IN5550: Neural Methods in Natural Language Processing
Introduction

Andrey Kutuzov, Vladislav Mikhailov, Meihan Tong,
Shenbin Qian, David Samuel & You

University of Oslo

January 19, 2026



Contents

1. Introduction
2. Paradigm shifts in NLP and AI
3. From feature engineering to representation learning
4. Neural methods in NLP
5. Language models
6. Course logistics

Introduction

Who we are?



Andrey Kutuzov, Language Technology Group



David Samuel, Language Technology Group



Meihan Tong, Language Technology Group, Integreat

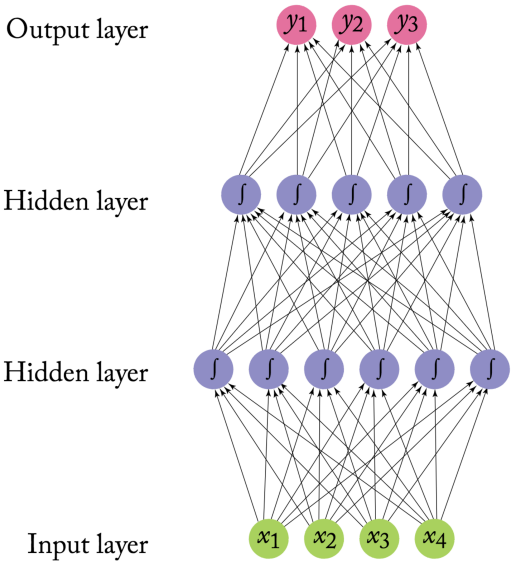


Vladislav Mikhailov, Language Technology Group



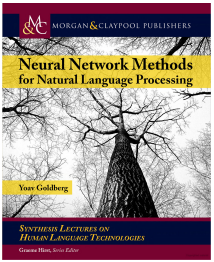
Shenbin Qian, Language Technology Group

What is a neural network (NN) model?



- Artificial neural networks (NNs) are a family of powerful machine learning models.
- Weakly based on the metaphor of a human neuron.
- Non-linear transformations of the input in the 'hidden layers'.
- Learns not only to make predictions, but also to represent the data
- 'Deep Learning': NNs with several hidden layers.

Knowledge sources for IN5550 (syllabus)



Speech and Language Processing (3rd ed. draft)

Dan Jurafsky and James H. Martin

Here's our Jan 6, 2026 release!

This release has a mostly-cosmetic and bug-fixing release, with some updated figures for the transformer in various chapters. The August release made larger changes, including DPO in chapter 9, new ASR and TTS-restructured LLM chapter, and a couple in Chapter 3. Individual chapters and updated slides are below:

Here is a (new) pdf of Jan 6, 2026 draft

1. **Feel free to use the draft chapters and slides in your classes**, print it out, whatever, the resulting feedback we get from you makes the book better!
2. **Typos and comments** are very welcome (just email slp3@stanford.edu) and let us know the date on the draft! (Don't bother reporting missing refs due to cross-chapter cross-reference problems in the draft pdfs, those are fixed in the full book draft!)
3. **Contributors**: We've had a lot of the amazing people who have sent us many fantastic suggestions and bug-fixes for improving the book. We are really grateful to all of you for your help, the book would not t without all of you.
4. **How to cite the book:**

Daniel Jurafsky and James H. Martin. 2026. Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition with Language Models, 3rd ed manuscript released January 6, 2026. https://web.stanford.edu/~jurafsky/slp3/

5. **A bib entry** for the book is here:

```
@book{jurafsky_martin_slp3_draft,
    author = "Daniel Jurafsky and James H. Martin",
    title = "Speech and Language Processing: An Introduction to Natural Language Processing,
             Computational Linguistics, and Speech Recognition,
             with Language Models",
    year = "2026",
    note = "https://web.stanford.edu/~jurafsky/slp3/",
    edition = "3rd ed manuscript released January 6, 2026",
    address = "Stanford",
}
```

6. **When** will the book be finished? Don't ask.
7. If you need the previous Aug 2025 draft chapters, they are here; if you need the previous Jan 2025 draft chapters, they are here.

Volume 1: Large Language Models

Chapter	Slides
1 Introduction	
2 Words and Tokens	2: Words and Tokens [send] [pdf] 2: Edit Distance [send] [pdf]
3 N-gram Language Models	3 [send] [pdf]
4 Logistic Regression and Text Classification	4 [send] [pdf]
5 Embeddings	5 [send] [pdf]
6 Neural Networks	6 [send] [pdf]
7 Large Language Models	7 [send] [pdf]
8 Transformers	8 [send] [pdf]

- The field is developing rapidly: no definitive textbook exists
- *'Neural Network Methods for Natural Language Processing'* [Goldberg, 2017]
  - For more foundational topics
  - Free e-version available through UiO; `https://oria.no/`
- *'Speech and Language Processing'* by Dan Jurafsky and James Martin (online, 2026)
  - For more advanced and recent topics
  - `https://web.stanford.edu/~jurafsky/slp3/`
- Supplementary research papers will be provided.
- Check the course schedule for the syllabus for each lecture!

Contents

1 Introduction

2 Paradigm shifts in NLP and AI

3 From feature engineering to representation learning

4 Neural methods in NLP

5 Language models

6 Course logistics

Paradigm shifts in NLP and AI

- 50s–80s: mostly rule-based (symbolic / rationalist) approaches.
- Hand-crafted formal rules and manually encoded knowledge.
- (Though some AI research on neural networks in the 40s and 50s).

Paradigm shifts in NLP and AI

- 50s–80s: mostly rule-based (symbolic / rationalist) approaches.
- Hand-crafted formal rules and manually encoded knowledge.
- (Though some AI research on neural networks in the 40s and 50s).
- Late 80s: success with statistical ('empirical') methods in the fields of speech recognition and machine translation.
- Late 90s: massive shift towards machine learning (ML).
  - Based on automatically inferring statistical patterns from data.
- 00s: Machine learning methods dominant.

Paradigm shifts in NLP and AI

- 50s–80s: mostly rule-based (symbolic / rationalist) approaches.
- Hand-crafted formal rules and manually encoded knowledge.
- (Though some AI research on neural networks in the 40s and 50s).
- Late 80s: success with statistical ('empirical') methods in the fields of speech recognition and machine translation.
- Late 90s: massive shift towards machine learning (ML).
  - Based on automatically inferring statistical patterns from data.
- 00s: Machine learning methods dominant.
- 2010–present: neural methods increasingly replacing traditional ML.

Paradigm shifts in NLP and AI

- 50s–80s: mostly rule-based (symbolic / rationalist) approaches.
- Hand-crafted formal rules and manually encoded knowledge.
- (Though some AI research on neural networks in the 40s and 50s).
- Late 80s: success with statistical ('empirical') methods in the fields of speech recognition and machine translation.
- Late 90s: massive shift towards machine learning (ML).
  - Based on automatically inferring statistical patterns from data.
- 00s: Machine learning methods dominant.
- 2010–present: neural methods increasingly replacing traditional ML.
- Many fundamental neural methods were proposed in the **80s and 90s** like recurrent and convolutional networks, but due to computational power and data scarcity, they only got great breakthroughs after the 2010s.
- Today, AI is almost synonymous with ML, and ML synonymous with NNs/Deep learning.

Contents

1 Introduction

2 Paradigm shifts in NLP and AI

3 From feature engineering to representation learning

4 Neural methods in NLP

5 Language models

6 Course logistics

What is a machine learning classifier?

- Given labeled examples $\{(x_i, y_i)\}_{i=1}^{N}$, a classifier learns a function $f_\theta : x \mapsto y$.
- Example (NLP): spam detection.

Mini example: spam detection

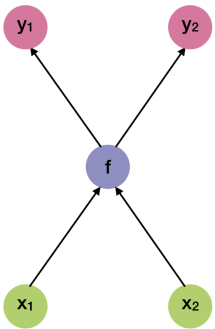
Spam (x_1)

```
Congrats!  You won a FREE iPhone.  Click to claim.
```

Ham (x_2)

```
Hi!  The meeting is moved to 10:15.  See you in Zoom.
```

$y \in \{\text{spam}, \text{ham}\}$



From raw text to a classifier: why representations?

Key idea

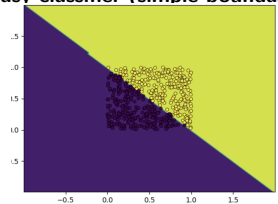
- Classifiers operate on vectors (numbers), not strings.
- We map raw text x to a fixed-size representation $\phi(x) \in \mathbb{R}^d$.
- Then we predict a discrete label:

$$\hat{y} = g_\theta(\phi(x)).$$

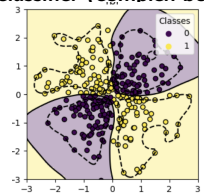
- Representation matters: good $\phi(x) \Rightarrow$ simple boundary; poor $\phi(x) \Rightarrow$ complex boundary.

"A good representation does most of the work."
[Anderson and Rosenfeld, 1988]

Easy classifier (simple boundary)



Hard classifier (complex boundary)

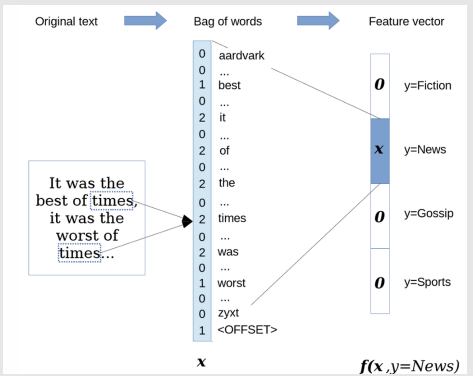


Representation in ML - Feature Engineering

Key idea

- Manual design of features for automatic extraction.
- Performance depends strongly on the task and dataset.
- Examples: bag-of-words, n-grams, POS, dependency features.
- Drawback: feature vectors are often sparse (poor feature sharing and data inefficiency); features that work well for one dataset/task often need to be redesigned for another.

Bag-of-Words



Representation in NN: Representation Learning

- Neural NLP models can learn dense representations from data, enabling better feature sharing and transfer across tasks.

Representation in NN: Representation Learning

- Neural NLP models can learn dense representations from data, enabling better feature sharing and transfer across tasks.

- **Representation Learning(Word embeddings)**
  - **Training:** pre-train on unlabeled text with prediction objectives.
  - **Benefits:** dense, low-dimensional vectors for words.
  - **Usage:** reuse as input features for downstream supervised tasks.

Distributional hypothesis

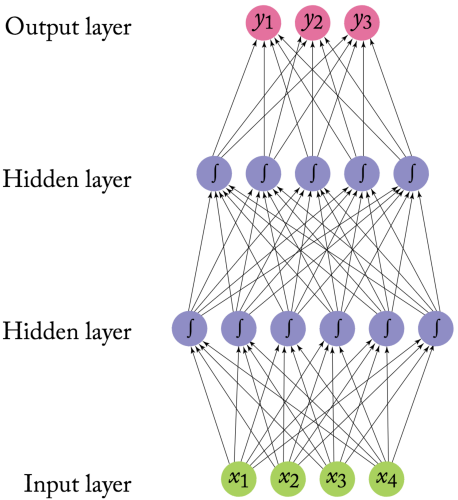
Words occurring in similar contexts tend to have similar meanings.

Example (minimal pair)

`The students submitted their assignments yesterday.`

`The scholar submitted their assignments yesterday.`

Pros and Cons of Representation Learning



Pros

- Less manual feature engineering: representations are learned from data.
- Dense vectors share statistical evidence across inputs $\Rightarrow$ better generalization and data efficiency.
- Often scales well with more data and compute.

Cons

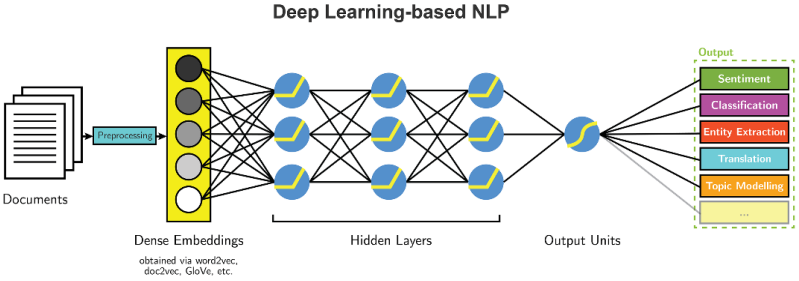
- Can be less interpretable: hard to explain what each dimension "means".
- Often more complex and parameter-heavy $\Rightarrow$ higher compute/memory cost.

Contents

1. Introduction
2. Paradigm shifts in NLP and AI
3. From feature engineering to representation learning
4. Neural methods in NLP
5. Language models
6. Course logistics

Neural methods in NLP

- Most NLP tasks are approached as classification problems, using supervised machine learning
  - ...but recently also as text generation.
- Since at least 2014, neural networks completely dominating.
- Very successful on standard NLP benchmarks.



Neural NLP tasks

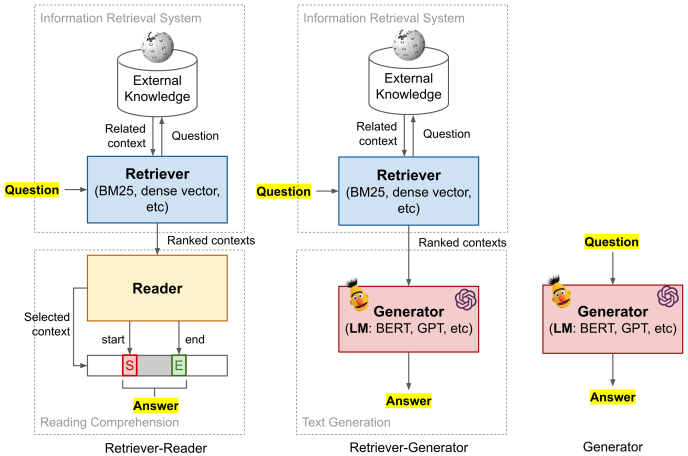
- Text classification:
  - Sentiment analysis;
  - Spam detection;
  - Language identification.
- Sequence labeling:
  - Named entity recognition;
  - PoS-tagging.
- Natural language generation:
  - Summarization;
  - Paraphrase generation;
  - Headline generation;
  - Machine translation.



https://blog.xiteb.com/
best-natural-language-processing-models-available-for-nl

Neural NLP tasks

Question answering



(https://lilianweng.github.io/posts/2020-10-29-odqa/)

Neural NLP tasks

Visual question answering

Who is wearing glasses?

man



woman

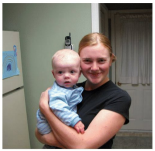


Where is the child sitting?

fridge

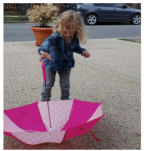


arms

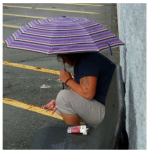


Is the umbrella upside down?

yes



no



How many children are in the bed?

2



1



(http://visualqa.org/)

Neural NLP tasks

Image captioning

a mexican food sign on the side of a building



a group of people standing next to a brick building



an old photo of a man in a military uniform



a man kneeling down next to a dog in a field



an old tea pot shaped building on the side of the street



a painting of a young woman with a black shirt on



(https://towardsdatascience.com/using-openclip-for-image-search-and-automatic-captioning-fa1cbbd48ce4/)

Neural NLP tasks

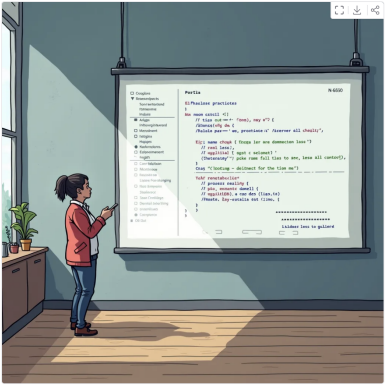
Text-to-image generation

FLUX.1 [dev]

12B param rectified flow transformer guidance-distilled from FLUX.1 [pro]
[non-commercial license] [blog] [model]

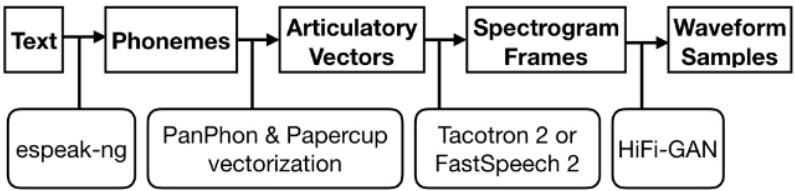
Create an illustration of the IN5550 (Neural Methods

Run



Neural NLP tasks

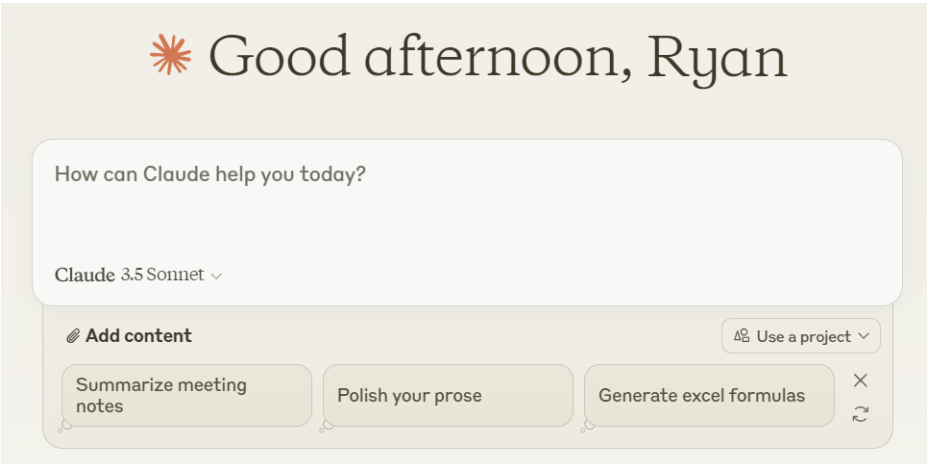
Text-to-Speech:



[Lux and Vu, 2022]

Neural NLP tasks

Dialog systems:



https://claude.ai

Neural NLP tasks

To sum up

- Neural ('deep learning') methods have caused great advances in the field of natural language processing, image processing and signal processing.
- All of these have converged to the same neural architecture and learning pipelines.
- We will focus only on text $\rightarrow$ but you can easily use these methods in other domains.

Neural NLP tasks

To sum up

- Neural ('deep learning') methods have caused great advances in the field of natural language processing, image processing and signal processing.
- All of these have converged to the same neural architecture and learning pipelines.
- We will focus only on text $\rightarrow$ but you can easily use these methods in other domains.

...and the most well known gems of modern neural NLP are large language models (LLMs).

Contents

1 Introduction

2 Paradigm shifts in NLP and AI

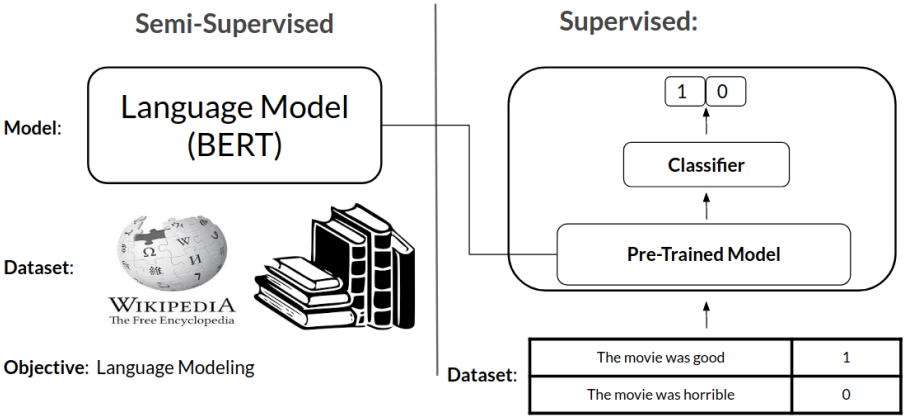
3 From feature engineering to representation learning

4 Neural methods in NLP

5 Language models

6 Course logistics

“Old-school” neural NLP system (finetuning)



(https://blog.insightdatascience.com)

Masked Language models

- Given raw (i.e. unlabeled) text as input,
- Masked language Models (MLMs) are trained to predict masked out parts of a sentence

A quick brown ??? jumps over the ??? dog

- 'Self-supervised'
- Neural MLMs are used as representation learners;

“New-school” neural NLP system (prompting)

ENG: She is making a mountain out of a molehill.
NOB: Hun gjør en mygg til en elefant.

ENG: Do you remember that baffling murder case?
NOB: Husker du den utrolige mordsaken?

ENG: What’s the best way to get there?
NOB: Hva er beste måte å komme seg dit på?

ENG: They say that Tom won’t do that.
NOB:

“New-school” neural NLP system (prompting)

ENG: She is making a mountain out of a molehill.
NOB: Hun gjør en mygg til en elefant.

ENG: Do you remember that baffling murder case?
NOB: Husker du den utrolige mordsaken?

ENG: What’s the best way to get there?
NOB: Hva er beste måte å komme seg dit på?

ENG: They say that Tom won’t do that.
NOB: De sier at Tom ikke kommer til å gjøre det.

Autoregressive language models

- Again, given only raw text as input,
- Autoregressive or Causal language Models (CLMs) are trained to predict the next word

A

Autoregressive language models

- Again, given only raw text as input,
- Autoregressive or Causal language Models (CLMs) are trained to predict the next word

A quick

Autoregressive language models

- Again, given only raw text as input,
- Autoregressive or Causal language Models (CLMs) are trained to predict the next word

A quick brown

Autoregressive language models

- Again, given only raw text as input,
- Autoregressive or Causal language Models (CLMs) are trained to predict the next word

A quick brown fox

Autoregressive language models

- Again, given only raw text as input,
- Autoregressive or Causal language Models (CLMs) are trained to predict the next word

A quick brown fox jumps

Autoregressive language models

- Again, given only raw text as input,
- Autoregressive or Causal language Models (CLMs) are trained to predict the next word

A quick brown fox jumps over the lazy dog

- These models are not great representation learners, but their ability to generate text gives them a big flexibility
- We can combine finetuning with prompting to teach CLMs to follow instructions

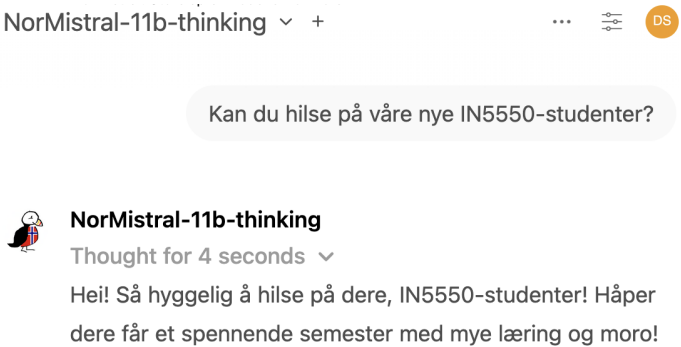
Language models

Our group has been training language models for many years



Language models

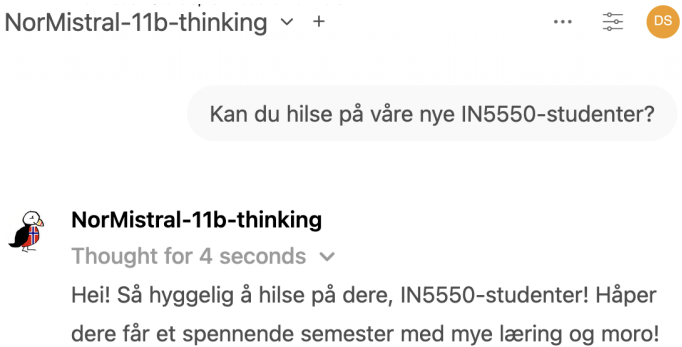
Our group has been training language models for many years



https://chat.llm.sigma2.no/

Language models

Our group has been training language models for many years



`https://chat.llm.sigma2.no/`

- Now, we have a collection of Norwegian LMs of different flavors and sizes
- ...and pre-trained LMs and Transformers lectures are coming later in the course

Contents

1 Introduction

2 Paradigm shifts in NLP and AI

3 From feature engineering to representation learning

4 Neural methods in NLP

5 Language models

6 Course logistics

Getting to know each other

We need to know who you are!

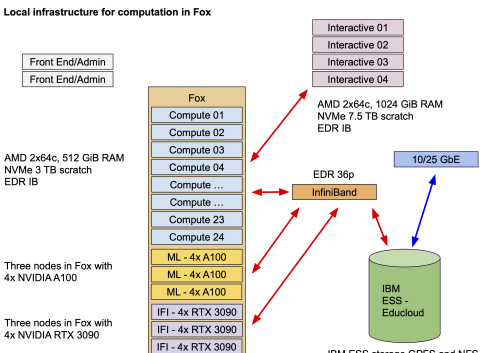
A brief anonymous survey:

https://nettskjema.no/a/582613

Course content

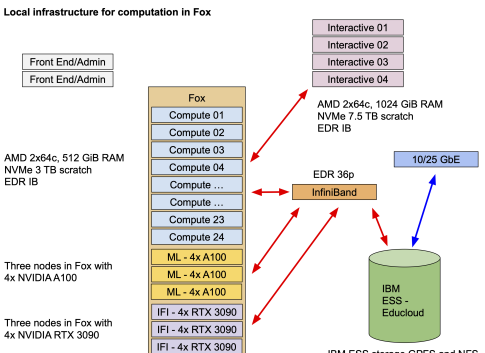
- Training deep neural networks;
- Dense feature representations and word embeddings;
- Language modeling, contextualized embeddings;
- Attention mechanism, Transformer;
- Modern masked and generative Transformer-based language models;
- Generation, finetuning and prompting;
- Evaluation and benchmarking
- Trustworthiness and Green AI;
- Mechanistic interpretability;
- a guest lecture and an invited lecture from the industry (stay tuned)

Scientific programming in IN5550



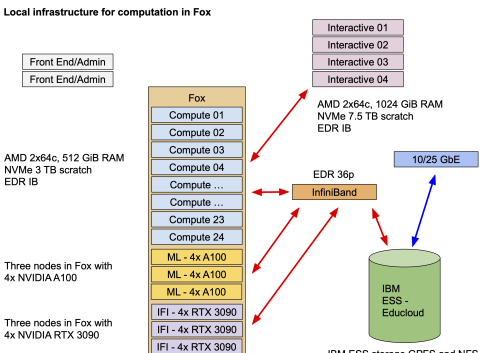
- Strong practical and programming elements in this course;

Scientific programming in IN5550



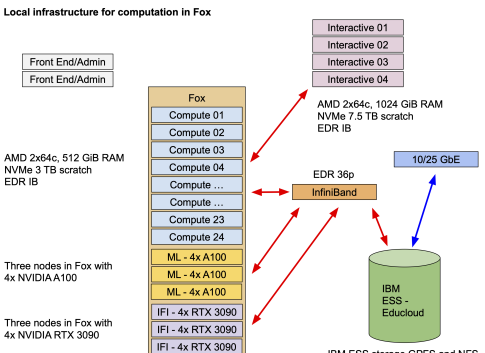
- Strong practical and programming elements in this course;
- three obligatory assignments; all predominantly ‘hands-on’;

Scientific programming in IN5550



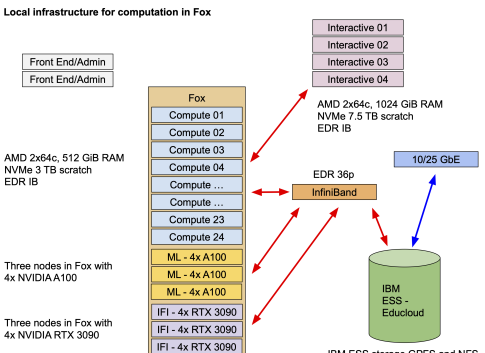
- Strong practical and programming elements in this course;
- three obligatory assignments; all predominantly 'hands-on';
- representative of sub-problems in typical MSc theses at the Language Technology Group (LTG);

Scientific programming in IN5550



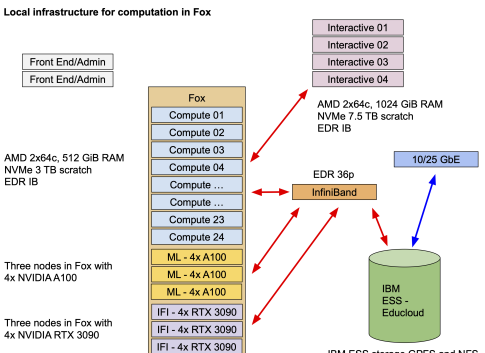
- Strong practical and programming elements in this course;
- three obligatory assignments; all predominantly 'hands-on';
- representative of sub-problems in typical MSc theses at the Language Technology Group (LTG);
- fairly data- and compute-intensive throughout the semester;

Scientific programming in IN5550



- Strong practical and programming elements in this course;
- three obligatory assignments; all predominantly 'hands-on';
- representative of sub-problems in typical MSc theses at the Language Technology Group (LTG);
- fairly data- and compute-intensive throughout the semester;
- learn how to use the UiO Fox computing cluster

Scientific programming in IN5550



- Strong practical and programming elements in this course;
- three obligatory assignments; all predominantly 'hands-on';
- representative of sub-problems in typical MSc theses at the Language Technology Group (LTG);
- fairly data- and compute-intensive throughout the semester;
- learn how to use the UiO Fox computing cluster (3,000$^+$ CPUs and GPUs);

Deep learning through Python



- Python is a relatively simple programming language with great popularity for neural network–based machine learning;

Deep learning through Python



- Python is a relatively simple programming language with great popularity for neural network–based machine learning;
- it provides a very convenient, high-level coding with a gentle learning curve; works easily across different platforms;

Deep learning through Python



- Python is a relatively simple programming language with great popularity for neural network–based machine learning;
- it provides a very convenient, high-level coding with a gentle learning curve; works easily across different platforms;
- comprehensive standard library; ecosystem of community-maintained add-on modules with specialized (and optimized) functionality;

Deep learning through Python



- Python is a relatively simple programming language with great popularity for neural network–based machine learning;
- it provides a very convenient, high-level coding with a gentle learning curve; works easily across different platforms;
- comprehensive standard library; ecosystem of community-maintained add-on modules with specialized (and optimized) functionality;
- pretty much everything open-source; we provide ready-to-use ‘virtual lab’ on Fox; in principle possible to reproduce it ‘at home’ (for development).

A complex of interoperable modules







- NumPy for efficient multi-dimensional arrays (a.k.a. ‘tensors’) and general linear algebra;

A complex of interoperable modules







- NumPy for efficient multi-dimensional arrays (a.k.a. ‘tensors’) and general linear algebra;
- PyTorch (by Meta) is a mature software infrastructure for deep learning (built natively in C^{++});
  - open source $\rightarrow$ community involvement;

A complex of interoperable modules







- NumPy for efficient multi-dimensional arrays (a.k.a. 'tensors') and general linear algebra;
- PyTorch (by Meta) is a mature software infrastructure for deep learning (built natively in C^{++});
  - open source $\rightarrow$ community involvement;
- HuggingFace Transformers for easy handling of pre-trained models and datasets;

A complex of interoperable modules







- NumPy for efficient multi-dimensional arrays (a.k.a. ‘tensors’) and general linear algebra;
- PyTorch (by Meta) is a mature software infrastructure for deep learning (built natively in C^{++});
  - open source $\rightarrow$ community involvement;
- HuggingFace Transformers for easy handling of pre-trained models and datasets;
- all integrated in standard modules on Fox (see the course page).

Lectures

No traditional lectures (except this one and guest lectures)

- Pre-recorded short videos;
- Published: every Wednesday afternoon (starting this week);
- watch whenever convenient;
- find them in the course schedule in the corresponding lecture entries;
- every Monday at 10:15: interactive Q&A in Zoom.



Labs and obligatory assignments (*grupper & obliger*)

- Labs or 'group sessions': Wednesday, 10:15–12:00 (starting this week!);

Labs and obligatory assignments (*grupper & obliger*)

- Labs or 'group sessions': Wednesday, 10:15–12:00 (starting this week!);
- Three obligatory assignments;
- minimum 1p required per assignment;
- minimum 56% of points across all three required to qualify for exam (63% for PhD students), i.e. 18 and 20 points respectively.



Labs and obligatory assignments (*grupper & obliger*)

- Labs or 'group sessions': Wednesday, 10:15–12:00 (starting this week!);
- Three obligatory assignments;
- minimum 1p required per assignment;
- minimum 56% of points across all three required to qualify for exam (63% for PhD students), i.e. 18 and 20 points respectively.



Mechanics of the obligatory assignments

- No re-submissions, rigid schedule:
- `https://www.uio.no/studier/emner/matnat/ifi/IN5550/v26/obligatory-assignments/`
- Group work warmly encouraged (1–3 students); please give it a shot!

Home Exam

Final exam: baby research project in May

- Our guiding metaphor: preparing a scientific paper for publication.
- again, team work possible (for MSc students).

Home Exam

Final exam: baby research project in May

- Our guiding metaphor: preparing a scientific paper for publication.
- again, team work possible (for MSc students).

8th IN5550 Workshop on Neural NLP (WNNLP 2026)

Home Exam

Final exam: baby research project in May

- Our guiding metaphor: preparing a scientific paper for publication.
- again, team work possible (for MSc students).

8th IN5550 Workshop on Neural NLP (WNNLP 2026)

Process

(1) Experimentation

(2) Analysis

(3) Paper Submission

(4) Reviewing

(5) Camera-Ready Submission

(6) Presentation



For Example: The Annual ACL Conference

Main Conference Papers

Submission template available	~~November 1, 2022~~
Anonymity period for ARR papers	~~November 15, 2022~~
Submission deadline for papers submitted to ARR	~~December 15, 2022~~
Anonymity period for papers submitted through Softconf START	~~December 20, 2022~~
Abstract deadline for Softconf START direct submissions	January 13, 2023
Direct paper submission deadline	January 20, 2023
Anonymity period for ARR papers to be committed to ACL 2023	Februrary 17, 2023
Commitment deadline for ARR papers	March 17, 2023
Author response period	March 17-24, 2023
Notification of acceptance	May 1, 2023
Withdrawal deadline	May 8, 2023
Camera-ready papers due	May 22, 2023

:

Same here: WNNLP 2026 (IN5550 Final Exam)

Three weeks for preparing a paper; 10 days for reviewing and revision.

`https://www.uio.no/studier/emner/matnat/ifi/IN5550/v26/final-exam/`

Home Exam

Our timeline in 2026

- Track presentations: April 27th
- Home Exam: April 27th – May 20th
- Paper submission – May 20th
- Review period: May 21st – May 25th
- Acceptance decisions: May 27th
- Camera Ready submission (if accepted): May 28th
- Workshop with oral presentations: June 1st

Proceedings of the Sixth IN5550 Workshop on Neural Natural Language Processing (WNNLP 2024)

Andrey Kutuzov, David Samuel, Vladislav Mikhailov, Roxana Pop and Sondre Wold (editors)

June 03, 2024

University of Oslo, Norway

Course Communication

- Questions?

Course Communication

- Questions?
  - On-line chat room: https://mattermost.uio.no/ifi-undervisning/channels/in5550
  - GitHub issues: https://github.uio.no/in5550/2026/issues
  - email: in5550-help@ifi.uio.no reaches all course staff.
- submissions, solutions, etc. on the GitHub instance at UiO:
  - https://github.uio.no/in5550/2026/

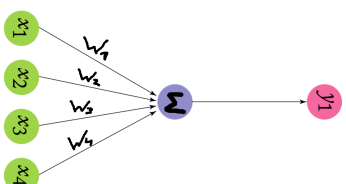
Course Communication

- Questions?
  - On-line chat room: https://mattermost.uio.no/ifi-undervisning/channels/in5550
  - GitHub issues: https://github.uio.no/in5550/2026/issues
  - email: in5550-help@ifi.uio.no reaches all course staff.
- submissions, solutions, etc. on the GitHub instance at UiO:
  - https://github.uio.no/in5550/2026/
- Messages:
  - Read your UiO email regularly;
  - Check the course page and schedule regularly;
  - Participate in the discussions on Mattermost and GitHub

Next week (lecture recording available this Wednesday)

Foundations

- Introducing mathematical notation for describing neural and not-neural classifiers.
- From linear regression to feed-forward networks and multi-layer perceptrons.
- Linear vs. non-linear decision boundaries



References I

Anderson, J. A. and Rosenfeld, E. (1988).
Neurocomputing: Foundations of Research.
MIT Press.

Goldberg, Y. (2017).
Neural network methods for natural language processing.
Synthesis Lectures on Human Language Technologies, 10(1):1–309.

Lux, F. and Vu, T. (2022).
Language-agnostic meta-learning for low-resource text-to-speech with articulatory features.

In Muresan, S., Nakov, P., and Villavicencio, A., editors, Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 6858–6868, Dublin, Ireland. Association for Computational Linguistics.