

Conceptual Bridges in Machine Learning

Exploring the Effect of Analogies on Multilayer Perceptron Understanding

Maria Cristescu[1]

Supervisors: Ilinca Rențea[1]**, Yuri Noviello**[1]**, Gosia Migut**[1]

[1]**EEMCS, Delft University of Technology, The Netherlands**

A Thesis Submitted to EEMCS Faculty Delft University of Technology,
In Partial Fulfilment of the Requirements
For the Bachelor of Computer Science and Engineering
June 22, 2025

Name of the student: Maria Cristescu
Final project course: CSE3000 Research Project
Thesis committee: Ilinca Rențea, Yuri Noviello, Gosia Migut, David Tax

An electronic version of this thesis is available at http://repository.tudelft.nl/.

Abstract

Machine Learning education faces significant challenges due to the abstract and mathematically-complex nature of fundamental models, such as Multilayer Perceptrons (MLPs). This paper investigates the effectiveness of conceptual metaphors and analogies as pedagogical tools to improve novice learner's understanding of key MLP concepts. Using large language models, we generated a set of analogies for core MLP topics. These analogies were then evaluated by experts to assess their quality, followed by a user study with novice learners employing a between-subject A/B test comparing analogy-based explanations to formal definitions. Although the study found no statistically significant improvement in knowledge gain or engagement that could be attributed to analogy-based explanations, trends suggest potential benefits in learner confidence and motivation. The research contributes a curated set of expert-evaluated analogies for ML education and discusses methodology limitations and directions for future work. This study highlights both the promise and complexity of integrating analogy-based teaching approaches into ML education.

1 Introduction

In the new era of AI, Machine Learning (ML) has become crucial, impacting a wide range of industries, from healthcare and finance to transportation and education [1]. As its impact on society grows, so does the need for ML literacy among specialists, students and the general public. To address this, many universities have incorporated ML courses in their curricula, in the hope of preparing students to be well-rounded thinkers and engineers, prepared for the professional world that awaits them [2].

However, ML is often considered a complex topic, with fundamental models like Multilayer Perceptrons (MLPs) rooted in mathematics and neurobiology [3], theoretical fields which can feel intimidating and inaccessible to individuals without prior exposure. Without accessible teaching education strategies, learners may struggle to grasp core concepts and misuse powerful technologies due to shallow understanding.

One promising approach involves the use of conceptual metaphors and analogies, tools recognized as effective tools for making abstract systems more understandable [4]. For example, in the field of Computer Science, Notional Machines have been used to guide students' mental models, helping them focus their attention on the right concepts [5].

Yet, while the value of such tools has been explored in traditional programming education [6], such contributions are notably sparse in the domain of ML. Pendyala [7] states, in a relatively recent publication (2022), that their work is the first to present "a number of real-world analogies to simplify and explain machine learning concepts and paradigms" [7]. Initial discussions, such as those by Shapiro and Fiebrink [8] emphasize the need for better ways to support student learning in ML, but do not answer the question of how analogies might serve this goal, particularly for models like MLPs.

Research Question

This paper aims to answer the following Research Question:

> To what extent do conceptual metaphors and analogies improve novice learners' ability to understand MLPs?

From this, the following sub-questions arise:

- **Sub-Question 1**: Based on expert survey responses, which analogies are considered most effective for explaining key MLP concepts?
- **Sub-Question 2**: How do analogy-based explanations impact student understanding of MLPs compared to traditional explanations?
- **Sub-Question 3**: How does prior Computer Science experience influence the effectiveness of analogy-based ML learning?

In pursuit of this goal, this paper makes three main contributions: (1) it generates and evaluates a set of conceptual metaphors for key MLP concepts, (2) it evaluates the quality of these analogies through an expert evaluation (addressing Sub-Questions 1), and (3) it analyzes their effect on students with no prior ML education (addressing Sub-Questions 2 and 3).

Paper Structure

The remainder of the paper's structure is as follows: Section 2 discusses related work in more detail, establishing the theoretical background for using analogies in ML education. Section 3 focuses on the analogy generation process, detailing both the methodology and the resulting set of analogies. Section 4 details the expert evaluation, presenting its methodology, results, and discussion. Section 5 outlines the user study, including the design, results, and corresponding discussion. Section 6 goes into detail about the ethical considerations of this research. Finally, Section 7 concludes the paper and suggests directions for future research.

2 Background

Teaching complex technical concepts to novice learners, particularly in fields like ML, presents well-known challenges. Learners often struggle to grasp abstract concepts, and educational research has shown that concrete examples can significantly help understanding [9]. To address this, computer science education has long relied on conceptual tools such as analogies, metaphors, and mental models. This section introduces the term of 'analogy' and reviews the educational literature surrounding Notional Machines (NMs) and analogy-based teaching approaches, highlighting their value in traditional CS education and their current under-use in the ML domain.

2.1 Defining Analogies

An *analogy* represents a comparison of different concepts, based on their similarities[1]. It maps features from an unknown, abstract, "target" domain, to features from another known, concrete, "source" domain. A *concept mapping* defines the specific parallels between elements of the target and source domains, identifying which elements from the source domain are used to explain the target concept. For example, gradient descent, a core algorithm in ML, can be explained using the analogy of a hiker going down a mountain[2]. In this analogy, gradient descent (target domain) is compared with a hiker trying to find the lowest point of a mountain (source domain). The concept mapping includes:

- the hiker → the algorithm
- the hiker's position → current value of model's parameters
- the size of the hiker's steps → the learning rate of the algorithm
- the global minimum → the lowest point in the valley

2.2 Notional Machines and Analogies in Computer Science

Teaching technical subjects to novices without the necessary level of tech knowledge can be difficult, especially when trying to choose the level of detail which would be most helpful for the learner [10]. To aid in this pedagogical process, Boulay, O'Shea, and Monk [10] propose the use of NMs, simplified, idealized descriptions of how a system operates, designed to help learners map unfamiliar computational processes to familiar concepts. They define a notional machine as

> "an idealized, conceptual computer whose properties are implied by the constructs in the programming language employed." [10].

Recent works have built on this principle, studying the use of conceptual methods in computing education. Munasinghe, Bell, and Robins [11], for example, argue about the usefulness of NMs and Computational Agents in a pedagogical context, being a critical tool in aligning the learners' mental models with the conceptual model of a program. This corresponds with Colburn and Shute's [6] view of the pedagogical role of analogies in Computer Science (CS), as they "provide a conceptual framework in which to situate constantly emerging new ontologies in computational environments" [6].

Among the most thorough studies on this topic is the work of Fincher, Jeuring, Miller, *et al.* [5], whose research results in a comprehensive list of NMs used in CS, also presented in an accessible format at https://notionalmachines.github.io. However, what is notable is the absence of any analogies related to ML concepts. In addition to that, Fincher, Jeuring, Miller, *et al.* [5] also note the lack of scientific works studying how effective NMs are in CS-education settings.

[1]https://www.merriam-webster.com/dictionary/analogy

[2]https://rushhabhh.medium.com/gradient-descent-heartbeat-of-machine-learning-099f9e9d5272

2.3 Gap in ML Education

This gap is further reinforced by recent literature in ML education. While Pendyala [7] introduces a set of intuitive, real-world analogies for explaining ML concepts, the work explicitly acknowledges its exploratory nature, lacking empirical evaluation. The need for more structured pedagogical approaches in ML is echoed by Shapiro and Fiebrink [8], who stress the importance of supporting students' conceptual understanding of ML systems, though without offering concrete tools to do so.

In a related effort, Pineda, Ashar, and Liu [12] investigated approaches to demystify ML concepts and increase ML literacy among K-12 students, such as group work, project-based learning and activity-based instructions. Similarly, Rodríguez-García, Moreno-León, Román-González, *et al.* [13] describe the design of *LearningML*, an engaging and easy-to-use platform that aims to introduce fundamental ML concepts, including the training, learning and evaluation phases of a model.

Together, these works highlight a promising, unexplored research opportunity: the adaptation of analogy-based education strategies in the teaching of complex, fundamental ML concepts, such as MLPs. This paper addresses this gap by developing and evaluating a set of conceptual metaphors aimed at improving students' understanding of MLPs.

3 Phase I: Analogy Generation

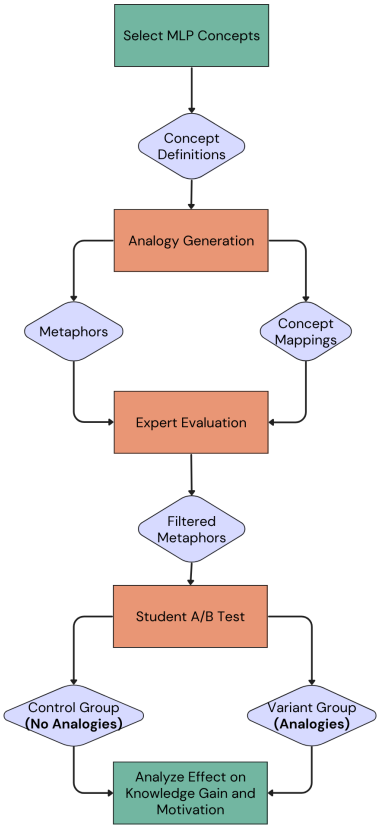


Figure 1: Methodology Flowchart

3.1 Methodology

The methodology of this study was structured in three distinct phases, as summarized in Figure 1. The first phase of the study focused on generating analogies for key MLP topics. As a starting point, we collected formal definitions from various sources for the following core MLP concepts:

1. Artificial Neurons
2. Activation Functions
3. Weights
4. Empirical Risk Minimisation
5. Biases
6. Forward Propagation
7. Backpropagation
8. Overfitting/Early Stopping
9. Learning Rate
10. Hidden Layers

We collected the definitions from websites and educational material of level similar to university-level ML courses. The criteria for selection were that they (1) reflected common teaching explanations and (2) clearly showed the individual components involved.

Designing analogies manually is often time-consuming and requires both pedagogical and domain expertise. Large Language Models (LLMs), such as GPT-3, have demonstrated a strong ability to generate high-quality, creative and contextually-relevant analogies [14]. Consequently, LLMs were the tool chosen for the analogy generation step of the experiment (specifically, GPT-4o).

For each concept, the LLM was prompted to generate analogies and their corresponding conceptual mappings. Prompts included the formal definition and a requirement that each individual concept referenced in the definition should be explicitly present in the analogy.

For example, given the formal explanation of hidden layers, the following analogy and concept mapping were generated:

- **Definition**: Between the input and output layers, there can be one or more layers of neurons. Each neuron in a hidden layer receives inputs from all neurons in the previous layer (either the input layer or another hidden layer) and produces an output that is passed to the next layer. The number of hidden layers and the number of neurons in each hidden layer are hyperparameters that need to be determined during the model design phase.
- **Analogy**: Imagine a factory assembly line that creates customized products. The production process involves multiple relay teams, each performing a specific task on the product before passing it along to the next team. The first team (input layer) receives raw materials and hands them to the next team, and this continues until the final team (output layer) delivers the finished product. The middle teams are the hidden layers—they do not interact directly with the customer or the raw materials, but their processing is essential for producing the final result.
- **Target Domain**: Hidden Layers
- **Source Domain**: Relay teams in a factory assembly line
- **Conceptual Mapping**: Presented in Table 1

MLP Concept	Analogy Element
Hidden layers	Middle relay teams in the assembly line.
Neurons in hidden layers	Workers in each relay team.
Receives inputs from previous layer neurons	Each worker gets processed parts from all workers in the prior team.
Produces outputs for next layer	Each worker refines the product and passes it on.
Number of hidden layers	Number of relay teams.
Number of neurons per hidden layer	Number of workers per relay team.
Hyperparameters in model design	Decisions about how many teams and workers are needed to optimize the assembly process.

Table 1: Concept Mapping Between Hidden Layers and Factory Analogy

Full transcripts of these interactions, including a list of all generated analogies, have been included in Appendix A, for transparency and reproducibility purposes.

4 Phase II: Expert Evaluation

To assess the quality of the generated analogies, we conducted an expert evaluation. This evaluation focused on three key criteria that capture both the quality and the structure of each analogy.

4.1 Methodology

Participant Selection

The study targeted Machine Learning experts. Their expertise was measured using a self-assessment question, included at the beginning of the survey: "What is your Machine Learning knowledge level? Select the one that fits your situation best." Only those identifying as having at least a bachelor-level knowledge of ML were considered expert participants. This criterion ensured that all evaluators were sufficiently familiar with ML concepts to provide informed feedback.

Evaluation Criteria

Each analogy was presented next to the formal concept definition and a corresponding concept mapping. Experts were then asked to assess each analogy on three criteria, inspired from an updated version[3] of the metrics suggested by Bhavya, Palaguachi, Zhou, *et al.* [15]:

- Target Concept Coverage - the degree to which each element in the definition is addressed by the metaphor
- Mapping Strength - the consistency of mappings between the source and target concepts

[3] Available at: https://sites.google.com/illinois.edu/analogyeval24/analogy-evaluation-criteria

- Metaphoricity - the extent of conceptual separation between the source and target domains

Survey Implementation

The expert evaluation was implemented via an anonymous, online survey. Each expert evaluated as many analogies as they were comfortable with, received in a random order, to maintain the quality of assessment.

4.2 Results

A total of 15 expert responses were collected for the evaluation. The participant pool consisted of one university lecturer, one master's student, three ML teaching assistants, and ten bachelor-level students.

To account for the subjective nature of analogy evaluation and the variable number of ratings per analogy, Krippendorff's alpha was used to measure inter-rater reliability. Krippendorff's alpha is a statistical tool used to assess the agreement between multiple evaluators [16]. This statistic allows for missing data and a varying number of raters per item.

Using the "K-Alpha Calculator" computational tool developed by Marzi, Balzano, and Marchiori [17], the alpha value obtained was **0.119**, which reflects a low overall inter-rater reliability across all individual evaluations. In other words, this single value captures the consistency of ratings across all expert assessments, treating each criterion - analogy pair as a distinct item.

The relationship between analogy ratings and (per-analogy) inter-rater reliability is shown in Figure 2. Based on the aggregated expert ratings, four analogies were selected for inclusion in the user study:

- The **learning rate** is like the stride of a hiker
- **Overfitting** is like memorizing practice exam answers
- **Hidden layers** are like relay teams in a factory assembly line
- The **bias** is like a baseline setting in a thermostat

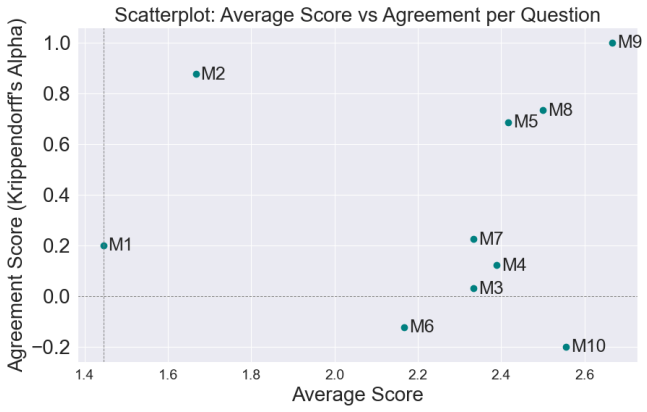


Figure 2: Relationship between analogy ratings and inter-rater agreement.

4.3 Discussion

What recommends Krippendorff's alpha for the analysis of the data is its ability to handle missing data and the varying number of raters per item. The observed Krippendorff's alpha value of 0.119 reflects the subjective nature of analogy interpretation, especially within complex technical domains like Machine Learning, and shows an opportunity for further studies on analogy evaluation techniques.

Since the inter-rater reliability value is especially dependent on the number of evaluations - the higher the number of raters, the more statistically accurate the alpha value [18] - the selection of analogies for the user study was based predominantly on the average score given by experts, while also taking into account the complexity and conceptual proximity of the topics. Given the moderate complexity of the selected concepts, they are believed to be suitable for the next step of the study, as they are unlikely to add any unnecessary cognitive load on the participants.

5 Phase III: User Study

To investigate the pedagogical effectiveness of conceptual metaphors, we conducted a controlled experiment, using a between-subject A/B test design. This methodology aligns with previous studies in ML education [19], which also used a between-subjects framework to evaluate ML education techniques.

The objective of this experiment was to evaluate whether the inclusion of analogy-based explanations improves student understanding of key MLP concepts: Learning Rate, Overfitting/Early Stopping, Hidden Layers and Biases. The following sections describe the participant selection, design and evaluation criteria for the experiment.

5.1 Methodology

Participant Selection

The target demographic consisted of students with no prior ML experience, to mirror the intended audience of an introductory ML course. A total of 23 participants were recruited through personal channels and chain-referral sampling. Inclusion criteria were self-reported novice ML knowledge.

To address Sub-Question 3, which examines how prior CS experience influences the effectiveness of analogy-based education, the participant pool was intentionally balanced between students with and without CS backgrounds.

Participation was voluntary and anonymized, and informed consent was obtained prior to the study. The experiment was conducted according to the ethical guidelines mentioned in Section 6.

Experiment Design

The experiment is designed as a between-subject A/B study, where participants were randomly assigned to one of two groups: Group A (the control group, N = 12; 8 with CS background, 4 without), which received only the formal definition of each concept, and Group B (the analogy group, N = 11; 7 with CS background, 4 without), which received both the formal definition and a conceptual metaphor.

Each ML concept was associated with one learning objective (LO), from which two multiple-choice questions were

generated: A pre-explanation question to assess baseline knowledge, and a post-explanation question designed to test comprehension of the concept after the provided explanation. Both questions include an "I don't know" option to discourage guessing. Additionally, participants rated their confidence in their post-question answers on a 5-point Likert scale [20], with options spanning from "Not at all confident" to "Very confident". The LOs were developed using the second level of Bloom's Taxonomy [21], to ensure the questions targeted concept comprehension. The entire list can be found in Appendix B.1.

After completing the instructional blocks, all participants filled out the Reduced Instructional Materials Motivation Survey (RIMMS) [22]. The RIMMS is a validated 12-item questionnaire measuring four motivational constructs from the ARCS model: Attention, Relevance, Confidence and Satisfaction. Each subscale is represented by three items, using a 5-point Likert response format [22]. This survey provides insight into learners' non-cognitive engagement with the instruction materials. Its validated structure and common use in measuring learner interaction with self-directed instructional materials made this survey an appropriate choice for this experiment.

Data Collection

The collected data included accuracy scores on pre- and post-questions, knowledge gain, confidence ratings and RIMMS scores across the four motivational dimensions.

5.2 Results

Knowledge gain

Due to non-normal distribution (all p-values $< .05$), Mann-Whitney U test was used to compare groups for all 4 concepts. As shown in Table 2, the result showed no statistically significant differences between the control (No Analogies) and experiment (Analogies) groups for any of the knowledge concepts.

Topic	Group	Mean	U-stat	p-value
Learning Rate	Control	0.167	74.0	.566
	Experiment	0.273		
Early Stopping	Control	0.000	81.0	.270
	Experiment	0.273		
Hidden Layer	Control	0.333	68.0	.918
	Experiment	0.364		
Bias	Control	0.667	40.0	.070
	Experiment	0.273		

Table 2: Knowledge gain by topic and condition (No Analogies vs Analogies). Mann-Whitney U test results included.

A two-way ANOVA test was performed to analyze the effect of analogies and CS background on knowledge gain. The results, as shown in Table 3, reveal that there is no statistically significant interaction between the two variables ($F(1, 10) = .117$, $p = .740$).

[3]Table structure adapted from https://www.statology.org/how-to-report-two-way-anova-results/

Source	Sum of Squares	df	F	p-value
Analogy Condition	0.286	1	0.156	.701
CS Background Condition	6.881	1	3.753	.081
Condition × Background	0.214	1	0.117	.740
Within	18.333	10	–	–

Table 3: Two-way ANOVA results comparing the effects of Analogies and CS background on knowledge gain[3].

Student Engagement and Motivation

Independent samples t-tests revealed no statistical significant differences between the analogy and no-analogy conditions on any of the ARCS motivation sub scales, nor the overall motivation, calculated as the average of these sub scales, as shown in Table 6.

Subscale	Group	Mean	t-stat	p-value
Attention	Control	3.583	0.295	.7713
	Experiment	3.697		
Relevance	Control	3.361	0.882	.3899
	Experiment	3.667		
Confidence	Control	3.028	1.297	.2087
	Experiment	3.606		
Satisfaction	Control	3.667	0.068	.9461
	Experiment	3.697		
Overall Motivation	Control	3.410	0.793	.4377
	Experiment	3.667		

Table 4: Engagement subscales and overall motivation by group (No Analogies vs Analogies) with t-test statistics.

5.3 Discussion

Knowledge gain

The results of the user study did not show any significant differences between instruction with and without analogies. Figure 3 illustrates the knowledge gain in participants with no prior ML experience, indicating that analogies had little to no effect, regardless of whether the participants had a background in Computer Science or not. The higher average knowledge gain in CS students might be explained by their familiarity with technical concepts and terms, such as overfitting or bias. In addition to that, CS students might have more experience in interpreting technical definitions, lowering the added effect of analogies.

Additionally, there seems to be no consistent trend in the effect of analogies on knowledge gain across concepts, as shown in Figure 4.



Figure 3: Knowledge Gain for People with no ML Experience by CS Experience and Experiment Condition.

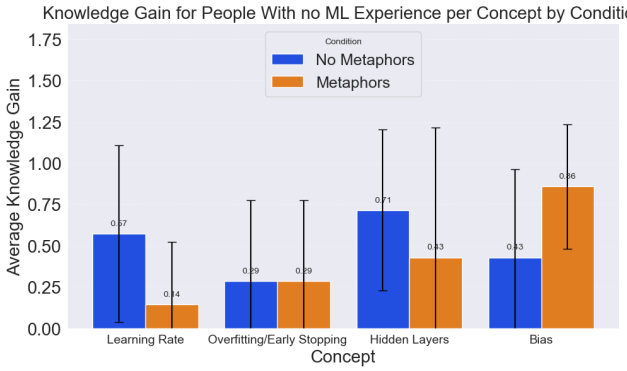


Figure 4: Knowledge Gain for People with no ML Experience by Concept and Experiment Condition.

The results of the study do not correspond with the existing literature. Similar experiments have shown significant improvements in learning when analogies are integrated in the material [23]. A possible explanation for this discrepancy could be the nature and complexity of the subject. ML concepts may require multiple exposures or more sustained practice for analogies to fully support learning. It is also possible that the setup of the instructional material, with only one question in each pre- and post-test, may not have captured the more nuanced or partial understanding of the participants.

Student Engagement and Motivation

Although the study results do not show statistically significant differences between the control and experiment groups, the Analogy group showed numerically higher scores in all categories, most notably in the Confidence scale, as shown in Figure 5. This suggests that analogies may help students feel more certain in their understanding, even if the effect is not strong enough to be statistically significant within this sample.

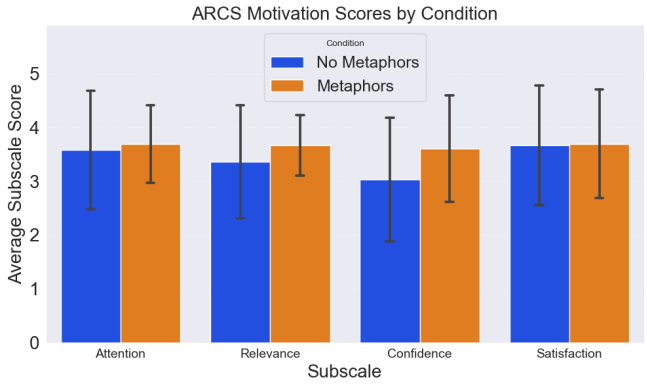


Figure 5: Engagement subscales by Experiment Condition.

Additionally, the overall motivation of participants who learned with the help of analogies was also numerically higher, which shows the potential value of using analogies not only as cognitive aids, but also as non-cognitive tools that can enable a more engaging and emotionally-supportive learning environment. In particular, the increase in confidence scores may indicate that analogies decrease the complexity of abstract concepts in the eyes of students, while also allowing them to approach the materials with increased self-confidence.

These trends, although not conclusive, align with previous literature on the motivational benefits of analogies in educational contexts. Prior research has highlighted their positive effect on student interest and confidence in their abilities and knowledge [24].

6 Responsible Research

6.1 Ethical Aspects

This research was designed with consideration for responsible research practices. The study involved both expert evaluations and user studies. All participants were informed of the nature of the study, and their consent was collected before participation. No personally identifiable information was collected and all responses were anonymized. Additionally, an application was submitted for Human Research Ethics Committee (HREC) approval, to ensure alignment with the academic integrity policies of Delft University of Technology.

Artificial Intelligence (AI) tools were used responsibly to assist in the research process. AI was used to transform data into tables suitable for the 2-column paper format. To ensure transparency, prompts are included in Appendix C. The Overleaf-integrated tool Writefull was also used to support language quality throughout the paper.

6.2 Reproducibility

To ensure transparency and reproducibility, the methodology and tools used in this study are documented in detail:

- Analogy Generation: All LLM-generated analogies were based on formal definitions and consistent prompts. All prompts and responses are included in Appendix A for full traceability.

- Expert Evaluation: The survey structure, evaluation criteria and participant selection process are clearly outlined in Section 4. Randomized question order and optional participation were used to minimize bias.
- User Study: The experimental design, participant selection criteria, instructional materials, and evaluation instruments are detailed in Section 5.

While some unpredictability can be expected due to the use of LLMs and human evaluators, the study's design allows others to replicate the process and conduct similar evaluations.

7 Conclusions and Future Work

This paper explored the use of conceptual metaphors and analogies as pedagogical tools for teaching fundamental MLP concepts. Through a multi-phased approach involving analogy generation using LLMs, expert evaluation and a controlled user study, it aimed to evaluate the educational value of these analogies in both cognitive (conceptual understanding) and non-cognitive (motivation and confidence) domains.

While the results did not show statistically significant effect in either domains, they highlight some important potential discussions regarding the use of analogies in ML education. Analogies may not yield immediate improvements in test performance, especially in one-session formats, but may still offer motivation and confidence benefits over time.

The study also contributes a set of expert-evaluated analogies for key MLP concepts, which can be found in an aggregated form on this website: https://ml-teaching-analogies.github.io/.

7.1 Limitations and Future Work

A possible limitation of this study is the low and inconsistent number of expert evaluations per analogy. Since Krippendorff's alpha depends on the number of evaluators, a larger number of responses could have led to a more accurate ranking of the analogies. Another limitation of this study is the linear nature of the evaluation process due to time constraints. A better approach could have been to repeat this experiment multiple times, which could have shown more stable evaluations and better expert consensus. Additionally, a different pool of experts, focusing on either ML lecturers and professors or industry practitioners, could have yielded results with higher inter-rater reliability. Future studies could look into more consistent strategies of analogy evaluation, which would aid further studies of analogy effects in ML education.

The study included only 23 participants, which limited its statistical significance. Additionally, due to time constraints, the instructional material was presented to the students during only one session, which may not reflect how analogies function in real educational environments, since students are often exposed multiple times to an explanation before having their knowledge tested. Furthermore, the scope of the study limited the instruction to four MLP-related topics, which may not fully represent the range of topics where analogies could have stronger pedagogical effects. Further works could investigate the effects of analogies on long-term retention in ML students.

While the RIMMS questionnaire is validated for motivational analysis, it may not fully apply to a questionnaire-based instruction material (i.e., it includes questions about the layout of the information on the page), affecting the final results of the study. Future work could address this by either adapting the RIMMS to better suit non-interactive, text-based formats or by developing a new instrument specifically designed for such contexts.

References

[1] A. Mishra, "A Comprehensive Review of Artificial Intelligence and Machine Learning : Concepts, Trends, and Applications," en, *International Journal of Scientific Research in Science and Technology*, vol. 11, no. 5, pp. 126–142, Sep. 2024, ISSN: 2395-602X, 2395-6011. DOI: 10.32628/IJSRST2411587. [Online]. Available: https://ijsrst.com/index.php/home/article/view/IJSRST2411587 (visited on 06/05/2025).

[2] J. Southworth, K. Migliaccio, J. Glover, *et al.*, "Developing a model for AI Across the curriculum: Transforming the higher education landscape via innovation in AI literacy," en, *Computers and Education: Artificial Intelligence*, vol. 4, 2023, ISSN: 2666920X. DOI: 10.1016/j.caeai.2023.100127. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S2666920X23000061 (visited on 06/05/2025).

[3] M. Paliwal and U. A. Kumar, "Neural networks and statistical techniques: A review of applications," en, *Expert Systems with Applications*, vol. 36, no. 1, pp. 2–17, Jan. 2009, ISSN: 09574174. DOI: 10.1016/j.eswa.2007.10.005. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0957417407004952 (visited on 06/04/2025).

[4] G. Lakoff and M. Johnson, *Metaphors We Live By*. The University of Chicago Press, 1980.

[5] S. Fincher, J. Jeuring, C. Miller, *et al.*, "Notional machines in computing education: The education of attention," Jun. 2020, pp. 21–50. DOI: 10.1145/3437800.3439202.

[6] T. Colburn and G. Shute, "Metaphor in computer science," *Journal of Applied Logic*, vol. 6, pp. 526–533, Dec. 2008. DOI: 10.1016/j.jal.2008.09.005.

[7] V. Pendyala, "Relating machine learning to the real-world: Analogies to enhance learning comprehension," in May 2022, pp. 127–139, ISBN: 978-3-031-05766-3. DOI: 10.1007/978-3-031-05767-0_11.

[8] R. Shapiro and R. Fiebrink, "Introduction to the special section: Launching an agenda for research on learning machine learning," *ACM Transactions on Computing Education*, vol. 19, pp. 1–6, Oct. 2019. DOI: 10.1145/3354136.

[9] A. Micallef and P. M. Newton, "The Use of Concrete Examples Enhances the Learning of Abstract Concepts: A Replication Study," en, *Teaching of Psychology*, vol. 51, no. 1, pp. 22–29, Jan. 2024, ISSN: 0098-6283, 1532-8023. DOI: 10.1177/00986283211058069. [Online]. Available: https://journals.sagepub.

com/doi/10.1177/00986283211058069 (visited on 06/06/2025).

[10] B. d. Boulay, T. O'Shea, and J. Monk, "The black box inside the glass box: Presenting computing concepts to novices," *International Journal of Man-Machine Studies*, vol. 14, no. 3, pp. 237–249, 1981. DOI: 10.1016/s0020-7373(81)80056-9.

[11] B. Munasinghe, T. Bell, and A. Robins, "Computational thinking and notional machines: The missing link," *ACM Trans. Comput. Educ.*, vol. 23, no. 4, Dec. 2023. DOI: 10.1145/3627829. [Online]. Available: https://doi-org.tudelft.idm.oclc.org/10.1145/3627829.

[12] C. L. Pineda, A. A. K. Ashar, and J. Liu, "Fostering AI Literacy: A Survey of Student Perceptions and Effective Practices in K-12 Machine Learning," en, in *2024 IEEE Frontiers in Education Conference (FIE)*, Washington, DC, USA: IEEE, Oct. 2024, pp. 1–7, ISBN: 979-8-3503-5150-7. DOI: 10.1109/FIE61694.2024.10893365. [Online]. Available: https://ieeexplore.ieee.org/document/10893365/ (visited on 06/05/2025).

[13] J. D. Rodríguez-García, J. Moreno-León, M. Román-González, and G. Robles, "Introducing Artificial Intelligence Fundamentals with LearningML: Artificial Intelligence made easy," en, in *Eighth International Conference on Technological Ecosystems for Enhancing Multiculturality*, Salamanca Spain: ACM, Oct. 2020, pp. 18–20, ISBN: 978-1-4503-8850-4. DOI: 10.1145/3434780.3436705. [Online]. Available: https://dl.acm.org/doi/10.1145/3434780.3436705 (visited on 06/06/2025).

[14] J. Kim, S. Suh, L. B. Chilton, and H. Xia, "Metaphorian: Leveraging Large Language Models to Support Extended Metaphor Creation for Science Writing," en, in *Proceedings of the 2023 ACM Designing Interactive Systems Conference*, Pittsburgh PA USA: ACM, Jul. 2023, pp. 115–135, ISBN: 978-1-4503-9893-0. DOI: 10.1145/3563657.3595996. [Online]. Available: https://dl.acm.org/doi/10.1145/3563657.3595996 (visited on 05/22/2025).

[15] B. Bhavya, C. Palaguachi, Y. Zhou, S. Bhat, and C. Zhai, "Long-Form Analogy Evaluation Challenge," en, pp. 1–16, Sep. 2024.

[16] K. Krippendorff, "Computing Krippendorff's Alpha-Reliability," en, Jan. 2011.

[17] G. Marzi, M. Balzano, and D. Marchiori, "K-Alpha Calculator–Krippendorff's Alpha Calculator: A user-friendly tool for computing Krippendorff's Alpha inter-rater reliability coefficient," en, *MethodsX*, vol. 12, p. 102 545, Jun. 2024, ISSN: 22150161. DOI: 10.1016/j.mex.2023.102545. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S2215016123005411 (visited on 05/29/2025).

[18] A. Zapf, S. Castell, L. Morawietz, and A. Karch, "Measuring inter-rater reliability for nominal data – which coefficients and confidence intervals are appropriate?" en, *BMC Medical Research Methodology*, vol. 16, no. 1, p. 93, Dec. 2016, ISSN: 1471-2288. DOI: 10.1186/s12874-016-0200-9. [Online]. Available: http://bmcmedresmethodol.biomedcentral.com/articles/10.1186/s12874-016-0200-9 (visited on 05/29/2025).

[19] H. Gao, Y. Xie, and E. Kasneci, "PerVRML: ChatGPT-Driven Personalized VR Environments for Machine Learning Education," en, *International Journal of Human–Computer Interaction*, pp. 1–15, May 2025, ISSN: 1044-7318, 1532-7590. DOI: 10.1080/10447318.2025.2504188. [Online]. Available: https://www.tandfonline.com/doi/full/10.1080/10447318.2025.2504188 (visited on 06/06/2025).

[20] A. Joshi, S. Kale, S. Chandel, and D. Pal, "Likert Scale: Explored and Explained," en, *British Journal of Applied Science & Technology*, vol. 7, no. 4, pp. 396–403, Jan. 2015, ISSN: 22310843. DOI: 10.9734/BJAST/2015/14975. [Online]. Available: https://journalcjast.com/index.php/CJAST/article/view/381 (visited on 06/04/2025).

[21] B. Bloom, M. Englehart, E. Furst, W. Hill, and D. Krathwohl, *Taxonomy of Educational Objectives, Handbook 1: Cognitive Domain*. 1956.

[22] N. Loorbach, O. Peters, J. Karreman, and M. Steehouder, "Validation of the Instructional Materials Motivation Survey (IMMS) in a self-directed instructional setting aimed at working with technology," en, *British Journal of Educational Technology*, vol. 46, no. 1, pp. 204–218, Jan. 2015, ISSN: 0007-1013, 1467-8535. DOI: 10.1111/bjet.12138. [Online]. Available: https://bera-journals.onlinelibrary.wiley.com/doi/10.1111/bjet.12138 (visited on 05/29/2025).

[23] G.-J. Hwang, S.-Y. Chien, T.-W. Chen, and C.-C. Chang, "From analogy to reflective thinking: An observation-identification-analogy-based virtual learning approach," en, *Educational technology research and development*, Mar. 2025, ISSN: 1042-1629, 1556-6501. DOI: 10.1007/s11423-025-10479-w. [Online]. Available: https://link.springer.com/10.1007/s11423-025-10479-w (visited on 06/20/2025).

[24] F. Petri-Ouani and S. N. Löffler, "Effects of different learning materials on student motivation and learning results; [wie wirken sich unterschiedlich lehr- / lernmaterialien auf die studentische lernmotivation und das lernergebnis aus?]" *Zeitschrift fur Padagogische Psychologie*, vol. 39, no. 3, pp. 183–196, 2025. DOI: 10.1024/1010-0652/a000361. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85156214964&doi=10.1024%2f1010-0652%2fa000361&partnerID=40&md5=08c3394645135722c5ee0eb764dc9e02.

A LLM Generated Analogies

A.1 Artificial Neurons

User

For the following concept and its definition, provide a metaphor/analogy and its concept mapping. The metaphor and its mapping need to refer to all elements present in the definition.

Artificial Neurons

In a multilayer perceptron, neurons process information in a step-by-step manner, performing computations that involve weighted sums and nonlinear transformations.

ChatGPT

Metaphor/Analogy:

Artificial Neurons are like chefs in a multi-course kitchen.

Each chef receives ingredients (inputs), adjusts their proportions using a recipe (weights), combines them into a dish (weighted sum), and adds a unique twist with seasoning or technique (nonlinear transformation). The dish is then passed to the next chef in the line (next layer) for the next course.

Concept Mapping

Table 5: Concept Mapping Between Artificial Neurons and Kitchen Metaphor

Artificial Neuron Concept	Metaphor Element
Neurons in a multilayer perceptron	Chefs in a multi-course kitchen
Inputs	Ingredients received by each chef
Weights	Recipe proportions guiding ingredient use
Weighted sum	Combined dish prepared using adjusted ingredients
Nonlinear transformation	Chef's unique twist: seasoning or special technique
Step-by-step processing across layers	Course-by-course preparation passed between chefs

Would you like a visual diagram for this metaphor?

A.2 Activation Function

User

For the following concept and its definition, provide a metaphor/analogy and its concept mapping. The metaphor and its mapping need to refer to all elements present in the definition.

Activation Functions

An activation function is a mathematical function applied to the output of a neuron. It introduces non-linearity into the model, allowing the network to learn and represent complex patterns in the data. Without this non-linearity feature, a neural network would behave like a linear regression model, no matter how many layers it has. Activation function decides whether a neuron should be activated by calculating the weighted sum of inputs and adding a bias term. This helps the model make complex decisions and predictions by introducing non-linearities to the output of each neuron.

ChatGPT

Metaphor/Analogy:

Activation Functions are like decision-making gates in a maze of mirrors.

Concept Mapping

Neural Network Concept	Analogy Element
Neuron	A mirror in the maze that reflects light (information)
Input (weighted sum + bias)	The intensity and angle of incoming light hitting the mirror
Activation Function	A gate that decides whether to let the light continue through the maze
Non-linearity	The twisting, branching paths of the maze, which change the direction of light
Linear network (no activation)	A straight hallway of mirrors with no turns, reflecting light in a direct line
Learning complex patterns	Navigating a maze with curves, forks, and dead ends, requiring careful redirection
Decision to activate	Gate opens or closes based on how the light hits (intensity/angle calculation)

Summary:

Just as gates in a mirror maze decide if and how light continues based on its characteristics, activation functions decide whether a neuron should pass its signal forward, enabling the network to navigate complex, nonlinear paths to reach its goal—accurate predictions.

Would you like a visual diagram of this metaphor?

User

Try to map it to a more common thing. Decision-making gates in a maze of mirrors is too abstract.

ChatGPT

Great point — let's ground it in something more familiar.

Metaphor/Analogy:

An activation function is like a bouncer at a club entrance.

Concept Mapping

Summary:

Just like a bouncer decides who gets into a club based on multiple cues, the activation function decides whether a neuron's output is important enough to pass forward. This helps the neural network learn more subtle and complex patterns, instead of just blindly processing everything.

Would you like a few metaphor variations for different audiences (e.g., kids, businesspeople, gamers)?

A.3 Weights

User

Neural Network Concept	Analogy Element
Neuron	A club/restaurant that may let people (signals) in
Input (weighted sum + bias)	A guest's profile — their ID, dress, attitude (summed info used for evaluation)
Activation Function	The bouncer deciding who gets in and who doesn't
Non-linearity	The bouncer's nuanced judgment, not just a fixed rule (e.g., dress code vs. vibes)
Linear network (no activation)	A club that lets everyone in or uses a rigid rule, leading to predictable outcomes
Learning complex patterns	The bouncer adapting rules over time to admit a crowd that matches the vibe
Decision to activate	The bouncer gives a nod (signal continues) or denies entry (signal blocked)

For the following concept related to Multilayer Perceptrons and its definition, provide a metaphor/analogy and its concept mapping. The metaphor and its mapping need to refer to all elements present in the definition.

Weights

Weights are numerical values associated with the connections between neurons. They determine the strength of these connections and, in turn, the influence that one neuron's output has on another neuron's input.

ChatGPT

Metaphor: Volume Knobs on a Mixing Console

Imagine a sound mixing console used by a music producer. Each instrument (like guitar, drums, vocals) has its own volume knob. These knobs control how loud or soft each instrument sounds in the final mix. Similarly, weights in a neural network control how much influence each neuron's output has on the next neuron's input.

Concept Mapping of the Metaphor to the Definition

MLP Concept	Mixing Console Metaphor
Neurons	Instruments (e.g., guitar, drums, vocals)
Connections between neurons	Audio channels between instruments and output speaker
Weights	Volume knobs
Numerical values of weights	Volume level (0 = mute, 10 = loud)
Influence of one neuron's output	Loudness of an instrument in the final mix

Would you like a visual of this metaphor as a diagram?

A.4 Empirical Risk Minimisation

User

For the following concept and its definition, provide a metaphor/analogy and its concept mapping. The metaphor and its mapping need to refer to all elements present in the definition.

Empirical Risk Minimisation

Empirical Risk Minimization is a critical concept in the theory of statistical learning and ML, highlighting the process of optimizing model parameters by minimizing the empirical risk or the average loss on the training dataset.

ChatGPT

Metaphor/Analogy:

Empirical Risk Minimization is like a chef perfecting a recipe by tasting and adjusting it based only on a sample of guests (training data) rather than the entire population of diners (true data distribution).

Concept Mapping

Original Concept	Metaphorical Element
Empirical Risk Minimization	Chef perfecting a recipe
Optimizing model parameters	Adjusting ingredients or cooking technique
Empirical risk / average loss	Guests' feedback on taste
Training dataset	Sample of guests
True (unknown) data distribution	Entire population of potential diners

This analogy emphasizes that the chef (like the learning algorithm) can only base improvements on a limited sample, aiming to generalize the adjustments as best as possible to suit everyone's palate—even those not present at the tasting.

Would you like another analogy with a different domain (e.g., sports, construction, or music)?

A.5 Biases

User

For the following concept related to Multilayer Perceptrons and its definition, provide a metaphor/analogy and its concept mapping. The metaphor and its mapping need to refer to all elements present in the definition.

Biases

Biases are essentially constants associated with each neuron. Unlike weights, biases are not connected to specific inputs but are added to the neuron's output. Biases serve as a form of offset or threshold, allowing neurons to activate even when the weighted sum of their inputs is not sufficient on its own.

ChatGPT

Metaphor: Thermostat with an Adjustable Baseline

Imagine a thermostat-controlled heater in a house. The temperature sensor (like the inputs and weights) reads the room's temperature and decides whether to turn on the heater. However, there's also a baseline setting — a knob that allows you to set a desired minimum temperature. This knob doesn't depend on the sensor readings; it's always there, shifting the behavior of the system. That knob is like the bias in a neuron.

Concept Mapping

Would you like a visual diagram of this metaphor and its mapping?

MLP Concept	Thermostat Analogy
Neuron	Thermostat
Inputs	Current room temperature
Weights	Sensitivity of thermostat to temperature
Bias	Thermostat baseline setting (minimum desired temperature)
Weighted Sum	Evaluation of temperature difference from target
Output Activation	Heater turning on or staying off
Adaptability	Changing the baseline setting

A.6 Forward Propagation

User

For the following concept related to Multilayer Perceptrons and its definition, provide a metaphor/analogy and its concept mapping. The metaphor and its mapping need to refer to all elements present in the definition.

Forward Propagation

Forward propagation (or forward pass) refers to the calculation and storage of intermediate variables (including outputs) for a neural network in order from the input layer to the output layer.

ChatGPT

Sure! Here's a metaphor and concept mapping for Forward Propagation in the context of Multilayer Perceptrons (MLPs):

Metaphor: The Assembly Line in a Factory

Imagine a factory with an assembly line where raw materials enter at one end, and a finished product comes out the other. At each station along the line, workers (machines) perform a specific task using the partially assembled product from the previous station, and then pass it along to the next. Each station also logs its work for quality control.

Concept Mapping

MLP Concept	Factory Metaphor Element
Input Layer	Row materials entering the factory
Hidden Layers	Intermediate workstations along the assembly line
Output Layer	Final product at the end of the assembly line
Neurons in each layer	Workers/machines at each station
Forward Propagation	Flow of product through the assembly line
Intermediate variables (outputs)	Logs or records of work done at each station
Order from input to output	Sequential processing from the first to last station

Would you like a visual diagram to go with this analogy?

A.7 Backpropagation

User

For the following concept related to Multilayer Perceptrons and its definition, provide a metaphor/analogy and its concept mapping. The metaphor and its mapping need to refer to all elements present in the definition.

Backpropagation

Backpropagation refers to the method of calculating the gradient of neural network parameters. In short, the method traverses the network in reverse order, from the output to the input layer, according to the chain rule from calculus. The algorithm stores any intermediate variables (partial derivatives) required while calculating the gradient with respect to some parameters.

ChatGPT

Metaphor: Baking a Cake and Discovering What Went Wrong

Imagine you've baked a cake and it didn't turn out right—it's too dry. To figure out what went wrong, you work backwards from the final cake to each ingredient and step in the recipe. You consider: Was the oven too hot? Did you use too little milk? You examine each stage to see how much it contributed to the dryness. Along the way, you write down the effect of each step to understand how each one impacted the outcome.

Concept Mapping

Backpropagation Concept	Cake Baking Analogy
Gradient of neural network parameters	Understanding how much each ingredient/step contributed to the dry cake
Traversing network in reverse order (output to input)	Starting from the final cake and retracing steps backward
Chain rule from calculus	Step-by-step influence tracing—how each decision led to the final result
Intermediate variables (partial derivatives)	Notes on each step's impact (e.g., "less milk = drier texture")
Gradient w.r.t. parameters	How changing ingredients (parameters) would change the cake outcome

Would you like similar metaphors for other MLP concepts like activation functions or gradient descent?

A.8 Overfitting/Early stopping

User

For the following concept related to Multilayer Perceptrons and its definition, provide a metaphor/analogy and its concept mapping. The metaphor and its mapping need to refer to all elements present in the definition.

Overfitting/Early Stopping

Overfitting [...] occurs when a model learns to memorize the training data rather than generalize patterns from it, leading to poor performance on unseen data. Early stopping is a form of regularization that halts the training process when the performance of the model on a validation dataset starts to degrade. Instead of training the model until convergence, early stopping monitors the validation error during training

and stops the training process when the validation error begins to increase.

ChatGPT

Metaphor: "Cramming for an Exam vs. Learning for Mastery"

Imagine a student preparing for an exam. If the student crams by memorizing exact answers from practice exams without understanding the material, they might do well on those practice tests but struggle with new questions on the real exam. This is like overfitting.

A wise teacher watches the student's progress and tells them to stop studying when they begin to show signs of mental fatigue and declining performance on mock exams — preventing burnout and poor generalization. This is like early stopping.

Concept Mapping

ML Concept	Metaphor
Training data	Practice exam questions
Model memorizing	Student cramming exact answers
Overfitting	Student can't answer new/different questions
Unseen data	Real exam with new questions
Validation dataset	Mock exams during study
Validation error	Mistakes made on mock exams
Early stopping	Teacher stopping the student before burnout
Convergence	Studying until exhaustion

Would you like a diagram to visually represent this metaphor?

A.9 Learning Rate

User

For the following concept related to Multilayer Perceptrons and its definition, provide a metaphor/analogy and its concept mapping. The metaphor and its mapping need to refer to all elements present in the definition.

Learning Rate

The learning rate is arguably the most important hyperparameter in training neural networks. It determines the size of the steps the optimization algorithm takes when adjusting the weights of the model in response to the gradients.

ChatGPT

Metaphor/Analogy: "The learning rate is like the stride of a hiker climbing a mountain with a map (the gradient) to reach the summit (optimal model weights)."

Concept Mapping

The metaphor highlights that if the stride is too large, the hiker might overshoot or trip, and if it's too small, progress is slow and may never reach the summit efficiently—just as in training with too high or low a learning rate.

Would you like metaphors for more MLP concepts?

MLP Concept	Metaphor/Analogy Element
Learning rate	The stride length of the hiker
Optimization algorithm	The hiker
Adjusting weights	Climbing or stepping toward the summit
Gradients	The map or compass that points the hiker in the right direction
Model	The hiker's route
Optimal model weights	The summit (goal of the hike)
Training	The hiking journey

A.10 Hidden Layers

User

For the following concept related to Multilayer Perceptrons and its definition, provide a metaphor/analogy and its concept mapping. The metaphor and its mapping need to refer to all elements present in the definition.

Hidden Layers

Between the input and output layers, there can be one or more layers of neurons. Each neuron in a hidden layer receives inputs from all neurons in the previous layer (either the input layer or another hidden layer) and produces an output that is passed to the next layer. The number of hidden layers and the number of neurons in each hidden layer are hyperparameters that need to be determined during the model design phase.

ChatGPT

Metaphor: "Relay Teams in a Factory Assembly Line"

Imagine a factory assembly line that creates customized products. The production process involves multiple relay teams, each performing a specific task on the product before passing it along to the next team. The first team (input layer) receives raw materials and hands them to the next team, and this continues until the final team (output layer) delivers the finished product. The middle teams are the hidden layers—they do not interact directly with the customer or the raw materials, but their processing is essential for producing the final result.

Concept Mapping

MLP Concept	Metaphor Element
Hidden layers	Middle relay teams in the assembly line
Neurons in hidden layers	Workers in each relay team
Receives inputs from previous layer neurons	Each worker gets processed parts from all workers in the prior team
Produces outputs for next layer	Each worker refines the product and passes it on
Number of hidden layers	Number of relay teams
Number of neurons per hidden layer	Number of workers per relay team
Hyperparameters in model design	Decisions about how many teams and workers are needed to optimize the assembly process

Would you like similar metaphors for other MLP components like activation functions or weights?

B User Study

B.1 Learning Objectives

- Explain the function of the learning rate in the training process of a neural network.
- Explain the concept of overfitting and demonstrate how early stopping serves as a regularization technique to improve model generalization.
- Interpret the role of hidden layers in transforming input data to enable the learning of complex patterns within a neural network.
- Show the function of the bias term in a neuron, specifically how it influences activation by shifting the decision boundary.

B.2 Survey Questions

Learning Rates

Pre-Question: Which of the following best describes the role of the learning rate in training a neural network?

- It decides how many layers the network has
- It determines the size of each step the model takes to adjust itself
- It tells the network when to stop training
- It sets the initial values of the model's weights
- I don't know

Explanation: The learning rate is arguably the most important hyperparameter in training neural networks. It determines the size of the steps the optimization algorithm takes when adjusting the weights of the model in response to the gradients.

Analogy: The learning rate is like the stride of a hiker climbing a mountain with a map (the gradient) to reach the summit (optimal model weights).

In this situation:

- The optimization algorithm would be the hiker
- Adjusting the weights would be climbing or stepping toward the summit
- The gradients would be the map or compass that points the hiker in the right direction
- The training process would be the hiking journey

Post-Question: What does the learning rate control during neural network training?

- The number of steps taken during training
- How much the model's weights are adjusted in response to the gradients
- The technique used to calculate the gradients
- The method used to collect training data
- I don't know

Overfitting/Early Stopping

Pre-Question: Which statement best captures a key risk associated with a model that overfits its training data?

- The model performs well on unseen data but poorly on training data
- The model captures noise and irrelevant details, reducing its ability to generalize
- The model ignores the training data and relies solely on validation data
- The model requires less computational power due to simplified learning
- I don't know

Explanation: Overfitting occurs when a model learns to memorize the training data rather than generalize patterns from it, leading to poor performance on unseen data. Early stopping is a form of regularization that halts the training process when the performance of the model on a validation dataset starts to degrade. Instead of training the model until convergence, early stopping monitors the validation error during training and stops the training process when the validation error begins to increase.

Analogy: Imagine a student preparing for an exam. If the student crams by memorizing exact answers from practice exams without understanding the material, they might do well on those practice tests but struggle with new questions on the real exam. This is like overfitting. A wise teacher watches the student's progress and tells them to stop studying when they begin to show signs of mental fatigue and declining performance on mock exams — preventing burnout and poor generalization. This is like early stopping.

In this example:

- The training data would be the practice exam questions
- The training error would be student's mistakes on the original practice exam questions
- The test (unseen) data would be the real exam with new questions
- The validation dataset would be the different sets of practice questions used to check progress during studying
- The validation error would be the mistakes made on the different set of practice exams

Post-Question: During training, early stopping is used because continuing training past a certain point can cause which of the following?

- The training error will increase, while validation error decreases
- Both training error and validation error will increase
- Validation error will begin to increase, while training error decreases
- Validation error will remain constant but training error will fluctuate randomly
- I don't know

Hidden Layers

Pre-Question: In a neural network, what do hidden layers do with the information they receive from previous layers?

- They memorize the input values for later comparison
- They pass the inputs directly to the next layer
- They transform the inputs and then pass them to the next layer
- They ensure that the output layer has access to the original input values
- I don't know

Explanation: Between the input and output layers, there can be one or more layers of neurons. Each neuron in a hidden layer receives inputs from all neurons in the previous layer (either the input layer or another hidden layer) and produces an output that is passed to the next layer. The number of hidden layers and the number of neurons in each hidden layer are parameters that need to be determined during the model design phase.

Analogy: Imagine a factory assembly line that creates customized products. The production process involves multiple relay teams, each performing a specific task on the product before passing it along to the next team. The first team (input layer) receives raw materials and hands them to the next team, and this continues until the final team (output layer) delivers the finished product. The middle teams are the hidden layers - they do not interact directly with the customer or the raw materials, but their processing is essential for producing the final result.

In this situation:

- The neurons in hidden layers would be the workers in each relay team
- The received inputs from previous layer neurons would be each worker getting processed parts from all workers in the prior team
- The produced outputs for next layer would be each worker refining the product and passing it on

Post-Question: Which of the following best describes the function of hidden layers in a neural network?

- They store the training data for fast retrieval during testing
- They apply transformations to the inputs, enabling the network to learn complex patterns
- They control the learning rate and adjust the weights between input and output
- They evaluate model performance using a separate validation set
- I don't know

Bias

Pre-Question: In a neural network, what is the role of a bias in a neuron's output calculation?

- To multiply the input by a constant factor before passing it to the next layer
- To ignore low-valued inputs that don't meet a threshold
- To add a constant that shifts the neuron's activation threshold
- To reduce the total number of weights that need to be trained
- I don't know

Explanation: Biases are essentially constants associated with each neuron. Biases are not connected to specific inputs, but are added to the neuron's output. Biases serve as a form of offset or threshold, allowing neurons to activate even when the weighted sum of their inputs is not sufficient on its own.

Analogy: Imagine a thermostat-controlled heater in a house. The temperature sensor (like the inputs and weights) reads the room's temperature and decides whether to turn on the heater. However, there's also a baseline setting - a knob that allows you to set a desired minimum temperature. This knob doesn't depend on the sensor readings; it's always there, shifting the behavior of the system. That knob is like the bias in a neuron.

In this example:

- The neuron would be the thermostat
- The inputs would be the current room temperature
- The weights would be the sensitivity of thermostat to temperature
- The bias would be the thermostat baseline setting (minimum desired temperature)
- The Output Activation would be the heater turning on or staying off

Post-Question: Why is a bias important for a neuron?

- It ensures that only the strongest inputs can activate the neuron.
- It allows the neuron to deactivate automatically when inputs are too strong.
- It provides flexibility by shifting the conditions under which the neuron activates
- It balances all input weights to keep output stable
- I don't know

C AI Use

Prompt 1:

Subscale No Metaphors Mean No Metaphors SD Metaphors Mean 0 Attention 3.5833 1.1021 3.6970 1 Relevance 3.3611 1.0489 3.6667 2 Confidence 3.0278 1.1500 3.6061 3 Satisfaction 3.6667 1.1192 3.6970

Metaphors SD t-statistic p-value 0 0.7220 -0.2948 0.7713 1 0.5578 -0.8822 0.3899 2 0.9868 -1.2972 0.2087 3 1.0050 -0.0684 0.9461

I need this data formatted into a latex table suitable for a 2-column paper. it should have 4 rows, one for each subscale. each row should then have 2 sub rows, showing the 2 groups for each (Control - no analogy and experiment - analogies), the means, then the 1 t-stat and p-value for each of the 4 rows.

Prompt 2:

Subscale	Group	Mean	t-stat	p-value
Attention	Control Experiment	3.583 3.697	0.295	.7713
Relevance	Control Experiment	3.361 3.667	0.882	.3899
Confidence	Control Experiment	3.028 3.606	1.297	.2087
Satisfaction	Control Experiment	3.667 3.697	0.068	.9461

Table 6: Engagement subscales and overall motivation by group (No Analogies vs Analogies) with t-test statistics.

following this latex table, format another table with this data:

Source SS DF MS F p-unc 0 Dataset 0.285714 1.0 0.285714 0.155844 0.701301 1 CSsimplified 6.880952 1.0 6.880952 3.753247 0.081440 2 Dataset * CSsimplified 0.214286 1.0 0.214286 0.116883 0.739516 3 Residual 18.333333 10.0 1.833333 NaN NaN

np2 0 0.015345 1 0.272899 2 0.011553 3 NaN