Seminar: Deep Learning for Molecular Biology

Alice McHardy, Giorgos Kallergis, Mohammad Hadi Foroughmand Araabi

Helmholtz Center for Infection Research & TU Braunschweig
Summer 2025

Seminar overview

Max. number of participants: 10

Language: English

Requirements:

- Student pairs must give both practical (implementation in Python) and theoretical presentations.
- At least one meeting with the assistants.
- >5 page summary of the topic, with scientific report template, for example with literature references (to be sent two weeks before seminar date)
- approx. 15 minutes of presentation per person, plus 10 discussion

Designated for Bachelor and Master students of Computer Science.

Course Takeaways

Basic Knowledge of Machine Learning: Understanding of fundamental concepts in machine learning.

- Machine Learning:
  a. A hot topic in both scientific research and industry applications.
  b. Wide-ranging impact across various domains, from healthcare to finance.

Basic Knowledge of Deep Learning: Familiarity with the principles of deep learning.

- Deep Learning:
  - Specialized models built upon neural network architectures.
  - Remarkable success in tackling complex and challenging problems across domains.

Basic Knowledge of Bioinformatics

- Understanding of bioinformatics principles and applications.
- Awareness of how machine learning and deep learning are applied in bioinformatics research.

Experience in Problem Solving:

- Practical experience in applying machine learning and deep learning techniques to bioinformatics problems.

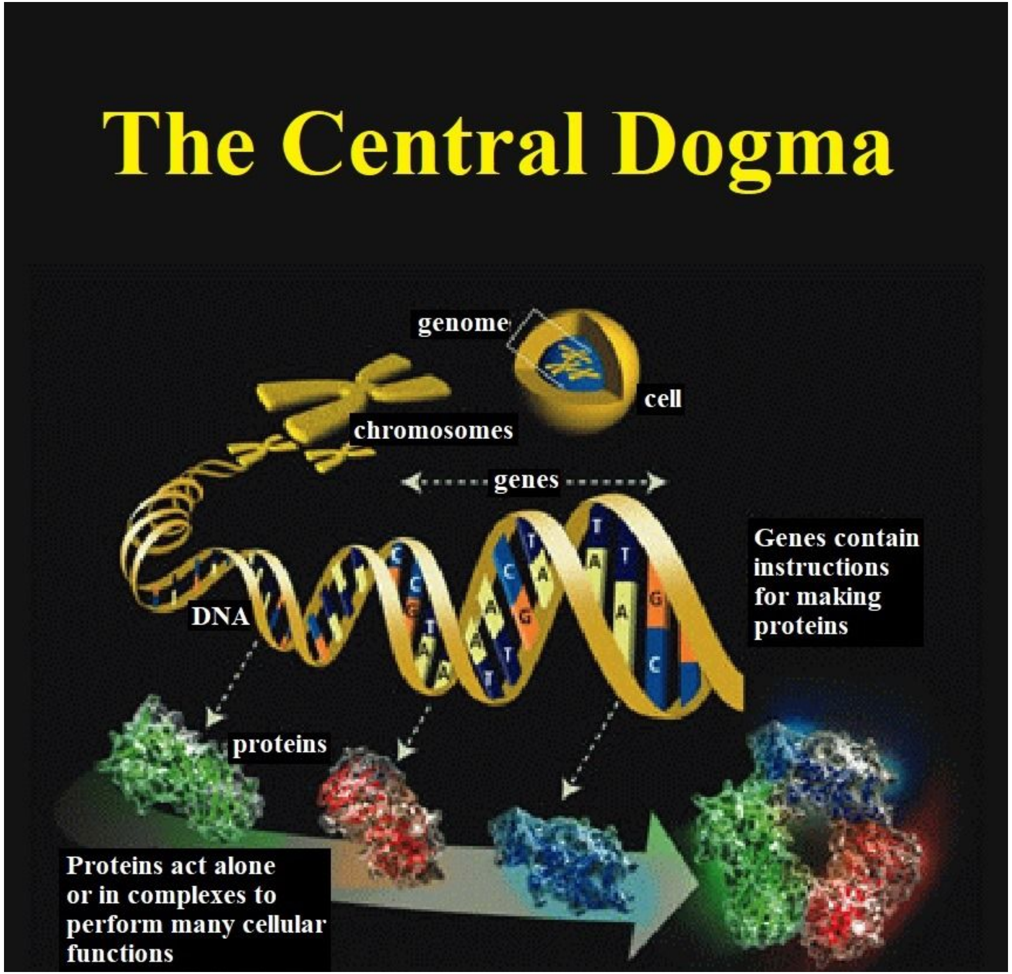
Introduction to Bioinformatics, Machine Learning, and Deep Learning

What is bioinformatics

(Current) Biology =

- DNA + RNA + Protein + Interactions

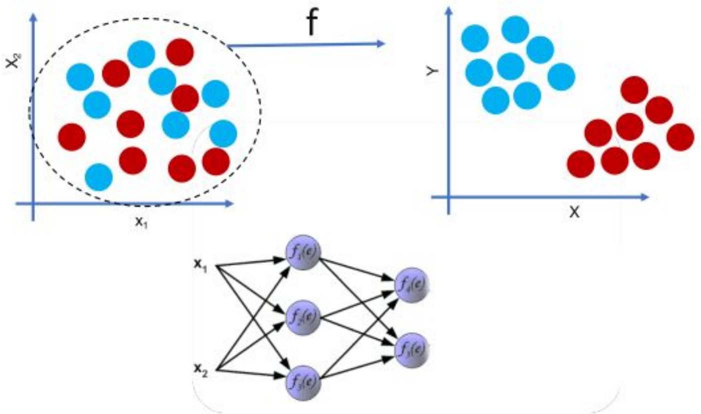
Bioinformatics: Computational analysis of the biological data



Machine Learning & Representation learning

Machine Learning is a subset of artificial intelligence focused on developing algorithms that enable systems to learn patterns, representations, or behaviors from empirical data

- Constructs models that generalize from training data to unseen instances by optimizing a loss function
- Includes supervised, unsupervised, and reinforcement learning, with applications in classification, regression, clustering, and decision-making.

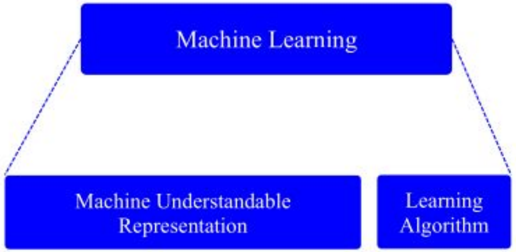


Representation learning:

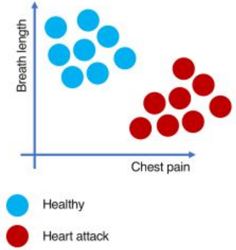
- Data can be represented as points in n-dimensional space (of features)
- Learns hierarchical or latent representations that capture underlying structures in data
    - Then, we can solve several problems, e.g. classification.

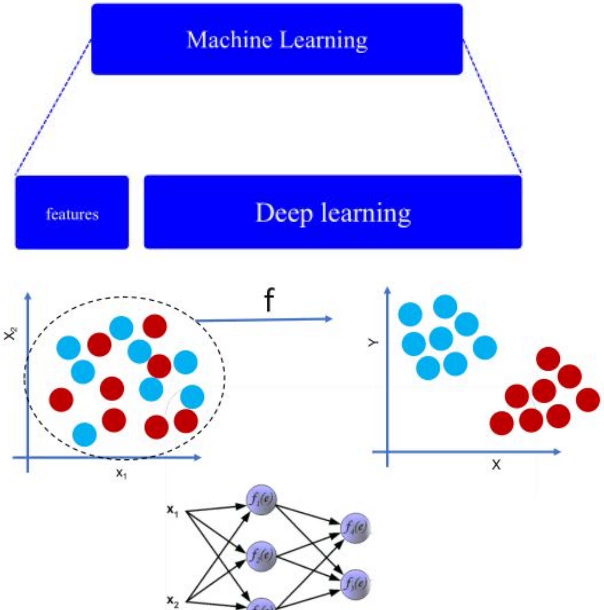
What is machine learning

- We alter data appearance to be interpretable by the audience.
  - Machine as the audience? Numerical values, vectors, matrices
- Finding a proper representation has been critical in machine learning



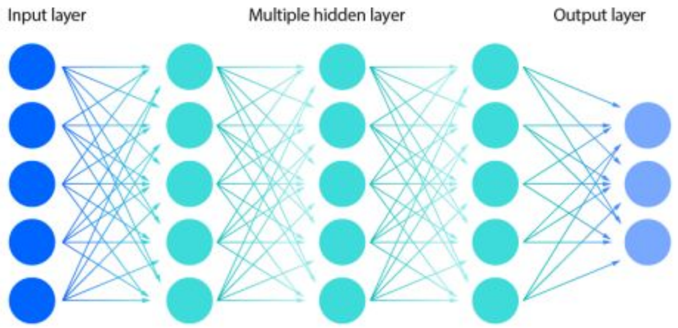
Domain knowledge
for designing representation



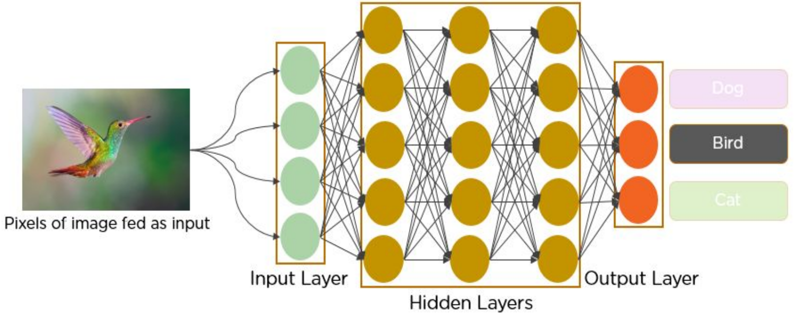


Specific ML models: NN, RNN, CNN

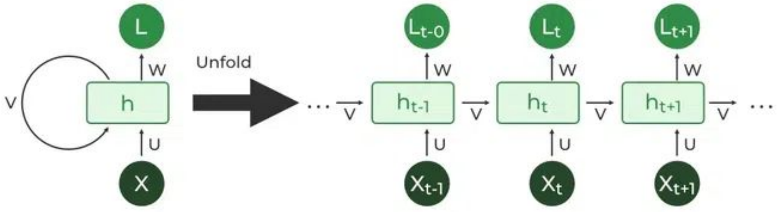
Neural network (NN): Inspired by brain



Convolutional Neural Network (CNN): NN with fewer (redundant) parameters

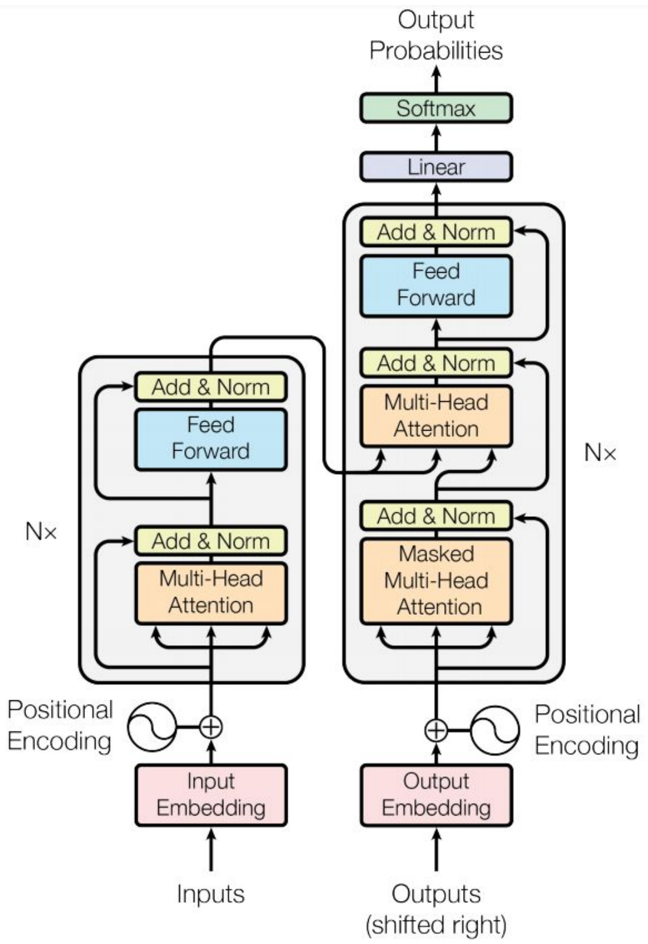


Recurrent Neural Network (RNN): Dealing with sequential information



What is a transformer?

- A specific architecture with breakthrough performances
- Encoder/Decoder architecture
- Used in a wide range of applications
- Effective
- Highly parallelizable
- Ideal for transfer learning



Transformers model architecture.

Vaswani et al, Attention is all you need. (NIPS 2017)

Applications of deep learning

Computer Vision

Natural Language Processing

Bioinformatics

Advertising

…

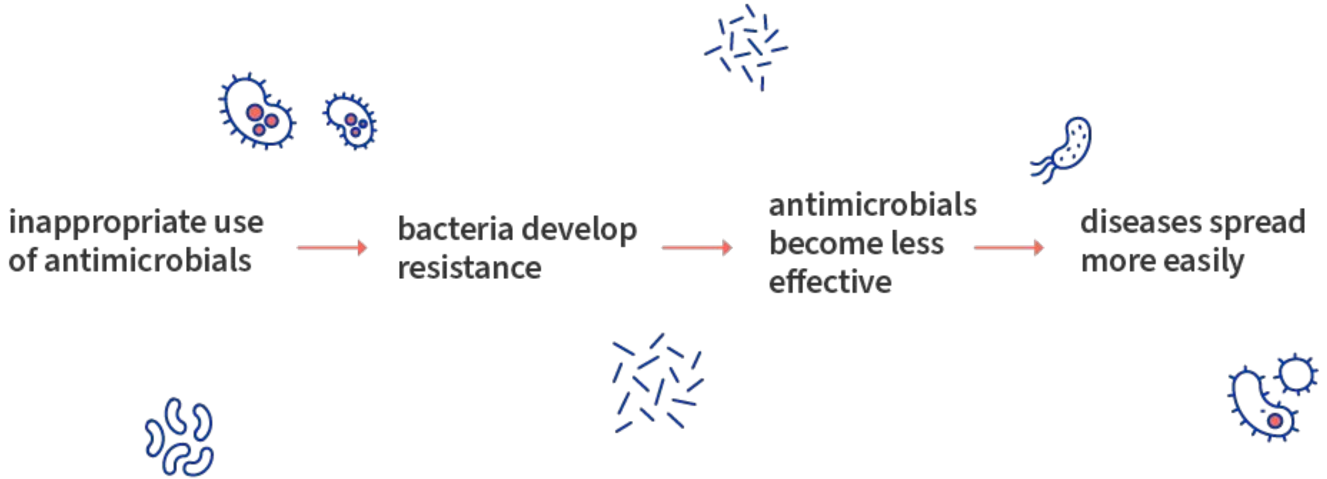
Your task

- Find a partner to form a group
- Choose a topic (from provided list)
- Study the topic
    - Some literature is provided
    - Meet and consult with the lecturers (at least once)
    - Implement and test your topic with the dataset (if applicable)
    - Evaluate (Metrics, precision/recall, TP, TN, ...)
- Create a written report
    - **Using AI tools is permitted; please detail their use in the acknowledgements section!**
- Present your topic on the presentation day

Implementation Tasks

Antimicrobial Resistance of pathogens (AMR)

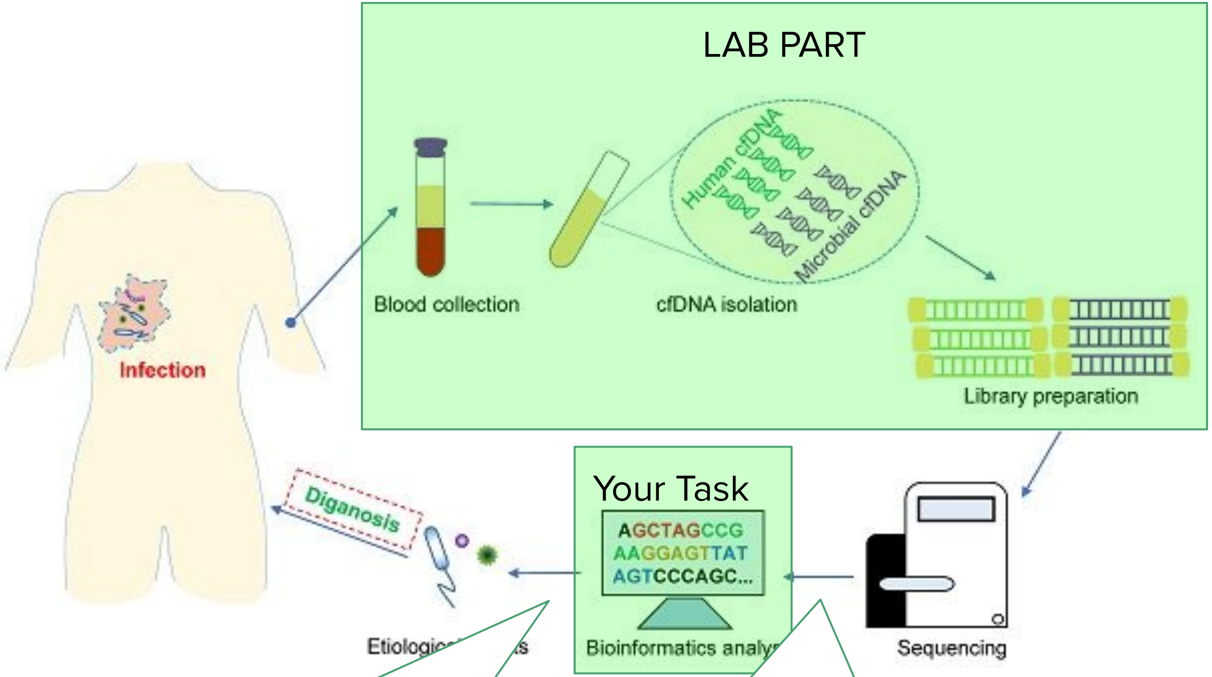
> “AMR occurs when bacteria, viruses, fungi and parasites change over time and no longer respond to medicines making infections harder to treat and increasing the risk of disease spread, severe illness and death.” WHO



Implementation Tasks

Antimicrobial Resistance of pathogens (AMR)

“AMR occurs when bacteria, viruses, fungi and parasites change over time and no longer respond to medicines making infections harder to treat and increasing the risk of disease spread, severe illness and death.” WHO



TASK: For each antibiotics, is this pathogen resistant to these antibiotics?

Genomic sequence: ATTCGGATTGC....

Source: https://www.thno.org/v10p5501.htm

Dataset description

- Git repository: https://github.com/hzi-bifo/seminar-dlmb-2025-summer-public
- 150 genomes
  - Training set (135 samples) and test set (15 samples)
- Labels: 0 (non-resistant), 1 (resistant)
- Primary implementation task:
  - Predict AMR for a pathogen (*Staphylococcus aureus*) against an antibiotic (Cefoxitin) given the genomic sequence of one of its genes (gene pbp4) as input
- Extended implementation task:
  - Pathogen: *Klebsiella pneumoniae*, antibiotics: Aztreonam, gene: all genes

Topics

Topics

- Feed-forward Neural Networks and back propagation [2 students]
- Convolutional Neural Networks (CNNs) [2 students]
- Recurrent Neural Networks and LSTMs [2 students]
- Transformers - Encoders [2 students]
- Transformers - Decoders, Encoder-Decoders [2 students]

Feed-forward Neural Networks and back propagation [2 students]

Goals: Getting familiar with the basics of neural networks

- Introduction to linear classification, multilayer perceptron (MLP), and back propagation algorithm
- Implementation in python using MLP

Suggested references

- Linear prediction: Lec. 1,2 at (https://bit.ly/1DIpc51) and (Chapter 4: https://stanford.io/2voWjra)
- G. Hinton’s lecture 2, 3: https://bit.ly/3TNBPqw
- Lecture from U of Waterloo: https://bit.ly/2A2mzgN
- Stanford Tutorial: https://stanford.io/1FRrkZw
- Practicals: In Keras (keras.io) or Pytorch (pytorch.org)

Convolutional Neural Networks (CNN) [2 students]

Goals: Getting familiar with the convolutional neural network

- CNN
- Implementation in python using only CNN

Suggested references

- MIT notes: https://tinyurl.com/3c8fk4mz
- G. Hinton's lecture: https://bit.ly/3TNBP qw
- Stanford Tutorial: https://stanford.io/1FRrkZw
- A more advanced reference: deeplearningbook.org
- Practicals: In Keras (keras.io) or Pytorch (pytorch.org), e.g. https://tinyurl.com/yfy56ay5

Recurrent Neural Networks [2 students]

Goals: Getting familiar with the RNNs and in particular LSTM

- Understanding “Vanilla” RNN
- Understanding the LSTM architecture (in particular read: https://bit.ly/1S6gmjZ)
- Implementation in python using only RNNs and LSTM

Suggested references

- Lecture from U of Waterloo: https://bit.ly/2RCNEhn
- Understanding LSTMs: https://bit.ly/1S6gmjZ
- MIT notes: https://tinyurl.com/2x4z77fz
- G. Hinton’s lecture: https://bit.ly/3TNBPqw
- A more advanced reference: deeplearningbook.org
- Practicals: In Keras (keras.io) or Pytorch (pytorch.org)

Transformers - Encoders only models [2 students]

Goals: Getting familiar with the concept of transformers architecture

- Encoder part of a transformer model (e.g., Bert)
- Implementation in python using encoders-based transformers models
    - **You are allowed to use a pretrained model**

Suggested references

- Representation learning: https://arxiv.org/pdf/1206.5538.pdf
- Transformer paper: https://arxiv.org/abs/1706.03762
- Simple explanation of transformers: https://jalammar.github.io/illustrated-transformer/
- Simple explanation of the details: https://serrano.academy/
- Practicals: In Keras (keras.io) or Pytorch (pytorch.org)
- Practicals: https://huggingface.co/docs/transformers/notebooks

Transformers - Decoder-only and Encoder - Decoders [2 students]

Goals: Getting familiar with the concept of these architectures

- Decoder-only models (GPT)
- Encoder-decoder models
- Their applications in bioinformatics

Suggested references

- Representation learning: https://arxiv.org/pdf/1206.5538.pdf
- Transformer paper: https://arxiv.org/abs/1706.03762
- Simple explanation of transformers: https://jalammar.github.io/illustrated-transformer/
- Simple explanation of the details: https://serrano.academy/
- Practicals: In Keras (keras.io) or Pytorch (pytorch.org)
- Practicals: https://huggingface.co/docs/transformers/notebooks

Further steps

Send an email (to both of us)

- From one member CC all other members
- Send three preferred topics in the order of preference
- Until **April 20th**
- If you have problem forming a team, send us an email!

Up to you how to split the tasks: collaborate!

Meeting and consultation with the lecturers (at least once)

Send report two weeks before the seminar day

Final seminar date:

- TBD 9:00 - 13:00 at BRICS (early July).

Any questions? Contact us:

- mohammad-hadi.foroughmand-araabi@helmholtz-hzi.de
- georgios.kallergis@helmholtz-hzi.de

The End

References

- Deep learning in Life Science Youtube series
- AMR image https://www.consilium.europa.eu/en/infographics/antimicrobial-resistance/