SYLLABUS
Academic year 2025-2026

Dean,
Prof. dr. eng. Adrian Burlacu

1. Program data

1.1 Higher education institution	"Gheorghe Asachi" Technical University of Iași
1.2 Faculty	Automatic Control and Computer Engineering
1.3 Department	Computers
1.4 Field of studies	Computers and Information Technology
1.5 The cycle of studies[1]	Master
1.6 Study program	Artificial Intelligence

2. Subject data

2.1 Name of the subject / Code	**Deep Learning** *(Învățare profundă)* / **AI.106**						
2.2 Course coordinator	Assoc. prof. dr. eng. Marius Gavrilescu						
2.3 Application instructor	Assoc. prof. dr. eng. Marius Gavrilescu						
2.4 Year of study[2]	1	2.5 Semester[3]	2	2.6 Type of assessment[4]	colloquium	2.7 Type of subject[5]	DS

3. Estimated total time of daily activities (hours per semester)

3.1 Number of hours per week	3	3.2 lectures	1	3.3a sem.		3.3b laboratory	2	3.3c project	
3.4 Total hours in curriculum[6]	42	3.5 lectures	14	3.6a sem.		3.6b laboratory	28	3.6c project	

Distribution of the time fund[7]	No. hours
Study by textbook, course support, bibliography and notes	23
Additional documentation in the library, on specialist electronic platforms and in the field	40
Preparation of seminars/labs/projects, assignments, reports and portfolios	20
Tutorial[8]	
Examinations[9]	4
Other activities:	

3.7 Total hours of individual study[10]	83
3.8 Total hours per semester[11]	125
3.9 Number of credits	5

4. Prerequisites (where applicable)

4.1 curriculum[12]	
4.2 competences	

5. Conditions (where applicable)

5.1 conducting the lectures[13]	• Video projector
5.2 conducting the seminar / laboratory / project[14]	• Laboratory room with computers and Internet access • The Visual Studio programming environment (academic license)

6. Specific competences accumulated[15]

		Number of credits assigned to the subject[16]:	**5**	Distribution of credits per competences[17]
Professional competences	CP1	Knowledge of advanced concepts of computer science and information technology and the ability to work with these concepts.		1
	CP2	Scientific and practical research in the field of artificial intelligence.		0.5
	CP3	Design and development of artificial intelligence systems.		1
	CP4	Problem solving using artificial intelligence methods and techniques.		1
	CP5	Utilization of artificial intelligence tools and technologies.		1
	CP6			
	CPS1			
	CPS2			
Transversal competences	CT1	Legislation compliant application of the intellectual property rights and of the principles, norms and values of the professional ethics code within their own strategies for rigorous, effective and responsible work.		0.1
	CT2	Application of communication techniques and effective group work; developing empathic interpersonal communication skills and assuming leadership roles/functions in a multi-specialized team.		0.2

	CT3	Creating opportunities for continuous training and the effective utilization of learning resources and techniques for personal development.	0.2
	CTS		

7. Objectives of the subject (resulting from the grid of specific competences accumulated)

7.1 General objective	Providing students with a comprehensive understanding of deep learning principles and methodologies, with emphasis on the design, implementation, optimization and deployment of neural network models. The course focuses on both theoretical foundations and practical aspects, including tensor-based computation, data management, training strategies, augmentation, transfer learning and inference optimization, enabling students to develop, analyze and apply deep learning solutions to real-world problems using modern computational frameworks.
7.2 Specific objectives	• Understanding the fundamental concepts and principles of deep learning, including neural network architectures, training mechanisms and optimization strategies. • Developing the ability to work with tensor-based computational models and automatic differentiation within modern deep learning frameworks. • Acquiring skills for efficient data handling, including preprocessing, augmentation, batching and dataset management in large-scale learning scenarios. • Understanding and applying advanced training techniques, including adaptive batching, regularization, handling class imbalance and hyperparameter optimization. • Applying transfer learning, dynamic modeling and curriculum learning strategies to improve model performance and generalization. • Designing, implementing and evaluating deep learning models for practical applications, including deployment and inference optimization in real-world environments.

8. Contents

8.1 Course[18]	Teaching methods[19]	Remarks
1. Introduction Definition of deep learning; relationship between AI, ML and DL; importance, applications, scalability and role of data and computational power. Neural network fundamentals: layers, neurons, weights, activation functions; loss functions, backpropagation, gradient descent (BGD, SGD, MBGD), regularization, batching, data handling and deep learning frameworks overview. **2. Tensor-based Computing** Tensors: definition, order vs rank, shape, size, data types, memory layout, contiguity and strides; tensor representation of multidimensional data and role in deep learning. Tensor operations and differentiation: creation, indexing, slicing, reshaping, concatenation, broadcasting, mathematical and matrix operations; automatic differentiation, computational graphs, gradient computation and custom autograd functions. **3. Data Set Management** Dataset challenges and importance: memory constraints, slow I/O, transformation overhead, distributed data, labeling and imbalance; role of data quality, mini-batching, shuffling, augmentation and efficient loading. Dataset vs DataLoader and PyTorch implementation: Dataset structure and methods, DataLoader functionality (batching, shuffling, parallelism), built-in datasets, custom datasets, streaming data with IterableDataset and large-scale data handling. **4. Batching Strategies** Batching fundamentals and gradient descent variants: epochs, batches, mini-batches, batch size; batch, stochastic and mini-batch gradient descent, gradient estimation, variance, learning rate interaction and implicit regularization through noisy gradients. Batch size strategies: constant vs dynamic batching, advantages and limitations; adaptive batching based on epoch, loss, confidence, gradient noise and curriculum learning for balancing exploration, stability and generalization. **5. Augmentation Techniques for Deep Models** Objectives and motivation of data augmentation: addressing limited data, overfitting, class imbalance and robustness; label-preserving transformations for	Lectures via Powerpoint presentations, explanations and discussions	

images and text, including geometric, photometric, noise, occlusion and NLP-based transformations. Advanced augmentation methods: oversampling techniques (random, SMOTE, ADASYN), mixed-sample methods (Mixup, CutMix) and learning-based approaches (AutoAugment, RandAugment, GANs, autoencoders) for improving generalization and diversity. **6. Transfer Learning** Definition and motivation of transfer learning: reuse of knowledge (features, weights, embeddings) from source to target tasks; domain vs task adaptation, data scarcity, computational constraints and benefits over training from scratch.Transfer learning taxonomy and approaches: inductive, transductive and unsupervised transfer; feature-based, instance-based and parameter-based methods including domain adaptation (MMD, adversarial), instance weighting/translation and fine-tuning with layer freezing and multi-task learning. **7. Model Deployment and Inference Optimization** From training to deployment: model serialization, reproducibility, inference pipelines, hardware considerations (CPU, GPU), latency vs throughput trade-offs and constraints in real-world applications. Inference optimization techniques: model compression (pruning, quantization, distillation), batching and streaming inference, efficient architectures and deployment frameworks for scalable and resource-constrained environments.		

Course references:

[1] C. F. G. D. Santos and J. P. Papa, “Avoiding Overfitting: A Survey on Regularization Methods for Convolutional Neural Networks,” ACM Computing Surveys, vol. 54, no. 10s, pp. 1–25, Jan. 2022, doi: 10.1145/3510413.
[2] J. E. van Engelen and H. H. Hoos, “A survey on semi-supervised learning,” Machine Learning, vol. 109, no. 2, pp. 373–440, Nov. 2019, doi: 10.1007/s10994-019-05855-6.
[3] M. H. Farrell, T. Liang, and S. Misra, “Deep Neural Networks for Estimation and Inference,” Econometrica, vol. 89, no. 1, pp. 181–213, 2021, doi: 10.3982/ecta16901.
[4] C. Chen, D. Han, and J. Wang, “Multimodal Encoder-Decoder Attention Networks for Visual Question Answering,” IEEE Access, vol. 8, pp. 35662–35671, 2020, doi: 10.1109/access.2020.2975093.
[5] K. Zhou, Z. Liu, Y. Qiao, T. Xiang, and C. C. Loy, “Domain Generalization: A Survey,” IEEE Transactions on Pattern Analysis and Machine Intelligence, pp. 1–20, 2022, doi: 10.1109/tpami.2022.3195549.
[6] A. Saxe, S. Nelli, and C. Summerfield, “If deep learning is the answer, what is the question?,” Nature Reviews Neuroscience, vol. 22, no. 1, pp. 55–67, Nov. 2020, doi: 10.1038/s41583-020-00395-8.
[7] K. Han et al., “A Survey on Vision Transformer,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 45, no. 1, pp. 87–110, Jan. 2023, doi: 10.1109/tpami.2022.3152247.
[8] X. Dong, Z. Yu, W. Cao, Y. Shi, and Q. Ma, “A survey on ensemble learning,” Frontiers of Computer Science, vol. 14, no. 2, pp. 241–258, Aug. 2019, doi: 10.1007/s11704-019-8208-z.
[9] R. Miikkulainen et al., “Evolving deep neural networks,” Artificial Intelligence in the Age of Neural Networks and Brain Computing, pp. 269–287, 2024, doi: 10.1016/b978-0-323-96104-2.00002-6.
[10] J. A. Livezey and J. I. Glaser, “Deep learning approaches for neural decoding across architectures and recording modalities,” Briefings in Bioinformatics, vol. 22, no. 2, pp. 1577–1591, Dec. 2020, doi: 10.1093/bib/bbaa355.
[11] Y. Ji, H. Zhang, Z. Zhang, and M. Liu, “CNN-based encoder-decoder networks for salient object detection: A comprehensive review and recent advances,” Information Sciences, vol. 546, pp. 835–857, Feb. 2021, doi: 10.1016/j.ins.2020.09.003.
[12] L. Zhang and X. Gao, “Transfer Adaptation Learning: A Decade Survey,” IEEE Transactions on Neural Networks and Learning Systems, vol. 35, no. 1, pp. 23–44, Jan. 2024, doi: 10.1109/tnnls.2022.3183326.
[13] Y. Han, G. Huang, S. Song, L. Yang, H. Wang, and Y. Wang, “Dynamic Neural Networks: A Survey,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 44, no. 11, pp. 7436–7456, Nov. 2022, doi: 10.1109/tpami.2021.3117837.
[14] B. Bischl et al., “Hyperparameter optimization: Foundations, algorithms, best practices, and open challenges,” WIREs Data Mining and Knowledge Discovery, vol. 13, no. 2, Jan. 2023, doi: 10.1002/widm.1484.
[15] E. Hancer, B. Xue, and M. Zhang, “A survey on feature selection approaches for clustering,” Artificial Intelligence Review, vol. 53, no. 6, pp. 4519–4545, Jan. 2020, doi: 10.1007/s10462-019-09800-w.
[16] X. He, K. Zhao, and X. Chu, “AutoML: A survey of the state-of-the-art,” Knowledge-Based Systems, vol. 212, p. 106622, Jan. 2021, doi: 10.1016/j.knosys.2020.106622.
[17] G. James, D. Witten, T. Hastie, R. Tibshirani, and J. Taylor, “Unsupervised Learning,” An Introduction to Statistical Learning, pp. 503–556, 2023, doi: 10.1007/978-3-031-38747-0_12.
[18] I. D. Mienye and Y. Sun, “A Survey of Ensemble Learning: Concepts, Algorithms, Applications, and Prospects,” IEEE Access, vol. 10, pp. 99129–99149, 2022, doi: 10.1109/access.2022.3207287.
[19] G. Matteucci, E. Piasini, and D. Zoccolan, “Unsupervised learning of mid-level visual representations,” Current Opinion in Neurobiology, vol. 84, p. 102834, Feb. 2024, doi: 10.1016/j.conb.2023.102834.
[20] T. Jiang, J. L. Gradus, and A. J. Rosellini, “Supervised Machine Learning: A Brief Primer,” Behavior Therapy, vol.

51, no. 5, pp. 675–687, Sep. 2020, doi: 10.1016/j.beth.2020.05.002.

8.2a Seminar	Teaching methods[20]	Remarks
8.2b Laboratory	Teaching methods[21]	Remarks
1. Fundamentals of Neural Network Implementation Implementation of a feedforward neural network for regression using a deep learning framework. Training, optimization and evaluation of the model, including architecture, loss functions, optimizers and regularization techniques. **2. Tensor-Based Computing** Practical exercises with tensor creation, indexing, slicing, reshaping, broadcasting and linear algebra operations using a deep learning framework. Implementation of automatic differentiation and efficient tensor computations on CPU/GPU, including gradient tracking and optimized numerical operations. **3. Tensor Computing in Neural Networks** Implementation of neural network computations using explicit tensor operations, including forward propagation and manual backpropagation for perceptrons and shallow networks. Comparison between manual tensor-based implementations and framework-based approaches, analyzing gradient computation, training dynamics and convergence behavior. **4. Tensor Differentiation and Optimization** Practical implementation of automatic differentiation using computational graphs and autograd, including gradient computation for scalar and tensor functions. Application of gradient-based optimization methods, including numerical gradient approximation, momentum techniques and custom loss functions with explicit gradient definitions. **5. Data Sets and Data Loaders** Implementation of dataset management using custom and built-in datasets, including data loading, batching, shuffling and parallel processing with DataLoader. Application of data preprocessing, transformations and custom collation functions for handling diverse and variable-size data in deep learning pipelines. **6. Adaptive Data Batching** Implementation and analysis of batching strategies, including fixed, dynamic and adaptive batching, with emphasis on their impact on training efficiency, convergence and generalization. Application of adaptive batch size adjustment techniques based on training dynamics, including epoch-based, loss-based, variance-based, confidence-based and gradient noise scale methods. **7. Data Preprocessing Techniques** Application of data normalization methods, including standardization and min-max scaling, and analysis of their impact on training stability and convergence. Handling real-world data issues through missing value imputation (mean, median, KNN) and encoding categorical variables using one-hot representations. **8. Dynamic Models and Curriculum Learning** Implementation of dynamic neural networks in PyTorch using define-by-run principles, enabling runtime adaptation of computational graphs and model behavior. Application of curriculum learning strategies, including progressive data complexity and dynamic architecture modification, with comparison to static models in terms of performance and convergence. **9. Handling Class Imbalance** Implementation and evaluation of neural network models on imbalanced datasets, analyzing performance using metrics such as precision, recall, F1-score and accuracy. Application of imbalance mitigation techniques, including class weighting, oversampling (random, SMOTE, ADASYN) and undersampling (Tomek Links, NearMiss), and comparison of their impact on model performance.	General and individual explanations, individual computer work.	

10. Data Augmentation Implementation of image augmentation techniques, including geometric, photometric and occlusion-based transformations, integrated into data loading pipelines. Application of advanced augmentation methods such as MixUp and CutMix, with analysis of their impact on model generalization and training performance. **11. Hyperparameter Tuning and Optimization** Implementation of hyperparameter optimization strategies, including manual search, grid search, random search and automated tuning using Optuna for neural network models. Design and evaluation of optimized architectures and training configurations, including multilayer and convolutional networks, based on validation performance. **12. Model Deployment and Inference Optimization** Implementation of model serialization and deployment workflows, including saving/loading trained models and performing inference on new data. Application of inference optimization techniques, including batching, hardware-aware execution (CPU/GPU) and basic model compression methods, with analysis of latency and throughput. **13. Transfer Learning and Fine-Tuning** Implementation of transfer learning using pretrained models, including feature extraction and fine-tuning for new tasks. Evaluation of different fine-tuning strategies, including layer freezing, learning rate adjustment and domain adaptation, with analysis of performance improvements.		
8.2c Project	Teaching methods[22]	Remarks

Applications (laboratory / project) references:

[1] G. James, D. Witten, T. Hastie, R. Tibshirani, and J. Taylor, “Unsupervised Learning,” An Introduction to Statistical Learning, pp. 503–556, 2023, doi: 10.1007/978-3-031-38747-0_12.

[2] J. E. van Engelen and H. H. Hoos, “A survey on semi-supervised learning,” Machine Learning, vol. 109, no. 2, pp. 373–440, Nov. 2019, doi: 10.1007/s10994-019-05855-6.

[3] Y. Ji, H. Zhang, Z. Zhang, and M. Liu, “CNN-based encoder-decoder networks for salient object detection: A comprehensive review and recent advances,” Information Sciences, vol. 546, pp. 835–857, Feb. 2021, doi: 10.1016/j.ins.2020.09.003.

[4] X. He, K. Zhao, and X. Chu, “AutoML: A survey of the state-of-the-art,” Knowledge-Based Systems, vol. 212, p. 106622, Jan. 2021, doi: 10.1016/j.knosys.2020.106622.

[5] E. Hancer, B. Xue, and M. Zhang, “A survey on feature selection approaches for clustering,” Artificial Intelligence Review, vol. 53, no. 6, pp. 4519–4545, Jan. 2020, doi: 10.1007/s10462-019-09800-w.

[6] Y. Han, G. Huang, S. Song, L. Yang, H. Wang, and Y. Wang, “Dynamic Neural Networks: A Survey,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 44, no. 11, pp. 7436–7456, Nov. 2022, doi: 10.1109/tpami.2021.3117837.

[7] A. Saxe, S. Nelli, and C. Summerfield, “If deep learning is the answer, what is the question?,” Nature Reviews Neuroscience, vol. 22, no. 1, pp. 55–67, Nov. 2020, doi: 10.1038/s41583-020-00395-8.

[8] R. Miikkulainen et al., “Evolving deep neural networks,” Artificial Intelligence in the Age of Neural Networks and Brain Computing, pp. 269–287, 2024, doi: 10.1016/b978-0-323-96104-2.00002-6.

[9] M. H. Farrell, T. Liang, and S. Misra, “Deep Neural Networks for Estimation and Inference,” Econometrica, vol. 89, no. 1, pp. 181–213, 2021, doi: 10.3982/ecta16901.

[10] J. A. Livezey and J. I. Glaser, “Deep learning approaches for neural decoding across architectures and recording modalities,” Briefings in Bioinformatics, vol. 22, no. 2, pp. 1577–1591, Dec. 2020, doi: 10.1093/bib/bbaa355.

[11] C. Chen, D. Han, and J. Wang, “Multimodal Encoder-Decoder Attention Networks for Visual Question Answering,” IEEE Access, vol. 8, pp. 35662–35671, 2020, doi: 10.1109/access.2020.2975093.

[12] K. Han et al., “A Survey on Vision Transformer,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 45, no. 1, pp. 87–110, Jan. 2023, doi: 10.1109/tpami.2022.3152247.

[13] C. F. G. D. Santos and J. P. Papa, “Avoiding Overfitting: A Survey on Regularization Methods for Convolutional Neural Networks,” ACM Computing Surveys, vol. 54, no. 10s, pp. 1–25, Jan. 2022, doi: 10.1145/3510413.

[14] N. Syam and R. Kaul, “Overfitting and Regularization in Machine Learning Models,” Machine Learning and Artificial Intelligence in Marketing and Sales, pp. 65–84, Mar. 2021, doi: 10.1108/978-1-80043-880-420211004.

[15] B. Bischl et al., “Hyperparameter optimization: Foundations, algorithms, best practices, and open challenges,” WIREs Data Mining and Knowledge Discovery, vol. 13, no. 2, Jan. 2023, doi: 10.1002/widm.1484.

[16] L. Zhang and X. Gao, “Transfer Adaptation Learning: A Decade Survey,” IEEE Transactions on Neural Networks and Learning Systems, vol. 35, no. 1, pp. 23–44, Jan. 2024, doi: 10.1109/tnnls.2022.3183326.

[17] K. Zhou, Z. Liu, Y. Qiao, T. Xiang, and C. C. Loy, “Domain Generalization: A Survey,” IEEE Transactions on Pattern Analysis and Machine Intelligence, pp. 1–20, 2022, doi: 10.1109/tpami.2022.3195549.

9. Corroboration of the contents of the subject with the expectations of representatives of the epistemic community, professional associations and representative employers in the field related to the program[23]

The course serves as an important means of connecting various related courses in the fields of artificial intelligence and computer science. It is strongly linked to fundamental topics such as machine learning, neural networks, and data science. Furthermore, it forms a symbiotic relationship with courses such as computer vision, natural language processing, and reinforcement learning, where the principles of deep learning have multiple practical applications. Additionally, the knowledge and skills gained from this course have significant applicability in the professional landscape. Its practical utility in solving real-world problems, optimizing processes, and enhancing decision-making makes the techniques studies within this course sought-after in the labor market. As industries increasingly incorporate automation and data-driven solutions, professionals equipped with a deep understanding of neural networks and deep learning techniques become essential contributors to innovation and efficiency, making this course both intellectually stimulating and advantageous for future career prospects.

10. Evaluation

Type of activity	10.1 Evaluation criteria	10.2 Evaluation methods		10.3 Weight in the final grade
10.4a Exam	Acquired theoretical and practical knowledge (quantity, correctness, accuracy)	Periodic tests[24]:		50% (minimum 5)
		Homework:		
		Other activities[25]:		
		Final evaluation:	100%	
10.4b Seminar				
10.4c Laboratory	Knowledge of related techniques, ability to use dedicated frameworks, evaluation and interpretation of results	• Practical demonstrations • Oral answers • Written questionnaires		50% (minimum 5)
10.4d Project		•		
10.5 Minimum performance standard[26]: grade 5 in the exam and applications. Developing an understanding of deep neural network architectures, including convolutional and recurrent networks, as well as the ability to work with such architectures in popular machine learning frameworks, for processing, analyzing and organizing data of various types.				

Date of completion,
9.09.2025

Signature of course coordinator,
Assoc. prof. dr. eng. Marius Gavrilescu

Signature of application instructor,
Assoc. prof. dr. eng. Marius Gavrilescu

Date of approval in the department,
23.09.2025

Director of department,
Assoc. prof. dr. eng. Andrei Stan

[1]Bachelor / Master
[2]1-4 for Bachelor's, 1-2 for Master's
[3]1-8 for Bachelors, 1-3 for Masters
[4]Exam, colloquium or VP A/R – from the curriculum
[5]DF - fundamental subject, DID - subject in the field, DS - specialized subject or DC - complementary subject - from the education plan
[6]It is equal to 14 weeksx number of hours from point 3.1 (similar for 3.5, 3.6abc)
[7]The lines below refer to the individual study; the total is completed at point 3.7.
[8]Between 7 and 14 hours
[9]Between 2 and 6 hours
[10]The sum of the values on the previous lines, which refer to the individual study.
[11]The sum of the number of hours of direct teaching activity (3.4) and the number of hours of individual study (3.7); must be equal to the number of credits allocated to the subject (point 3.9)x 24 hours per credit.

[12]*Mention the subjects that must be passed previously or equivalent*
[13]*Blackboard, video projector, flipchart, specific teaching materials, etc.*
[14]*Computing technique, software packages, experimental stands, etc.*
[15]*Competencies from the G1 and G1bis Grids of the study program, adapted to the specifics of the subject, for which credits are allocated (www.rncis.ro or the faculty website)*
[16] *From the education plan*
[17]*The credits allocated to the subject are distributed on professional and transversal competences according to the specifics of the subject*
[18]*Chapter and paragraph headings*
[19]*Exposition, lecture, blackboard presentation of the studied issue, use of video projector, discussions with students (for each chapter, if applicable)*
[20]*Discussions, debates, presentation and/or analysis of papers, solving exercises and problems*
[21]*Practical demonstration, exercise, experiment*
[22]*Case study, demonstration, exercise, error analysis, etc.*
[23]*The connection with other subjects, the usefulness of the subject on the labor market*
[24]*The number of tests and the weeks in which they will be held will be specified.*
[25]*Scientific circles, professional competitions, etc.*
[26]*The minimum performance standard from the competences grid of the study program is customized to the specifics of the subject, if applicable.*