Imperial College London

Image Classification II

Dr Wenjia Bai

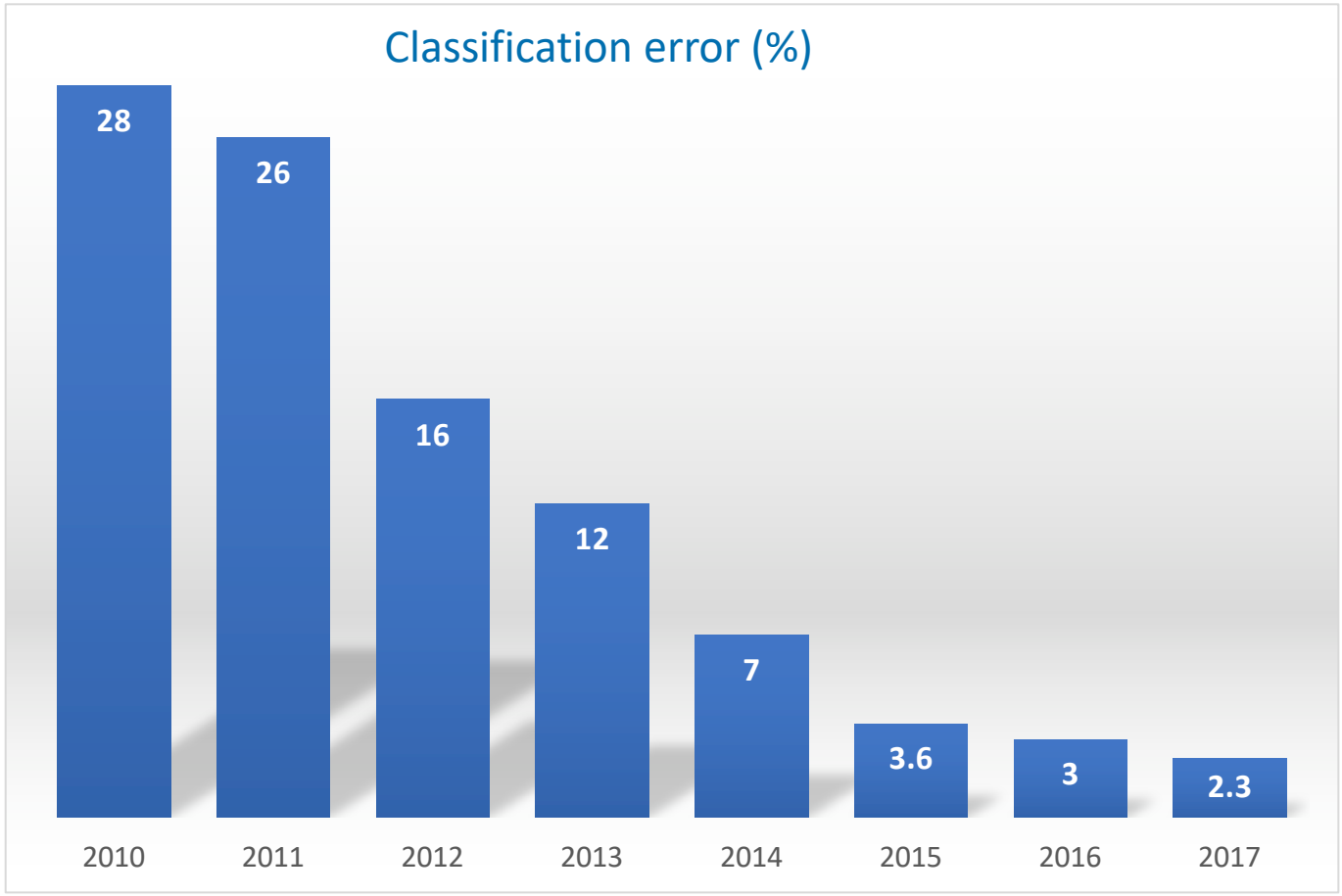
Department of Computing & Brain Sciences

Image classification

- Previously, we described K nearest neighbours (KNN) for image classification.
  - KNN is simple to implement but computationally expensive during test time.

- Today, we are going to introduce several other machine learning models for image classification.
  - Support vector machine (SVM)
  - Neural networks

ImageNet classification challenge



SVM Neural Networks

ILSVRC 2010

Codename	CLS	Insitutions	Contributors and references
Hminmax	54.4	Massachusetts Institute of Technology	Jim Mutch, Sharat Chikkerur, Hristo Paskov, Ruslan Salakhutdinov, Stan Bileschi, Hueihan Jhuang
IBM	70.1	IBM research†, Georgia Tech‡	Lexing Xie†, Hua Ouyang‡, Apostol Natsev†
ISIL	44.6	Intelligent Systems and Informatics Lab., The University of Tokyo	Tatsuya Harada, Hideki Nakayama, Yoshitaka Ushiku, Yuya Yamashita, Jun Imura, Yasuo Kuniyoshi
ITNLP	78.7	Harbin Institute of Technology	Deyuan Zhang, Wenfeng Xuan, Xiaolong Wang, Bingquan Liu, Chengjie Sun
LIG	60.7	Laboratoire d'Informatique de Grenoble	Georges Quénot
NEC	**28.2**	NEC Labs America†, University of Illinois at Urbana-Champaign‡, Rutgers∓	Yuanqing Lin†, Fengjun Lv†, Shenghuo Zhu†, Ming Yang†, Timothee Cour†, Kai Yu†, LiangLiang Cao‡, Zhen Li‡, Min-Hsuan Tsai‡, Xi Zhou‡, Thomas Huang‡, Tong Zhang∓ (Lin et al., 2011)
NII	74.2	National Institute of Informatics, Tokyo,Japan†, Hefei Normal Univ. Heifei, China‡	Cai-Zhi Zhu†, Xiao Zhou‡, Shiníchi Satoh†
NTU	58.3	CeMNet, SCE, NTU, Singapore	Zhengxiang Wang, Liang-Tien Chia
Regularities	75.1	SRI International	Omid Madani, Brian Burns
UCI	46.6	University of California Irvine	Hamed Pirsiavash, Deva Ramanan, Charless Fowlkes
XRCE	33.6	Xerox Research Centre Europe	Jorge Sanchez, Florent Perronnin, Thomas Mensink (Perronnin et al., 2010)

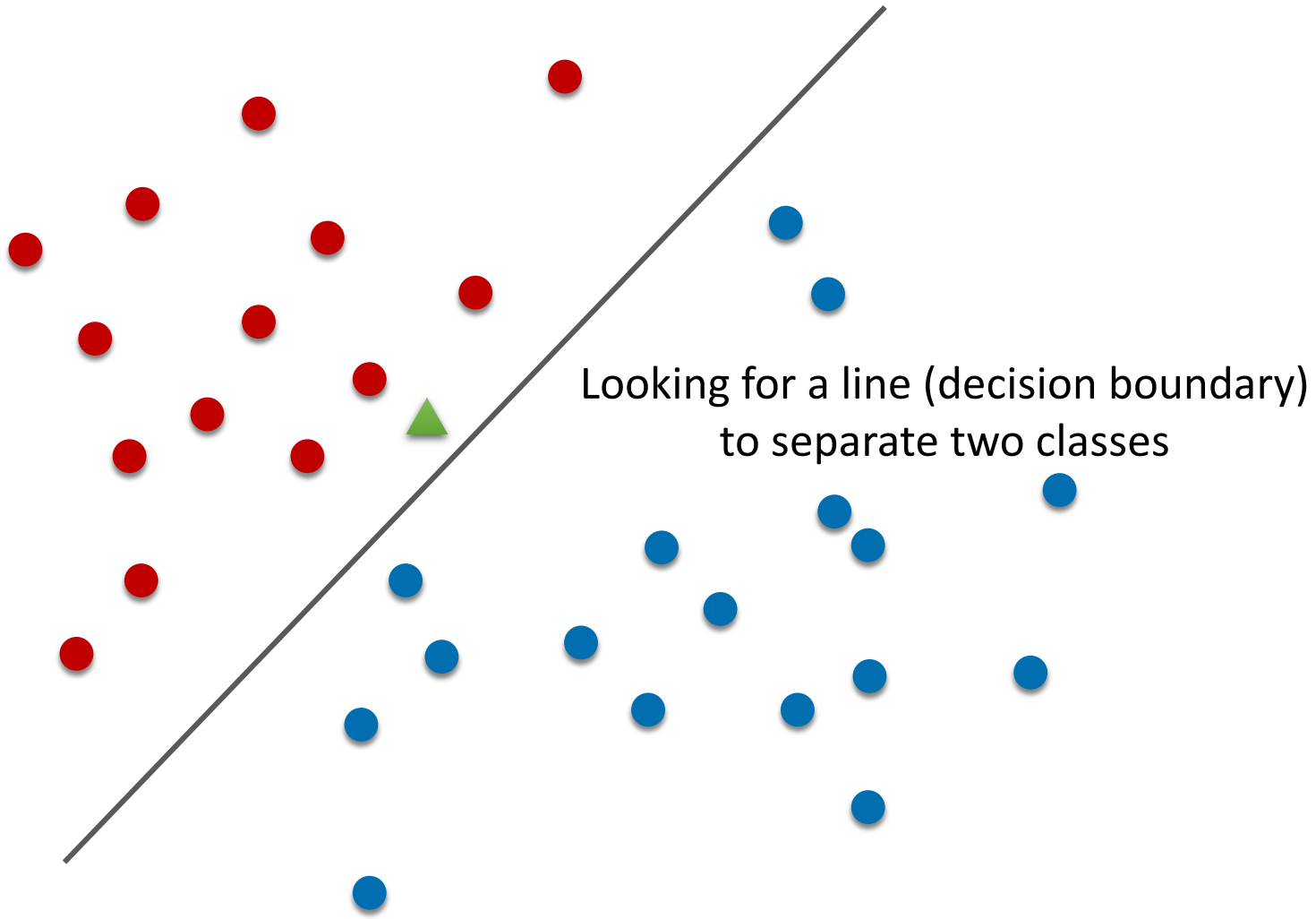
Image classification error rates in the first ImageNet challenge.
The top team and many other teams use SVM as the classifier and HOG as the feature.

Olga Russakovsky et al. ImageNet Large Scale Visual Recognition Challenge. IJCV, 2015.
Yuanqing Lin et al. Large-scale image classification: Fast feature extraction and SVM training. CVPR, 2011.

Support vector machine (SVM)

- What is SVM?
  - In its simplest form (linear SVM), it is simply a line that separates two different classes.

Linear classifier



Which class does the test data (triangle) belong to?

Linear classifier

- For 2D case (input data is 2D), the line has the form

$$w_1 x_1 + w_2 x_2 + b = 0$$

- The rule of the linear classifier is to assign a class c to data x,

$$c = \begin{cases} +1, \text{ if } w_1 x_1 + w_2 x_2 + b \geq 0 \\ -1, \qquad \text{otherwise} \end{cases}$$

- For KNN, we need to store all the training data.
- For linear classifier, once we know $\boldsymbol{w}$ and b, we can discard training data.

Linear classifier

- For general cases, the linear model is formulated as,

$$\boldsymbol{w} \cdot \boldsymbol{x} + b = 0$$

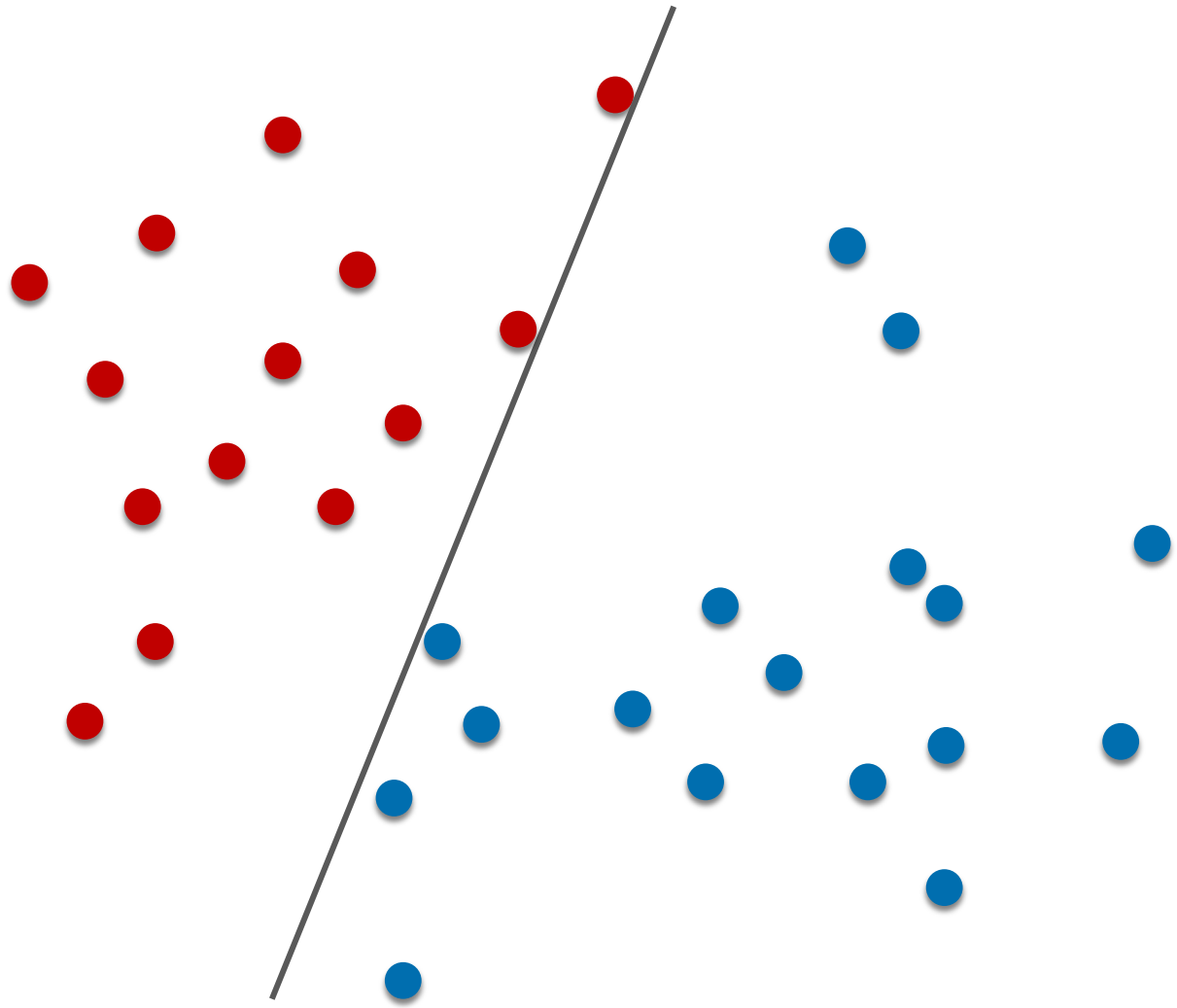
weights, the normal — data, feature — bias

- The rule of the linear classifier is to assign a class c to data x,

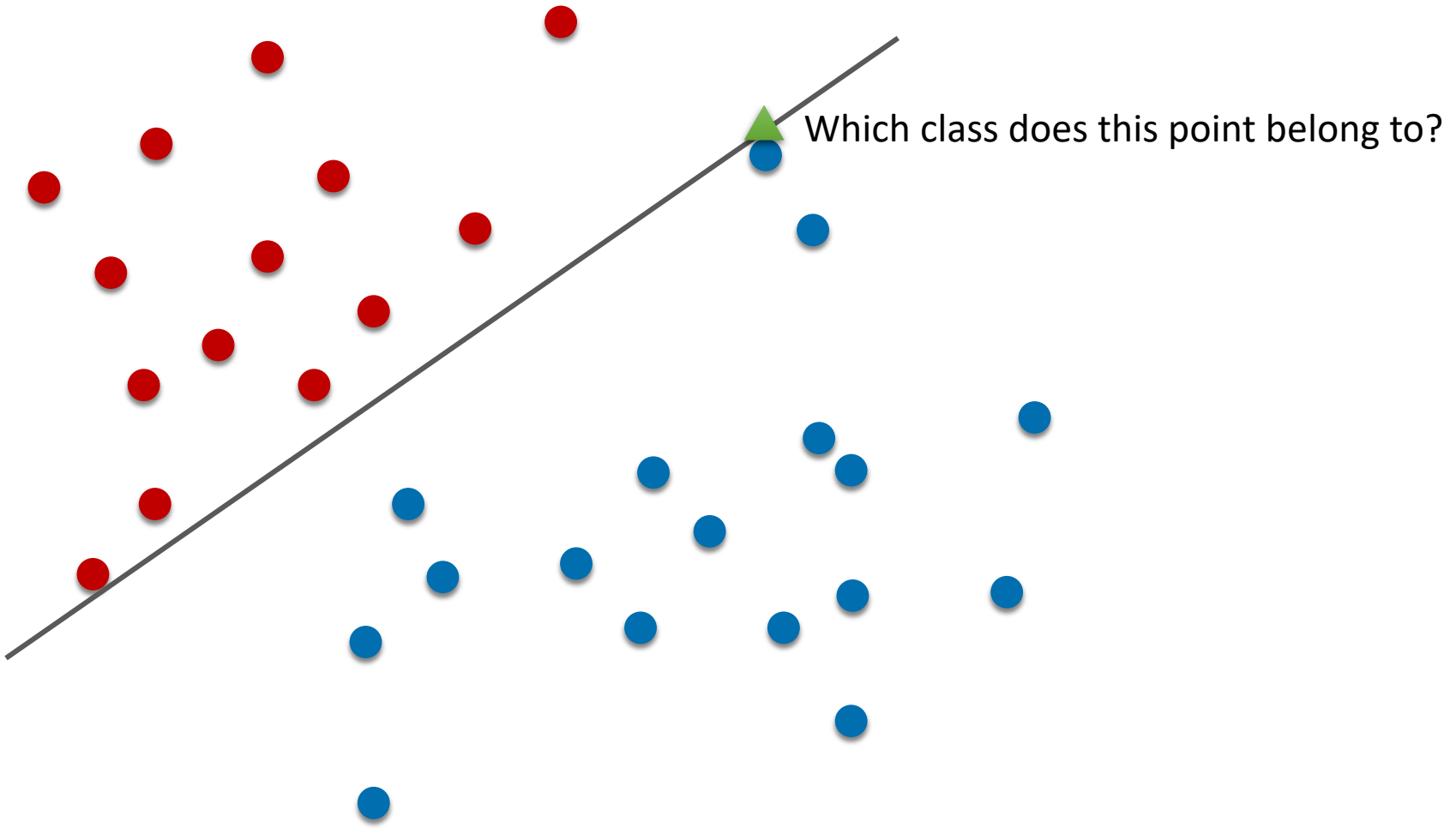
$$c = \begin{cases} +1, \text{ if } \boldsymbol{w} \cdot \boldsymbol{x} + b \geq 0 \\ -1, \quad \text{otherwise} \end{cases}$$

- To train the classifier, what we need is to estimate parameters $\boldsymbol{w}$ and b, which determines the decision boundary or hyperplane.

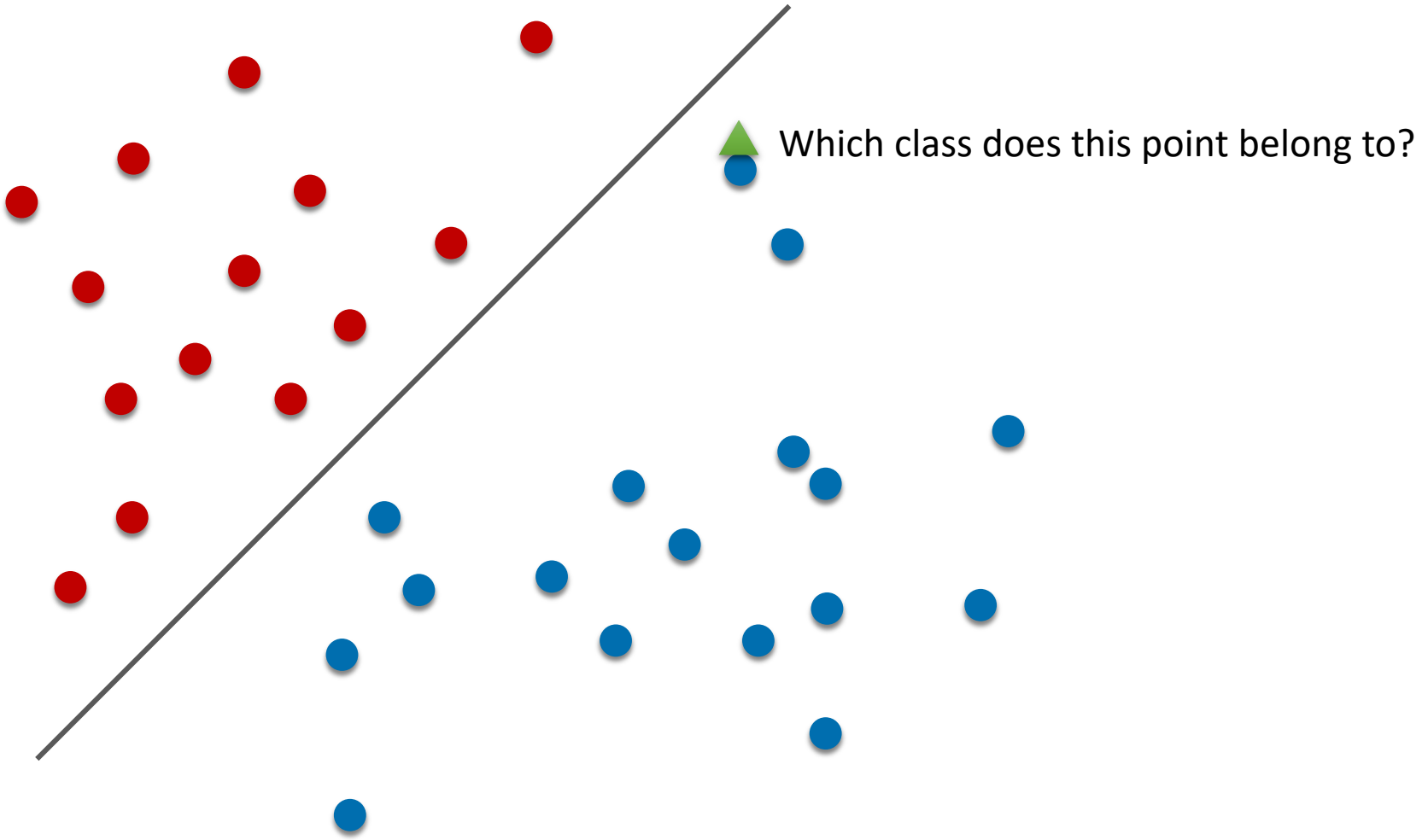
Where is best separating hyperplane?



Where is best separating hyperplane?

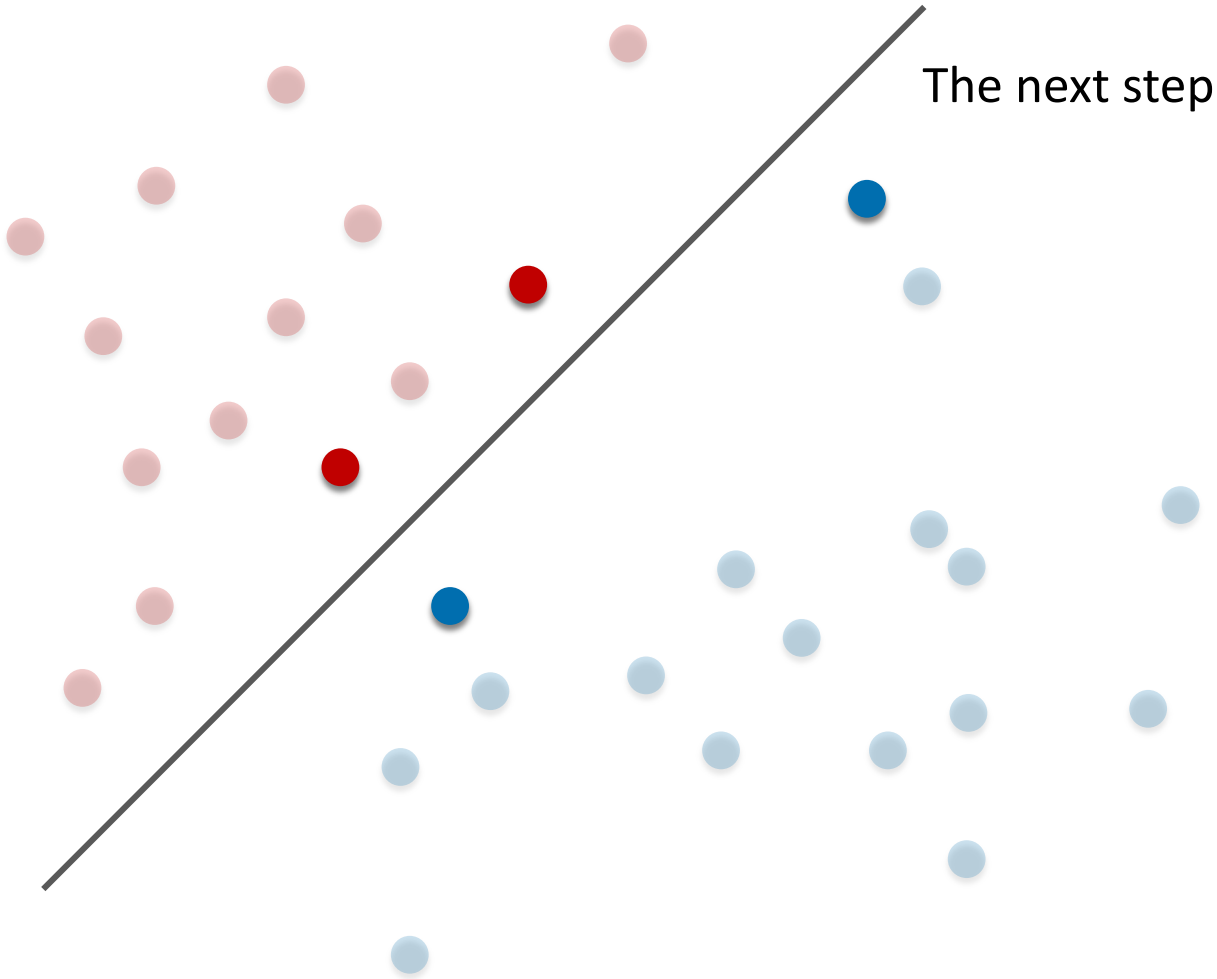


Where is best separating hyperplane?



Intuitively, a line that are far from both classes of points (**maximum margin** hyperplane).

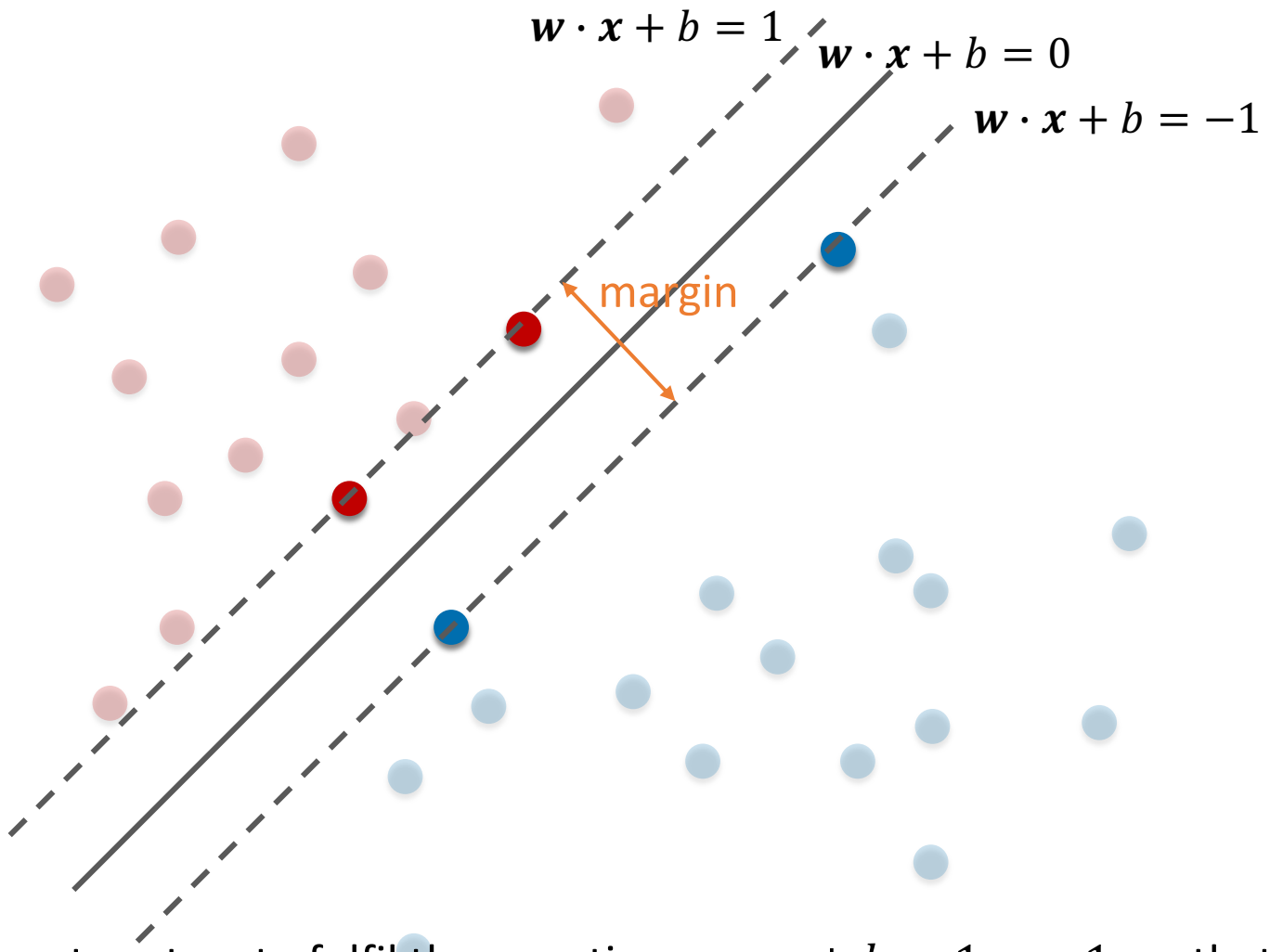
Where is best separating hyperplane?



The next step is to solve $\boldsymbol{w}$ and b.

To determine the maximum margin hyperplane, we only need the innermost points, which are called the **support vectors**.

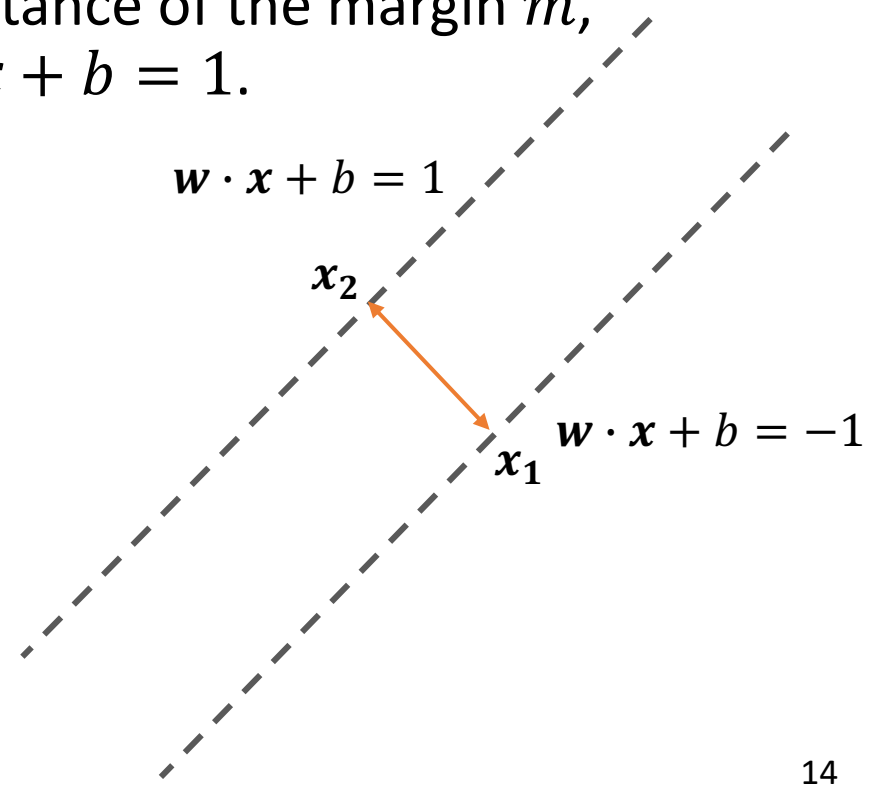
Where is best separating hyperplane?



- We would like the support vectors to fulfil the equations $\boldsymbol{w} \cdot \boldsymbol{x} + b = 1$ or -1, so that all the other points are easily classified.
- We would also like to maximise the margin between the dashed lines.

Maximum margin

- What is the margin between $\boldsymbol{w} \cdot \boldsymbol{x} + b = 1$ and $\boldsymbol{w} \cdot \boldsymbol{x} + b = -1$?
- We can derive it like this.
  - Let $\boldsymbol{x_1}$ be a point on the plane $\boldsymbol{w} \cdot \boldsymbol{x} + b = -1$, we have $\boldsymbol{w} \cdot \boldsymbol{x_1} + b = -1$.
  - Let us move $\boldsymbol{x_1}$ along the normal direction $\boldsymbol{n}$ for the distance of the margin m, it should arrive at the point $\boldsymbol{x_2}$ on the other plane $\boldsymbol{w} \cdot \boldsymbol{x} + b = 1$.
  - We have $\boldsymbol{w} \cdot (\boldsymbol{x_1} + m\boldsymbol{n}) + b = 1$.
  - It follows that $m\boldsymbol{w} \cdot \boldsymbol{n} = 2$.
  - We know that the normal $\boldsymbol{n} = \frac{\boldsymbol{w}}{\|\boldsymbol{w}\|}$.
  - As a result, the margin $m = \frac{2}{\|\boldsymbol{w}\|}$.



Support vector machine (SVM)

- SVM aims to maximise the margin,

$$\max_{\boldsymbol{w},b} \frac{2}{\|\boldsymbol{w}\|}$$

maximise the margin

$$\text{subject to } \boldsymbol{w} \cdot \boldsymbol{x_i} + b \begin{matrix} \geq +1, \text{if } y_i = +1 \\ \leq -1, \text{if } y_i = -1 \end{matrix}, \text{for } i = 1,2,\cdots,N.$$

separate the two classes

Optimisation

- It can be formulated as this optimisation problem

$$\min_{\boldsymbol{w},b} \|\boldsymbol{w}\|^2$$

$$\text{subject to } y_i(\boldsymbol{w} \cdot \boldsymbol{x_i} + b) \geq 1, \text{for } i = 1,2,\cdots,N.$$

- This is a quadratic optimisation problem subject to linear constraints.
  - It can be solved analytically using libraries such as libsvm.

Optimisation

- Alternatively, we can formulate the following optimisation problem,

$$\min_{\boldsymbol{w},b} \|\boldsymbol{w}\|^2 + C\sum_{i=1}^{N} \max(0, 1 - y_i\,(\boldsymbol{w}\cdot\boldsymbol{x_i} + b))$$

- The second term is called the hinge loss, $h = \max(0, 1 - y_i\,(\boldsymbol{w}\cdot\boldsymbol{x_i} + b))$.

- This function can be solved using gradient descent.

SVM for image classification

- Classifier

$$\boldsymbol{w} \cdot \boldsymbol{x} + b = 0$$

- What is the feature $\boldsymbol{x}$?
- How is SVM trained, i.e. how do we estimate $\boldsymbol{w}$ and b?

Large-scale Image Classification:
Fast Feature Extraction and SVM Training

Yuanqing Lin, Fengjun Lv, Shenghuo Zhu, Ming Yang, Timothee Cour and Kai Yu
NEC Laboratories America, Cupertino, CA 95014

Liangliang Cao and Thomas Huang
Beckman Institute, University of Illinois at Urbana-Champaign, IL 61801

Abstract

Most research efforts on image classification so far have been focused on medium-scale *datasets, which are often defined as datasets that can fit into the memory of a desktop (typically 4G~48G). There are two main reasons for the limited effort on* large-scale *image classification. First, until the emergence of ImageNet dataset, there was almost no publicly available large-scale benchmark data for image classification. This is mostly because class labels are expensive to obtain. Second, large-scale classification is hard because it poses more challenges than its medium-scale counterparts. A key challenge is how to achieve efficiency in both feature extraction and classifier training without compromising performance. This paper is to show how we address this challenge using ImageNet dataset as an example. For feature extraction, we develop a Hadoop scheme that performs feature extraction in parallel using hundreds of mappers. This allows us to extract fairly sophisticated features (with dimensions being hundreds of thousands) on 1.2 million images* within one day. *For SVM training, we develop a parallel* averaging stochastic gradient descent *(ASGD) algorithm for training one-against-all 1000-class SVM classifiers. The ASGD algorithm is capable of dealing with terabytes of training data and converges very fast – typically 5 epochs are sufficient. As a result, we achieve state-of-the-art performance on the ImageNet 1000-class classification,* i.e., 52.9% *in classification accuracy and* 71.8% *in top 5 hit rate.*

1. Introduction

It is needless to say how important of image classification/recognition is in the field of computer vision – image recognition is essential for bridging the huge semantic gap between an image, which is simply a scatter of pixels to untrained computers, and the object it presents. Therefore, there have been extensive research efforts on developing effective visual object recognizers [10]. Along the line, there are quite a few benchmark datasets for image classification, such as MNIST [1], Caltech 101 [9], Caltech 256 [11], PASCAL VOC [7], LabelMe[19], etc. Researchers have developed a wide spectrum of different local descriptors [17, 16, 5, 22], bag-of-words models [14, 24] and classification methods [4], and they compared to the best available results on those publicly available datasets – for PASCAL VOC, many teams from all over the world participate in the PASCAL Challenge each year to compete for the best performance. Such benchmarking activities have played an important role in pushing object classification research forward in the past years.

In recent years, there is a growing consensus that it is necessary to build general purpose object recognizers that are able to recognize many different classes of objects – e.g. this can be very useful for image/video tagging and retrieval. Caltech 101/256 are the pioneer benchmark datasets on that front. Newly released ImageNet dataset [6] goes a big step further, as shown in Fig. 1 – it further increases the number of classes to 1000[1], and it has more than 1000 images for each class on average. Indeed, it is necessary to have so many images for each class to cover visual variance, such as lighting, orientation as well as fairly wild appearance difference within the same class – like different cars may look very differently although all belong to the same class.

However, compared to those previous medium-scale datasets (such as PASCAL VOC datasets and Caltech101&256, which can fit into desktop memory), large-scale ImageNet dataset poses more challenges in image classification. For example, those previous datesets

[1]The overall ImageNet dataset consists of 11,231,732 labeled images of 15589 classes by October 2010. But here we only concern about the subset of ImageNet dataset (about 1.2 million images) that was used in 2010 ImageNet Large Scale Visual Recognition Challenge

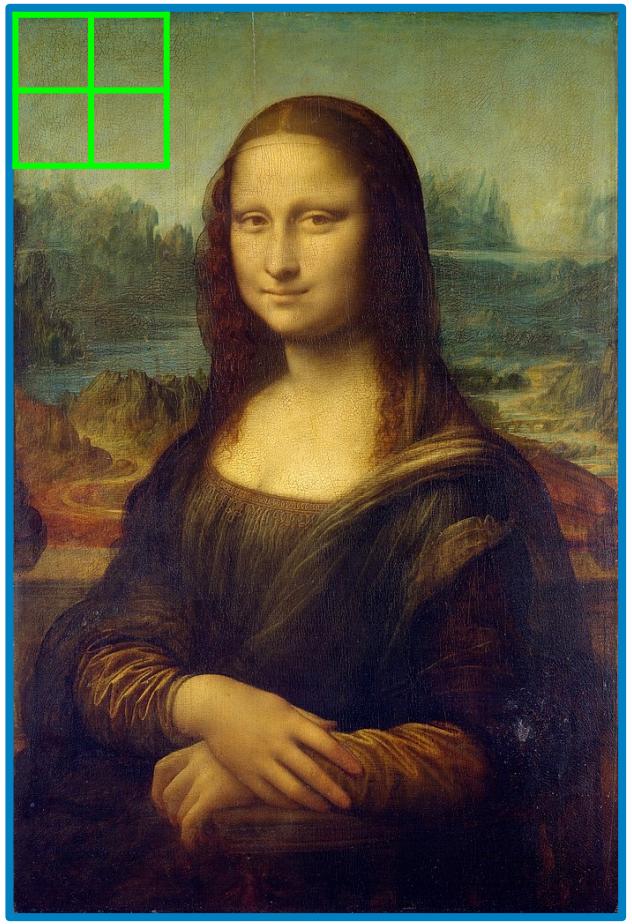
1689

Yuanqing Lin et al. Large-scale image classification: Fast feature extraction and SVM training. CVPR, 2011.

HOG

- Suppose we are to describe the feature for Mona-Lisa.
- We can divide the image into equally spaced cells.
  - Each cell contains 8 x 8 pixels.
  - 4 cells form a block.
- The orientation histograms of this block describes the top-left corner of this image.

Calculate gradient orientation histograms for this block.



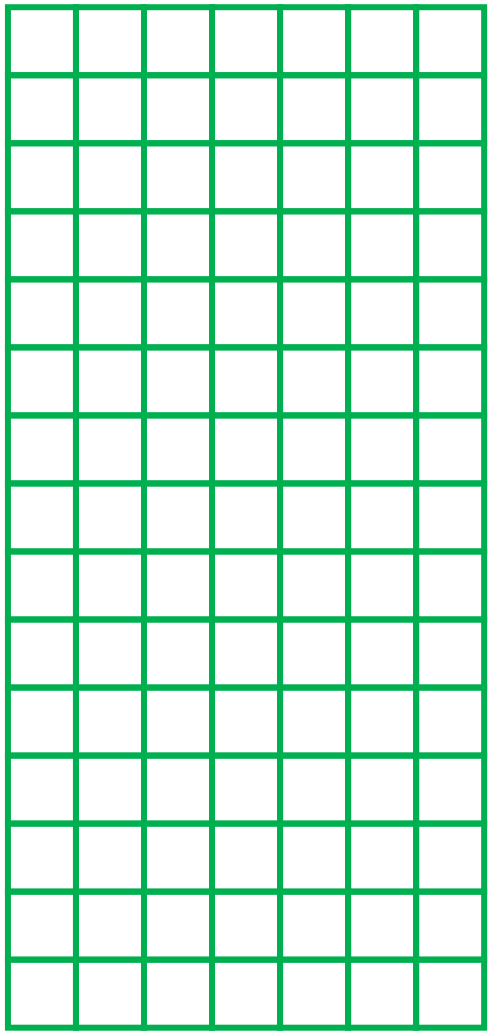
128x64 pixel image.

N. Dalal et al. Histograms of Oriented Gradients for Human Detection, CVPR 2005.

HOG

- The HOG descriptor is formed by concatenating the normalised local descriptors for all blocks.
- In this way, we can describe a full image or a large image region.
- We can use the HOG descriptor as $\boldsymbol{x}$.

The descriptors for all blocks in this image.



N. Dalal et al. Histograms of Oriented Gradients for Human Detection, CVPR 2005.

Large-scale image classification

- Since ImageNet contains 1.2 million images, how do we calculate the features for all the images?
  - Parallel computing
  - Distribute the computation to many machines e.g. using Hadoop

Gradient descent

- How is SVM trained?

over a million!

$$\min_{\boldsymbol{w},b} L(\boldsymbol{w},b) = \|\boldsymbol{w}\|^2 + C\sum_{i=1}^{N} \max(0, 1 - y_i(\boldsymbol{w}\cdot\boldsymbol{x_i} + b))$$

- The gradient of the loss function L w.r.t. $\boldsymbol{w}$ is,

$$\nabla_{\boldsymbol{w}} L = 2\boldsymbol{w} - C\sum_{i=1}^{N} \nabla_{\boldsymbol{w}} h$$

where $h = \max(0, 1 - y_i(\boldsymbol{w}\cdot\boldsymbol{x_i} + b))$. Its gradient is,

$$\nabla_{\boldsymbol{w}} h = \begin{cases} -y_i\boldsymbol{x_i}, \text{ if } y_i(\boldsymbol{w}\cdot\boldsymbol{x_i} + b) < 1 \\ 0, \qquad \text{otherwise} \end{cases}$$

- Similarly, we can derive $\nabla_{\boldsymbol{b}} L$.

Gradient descent

- We can optimise the loss function using gradient descent.
- For each iteration

$$\boldsymbol{w}^{(k+1)} = \boldsymbol{w}^{(k)} - \eta \nabla_{\boldsymbol{w}} L\left(\boldsymbol{w}^{(k)}, b^{(k)}\right)$$
$$= \boldsymbol{w}^{(k)} - \eta (2\boldsymbol{w}^{(k)} + C \sum_{i=1}^{N} \nabla_{\boldsymbol{w}} h)$$

step length, learning rate

Stochastic gradient descent (SGD)

- Since N is very big (a million), it would be computationally expensive to evaluate the gradient for the whole training set.
- Instead, we can perform stochastic gradient descent (SGD).

- For each iteration in optimisation
  - Randomly select a batch of B training samples
  - Calculate the gradient $\nabla_{\boldsymbol{w}} L_B$ and $\nabla_b L_B$ only for this batch
  - Update the parameters

$$\boldsymbol{w}^{(k+1)} = \boldsymbol{w}^{(k)} - \eta \nabla_{\boldsymbol{w}} L_B\left(\boldsymbol{w}^{(k)}, b^{(k)}\right)$$
$$b^{(k+1)} = b^{(k)} - \eta \nabla_b L_B\left(\boldsymbol{w}^{(k)}, b^{(k)}\right)$$

L_B denotes the loss just for this batch

SVM for image classification

- Classifier

$$\boldsymbol{w} \cdot \boldsymbol{x} + b = 0$$

- So now we know what the feature $\boldsymbol{x}$ is and how $\boldsymbol{w}$ and b are estimated.
- At test time, we perform classification,

$$c = \begin{cases} +1, \text{ if } \boldsymbol{w} \cdot \boldsymbol{x} + b \geq 0 \\ -1, \quad \text{otherwise} \end{cases}$$

- Next question: this only separates two classes.

Large-scale Image Classification:
Fast Feature Extraction and SVM Training

Yuanqing Lin, Fengjun Lv, Shenghuo Zhu, Ming Yang, Timothee Cour and Kai Yu
NEC Laboratories America, Cupertino, CA 95014

Liangliang Cao and Thomas Huang
Beckman Institute, University of Illinois at Urbana-Champaign, IL 61801

Abstract

Most research efforts on image classification so far have been focused on medium-scale *datasets, which are often defined as datasets that can fit into the memory of a desktop (typically 4G~48G). There are two main reasons for the limited effort on* large-scale *image classification. First, until the emergence of ImageNet dataset, there was almost no publicly available large-scale benchmark data for image classification. This is mostly because class labels are expensive to obtain. Second, large-scale classification is hard because it poses more challenges than its medium-scale counterparts. A key challenge is how to achieve efficiency in both feature extraction and classifier training without compromising performance. This paper is to show how we address this challenge using ImageNet dataset as an example. For feature extraction, we develop a Hadoop scheme that performs feature extraction in parallel using hundreds of mappers. This allows us to extract fairly sophisticated features (with dimensions being hundreds of thousands) on 1.2 million images* within one day. *For SVM training, we develop a parallel* averaging stochastic gradient descent *(ASGD) algorithm for training one-against-all 1000-class SVM classifiers. The ASGD algorithm is capable of dealing with terabytes of training data and converges very fast – typically 5 epochs are sufficient. As a result, we achieve state-of-the-art performance on the ImageNet 1000-class classification,* i.e., 52.9% *in classification accuracy and* 71.8% *in top 5 hit rate.*

1. Introduction

It is needless to say how important of image classification/recognition is in the field of computer vision – image recognition is essential for bridging the huge semantic gap between an image, which is simply a scatter of pixels to untrained computers, and the object it presents. Therefore, there have been extensive research efforts on developing effective visual object recognizers [10]. Along the line, there are quite a few benchmark datasets for image classification, such as MNIST [1], Caltech 101 [9], Caltech 256 [11], PASCAL VOC [7], LabelMe[19], etc. Researchers have developed a wide spectrum of different local descriptors [17, 16, 5, 22], bag-of-words models [14, 24] and classification methods [4], and they compared to the best available results on those publicly available datasets – for PASCAL VOC, many teams from all over the world participate in the PASCAL Challenge each year to compete for the best performance. Such benchmarking activities have played an important role in pushing object classification research forward in the past years.

In recent years, there is a growing consensus that it is necessary to build general purpose object recognizers that are able to recognize many different classes of objects – e.g. this can be very useful for image/video tagging and retrieval. Caltech 101/256 are the pioneer benchmark datasets on that front. Newly released ImageNet dataset [6] goes a big step further, as shown in Fig. 1 – it further increases the number of classes to 1000[1], and it has more than 1000 images for each class on average. Indeed, it is necessary to have so many images for each class to cover visual variance, such as lighting, orientation as well as fairly wild appearance difference within the same class – like different cars may look very differently although all belong to the same class.

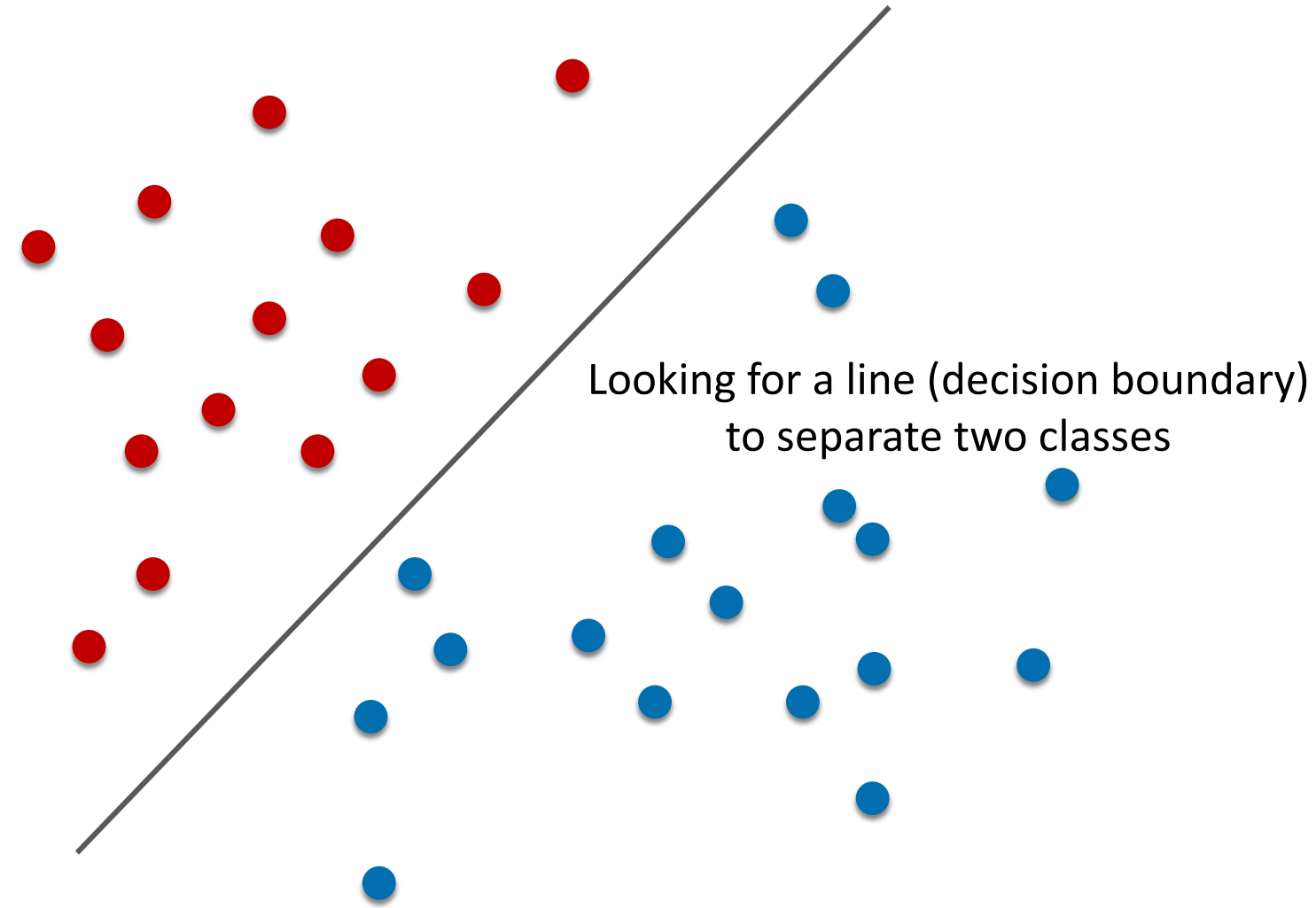
However, compared to those previous medium-scale datasets (such as PASCAL VOC datasets and Caltech101&256, which can fit into desktop memory), large-scale ImageNet dataset poses more challenges in image classification. For example, those previous datesets

[1]The overall ImageNet dataset consists of 11,231,732 labeled images of 15589 classes by October 2010. But here we only concern about the subset of ImageNet dataset (about 1.2 million images) that was used in 2010 ImageNet Large Scale Visual Recognition Challenge

1689

Yuanqing Lin et al. Large-scale image classification: Fast feature extraction and SVM training. CVPR, 2011.

Linear classifier



ImageNet

- There are 1,000 classes in the ImageNet classification challenge.

http://www.image-net.org

Class ID	Category
0	tench
1	goldfish
2	great white shark
3	tiger shark
4	hammerhead
5	electric ray
6	stingray
7	cock
8	hen
9	ostrich
…	…

The 1,000 classes in the ImageNet challenge dataset.

https://s3.amazonaws.com/deep-learning-models/image-models/imagenet_class_index.json

How to perform multi-class classification?

- One vs rest strategy
  - Train a classifier for each of the 1000 classes.
    - A classifier between class 1 and others
    - A classifier between class 2 and others
    - A classifier between class 3 and others
    - …
  - During testing, apply all the 1000 classifiers to the test data.
  - The classifier which produces the highest response will determine the result.

$$c = \underset{k=1,2,\cdots,K}{\operatorname{argmax}} f_k(\boldsymbol{x})$$

where $f_k(x) = \boldsymbol{w}_k \cdot \boldsymbol{x} + b_k$ denotes the k-th classifier.

How to perform multi-class classification?

- One vs one strategy
  - Train a classifier between each pair of classes.
    - A classifier between class 1 and class 2
    - A classifier between class 1 and class 3
    - A classifier between class 1 and class 4
    - …
    - A classifier between class 2 and class 3
    - A classifier between class 2 and class 4
    - …
  - For K classes, we need $\frac{K(K-1)}{2}$ classifiers.
  - At test time, each classifier will vote for a class.
  - Count the vote for each class and perform majority voting.

SVM for image classification

- In this paper, 1,000 binary classifiers were trained.
- And again, it was implemented using parallel computing because the training of 1,000 classifiers were independent.

Large-scale Image Classification:
Fast Feature Extraction and SVM Training

Yuanqing Lin, Fengjun Lv, Shenghuo Zhu, Ming Yang, Timothee Cour and Kai Yu
NEC Laboratories America, Cupertino, CA 95014

Liangliang Cao and Thomas Huang
Beckman Institute, University of Illinois at Urbana-Champaign, IL 61801

Abstract

Most research efforts on image classification so far have been focused on medium-scale *datasets, which are often defined as datasets that can fit into the memory of a desktop (typically 4G~48G). There are two main reasons for the limited effort on* large-scale *image classification. First, until the emergence of ImageNet dataset, there was almost no publicly available large-scale benchmark data for image classification. This is mostly because class labels are expensive to obtain. Second, large-scale classification is hard because it poses more challenges than its medium-scale counterparts. A key challenge is how to achieve efficiency in both feature extraction and classifier training without compromising performance. This paper is to show how we address this challenge using ImageNet dataset as an example. For feature extraction, we develop a Hadoop scheme that performs feature extraction in parallel using hundreds of mappers. This allows us to extract fairly sophisticated features (with dimensions being hundreds of thousands) on 1.2 million images* within one day. *For SVM training, we develop a parallel* averaging stochastic gradient descent *(ASGD) algorithm for training one-against-all 1000-class SVM classifiers. The ASGD algorithm is capable of dealing with terabytes of training data and converges very fast – typically 5 epochs are sufficient. As a result, we achieve state-of-the-art performance on the ImageNet 1000-class classification,* i.e., 52.9% *in classification accuracy and* 71.8% *in top 5 hit rate.*

1. Introduction

It is needless to say how important of image classification/recognition is in the field of computer vision – image recognition is essential for bridging the huge semantic gap between an image, which is simply a scatter of pixels to untrained computers, and the object it presents. Therefore, there have been extensive research efforts on developing effective visual object recognizers [10]. Along the line, there are quite a few benchmark datasets for image classification, such as MNIST [1], Caltech 101 [9], Caltech 256 [11], PASCAL VOC [7], LabelMe[19], etc. Researchers have developed a wide spectrum of different local descriptors [17, 16, 5, 22], bag-of-words models [14, 24] and classification methods [4], and they compared to the best available results on those publicly available datasets – for PASCAL VOC, many teams from all over the world participate in the PASCAL Challenge each year to compete for the best performance. Such benchmarking activities have played an important role in pushing object classification research forward in the past years.

In recent years, there is a growing consensus that it is necessary to build general purpose object recognizers that are able to recognize many different classes of objects – e.g. this can be very useful for image/video tagging and retrieval. Caltech 101/256 are the pioneer benchmark datasets on that front. Newly released ImageNet dataset [6] goes a big step further, as shown in Fig. 1 – it further increases the number of classes to 1000[1], and it has more than 1000 images for each class on average. Indeed, it is necessary to have so many images for each class to cover visual variance, such as lighting, orientation as well as fairly wild appearance difference within the same class – like different cars may look very differently although all belong to the same class.

However, compared to those previous medium-scale datasets (such as PASCAL VOC datasets and Caltech101&256, which can fit into desktop memory), large-scale ImageNet dataset poses more challenges in image classification. For example, those previous datesets

[1]The overall ImageNet dataset consists of 11,231,732 labeled images of 15589 classes by October 2010. But here we only concern about the subset of ImageNet dataset (about 1.2 million images) that was used in 2010 ImageNet Large Scale Visual Recognition Challenge

1689

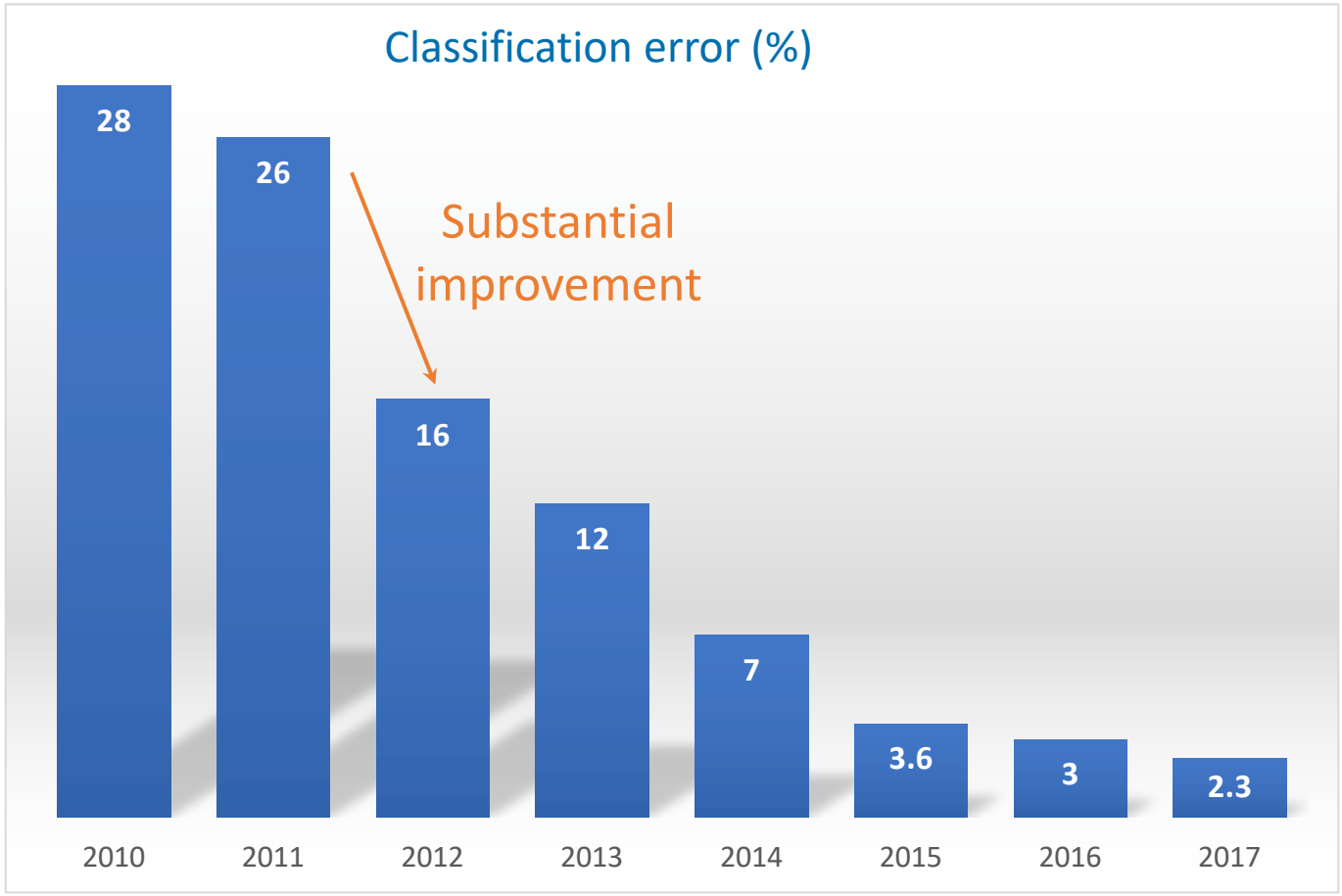
Yuanqing Lin et al. Large-scale image classification: Fast feature extraction and SVM training. CVPR, 2011.

Image classification

- SVM won the ImageNet challenge in 2010.
- Something happened afterwards.
  - In 2011, a method based on deep belief networks (DBN) increased the speech recognition accuracy by a large margin, published in ICASSP [1].
  - In 2012, neural networks won the ImageNet challenge [2].

[1] G.E. Dahl et al. Large vocabulary continuous speech recognition with context-dependent DBN-HMMs. ICASSP 2011.
[2] A. Krizhevsky et al. ImageNet classification with deep convolutional neural networks. NIPS 2012.

ImageNet classification challenge

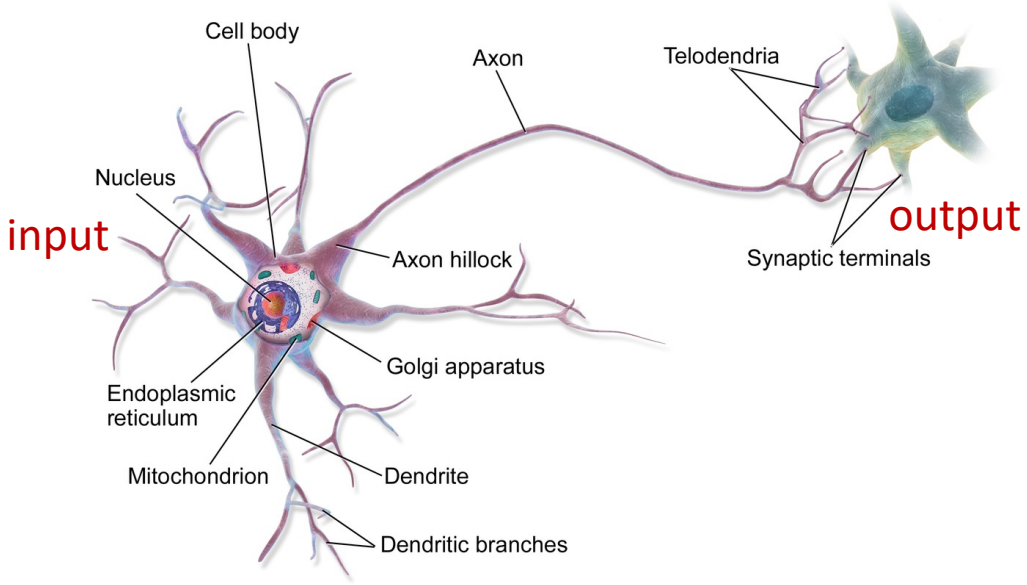


Neural networks

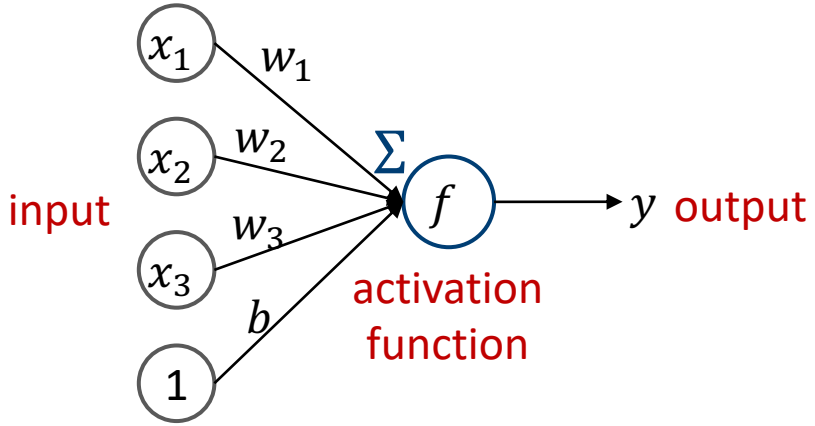
- Perceptron
- Multi-layer perceptron (MLP)
- Convolutional neural networks (CNN)

From biological neuron to artificial neuron

- Biology
  - Neurons are inter-connected.
  - Signal (electrical, chemical) flows from input at dendrites to output at axon terminals.



- Artificial neural networks
  - Inspired by biology.
  - But it is not the exact model of how brain or neuron works.

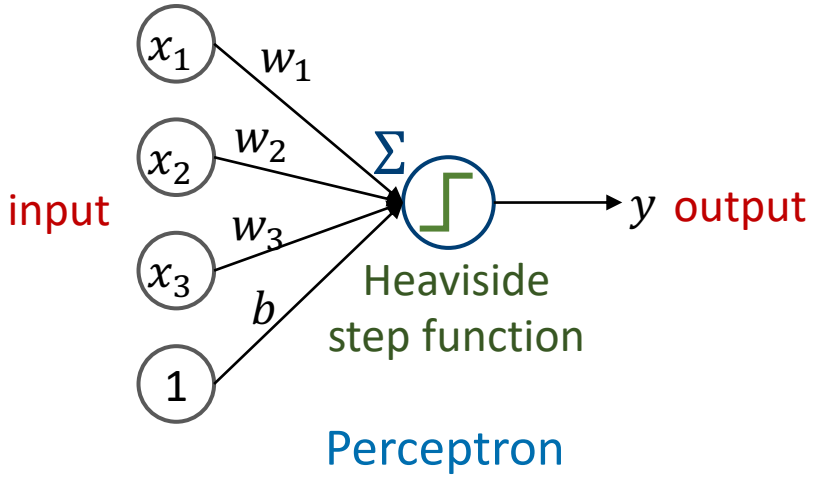


Perceptron

- The simplest form, perceptron, was developed by Frank Rosenblatt in 1957.
  - Perceptron consists of only a single layer and uses the Heaviside step function as activation function.

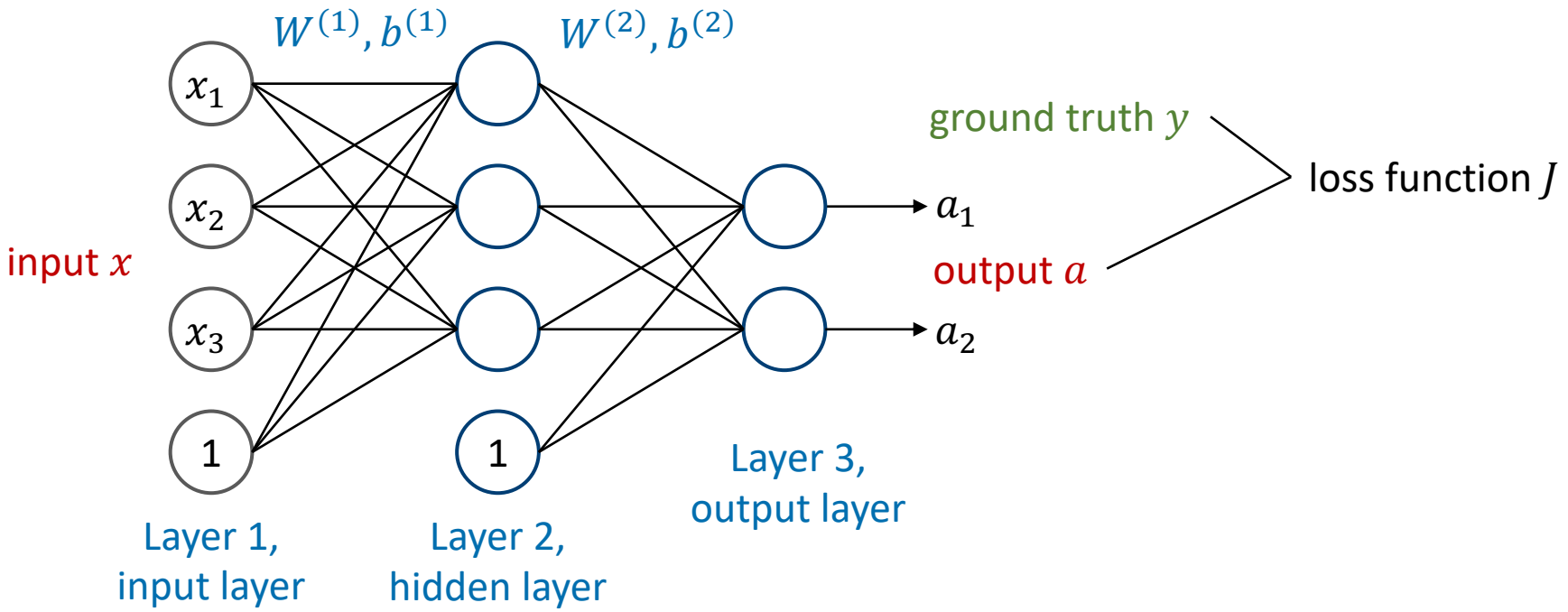
$$y = \begin{cases} 1, & \text{if } \boldsymbol{w} \cdot x + b > 0 \\ 0, & \text{otherwise} \end{cases}$$

  - We optimise $\boldsymbol{w}$ and b so that y matches ground truth.
- Multi-layer perceptron was developed later.



Multi-layer perceptron

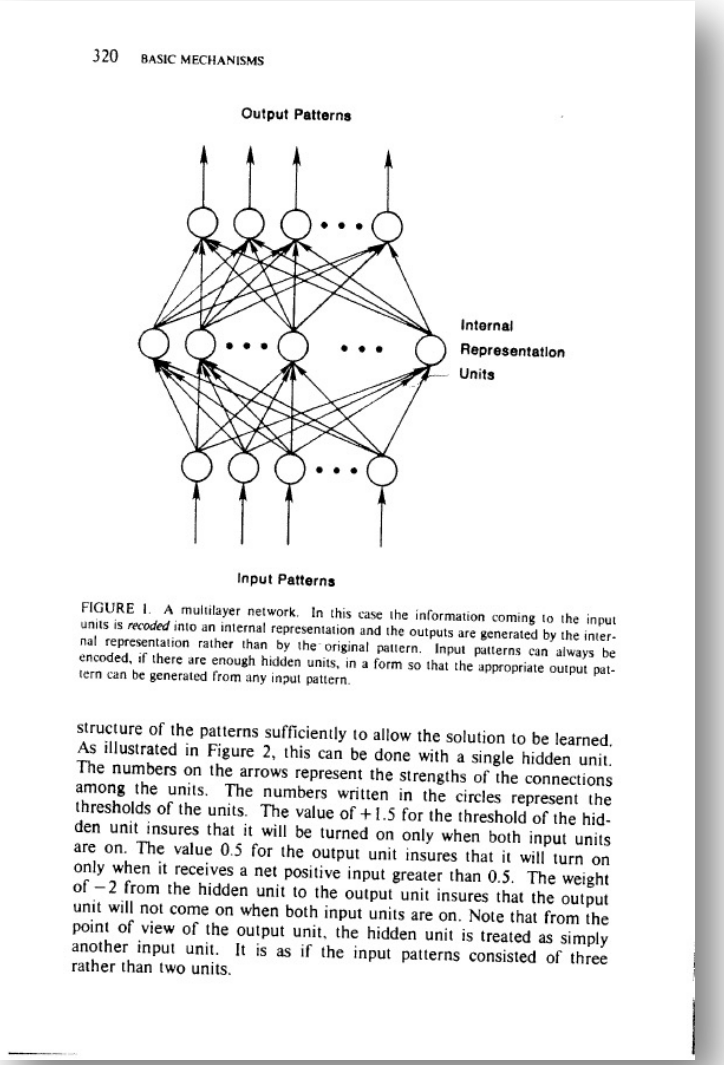
- A multi-layer perceptron (MLP) is formed by putting several layers of neurons into connection, where the output of a neuron can be the input to another.



Multi-layer perceptron (MLP), which is a fully connected multi-layer network.

Neural networks

- The form of neural networks has not changed much since 1980s.
- It optimisation algorithm, backpropagation, developed by David Rumelhart, Geoffery Hinton and Ronald J. Williams in 1986, has not changed either.



320 BASIC MECHANISMS

FIGURE 1. A multilayer network. In this case the information coming to the input units is *recoded* into an internal representation and the outputs are generated by the internal representation rather than by the original pattern. Input patterns can always be encoded, if there are enough hidden units, in a form so that the appropriate output pattern can be generated from any input pattern.

structure of the patterns sufficiently to allow the solution to be learned. As illustrated in Figure 2, this can be done with a single hidden unit. The numbers on the arrows represent the strengths of the connections among the units. The numbers written in the circles represent the thresholds of the units. The value of +1.5 for the threshold of the hidden unit insures that it will be turned on only when both input units are on. The value 0.5 for the output unit insures that it will turn on only when it receives a net positive input greater than 0.5. The weight of −2 from the hidden unit to the output unit insures that the output unit will not come on when both input units are on. Note that from the point of view of the output unit, the hidden unit is treated as simply another input unit. It is as if the input patterns consisted of three rather than two units.

D.E. Rumelhart et al. Learning internal representations by error-propagation, 1986.

Neural networks

- So what has changed in the past 40 years?
  - More layers of neural connections (i.e. deeper)
  - Better hardware to enable faster computation (i.e. GPUs)
  - Larger datasets (e.g. ImageNet)
  - Other technical improvements
    - Activation functions, optimisation, data normalisation, data augmentation etc
- Why was it not popular in 1990s and 2000s?
  - Because people were using SVM.
  - Also because many people did not believe deep neural networks would work.

Neural networks

Turing Award 2018

"For conceptual and engineering breakthroughs that have made **deep neural networks** a critical component of computing."

Contributions to neural networks, including RNN and LSTM.

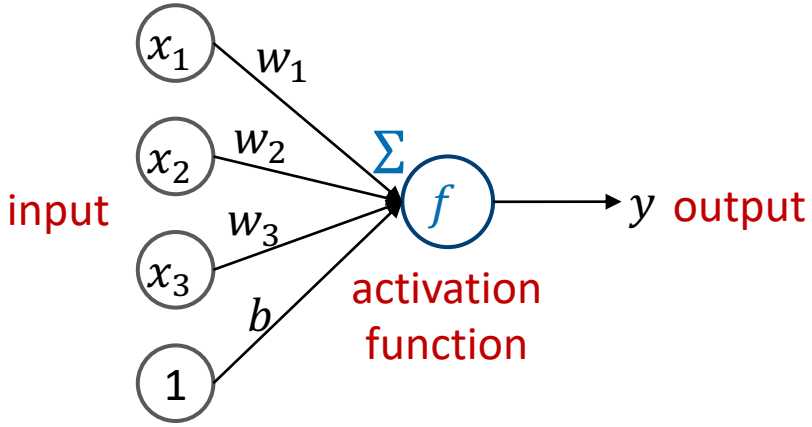
How does a neural network work?

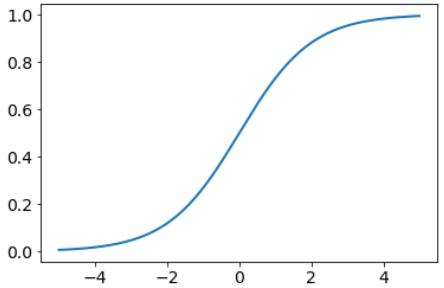
- Let us start from the simplest case, a single neuron.
- The neuron is a computational unit that takes an input, applies an activation function and generates an output.

$$y = f\left(\sum_{i=1}^{3} w_i x_i + b\right)$$

- A commonly used activation function is the sigmoid function, also known as the logistic function.

$$f(z) = \frac{1}{1 + e^{-z}}$$

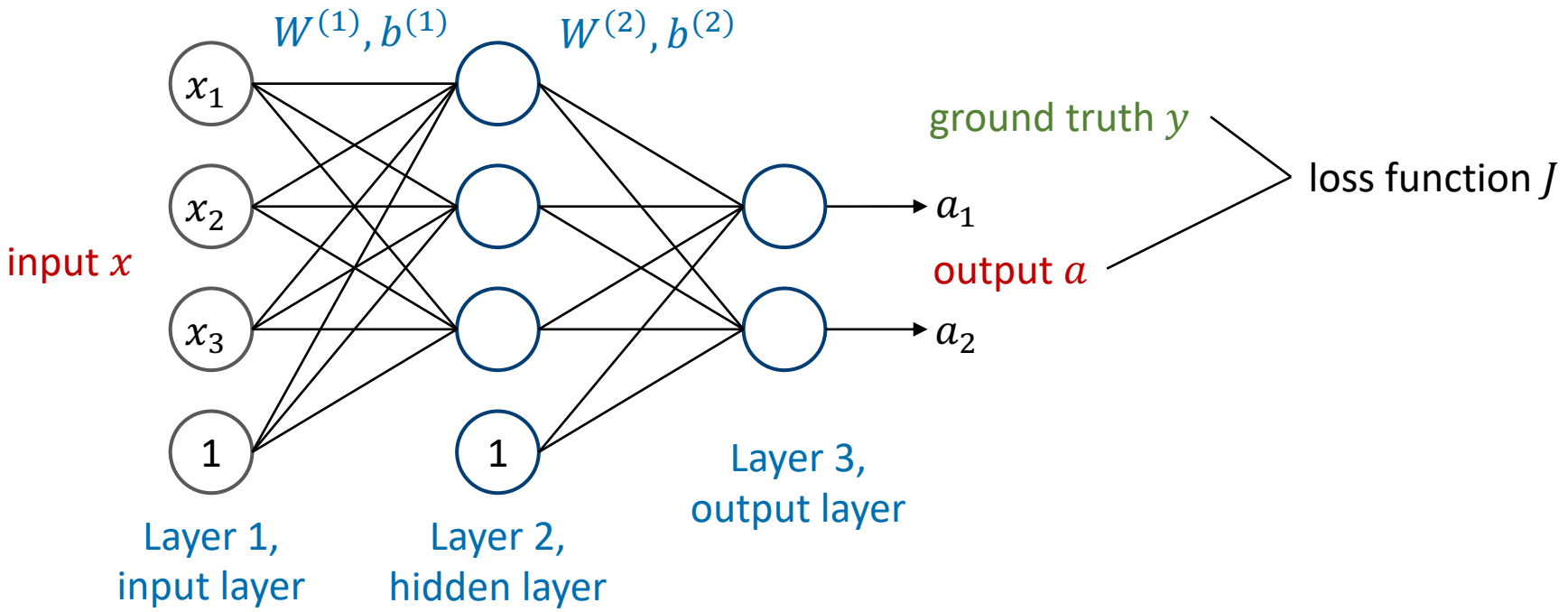




Sigmoid activation function

Multi-layer perceptron

- A multi-layer perceptron (MLP) is formed by putting several layers of neurons into connection, where the output of a neuron can be the input to another.



Multi-layer perceptron (MLP), which is a fully connected multi-layer network.

Training a neural network

- To train a neural network, we need a training set $\{(x_1, y_1), (x_2, y_2), \cdots, (x_m, y_m), \cdots, (x_M, y_M)\}$, which are paired data and ground truth labels.
- We would like to find parameters W and b so that given input x, the output of the network a matches y as much as possible.
- For example, we can define a loss function like this,

$$J(W, b) = \frac{1}{M} \sum_{m=1}^{M} \frac{1}{2} \| a_m - y_m \|^2$$

m-th sample

Training a neural network

- There are two technical questions here.
- Q1: How do we calculate the output of the network a given input x?
  - A1: We use **forward propagation**.

- Q2: How do we find parameters W and b that minimise the loss function, so that a matching ground truth y as much as possible?
  - A2: We use **backpropagation** to calculate the gradient and then perform stochastic gradient descent for optimisation.

- You can learn the details in machine learning or deep learning modules.

Neural networks for image classification

- Suppose we know how a neural network works.
- How do we use it for image classification?
  - We just need to define a loss function for image classification.
  - Then train the network, i.e. optimise the loss w.r.t. to network parameters.

Loss function

- Mean squared error

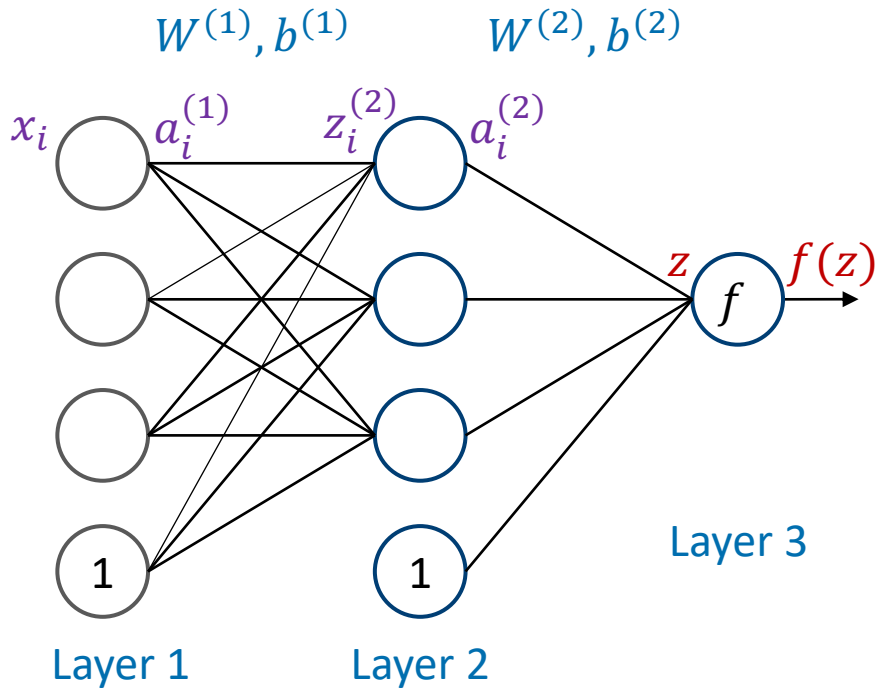
$$J(W,b) = \frac{1}{M}\sum_{m=1}^{M}\frac{1}{2}\|a_m - y_m\|^2$$

- This works for regression problems, i.e. y is a continuous variable.
- But is may not be optimal for image classification problems, where y is a categorical variable.
  - For binary classification, y can be 0 or 1.
  - For multi-class classification, for example,
    - in MNIST, y is one of the 10 label classes.
    - in ImageNet, y is one of the 1,000 label classes.

Binary classification

- Let us first consider the simplest case, binary classification.
- For binary classification, the output layer only needs 1 neuron.
- We can use the sigmoid activation function for the last layer. Its output is a value between 0 and 1, which is in the range of probability.

$$f(z) = \frac{1}{1 + e^{-z}}$$



Binary classification

- For example, the network predicts the probability to be 0.9 for class 1 and $1 - 0.9 = 0.1$ for class 0.
- Suppose the ground truth is $y = 1$, i.e. class 1. In other words, the probability is 1 for class 1 and 0 for class 0.

- Can we evaluate a distance metric between the predicted probability and the true probability, using this as the loss function?
  - Yes.
  - Information theory provides us with a metric for probability distributions, called the cross entropy.

Loss function

- The cross entropy between a true probability distribution p and an estimatd probability q is defined as,

$$H(p,q) = -\sum_{i} p_i \log(q_i)$$

number of classes

- In our example (two classes), $p = [1,0]$, $q = [0.9,0.1]$,

$$H(p,q) = -(1 * \log(0.9) + 0 * \log(0.1)) = 0.105$$

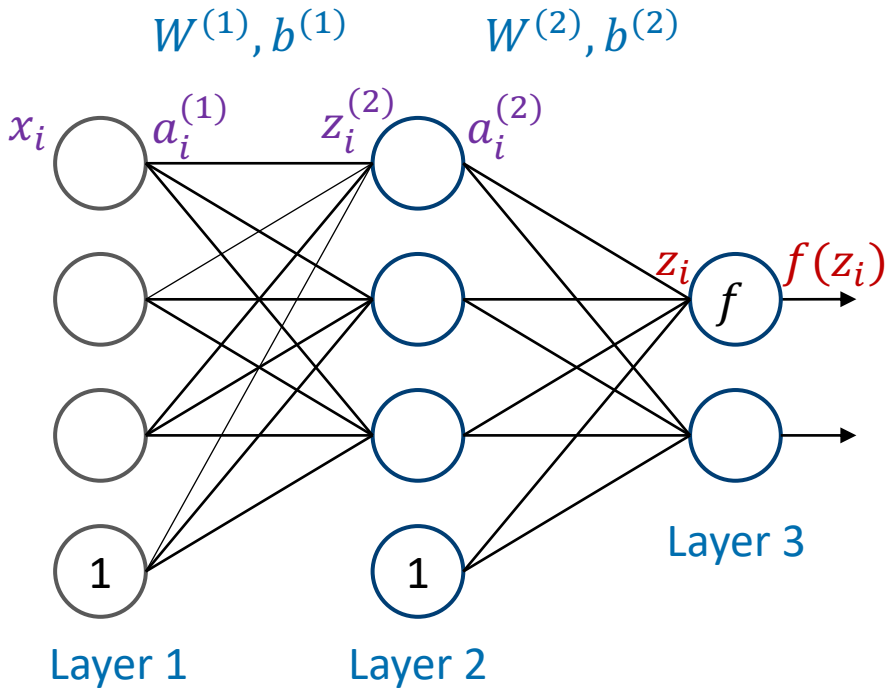
- For general cases, $p = [y, 1-y]$, $q = [f(z), 1-f(z)]$,

$$H(p,q) = -[y \log(f(z)) + (1-y) \log(1-f(z))]$$

Multi-class classification

- For K classes, we put K neurons at output layer.
- To make the output of the K neurons to form a probability vector, which have positive values and sum to 1, we use the softmax function.

$$f(z_i) = \frac{e^{z_i}}{\sum_{k=1}^{K} e^{z_k}}$$



Loss function

- The true probability $p = [y_1, \cdots, y_i, \cdots, y_K] = [0, \cdots, 1, \cdots, 0]$.
  - It is called one-hot encoding or representation.
  - Only one element is 1 (hot), all the others are 0 (cold).
- The estimated probability $q = [f(z_1), \cdots, f(z_i), \cdots, f(z_K)]$.

- The cross entropy loss is defined,

$$J(W, b; x, y) = H(p, q) = -\sum_{i=1}^{K} y_i \log(f(z_i))$$

Sigmoid vs softmax

- Sigmoid
  - Binary classification
  - The probabilities for two classes are

$$f(z) = \frac{1}{1 + e^{-z}} = \frac{e^z}{e^z + e^0}$$

$$1 - f(z) = \frac{e^0}{e^z + e^0}$$

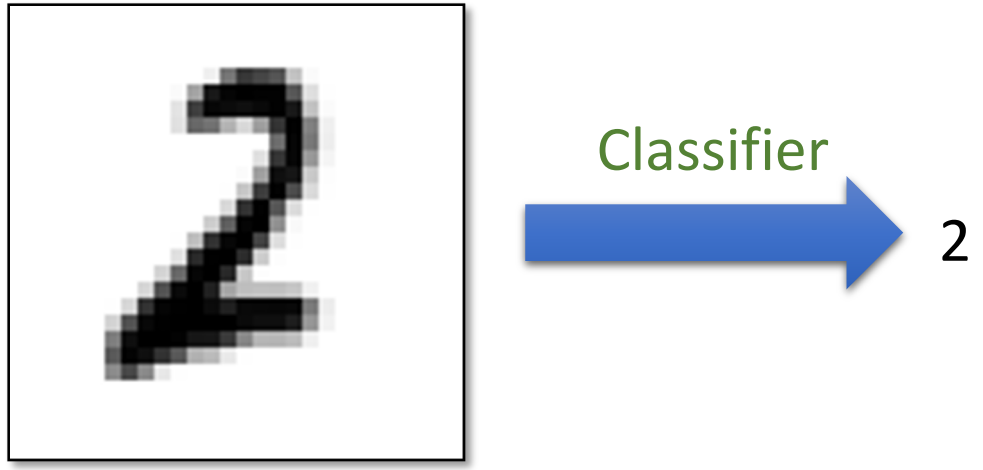
- Softmax
  - Multi-class classification
  - The probability for each of the K classes is

$$f(z_i) = \frac{e^{z_i}}{\sum_{k=1}^{K} e^{z_k}}$$

  - It generalises the sigmoid function to multiple classes.

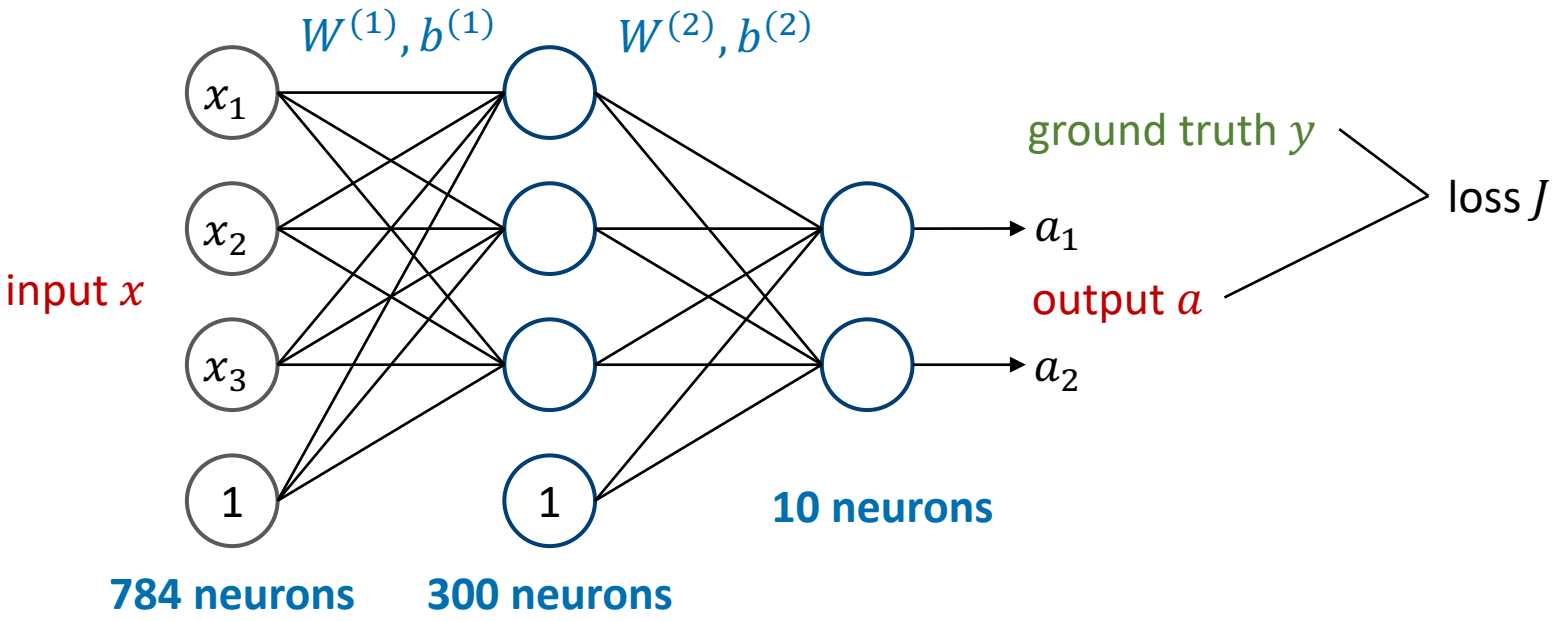
Neural networks for image classification

- Now we know how to define a loss function for multi-class classification.
- Let us try this on MNIST dataset.



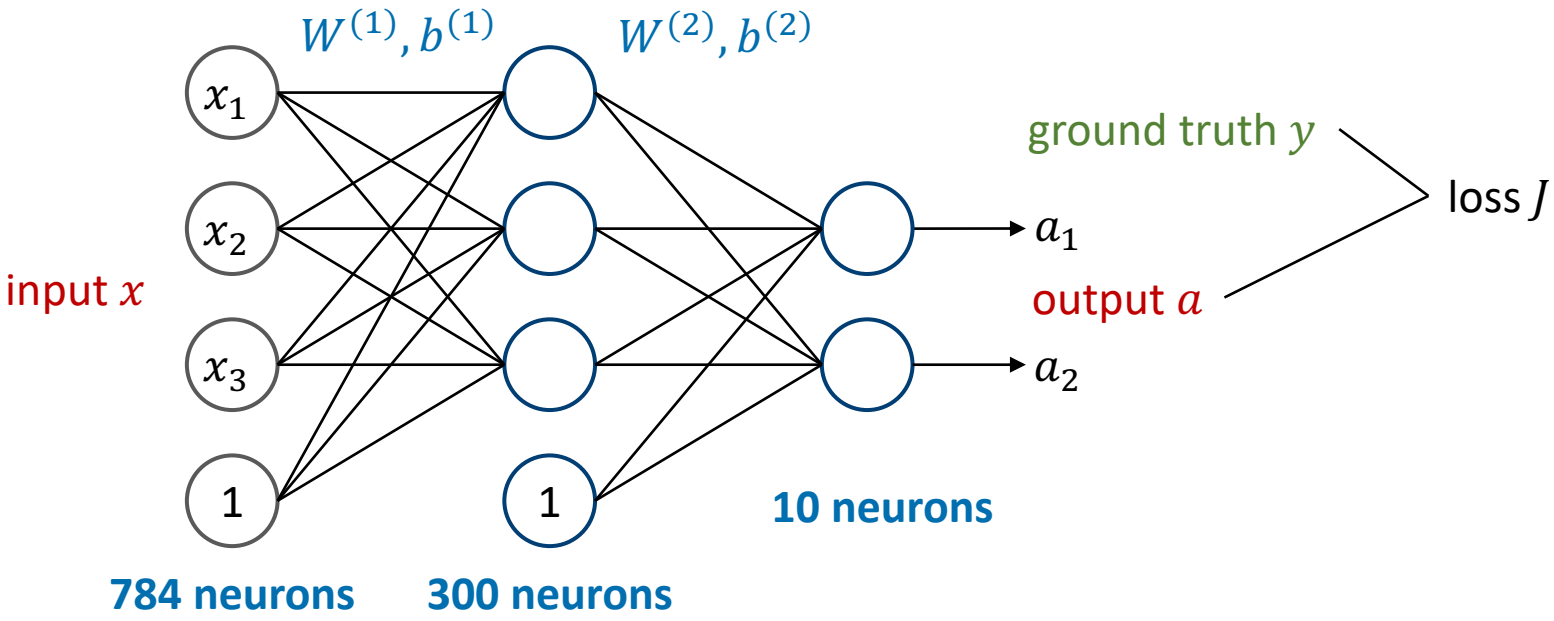
Multi-layer perceptron

- We build a 3-layered network
  - 784-300-10
  - cross entropy loss function



Multi-layer perceptron

- The network parameters can be trained using 60,000 training images.
- Then we can evaluate its performance on 10,000 test images.



Performance on MNIST

- This table reports the error rate.
  - Out of 10,000 test samples, how many have been wrongly classified.
  - Error rate = 100% - classification accuracy

Methods	Error Rate
KNN	5.0%
KNN (deslanted)	2.4%
SVM (polynomial, d=4)	1.1%
MLP (784-300-10)	4.7%

MNIST classification performance

Y. LeCun et al. Gradient-based learning applied to document recognition, Proceedings of IEEE, 1998.

Performance on MNIST

- We can also try different network architectures.
  - 784-300-10
  - 784-1000-10
  - 784-500-150-10 (4-layered)

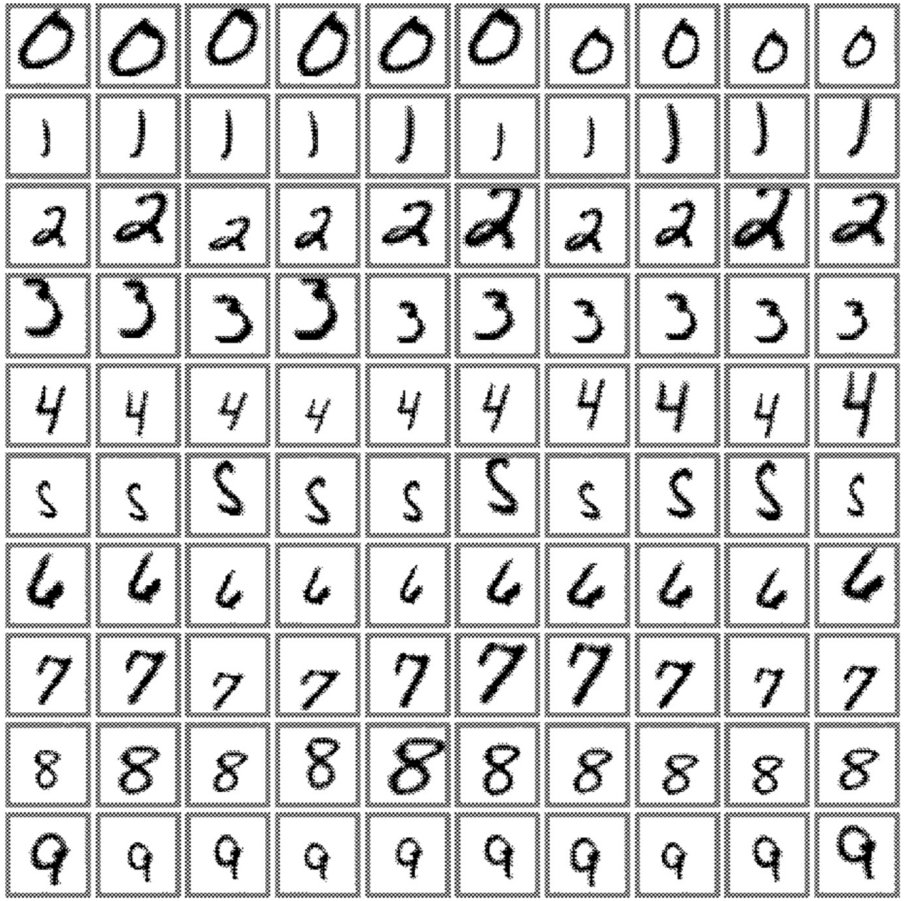
Methods	Error Rate
KNN	5.0%
KNN (deslanted)	2.4%
SVM (polynomial, d=4)	1.1%
MLP (784-300-10)	4.7%
MLP (784-1000-10)	4.5%
MLP (784-500-150-10)	2.95%

MNIST classification performance

Y. LeCun et al. Gradient-based learning applied to document recognition, Proceedings of IEEE, 1998.

Performance on MNIST

- We can perform data augmentation to create more training samples.
  - Apply affine transformation to the original 60,000 training samples
    - Translation
    - Scaling
    - Squeezing
    - Shearing
  - Generate augmented 540,000 samples



Examples of augmented samples.

Y. LeCun et al. Gradient-based learning applied to document recognition, Proceedings of IEEE, 1998.

Performance on MNIST

- Data augmentation improves classification performance.
- A 4-layered network with data augmentation can now achieve 2.45% error rate.

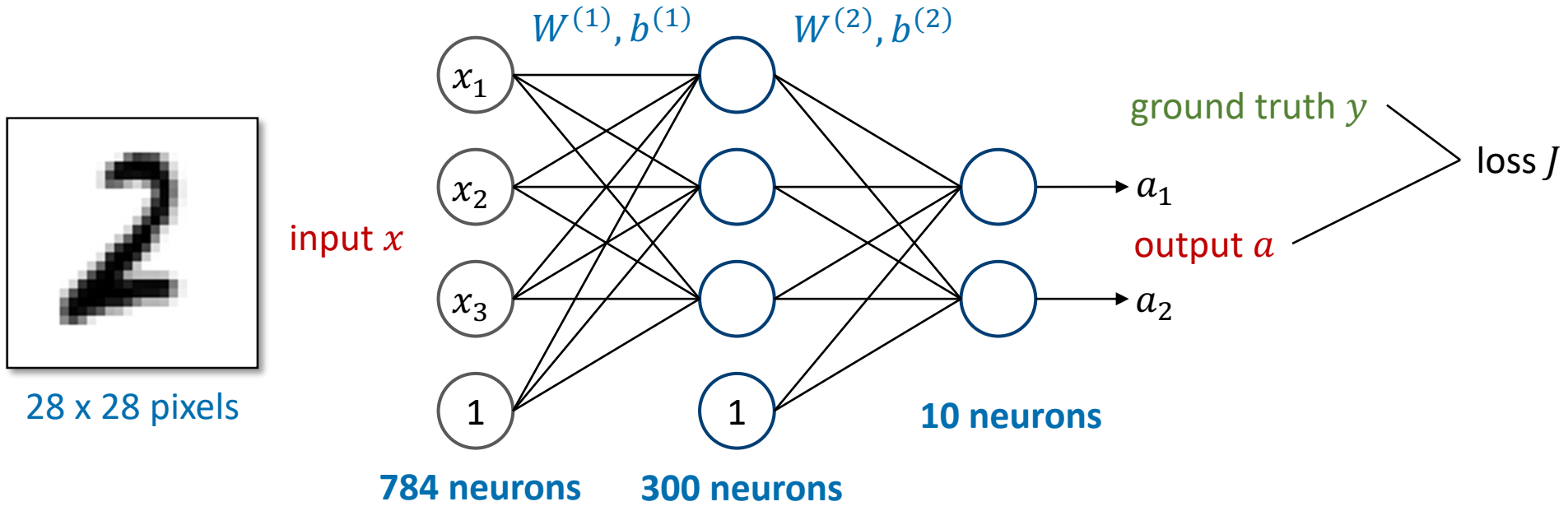
Methods	Error Rate
KNN	5.0%
KNN (deslanted)	2.4%
SVM (polynomial, d=4)	1.1%
MLP (784-300-10)	4.7%
MLP (784-300-10, aug.)	3.6%
MLP (784-1000-10)	4.5%
MLP (784-1000-10, aug.)	3.8%
MLP (784-500-150-10)	2.95%
MLP (784-500-150-10, aug.)	2.45%

MNIST classification performance

Y. LeCun et al. Gradient-based learning applied to document recognition, Proceedings of IEEE, 1998.

Limitations with MLP

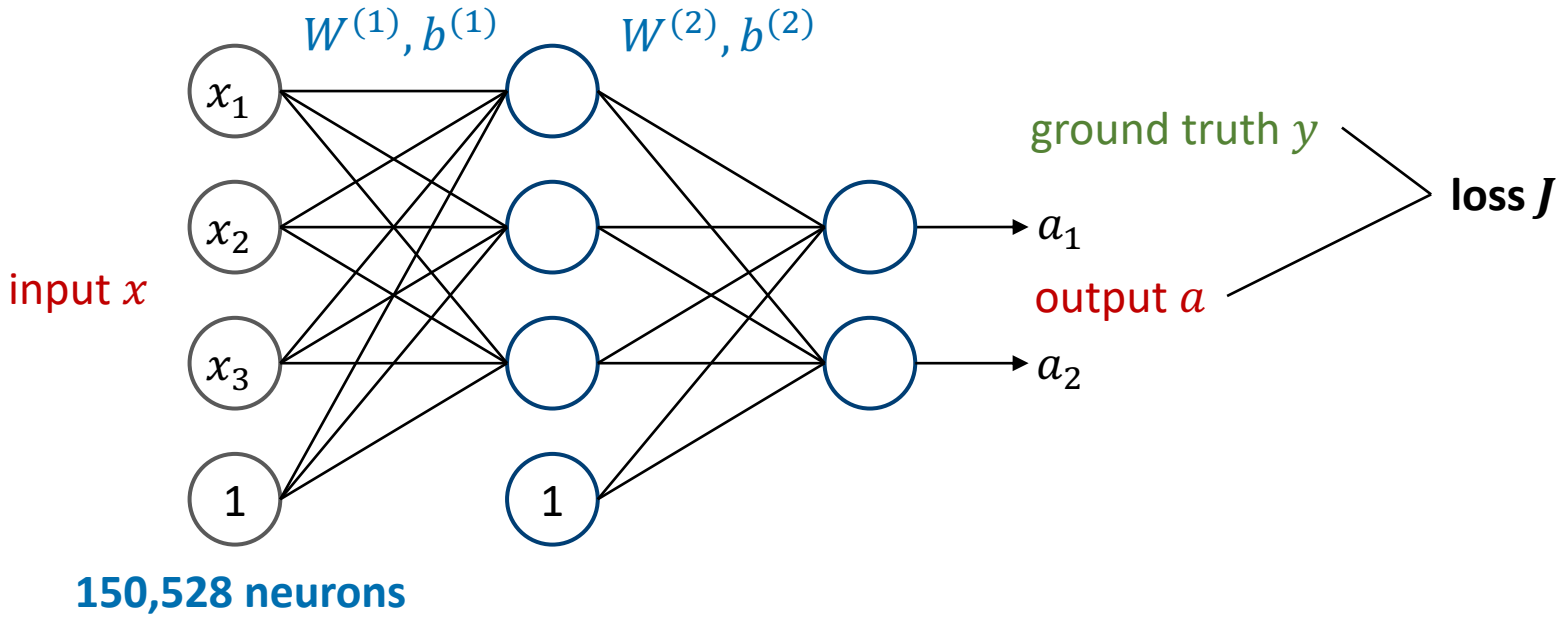
- MLP uses many parameters. Even for a 3-layered network (784-300-10), without considering the bias parameters, it will use
  - 784x300 + 300x10 = 238,200 parameters



Limitations with MLP

- It may not scale up to bigger images, e.g. for ImageNet images,
  - For a 224 x 224 RGB images, 224 x 224 x 3 channels = 150,528 neurons for Layer 1.
  - A single neuron in Layer 2 has 150,528 parameters, not to mention all neurons on this layer and following layers.

224 x 224 pixels by 3 channels



150,528 neurons

Limitations with MLP

- A reason why so many parameters are needed is because a 2D image is considered as a flattened vector, without considering its 2D nature.
- If we could develop an appropriate operator for 2D images, we may be able to have a more efficient network.

Summary

- SVM for image classification
- MLP for image classification

References

- Ch. 6, Deep Feedforward Networks. Ian Goodfellow et al. Deep Learning (https://www.deeplearningbook.org/).