



MACHINE LEARNING

Syllabus

Catalog Description

Higher education level	<i>First (Undergraduate)</i>
Knowledge field	<i>Information Technologies</i>
Profession	<i>121 Software Engineering</i>
Curriculum	<i>Software Engineering of Intelligent Cyberphysical Systems in Energy Industry</i>
Course status	<i>Elective</i>
Form of training	<i>Full-time</i>
Grade, term	<i>Fourth grade, fall semester</i>
Credits (hours)	<i>4 credits / 120 hours (part time: 6 hours of lectures, 4 hours of practice, 110 hours of individual assignments)</i>
Term control	<i>Exam, modular test</i>
Schedule	<i>http://schedule.kpi.ua/</i>
Teaching language	<i>Ukrainian/English</i>
Instructors	Lecturer: <i>DSc. (Econ), professor Andrii Sihaiov</i> Seminars: Laboratory work: <i>Andrii Sihaiov</i>
URL	<i>https://course.fast.ai</i>

Course Program

1 Course description, aim, subject, and expected outcomes

Why future specialist should study this course?

Machine learning is a method of data analysis that automates analytical model building. It is a branch of artificial intelligence based on the idea that systems can learn from data, identify patterns and make decisions with minimal human intervention.

Because of new computing technologies, machine learning today is not like machine learning of the past. It was born from pattern recognition and the theory that computers can learn without being programmed to perform specific tasks; researchers interested in artificial intelligence wanted to see if computers could learn from data. The iterative aspect of machine learning is important because as models are exposed to new data, they are able to independently adapt. They learn from previous computations to produce reliable, repeatable decisions and results. It's a science that's not new — but one that has gained fresh momentum.

While many machine learning algorithms have been around for a long time, the ability to automatically apply complex mathematical calculations to big data — over and over, faster and faster — is a recent development. Here are a few widely publicized examples of machine learning applications you may be familiar with

- The heavily hyped, self-driving Google car? The essence of machine learning.*
- Online recommendation offers such as those from Amazon, Netflix, Spotify, and YouTube ? Machine learning applications for everyday life.*

- *Knowing what customers are saying about you on Twitter? Machine learning combined with linguistic rule creation.*
- *Fraud detection? One of the more obvious, important uses in our world today.*

Who's using it? Most industries working with large amounts of data have recognized the value of machine learning technology. By gleaning insights from this data — often in real time — organizations are able to work more efficiently or gain competitive advantage.

Among the reasons to study machine learning important one in our opinion has been the huge salaries and bonuses that experts in artificial intelligence can command. Nearly all big tech companies have an artificial intelligence project, and they are willing to pay experts millions of dollars to help get it done. Though the pool of available A.I. researchers is growing, it is not growing fast enough. At the moment demand for skilled workforce in the field severely outweighs supply, salaries are skyrocketing, and this situation is not going to change any time soon.

Course Aim. *To familiarize students with modern state of machine learning.*

Course Subject. *An overview of machine learning, including supervised learning, unsupervised learning, deep learning, computer vision problems, collaborative filtering, convolutional neural networks, recurrent neural networks, decision trees, random forests, natural language processing.*

Expected Outcomes.

Professional Competencies.

FC 18. Ability to implement applications using concepts of asynchronous programming, latest programming languages, artificial intelligence, robotic systems and complexes control, machine learning technologies.

Program Learning Outcomes.

PLO 34. Ability to develop applications using the concepts of artificial intelligence and machine learning taking into account the specifics of the subject area of the energy industry.

2 Course prerequisites (Where the course fits into our curriculum)

The course is taken in fall semester of final year. Calculus, Probability Theory, and one of the Python using courses are the prerequisites. There is no required course that has this course as a prerequisite.

3 Course contents

- 1. Introduction to Deep Learning.*
- 2. From Model to Production.*
- 3. Under the Hood: Training a Digit Classifier.*
- 4. Image Classification.*
- 5. Other Computer Vision Problems.*
- 6. Training a State-of-the-Art Model.*
- 7. Collaborative Filtering Deep Dive.*
- 8. Tabular Modeling Deep Dive.*
- 9. NLP Deep Dive: RNNs.*
- 10. Data Munging with fastai's Mid-Level API.*
- 11. A Language Model from Scratch.*
- 12. Convolutional Neural Networks.*
- 13. ResNets.*

14. *Application Architectures Deep Dive.*
15. *The Training Process.*
16. *A Neural Net from the Foundations.*
17. *CNN Interpretation with CAM.*
18. *A fastai Learner from Scratch. Concluding Thoughts.*

4 Course textbooks and materials

Required reading:

Howard, J., Gugger, S. *Deep Learning for Coders with fastai and PyTorch: AI Applications Without a PhD:* Sebastopol, CA: O'Reilly Media, 2020. 624 c. URL: <http://gen.lib.rus.ec/book/index.php?md5=2668D405E21C8708BB3192CDD92D8DAF>

Optional reading:

1. Albon, C. *Machine Learning with Python Cookbook: Practical Solutions from Preprocessing to Deep Learning:* Sebastopol, CA: O'Reilly Media, 2018. 366 c. URL: <http://gen.lib.rus.ec/book/index.php?md5=1b8a00c4b487665f8c785761b3bb8f4b>
2. Harrison, M. *Machine Learning Pocket Reference: Working with Structured Data in Python:* Sebastopol, CA: O'Reilly Media, 2019. 320 c. URL: <http://gen.lib.rus.ec/book/index.php?md5=904606F860FEA95C671B3821CBBDE6C7>
3. Géron, A. *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems:* Sebastopol, CA: O'Reilly Media, 2019. 856 c. URL: <http://gen.lib.rus.ec/book/index.php?md5=E98CBB67C93AD4EF9C550E215E6248BD>
4. Chollet, F. *Deep Learning with Python:* Shelter Island, NY: Manning Publications, 2017. 384 c. URL: <http://libgen.is/item/index.php?md5=584B39E75A5B9E072467AFD6A684D0FB>
5. *Machine Learning Crash Course | Google Developers:* URL: <https://developers.google.com/machine-learning/crash-course>
6. Elger, P., Shanaghy, E. *AI as a Service:* Shelter Island, NY: Manning Publications, 2020. 325 c. URL: <http://gen.lib.rus.ec/book/index.php?md5=68556641E787AFDC9C8BD105D707B27E>
7. Ameisen, E. *Building Machine Learning Powered Applications: Going from Idea to Product:* Sebastopol, CA: O'Reilly Media, 2020. 260 c. URL: <http://libgen.is/book/index.php?md5=7253B8C7533DAA5F7747F332B2ED7D2F>
8. Hurbans, R. *Grokking Artificial Intelligence Algorithms:* Shelter Island, NY: Manning Publications, 2020. 392 c. URL: <http://gen.lib.rus.ec/book/index.php?md5=050DE26CAF73D5D7196B5543ED15530F>
9. Trask, A. W. *Grokking Deep Learning:* Shelter Island, NY: Manning Publications, 2019. 325 c. URL: <http://gen.lib.rus.ec/book/index.php?md5=9beb350de2902304edb5e437a1c5ab3f>
10. Rao, D., McMahan, B. *Natural Language Processing with PyTorch: Build Intelligent Language Applications Using Deep Learning:* Sebastopol, CA: O'Reilly Media, 2019. 256 c. URL: <https://learning.oreilly.com/library/view/natural-language-processing/9781491978221/>
11. Morales, M. *Grokking Deep Reinforcement Learning:* Shelter Island, NY: Manning Publications, 2019. 450 c. URL: <http://gen.lib.rus.ec/book/index.php?md5=F83946E7A76E64E0C309CB9A2086551E>
12. Zai, A., Brown, B. *Deep Reinforcement Learning in Action:* Shelter Island, NY: Manning Publications, 2020. 277 c. URL: <http://gen.lib.rus.ec/book/index.php?md5=F2F28879523CAE239FFB89EA6A533A1A>

13. Koul, A., Ganju, S., Kasam, M. *Practical Deep Learning for Cloud, Mobile, and Edge: Real-World AI & Computer-Vision Projects Using Python, Keras & TensorFlow*: Sebastopol, CA: O'Reilly Media, 2019. 620 c. URL: <http://gen.lib.rus.ec/book/index.php?md5=4EE9011651DEBD8F5F3B5A9503E48921>
14. Pointer, I. *Programming PyTorch for Deep Learning: Creating and Deploying Deep Learning Applications*: Sebastopol, CA: O'Reilly Media, 2019. 220 c. URL: <http://gen.lib.rus.ec/book/index.php?md5=38984CF4F0C132A92DD1B20578A116DA>
15. Foster, D. *Generative Deep Learning: Teaching Machines to Paint, Write, Compose, and Play*: Sebastopol, CA: O'Reilly Media, 2019. 330 c. URL: <http://libgen.is/book/index.php?md5=41fd02aff65fdb65143d0a9ff0c60fda>
16. Lane, H., Hapke, H., Howard, C. *Natural Language Processing in Action: Understanding, analyzing, and generating text with Python*: Shelter Island, NY: Manning Publications, 2019. 544 c. URL: <http://libgen.is/item/index.php?md5=dc168df28d64176a28151ccee6674d5a>
17. Bengfort, B., Bilbro, R., Ojeda, T. *Applied Text Analysis with Python: Enabling Language-Aware Data Products with Machine Learning*: Sebastopol, CA: O'Reilly Media, 2018. 332 c. URL: <http://libgen.io/book/index.php?md5=e0efd0aa7b373e509d189655ba3189b0>
18. Raschka, S., Mirjalili, V. *Python Machine Learning: Machine Learning and Deep Learning with Python, scikit-learn, and TensorFlow 2*: Birmingham, UK: Packt Publishing, 2019. 770 c. URL: <http://gen.lib.rus.ec/book/index.php?md5=6ab53d6bd918569dbbf95eed84ac51ab>
19. Cai, S., Bileschi, S., Nielsen, E. *Deep Learning with JavaScript: Neural networks in TensorFlow.js*: Shelter Island, NY: Manning Publications, 2020. 350 c. URL: <http://gen.lib.rus.ec/book/index.php?md5=0c26a4ef4a5e3f77c4f3a20a76af4c48>

Educational Content

5 Pedagogical advice

1. *Introduction to Deep Learning.*
 - 1.1. *Neural Networks: A Brief History.*
 - 1.2. *How to Learn Deep Learning.*
 - 1.2.1. *Your Projects and Your Mindset.*
 - 1.3. *The Software: PyTorch, fastai, and Jupyter (And Why It Doesn't Matter).*
 - 1.4. *Your First Model.*
 - 1.4.1. *Getting a GPU Deep Learning Server.*
 - 1.4.2. *Running Your First Notebook.*
 - 1.4.3. *What Is Machine Learning?*
 - 1.4.4. *What Is a Neural Network?*
 - 1.4.5. *A Bit of Deep Learning Jargon*
 - 1.4.6. *Limitations Inherent to Machine Learning*
 - 1.4.7. *How Our Image Recognizer Works*
 - 1.4.8. *What Our Image Recognizer Learned*
 - 1.4.9. *Image Recognizers Can Tackle Non-Image Tasks*
 - 1.4.10. *Jargon Recap*

- 1.5. *Deep Learning Is Not Just for Image Classification*
- 1.6. *Validation Sets and Test Sets*
- 1.7. *Use Judgment in Defining Test Sets*
- 1.8. *A “Choose Your Own Adventure” Moment*
2. *From Model to Production.*
 - 2.1. *The Practice of Deep Learning.*
 - 2.1.1. *Starting Your Project.*
 - 2.1.2. *The State of Deep Learning.*
 - 2.1.3. *The Drivetrain Approach.*
 - 2.2. *Gathering Data.*
 - 2.3. *From Data to DataLoaders.*
 - 2.3.1. *Data Augmentation.*
 - 2.4. *Training Your Model, and Using It to Clean Your Data.*
 - 2.5. *Turning Your Model into an Online Application.*
 - 2.5.1. *Using the Model for Inference.*
 - 2.5.2. *Creating a Notebook App from the Model.*
 - 2.5.3. *Turning Your Notebook into a Real App.*
 - 2.5.4. *Deploying Your App.*
 - 2.6. *How to Avoid Disaster.*
 - 2.6.1. *Unforeseen Consequences and Feedback Loops.*
 - 2.7. *Get Writing!*
3. *Under the Hood: Training a Digit Classifier.*
 - 3.1. *Pixels: The Foundations of Computer Vision.*
 - 3.2. *First Try: Pixel Similarity.*
 - 3.2.1. *NumPy Arrays and PyTorch Tensors.*
 - 3.3. *Computing Metrics Using Broadcasting.*
 - 3.4. *Stochastic Gradient Descent.*
 - 3.4.1. *Calculating Gradients.*
 - 3.4.2. *Stepping with a Learning Rate.*
 - 3.4.3. *An End-to-End SGD Example.*
 - 3.4.4. *Summarizing Gradient Descent.*
 - 3.5. *The MNIST Loss Function.*
 - 3.5.1. *Sigmoid.*
 - 3.5.2. *SGD and Mini-Batches.*
 - 3.6. *Putting It All Together.*
 - 3.6.1. *Creating an Optimizer.*
 - 3.7. *Adding a Nonlinearity.*

- 3.7.1. *Going Deeper.*
 - 3.8. *Jargon Recap.*
 - 4. *Image Classification.*
 - 4.1. *From Dogs and Cats to Pet Breeds.*
 - 4.2. *Resizing.*
 - 4.2.1. *Checking and Debugging a DataBlock.*
 - 4.3. *Cross-Entropy Loss.*
 - 4.3.1. *Viewing Activations and Labels.*
 - 4.3.2. *Softmax.*
 - 4.3.3. *Log Likelihood.*
 - 4.3.4. *Taking the log.*
 - 4.4. *Model Interpretation.*
 - 4.5. *Improving Our Model.*
 - 4.5.1. *The Learning Rate Finder.*
 - 4.5.2. *Unfreezing and Transfer Learning.*
 - 4.5.3. *Discriminative Learning Rates.*
 - 4.5.4. *Selecting the Number of Epochs.*
 - 4.5.5. *Deeper Architectures.*
 - 4.6. *Conclusion.*
 - 5. *Other Computer Vision Problems.*
 - 5.1. *Multi-Label Classification.*
 - 5.1.1. *The Data.*
 - 5.1.2. *Constructing a DataBlock.*
 - 5.1.3. *Binary Cross Entropy.*
 - 5.2. *Regression.*
 - 5.2.1. *Assembling the Data.*
 - 5.2.2. *Training a Model.*
 - 5.3. *Conclusion.*
 - 6. *Training a State-of-the-Art Model.*
 - 6.1. *Imagenette.*
 - 6.2. *Normalization.*
 - 6.3. *Progressive Resizing.*
 - 6.4. *Test Time Augmentation.*
 - 6.5. *Mixup.*
 - 6.6. *Label Smoothing.*
 - 6.7. *Conclusion.*
 - 7. *Collaborative Filtering Deep Dive.*

- 7.1. *A First Look at the Data.*
- 7.2. *Learning the Latent Factors.*
- 7.3. *Creating the DataLoaders.*
- 7.4. *Collaborative Filtering from Scratch.*
 - 7.4.1. *Weight Decay.*
 - 7.4.2. *Creating Our Own Embedding Module.*
- 7.5. *Interpreting Embeddings and Biases.*
 - 7.5.1. *Using fastai.collab.*
 - 7.5.2. *Embedding Distance.*
- 7.6. *Bootstrapping a Collaborative Filtering Model.*
- 7.7. *Deep Learning for Collaborative Filtering.*
- 7.8. *Conclusion.*
- 8. *Tabular Modeling Deep Dive.*
 - 8.1. *Categorical Embeddings.*
 - 8.2. *Beyond Deep Learning.*
 - 8.3. *The Dataset.*
 - 8.3.1. *Kaggle Competitions.*
 - 8.3.2. *Look at the Data.*
 - 8.4. *Decision Trees.*
 - 8.4.1. *Handling Dates.*
 - 8.4.2. *Using TabularPandas and TabularProc.*
 - 8.4.3. *Creating the Decision Tree.*
 - 8.4.4. *Categorical Variables.*
 - 8.5. *Random Forests.*
 - 8.5.1. *Creating a Random Forest.*
 - 8.5.2. *Out-of-Bag Error.*
 - 8.6. *Model Interpretation.*
 - 8.6.1. *Tree Variance for Prediction Confidence.*
 - 8.6.2. *Feature Importance.*
 - 8.6.3. *Removing Low-Importance Variables.*
 - 8.6.4. *Removing Redundant Features.*
 - 8.6.5. *Partial Dependence.*
 - 8.6.6. *Data Leakage.*
 - 8.6.7. *Tree Interpreter.*
 - 8.7. *Extrapolation and Neural Networks.*
 - 8.7.1. *The Extrapolation Problem.*
 - 8.7.2. *Finding Out-of-Domain Data.*

- 8.7.3. *Using a Neural Network.*
- 8.8. *Ensembling.*
 - 8.8.1. *Boosting.*
 - 8.8.2. *Combining Embeddings with Other Methods.*
- 8.9. *Conclusion*
- 9. *NLP Deep Dive: RNNs.*
 - 9.1. *Text Preprocessing.*
 - 9.1.1. *Tokenization.*
 - 9.1.2. *Word Tokenization with fastai.*
 - 9.1.3. *Subword Tokenization.*
 - 9.1.4. *Numericalization with fastai.*
 - 9.1.5. *Putting Our Texts into Batches for a Language Model.*
 - 9.2. *Training a Text Classifier.*
 - 9.2.1. *Language Model Using DataBlock.*
 - 9.2.2. *Fine-Tuning the Language Model.*
 - 9.2.3. *Saving and Loading Models.*
 - 9.2.4. *Text Generation.*
 - 9.2.5. *Creating the Classifier DataLoaders.*
 - 9.2.6. *Fine-Tuning the Classifier.*
 - 9.3. *Disinformation and Language Models.*
 - 9.4. *Conclusion.*
- 10. *Data Munging with fastai's Mid-Level API.*
 - 10.1. *Going Deeper into fastai's Layered API.*
 - 10.1.1. *Transforms.*
 - 10.1.2. *Writing Your Own Transform.*
 - 10.1.3. *Pipeline.*
 - 10.2. *TfmdLists and Datasets: Transformed Collections.*
 - 10.2.1. *TfmdLists.*
 - 10.2.2. *Datasets.*
 - 10.3. *Applying the Mid-Level Data API: SiamesePair.*
 - 10.4. *Conclusion.*
- 11. *A Language Model from Scratch.*
 - 11.1. *The Data.*
 - 11.2. *Our First Language Model from Scratch.*
 - 11.2.1. *Our Language Model in PyTorch.*
 - 11.2.2. *Our First Recurrent Neural Network.*
 - 11.3. *Improving the RNN.*

11.3.1. *Maintaining the State of an RNN.*

11.3.2. *Creating More Signal.*

11.4. *Multilayer RNNs.*

11.4.1. *The Model.*

11.4.2. *Exploding or Disappearing Activations.*

11.5. *LSTM.*

11.5.1. *Building an LSTM from Scratch.*

11.5.2. *Training a Language Model Using LSTMs.*

11.6. *Regularizing an LSTM.*

11.6.1. *Dropout.*

11.6.2. *Activation Regularization and Temporal Activation Regularization.*

11.6.3. *Training a Weight-Tied Regularized LSTM.*

11.7. *Conclusion.*

12. *Convolutional Neural Networks.*

12.1. *The Magic of Convolutions.*

12.1.1. *Mapping a Convolutional Kernel.*

12.1.2. *Convolutions in PyTorch.*

12.1.3. *Strides and Padding.*

12.1.4. *Understanding the Convolution Equations.*

12.2. *Our First Convolutional Neural Network.*

12.2.1. *Creating the CNN.*

12.2.2. *Understanding Convolution Arithmetic.*

12.2.3. *Receptive Fields.*

12.2.4. *A Note About Twitter.*

12.3. *Color Images.*

12.4. *Improving Training Stability.*

12.4.1. *A Simple Baseline.*

12.4.2. *Increase Batch Size.*

12.4.3. *1cycle Training.*

12.4.4. *Batch Normalization.*

12.5. *Conclusion.*

13. *ResNets.*

13.1. *Going Back to Imagenette.*

13.2. *Building a Modern CNN: ResNet.*

13.2.1. *Skip Connections.*

13.2.2. *A State-of-the-Art ResNet.*

13.2.3. *Bottleneck Layers.*

- 13.3. *Conclusion.*
- 14. *Application Architectures Deep Dive.*
 - 14.1. *Computer Vision.*
 - 14.1.1. *cnn_learner.*
 - 14.1.2. *unet_learner.*
 - 14.1.3. *A Siamese Network.*
 - 14.2. *Natural Language Processing.*
 - 14.3. *Tabular.*
 - 14.4. *Conclusion.*
- 15. *The Training Process.*
 - 15.1. *Establishing a Baseline.*
 - 15.2. *A Generic Optimizer.*
 - 15.3. *Momentum.*
 - 15.4. *RMSProp.*
 - 15.5. *Adam.*
 - 15.6. *Decoupled Weight Decay.*
 - 15.7. *Callbacks.*
 - 15.7.1. *Creating a Callback.*
 - 15.7.2. *Callback Ordering and Exceptions.*
 - 15.8. *Conclusion.*
- 16. *A Neural Net from the Foundations.*
 - 16.1. *Building a Neural Net Layer from Scratch.*
 - 16.1.1. *Modeling a Neuron.*
 - 16.1.2. *Matrix Multiplication from Scratch.*
 - 16.1.3. *Elementwise Arithmetic.*
 - 16.1.4. *Broadcasting.*
 - 16.1.5. *Einstein Summation.*
 - 16.2. *The Forward and Backward Passes.*
 - 16.2.1. *Defining and Initializing a Layer.*
 - 16.2.2. *Gradients and the Backward Pass.*
 - 16.2.3. *Refactoring the Model.*
 - 16.2.4. *Going to PyTorch.*
- 17. *CNN Interpretation with CAM.*
 - 17.1. *CAM and Hooks.*
 - 17.2. *Gradient CAM.*
 - 17.3. *Conclusion.*
- 18. *A fastai Learner from Scratch. Concluding Thoughts.*

18.1. Data.

18.1.1. Dataset.

18.2. Module and Parameter.

18.2.1. Simple CNN.

18.3. Loss.

18.4. Learner.

18.4.1. Callbacks.

18.4.2. Scheduling the Learning Rate.

18.5. Conclusion.

6 Individual Assignments

Students are expected to spend 3-4 hours a week outside of class on course material.

Політика та контроль

7 Course study rules

Students receive points relative to the correct and timely completion of coursework. The total grade consists of: 1) laboratories (programming assignments) 60%, 2) final exam 40%.

Presently there are three programming assignments, each worth up to 20% of the total grade. Student have to submit correctly fulfilled assignment during fortnight period from the date of give out to obtain full score for it, otherwise penalty points are applied but not more than 40% of the total score for laboratory work.

8 Assessment policy

How students are assessed: *modular test, programming assignments*

Calendar control: *not provided for part-time training.*

Term assessment: *final exam*

Term assessment admission condition: *all programming assignment submission, start score not less than 40 points.*

Exam scores map to the course grade according to the table:

Score	Grade
100-95	Excellent
94-85	Very Good
84-75	Good
74-65	Satisfactory
64-60	Sufficient
Less than 60	Unsatisfactory
Conditions for exam admission not met	Not allowed

9 Additional topics

- *Exam questions (see appendix).*

Syllabus:

Developed by Software Engineering in Energy Industry Department Professor, Sc. D. Andrii Sihaiov

Approved by Software Engineering in Energy Industry Department (minutes #28 on May 15, 2023)

Endorsed by Methodical Commission of Heat Power Faculty (minutes #9 on May 26, 2023)