

Online Joint GlueX-EIC-PANDA Machine Learning Workshop

Machine Learning for Beginners

Thomas Stibor

GSI
Helmholtzzentrum für Schwerionenforschung GmbH
t.stibor@gsi.de

21th September 2020 - 25th September 2020

Organizational

- Machine Learning for Beginners I, September 21th, 14:00 - 14:45
- Machine Learning for Beginners II, September 21th, 15:00 - 15:45
- Machine Learning for Beginners III, September 22th, 14:15 - 15:00
- Machine Learning for Beginners IV, September 23th, 14:15 - 15:00
- Support Vector Machines, September 24th, 15:15 - 16:00

Overview

- Literature
- Introductory Example
- Historical Overview
- Linear Classifiers
- Gradient Descent
- Neural Networks Learning (Backpropagation)
- Overfitting vs. Underfitting
- Bias-Variance Dilemma
- Support Vector Machines

Machine Learning is a large field, here we will focus and Neural Networks and Support Vector Machines.

Literature History of Artificial Intelligence & Machine Learning



Some figures are from:

The Quest for Artificial Intelligence (Nils J. Nilsson)

Literature Machine Learning



Some figures are from:

Pattern Recognition and Machine Learning (Christopher M. Bishop)


Literature Neural Networks



Literature Support Vector Machines



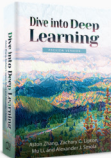
Literature Deep Learning



Dive into Deep Learning

Courses PDF All Notebooks Discuss GitHub 中文版

Preface
Installation
Notation
1. Introduction
2. Preliminaries
3. Linear Neural Networks
4. Multilayer Perceptrons
5. Deep Learning Computation
6. Convolutional Neural Networks
7. Modern Convolutional Neural Networks
8. Recurrent Neural Networks
9. Modern Recurrent Neural Networks
10. Attention Mechanisms
11. Optimization Algorithms
12. Computational Performance
13. Computer Vision
14. Natural Language Processing: Pretraining
15. Natural Language Processing: Applications
16. Recommender Systems
17. Generative Adversarial Networks



Dive into Deep Learning

An interactive deep learning book with code, math, and discussions

Provides NumPy/MXNet, PyTorch, and TensorFlow implementations

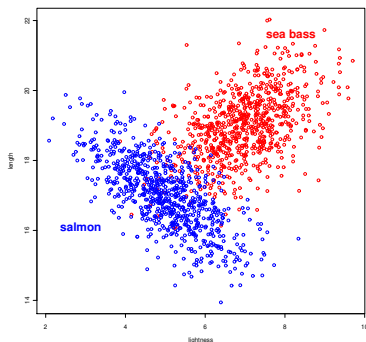
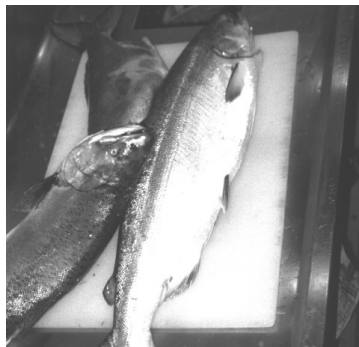
Announcements

- [Jul 2020] We have added TensorFlow implementations up to Chapter 7 (Modern CNNs).
- [Jun 2020] We have added PyTorch implementations up to Chapter 7 (Modern CNNs).
- [Apr 2020] We have re-organized [Chapter: NLP pretraining](#) and [Chapter: NLP applications](#), and added sections of BERT ([model](#), [data](#), [pretraining](#), [fine-tuning](#), [application](#)) and natural language inference ([data](#), [model](#)). To keep track of the latest updates, please follow D2L's [open source project](#).
- [Dec 2019] All the code has been rewritten with the NumPy API. Check out new instructions to run this book on [Amazon SageMaker](#) and [Google Colab](#).
- [Oct 2019] We have added [Chapter: Recommender Systems](#) and [Appendix: Mathematics for Deep Learning](#).
- [Jul 2019] The Chinese version is the [No. 1 best seller](#) of new books in "Computers and Internet" at the largest Chinese online bookstore.
- [May 2019] Slides, Jupyter notebooks, assignments, and videos of the Berkeley course can be found at the [syllabus page](#).

Navigation icons: back, forward, search, etc.

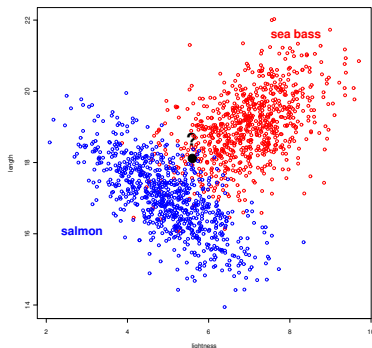
An Introductory Example

Suppose that a fishpacking factory wants to automate the process of sorting incoming fish (salmon and sea bass).



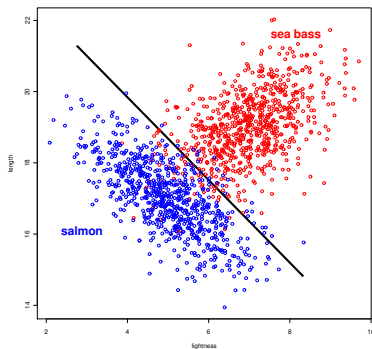
After some preprocessing, each fish is characterized by feature vector $\mathbf{x} = (x_1, x_2) \in \mathbb{R}^2$ (pattern), where the first component is the lightness and the second component the length.

Pattern belongs to Class?



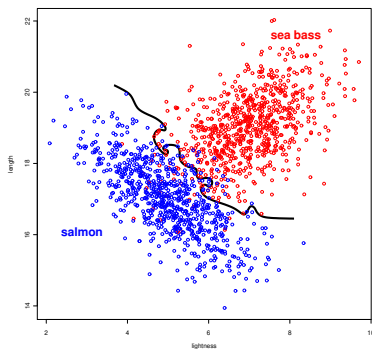
Given labeled training data $(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_N, y_N) \in \mathbb{R}^n \times Y$ coming from some unknown probability distribution $P(\mathbf{x}, y)$. In this example, $Y \in \{\text{salmon}, \text{sea bass}\}$ and $n = 2$. Unseen (unlabeled) pattern belongs to class salmon or sea bass?

A (too underfitted) Classifier



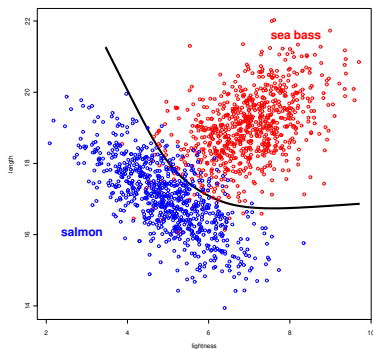
This linear separation suggests the rule: Classify the fish as salmon if its features falls below the *decision boundary*, otherwise as sea bass.

A (too overfitted) Classifier



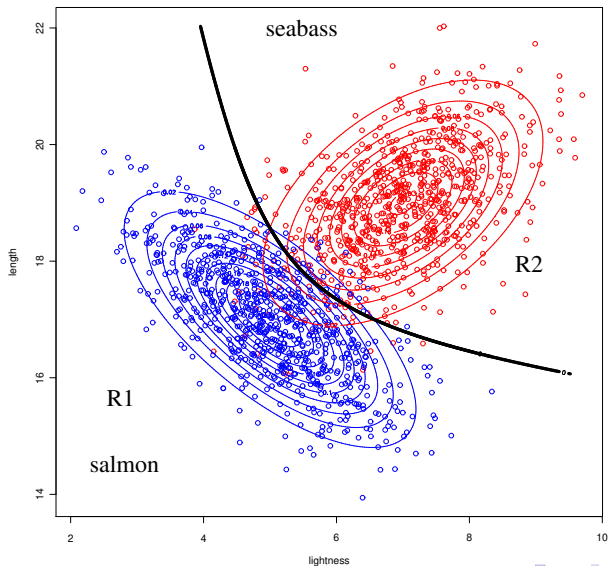
A too complex model will lead to decision boundary that gives perfect classification accuracy on training set (seen patterns), but poor classification on *unseen* patterns.

A Good Classifier

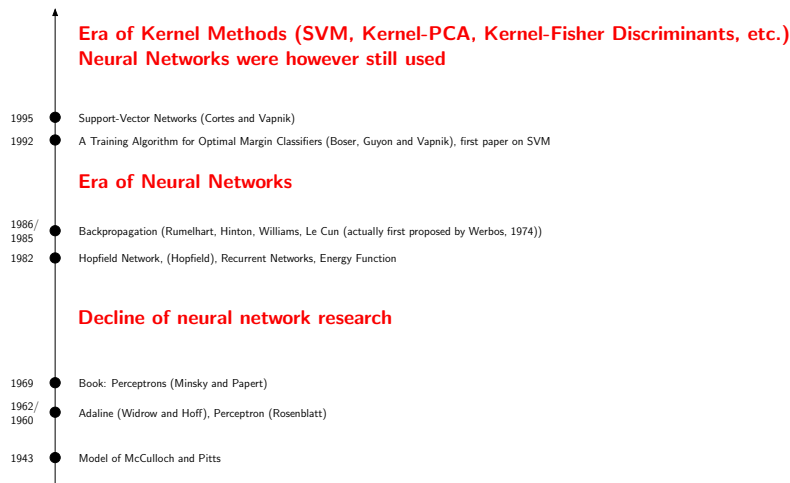


Optimal tradeoff between performance on the training set and simplicity of the model. This gives high classification accuracy on unseen patterns, i.e. it gives good *generalization*.

An Optimal Classifier



History of Neural Networks



Note, this historical overview is far from being complete
(c.f. [The Quest for Artificial Intelligence \(Nils J. Nilsson\)](#))

Neuron & Model of McCulloch and Pitts

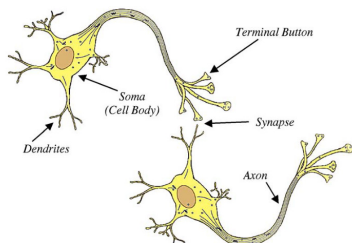


Figure 2.6: Two neurons. (Adapted from *Science*, Vol. 316, p. 1416, 8 June 2007. Used with permission.)

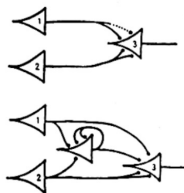


Figure 2.8: Networks of McCulloch–Pitts neural elements. (Adapted from Fig. 1 of Warren S. McCulloch and Walter Pitts, “A Logical Calculus of Ideas Immanent in Nervous Activity,” *Bulletin of Mathematical Biophysics*, Vol. 5, pp. 115–133, 1943.)

Taken from: [The Quest for Artificial Intelligence \(Nils J. Nilsson\)](#)

Book Perceptrons (Minsky and Papert)



Frank Rosenblatt
1928–1969

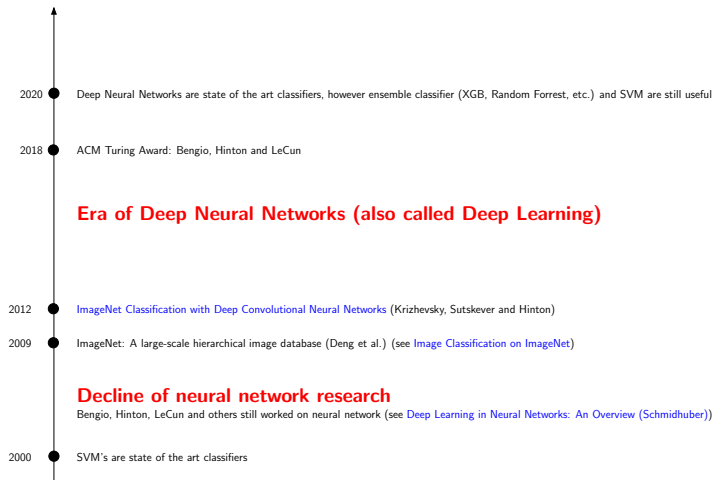
Rosenblatt's perceptron played an important role in the history of machine learning. Initially, Rosenblatt simulated the perceptron on an IBM 704 computer at Cornell in 1957, but by the early 1960s he had built special-purpose hardware that provided a direct, parallel implementation of perceptron learning. Many of his ideas were encapsulated in "Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms" published in 1962. Rosenblatt's work was criticized by Marvin Minsky, whose objections were published in the book "Perceptrons", co-authored with

Seymour Papert. This book was widely misinterpreted at the time as showing that neural networks were fatally flawed and could only learn solutions for linearly separable problems. In fact, it only proved such limitations in the case of single-layer networks such as the perceptron and merely conjectured (incorrectly) that they applied to more general network models. Unfortunately, however, this book contributed to the substantial decline in research funding for neural computing, a situation that was not reversed until the mid-1980s. Today, there are many hundreds, if not thousands, of applications of neural networks in widespread use, with examples in areas such as handwriting recognition and information retrieval being used routinely by millions of people.

Taken from:

Pattern Recognition and Machine Learning (Christopher M. Bishop)

History of Neural Networks (cont.)

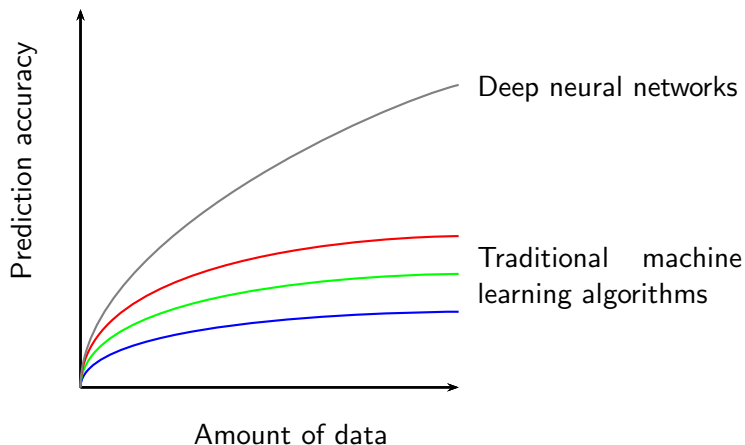


Overview ImageNet



- \approx 14 million annotated images to indicate what objects are pictured.
- Objects categorized into 1000 classes (e.g. *'Tibetan mastiff'*, *'Great Dane'*, *'Eskimo dog, husky'*, ...)
- Top-1 score: Check if predicted class with highest probability is the same as the target label.
- Top-5 score: Check if target label is one of your 5 predictions with highest probability.

Why are Deep Neural Networks so successful?



Deep Neural Networks (Backpropagation) are *universal*, that is, applicable to a large class of problems: Vision, speech, text, ... and *scale* with data. Backpropagation (forward + backward pass) is intrinsically linked to matrix multiplication (GPU's, TPU's).

Attendance AI & ML conferences (1984 - 2019)

Attendance at large conferences (1984-2019)

Source: Conference provided data.

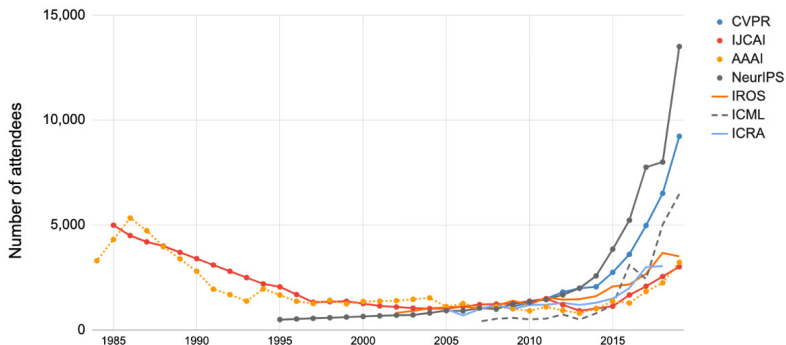


Fig. 2.1a

Note: IJCAI occurred every other year till 2014. The missing year between 1984 and 2014 are interpolated as the mean between the two known conference attendance dates to provide a comparative view across conferences.

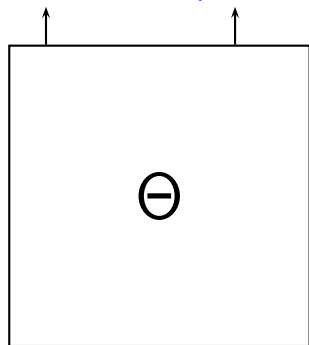
Taken from: [Artificial Intelligence Index, 2019 Annual Report \(pp. 39\)](#)

Machine Learning Framework

Machine Learning \equiv Optimization & Statistics

Data \equiv (input data, target data)

predicted data probability



```
while not min Loss $_{\Theta}$ (target data, predicted data) {  
    fit parameters  $\Theta$   
}
```

```
while not max Prob(target data, input data |  $\Theta$ ) {  
    fit parameters  $\Theta$   
}
```

Machine Learning Framework (Example SVM)

Machine Learning \equiv Optimization & Statistics

Data \equiv (input data \mathbf{x}_n , target data y_n)

while not min Loss $_{\Theta}$ (target data, predicted data) {

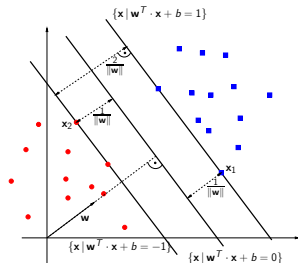
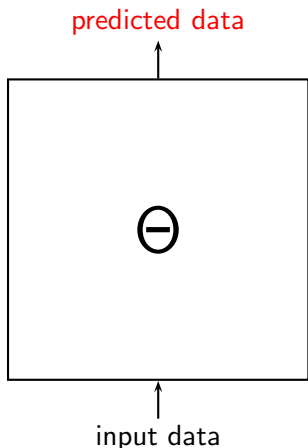
fit parameters $\Theta := \mathbf{w}, b$ (normal, offset)

}

$$\text{minimize } \frac{1}{2} \|\mathbf{w}\|^2$$

$$\text{subject to } y_n(\mathbf{w}^T \cdot \mathbf{x}_n + b) \geq 1$$

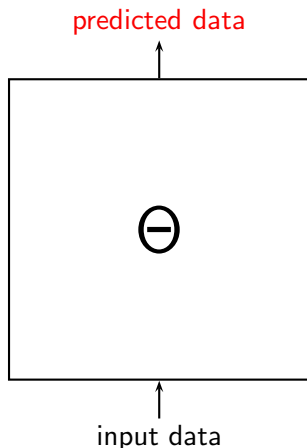
$$n = 1, \dots, N$$



Machine Learning Framework (Example One-Class SVM)

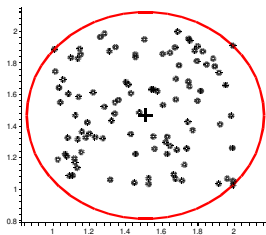
Machine Learning \equiv Optimization & Statistics

Data \equiv (input data \mathbf{x}_n)



while not $\min \text{Loss}_{\Theta}(\text{input data}) \{$
fit parameters $\Theta := \mathbf{c}, r$ (sphere center, radius)
 $\}$

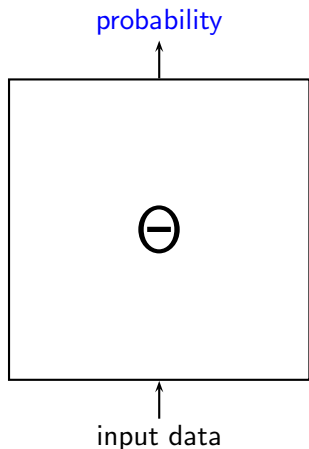
minimize r^2
subject to $\|\mathbf{x}_n - \mathbf{c}\|^2 \leq r^2$
 $n = 1, \dots, N$



Machine Learning Framework (Example HMM)

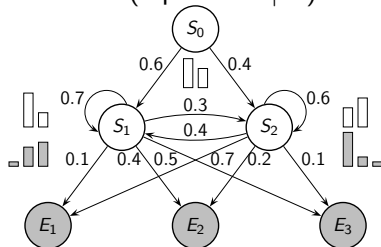
Machine Learning \equiv Optimization & Statistics

Data \equiv (input data)



```
while not max Prob(input data |  $\Theta$ ) {  
    fit parameters  $\Theta := \mathbf{s}, \mathbf{H}, \mathbf{E}$  (start vector, hidden  
    matrix, emission matrix)  
}
```

max Prob(input data | Θ)



Machine Learning Framework (Ex. Neural Networks (NN))

Machine Learning \equiv Optimization & Statistics

Data \equiv (input data \mathbf{X} , target data \mathbf{Y})

while not $\min \text{Loss}_{\Theta}(\text{input data, predicted data})$ {
fit parameters $\Theta := \mathbf{W}^{(1,2,3)}, \mathbf{b}^{(1,2,3)}$ (matrices,
vectors)
}

$$\text{minimize } \frac{1}{2} \| f(\mathbf{W}^{(3)} f(\mathbf{W}^{(2)} f(\mathbf{W}^{(1)} \mathbf{X} + \mathbf{b}^{(1)} + \mathbf{b}^{(2)} + \mathbf{b}^{(3)}) - \mathbf{Y} \|^2$$

