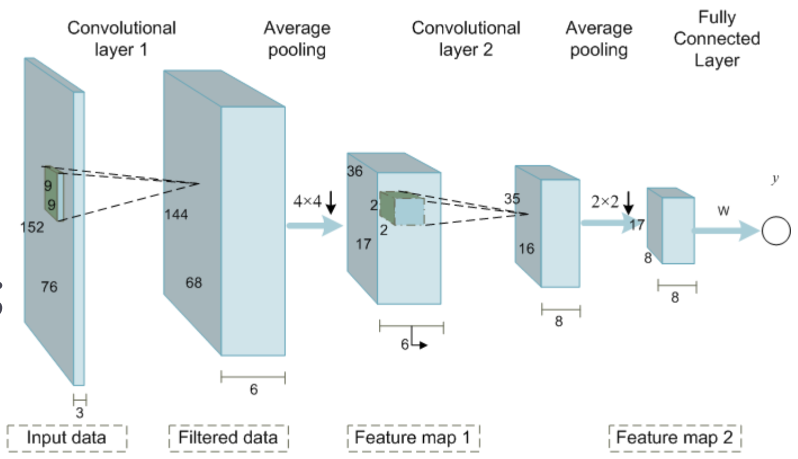
Notes:

1. cnn.o, the output values of the output layer
2. cnn.fv, the feature of the input samples
3. cnn.o=sigm(cnn.ffW * cnn.fv + repmat(cnn.ffb, 1, size(cnn.fv, 2)));

DeepLearnToolbox

- cnntest.m

Usage example:



```
[er, bad] = cnntest(cnn, test_x, test_y);
```

Notes:

1. er, error fraction value
2. bad, index of misclassified testing samples

DeepLearnToolbox

- More examples can be found in

https://github.com/rasmusbergpalm/DeepLearnToolbox/blob/master/README.md

- More details about deep learning,

Book, 'Learning Deep Architectures for AI'