Neural Networks and Deep Learning

Dr. Jerome J. Braun

This Lecture:
Course Description and Logistics

Course: Neural Networks and Deep Learning
IE 7615
Spring 2023

This Lecture

- Course description
- Course logistics

Course Objectives

After successfully completing this course, you should:

- **Have learned fundamental concepts of machine learning, neural networks and deep learning**
- **Have learned mathematical foundations of selected mainstream approaches and algorithmic paradigms in neural networks and deep learning**
- **Understand selected theoretical issues in neural networks and deep learning**
- **Understand the function, structure, and characteristics of neural networks**
- **Have learned to apply various neural networks to practical problems**
- **Be able to do hands-on work to design, implement, test, and use neural networks**
- **Have learned to design, implement, train, test, evaluate, and use neural networks**

Neural Networks and Deep Learning
Course Topics (tentative)

Topic numbers are not necessarily the order of lectures

1) Introduction and review of basic mathematical preliminaries.
2) Artificial neural networks fundamentals.
3) Probabilistic approaches, maximum likelihood principle, linear regression.
4) Information theory basics for deep learning.
5) Generative and discriminative approaches, logistic regression, linear discriminant analysis.
6) Regularization, Bayesian models.
7) Biological neural networks and selected neuroscience basics.
8) Multilayer perceptron networks.
9) Backpropagation and training of deep networks.
10) Computational graphs for deep networks.
11) TensorFlow, Keras, PyTorch.
12) Regularization for deep learning.
13) Convolutional neural networks (CNNs) and CNN training.
14) CNN architectures, AlexNet, VGGNet, ResNet, Inception, Xception.
15) Adaptive learning-rate algorithms, RMSProp, ADAM.
16) Deep learning in practice, practical advice for deep network training.
17) Recurrent neural networks, LSTMs, GRUs, encoder-decoder architectures.
18) Autoencoders.
19) Neural attention, transformers (Transformer neural networks), BERT
20) Representation learning, unsupervised and semi-supervised learning.
21) Deep generative models, generative adversarial networks (GANs).
22) Probabilistic graphical models for deep learning.
23) Deep-learning-related topics from statistical learning theory.
24) Selected additional topics in deep learning.

Neural Networks and Deep Learning
What We Will Study in This Course, and Why? (1 of 3)

- Modern AI (artificial intelligence) in reality is neural networks and deep learning
    - AI — currently a "street name" (media, popular press, advertising) for neural networks and deep learning
- Methods, techniques and algorithms of neural networks and deep learning are behind operation and successes of modern AI
- As such, neural networks and deep learning are indispensable to modern AI

Neural Networks and Deep Learning
What We Will Study in This Course, and Why? (2 of 3)

- **Neural networks and deep learning skills— among top in-demand skills now and in foreseeable future**
  - **Big and growing demand for neural-networks-and-deep-learning professionals**
    - **Companies across industry and commerce need this expertise for their AI applications**
    - **Trend will continue as AI continues to pervade more aspects of human activity**
  - **Numerous opportunities in industry and in academia**
- **High potential for "transformative" and "disruptive" advances**
- **BUT key to your success in neural networks and deep learning is learning the SCIENCE aspects of this field !**
  - **Implementation skills are NOT enough — it is NOT about being a "toolkit driver" !**
  - **You need to KNOW underlying theory and math behind code and software libraries (toolkits)**
    - **And you need to be able to demonstrate it (e.g., in job interviews)**

Neural Networks and Deep Learning
What We Will Study in This Course, and Why? (3 of 3)

- **To succeed in this field, you need to know theory and mathematics of neural networks and deep learning**
  - **I will teach you these in this course**
- **In this course, you will also gain practical hands-on experience**
  - **Course project (referred to as "final project")**

Learn mathematical and theoretical foundations
- **Learn science of neural networks**

Gain insights and intuitions

Learn techniques and algorithms

Learn the science behind modern deep-learning software-libraries, such as TensorFlow and Keras
- **To enable you to use these toolkits <u>knowledgeably</u> (wisely)**

Gain practical HANDS-ON experience
- **Course Project (referred to as "Final Project")**
  - **Learn to *implement* neural networks and deep-learning algorithms and techniques**
  - **Learn to use modern popular software-libraries such as**
    - **TensorFlow**
    - **Keras**
    - **PyTorch**

This Lecture

- Course description
- Course logistics

Main Study Sources

- My lectures
  - Notes you take during class
  - Discussions in class
- Lecture slides for <u>your</u> study in this course will be placed on the Canvas system (“Canvas”)
- Required textbook (see next slide “Books”)
- Check Canvas often
  - Lecture slides
  - Syllabus
  - Homework assignments
  - Announcements (in addition to announcements made in class)
  - Other course-related items

Books

Refer to course syllabus.

Required textbook:

Goodfellow, I., Bengio, Y., Courville, A., '*Deep Learning,*' MIT Press, 2016.

Supplemental references:

Bishop, C. M., '*Pattern Recognition and Machine Learning*,' Springer, 2006.

Hastie, T., Tibshirani, R. and Friedman, J., '*The Elements of Statistical Learning*,' 2nd Ed., Springer. 2017.

Haykin, S., '*Neural Networks and Learning Machines*,' 3rd Ed., Pearson, 2009.

Koller, D. and Friedman, N. '*Probabilistic Graphical Models*,' MIT Press. 2009.

Murphy, K.P., '*Machine Learning: A Probabilistic Perspective,*' MIT Press, 2012.

Duda, R.O., Hart, P.E., and Stork, D.G., '*Pattern Classification*,' Wiley, 2001.

Scholkopf B. and Smola A., '*Learning with Kernels,*' MIT Press, 2002.

Dayan, P. and Abbott, L.F., '*Theoretical Neuroscience: computational and mathematical modeling of neural systems,*' MIT Press, 2001.

Grading

Refer to course syllabus.

- Exams
  - Midterm exam: 17% of total course-grade
  - Final exam: 30% of total course-grade

  If either midterm exam or final exam is not administered *to entire class*, respective percentage weight of non-administered exam will be distributed to remaining grade-components — however please note that if exam is administered to class, redistribution option will NOT be available to any individual student(s)

  Participation in all administered exams is required
  Non-participation or receiving a zero-score in either the midterm exam or in the final exam when administered, may result (regardless of any other grade-components, e.g., scores on other exams, final project, etc.) in failing the course

- Homework problem-sets — 8% total
  - Self-graded, submit completed homework and grade, will be verified at random

- Final Project (40%)
  - More on this in subsequent slide and in subsequent lectures

- Class participation (5%)
  - Attendance, participation in discussions

Homework — Problem-Sets

- Problem-set may include mix of
  - Problems to solve (written)
  - Coding assignments
    - In Python
  - Homework — important part of your study, and helps you to prepare for project and exams
- Self-grading
  - You will submit your solutions (including code for coding assignments), and your own grade
  - We will randomly check to verify

Expectations, Policies, Student Academic Honesty and Integrity

Refer to course syllabus.

- Comply with <u>all</u> Northeastern University policies, college/department policies, and rules of this course
- Academic honesty and integrity
  - Cheating not tolerated — risk failing course and/or facing administrative actions
- All work on exams must be student's own work
- Homework assignments must be student's own work
  - Using past-years' solutions (from *any* source) is <u>prohibited</u>
  - Using websites to obtain solutions is <u>prohibited</u>
  - Students *may* engage in discussions of assignments with their peers <u>currently</u> enrolled in this course, and the TA or the instructor, provided that:
    - Following such interactions each student performs the assignment independently on his/her own (without referring to and/or copying notes from those discussions)
- Refer to syllabus of this course and comply with all rules listed in that syllabus

Final Project

Refer to course syllabus.

- Final Project is mandatory
- Students are expected to form self-organized teams of up to maximum of three students per team (team-members within each team collaborating to produce respective team's project)
- Mandatory project-related items include
  - Project proposal submission, project proposal presentation, proposal acceptance (proposals need to be approved by me), final paper submission, final presentation
    - All of these are mandatory <u>no later than</u> on their respective due-dates
      - These due-dates will be announced well ahead of time
  - Project-specific deliverables, in accordance with any particular accepted project-proposal
- Projects can be either application-oriented (most common) or theoretical
  - Datasets used must be from reputable public machine-learning research repositories, and must be available without restrictions
- Project proposals subject to acceptance and approval by me

- Detailed *Project Guidelines* will be provided, and will be covered in class.
- First part of Project Guidelines will describe project-proposal preparation in detail, and will also be covered in detail in class.

Submitting Course-Deliverables into Canvas

- Course-deliverables will be submitted into Canvas ONLY
  - E.g., homework assignments, any and all final-project deliverables
- Any other form of submission is <u>not</u> allowed
  - Submissions by email are <u>not</u> allowed

Course Prerequisites

- Knowledge of fundamentals of
  - Probability
  - Basic linear algebra (matrix algebra)
  - Calculus

Slideset covering review of some of these will be provided

- At least rudimentary knowledge of Python programming language
  - Knowledge of Matlab or other programming languages such as C/C++ could be an additional advantage

Making up any deficiencies is, obviously, individual responsibility of each student

Questions?