Project topics
IB3702 Mathematics for Machine Learning

Deadline: November 15, 2025

1 Introduction

This project exploration will allow you to apply the concepts and techniques that you have acquired throughout the course.

Our expectations

- Familiarity with the study material: We assume you have a solid grasp of the material covered in the course. Make sure to apply the mathematical concepts and techniques you've learned throughout the entire course, encompassing both calculus and linear algebra.
- Learning something new: Strive to expand your understanding in both calculus and linear algebra beyond what you learned during the lectures by exploring new concepts, techniques, or applications.
- Aim for understanding: It's crucial that you don't include more in the project than you can clearly explain to yourselves, your team member, and the teachers. Focus on achieving a deep understanding of the chosen topic.
- Collaborative decision-making: Both of you will be responsible for making choices and decisions together. This includes setting up a presentation appointment, selecting the topic, and defining the scope of the project.
- Collaborative report writing: Work together to create the project report, following the instructions provided in the Study guide and adhering to the assessment criteria[1].
- Timely submission and presentation: Submit the project report on time, make sure that you have a time slot for the presentation session at least two days ahead, and be prepared for the presentation of the project. You have **30 minutes for presentation** covering both Calculus and Linear Algebra. We have then 30 minutes for asking questions.
- Both of you should understand and be able to explain everything from the report.

We have confidence in your abilities and look forward to witnessing your progress and success in the project!

[1]See the rubric in Brightspace under *Cursusinformatie / Tentamen*.

Additional advice

- You do not need to try to cover everything. Find the boundaries of the project as soon as possible; this is called *scoping*. Within your scope, clarify all the concepts, ideas, and techniques.
- We lay the groundwork for each topic, offering a starting point to delve into the subject matter. Consider this as an introduction. If you encounter aspects that are not fully understood in the given resource, see it as an opportunity to learn something new.
- Feel free to use external resources, including books, reliable online texts, and educational videos from platforms like Khan Academy and 3Blue1Brown on YouTube.
- When studying new material, you should make numerical examples for yourselves. It is often useful to keep on extending one example when you study increasingly more complicated things.
- By using numerical examples while studying, you achieve two things simultaneously. First, this helps you truly grasp novel ideas. And second, you will need some of these examples in the project report.
- Do most computation with pen and paper for understanding. Computers can help make bigger examples, visualise, give intuition about changes, verify your manual results, and execute many related computations. Tools like the slider in GeoGebra or variables in Jupyter/Python can be particularly helpful in this regard.
- Include simple examples about all steps that show how you made sense of the mathematical techniques in this context.
- Limit the project **up to maximum 15 pages** excluding references (and probably auxiliary material).
- Before the presentation session, spend some time rehearsing the explanations with each other.

Remember, these tips will help you make the most of your project experience. Good luck, and we're excited to see your progress!

2 List of topics

The project aims to **deepen understanding and application of mathematical concepts** pivotal in machine learning. You will explore various mathematical frameworks and their direct relevance to real-world AI/ML tasks.

Important Notes:

- **The project is divided into two modules—Calculus and Linear Algebra**—each focusing on theoretical understanding and practical implementation within ML algorithms. From each module, you'll pick one topic, that is one from Calculus and another from Linear Algebra. Your focus will be on exploring the mathematical foundation rather than the ML model itself.

- Note that the topics of linear algebra will be revealed on October 9, 2025. **You must then submit the complete report by November 15, 2025** (in total 15 pages excluding references and probably auxiliary material).

- Allocate most of the space in your report to presenting **new mathematical insights**, rather than revisiting material previously covered in class.

2.1 Calculus in Machine Learning

Calculus serves as the mathematical backbone empowering machine learning (ML) through its multifaceted applications. At its core, calculus provides the tools necessary to understand the dynamics of change, pivotal in modeling and optimizing complex systems within ML algorithms.

Select one topic out of the following three topics for your own understanding of the calculus foundations supporting that specific ML models.

2.1.1 Topic 1: Derivative in Neural Networks (NN)

Derivatives in neural networks are fundamental for optimizing model parameters, facilitating learning, and enabling the network to make adjustments that gradually improve its performance in various tasks like classification, regression, and pattern recognition. Delve into understanding on how derivative supports the following foundations of NN models.

Gradient Descent and Weight Updates: In a neural network, the derivative of the loss function with respect to the network's weights tells us how the loss would change if we made a small change in the weights. This information guides the network's optimization process, specifically in gradient descent. The derivative helps identify the direction and magnitude in which to adjust the weights to minimize the loss.

Backpropagation: Backpropagation is a process that computes gradients efficiently using the chain rule of calculus. It involves calculating the derivatives of the loss function with respect to each parameter in the network, backward from the output layer to the input layer. These derivatives are used to update the weights in each layer, allowing the network to learn from its mistakes and improve its predictions.

Optimization Techniques: Advanced optimization techniques, such as momentum, Adam, RMSprop, etc., rely on derivatives to adjust learning rates and update parameters effectively. The derivatives provide information about the rate of change of the loss function concerning the parameters, crucial for optimizing the learning process and convergence speed.

2.1.2 Topic 2: Taylor Series Expansion and Approximation in ML Models

Taylor series expansions serve as a valuable mathematical tool in machine learning by enabling the approximation of complex functions with simpler polynomials. This approximation facilitates easier computations, aids in optimization techniques, and allows for the creation of more flexible models capable of handling nonlinear relationships within data.

Approximating Nonlinear Functions: In machine learning, many relationships between input and output data are nonlinear and complex. The Taylor series expansion helps in approximating these nonlinear functions by representing them as polynomial functions. This simplification enables easier manipulation and computation.

Local Approximation around a Point: The Taylor series expansion approximates a function by using its derivatives evaluated at a specific point. By calculating the function's value and its derivatives at that point, the expansion constructs a polynomial that closely mimics the behavior of the function locally around that point. You can also explore the convergence radius around a point.

2.1.3 Topic 3: Hessian Matrix and Second-Order Optimization in Machine Learning

The Hessian matrix and second-order optimization methods offer a more sophisticated approach to optimization in machine learning. They leverage curvature information to guide parameter updates, enabling faster convergence and improved handling of complex optimization landscapes, making them valuable tools for enhancing the efficiency and effectiveness of machine learning algorithms.

Curvature of the Loss Surface: The Hessian matrix in machine learning serves as a representation of the curvature of the loss surface within the parameter space of models. While gradients indicate the slope or direction of steepest ascent or descent, the Hessian matrix extends this understanding by capturing how the gradients change across different directions in the parameter space. It offers insights into the shape of the loss function's surface, identifying critical points such as minima, maxima, or saddle points during optimization processes.

Faster Convergence: Second-order optimization methods, like Newton's method or the Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm, often exhibit faster convergence rates compared to first-order methods like gradient descent. By leveraging information about both the gradient and curvature, these methods can take larger and more directed steps towards the optimal solution, leading to quicker convergence in many scenarios.

2.2 Linear Algebra in Machine Learning

Linear algebra acts as the structural framework bolstering machine learning (ML) with its diverse applications. At its essence, linear algebra equips us with fundamental tools to comprehend and manipulate multi-dimensional data, essential for modeling intricate relationships within ML systems. Matrices and vectors serve as the building blocks, enabling operations that capture data patterns, reduce dimensions, and solve optimization problems.

Select one topic out of the following three ones for your own understanding of the Linear Algebra foundations. Search for applications and choose one that uses the selected method as internal machinery to make a certain machine learning task. Unfold the modeling of the application to explore how the Linear Algebra technique is used and, if possible and applicable, generate a manageable small instance data/case such that you can implement and get results to present. Defining and explaining some basic topics is good, however it is not necessary to elaborate much the topics that we already covered during our lectures.

2.2.1 Topic 1: Eigenvalues and Eigenvectors in Principal Component Analysis (PCA)

Eigenvalues and eigenvectors in PCA serve as the foundation for dimensionality reduction in machine learning. They enable the extraction of essential information, reduction of noise, and efficient representation of high-dimensional data, facilitating improved data analysis and model efficiency. Delve into the underlying mathematics of PCA to uncover its role in delivering the following key properties

Basis of Dimensionality Reduction: Eigenvalues and eigenvectors form the core of PCA, a popular dimensionality reduction technique in ML. Eigenvectors represent directions in the original feature space, while eigenvalues denote their magnitudes, guiding PCA to identify the most informative axes or principal components.

Transformation for Feature Compression: PCA performs a linear transformation by projecting high-dimensional data onto a lower-dimensional subspace defined by eigenvectors. This transformation preserves the most important information while reducing the dataset's dimensionality, making it computationally more efficient without significant loss of information.

Visualization and Exploratory Analysis: PCA's transformation using eigenvectors allows for easy visualization of high-dimensional data in lower dimensions. It aids in exploratory analysis, enabling better understanding of data clusters, patterns, and relationships between variables.

2.2.2 Topic 2: Matrix Factorization in Machine Learning

Matrix factorization stands as a versatile and fundamental concept in machine learning, providing means to uncover latent structures, reduce data dimensions, enhance computational efficiency, and facilitate interpretability across diverse domains. Explore the foundational mathematics of matrix factorization to reveal its fundamental role in yielding the following essential properties.

Foundation for Data Decomposition: Matrix factorization techniques like Singular Value Decomposition (SVD) and Non-Negative Matrix Factorization (NMF) serve as pivotal tools in breaking down complex datasets into simpler components. They offer means to extract latent structures, reduce dimensionality, and unveil underlying patterns within the data.

Dimensionality Reduction and Feature Extraction: Matrix factorization methods play a key role in dimensionality reduction, allowing for the extraction of essential features from high-dimensional datasets. Techniques such as NMF aid in feature engineering by revealing meaningful representations, crucial in tasks like text mining, image processing, and recommender systems.

Computational Efficiency and Noise Reduction: Matrix factorization assists in noise reduction and computational efficiency by simplifying data representations. Through methods like SVD or NMF, it filters out noise, reduces redundancy, and represents data more compactly, facilitating faster computations and enhancing model performance.

2.2.3 Topic 3: Kernel Methods and the Kernel Trick

Kernel methods, supplemented by the kernel trick, stand as indispensable techniques in machine learning, facilitating the handling of non-linear relationships, expansion of feature spaces, and superior performance in various tasks. Their adaptability across diverse domains and critical role in enhancing algorithm capabilities highlight their significance in modern machine learning paradigms.

Dive into the mathematical framework underpinning Kernel methods to unveil their pivotal role in achieving the following crucial properties within machine learning.

Foundation for Non-linear Transformations: Kernel methods serve as powerful tools for handling non-linear relationships within data by implicitly mapping inputs into higher-dimensional spaces. The kernel trick, an integral part of these methods, enables efficient computation in these high-dimensional spaces without explicitly transforming the data.

Handling Non-linearity in Learning Algorithms: Kernel methods, through the kernel trick, allow linear algorithms to operate effectively in non-linear feature spaces. By employing diverse kernel functions like polynomial, radial basis function (RBF), or sigmoid kernels, they enable learning algorithms to capture intricate patterns and relationships in the data that linear methods alone might not capture.

Support in SVMs and Classification Tasks: Kernel methods play a pivotal role in Support Vector Machines (SVMs), facilitating effective separation of classes in non-linearly separable datasets. Leveraging different kernels, SVMs achieve superior classification performance by finding complex decision boundaries that optimize the margin between classes.