Enhancing LSTM-Based Sarcasm Detection on Social Media with LLM-Generated Sentences.

Gabriel Francisco dos Santos Silva
gabfssilva@gmail.com / gabfssilva@usp.br
PPgSI / EACH / USP

June 10, 2024

Abstract

Identifying sarcasm can be challenging, even for humans, due to the nuances of sarcastic sentences that depend on factors like culture, context, current and past events, and the language itself. The inability to detect sarcasm, especially on social networks, can lead to misinterpretations, difficult moderation, cyberbullying, hate speech, and other issues. These challenges underscore the complexity of interpreting sarcasm in text. Although Long Short-Term Memory Networks (LSTMs) are recognized for their effectiveness in handling the complexities of human language, which is inherently sequential and context-sensitive, the lack of sufficiently labeled datasets of sarcastic sentences still limits their potential. This project proposes using Large Language Models (LLMs), specifically GPT-4o, to generate synthetic sentences as additional training data, thereby enhancing the learning capabilities of LSTMs. By incorporating these synthetic sentences generated from the LLM, the model is expected to perform better than when trained solely with existing data. This approach allows for exposure to a wider variety of sarcasm, promoting a more robust understanding. The effectiveness of this enhanced model will be measured via F1-Score, Accuracy, Precision and Recall using the iSarcasmEval dataset, which is a collection of intended sarcastic sentences from Twitter.

Keywords sarcasm detection · recurrent neural networks · rnn · lstm · llm · transformers · social network · social media

1. Introduction

Even with the best current techniques, sarcasm is known to be difficult to detect. The most common approach involves using neural networks to identify contrasts in sentences that contain positive tones in negative situations and vice versa (Riloff et al. 2013). The challenge posed by sarcastic sentences, or any type of irony for that matter, stems from their rich and context-dependent nature. Although both sarcasm and irony can be used in humorous contexts, they are also frequently employed in negative tones (Johnson and Kreuz 2023), which can lead to misunderstandings.

A core hypothesis of this proposal is that, the contrast in tone and context — where positive expressions are used in negative situations or vice versa — is a definitive marker of sarcasm.

However, the effectiveness of current detection models is limited by the lack of diverse and adequately labeled training data. To address this gap, we propose enhancing training datasets with synthetic data generated by Large Language Models (LLMs), specifically GPT-4o (OpenAI 2024a), which is based on transformative transformer architectures (Vaswani et al. 2017). This approach takes advantage of the generative capabilities of the LLM to simulate a wide range of sarcastic expressions, enriching the dataset with complex contrasts in linguistic contexts (Shrivastava and Kumar 2021).

By incorporating synthetic examples that exhibit clear contrasts, the proposed Long Short-Term Memory (LSTM) model is expected to develop a more refined understanding of sarcasm, leading to significant improvements in detection accuracy.

The effectiveness of this methodology will be quantitatively assessed using several evaluation metrics (see Section A.3) on the iSarcasmEval dataset, which comprises deliberately sarcastic sentences curated from Twitter (Oprea and Magdy 2020).

2. The Problem

Sarcasm is a specific type of irony mostly used to mock or express contempt about something, often used to undermine or insult someone. In simple terms, irony itself is a rhetorical device that allows the speaker to express themselves in the opposite direction from what the words mean in the literal sense.

> Don't you know what sarcasm means? What a surprise!
>
> — Someone at Twitter, probably

The use of sarcasm varies and can be applied in multiple contexts, from serving humorous purposes to pointing out something obvious, or even negatively impacting another person due to their personal opinions (Kreuz 2020), which is the main goal of this work.

2.1. Miscommunications and negative interactions on social networks

Negative interactions on social media can lead to several problems. Some are relatively minor and easier to handle, such as getting upset over someone's disagreement on the Internet. However, these interactions can also escalate to cyberbullying and even threaten democratic values (Klofstad, Sokhey, and McClurg 2013).

Moreover, sarcasm can serve as a tool for hate speech (Pasa, Nuriadi, and Lail 2021), particularly since properly identifying hate speech within a sarcastic comment often requires a considerable amount of context (Riloff et al. 2013; Oprea and Magdy 2020; Sarsam et al. 2020). Multilingual platforms presents even greater challenges due to the particularities of each language (Frenda 2018).

3. Related Work

According to Maynard and Greenwood (2014), a significant challenge in sentiment analysis is the difficulty in distinguishing genuine expressions from sarcastic ones in tweets. This obstacle complicates the development of accurate sentiment analysis classifiers. Their proposal includes employing several heuristics to identify sarcastic tweets by analyzing the presence of specific

hashtags, such as `#sarcasm` and `#notreally`. This solution, however, does not properly account for sarcastic tweets that lack these hashtags, which is a limitation.

Hazarika et al. (2018) present a unique approach capable of identifying sarcasm in sentences that lack context. It achieves this by using a hybrid approach that considers the context of the author, based on the user's profile and behavior on Reddit. This solution has shown to be quite effective as it is not tied to a single sentence alone.

Other correlated work, now concerning the use of machine learning techniques for synthetic data, includes the work of Kruschwitz and Schmidhuber, where they explore the capabilities of GPT-3 Curie for categorizing toxic content. Although not related to sarcasm, the tests included categorizations of hate speech and patronizing sentences. The model did not show any improvements over previous tests on GPT-2 regarding hate speech, but it did for patronizing sentences. The authors mentioned that safety filters might be affecting the model's performance specifically on hate speech, since it is incapable of generating real-world hate speech in a faithful manner.

LLMs have also been used specifically for sarcasm detection. Shrivastava and Kumar (2021) shows how BERT (Bidirectional Encoder Representations from Transformers) outperforms other models, such as Logistic Regression (LR), SVMs (Support Vector Machines), and Bayesian Probabilities (BP). It also outperformed CNNs and LSTMs.

Although Transformers have demonstrated better performance compared to LSTMs (Vaswani et al. 2017), LSTMs have not only inspired this architecture, but have also significantly evolved over the years (Gers, Schmidhuber, and Cummins 1999; Bahdanau, Cho, and Bengio 2016). Nevertheless, LSTMs show better results over Transformers for smaller datasets (Ezen-Can 2020), which is the current situation of this present work.

4. Proposal

Language models have existed for quite some time, but it was only after the introduction of GPT by OpenAI that significantly boosted the popularity of Large Language Models (LLMs). These models are not only large, as their name suggests — they are *sufficiently large* to be useful to a broad audience — which partly explains their quick popularity growth. This is due to the fact that Transformers, unlike other types of RNNs that incorporated some form of attention mechanism, can be trained in a distributed manner, therefore are capable of scaling much more efficiently (Vaswani et al. 2017).

Compared to other common neural network models, LLMs are particularly effective in recognizing sarcasm (Shrivastava and Kumar 2021), largely due to their pre-trained nature. However, researchers have shown concerns about the resource hungriness of LLMs (Bai et al. 2024), which is why this current project aims to use LLMs only for data generation, rather than relying solely on using a pre-trained model.

4.1. How can Recurrent Neural Networks help?

Although Recurrent Neural Networks (RNNs) seem quite different from other Neural Networks (NNs), they share more similarities than differences with regular Multilayer Perceptrons (MLPs), for instance (see Figure 1 and Figure 2). The defining feature of RNNs, which sets them apart from MLPs, is their ability to process sequences effectively by using hidden layers that retain a memory of previous outputs (9) (10). This feature allows RNNs to handle sequential

data, such as natural language, by looping through their hidden layers rather than simply producing outputs at each layer. (Goodfellow, Bengio, and Courville 2016)

It is indeed possible to perform sarcasm detection using classic machine learning techniques, such as Support Vector Machines (SVMs) (Sarsam et al. 2020); however, Recurrent Neural Networks (RNNs) have demonstrated superior results in these tasks.

4.1.1. The problem with long-term dependencies

Basic RNNs have a major limitation due to their architecture: gradient vanishing. Gradients are products of derivatives, and repeatedly multiplying numbers smaller than one (most often through the sigmoid (1) or tanh (2) functions) via backpropagation tends to make the gradient approach zero. In simple terms, these basic RNNs struggle to learn from longer sequences because older terms become less important over time until they are mostly 'forgotten' by the network, as seen in Figure 3.

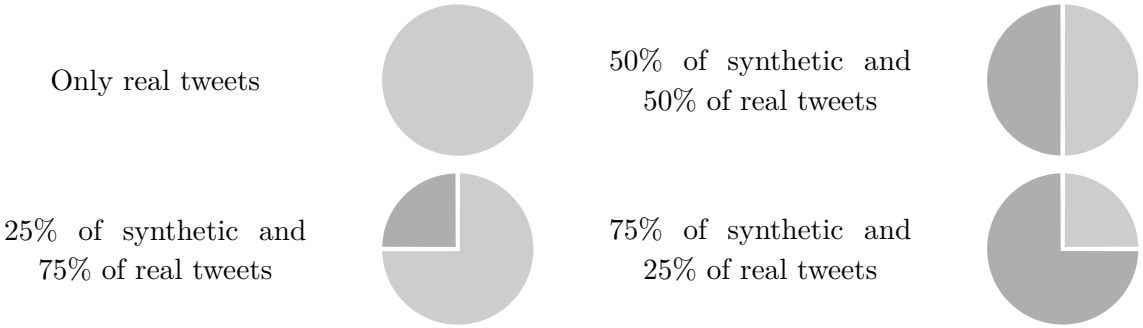
4.2. How do Long Short-Term Memory Networks overcome the gradient limitation?

Long Short-Term Memory Networks (LSTMs) (Hochreiter and Schmidhuber 1997) implement a much more robust architecture in order to avoid gradient vanishing, as seen in Figure 4. They introduce four new concepts – input gate (11), forget gate (12), update cell state (13), output gate (14), and finally the cell state (see Section A.6.2). This structure allows LSTMs to discard noisy information (12), at the same time maintaining all the relevant information in memory (see Section A.6.2).

4.3. Synthetic Data Generation

GPT-4o has been chosen as the LLM to generate synthetic sarcastic sentences due to its performance (OpenAI 2024b). The iSarcasmEval dataset contains 1,267 sarcastic and 4,868 non-sarcastic sentences (Oprea and Magdy 2020; Abu Farha et al. 2022), totaling 6,135 sentences. The strategy is to leverage the existing imbalance by generating additional sarcastic sentences until the dataset is balanced. The non-sarcastic sentences are approximately 3.8 times more numerous than the sarcastic ones, and the difference will be addressed with LLM-generated synthetic data.

Firstly, 20% of the real dataset will be reserved solely for testing, and then, for the training and validation phases, four different datasets will be constructed, each varying in the proportion of real to synthetic data:



This approach aims to explore how well the LSTM will perform under different amounts of synthetic data. Once again, it is important to note that only real tweets will be used for the evaluation to ensure the performance of the proposed model in a real-world scenario.

5. Evaluation

The effectiveness of the enhanced LSTM model will be evaluated using F1-score (8), along with other metrics as well, such as accuracy (5), precision (6) and recall (7). As mentioned previously, the dataset to be used for the evaluation is the iSarcasm dataset, which consists of sarcastic sentences curated from Twitter. The F1-score works as a single and harmonic metric, capable of capturing both accuracy and recall, therefore providing a comprehensive indicator of model performance.

For the validation step, Stratified Cross-Validation (see Section A.2.1) will be employed as an additional layer of variance ensuring.

5.1. Comparision with other Language Models

The effectiveness of the enhanced LSTM model will be benchmarked against a range of state-of-the-art language models, each with unique capabilities and architectures. The models selected for comparison are the GPT 4, GPT 4.0o, BERT, Llama 2 and Llama 3 (Achiam et al. 2023; OpenAI 2024a; Devlin et al. 2019; Touvron et al. 2023; Meta 2024).

5.2. iSarcasm dataset

The iSarcasm dataset comprises a collection of sarcastic tweets that have been manually annotated to ensure the accuracy of sarcasm labeling. This dataset includes self-annotations by the authors, reducing potential biases and inaccuracies often found in datasets labeled via distant supervision or third-party annotations (Oprea and Magdy 2020).

5.2.1. Collection and labeling

Each text in the dataset was labeled by its respective author, ensuring that the sarcasm was intended and not merely perceived by external annotators. This method is particularly effective as it captures the author's true intent, the major aspect of sarcasm that can be often lost in other labeling techniques (Oprea and Magdy 2020).

6. Expected results

In a nutshell, by using synthetic data, the LSTM model is expected to gain exposure to a wider variety of sarcasm, enhancing its ability to recognize and differentiate sarcastic expressions in text. More importantly, it enables the LSTM to better learn the contrasts between literal and intended meanings, which are essential for detecting sarcasm. This is a well-known strategy (Riloff et al. 2013) and is based on the idea that Recurrent Neural Networks are particularly good at identifying context and the relationship between words within a given context, provided they are fed with enough data.

One of the key aspects of not solely relying on a pre-trained LLM is to make the model accessible and mostly being able to use very few resources. By having a lightweight model, it is expected for this model to be easily embedded via libraries such as TensorflowJS (Smilkov et al. 2019)

or KotlinDL (JetBrains 2023), ensuring that the model is not only effective but also accessible and adaptable to low resource environments.

6.1. Challenges

There are several challenges that may impact the overall results of this project. Most concerns revolve around how effective synthetic data can be, which is the initial hypothesis of this work.

6.1.1. Too much synthetic data

One potential outcome of this project is a model that performs well on LLM-generated sentences but fails to perform effectively on real data. The choice of LSTMs over Transformers is motivated by their architecture, which is less prone to overfitting on small datasets. (Ezen-Can 2020) Additionally, the aim is not only for the network to learn from the data itself but also to recognize the contrast typically present in sarcastic sentences. This is crucial to the hypothesis that LLMs can assist training smaller, more specialized models.

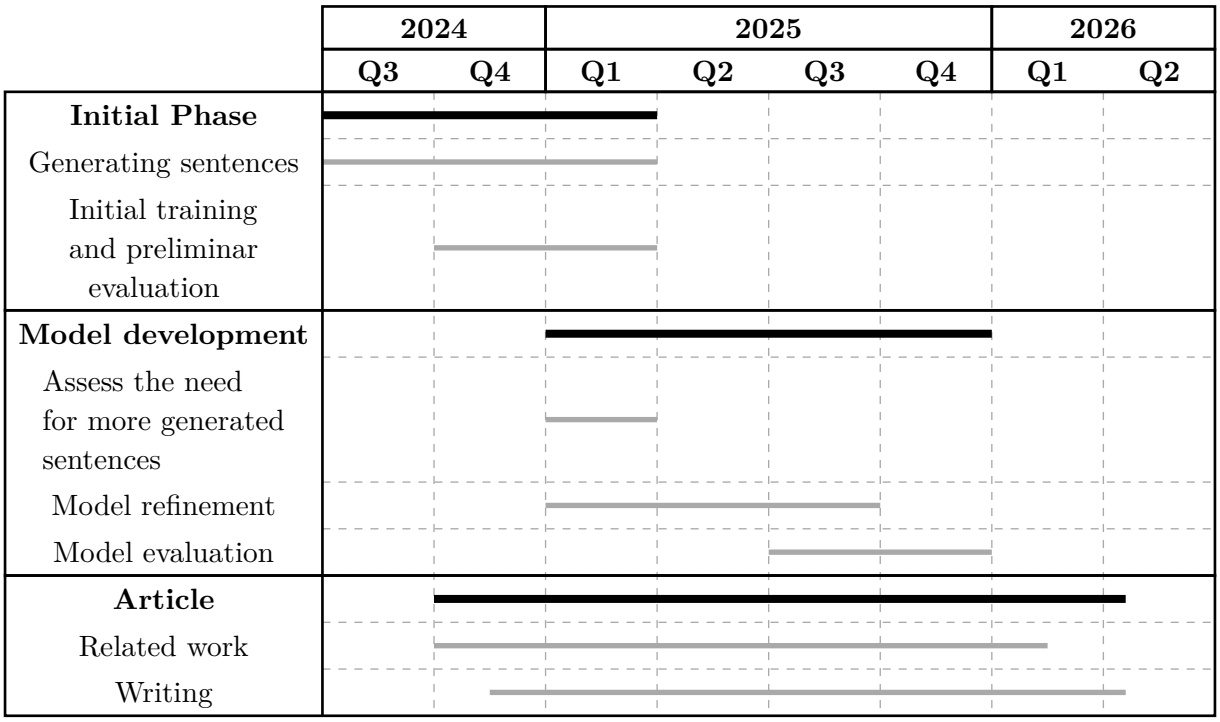
6.1.2. As effective as the LLM

The effectiveness of the trained network might be constrained by the LLM's ability to generate convincing sarcastic sentences. This could become a problem depending on the final evaluation, especially since modern LLMs are quite good at recognizing sarcasm. (Bai et al. 2024)

6.1.3. Sarcasm is not static

Sarcasm, like any aspect of human language, evolves rapidly (Kreuz 2020). This presents a challenge for any model because recognizing contrast is not the sole determinant of the model's effectiveness. Once evaluations confirm that LLMs are effective in training smaller, more specific models, continual generation of new sequences will be necessary to keep the model as up-to-date and effective as possible.

7. Timeline



Bibliography

[1] E. Riloff, A. Qadir, P. Surve, L. De Silva, N. Gilbert, and R. Huang, "Sarcasm as Contrast between a Positive Sentiment and Negative Situation," in *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, D. Yarowsky, T. Baldwin, A. Korhonen, K. Livescu, and S. Bethard, Eds., Seattle, Washington, USA: Association for Computational Linguistics, Oct. 2013, pp. 704–714. [Online]. Available: https://aclanthology.org/D13-1066

[2] A. A. Johnson and R. J. Kreuz, "Sarcasm Across Time and Space: Patterns of Usage by Age, Gender, and Region in the United States," *Discourse Processes*, vol. 60, no. 1, pp. 1–17, 2023, doi: 10.1080/0163853X.2022.2085475.

[3] OpenAI, "Hello GPT-4o." Accessed: May 26, 2024a. [Online]. Available: https://openai.com/index/hello-gpt-4o/

[4] A. Vaswani *et al.*, "Attention Is All You Need," *CoRR*, 2017, [Online]. Available: http://arxiv.org/abs/1706.03762

[5] M. Shrivastava and S. Kumar, "A pragmatic and intelligent model for sarcasm detection in social media text," *Technology in Society*, vol. 64, p. 101489–101490, 2021, doi: https://doi.org/10.1016/j.techsoc.2020.101489.

[6] S. Oprea and W. Magdy, "iSarcasm: A Dataset of Intended Sarcasm," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, D. Jurafsky, J. Chai, N. Schluter, and J. Tetreault, Eds., Online: Association for Computational Linguistics, Jul. 2020, pp. 1279–1289. doi: 10.18653/v1/2020.acl-main.118.

[7] R. J. Kreuz, *Irony and sarcasm*. The Mit Press, 2020. [Online]. Available: https://mitpress.mit.edu/9780262538268/

[8] C. A. Klofstad, A. E. Sokhey, and S. D. McClurg, "Disagreeing about Disagreement: How Conflict in Social Networks Affects Political Behavior," *American Journal of Political Science*, vol. 57, no. 1, pp. 120–134, 2013, doi: https://doi.org/10.1111/j.1540-5907.2012.00620.x.

[9] T. A. Pasa, Nuriadi, and H. Lail, "AN ANALYSIS OF SARCASM ON HATE SPEECH UTTERANCES ON JUST JARED INSTAGRAM ACCOUNT," *Journal of English Education Forum (JEEF)*, vol. 1, no. 1, pp. 10–19, Jun. 2021, [Online]. Available: https://jeef.unram.ac.id/index.php/jeef/article/view/94

[10] S. M. Sarsam, H. Al-Samarraie, A. I. Alzahrani, and B. Wright, " Sarcasm detection using machine learning algorithms in Twitter: A systematic review ," *International Journal of Market Research*, vol. 62, no. 5, pp. 578–598, 2020, doi: 10.1177/1470785320921779.

[11] S. Frenda, "The role of sarcasm in hate speech. A multilingual perspective," 2018, pp. 13–17. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85057593937&partnerID=40&md5=2c898e27c0620ef5e1a9e78397aacbb3

[12] D. Maynard and M. A. Greenwood, "Who cares about Sarcastic Tweets? Investigating the Impact of Sarcasm on Sentiment Analysis.," in *International Conference on Language Resources and Evaluation*, 2014. [Online]. Available: https://api.semanticscholar.org/CorpusID:14079970

[13] D. Hazarika, S. Poria, S. Gorantla, E. Cambria, R. Zimmermann, and R. Mihalcea, "CASCADE: Contextual Sarcasm Detection in Online Discussion Forums," in *Proceedings of the 27th International Conference on Computational Linguistics*, E. M. Bender, L. Derczynski, and P. Isabelle, Eds., Santa Fe, New Mexico, USA: Association for Computational Linguistics, Aug. 2018, pp. 1837–1848. [Online]. Available: https://aclanthology.org/C18-1156

[14] U. Kruschwitz and M. Schmidhuber, "LLM-Based Synthetic Datasets: Applications and Limitations in Toxicity Detection," in *Proceedings of the Fourth Workshop on Threat, Aggression & Cyberbullying @ LREC-COLING-2024*, R. Kumar, A. K. Ojha, S. Malmasi, B. R. Chakravarthi, B. Lahiri, S. Singh, and S. Ratan, Eds., Torino, Italia: ELRA, ICCL, May 2024, pp. 37–51. [Online]. Available: https://aclanthology.org/2024.trac-1.6

[15] F. Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: continual prediction with LSTM," in *1999 Ninth International Conference on Artificial Neural Networks ICANN 99. (Conf. Publ. No. 470)*, Sep. 1999, pp. 850–855. doi: 10.1049/cp:19991218.

[16] D. Bahdanau, K. Cho, and Y. Bengio, "Neural Machine Translation by Jointly Learning to Align and Translate." 2016.

[17] A. Ezen-Can, "A Comparison of LSTM and BERT for Small Corpus." 2020.

[18] G. Bai *et al.*, "Beyond Efficiency: A Systematic Survey of Resource-Efficient Large Language Models." 2024.

[19] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.

[20] S. Hochreiter and J. Schmidhuber, "Long Short-term Memory," *Neural computation*, vol. 9, pp. 1735–1780, 1997, doi: 10.1162/neco.1997.9.8.1735.

[21] OpenAI, "simple-evals: Repository for Evaluating Language Models." Accessed: Jun. 10, 2024b. [Online]. Available: https://github.com/openai/simple-evals

[22] I. Abu Farha, S. V. Oprea, S. Wilson, and W. Magdy, "SemEval-2022 Task 6: iSarcasmEval, Intended Sarcasm Detection in English and Arabic," in *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*, Seattle, United States: Association for Computational Linguistics, Jul. 2022, pp. 802–814. [Online]. Available: https://aclanthology.org/2022.semeval-1.111

[23] O. J. Achiam *et al.*, "GPT-4 Technical Report," 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:257532815

[24] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding." 2019.

[25] H. Touvron *et al.*, "Llama 2: Open Foundation and Fine-Tuned Chat Models," *ArXiv*, 2023, [Online]. Available: https://api.semanticscholar.org/CorpusID:259950998

[26] Meta, " Introducing Meta Llama 3: The most capable openly available LLM to date ." [Online]. Available: https://ai.meta.com/blog/meta-llama-3/

[27] D. Smilkov *et al.*, "TensorFlow.js: Machine Learning for the Web and Beyond," *CoRR*, 2019, [Online]. Available: http://arxiv.org/abs/1901.05350

[28] JetBrains, "KotlinDL: High-level Deep Learning API in Kotlin." Accessed: Jun. 10, 2024. [Online]. Available: https://github.com/Kotlin/kotlindl

APPENDIX A

A.1 Activation functions

A.1.1 Sigmoid

$$\sigma(x) = \frac{1}{1} + e^{-x} \tag{1}$$

A.1.2 Tanh

$$\tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} \tag{2}$$

A.2 Cross Validation

A dataset D is split into k equal parts. The model is trained and validated k times, with one of the parts D_k being used as the validation set V_i in each iteration, while the other parts T_i are used for training. Performance metrics are calculated at each step. The overall error rate is the average of the error rates from each step, where e_i is the error rate in the i-th iteration:

$$E = \frac{1}{k} \cdot \sum_{i=1}^{k} e_i \tag{3}$$

A.2.1 Stratified Cross Validation

The dataset D is first stratified into k parts (D_1, D_2, ..., D_k), ensuring that the sample ratio for each class in each part D_i is as close as possible to the ratio of that class in the complete dataset D. If C represents a class label, then the class ratio C in each D_i, denoted as p_{C_i}, should closely match p_C, the ratio of the class C throughout the dataset D:

$$p_{C_i} \approx p_C, \quad \forall i = 1, 2, ..., k \tag{4}$$

Once the data is stratified, cross validation proceeds as usual (Section A.2). Each part D_i is used once as a validation set V_i, while the remaining combined parts form the training set T_i.

A.3 Evaluation metrics

- TP: True Positives
- TN: True Negatives
- FP: False Positives
- FN: False Negatives

$$accuracy = \frac{TP + TN}{TP + TN + FN + FP} \tag{5}$$

$$precision = \frac{TP}{TP + FP} \tag{6}$$

$$recall = \frac{TP}{TP + FN} \tag{7}$$

$$f1 = \frac{2 \cdot (precision \cdot recall)}{precision + recall} \tag{8}$$

A.4 MLP Architecture

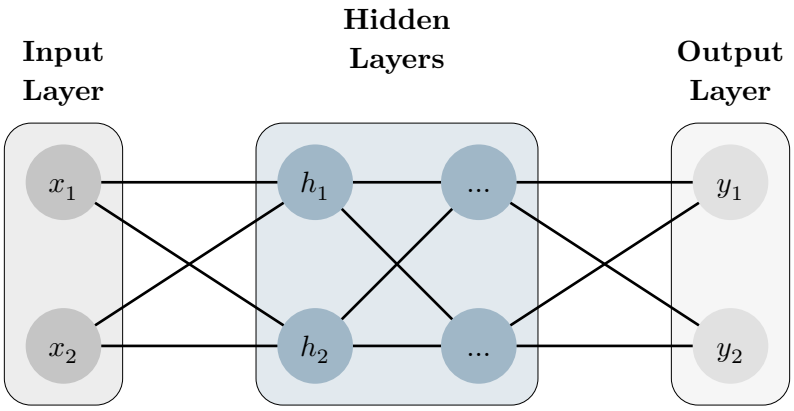


Figure 1: The architecture of a Multilayer Perceptron

A.5 Recurrent Neural Networks

A.5.1 Architecture

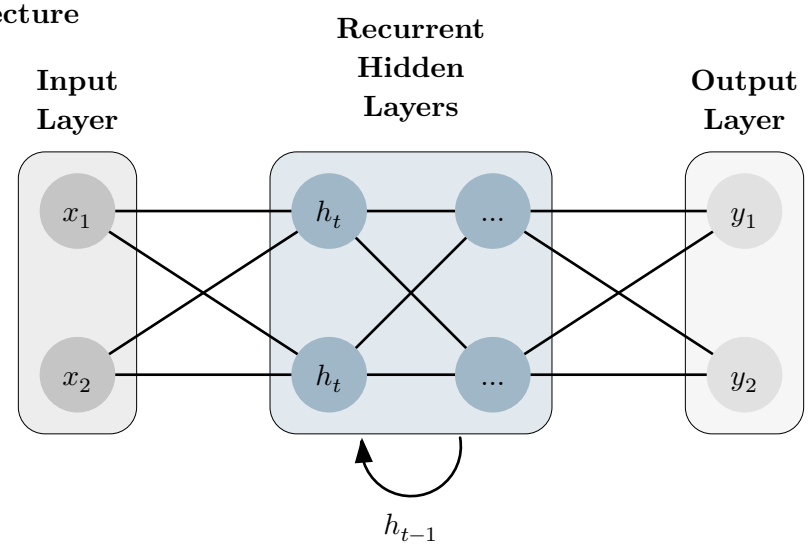


Figure 2: The architecture of a Recurrent Neural Network

A.5.2 Hidden State Update

$$h_t = f(W_h c \cdot h_{t-1} + W_x c \cdot x_t + b_h) \tag{9}$$

- h_t: Hidden state at time t.
- h_{t-1}: Hidden state from the previous time step.
- x_t: Input at time t.
- W_h: Weight matrix for the hidden state.
- W_x: Weight matrix for the input.
- b_h: Bias term for the hidden state.
- f: Activation function.

A.5.3 Output

$$y_t = W_y c \cdot h_t + b_y \tag{10}$$

- y_t: Output at time t.
- W_y: Weight matrix for the output.
- b_y: Bias term for the output y

A.5.4 The Vanishing Gradient Problem

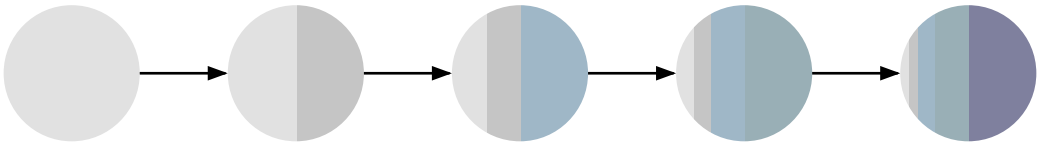


Figure 3: A simple representation of the vanishing problem

A.6 Long Short-Term Memory Networks

A.6.1 Architecture

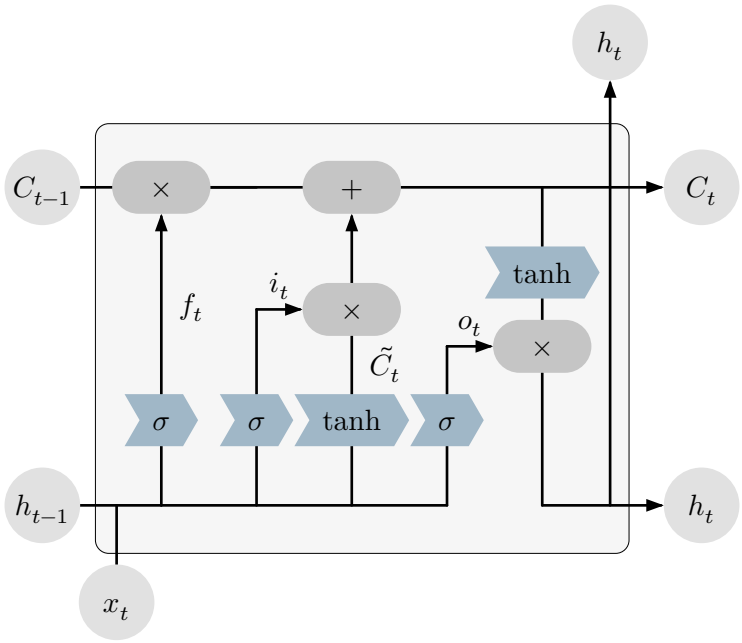


Figure 4: The architecture of a Long Short-Term Memory Network

A.6.2 Cell state

This is the "memory" part of the LSTM, carrying relevant information throughout the processing of the sequence. Plays a major role in transferring past knowledge to future states.

A.6.3 Input gate

Decides how much of the newly computed state for the current input x_t should be added to the cell state.

$$\begin{aligned} i_t &= \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \\ \tilde{C}_t &= \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \end{aligned} \tag{11}$$

A.6.4 Forget gate

Decides how much of the current cell state should be kept. Anything that was not forgotten is passed along to the next step.

$$f_t = \sigma\big(W_f \cdot [h_{t-1}, x_t] + b_f\big) \tag{12}$$

A.6.5 Update Cell State

This step combines the old state C_{t-1} and the new candidate values, modulated by the forget gate and the input gate.

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \tilde{C}_t \tag{13}$$

A.6.6 Output

Decides what part of the cell state should be output at this step

$$\begin{aligned} o_t &= \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \\ h_t &= o_t \cdot \tanh(C_t) \end{aligned} \tag{14}$$