Baština Akademije nauka i umjetnosti Bosne i Hercegovine

Proceedings of the Conference on March 14 - International Day of Mathematics

Vuković, Mirjana, urednik; Nurkanović, Mehmed, urednik

2024-12-26

Academy of Sciences and Arts of Bosnia and Herzegovina

https://bastina.anubih.ba/handle/123456789/798

Preuzeto s Baštine Akademije nauka i umjetnosti Bosne i Hercegovine

https://bastina.anubih.ba/

PROCEEDINGS OF THE CONFERENCE ON MARCH 14 – INTERNATIONAL DAY OF MATHEMATICS
Special Editions ANUBiH, Book CCXVI, OPMN Book 30
Sarajevo 2024, **p. 100–113** DOI: 10.5644/PI2024.216.12

THE MATHEMATICS OF ARTIFICIAL INTELLIGENCE

ERVIN MACIĆ, TARIK HUBANA AND MIGDAT HODŽIĆ

Dedicated to the 75th birthday of our dear Professor Mirjana Vuković

ABSTRACT. Although artificial intelligence (AI) is often perceived as the field of computer science that experienced the greatest development, without the wide scope of mathematical fields and methods, AI would not be possible. Mathematical areas in AI include standard methods of analysis (continuous or discrete), set theory and various logic's (propositional, first-order logic, ineffable logic), statistics, probability theory and random processes, vector theory and linear algebra, matrix, optimization, estimation theory and filtering, harmonic analysis, information theory, entropy analysis, graph theory, search methods and other related fields. All the mentioned areas and methods are intertwined in many ways, and specific applications and research determine which specific methods and areas are used. In this sense, this paper provides an overview of a wide range of mathematical fields, methods and concrete applications in a comprehensive manner. Therefore, this paper contributes to the existing knowledge base by summarizing the main mathematical areas, methods and applications in AI, providing mathematicians and artificial intelligence engineers with a basis for further research.

1. INTRODUCTION

Artificial intelligence (AI) is a branch of computing that deals with the development of computer systems capable of performing tasks and solving problems that typically require human intelligence. These include abilities such as learning, reasoning, pattern recognition, decision making and communicating with people. Mathematics forms an integral foundation of AI by providing the key tools and concepts needed to understand, model and solve complex problems. Mathematical methods play a key role in the development of algorithms and models that enable computers to learn, reason, make decisions and work with data. In addition, statistical methods in mathematics are essential for analyzing data, studying patterns and drawing conclusions based on them. Logical formalization enables precise representation of knowledge, drawing conclusions and making decisions in AI systems. Geometry and topology find applications in areas such as computer vision, where they are used to analyze shapes and spatial relationships in images and 3D models. Information theory, another branch of mathematics, plays a key role in signal processing, data compression, and the analysis of complex systems.

2020 *Mathematics Subject Classification.* 68T01.

Key words and phrases. Artificial intelligence, mathematical methods, machine learning, deep learning, large language models.

Because of this key role of mathematics in the large scope of the field of artificial intelligence, the field of research is very active. The greatest achievements and improvements in the field of artificial intelligence were actually inspired by research works. For example, the current interest and advancement of large language models (the model behind the popular application ChatGPT [1]) is a direct consequence of the popular research paper "Attention is all that is needed" [2] where a new encoder-decoder configuration is proposed. A similar effect was achieved by the research that proposed ADAM, a gradient optimization method for stochastic objective functions based on adaptive estimates of lower order moments [3]. The introduction of the perceptron model, which represents a simple mathematical model of a neural network [4] and the backpropagation algorithm [5] provided the basis for the further development of neural network models leading to more complex mathematical models and network structures such as recurrent neural networks that can analyze the flow of information between layers as a two-way neural network [6], deep learning networks [7], Long Short-Term Memory (LSTM) networks [8] and models such as Support vector machine (SVM) models [9] and Random forest (RF) models [10]. Therefore, it is clear that mathematics is essential to the understanding, development and application of algorithms and models in various areas of artificial intelligence, providing the foundation for progress in this rapidly growing field.

2. INTELLIGENT AGENTS

2.1. **What is an intelligent agent?**

An agent can be considered anything that can perceive its environment through sensors and act in that environment. On the other hand, intelligent agents usually represent software that has the ability to perform a task flexibly, independently and without user intervention, and informs the end user about the completion of the task or the very occurrence of the expected event. The agent itself interacts with the environment in order to perform the set task as precisely as possible. Intelligence in this context can be seen as the agent's ability to accept given goals, and the way in which they execute them. Intelligence reflects the level of quality of thinking and learned behavior [11]. A simple agent can be mathematically defined with an agent function that maps each possible sequence of perception to a possible action that the agent can perform or to a coefficient, feedback, function or constant that affects the eventual action [11]:

$$f : P^* \to A. \tag{2.1}$$

The function of an agent is an abstract concept that includes various decision-making principles such as the calculation of individual possibilities, deductions beyond logical rules, and the like, so there is a whole range of diverse agents. However, what they all have in common is that they can improve their performance through learning.

2.2. **Problem solving by searching**

When agents are faced with uncertainty regarding a certain action, it is necessary to carry out planning in advance, and determine the sequence of actions that form a road map to the desired state. This group of agents are called problem-solving agents, and the

problem-solving process is called searching. Currently, there are a number of search algorithms, such as best-first search, breadth-first search, uniform-cost search, depth-first search, depth-first iterative search, and bidirectional search, all of which attempt to find solutions in environments that are episodic, contain one agent, observable, deterministic, static, discrete and fully known. These algorithms make a trade-off between search time, memory usage and solution quality. The search is preceded by a well-defined problem formulation, consisting of an initial state, a set of actions, a transition model, target states, and an action cost function. Search algorithms traverse state space graphs, treating states and actions atomically. Evaluation criteria for search algorithms include completeness, cost optimality, time and space complexity. Uninformed methods, such as breadth-first search and depth-first search, operate solely on problem definitions. Informed methods, such as greedy best-first search, A* search, two-way A* search, IDA*, RBFS (recursive first-best search), SMA* (simplified memory bound A* search), and weighted A* search use heuristics functions for estimating solution costs. The quality of a heuristic search depends on the accuracy of the heuristic, often improved by problem relaxation, pre-calculated solution costs, landmark identification, or learning from problem experience. One of the most frequently used algorithms in the field of artificial intelligence is the A* search, which mathematically determines its main loop in each iteration where paths of the weighted graph extend starting from the initial node. This is done on the basis of the journey cost and the assessment of the costs required to extend the journey to the destination. Mathematically, A* chooses the path that minimizes the following expression [11]:

$$f(n) = g(n) + h(n) \tag{2.2}$$

where n is the next node on the path, $g(n)$ is the cost of the path from the starting node to n, and $h(n)$ is a heuristic function that estimates the cost of the cheapest path from n to the goal.

2.3. **Searching in complex spaces**

In contrast to the search algorithms presented in the previous chapter, search in complex situations explores the relaxation of constraints in search problems, extending beyond fully observable, deterministic, and static environments. In this case, it is necessary to deal with the task to find optimal states without considering the path to them, covering discrete and continuous states. Local search methods such as the peak-climbing algorithm, simulated annealing, and evolutionary algorithms are aimed at direct state optimization, which is useful for both discrete and continuous problems, and offer efficient solutions to optimization tasks with appropriate parameterization. Linear programming and convex optimization excel in certain forms of state space, while evolutionary algorithms maintain a population of states, using mutation and crossover to generate states. In nondeterministic environments, the AND–OR search facilitates contingency planning. When the environment is partially observable, the belief state method represents the set of possible states the agent could be in, where standard search algorithms can be used for sensorless problems, while the belief state AND-OR search can generally solve partially observable problems [11]. Often the objective functions are expressed in mathematical form in such a way that it is possible to use calculus to

solve the problem analytically rather than empirically. There are a number of methods that use gradient environments to find the maximum. However, for many problems, the most efficient algorithm is the Newton-Raphson method, which represents a general technique for finding the roots of a function, i.e. solving an expression in the form $g(x) = 0$. The method works by computing a new estimate for the root using Newton's formula [11]:

$$x \leftarrow x - \frac{g(x)}{g'(x)} \tag{2.3}$$

In order to find the maximum, it is necessary to find such x for which the gradient is a zero vector (ie $\nabla \mathrm{f}(x) = 0$). Therefore, $g(x)$ in the previous formula becomes $\nabla \mathrm{f}(x)$, and the updated equation can be written in matrix form as:

$$x \leftarrow x - H_f^{-1}(x)\nabla \mathrm{f}(x) \tag{2.4}$$

where $H_f(x)$ is the Hessian matrix of second derivatives, whose elements H_{i_j} are given as $\partial^2 f / \partial x_i \partial x_j$.

2.4. **Adversial search and games**

When the agent environment becomes competitive, adversarial search problems arise where multiple agents pursue conflicting goals. Therefore, as a solution for an optimal result, games that represent practical strategies and behavior of agents in adversarial environments are imposed. Key concepts include defining a game in terms of initial state, legal actions, outcomes of actions, terminal conditions, and utility function. Well-known methods include the minimax algorithm, the Alpha-beta algorithm and the Monte Carlo algorithm. Also, many programs use pre-calculated tables of moves that help decisions in certain games. For games based on chance, the Expectiminimax algorithm is used. Using these methods, artificial intelligence has triumphed in games like chess and GO, while humans still maintain superiority in some imperfect information games. In video games, AI competes effectively, using rapid decision-making capabilities [11]. In the context of artificial intelligence, a commonly used method is Monte Carlo Tree Search (MCTS) which is an algorithm that searches the state space and makes statistical evidence of the decisions available in the corresponding states. Since it can be modeled as a Markov decision process, the process is modeled as an ordered sequence (S, A_s, P_a, R_a), where [12]:

S– set of states that are possible in the environment (state space).
A_s -set of actions available in state s.
$P_a(s, s\prime)$ – transition function modeled as the probability that action a taken in state s will lead to state s'.
$P_a(s, s\prime)$ – transition function modeled as the probability that action a taken in state s will lead to state s'.
$R_a(s)$– current reward for reaching state s through action a.

MCTS represents an algorithm that is time-limited, so it can be stopped at any time while finding the best action (decision) at that moment using the following formula [12]:

$$a^* = argmax_{a \in A(s)} Q(s, a) \tag{2.5}$$

where A(s) is the set of actions available in state s in which a decision needs to be made, and Q(s,a) is the empirical average result of executing action a in state s.

2.5. **Constraint Satisfaction Problems**

Constraint satisfaction problems (CSP) represent a state by variable/value pairs and represent the conditions for a solution by setting constraints on the variables. The backward search algorithm, often used for CSPs, uses minimum residual, degree, and least limiting value heuristics to guide variable selection and value assignment. A conflict-oriented backtracking algorithm goes back to the source of the conflict, while a constraint learning algorithm records the conflicts it encounters to prevent them from recurring. Local search, especially using the minimum conflict heuristic, has been shown to be successful for CSPs. The complexity of solving CSPs is strongly related to the structure of the constraint graph, so reduction techniques such as cut conditioning and tree decomposition are often applied, which enable a better ratio of memory consumption and algorithm execution time.

3. KNOWLEDGE AND REASONING

3.1. **Logical agents**

Understanding logical agents is essential for a deeper understanding in the field of artificial intelligence. Logical agents are computer programs or systems that use logic to make decisions in their environment. They use formal logical languages and reasoning rules to process information and draw conclusions. After that, based on these conclusions, they make certain actions or decisions. Logical agents can be applied in various areas such as recommendation systems, process management, diagnostics, planning and many other areas where it is necessary to make decisions based on available information. One of the main advantages of logical agents is their ability to make clear and transparent reasoning in contrast to non-transparent neural networks. These agents use formal logical rules, which allow humans to understand and verify the way the agent makes decisions. Despite their advantages, logical agents face challenges in situations where the environment is dynamic or when it is necessary to handle large amounts of vague or imprecise information. Also, the complexity of the problem can limit the application of pure logical rules. Further development of logical agents includes research into advanced inference techniques, such as probabilistic logical reasoning or combining logic with machine learning techniques to improve their adaptability and robustness in different environments.

3.2. **First-order logic**

First-order logic, also known as predicate logic, is a formal system for representing and reasoning about the relationships and properties of objects in the world. This type of logic makes it possible to precisely express statements about individual elements, their relationships and quantification. The basic elements of first-order logic include objects, relations (or predicates) that describe relationships between objects, functions that map objects to other objects, quantifiers that denote general statements about sets of objects,

and logical conjunctions that connect statements. Expressions in first-order logic consist of terms that represent individual objects or functions of objects, atoms that represent basic propositions about relationships between objects, and complex expressions that consist of atoms, logical conjunctions, and quantifiers. An example of the use of quantifiers in the logic of the first order "All kings are persons": $\forall x\ King(x) \Rightarrow Person(x)$.

3.3. **Inference in first-order logic**

First-order logic uses rules of inference, such as Modus Ponens or rules of inference by generalization, to derive new conclusions from existing propositions. First-order logic is widely used in artificial intelligence for formally expressing knowledge about a problem domain, making inferences, and solving problems such as diagnostics, planning, and reasoning about agent behavior. Although first-order logic is a powerful tool, it faces challenges in efficiently handling large amounts of information and complex problem domains. Also, it is sometimes necessary to extend first-order logic to deal with uncertainty or temporal aspects in problems.

3.4. **Knowledge representation**

Knowledge representation plays a key role in artificial intelligence because it allows computers to represent, manipulate, and understand the information needed to make decisions and solve problems. Knowledge representation is the process of converting information from the real world into a form that computer systems can understand and process. This includes the identification of essential entities, relationships and properties in the problem domain and their formal expression. There are different approaches to representing knowledge, including declarative (through statements and facts), procedural (through algorithms and procedures), and example-based (through sets of examples or patterns). In AI, formal languages such as first-order logic, ontology's, knowledge graphs, or rule-based languages are often used to precisely describe the problem domain and knowledge about it. Ontology's are formal models that describe concepts in a certain area, their properties and relationships between them. They enable the precise definition of conceptual structures and common understanding of information within a domain. As an example, knowledge graphs represent knowledge in the form of a graph where objects or subjects are represented by nodes, and the relationships between them are shown by branches. This structure enables an intuitive understanding of the relationships and connections between different objects.

4. UNCERTAIN KNOWLEDGE AND REASONING

4.1. **Quantitfying uncertainty**

Uncertainty measurement refers to the process of assessing and managing the degree of uncertainty or imprecision in the information we possess. It is used to quantify and understand the degree of reliability or unreliability in data or claims, which is essential for making informed decisions. Uncertainty measurement methods include different approaches, such as fuzzy logic, possibility theory or Dempster-Shafer theory, which allow modeling and treating different degrees of vagueness or indeterminacy in data.

4.2. **Probabilistic reasoning**

Probabilistic reasoning is the process of making inferences or predictions based on the likelihood of an event or outcome. This method takes into account not only the available information, but also its probability, which allows computers to make better decisions even in situations where there is uncertainty.

4.3. **Probabilistic reasoning over Time**

Reasoning based on probability and time extends the concept of reasoning based on probability to dynamic situations where events unfold over time. This method makes it possible to model and predict the probability of an event or outcome in the future taking into account both the current state and temporal features and patterns. Applications include time series forecasting, dynamic planning and resource management in time-sensitive environments. Weighing uncertainty and probabilistic reasoning, including the aspect of time, enable computers to make informed decisions even in situations where there is uncertainty or changing conditions, which is key to creating robust and adaptive artificial intelligence systems.

$$f(x) = a_0 + \sum_{n=1}^{\infty} \left(a_n \cos \frac{n\pi x}{L} + b_n \sin \frac{n\pi x}{L} \right) \tag{4.1}$$

$$F(k) = \int f(x)\, e^{-2\pi i k x} dx. \tag{4.2}$$

Using Fourier transforms in time series forecasting allows analysts to better understand periodic patterns and seasonal components in data and to develop efficient models for predicting future time series values. The ways in which the Fourier transform is used are: identification of periodic patterns, noise filtering, time series decomposition, spectral analysis and prediction based on frequency components.

4.4. **Probabilistic based programming**

Probability-based programming (PBP) is a technique used to solve decision-making problems where there are uncertainties and the probabilities of different outcomes are known or can be estimated. Graphical models are models that represent probabilistic dependencies between different variables using graphical structures such as Bayesian Networks or Markov Random Fields. Bayesian networks are based on conditional probabilities. As an example, let's look at the case of lung cancer diagnosis. Based on the established conditional probabilities in relation to the parameters of whether the patient is a smoker, whether he lives in a polluted environment, lung imaging results and symptoms of shortness of breath, the model can determine the probability that the patient really suffers from cancer. Various inference algorithms are used to draw conclusions based on models and data, such as exact inference, Monte Carlo methods, or variational methods.

4.5. **Making simple decisions**

This chapter explores decision-making strategies in situations where there is a clear problem structure and little uncertainty. Utility or value is a measure of utility that quan-

tifies agents' preferences regarding different outcomes. Utility can be defined mathematically as a function that maps different outcomes (that is, states) to real numbers, with higher values reflecting greater utility or satisfaction. Formally: $U(x)$ where x represents the outcome or state. Utility can be a function of a single outcome or an aggregated function that takes into account multiple outcomes. The basic principle of decision-making based on the selection of actions that maximize expected utility. Analogous to expected values, expected utility can be written as:

$$EU(a) = \sum_i P(s_i) \cdot U(s_i) \tag{4.3}$$

where s_i represents possible outcomes (states), and $P(s_i)$ is the probability of a particular outcome, and $U(s_i)$ is the utility or value of that outcome.

Bayesian theory is a theory that integrates Bayes' theorem with the concept of utility to make optimal decisions under uncertainty. Bayes' theorem allows updating the probability of hypotheses based on new evidence or information. If hypothesis H and evidence D are given, the probability of hypothesis H after taking into account evidence D (the so-called Posterior) can be calculated using the Bayesian formula:

$$P(H|D) = \frac{P(D|H) \cdot P(H)}{P(D)} \tag{4.4}$$

where: $P(H|D)$ posterior probability of hypothesis H after seeing evidence D, $P(D|H)$ probability of evidence D assuming that is the hypothesis H true (the so-called Probability of the evidence), $P(H)$ the prior value of the hypothesis H before we took into account the evidence D and $P(D)$ the probability of the evidence D regardless of the hypothesis.

4.6. **Making complex decisions**

Rational decision-making is a decision-making process that takes into account complex aspects of a problem, including multiple interests, multiple variables, and various aspects of uncertainty. There are many approaches to making complex decisions in AI, and some of the most well-known are decision schemes, Markov processes, probabilistic-graphical models, and Bayesian networks. Decision Trees are graphical representations that model the sequence of decisions and their consequences, helping to analyze and make decisions in complex situations. Markov Decision Processes are formal models that describe a sequence of decisions in a dynamic environment with uncertainty, and enable the optimization of long-term decisions.

4.7. **Making complex decisions with multiple agents**

There is a group of problems where different decision-making strategies need to be explored in situations where multiple agents interact with each other and have different goals. Nash equilibrium is a concept from game theory that describes a state in which no agent can increase his utility by changing his strategy, assuming that the other agents keep their strategies. The mathematical significance of the Nash equilibrium lies in its stability and relevance in the study of interactions between different agents or players. Cooperative vs. non-cooperative decision making are different approaches to

decision making in situations where agents may cooperate or compete with each other. Cooperative decision-making typically involves negotiation and resource sharing, while non-cooperative decision-making typically involves strategies to maximize benefits independent of the behavior of other agents.

5. Machine learning

When the agent is a computer, its learning process is called machine learning where the computer receives relevant data, builds a model based on the data, and uses this model both as a hypothesis about the environment and as software that can solve problems.

5.1. **Learning from Examples**

Machine learning can be supervised, where feedback provides accurate answers, or unsupervised, where patterns are inferred from data. Supervised learning includes regression for continuous values and classification for categorical outcomes. On the other hand, decision trees, informed by information gain, effectively represent Boolean functions. Linear regression and logistic regression are widely used models for linear and probabilistic classification. Non parametric models use the data to make each estimate instead of generalizing the data with parameters. Examples of these methods are nearest neighbor, support vector, and kernel methods. Ensemble methods such as bagging and boosting improve model performance by combining weak classifiers with the goal of obtaining a stronger classifier [11]. A very often used method in the field of artificial intelligence is linear regression with several variables. If the constant representing the intersection of the axis in the equation is multiplied by the "dummy" variable $x_{j,0}$ which we define as always equal to the number 1, the expression can be generalized as follows:

$$h_w\left(x_j\right) = \sum_i w_i x_{j,i} \tag{5.1}$$

where x are inputs, y are outputs and w are weight factors that need to be learned in the machine learning process. The best vector of weight factors, w^*, minimizes the squared loss error:

$$w^* = \underset{w}{argmin} \sum_j L_2(y_j, w\, x_j) \tag{5.2}$$

Using gradient descent, the minimum of the loss function will be achieved, and the equation of the updated coefficients will be:

$$w_i \leftarrow w_i + \alpha \sum_j \left(y_j - h_w\left(x_j\right)\right) \times x_{j,i}. \tag{5.3}$$

5.2. **Learning based on probabilistic models**

Statistical learning methods have a wide scope, from simple average calculations to complex models such as Bayesian networks. Bayesian methods use probabilistic inference to update hypotheses, while maximum posterior (MAP) learning selects the most likely hypotheses. The maximum likelihood learning method selects a hypothesis that maximizes the likelihood of the data, while naive Bayesian learning is effectively and

widely used. The expectation maximization (EM) algorithm facilitates learning using hidden variables. Statistical learning is a very lively research field that is advancing both theoretically and practically, with the goal of getting to the point where it is possible to learn any model for which inference is feasible [11]. An effective and widely used probabilistic model in the field of artificial intelligence is the Naive Bayesian model. The model is described by the following equation:

$$P(Cause, Consequence_1, \ldots, Consequence_n) = P(Cause) \prod_i P(Cause_i|Consequence). \tag{5.4}$$

This type of distribution is called naive, because it is usually used in cases where the *Consequence* variables are not strictly independent in relation to the causal variable.

5.3. **Deep learning**

Deep learning is a broad family of machine learning techniques in which hypotheses take the form of a complex algebraic circuit with adjustable connection strengths. The word "deep" refers to the fact that cars are usually organized in multiple layers. The basic model used for deep learning is artificial neural network, while for the minimization of the error function algorithms are used that propagate the error backward using gradient descent in the parameter space. Deep learning is proven to work well for object and speech recognition, and as reinforcement learning in complex environments. Convolutional networks stand out especially for image recognition, while recurrent networks are very efficient for processed strings of values/sequences [11]. The mathematical model of neural networks lies in the background of all described methods, and the basic unit within the network (neuron) can be described by the following equation:

$$a_j = g_j(\sum_i w_{i,j} a_i) \equiv g_j\left(in_j\right) \tag{5.5}$$

Where a_j is the output of unit $j, w_{i,j}$ is the weighting factor of the connection between units i and j, g_j is the nonlinear activation function associated with unit j, and in_j is the weighted sum of the inputs to unit j. The choice of activation functions is also diverse, but the most commonly used are logistic or sigmoid, ReLU (corrective activation function), softplus functions and the hyperbolic tangent [11] function.

5.4. **Reinforcment learning**

In supervised learning, the agent learns passively by observing examples of input and output pairs available to it. For real problems, there is often not enough data to learn from which the agent will be ready for new problems. An alternative to this approach is incentive learning, where an agent interacting with the environment periodically receives a reward/incentive that indicates the agent is on the right track. Agent design dictates the type of information that must be learned, so we distinguish between model-based agents where agents possess or acquire a model for the environment and a global goal, and model-free agents that learn a global goal or policy. Global objectives can be learned using multiple approaches, such as direct estimation, adaptive dynamic programming, temporal difference, Q-learning, deep support learning, reward shaping, and

policy search [11]. The simplest case of supported learning is a fully observable environment with a small number of actions and states, in which the agent already has a fixed policy $\pi(s)$ that determines the actions. The agent in this case tries to learn the global objective function $U^{\pi}(s)$ which represents the expected total reward if the policy π is executed starting in state s. This type of agent is called a passive learning agent. The global objective function can be defined as follows [11]:

$$U^{\pi}(s) = E\left[\sum_{t=0}^{\infty} \gamma^{r} R(S_t, \pi(S_t), S_{t+1})\right] \tag{5.6}$$

where $R(S_t, \pi(S_t), S_{t+1})$ is the reward/incentive received when the action $\pi(S_t)$ is taken in state S_t and reaches state S_{t+1}. Here S_t is a random variable that indicates the state reached at the moment t when policy π is executed, starting from the state $S_0 = s$. Given the nature of supported learning agents, and how environments become more and more complex, the advantages of this approach are likely to become more pronounced [11].

5.5. **Communicating, perceiving and acting**

To make actions in environments, computer systems communicate with users, perceive the environment, and make decisions based on this information. Key concepts in this area include: Understanding users' natural language and being able to respond to queries or commands in a way that is understandable and useful to users. The ability of computer systems to perceive and interpret their environment through sensors, cameras, microphones, and other input devices The process of making decisions based on collected information about the environment and system goals.

5.6. **Natural language processing**

Natural language processing is an area of artificial intelligence that deals with the understanding, generation and interpretation of human language. Text analysis involves understanding the structure and meaning of a text through techniques such as tokenization, lemmatization, entity extraction, and syntactic analysis. Understanding language involves understanding the semantic connections and contexts between words, sentences and text segments. Language generation means creating text outputs that are grammatically correct and meaningful based on entered queries or conditions.

5.7. **Natural Language Processing with deep learning**

When deep learning with neural networks with hundreds or thousands of layers is applied to the field of natural language processing, this technique enables computers to better understand and generate human language. The representation of words that a computer can understand is achieved as a vector representation that enables computers to efficiently process and understand the semantic connections between words. For Large Language Models (LLMs), vectors with multiple components are most often used, that is, members of an n-dimensional continuous vector space. By training these representation vectors over the corpus of words we use, we get that words that appear in

similar contexts within a language tend to be closer to each other. The process of learning vector representation is most often performed using unsupervised learning. Within the space of representations, there is a trained algebraic relationship between words, so that we can add and subtract words like for example

$$king - man + woman = queen. \tag{5.7}$$

Deep neural networks include various architectures such as: recurrent neural networks, transformers, deep convolutional neural networks, autoencoders, gated recurrent units (GRU) and generative adversarial network (GAN) models.

5.8. **Computer Vision**

Computer Vision is a popular subfield of AI or machine learning that deals with the ability of computer systems to analyze, interpret and understand visual information, such as images and videos. The goal is to enable computers to use visual perception in order to solve tasks or make decisions. This area encompasses a variety of tasks including object recognition, face detection, image classification, motion tracking, object segmentation, and more. At the beginning of the 21st century, with the advent of modern and capable graphics cards, deep learning strategies have shown record results in the fields of object detection and image classification. Among the most commonly used strategies in building models capable of real-time object detection from camera videos are deep neural networks. In the field of computer vision, convolutional neural networks are the most represented technique in the development of these models, next to newly created transformers. Convolutional neural networks consist of convolutional layers and pooling layers and can be represented by equations:

$$z_{ij} = (w * x)_{ij} + b \tag{5.8}$$

$$a_{ij} = f\left(z_{ij}\right) \tag{5.9}$$

$$a_{ij} = pooling\left(x_{ij}\right) \tag{5.10}$$

where x,w,b,f,z,a respectively are input data (eg image), convolution filter weights, bias, activation function, convoluted output, activation layer output. The inspiration for using convolution is partly the human brain and the ways in which we recognize the basic shapes of an image (edges, corners, colors), but also the mathematical operation of convolution. It is based on the concept of filtering and processing input data using filters. Let us denote the input data as X where X can be a $2D$ image or a multichannel image tensor ($X \in R^{C\times H\times W}$) where C is the number of channels (eg RGB). A filter is a smaller field or tensor applied to the input data to perform a convolution operation. Let K be the filter (kernel), where K is usually smaller than the input data and has dimension $C \times F \times F$ where F is the size of the filter. The convolution operation in the discrete case can be written as:

$$Y(i,j) = \sum_{m=0}^{F-1}\sum_{m=0}^{F-1} X(i+m, j+n)\cdot K(m,n) \tag{5.11}$$

where Y(i,j) is the map output value at position (i,j).

This form is a special case of the continuous, general form of the convolution of two functions in t:

$$(f * g)(t) = \int_{-\infty}^{\infty} f(\tau)\, g(t-\tau)\, d\tau. \tag{5.12}$$

6. Philosophy, ethics and safety of AI

The rapid development of powerful AI models brings with it risks and ethical questions that have yet to be resolved. Questions like how to ensure an equal set of data that adequately represents the population that will be modeled on, and how to act in situations when the AI model is not safe or the user himself doubts its output. These and many other issues of security and ethics are dealt with by the fields of AI security and AI ethics. The field of AI security is also concerned with discerning why the model produced the appropriate output for a given input. Various methods of interpretation of the structure of the interior of the neural network and approximation of individual parts of the network are used.

7. Future of AI and conclusions

The fusion of mathematics and artificial intelligence (AI) has the power and potential to transform the world and advance many industries. Artificial intelligence is deeply dependent on mathematics and logic, with the ability to process large amounts of data and make complex decisions. The mathematical framework of AI contains the results of linear algebra, statistics, probability theory, Bayes theorem, random processes, optimization, game theory, fractal mathematics, chaos theory, logic, vector and matrix theory, various methods of discrete and continuous mathematics. Together, mathematics and AI are transforming society, industries, education, productivity, and innovation. Mathematics becomes even more important and represents a constant "fuel" for the continuous development of AI. By providing an overview in this paper as a kind of catalog of leading mathematical fields, methods and applications in the field of AI, this paper provides mathematicians and artificial intelligence engineers with a basis for further research.

References

[1] OpenAI, *OpenAI*, OpenAI, [Online]. Available: https://openai.com/. [Last access 15 05 2024].
[2] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser and I. Polosukhin, *Attention is All you Need*, at 31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA, 2017.
[3] D. P. Kingma and J. Ba, *Adam: A Method for Stochastic Optimization*, at 3rd International Conference for Learning Representations, San Diego, 2015.
[4] F. Rosenblatt, *The perceptron: A probabilistic model for information storage and organization in the brain*, Psychological Review, vol 65, no. 6, pp. 386-408, 1958.
[5] D. E. Rumelhart, G. E. Hinton and R. J. Williams, *Learning representations by back-propagating errors*, Nature, tom 323, pp. 533-536, 1986.
[6] M. I. Jordan, *Serial order: a parallel distributed processing approach*, Technical report, 1986.
[7] Y. LeCun, Y. Bengio and G. Hinton, *Deep Learning*, Nature, vol 521, pp. 436-444, 2015
[8] S. Hochreiter and J. Schmidhuber, *Long Short-term Memory*, Neural Computation, vol 9, 1997.

[9] M. A. Hearst, S. T. Dumais, E. Osuna, J. Platt and B. Scholkopf, *Support vector machines*, IEEE Intelligent Systems and their Applications, vol 13, no. 4, pp. 12-28, 1998.
[10] L. Breiman, *Random Forests*, Machine Learning, vol 45, pp. 5-32, 2001.
[11] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, London, Great Britain, Pearson, 2021.
[12] M. Swiechowski, K. Godlewski, B. Sawicki and J. Mańdziuk, *Monte Carlo Tree Search: a review of recent modifications and applications*, Artificial Intelligence Review, vol 56, no. 3, p. 2497–2562, 2023.

(Received: May 17, 2024)
(Revised: September 13, 2024)

Ervin Macić
ARTI Analytics Inc.
European Operations
Marsala Tita 6
71000 Sarajevo
Bosnia and Herzegovina
e-mail: *ervin.macic@artianalytics.com*
and
Tarik Hubana
ARTI Analytics Inc.
European Operations
Marsala Tita 6
71000 Sarajevo
Bosnia and Herzegovina
e-mail: *tarik.hubana@artianalytics.com*
and
Migdat Hodžić
ARTI Analytics Inc.
European Operations
Marsala Tita 6
71000 Sarajevo
Bosnia and Herzegovina
e-mail: *migdat.hodzic@artianalytics.com*