Neural Networks

doc. RNDr. Iveta Mrázová, CSc.

DEPARTMENT OF THEORETICAL COMPUTER SCIENCE AND MATHEMATICAL LOGIC

FACULTY OF MATHEMATICS AND PHYSICS, CHARLES UNIVERSITY IN PRAGUE

Neural Networks:

Introduction to the Area

doc. RNDr. Iveta Mrázová, CSc.

DEPARTMENT OF THEORETICAL COMPUTER SCIENCE AND MATHEMATICAL LOGIC

FACULTY OF MATHEMATICS AND PHYSICS, CHARLES UNIVERSITY IN PRAGUE

Neural Networks:

Contents:

- **Introduction to the Field**
- **Perceptron and Linear Separability**

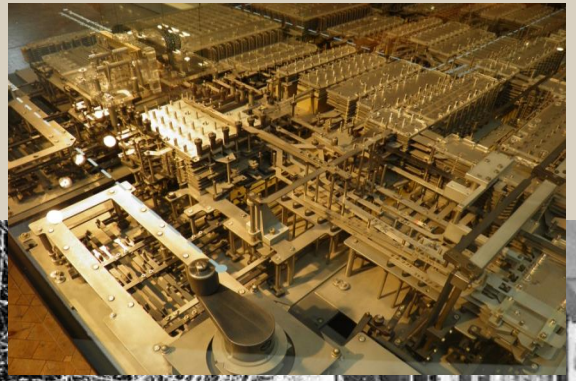
Contents:

Computer Versus Brain

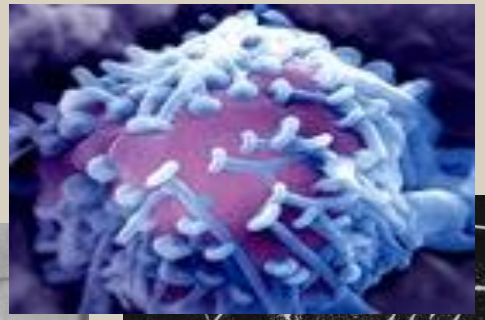
- The speed of information processing
- The kind of information processing
  - serial × parallel
- The kind of information storage
- Redundancy
- Control

Computer Versus Brain

The Z1 computer designed by Konrad Zuse in 1936/7



Mark I Perceptron machine built by F. Rosenblatt in 1957 (equipped with a camera to Process 20x20 pixel images)

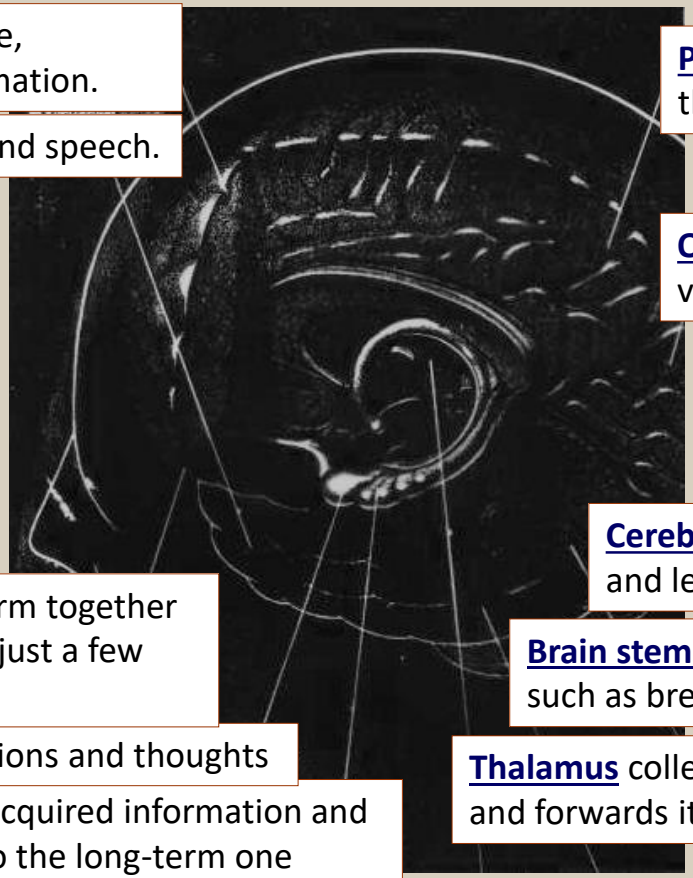


A biological neuron

Human brain

Frontal lobes plan for the future, movement control, speech formation.

Temporal lobes process and interpret music and speech.



Parietal lobes collect and process the data provided by the senses.

Occipital lobes specialize in visual information processing.

Cerebellum controls muscle coordination and learning of automated movements.

Cerebral cortex covers all the lobes, that form together the left and right cerebral hemisphere. It is just a few millimeters thick.

Brain stem controls automatically performer body functions such as breathing. It connects the brain with the spinal cord.

Amygdala creates emotions from the perceptions and thoughts

Thalamus collects the information from the senses and forwards it further to the brain cortex.

Hippocampus consolidates recently acquired information and transforms the short-term memory to the long-term one

Computer Versus Brain

The Structure of the Human Brain

Neural Networks – a Brief History

1943 – formal neuron (W. McCulloch, W. Pitts)

1949 – mathematical notion of learning (D. Hebb)

1958 – perceptron (F. Rosenblatt)

1962 – Adaline and sigmoidal transfer function (B. Widrow, M. Hoff)

1969 – The perceptrons (M. Minsky, S. Papert)

1980s – a further development

Neural Networks – a Brief History

since the eighties – **further developments**:

- The back-propagation training algorithm (P. Werbos, D. Rumelhart, G. Hinton, Y. Le Cun)
- Kohonen self-organizing feature maps (T. Kohonen)
- RBF-networks (Radial Basis Function, J. Moody, C. Darken)
- GNG-model (Growing Neural Gas, B. Fritzke)
- Convolutional neural networks (Y. Le Cun)
- SVM-machines (Support Vector Machines, V. Vapnik)
- ELM-networks (Extreme Learning Machines, G.-B. Huang)

Neural Networks – 21st Century

2003 - **Allen Brain Atlas** (Allen Institute for Brain Science, USA)

HBP – Human Brain Project, EU (January 2013)

Goal: mimic the human brain and identify faults its function

Expected costs – 1.2 billions Euro /10 year

https://www.humanbrainproject.eu/

http://www.nature.com/news/brain-simulation-and-graphene-projects-win-billion-euro-competition-1.12291

2013 – **BigBrain** (Montreal Neurological Institute and German Forschungszentrum Jülich)

https://bigbrain.loris.ca/main.php

Neural Networks – 21st Century

GRAND CHALLENGE BRAIN Initiative, President Obama, 2. 4. 2013, USA:

~ Brain Research Through Advancing Innovative Neurotechnologies

https://obamawhitehouse.archives.gov/BRAIN

Goal: understand, how we think, how we learn and how works our memory

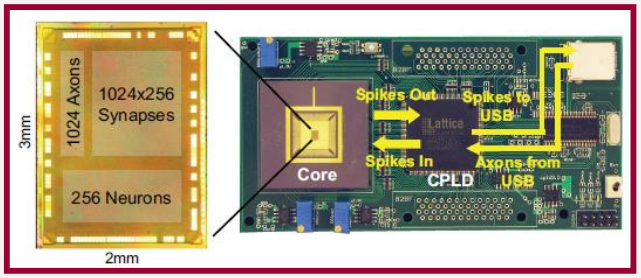
Expected costs – 3 billions USD / 10 years

Participants: DARPA ~ Defense Advanced Research Projects Agency
NIH ~ National Institutes of Health
NSF ~ National Science Foundation
private sector

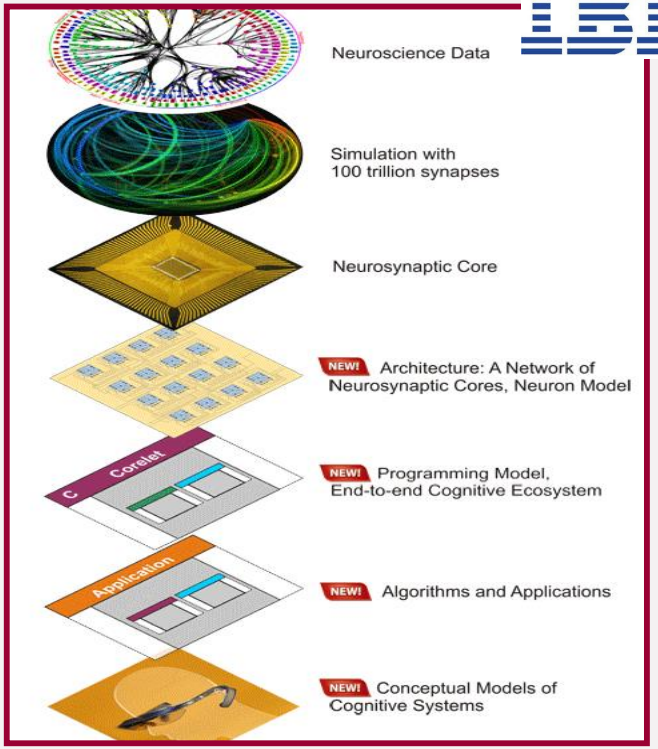
https://www.nimh.nih.gov/news/science-news/science-news-about-the-brain-initiative

http://www.nature.com/news/flashing-fish-brains-filmed-in-action-1.12621

New Technologies



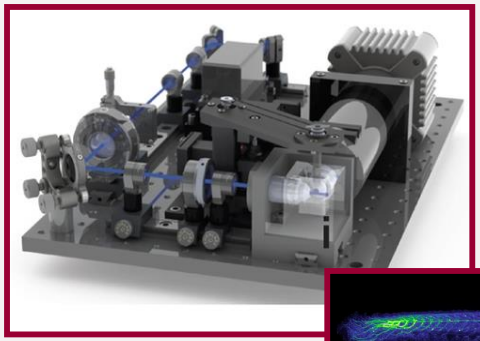
❑ Neurosynaptic chip

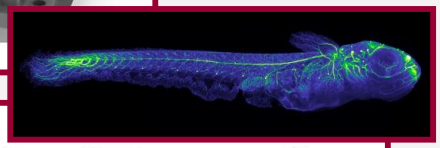


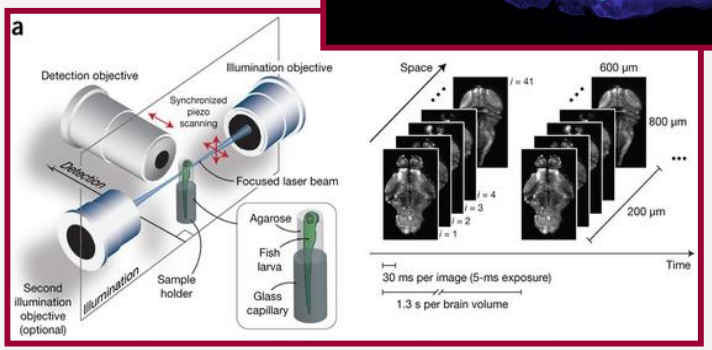
11. 8. 2024 – Pinnacle Business Systems: Scientists from IBM unveiled the first neurosynaptic chip to achieve a scale of:

- one million programmable neurons,
- 256 million programmable synapses, and
- 46 billion synaptic operations per second per Watt.

❑ Light-sheet microscopy







http://www.nature.com/nmeth/journal/v10/n5/fig_tab/nmeth.2434_SV4.html

Neural Networks – a General Introduction

Recent problems:

- Training strategies – parallelization and efficiency, few-shot training, generative models
- Architecture – generalization and robustness, non-standard forms of data
- Scalability – GPU, TPU, Google Cloud Platform, Edge TPU, …
- Convergence, (few-shot) training, and over-training
- Prediction and generative models
- Adversarial patterns

Applications:

- Data mining – „black-box“, „white-box“
- Clustering and classification
- Information processing – NLP, speech, vision, olfactory, tactile, motoric
- Artworks
- Solutions of optimization tasks
- and many others ….

Neural Networks – a General Introduction

Recent problems:

- Training strategies – parallelization and efficiency, few-shot training, generative models
- Architecture – generalization and robustness, non-standard forms of data
- Scalability – GPU, TPU, Google Cloud Platform, Edge TPU, …
- Convergence, (few-shot) training, and over-training
- Prediction and generative models
- Adversarial patterns

Applications:

- Data mining – „black-box“, „white-box“
- Clustering and classification
- Information processing – NLP, speech, vision, olfactory, tactile, motoric
- Artworks
- Solutions of optimization tasks
- and many others ….

Neural Networks:

Contents:

- **Introduction to the Field**
- **Perceptron and Linear Separability**

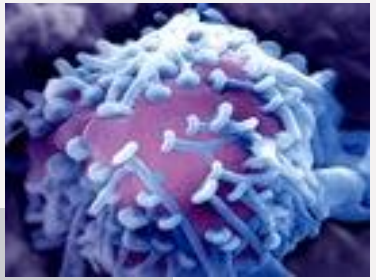
Contents:

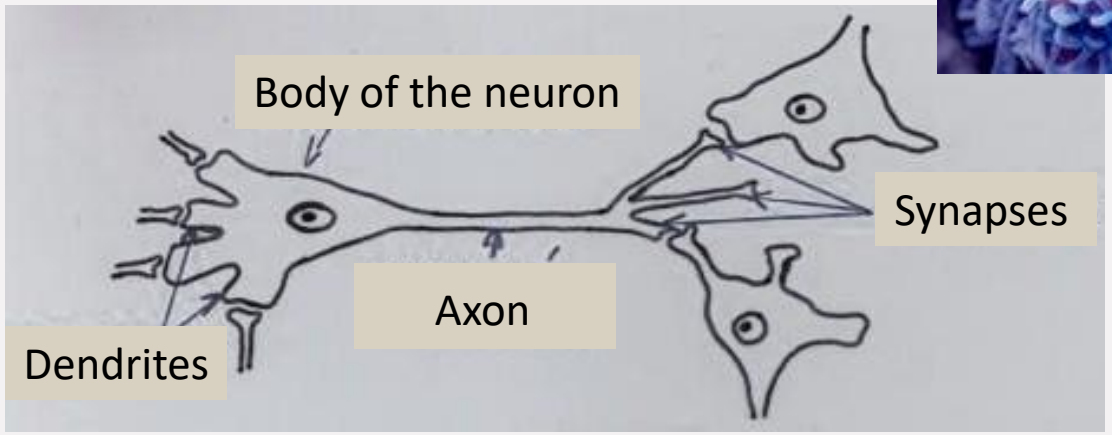
Biological Background (1)

Model of a neuron

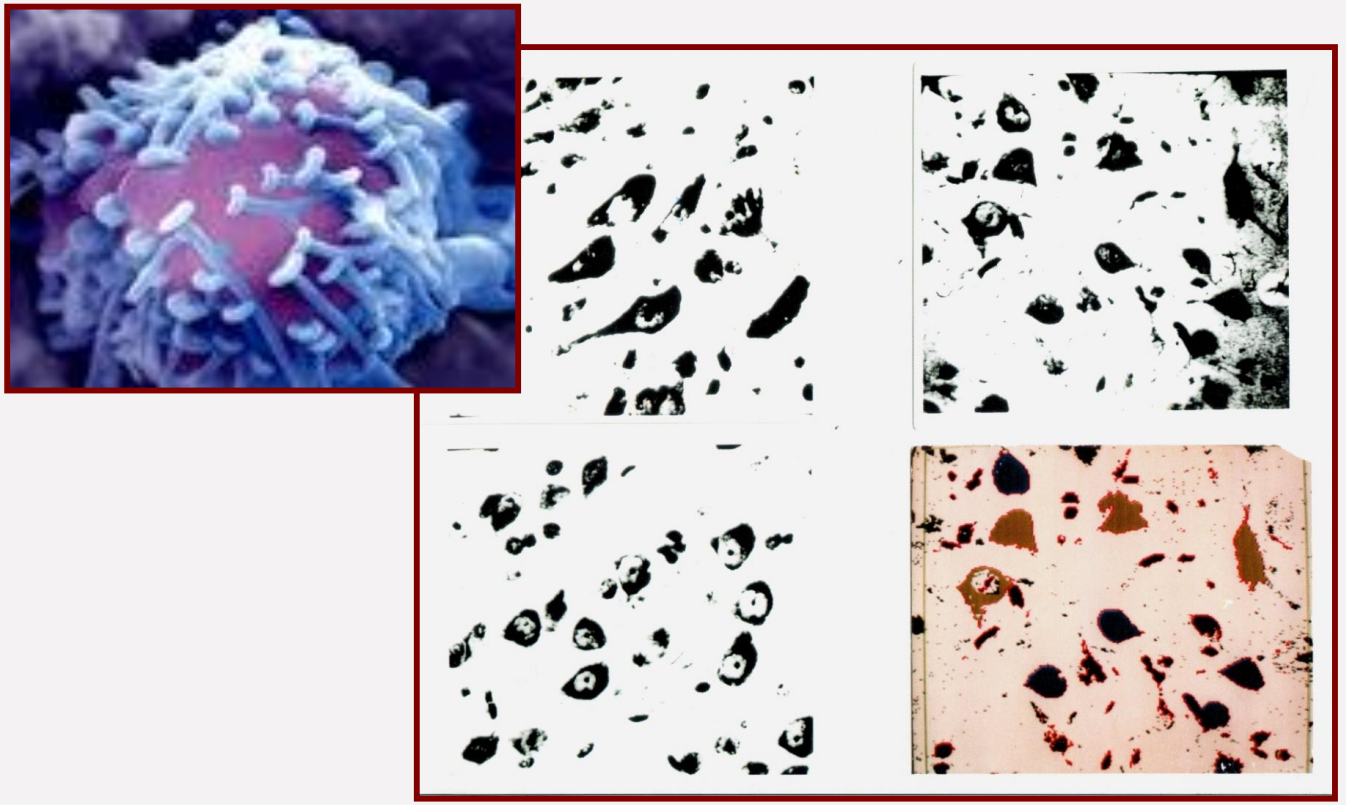
- basic „computational unit“ of a more complex system
  - neural network (contains cca 8.6×10^{10} neurons)
- biological neurons consist of:

  body (soma), dendrites, axon and synapses





Biological Background (1) - Biological Neuron



Biological Background (2) - Biological Neuron

Body (soma):

- summarizes signals transmitted by surrounding neurons → **potential**
- inner potential leads to the **excitation** of the neuron
- the size varies from several μm to several tens of μm (~ 10^{-6} m)

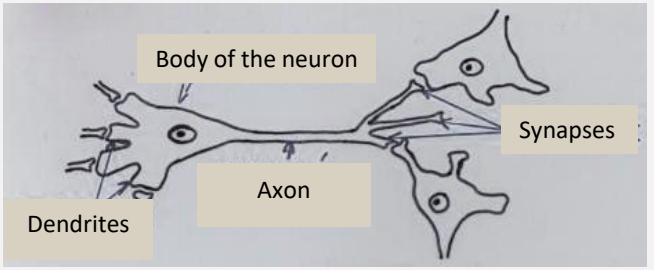
Dendrites:

- represent signal input to neuron body
- their length varies around 2-3 mm (~ 10^{-3} m)

Biological Background (3) - Biological Neuron

Axon:

- the only output of a neuron, but branched out widely at its end
- transmits the signal given by the level of excitation to the synapses
- its length can reach over 1 m



Synapse:

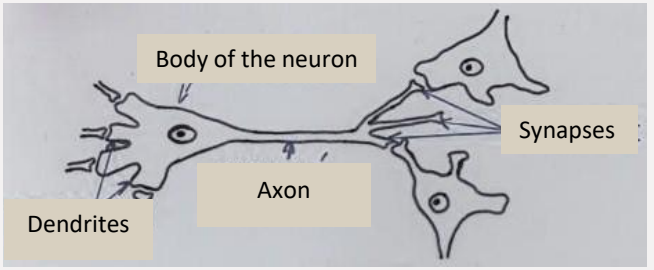
- represent the „output device“ of the neurons, can the signal amplify or diminish and transmit it to other neurons
- for each neuron, there are up to 10^6 connections to other neurons

Neuron output:

- Depends on neuron inputs and their processing inside neuron body

Biological Background (4) - Biological Networks

Biological neural networks:



- **neurons are mutually interconnected into networks**
  - by means of axons connected to dendrites of other neurons via synapses
- **density of the neurons:**
  - reaches cca $70 – 80 \cdot 10^3$ / mm^3 in the human brain
  - cca $10 \cdot 10^3$ neurons die every day without replacement
  - synapses are formed on the dendrites during the whole life

    → **new synapses are formed**, resp. **non-functioning synapses can be revived**

=> L E A R N I N G

Question:

What percentage of a human brain is lost while alive?

Assume the age at death to be 100 years.

Biological Background (5) – Memory Types

Memory types

- **Short-term memory mechanism**
  - based on cyclical circulation of signals in neural networks
  - after cca 300 circulations, fixation of the information starts in mid-term memory – this takes cca 30 s

- **Mid-term memory mechanism**
  - based on the changes of „neural weights“
  - the change of synaptic weight coefficients is caused by multiple actions of the same signal on the respective synapse

Biological Background (6) – Memory Types

Memory types

- **Mid-term memory mechanism**
  - some information stored in mid-term memory moves to long-term memory while sleeping
  - information stays in mid-term memory for several hours or days

- **Long-term memory mechanism**
  - consists in copying the information from mid-term memory to proteins inside the neurons – in particular in their nuclei
  - the stored information can remain in the organism for its entire life

Neural Networks:

Contents:

- **Introduction to the Field**
- **Perceptron and Linear Separability**

Contents:

Adaptation and Learning

Adaptation:

- ability to accommodate to the changes of the environment

Adaptive process: the process of the adjustment

- every adaptation represents for the system some costs (material, energy, …)
- living organisms are capable of reducing these costs during multiply repeated adaptations to environment changes

L E A R N I N G:

- its objective is to minimize the costs spent for adaptation
- is the result of a multiply repeated adaptation

Adaptation and Learning - the Formalism (1)

- **Manifestation of the environment:** $\boldsymbol{x}$

  ~ input pattern generated by the environment

- **Feature description of the objects** *(~ input patterns)* **:**
  - selection of n basic characteristics – features $x_1, \dots, x_n$
  - $\boldsymbol{x} = (x_1, \dots, x_n)$
- **Information about the desired system reaction to the manifested environment:** Ω

  ~ e.g., the true class assignment function for the input patterns

- The system reacts to any manifestation of the environment *(~ input pattern generated by the environment)* $\boldsymbol{x}$ and information Ω *(~ true class assignment)* by yielding one of the symbols ω_r; $r = 1, \dots, R$ at its output *(~ actual class labels)*.

Adaptation and Learning - the Formalism (2)

- Every assignment $[\boldsymbol{x}, \Omega] \rightarrow \omega_r$ is accompanied by some costs given by the function $Q(\boldsymbol{x}, \Omega, \omega_r)$ for each time unit
- **The goal of the system:**
  - find for any $\boldsymbol{x}$ and Ω such an assignment $[\boldsymbol{x}, \Omega] \rightarrow \omega_r$, for which the **cost is minimal:**

$$\boldsymbol{Q(x, \Omega, \omega_r) = \min_{\omega} Q(x, \Omega, \omega)}$$

Adaptive Systems (1)

Adaptive system ~ a system with two inputs and one output determined by:

1) a set X **of manifestations of the environment** $\boldsymbol{x}$ *(~ input patterns generated by the environment)*
2) a set O_1 **of information about the desired system reaction** Ω *(e.g., true class assignments for the input patterns)*
3) a set O_2 **of output symbols** ω *(~ actual class labels)*.
4) a set D **of** *(parametrized class label)* **decision rules** $\omega = d(\boldsymbol{x}, q)$
5) **the cost** $Q(\boldsymbol{x}, \Omega, q)$

For any pair $[\boldsymbol{x}, \Omega]$ we seek such a parameter q^*, for which it holds:

$$\boldsymbol{Q(x, \Omega, q^*) = \min_{q} Q(x, \Omega, q)}$$

Adaptive Systems (2)

- Initial assignment $[\boldsymbol{x}, \Omega] \rightarrow \omega_s$
- If the system stays for time T in its initial assignment, this will be associated with total costs corresponding to $\boldsymbol{T \cdot Q(x, \Omega, \omega_s)}$

- If the system is able to change its behavior based on an ongoing cost assessment, it finds **after the time τ necessary for evaluating** ω_r , for which the cost is minimal

Adaptive Systems (3)

Total costs after time T :

$$\tau\, Q(x, \Omega, \omega_s) + (T - \tau) Q(x, \Omega, \omega_r)$$

- are bigger than the least possible total costs $T\, Q(x, \Omega, \omega_r)$,
- but smaller than the total costs of a system, that cannot change its decision, $T\, Q(x, \Omega, \omega_s)$

Adaptive system

$$T\, Q(x, \Omega, \omega_r) < \tau\, Q(x, \Omega, \omega_s) + (T - \tau) Q(x, \Omega, \omega_r) < T\, Q(x, \Omega, \omega_s)$$

Learned system

Non-adaptive system

Learning Systems (1)

- **The result of adaptation is stored in the memory:**
  - Save the time τ necessary to find minimum costs for repeated manifestations of the environment *(~ input pattern sets generated by the environment)*
  - Further, it is not necessary to evaluate the costs

    → after training, the information Ω about the desired system reaction is not necessary anymore

- **The total costs of a learning system after training** $T\ Q(x, \Omega, \omega_r)$ are smaller than the total costs of an adaptive system

Learning Systems (2)

Learning system ~ a system with 2 inputs and 1 output determined by:

1) a set X **of manifestations of the environment** $\boldsymbol{x}$ *(~ input patterns generated by the environment)*
2) a set O_1 **of information about the desired system reaction** Ω *(e.g., true class assignments for the input patterns;* ***is not necessary after training****)*
3) a set O_2 **of output symbols** ω *(~ actual class labels)*
4) a set D **of** *(parametrized class label)* **decision rules** $\omega = d(\boldsymbol{x}, q)$
5) The **desired behavior** $\Omega = T(x)$ *(****in addition to adaptive systems****)*
6) **Mean costs** $J(q)$ evaluated over $X \times O_1$ *(****different from adaptive systems****)*

Learning Systems (3)

Learning system

- After presenting the pair elements from the sequence

$$\{ [\boldsymbol{x_k}, \boldsymbol{\Omega_k}] \};\ \boldsymbol{1 \leq k \leq \infty},\ \text{where}\ \boldsymbol{\Omega_k = T_k(x_k)},$$

it finds such a parameter $\boldsymbol{q^*}$, for which it holds:

$$\boldsymbol{J(q^*) = \min_{q} J(q)}$$

- **Sequential** ~ sequential presentation of the pairs $[\boldsymbol{x_k}, \boldsymbol{\Omega_k}]$
- **Inductive** ~ find after the evaluation of countably many pairs $[\boldsymbol{x_k}, \boldsymbol{\Omega_k}]$ the parameter $\boldsymbol{q^*}$, that minimizes the mean costs over the entire set $\boldsymbol{X}$

Efficiency of Adaptation and Learning

The efficiency of an adaptive system is the higher, the shorter is the time $\boldsymbol{\tau}$ necessary for its adaptation and the longer are the time intervals $\boldsymbol{T}$ when the environment does not change:

- $\boldsymbol{\tau/T \rightarrow 0}$:

  The efficiency of the AS is comparable with the efficiency of a learning system after training

- $\boldsymbol{\tau/T \rightarrow 1}$ $\boldsymbol{(\tau/T < 1)}$:

  AS has about the same efficiency as a non-adaptive system

- $\boldsymbol{\tau/T \geq 1}$: no adaptation takes place

The efficiency of the (trained) learning system is the highest possible

Neural Networks:

Contents:

- **Introduction to the Field**
- **Perceptron and Linear Separability**

Contents:

Data Standardization

- In the Euclidean space, standardization of attributes is recommended so that all attributes can have an equal impact on the computation of distances.
- **Example:** Consider the following pair of data points
  - $\mathbf{x}_i$: $(0{,}1; 20)$ and $\mathbf{x}_j$: $(0{,}9; 720)$,

$$dist(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{(0{,}9 - 0.1)^2 + (720 - 20)^2} = 700{,}000457$$

- The distance is dominated by $(720 - 20) = 700$.
- **Standardize the attributes** to have a common value range

Interval-Scaled Attributes

- Their values are real numbers following a linear scale.
  - e.g., the difference in Age between 10 and 20 is the same as that between 40 and 50.
  - **The key idea** is that the intervals keep the same importance throughout the scale
- Decimal scaling to the interval $[-1,1]$: divides the attribute values by the smallest power of 10 that keeps all the transformed values within the interval $[-1,1]$.
- Range standardization to the interval $[0,1]$ normalizes attribute values in the following way (f is an attribute):

$$range(x_{if}) = \frac{x_{if} - \min(f)}{\max(f) - \min(f)} \quad .$$

Interval-Scaled Attributes (cont …)

- Range standardization to the interval $[-1,1]$ normalizes attribute values in the following way (using $range(x_{if})$ defined above):

$$\boldsymbol{range^{[-1,1]}(x_{if}) = 2 \cdot range(x_{if}) - 1}$$

- Standardization according to the mean absolute deviation (MAD) transforms the attribute values so that they have zero mean and mean absolute deviation equal to 1.
  - The mean absolute deviation of attribute *f* (denoted by ${}^{MAD}S_f$) is computed as follows:

$$m_f = \frac{1}{n}(x_{1f} + x_{2f} + \ldots + x_{nf}),$$

$${}^{MAD}S_f = \frac{1}{n}(|x_{1f} - m_f| + |x_{2f} - m_f| + \cdots + |x_{nf} - m_f|),$$

MAD-score: $\boldsymbol{z(x_{if}) = \frac{x_{if} - m_f}{{}^{MAD}S_f}}$

Interval-Scaled Attributes (cont …)

- Standardization according to standard deviation transforms the attribute values so that they have zero mean and (corrected) sample standard deviation equal to 1.
  - The (corrected) sample standard deviation of attribute *f* (denoted by ${}^{cor}s_f$) is computed as it follows:

$$m_f = \frac{1}{n}\left(x_{1f} + x_{2f} + \cdots + x_{nf}\right),$$

$${}^{cor}s_f = \sqrt{\frac{1}{n-1}\left((x_{1f} - m_f)^2 + (x_{2f} - m_f)^2 + \cdots + (x_{nf} - m_f)^2\right)},$$

std-score: $\boldsymbol{std(x_{if}) = \frac{x_{if} - m_f}{{}^{cor}s_f}}$.

Question:

Let X be a dataset consisting of the following five elements:

$$X = \{\, \mathbf{0,1};\ \mathbf{-2,0};\ \mathbf{1,5};\ \mathbf{-0,2};\ \mathbf{-1,3} \,\}.$$

State its standardized forms.

Selection and Order of Features

Probability of a wrong decision

×

Information contained in the input patterns

- **Too many features:**
  - technical feasibility
  - speed of processing
  - danger of over-training
  - the number of variables × the number of training patterns
  - correlated features

Selection of Informative Features

- **Selection of the minimum number of features** from the considered set of features
  - the chosen set is not guaranteed to contain really informative features
  - the choice depends on the actual task solved

- **The order of features** from the considered set of features
  - according to the amount of information contained
  - can be used, e.g., in the case of sequential classifiers

Karhunen-Loeve Transform (1)

Properties of the Karhunen-Loeve transform:

1. For the given number of expansion members, it yields the **least mean squared error** between the original and the transformed patterns
2. After the application of the covariance matrix the approximated patterns are decorrelated

   → **decorrelation of features**

Karhunen-Loeve Transform (2)

3. Expansion members **do not contribute equally to the approximation**
   - The influence of each respective expansion member on the approximation accuracy falls with its index

     → The impact of members with high indexes will be small and we can thus omit them

4. The **magnitude of the approximation error does not influence the structure of the expansion**
   - Changed demands on the approximation error do not require the recomputation of the entire expansion

     → It is sufficient to add or remove a few of the last members

Of advantage especially for **sequential classification methods**

Karhunen-Loeve Transform (3)

- The choice of a suitable mapping of patterns from $\boldsymbol{X}^m$ to $\boldsymbol{X}^p$ such that the patterns from $\boldsymbol{X}^p$ will represent the best approximation of the original patterns from $\boldsymbol{X}^m$ in the sense of the mean squared error

  K patterns
  m features
  p orthonormal vectors $\boldsymbol{e}_i$ $(1 \leq i \leq p)$ in $\boldsymbol{X}^m$ $(p \leq m)$

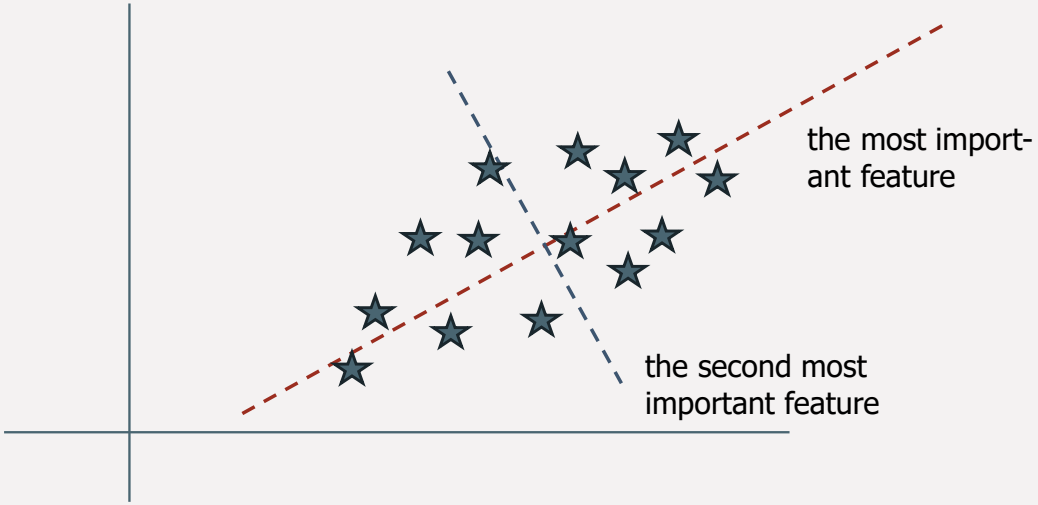
  → Approximate the vectors $\mathbf{x}_k$ from $\boldsymbol{X}^m$ $(1 \leq k \leq K)$ by a linear combination of $\boldsymbol{e}_i$:

$$\mathbf{y}_k = \sum_{i=1}^{p} c_{ki} e_i$$

  such that the squared error $\varepsilon_k^2 = \|\mathbf{x}_k - \mathbf{y}_k\|^2$, will be minimal

the data matrix $\mathbf{X} = \begin{pmatrix} x_{11} & \cdots & x_{1m} \\ \vdots & \ddots & \vdots \\ x_{K1} & \cdots & x_{Km} \end{pmatrix}$ 1st pattern / Kth pattern

Karhunen-Loeve Transform (4)



$$\mathbf{v} = (v_1, v_2, \ldots, v_m)^{\mathrm{T}},$$
$$\mathbf{x} = (x_1, x_2, \ldots, x_m)^{\mathrm{T}}$$
$$\mathbf{y} = (y_1, y_2, \ldots, y_p)^{\mathrm{T}}$$
$$\mathbf{y} = \mathbf{v}^{\mathrm{T}}\mathbf{x} = v_1 x_1 + v_2 x_2 + \cdots$$

From m measured features, we want to get the p most important features ($1 \leq p \ll m$)

Matrix $\mathbf{V}$: $m \times p$

the p-th EV

$$\mathbf{V} = \begin{pmatrix} v_{11} & \cdots & v_{1p} \\ \vdots & \ddots & \vdots \\ v_{m1} & \cdots & v_{mp} \end{pmatrix}$$

the 1-st EV

Compute the vector of the p most important features:

$$\mathbf{y} = \mathbf{V}^{\mathrm{T}}\,\mathbf{x}$$

Karhunen-Loeve Transform (5)

Observations are the rows

$$X = \begin{pmatrix} \cdots & x_{1j} & \cdots \\ \cdots & \cdots & \cdots \\ \cdots & x_{Kj} & \cdots \end{pmatrix}$$

$$\mu = \begin{pmatrix} \cdots & \mu_j & \cdots \end{pmatrix}$$

Computation of the matrix **V**:

- Center the data: $\mu_j = \frac{1}{K}\sum_{k=1}^{K} x_{kj}$
- Compute the covariance matrix for the training set:

$$w_{ij} = w_{ji} = \frac{1}{K}\sum_{k=1}^{K}(x_{ki} - \mu_i) \cdot (x_{kj} - \mu_j)$$

- The vectors defining the most important features correspond to the eigenvectors of the covariance matrix

Karhunen-Loeve Transform (6)

The eigenvalues correspond to the variance of the most important features

- the 1st column of the matrix $\mathbf{V}$ is the eigenvector corresponding to the biggest eigenvalue, …
- further columns of $\mathbf{V}$ will be added until the following eigenvalues are too small and can be omitted
- covariance matrices are positive semi-definite (~ their **eigenvalues are non-negative;** $\boldsymbol{\lambda_i \geq 0 \ \forall \ i}$, because they are computed from the sums of squares, which themselves are each non-negative).

Problem:

- The choice of an adequate number of eigenvalues (p)
- An optimal choice of p cannot be guaranteed as the expansion does not reflect the true importance of each respective feature

Question:

Let Σ be a covariance matrix of a two-dimensional dataset:

$$\boldsymbol{\Sigma} = \begin{pmatrix} 2{,}0 & 0{,}8 \\ 0{,}8 & 0{,}6 \end{pmatrix}.$$

Find its principal components.

Karhunen-Loeve Transform (7)

Modifications:

1. **Centered most important features:** $\mathbf{y} = \mathbf{V}^{\mathbf{T}}(\mathbf{x} - \boldsymbol{\mu})$,

   where $\boldsymbol{\mu} = (\mu_1, \dots)$ is the vector of mean values

2. **Normalized most important features:** $\mathbf{y} = \mathbf{L}^{-\frac{\mathbf{1}}{\mathbf{2}}}\mathbf{V}^{\mathbf{T}}(\mathbf{x} - \boldsymbol{\mu})$,

   where $\mathbf{L}$ is the matrix $p \times p$, diagonal elements are the eigenvalues corresponding to the columns of $\mathbf{V}$, the other elements are zero

Neural Networks:

Contents:

- **Introduction to the Field**
- **Perceptron and Linear Separability**

Contents:

Probability – Basic Notions (1)

Probability (of an event A from the space S):

- $\mathbf{P}(A) \geq 0$ ($\mathbf{P}(\emptyset) = 0$)
- $\mathbf{P}(S) = 1$
- For a finite number of mutually exclusive events $A_1, A_2, \ldots, A_n$ the probability $P(A_1 \cup A_2 \cup \cdots \cup A_n) = \sum_{i=1}^{n} P(A_i)$
- For an infinite number of mutually exclusive events $A_1, A_2, \ldots$ the probability $P(A_1 \cup A_2 \cup \cdots) = \sum_{i=1}^{\infty} P(A_i)$

Probability – Basic Notions (2)

- **Conditional probability** of the event $\boldsymbol{B}$ given that the event $\boldsymbol{A}$ has occurred ($\mathbf{P}(\boldsymbol{A}) > \mathbf{0}$):

$$\boldsymbol{P}(\boldsymbol{B}|\boldsymbol{A}) = \frac{\boldsymbol{P}(\boldsymbol{A} \cap \boldsymbol{B})}{\boldsymbol{P}(\boldsymbol{A})}$$

probability of B given A; probability of A and B; probability of A

- **Mutual independence** of the events $\boldsymbol{A}$ and $\boldsymbol{B}$**:**

$$\mathbf{P}(\boldsymbol{A} \cap \boldsymbol{B}) = \mathbf{P}(\boldsymbol{A}) \cdot \mathbf{P}(\boldsymbol{B})$$

- **Formula for the probability** of $\boldsymbol{A}$:

$$\boldsymbol{P}(\boldsymbol{A}) = \sum_{\boldsymbol{i}} \boldsymbol{P}(\boldsymbol{A}|\boldsymbol{B_i})\boldsymbol{P}(\boldsymbol{B_i})$$

Probability – Basic Notions (3)

- **Bayesian formula for the conditional probability:**

probability of B given A — probability of B given A — probability of B

$$\mathbf{P}(\boldsymbol{B}|\boldsymbol{A}) = \frac{\mathbf{P}(\boldsymbol{A}|\boldsymbol{B})\,\mathbf{P}(\boldsymbol{B})}{\mathbf{P}(\boldsymbol{A})}; \qquad \mathrm{P}(A),\ \mathrm{P}(B) > 0$$

probability of A

- **Random variable:**
  - ´the name of an experiment with a probabilistic outcome´
  - its value is the outcome of the experiment
- **Probability distribution (for the random variable $\boldsymbol{Y}$):**
  - probability $\mathbf{P}(\boldsymbol{Y} = \boldsymbol{y_i})$, that $\boldsymbol{Y}$ will take on the value $\boldsymbol{y_i}$
- **Expected value (~mean) of a random variable $\boldsymbol{Y}$:**

$$\boldsymbol{\mu_Y} = \mathbf{E}(\boldsymbol{Y}) = \sum_{\boldsymbol{i}} \boldsymbol{y_i}\ \boldsymbol{P}(\boldsymbol{Y} = \boldsymbol{y_i})$$

Probability – Basic Notions (4)

- **Variance (of a random variable):**

$$VAR(Y) = E[(Y - \mu_Y)^2]$$

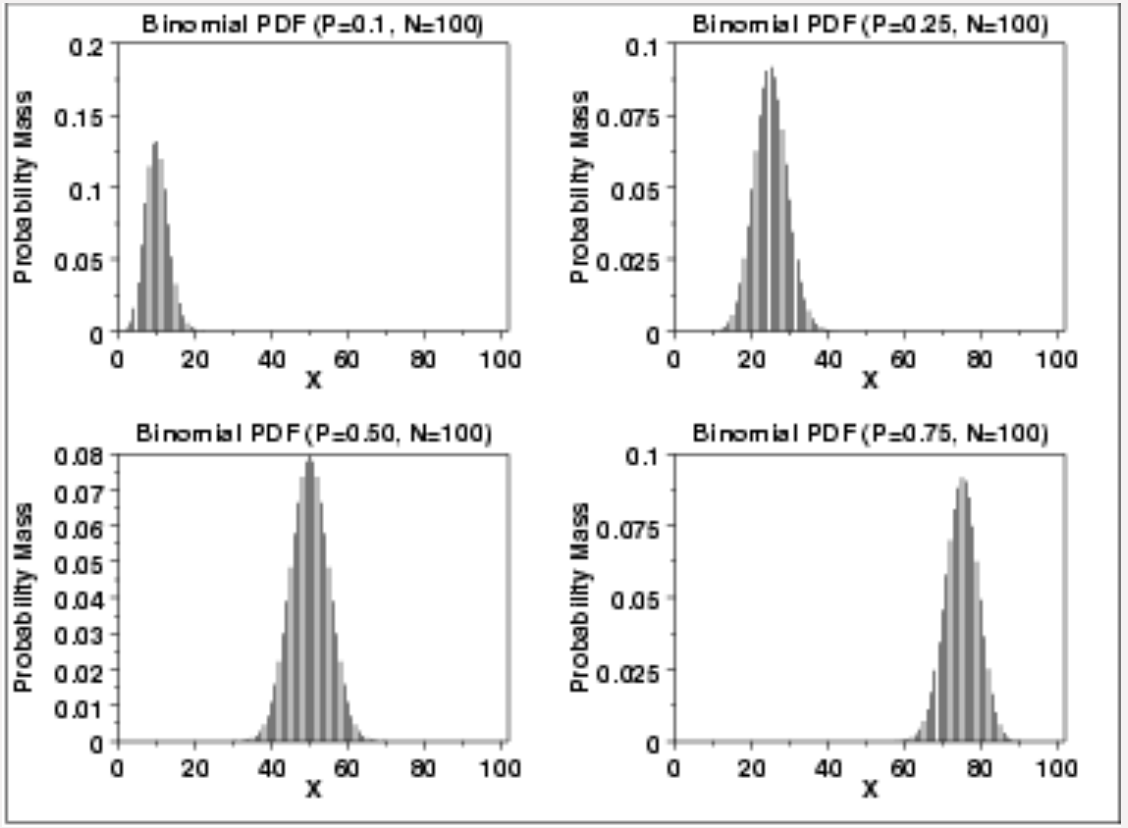
  - Characterizes the width (dispersion) of the distribution around its mean
- **Standard deviation of *Y***: $\sigma_y = \sqrt{VAR(Y)}$
- **Binomial distribution**
  - The probability of observing $\boldsymbol{r}$ ´heads´ in a series of $\boldsymbol{n}$ independent coin tosses
  - The probability of ´heads´ in a single toss is $\boldsymbol{p}$

Probability – Basic Notions (5)

Binomial distribution

- The probability of observing r ´heads´ in a series of n independent coin tosses
- The probability of ´heads´ in a single toss is p
- **Probability function** (probability that X will take on the value r)**:**

$$\mathbf{P}(r) = \frac{n!}{r!(n-r)!}\, p^r (1-p)^{n-r}$$



Probability – Basic Notions (6)

- Expected (mean) value of $\boldsymbol{X}$: $\mathbf{E}[\boldsymbol{X}] = \boldsymbol{np}$
- Variance: $\boldsymbol{VAR(X) = n\,p\,(1-p)}$
- Standard deviation: $\boldsymbol{\sigma_X = \sqrt{np(1-p)}}$
- For sufficiently large values of $\boldsymbol{n}$ the binomial distribution is closely approximated by a normal distribution with the same mean and variance
- **Recommendation:** use the normal approximation only when:

  $\boldsymbol{np(1-p) \geq 5}$ (i.e., for $\boldsymbol{n \geq \frac{5}{p(1-p)}}$)

Probability – Basic Notions (7)

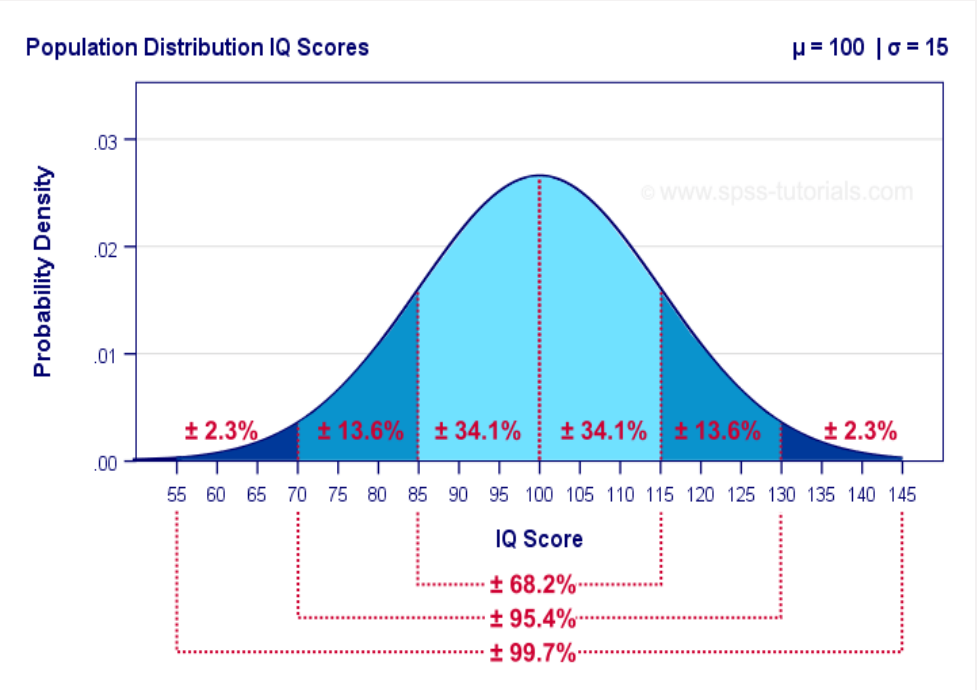
Normal distribution

- also called **Gaussian distribution**
- **Normal probability density function**

$$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

Probability that the value of the random variable X will fall into the interval (a, b):

$$\int_a^b p(x)dx$$



Probability – Basic Notions (8)

Normal distribution

- Suitable for a large number of natural phenomena
- Expected (mean) value of $\boldsymbol{X}$: $\mathbf{E}[\boldsymbol{X}] = \boldsymbol{\mu}$
- Variance: $\boldsymbol{VAR(X) = \sigma^2}$
- Standard deviation: $\boldsymbol{\sigma_X = \sigma}$
- **Central limit theorem:**

  ´The distribution of the mean of a large number of independent random variables of the same distribution approximates the normal distribution.´

Probability – Basic Notions (9)

- **Estimator** ~ random variable $\boldsymbol{Y}$
  - Used to estimate the parameter $\boldsymbol{p}$ from the tested population
- **Estimation bias** of $\boldsymbol{Y}$ for $\boldsymbol{p}$: $\mathbf{E}[\boldsymbol{Y}] - \boldsymbol{p}$
  - ´unbiased´ estimator for $\boldsymbol{p}$: $\mathbf{E}[\boldsymbol{Y}] = \boldsymbol{p}$
- $\boldsymbol{N\%}$ **confidence interval** for the parameter $\boldsymbol{p}$
  - Interval that contains $\boldsymbol{p}$ with probability $\boldsymbol{N\%}$
- **Test** ~ procedure deciding on the correctness of a statistical hypothesis $\boldsymbol{H}$
  - **Significance level** $\boldsymbol{\alpha}$ corresponds to the probability of rejecting the true hypothesis → usually set to $\boldsymbol{\alpha = 0.05}$

Hypotheses Testing (1)

1. Given the observed accuracy of a hypothesis over a limited sample of data
   → how well does this estimate its accuracy over additional examples?

2. Given that one hypothesis outperforms another over some sample of data
   → how probable is it that this hypothesis is more accurate in general?

3. When the data is limited → how to best use this data to both learn a hypothesis and estimate its accuracy as well as to compare the performance of two learning algorithms?

→ **limit the difference between the accuracy observed on the given data and the actual accuracy of the whole data distribution**

Hypotheses Testing (2)

Aim: 1) Understand whether to use the hypothesis or not

2) Evaluating hypotheses represents an integral component of many learning methods (e.g., when post-pruning decision trees to avoid overfitting)

Estimate future accuracy of a hypothesis given only a limited set of data:

- **Bias in the estimate**: over-training × unbiased estimate of future accuracy (mutually independent training and test sets)
- **Variance in the estimate**: the measured accuracy can vary from the true accuracy; bigger variance for fewer test examples

Hypotheses Testing (3)

Estimating hypothesis accuracy

- Space of possible instances $\boldsymbol{X}$, e.g., the set of all people
- Various target functions may be defined over $\boldsymbol{X}$, $\boldsymbol{f : X \rightarrow \{0, 1\}}$, e.g., people who plan to purchase new skis this year
- Different instances $\boldsymbol{x \in X}$ may be encountered with different frequencies, e.g., probability that $\boldsymbol{x}$ arrives at the ski resort
  - $\boldsymbol{D}$ … probability of encountering the instances in $\boldsymbol{X}$

Hypotheses Testing (4)

Task: learn the target function f from the space H of possible hypotheses

- provided are training examples $\boldsymbol{x}$, along with their correct target value $\boldsymbol{f(x)}$, drawn randomly from $\boldsymbol{X}$ according to the distribution $\boldsymbol{D}$

Questions:

Given a hypothesis $\boldsymbol{h}$ and a data sample containing $\boldsymbol{n}$ examples drawn at random according to the distribution $\boldsymbol{D}$:

1. **What is the best estimate of the accuracy of $\boldsymbol{h}$ over future instances drawn from the same distribution?**
2. **What is the probable error in this accuracy estimate?**

Hypotheses Testing (5)

The sample error on the training set $S \subset X$

~ the fraction of S, misclassified by h

$$ERROR_S(h) \equiv \frac{1}{n} \sum_{x \in S} \delta\big(f(x), h(x)\big)$$

- n … the number of examples in S
- $\delta(f(x), h(x)) = 1$ for $f(x) \neq h(x)$
- $\delta(f(x), h(x)) = 0$ for $f(x) = h(x)$
- Binomial distribution $ERROR_S(h)$: $ERROR_S(h) = \frac{r}{n}$
  - r … the number of examples from S, that were misclassified by h

Hypotheses Testing (6)

The true error of hypothesis h

~ probability of misclassification for an instance $x \in X$ drawn at random according to D : $ERROR_D(h) \equiv \Pr_{x \in D} [f(x) \neq h(x)]$

- Binomial distribution: $ERROR_D(h) = p$ $(= \frac{r}{n}$ … estimate for $p)$
  - p … probability of misclassifying a single instance drawn from D
  - unbiased estimator $ERROR_D(h)$ $(\sim p = r/n)$
    - The hypothesis h and the sample set S must be chosen independently.
    - The sample set S contains n (≥ 30) examples drawn at random from X according to the probability distribution D

Hypotheses Testing (7)

Estimator variance

- An unbiased estimator with the least variance would yield the smallest expected squared error between the estimate and the true value of the parameter
- Given no other information, the most probable value of $\boldsymbol{ERROR_D(h)}$ is $\boldsymbol{ERROR_S(h)}$
- With approximately $\boldsymbol{95\%}$ probability, the true error $\boldsymbol{ERROR_D(h)}$ lies in the interval

$$\boldsymbol{ERROR_S(h) \pm 1.96 \sqrt{\frac{ERROR_S(h) \cdot (1 - ERROR_S(h))}{n}}}$$

→ for approximately 95% of experiments, the calculated interval will contain the true error value

Hypotheses Testing (8)

The expression for general ($N\%$) confidence intervals – constant z_N :

$$ERROR_S(h) \pm z_N \sqrt{\frac{ERROR_S(h) \cdot \left(1 - ERROR_S(h)\right)}{n}}$$

	The values of z_N for two-sided $N\%$ confidence intervals						
$N\%$	50%	68%	80%	90%	95%	98%	99%
z_N	0.67	1.00	1.28	1.64	1.96	2.33	2.58

- Wider intervals for a higher probability
- Good approximation for $n \geq 30$, resp. $n \cdot ERRORS(h)\,(1 - ERRORS\,(h)) \geq 5$

Hypotheses Testing (9)

General approach to derive the confidence intervals:

1. **Identify the underlying population parameter** $\boldsymbol{p}$ to be estimated, e.g., $\boldsymbol{ERROR_{D(h)}}$.
2. **Define the estimator** $\boldsymbol{Y}$ (e.g., $\boldsymbol{ERROR_S(h)}$).
   – choose a minimum-variance, unbiased estimator
3. **Determine the probability distribution** $\boldsymbol{D_Y}$ that governs the estimator $\boldsymbol{Y}$ including its mean and variance.
4. **Determine the** $\boldsymbol{N\%}$ **confidence interval**
   – find the thresholds $\boldsymbol{L}$ and $\boldsymbol{U}$ such that $\boldsymbol{N\%}$ of the mass in the probability distribution $\boldsymbol{D_Y}$ falls between $\boldsymbol{L}$ a $\boldsymbol{U}$.

Hypotheses Testing (10)

Difference in error of two hypotheses D (with a discrete-valued target function):

- Hypothesis h_1 has been tested on a sample S_1 containing n_1 randomly drawn examples
- Hypothesis h_2 has been tested on an independent sample S_2 containing n_2 examples drawn from the same distribution
- **We want to estimate the difference d between the true errors of these two hypotheses:**

$$d = ERRORD(h_1) - ERRORD(h_2)$$

Hypotheses Testing (11)

→ Estimator $\hat{d}$: $\hat{d} \equiv ERROR_{S_1}(h_1) - ERROR_{S_2}(h_2)$

- $\hat{d}$ (~ difference between sample errors) yields an unbiased estimate of *d*
- Normal distribution with the mean $\mathrm{E}[\hat{d}] = d$ and variance $\sigma_{\hat{d}}^2$

$$\sigma_{\hat{d}}^2 \approx \frac{ERROR_{S_1}(h_1) \cdot \left(1 - ERROR_{S_1}(h_1)\right)}{n_1} + \frac{ERROR_{S_2}(h_2) \cdot \left(1 - ERROR_{S_2}(h_2)\right)}{n_2}$$

- ***N%*** confidence interval:

$$\hat{\boldsymbol{d}} \pm \boldsymbol{z_N} \sqrt{\frac{\boldsymbol{ERROR_{S_1}(h_1) \cdot (1 - ERROR_{S_1}(h_1))}}{\boldsymbol{n_1}} + \frac{\boldsymbol{ERROR_{S_2}(h_2) \cdot (1 - ERROR_{S_2}(h_2))}}{\boldsymbol{n_2}}}$$

Hypotheses Testing (12)

Comparing the learning algorithms:

- test for comparing the learning algorithms L_A and L_B
- statistical significance of the observed difference between the algorithms

→ **determine, which of the learning methods, L_A and L_B, is better for learning the target function f**

- Consider the relative performance of the two algorithms averaged over all the training sets of size n that might be drawn from the distribution D

Hypotheses Testing (13)

Comparing learning algorithms:

→ **estimate the expected value of the difference in the errors**

$$\mathop{\mathrm{E}}_{S \subset D}\left[ERROR_D\left(L_A(S)\right) - ERROR_D\left(L_B(S)\right)\right]$$

- $L(S)$ … hypothesis obtained by the learning algorithm $\boldsymbol{L}$ on the training set $\boldsymbol{S}$
- $S \subset D$ … the expected value is taken over the samples S drawn according to the underlying instance distribution $\boldsymbol{D}$

→ **in practice, just a limited number of training data $\boldsymbol{D_0}$ is available to compare the considered learning algorithms**

Hypotheses Testing (14)

- Divide the set $\boldsymbol{D_0}$ into a training set $\boldsymbol{S_0}$ and a disjoint test set $\boldsymbol{T_0}$
  - Training data are used to train both $\boldsymbol{L_A}$ and $\boldsymbol{L_B}$
  - Test data are used to compare the accuracy of the two learned hypotheses:

$$\boldsymbol{ERROR_{T_0}(L_A(S_0)) - ERROR_{T_0}(L_B(S_0))}$$

  - $\boldsymbol{ERROR_{T_0}(h)}$ approximates the true error $\boldsymbol{ERROR_D(h)}$
  - The difference in errors is measured only for the training set $\boldsymbol{S_0}$ (rather than taking the expected value of this difference over all samples $\boldsymbol{S}$ that might be drawn from the distribution $\boldsymbol{D}$)

k-Fold Cross Validation (1)

1. Partition the available data $\boldsymbol{D_0}$ into $\boldsymbol{k}$ disjoint subsets $\boldsymbol{T_1, T_2, \ldots, T_k}$ of equal size ($\geq \boldsymbol{30}$).

2. **FOR $\boldsymbol{i := 1}$ TO $\boldsymbol{k}$ DO**

   use $\boldsymbol{T_i}$ for the test set, and the remaining data to build the training set $\boldsymbol{S_i}$

   $\boldsymbol{S_i \leftarrow D_0 \backslash T_i}$

   $\boldsymbol{h_A \leftarrow L_A(S_i)}$

   $\boldsymbol{h_B \leftarrow L_B(S_i)}$

   $\boldsymbol{\delta_i \leftarrow ERROR_{T_i}(h_A) - ERROR_{T_i}(h_B)}$

3. Return the value $\overline{\boldsymbol{\delta}}$, where $\boldsymbol{\overline{\delta} = \frac{1}{k} \sum_{i=1}^{k} \delta_i}$

k-Fold Cross Validation (2)

- $\boldsymbol{N}$ **%** - confidence interval: $\overline{\boldsymbol{\delta}} \pm \boldsymbol{t_{N,k-1}} \frac{\boldsymbol{\sigma_{\bar{\delta}}}}{\sqrt{\boldsymbol{k}}}$
  - $\boldsymbol{\sigma}_{\bar{\delta}}$ … estimate of the standard deviation: $\boldsymbol{\sigma_{\bar{\delta}}} \equiv \sqrt{\frac{1}{k-1} \cdot \sum_{i=1}^{k} \left(\boldsymbol{\delta_i} - \bar{\boldsymbol{\delta}}\right)^2}$
  - $\boldsymbol{t_{N,k-1}}$ … constant (values of $\boldsymbol{t_{N,\nu}}$ for two-sided confidence intervals approach the values of $\boldsymbol{z_N}$ with $\boldsymbol{\nu} \to \infty$)
  - $\boldsymbol{N}$ …… the desired confidence level
  - $\boldsymbol{\nu}$ ……. Nr. of degrees of freedom (nr. of independent random events that influence the value of $\bar{\delta}$; $\nu = k - 1$ in the current setting)

k-Fold Cross Validation (3)

Values of $t_{N,\nu}$ for two-sided confidence intervals

	Confidence level N			
	90%	95%	98%	99%
$\nu = 2$	2.92	4.30	6.96	9.92
$\nu = 4$	2.13	2.78	3.75	4.60
$\nu = 5$	2.02	2.57	3.36	4.03
$\nu = 9$	1.83	2.26	2.82	3.25
$\nu = 10$	1.81	2.23	2.76	3.17
$\nu = 20$	1.72	2.09	2.53	2.84
$\nu = 30$	1.70	2.04	2.46	2.75
$\nu = 120$	1.66	1.98	2.36	2.62
$\nu = \infty$	1.64	1.96	2.33	2.58

N … desired confidence level

ν …. Nr. of degrees of freedom

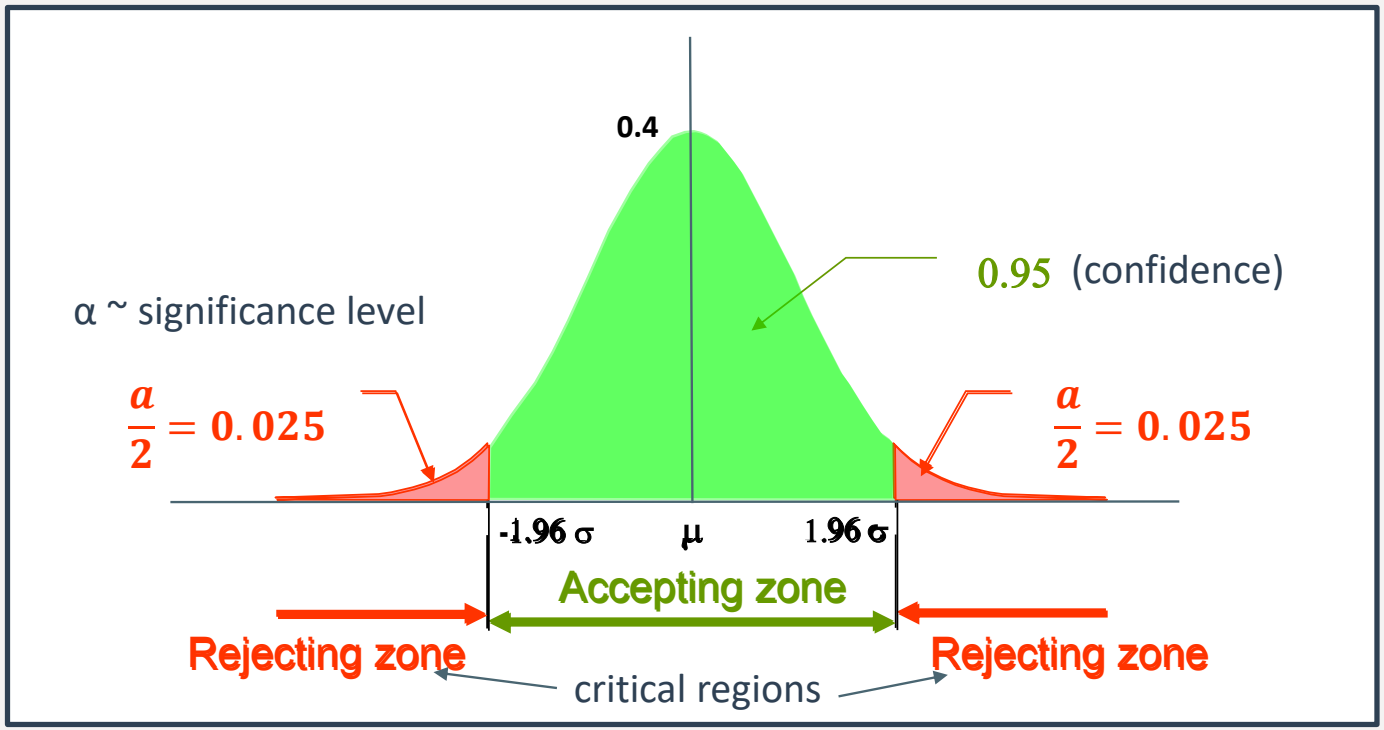
k-Fold Cross Validation (4)

- **Testing has to be done on identical test sets!**
  - in contrast to comparing hypotheses that requires independent test sets

→ **Paired tests**
  - typically produce tighter confidence intervals because any differences in observed errors are due to differences between the hypotheses and not due to differences in the makeup of the sampled data

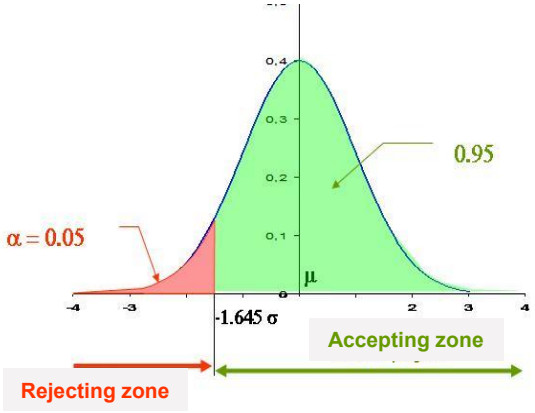
A Two-Sided Test

(looks for the change in the estimated parameter and has two critical regions)

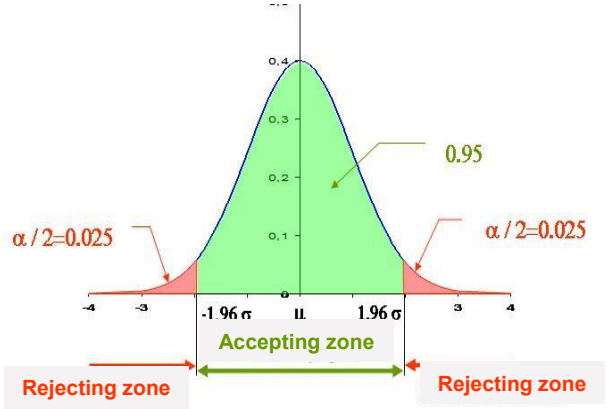


A One-Sided × a Two-Sided Test

A one-sided test



A two-sided test



A one-sided test specifies the direction of change in the estimated parameter (e.g., “precision is higher than”) and has only one critical region.

Neural Networks:

Contents:

- **Introduction to the Field**
- **Perceptron and Linear Separability**

Contents: