Національний технічний університет України
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

Information systems and technologies

Software tools for designing and implementing neural network systems

Working program of the academic discipline (Syllabus)

Details of the academic discipline	
Level of higher education	**Second (master's)**
Branch of knowledge	12 Information technologies
Specialty	121 Software engineering 123*Computer Engineering* 126 Information systems and technologies
Educational program	
Discipline status	*Study discipline of the students' free choice*
Form of education	*full-time (daytime / remote*
Year of training, semester	*3rd year, spring semester*
Scope of the discipline	*180 hours (36 hours – Lectures, 18 hours – Laboratory, 126 hours – SRS)*
Semester control/ control measures	*Assessment/Assessment work*
Lessons schedule	http://rozklad.kpi.ua/Schedules/ScheduleGroupSelection.aspx
Language of teaching	Ukrainian
Information about the course leader / teachers	Lecturer: candidate of technical sciences, associate professor Volodymyr Mykolayovych Shymkovich, v.shymkovych@kpi.ua, Telegram:@volodymyr_shymkovych Laboratory: candidate of technical sciences, associate professor Volodymyr Mykolayovych Shymkovich, v.shymkovych@kpi.ua, Telegram:@volodymyr_shymkovych
Placement of the course	https://campus.kpi.ua

Program of educational discipline

1. Description of the educational discipline, its purpose, subject of study and learning outcomes

Description of the discipline. When studying this discipline, students will learn the theoretical foundations of neural networks and get initial experience in developing software that implements neural network technologies. Laboratory classes will provide initial experience in creating software systems based on neural networks. Development and research of neural network systems for various purposes will be carried out using the Payton programming language and the TensorFlow and Keras libraries. The course provides quality control of the acquired knowledge in the form of express tasks with the help of software packages of control and modular control works.

The subject of the academic discipline:basic concepts of neural network systems, introduction to neural networks, topologies and types of neural networks, convolutional neural networks, software implementation and research of neural network systems on Payton and TensorFlow and Keras libraries.

Interdisciplinary connections. DisciplineSoftware tools for designing and implementing neural network systems is based on the following disciplines: Programming – 1. Fundamentals of programming; Programming - 2. Data structures and algorithms;Software development technologies.

- **The purpose of the educational discipline.**Training of highly qualified specialists who know the basic concepts of neural network theory, the terms of neural network systems, the structure and properties of neural network systems, methods of learning neural network systems, databases for learning neural network systems, stages of designing software neural network systems, software tools for implementing neural network systems, ways to increase the efficiency of software neural network systems systems

The main tasks of the academic discipline

Knowledge:

- structure and properties of neural networks;
- methods of learning neural networks;
- databases for training neural network systems;
- roles and places of neural network systems in the information technology environment of their use;
- stages of designing software neural network systems;
- software tools for implementing neural network systems;
- methods of increasing the efficiency of software neural network systems;

Skills:

- apply the error backpropagation method for training neural networks;
- apply adaptive learning rate procedures such as AdaGrad, RMSprop, and Adam to backpropagation to train neural networks;
- apply basic building blocks of TensorFlow;
- programmatically implement a neural network in TensorFlow
- programmatically implement a neural network that works well on the MNIST dataset;
- apply screening regularization in TensorFlow;
- apply packet normalization in Tensorflow;
- programmatically implement the network using Keras

2. Pre-requisites and post-requisites of the discipline (place in the structural and logical scheme of training according to the relevant educational program)

Prerequisites:be able to use a computer at the administrator level, be able to work with virtual machines (create, configure, modify), basic knowledge in the field of neural networks, be able to program in the Payton language.

Post-requisites:design and implementation of software using neural network applications.

After completing the course, students will be able to implement software tools with neural networks and apply error backpropagation to train neural networks, apply adaptive learning rate procedures such as AdaGrad, RMSprop, and Adam to backpropagation to train neural networks, apply the basic building blocks of TensorFlow, programmatically implement a neural network in TensorFlow, programmatically implement a neural network that works well on different datasets, create datasets for training a neural network, apply dropout regularization in TensorFlow, apply batch normalization in Tensorflow, programmatically implement a network using Keras.

3. Content of the academic discipline

1. Overview of basic concepts. MNIST data set and linear benchmarking.
2. Gradient descent: full, batch, stochastic. Implementation of gradient descent in the code;
3. Momentum and speed of adaptive learning: using momentum to accelerate learning; Nesterov impulse; impulse in the code; variable and adaptive learning rates; constant learning rate vs. code-in-code RMSP; Adam's optimization; adam in code; offer window;
4. Selection of hyperparameter. Hyperparameter optimization: cross-validation, grid search, and random search; sampling is logarithmic; grid search in the code; changing grid search; random search in the code.
5. Introduction to weight initialization, vanishing and exploding gradients, local versus global minima.
6. Basics of TensorFlow: variables, functions, expressions, optimization; building a neural network in TensorFlow;
7. Acceleration of a software product using a graphics processor. Configuring the GPU on Amazon Web Services, installing deep learning libraries with an accelerated NVIDIA GPU on a home computer. Can Big Data Be Used to Accelerate Backpropagation? How to improve your Theano and Tensorflow skills;
8. Face recognition. Introduction to the face recognition project, description of the face recognition problem. Class-based ANNs in Theano. Class-based ANNs in TensorFlow. Content of the face recognition project.
9. Modern methods of regularization. Screening regularization, screening intuition, noise injection.
10. Normalization of the batch. Exponentially smoothed averages. The theory of party normalization. Batch normalization flow tensor. Theano normalization batch; Noise perspective.
11. Basics of Keras. Keras in code, functional Keras API. How to easily convert Keras to Tensorflow 2.0 code.
12. Basics of PyTorch. Dropping PyTorch; PyTorch batch norm.
13. PyTorch, CNTK and MXNet
14. Deep learning. What is the difference between "neural networks" and "deep learning"?
15. Choice of learning rate and penalty for manual regularization.
16. How to install Numpy, Scipy, Matplotlib, Pandas, IPython, Theano and TensorFlow

Lecture classes

Section 1. General provisions

Chapter 2. Neural networks and their properties. Introduction

Chapter 3. Neural network training methods and datasets for training

Chapter 4. Introduction to software packages for the implementation of ANNs.

Section 5.Theano Basics

Section 6.Basics of TensorFlow

Chapter 7.Basics of Keras

Laboratory classes

1. Analysis of the subject area.
2. Construction of a structural diagram of a neural network system.
3. Building a data modelfor neural network training.
4. Implementation of ANNs using specialized software packages.
5. Implementation of ANN training using specialized software packages.
6. Development of a software product using trained ANN.
7. Preparation of a report on the developed neural network software system.

4. Educational materials and resources

Basic literature

1. Shakhovska N. B. Systems of artificial intelligence: study guide / N.B. Shakhovska, R. M. Kaminsky, O. B. Vovk. – Lviv: Publishing House of Lviv Polytechnic, 2018. – 392 p.
2. Zaichenko Yu.P. Fundamentals of designing intelligent systems. Study guide / Yu.P. Zaichenko. - K.: Slovo, 2004. - 352 p.
3. Kutkovetsky V.Ya. Image recognition: Study guide / V.Ya. Kutkovetskyi. - Mykolaiv: Department of the Moscow State University named after P.Mohyly, 2017. – 420 p.
4. Rutkovskaya D., Pylinsky M., Rutkovsky L. Neural networks, genetic algorithms and fuzzy systems: Trans. from Polish I.D. Rudinsky - M.: Hotline - Telecom, 2007. - 452 p.
5. Deep Learning with TensorFlow 2 and Keras: Regression, ConvNets, GANs, RNNs, NLP, and more with TensorFlow 2 and the Keras API, 2nd EditionPaperback – December 27, 2019. by. Antonio Gulli (Author)

Supporting literature

1. Kravets P.Y., Shimkovich V.N. The method of optimization of weight coefficients of neural networks with the help of a genetic algorithm when implemented on programmable logic integrated circuits / International Scientific and Technical Journal "Electronic Modeling". – 2013. – 35, No. 3. - pp. 65-75.
2. Li, Zewen & Yang, Wenjie & Peng, Shouheng & Liu, Fan. (2020). A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects.
3. Chen, Xiaoxue & Jin, Lianwen & Zhu, Yuanzhi & Luo, Canjie & Wang, Tianwei. (2020). Text Recognition in the Wild: A Survey.
4. Yang YX, Wen C, Xie K, Wen FQ, Sheng GQ, Tang XG. Face Recognition Using the SR-CNN Model. Sensors (Basel). 2018;18(12):4237. Published 2018 Dec 3. doi:10.3390/s18124237
5. Kocić J, Jovičić N, Drndarević V. An End-to-End Deep Neural Network for Autonomous Driving Designed for Embedded Automotive Platforms. Sensors. 2019; 19(9):2064.
6. A. Kumar, S. Verma and H. Mangla, "A Survey of Deep Learning Techniques in Speech Recognition," 2018 International Conference on Advances in Computing, Communication Control and Networking (ICACCCN), Greater Noida (UP), India, 2018, pp. 179-185, doi: 10.1109/ICACCCN.2018.8748399.
7. Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, Illia PolosukhinAttention is All you Need. Part ofAdvances in Neural Information Processing Systems 30 (NIPS 2017)
8. Thierry Bouwmans,Sajid Javed,Soon Ki Jung.Deep neural network concepts for background subtraction: A systematic review and comparative evaluation.Neural Networks.Volume 117, September 2019, Pages 8-66

Educational content

6. Methods of mastering an educational discipline (educational component)

Lecture classes

No. z/p	The name of the topic of the lecture and a list of the main questions (a list of didactic tools, references to the literature and tasks on the SRS)
1	Topic 1.1. Structure and content of the course. RSO Topic 1.2. General concepts of the discipline. Types of artificial neural networks. Lecture 1. Introduction. General concepts. Artificial neuron. Types of artificial neural networks. Structure of the discipline "Software tools for designing and implementing neural network systems", RSO. General concepts of the field of artificial neural networks. Properties of ANNs. The structure and principle of operation of an artificial neuron. Types of artificial neural networks. Literature: [2, Chapter 1], [3, Chapter 1] Tasks on SRS. Recurrent neural networks - general concepts, types, models.
2	Topic 1.3.Overview of basic concepts. Topic 2.1.MNIST data set. Lecture 2.Overview of basic concepts. MNIST data set and linear benchmarking. Overview of the main concepts of building neural network systems. Review and analysis of the databaseMNIST, analysis of training results on this database of different types of ANNs. Literature: [1, Chapter 1], [3, Chapter 1] Tasks on SRS. Databases Iris, CIFAR-10, ImageNet, ADE20K, Coco, Fashion-MNIST, Boston housing - detailed review, comparative analysis of application in different systems
3	Topic 2.2.The momentum and speed of adaptive learning. Lecture 3.The momentum and speed of adaptive learning. Momentum and speed of adaptive learning: using momentum to accelerate learning; Nesterov impulse; impulse in the code; variable and adaptive learning rates; constant learning rate vs. code-in-code RMSP; Literature: [1, Ch. 1.1.2; Ch. 3], [3, Ch. 1] Tasks on SRS.Adam optimization, adam in code, suggestion window.
4	Topic 3.1.Hyperparameter selection Lecture 4.Hyperparameter selection and hyperparameter optimization Hyperparameter selection. Hyperparameter optimization: cross-validation, grid search, and random search; sampling is logarithmic; grid search in the code; changing grid search; random search in the code. Literature: [1, Chapter 5], [3, Chapter 6]
5	Topic 3.2. Learning neural networks. Lecture 5. Methods of learning ANNs.

	Introduction to weight initialization, vanishing and exploding gradients, local versus global minima. Literature: [1, Ch. 14], [3, Ch. 6; Ch. 9], [6, Ch. 2] Tasks for SRS. Comparison of software implementations of learning methods and their speed.
6	Topic 3.3.Theano Basics Lecture 6.Theano Basics. Theano basics: variables, functions, expressions, optimization; building a neural network in Theano; prospects for working with Theano; Literature: [1, Ch. 5; Ch. 6], [3, Ch. 10], [6, Ch. 3; Ch. 4] Tasks for SRS. Overview and analysis of constructed neural network systems in a software packageTheano.
7	Topic 3.4.Basics of TensorFlow. Lecture 7.Basics of TensorFlow. TensorFlow basics: variables, functions, expressions, optimization; building a neural network in TensorFlow; Literature: [1, Chapter 12], [2, Chapter 6], [3, Chapter 9] Tasks for SRS. Overview and analysis of constructed neural network systems in a software packageTensorFlow.
8	Topic 4.1. Implementation of neural network systems on parallel computing systems. Lecture 8. Implementation of neural network systems on parallel computing systems. Acceleration of a software product using a graphics processor. Configuring the GPU on Amazon Web Services, installing deep learning libraries with an accelerated NVIDIA GPU on a home computer. Can Big Data Be Used to Accelerate Backpropagation? Literature: [1, Chapter 1; Chapter 4], [2, Chapter 4; Chapter 5] Tasks for SRS.Acceleration of Theano and Tensorflow packages on GPU;
9	Topic 4.2.Face recognition. Lecture 9.Face recognition. Introduction to the face recognition project, description of the face recognition problem. Class-based ANNs in Theano. Class-based ANNs in TensorFlow. Content of the face recognition project. Tasks for SRS. Review and analysis of software neural network projects for the implementation of facial recognition systems.
10	Topic 5.1.Modern methods of regularization Lecture 10.Modern methods of regularization. Modern methods of regularization. Screening regularization, screening intuition, noise injection.

	Literature: [7, Chapter 2]
11	Topic 5.2.Party normalization. Lecture 11.Party normalization. Exponentially smoothed averages. The theory of party normalization. Batch normalization flow tensor. Theano and TensorFlow batch normalization;Literature: [7, Chapter 4] Tasks for SRS. Independently master the materials onnoise perspectives.
12	Topic 5.3.Basics of Keras. Lecture 12.Basics of Keras. Basics of Keras. Keras in code, functional Keras API. How to easily convert Keras to Tensorflow 2.0 code. Literature: [7, Ch. 2; 3] Tasks for SRS. Overview and analysis of constructed neural network systems in a software packageKeras.
13	Topic 6.1.The basics of PyTorch. Lecture 13.The basics of PyTorch. Dropping PyTorch; PyTorch batch norm. Literature: [1, Ch. 9], [2, L. 9], [3, Ch. 10], [7, Ch. 11] Tasks for SRS. Review and analysis of software products for the implementation of neural network systemsCNTK and MXNet
14	Topic 6.3. Deep neural networks Lecture 14. Deep neural networks and their learning Overview and analysis of deep neural networks. Methods of their training and software applications using deep neural networks. Literature: [1, Ch. 4], [2, L. 13] Tasks for SRS. Rthe difference between "neural networks" and "deep learning"?
15	Lecture 15. Modular control work All previous material is submitted to the control work. Tasks include a theoretical part, a test question and a practical part. Tasks for SRS. Repeat the material of lectures 1-14.
16	Topic 7.1.Manual selection of learning rate and regularization penalty. Lecture 16.Choice of learning speed Manual selection of learning rate and regularization penalty. Literature: [7, Chapter 10]

	Tasks for SRS. Methods of learning neural networks
17	Topic 7.2. Network transformers Lecture 17. Transformer networks Network transformers. The structure of neural networks of transformers, their scope of application, data sets. Review and analysis of software applications with transformer networks. Literature: [7, Chapter 10], [8] Tasks for SRS. An attention mechanism for detecting global dependencies between input data and output data
18	Topic 7.3. Work with software applications Lecture 18. Working with software applications. How to install Numpy, Scipy, Matplotlib, Pandas, IPython, Theano and TensorFlow. Literature: [7, Chapter 8] Tasks for SRS. Self-practice and master the functions of neural network design and implementation software applications

Laboratory classes

No	The name of the laboratory work	Number of aud. hours
1	Laboratory work 1. Parceptron. Write a program that implements a Parceptron neural network and teach it to perform the XOR function. Literature: [3, Chapter 2]	4
2	Laboratory work 2. Implementation of basic architectures of neural networks. Write a program that implements neural networks for modeling a function of two variables. Literature: [3, Chapter 2]	2
3	Laboratory work 3. Neural network of forward propagation for image recognition. Write a program that implements a forward propagation neural network for recognizing handwritten digits. Literature: [3, Chapter 6]	2
4	Laboratory work 4. Convolutional neural networks. Write a program that implements the AlexNet convolutional neural network for recognizing objects from the CIFAR-10 dataset Literature: [7]	2
5	Laboratory work 5. Convolutional neural networks of the typeInception Write a program that implements the Inception V3 convolutional neural network for object recognition in images. Create your own dataset from a folder on the disk, train a neural network on this dataset to recognize the breed of your favorite dog or cat. Save the trained neural network on the computer and write a program that opens and analyzes the image. Literature: [7], [8]	2
6	Laboratory work 6. Convolutional neural networks of the typeXception. Write a program that implements a convolutional neural networkXceptionfor object recognition on video. Create your own dataset from a folder on the disk, train a neural network on this dataset to recognize the logo of your favorite brand, say Apple or BMW. Save the trained neural network on the computer, write a program that opens and analyzes the video, the result is the time at which the logo appeared. Literature: [7]	2
7	Laboratory work 7. LSTM recurrent neural networks Write a program that implements a recurrent neural networkLSTMto recognize the emotional coloring of the text, use the datasetYelp Dataset	2
8	Laboratory work 8. CNN-bi-LSTM neural networks for sound recognition Write a program that implements a CNN-bi-LSTM neural network for speech-to-text recognition.	2

7. Independent work of a student/graduate student

No. z/p	The name of the topic submitted for independent processing	Number of hours of SRS
1	Recurrent neural networks - general concepts, types, models.	6
2	Databases Iris, CIFAR-10, ImageNet, ADE20K, Coco, Fashion-MNIST, Boston housing - detailed review, comparative analysis of application in different systems	6
3	Creating a training database for a neural network.	8
4	Comparison of software implementations of learning methods and their speed.	8
5	Overview and analysis of constructed neural network systems in a software packageTheano.	6
6	Overview and analysis of constructed neural network systems in a software packageTensorFlow.	10
7	GPU acceleration of Theano and Tensorflow packages	6
8	Review and analysis of software neural network projects for the implementation of facial recognition systems.	8
9	Independently master the materials onnoise perspectives.	8
10	Overview and analysis of constructed neural network systems in a software packageKeras.	8
11	Review and analysis of software products for the implementation of neural network systemsCNTK and MXNet	8
12	Rthe difference between "neural networks" and "deep learning"?	8
13	Methods of learning neural networks	8
14	An attention mechanism for detecting global dependencies between input data and output data	4
15	Self-practice and master the functions of neural network design and implementation software applications	6
16	Preparation for the assessment on the entire material of the module.	10
	Total hours of SRS	126

Policy and control

8. Policy of academic discipline (educational component)

The system of requirements for the student:

- attending lectures and laboratory classes is a mandatory component of studying the material;
- the teacher uses his own presentation material at the lecture; uses Google Drive for teaching the material of the current lecture, additional resources, laboratory work, etc.; the teacher opens access to a certain Google Drive directory for downloading electronic laboratory reports;
- during lectures, it is forbidden to distract the teacher from teaching the material, all questions, clarifications, etc. students ask at the end of the lecture in the time allotted for this;
- laboratory works are defended in two stages - the first stage: students perform tasks for admission to the defense of laboratory work; the second stage is protection of laboratory work. Points for laboratory work are taken into account only if there is an electronic report;
- modular test papers are written in lectures without the use of aids (mobile phones, tablets, etc.); the result is forwarded in a file to the corresponding Google Drive directory;
- incentive points are awarded for: active participation in lectures; participation in faculty and institute olympiads in academic disciplines, participation in work competitions, preparation of reviews of scientific works; presentations on one of the topics of the SRS discipline, etc. The number of encouraged points is more than 10;
- penalty points are issued for: untimely submission of laboratory work. The number of penalty points is no more than 10.

9. Types of control and rating system for evaluating learning outcomes (RSO)

The student's rating in the discipline consists of the points he receives for:

1. execution and protection of 8 laboratory works;
2. execution of modular control work (MKR);
3. incentive and penalty points.

System of rating points and evaluation criteria

Laboratory works:
"excellent", a complete answer to the questions during the defense (at least 90% of the required information) and a properly prepared electronic protocol for laboratory work - 10 points;
"good", a sufficiently complete answer to the questions during the defense (at least 75% of the required information) and a properly prepared electronic protocol for laboratory work - 9/7 points;
"satisfactory", incomplete answer to the questions during the defense (at least 60% of the required information), minor errors and properly prepared electronic protocol for laboratory work - 6/5 points;
"unsatisfactory", an unsatisfactory answer and/or an improperly prepared electronic protocol for laboratory work - 0 points.

For late submission of laboratory work before the deadline, i.e. late submission of laboratory work by more than one class, the grade is reduced by 2 points.

Modular control works:
"excellent", complete completion of the task (at least 90% of the required information in the report to the MKR) - 20 points;
"good", a sufficiently complete answer (at least 75% of the required information), or a complete answer with minor errors - 16-18 points;
"satisfactory", incomplete answer (but not less than 60% of the required information) and minor errors - 10-16 points;
"unsatisfactory", unsatisfactory answer (incorrect problem solution), requires mandatory rewriting at the end of the semester - 0 points.

Incentive points
– for the performance of creative works from the credit module (for example, participation in faculty and institute olympiads in academic disciplines, participation in work competitions, preparation of reviews of scientific works, etc.); for active work at the lecture (questions, additions, comments on the topic of the lecture, when the lecturer invites students to ask their questions) 1-2 points, but not more than 10 in total;
– presentations on SRS - from 1 to 5 points.

Intersessional certification
According to the results of educational work for the first 7 weeks, the maximum possible number of points is 20 points (2 laboratory points). At the first certification (8th week), the student receives "credited" if his current rating is not less than 10 points.

According to the results of 13 weeks of training, the maximum possible number of points is 70 points (4 laboratory, MKR). At the second certification (14th week), the student receives "credited" if his current rating is not less than 40 points.

The maximum sum of weighted points of control measures during the semester is:

$$RD = 7*rlab+rmkr=7*10+30\ (rz - rsh)=100 + (rz - rsh),$$

where rlab is a score for laboratory work (0...10);
rmkr – score for writing MKR (0...20);

rz - incentive points for active participation in lectures, presentations, participation in Olympiads, work competitions, scientific works on the subject of the discipline (0...10);
rzsh - penalty points.

Test:
The condition for obtaining a credit is the enrollment of all laboratory works, writing modular control papers and a starting rating of at least 42 points.

During the test, students perform a written test. Each ticket contains three theoretical questions (tasks). The list of theoretical questions is given in Appendix 1. Each question (task) is valued at 15 points.

Question evaluation system:
"excellent", a complete answer, at least 90% of the required information, which was completed in accordance with the requirements for the "skills" level (complete, error-free solution of the task) - 14-15 points;
"good", sufficiently complete answer, at least 75% of the required information, fulfilled in accordance with the requirements for the "skill" level or there are minor inaccuracies (complete solution of the task with minor inaccuracies) - 10-14 points;
"satisfactory", incomplete answer, at least 60% of the required information, completed in accordance with the requirements for the "stereotypical" level and some errors (the task was completed with certain shortcomings) - 7-9 points;
"unsatisfactory", the answer does not meet the conditions for "satisfactory" - 0-6 points.

The sum of points for the final test is converted to the final grade according to the table:

Table 1. Conversion of rating points to grades on the university scale

Scores	Rating
100-95	Perfectly
94-85	Very good
84-75	Fine
74-65	Satisfactorily
64-60	Enough
Less than 60	Unsatisfactorily
There are uncredited laboratory works or modular control work is not counted	Not allowed

10. Additional information on the discipline (educational component)

- the list of theoretical questions submitted for semester control is given in Appendix 1;
- at the beginning of the semester, the teacher analyzes the existing courses on the topic of the discipline and offers students to take the corresponding free courses. After the student receives a certificate of completion of remote or online courses on the relevant topic, the teacher closes the relevant part of the course (laboratory or lectures) by prior agreement with the group.

Working program of the academic discipline (Syllabus):

FoldedPh.D., Associate Professor, Volodymyr Mykolayovych Shymkovich

ApprovedDepartment of ICT (protocol No. 13 dated 15.06.2022)

AgreedMethodical commission of the faculty[1](protocol No. 11 dated 07.07.2022)

[1]Methodical council of the university - for general university disciplines.

Appendix 1

A list of theoretical questions for assessment in the first part of the course

1. Properties of artificial neural networks.
2. Structure and description of an artificial neuron.
3. Activation functions of an artificial neuron.
4. Types of artificial neural networks
5. Forward propagation neural network, its architecture and properties.
6. Recurrent neural networks, their structure, activation functions and properties.
7. Radial-basis neural networks, their structure, activation functions and properties.
8. Parceptron neural network. Structure, activation functions and properties.
9. A dataset for training neural networksMNIST and establishment of a linear standard.
10. Databases Iris, CIFAR-10, ImageNet, ADE20K, – detailed review, comparative analysis of application in different systems.
11. Coco, Fashion-MNIST, Boston housing– detailed consideration, comparative analysis of application in different systems.
12. The principle of creating training databases for neural networks. Selection of informative features.
13.Momentum and rate adaptive learning: using momentum to accelerate learning.
14. Nesterov's impulse; impulse in the code; variable and adaptive learning rates.
15. Constant learning rate vs RMSP code-in-code.
16. Adam optimization, adam in code, suggestion window.
18. Hyperparameter selection.
19. Hyperparameter Optimization: Cross Validation, Grid Search and Random Search, Sampling Logarithmically, Grid Search in Code, Changing Grid Search, Random Search in Code.
20. General characteristics of ANN learning methods.
21. Vanishing and explosive gradients in ANN learning methods.
22. Local versus global minima in ANN learning methods.
23. Comparison of software implementations of learning methods and their speed.
24. Theano basics: variables, functions, expressions, optimization.
25. building a neural network in Theano.
26. Overview and analysis of constructed neural network systems in a software packageTheano.
27. TensorFlow basics: variables, functions, expressions, optimization.
28. Building a neural network in TensorFlow.
29. Overview and analysis of constructed neural network systems in a software packageTensorFlow.
28. Implementation of neural network systems on parallel computing systems.
29. Acceleration of a software product using a graphics processor.
30. Can Big Data Be Used to Accelerate Backpropagation?
31. GPU acceleration of Theano and Tensorflow packages.
32. Description of the face recognition problem.

33. Class-based ANNs in Theano.
34. Class-based ANNs in TensorFlow.
35. Content of the face recognition project.
36. Review and analysis of software neural network projects for the implementation of facial recognition systems.
37. Modern methods of regularization..
38. Screening regularization, screening intuition, noise injection.
39. The theory of party normalization.
40. Exponentially smoothed averages.
41. Batch normalization flow tensor.
42. Theano and TensorFlow batch normalization.
43. Basics of Keras.
44. Keras in code, functional Keras API.
45. How to easily convert Keras to Tensorflow 2.0 code.
46. Overview and analysis of constructed neural network systems in a software packageKeras.
47. The basics of PyTorch.
48. Dropping PyTorch;
49. PyTorch batch norm.
50. Review and analysis of software products for the implementation of neural network systemsCNTK
51. Review and analysis of software products for the implementation of neural network systemsMXNet
52. Types of deep neural networks.
53. Differences between deep learning and ANNs.
54. Convolutional neural networks, their architecture, principle of operation, properties.
55. Deep probabilistic neural networks, their architecture, principle of action, properties.
56. Deep fuzzy neural networks, their architecture, principle of action, properties.
57. Generative competitive networks, their architecture, principle of action, properties.
58. Deep automatic coding networks, their architecture, principle of operation, properties.
59. Deep convolutional neural networks.
60. Parallelization of calculations in the implementation of convolutional neural networks.
61. Methods of learning deep neural networks, their general characteristics.
62. Selection of learning speed during program implementation.
63. Penalty for manual regularizationspeed
64. The structure of neural networks of transformers
65. Scope of application of neural networks of transformers
66. Review and analysis of software applications with transformer networks
67. An attention mechanism for detecting global dependencies between input data and output data