Jinchao Xu

Deep Learning Algorithms and Analysis

Summer 2020

Contributors:

This set of notes are based on contributions from many of graduate students, post-doctoral fellows and other collaborators. Here is a partial list:

Juncai He, Qingguo Hong, Li Jiang.....

Contents

1 Machine Learning and Image Classification 5
- 1.1 A basic machine learning problem: image classification 5
- 1.2 Image classification problem 8
- 1.3 Some popular data sets in image classification 9
  - 1.3.1 MNIST(Modified National Institute of Standards and Technology Database) 9
  - 1.3.2 CIFAR 10
  - 1.3.3 ImageNet 12
- 1.4 Classification and decision boundaries 13
  - 1.4.1 Linear models: decision boundaries given by hyper-planes . . 14
  - 1.4.2 How to find these hyperplanes: logistic regressions 15

References 17

1

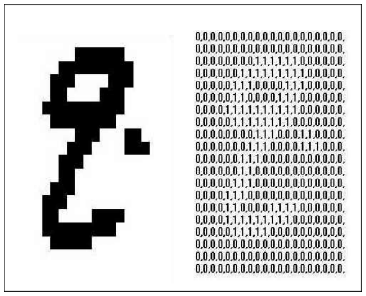
Machine Learning and Image Classification

1.1 A basic machine learning problem: image classification

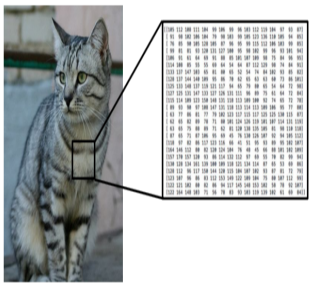
A basic AI problem: classification

- Can a machine (function) tell the difference ?

Mathematically, gray-scale image can be just taken as matrix in $\mathbb{R}^{n_0 \times n_0}$

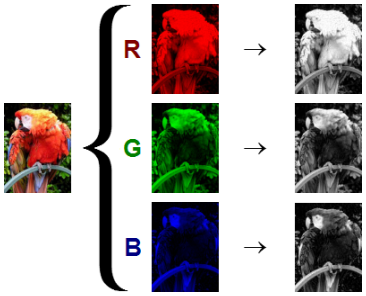


The next figure shows different result from: human vision and computer representation:



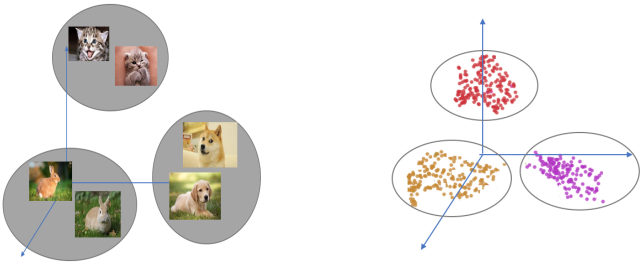
- An image is just a big grid of numbers between [0, 255]
  - e.g. 800×600×3 (3 channels RGB)

Furthermore, color image can be taken as 3D tensor (matrix with 3 channel (RGB)) in $\mathbb{R}^{n_0 \times n_0 \times 3}$



Then, let think about the general supervised learning case.

- Each image = a big vector of pixel values
  - $d = 1280 \times 720 \times 3$ (width× height × RGB channel) ≈ 3M.
- 3 different sets of points in $\mathbb{R}^d$, are they separable?



- Mathematical problem: Find $f(\cdot;\boldsymbol{\theta}) : \mathbb{R}^d \to \mathbb{R}^3$ such that:

$$f(\text{  };\boldsymbol{\theta}) \approx \begin{pmatrix}1\\0\\0\end{pmatrix} \quad f(\text{  };\boldsymbol{\theta}) \approx \begin{pmatrix}0\\1\\0\end{pmatrix} \quad f(\text{  };\boldsymbol{\theta}) \approx \begin{pmatrix}0\\0\\1\end{pmatrix}$$

  - Function interpolation
  - data fitting

  Then, the next question is how to formulate "learning"?

- Data: $\{x_j, y_j\}_{j=1}^N$
- Find f^* in some function class such that $f^*(x_j) \approx y_j$.
- Mathematically: solve the optimization problem by parameterizing the abstract function class

$$\min_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}) \tag{1.1}$$

  where

$$\mathcal{L}(\boldsymbol{\theta}) := \mathbb{E}_{(x,y)\sim\mathcal{D}}[\ell(f(x;\boldsymbol{\theta}), y)] \approx L(\boldsymbol{\theta}) := \frac{1}{N}\sum_{j=1}^{N} \ell(y_j, f(x_j;\boldsymbol{\theta}))$$

  Here $\ell(y_j, f(x_j;\boldsymbol{\theta}))$ is the general distance between real label y_j and predicted label $f(x_j;\boldsymbol{\theta})$. Two commonly used distances are
  - ℓ^2 distance:
$$\ell(y_j, f(x_j;\boldsymbol{\theta})) = \|y_j - f(x_j;\boldsymbol{\theta})\|^2.$$
  - KL-divergence distance:
$$\ell(y_j, f(x_j;\boldsymbol{\theta})) = \sum_{i=1}^{k} [y_j]_i \log \frac{[y_j]_i}{[f(x_j;\boldsymbol{\theta})]_i}.$$

- Application: image classification:

$$f(\text{  };\boldsymbol{\theta}) = \begin{pmatrix}0.7\\0.2\\0.1\end{pmatrix} \implies \text{  } = \text{cat}$$

1.2 Image classification problem

We consider a basic machine learning problem for classifying a collection of images into k distinctive classes. As an example, we consider a two-dimensional image which is usually represented by a tensor

$$x \in \mathbb{R}^{n_0 \times n_0 \times c} = \mathbb{R}^d.$$

Here $n_0 \times n_0$ is the original image resolution and

$$c = \begin{cases} 1 \text{ for grayscale image,} \\ 3 \text{ for color image.} \end{cases} \tag{1.2}$$

A typical supervised machine learning problem begins with a data set (training data)

$$D := \{(x_j, y_j)\}_{j=1}^N,$$

with

$$A = \{x_1, x_2, \cdots, x_N\} \quad \text{with} \quad A = A_1 \cup A_2 \cup \cdots \cup A_k,\ A_i \cap A_j = \emptyset, \forall i \neq j.$$

and $y_j \in \mathbb{R}^k$ is the label for data x_j, with $y_i[i]$ as the probability for x_i in classes i or $x_j \in A_i$.

Here for image classification problem,

$$y_i = e_{i_j}, \tag{1.3}$$

is $x_j \in A_{i_j}$ or we say x_j has real label i_j.

Roughly speaking, a supervised learning problem can be thought as a data fitting problem in a high dimensional space $\mathbb{R}^d$. Namely, we need to find a mapping $f : \mathbb{R}^d \mapsto \mathbb{R}^k$, such that, for a given data (x_j, y_j),

$$f(x_j) \approx y_j = e_{i_j} \in \mathbb{R}^k, \tag{1.4}$$

for all $x_j \in A$. For the general setting above, we use a probatilistic model for understanding the output $f(x) \in \mathbb{R}^k$ as a discrete distribution on $\{1, \cdots, k\}$, with $[f(x)]_i$ as the probability for x in the class i, namely

$$0 \leq [f(x)]_i \leq 1, \quad \sum_{i=1}^k [f(x)]_i = 1. \tag{1.5}$$

At last, we finish our model with a simple strategy to choose

$$\arg\max_i \{[f(x)]_i \ : \ i = 1 : k\}, \tag{1.6}$$

as the label for a test data x, which ideally is close to (1.4). The remaining key issue is the construction of the classification mapping f.

Generally speaking, there will be a test set

$$T = \{(x_j, y_j)\}_{j=1}^M, \tag{1.7}$$

with the same dimension of training data D, but is not known before we finish the training process. That is to say, we can use this test data T to verify the performance of trained model f.

1.3 Some popular data sets in image classification

In this subsection, we will introduce some popular and standard data sets in image classification.

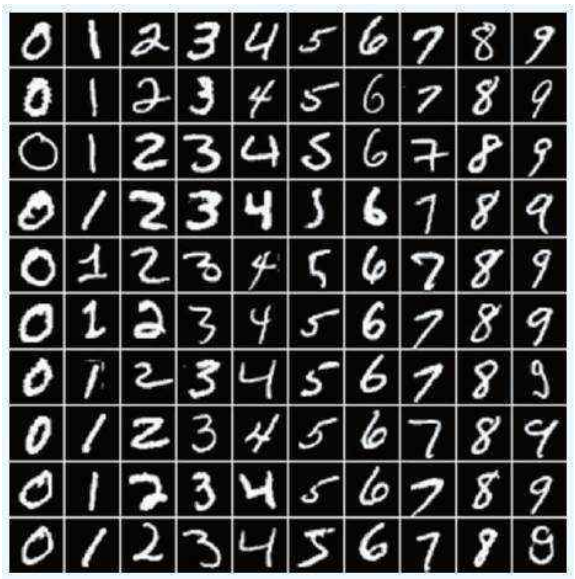
dataset	training (N)	test (M)	classes (k)	channels (c)	input size (d)
MNIST	60K	10K	10	Greyscale	28*28
CIFAR-10	50K	10K	10	RGB	32*32
CIFAR-100	50K	10K	100	RGB	32*32
ImageNet	1.2M	50K	1000	RGB	224*224

Table 1.1. Basic descriptions about popular datasets

1.3.1 MNIST(Modified National Institute of Standards and Technology Database)

This is a database for handwritten digits

- Training set : $N = 60,000$
- Test set : $M = 10,000$
- Image size : $d = 28 * 28 * 1 = 784$
- Classes: $k = 10$



$$x = \text{  } = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_{784} \end{pmatrix} \in \mathbb{R}^{784}$$

(1.8) $$A_1 = \left\{ \text{  }, \cdots \right\} \subset \mathbb{R}^{784}$$

(1.9) $$A_2 = \left\{ \text{  }, \cdots \right\} \subset \mathbb{R}^{784}$$

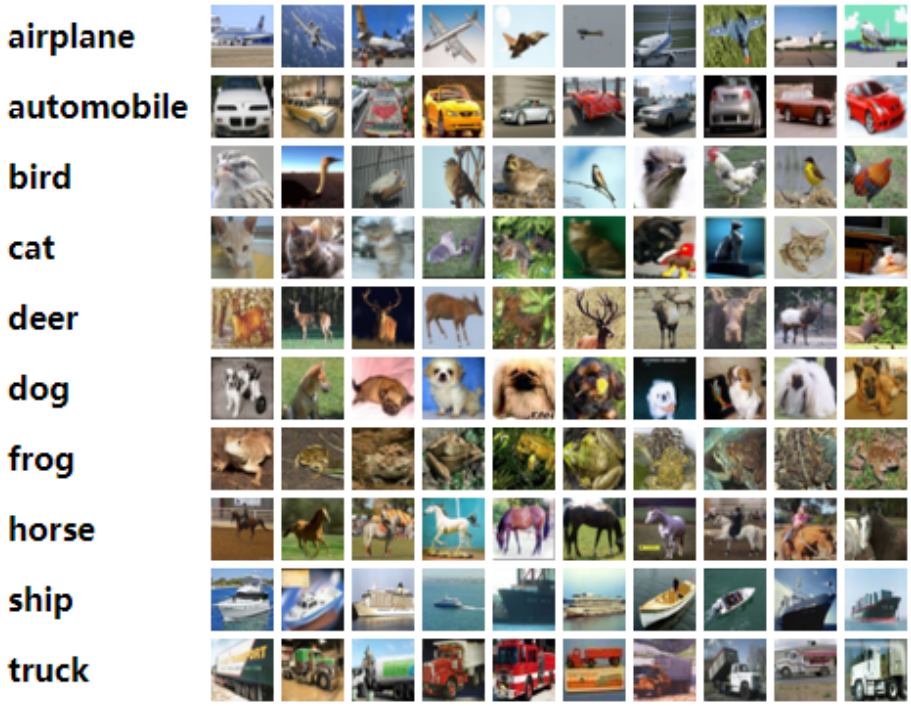
(1.10) $$A_9 = \left\{ \text{  }, \cdots \right\} \subset \mathbb{R}^{784}$$

(1.11) $$A_{10} = \left\{ \text{  }, \cdots \right\} \subset \mathbb{R}^{784}$$

1.3.2 CIFAR

CIFAR-10

- Training set : $N = 50,000$
- Test set : $M = 10,000$
- Image size : $d = 32 * 32 * 3$
- Classes: $k = 10$

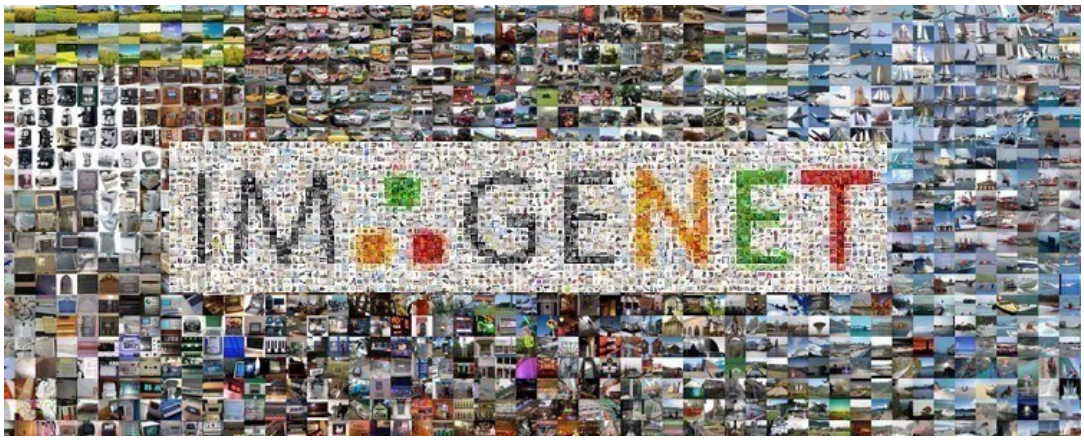


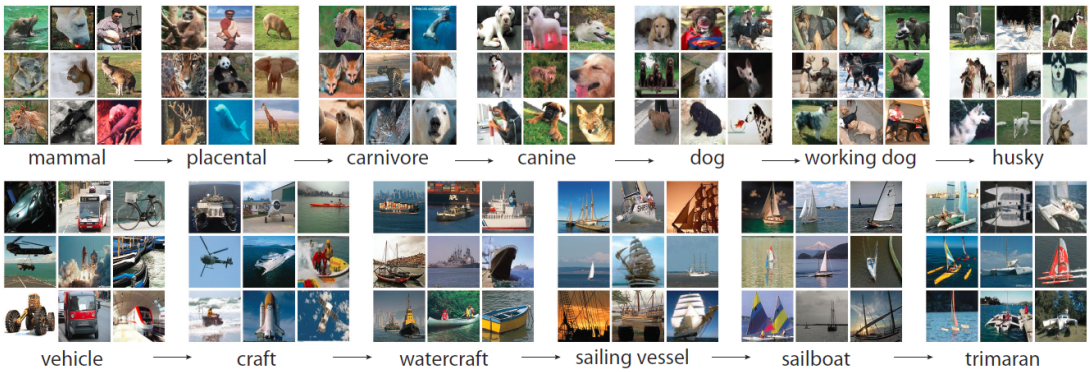
CIFAR-100

- Training set : $N = 50,000$
- Test set : $M = 10,000$
- Image size : $d = 32 * 32 * 3$
- Classes: $k = 100$

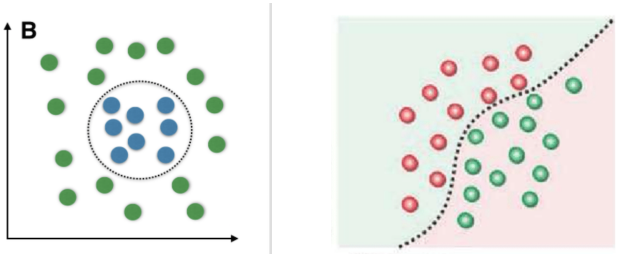
1.3.3 ImageNet

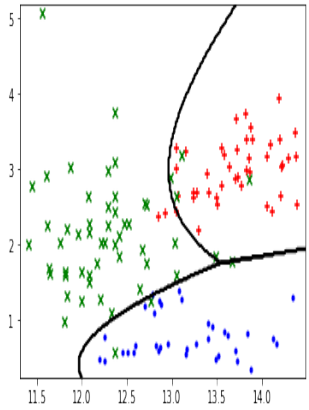
- All data set: $N + M = 1,431,167$
- Image size : $d = 224 * 224 * 3$
- Classes: $k = 1,000$





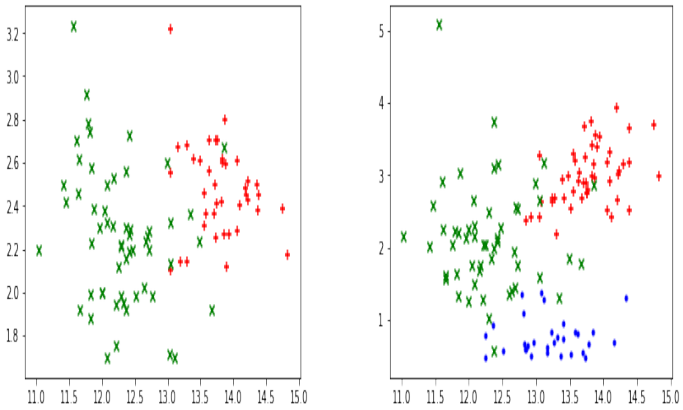
1.4 Classification and decision boundaries



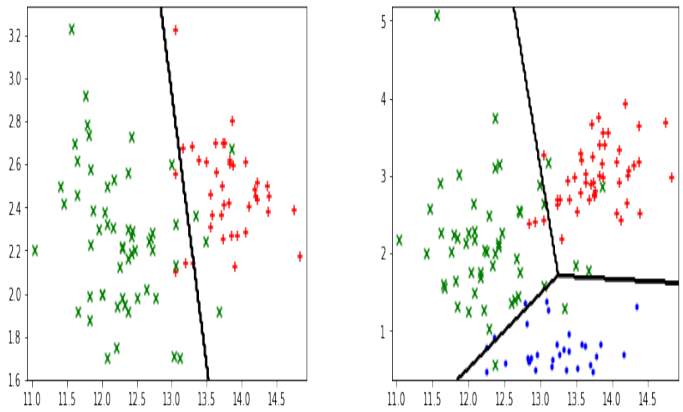


1.4.1 Linear models: decision boundaries given by hyper-planes

This is a demo for logistic regression classification for binary and multi-classed cases. The original data sets would be



Then the decision boundary for logistic regression models would be:



1.4.2 How to find these hyperplanes: logistic regressions

For a collection of subsets $A_1, ..., A_k \subset \mathbb{R}^d$, try to find

$$W = \begin{pmatrix} w_1 \\ \vdots \\ w_k \end{pmatrix} \in \mathbb{R}^{k\times d}, b = \begin{pmatrix} b_1 \\ \vdots \\ b_k \end{pmatrix} \in \mathbb{R}^{k\times d}, \tag{1.12}$$

such that, for each $1 \leq i \leq k$ and $j \neq i$

$$(Wx + b)_i > (Wx + b)_j, \ \forall x \in A_i, \tag{1.13}$$

or

$$w_i x + b_i > w_j x + b_j, \ \forall x \in A_i. \tag{1.14}$$

More details of logistic regression will be discussed later.

References