

# Fine-tuning the Gemma model for Kaggle Assistant

**Arman Zhalgasbayev**

Software Engineering Student

Astana IT University

Astana, Kazakhstan

[220650@astanait.edu.kz](mailto:220650@astanait.edu.kz)

<https://orcid.org/0000-0001-8507-6845>

**Ali Khauazkhan**

Big Data Analysis Student

Astana IT University

Astana, Kazakhstan

[230378@astanait.edu.kz](mailto:230378@astanait.edu.kz)

<https://orcid.org/0009-0007-2411-2145>

**Zhibek Sarsenova**

Computer Engineering department

Astana IT University

Astana, Kazakhstan

[zhibek.sarsenova@astanait.edu.kz](mailto:zhibek.sarsenova@astanait.edu.kz)

<https://orcid.org/0000-0002-7780-8070>

**Abstract:** The emergence of advanced open-source large language models (LLMs) opens new opportunities for researchers and accelerates the development of generative AI. This study focuses on harnessing the potential of Gemma, Google's recently released open source LLM, to develop the Kaggle assistant through fine-tuning. The main objective of the study was to investigate the Low-Rank Adaptation (LoRA) fine-tuning method of Gemma model for building a robust AI assistant in Kaggle. Using the Gemma 2B variant, this study applies the Latent Consistency Model - LoRA method to improve interpretability and performance. The fine-tuning of the model achieved a sparse categorical cross-entropy (Loss) of 0.4943 and a sparse categorical precision of 0.7753. Although experiments with epochs and batch sizes revealed potential overbuilding after 10 epochs, the model shows acceptable performance in this range. The integration of LoRA fine-tuning enhances the model's ability to provide accurate recommendations and conclusions to users solving Kaggle problems, thereby improving the user experience and reinforcing its utility as a valuable resource for data scientists. Further refinement and adaptation of the Gemma model, especially through techniques such as LoRA fine-tuning, are critical to ensure its continued effectiveness in assisting Kaggle users in their data science endeavors.

**Keywords:** LLM, Gemma model, LoRA fine-tuning, Kaggle

## I. INTRODUCTION

The release of new advanced open-source large language models (LLMs) by major technology companies opens new opportunities for researchers and accelerates the development of Generative AI. Linux Operating System (OS) founder Linus Torvalds said that "for complex technical issues, you really need open source, simply because the problem space turns out to be too complex to be managed within a single company" [1]. Transparency in LLM technology will be a key factor in creating Artificial General Intelligence (AGI) in the near future, which is why major corporations like: Google, Microsoft and Meta are actively working on openness of AI models so that young researchers learn and can be able to skillfully utilize the gifts of modern LLM technology.

Google recently released a new open-source LLM called Gemma. Gemma is a family of lightweight, state-of-the-art open-source models based on the same research and technology used to create Google DeepMind's Gemini models, released in 21st of February [2]. Gemma has 2 variants with different weights: Gemma 2B and Gemma 7B (nB denotes the range of the parameters). To widely popularize the model, the Google team has integrated the Gemma model with popular platforms for Data Science and Machine Learning (DS/ML) researchers, such as Kaggle, HuggingFace, MaxText, NVIDIA NeMo, published articles, guides and tutorials on how to use the model, also Google on the Kaggle platform, announced a major competition to develop DS/ML assistants based on the LLM model Gemma

called «Google – AI Assistants for Data Tasks with Gemma» competition [3].

The goal of this research, as part of the "Google - AI Assistants for Data Tasks with Gemma" competition, is to develop a Kaggle assistant based on fine-tuning of the Gemma model. A custom dataset will be used to train the model, as there are no well-organized Kaggle FAQ datasets in the public domain. The Gemma 2B model with 2 billion parameters will be used as the main model as it is lighter and faster. Latent Consistency Model - Low-Rank Adaptation (LoRA) technique will be used to fine tune the model as it enhance the model's interpretability and performance.

## II. BACKGROUND/LITERATURE REVIEW

Gemma is a new LLM released less than a month ago, so an article by the Gemma Team and Google DeepMind was used as the main resource of research. Articles on fine-tuning LLMs and LoRA techniques were also used to improve model performance for a specific domain.

### A. Gemma open-source LLM by Google

The article "Gemma: Open Models Based on Gemini" by the Gemma Team at Google DeepMind introduces Gemma, a family of open models derived from the research and technology underlying Gemini models. Gemini models are renowned for their robust performance in language understanding (fig. 1), reasoning, and safety across academic benchmarks. Gemma models, offered in two sizes with 2 billion and 7 billion parameters respectively, outperform similarly sized open models on 11 out of 18 text-based tasks (table 1) [2].

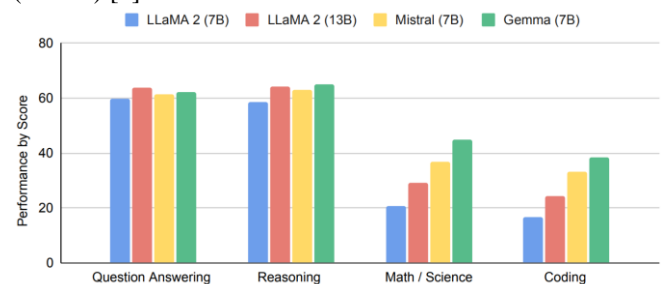


Fig. 1. Comparison of the Gemma 7B model with other open-source LLMs by language understanding and generation performance [2].

Benchmark	metric	LLaMA-2		Mistral		Gemma	
		7B	13B	7B	2B	7B	
MMLU	5-shot, top-1	45.3	54.8	62.5	42.3	<b>64.3</b>	
HellaSwag	0-shot	77.2	80.7	81.0	71.4	<b>81.2</b>	
PIQA	0-shot	78.8	80.5	<b>82.2</b>	77.3	81.2	
SIQA	0-shot	48.3	50.3	47.0*	49.7	<b>51.8</b>	
Boolq	0-shot	77.4	81.7	<b>83.2*</b>	69.4	<b>83.2</b>	
Winogrande	partial scoring	69.2	72.8	<b>74.2</b>	65.4	72.3	
CQA	7-shot	57.8	67.3	66.3*	65.3	<b>71.3</b>	
OBQA		<b>58.6</b>	57.0	52.2	47.8	52.8	
ARC-e		75.2	77.3	80.5	73.2	<b>81.5</b>	
ARC-c		45.9	49.4	<b>54.9</b>	42.1	53.2	
TriviaQA	5-shot	72.1	<b>79.6</b>	62.5	53.2	63.4	
NQ	5-shot	25.7	<b>31.2</b>	23.2	12.5	23.0	
HumanEval	pass@1	12.8	18.3	26.2	22.0	<b>32.3</b>	
MBPP <sup>†</sup>	3-shot	20.8	30.6	40.2*	29.2	<b>44.4</b>	
GSM8K	maj@1	14.6	28.7	35.4*	17.7	<b>46.4</b>	
MATH	4-shot	2.5	3.9	12.7	11.8	<b>24.3</b>	
AGIEval		29.3	39.1	41.2*	24.2	<b>41.7</b>	
BBH		32.6	39.4	<b>56.1*</b>	35.2	55.1	
Average		47.0	52.2	54.0	44.9	<b>56.4</b>	

Table 1. Comparison of the Gemma models with other open-source LLMs by Academic benchmark results [2].

The data from the diagram of fig. 1 and table 1 confirm the impressive results of the Gemma model in understanding the language, generating responses and academic knowledge.

### B. LoRA Fine-Tuning of LLMs

The article "LoRA: Low-Rank Adaptation of Large Language Models" by Edward J. Hu et al. introduces LoRA, a novel approach for adapting large language models to specific tasks or domains while mitigating the computational and memory overhead associated with full fine-tuning. LoRA addresses the challenge of fine-tuning large-scale models on general domain for performing downstream tasks by proposing a low-rank adaptation strategy that freezes the pre-trained model weights and introduces trainable rank decomposition matrices into each layer of the Transformer architecture. Compared to traditional fine-tuning methods, LoRA can achieve significant parameter reduction (up to 10,000 times) and reduce GPU memory requirements (up to 3 times) [4].

### C. Kaggle Platform for DS/ML enthusiasts

Kaggle stands as an online community platform designed for DS/ML enthusiasts. Founded in 2010 by Anthony Goldbloom and Jeremy Howard, and subsequently acquired by Google in 2017, Kaggle offers a wealth of resources and tools to assist researchers in their data science endeavors [5]. Kaggle Key features include:

- 1) **Collaboration:** Kaggle fosters collaboration among users, enabling them to work together on projects, share insights, and learn from one another.
- 2) **Datasets:** Users can discover and publish datasets on Kaggle, making it valuable resource for accessing diverse data for analysis and modeling.
- 3) **Competitions:** Kaggle provides data scientists with convenient platform for DS/ML competitions, challenging tasks, and generous rewards. Participating in these competitions allows researchers to apply their skills, learn new techniques, and potentially earn recognition and prizes.

4) **Notebooks:** Kaggle Notebooks provide integrated environments for writing and executing the code, leveraging cloud resources to handle heavy computations and eliminating the need for users to set up their own local development environment [6].

5) **Models:** Kaggle provides the opportunity to use the most advanced open-source ML models, and train custom ML models and publish them on the platform.

## III. METHODOLOGY

In this research we used content-analysis methodology to get more in-depth knowledge about the research object, these are: the Gemma model, Kaggle platform and LoRA fine-tuning technique. After conducting the theoretical part of the study, we implemented the practical part, in which we developed a Notebook for fine-tuning the LLM Gemma to perform the function of Kaggle assistant. Kaggle Notebook was implemented in 8 major steps. There are:

- 1) **Setup dependencies** – install all the necessary modules and dependencies;
- 2) **Load dataset** – load, preprocess and format 3 datasets (Kaggle QnA, Python QnA and Data Science QnA) into 1 common well-formatted dataframe;
- 3) **Data preparation** – text-size limiting to avoid memory limit exceed, transforming data to appropriate format for the LoRA training model fitting;
- 4) **Load model** – loading pre-trained Gemma 2B model;
- 5) **Inference before fine tuning** – 5 control questions about Kaggle to assess the not fine-tuned model performance;
- 6) **LoRA fine-tuning** – enhance model’s interpretability and performance for the Kaggle Assistant domain based on prepared dataset;
- 7) **Inference after fine-tuning** – 5 control questions about Kaggle, as before, to assess the fine-tuned model performance and improvements;
- 8) **Fine-tuned model save and load instructions** – for model reusing;

### A. Data Collection

In the data collection phase, we collected 1 custom dataset to train the Kaggle assistant and used 2 public datasets from Kaggle on Python & Data Science.

- To fine-tune the model, we collected all the data in question-and-answer (QnA) format from Stackoverflow for the keyword "Kaggle" using Stackexchange API v2.3 [7] [8].
- Additional Dataset for Python Knowledge “MBPP Python Problems jsonl” (Author’s username: Marília Prata) [9].
- Additional Dataset for Data Science Knowledge “Data Science questions and answers” (Author’s username: Nikhil) [10].

### B. Setup Dependencies

- keras – working with deep learning models;

- keras\_nlp – working with NLP-related functionalities;
- numpy – linear algebra and numerical computations;
- pandas – data processing;
- re – handling regular expressions;
- time – checking computation time;
- json – handling json-formatted files;
- os – operating system functions;
- warnings – for filtering warnings and produce less noise;
- IPython.display – for well-formatted Markdown and display functions;

### C. Load Dataset:

Dataset	Number of data rows
Kaggle QnA	821
Python QnA	974
Data Science QnA	158
Total QnA	1953

Table 2. Information about datasets.

	question	answer	question_length	answer_length
302	Problems loading csv files into mysql with spa...	I guess you are working on Kaggle's Grupo Bimb...	86	103
25	Write a function to check if the given tuple l...	def check_k_elements(test_list, K): res = ...	69	153
84	What is PCA (Principal Component Analysis)?	PCA is a linear dimensionality reduction techn...	43	170
643	Is number of tasks same as the number of fits ...	Deciphering it step by step Fitting 3 folds fo...	83	627
458	Merging in R returns columns full of NA values	Ok after fighting back and forth I found out l...	46	510

Table 3. Sample 5 data rows from Total QnA dataset.

### D. Data Preparation

We precalculated lengths of questions and answers for setting up maximum length for data preprocessing, to avoid memory limit exceed. Only 95 rows are greater than 1024, so we can limit the length of all texts to 1024.

```
def text_limit(text: str) -> str:
    max_len = 1024
    return text[:max_len] if len(text) > max_len else text

kaggle_df['question'] = kaggle_df['question'].apply(text_limit)
kaggle_df['answer'] = kaggle_df['answer'].apply(text_limit)
```

Fig. 2. Code snippet for limiting text sizes.

By limiting the text size, we converted our dataset to a list with string objects in the QnA format.

### E. Load Model

We used the GemmaCausalLM model using the “gemma\_2b\_en” pre-trained model settings (Gemma 2B). Model summary information:

Layer (type)	Output Shape	Param #
padding_mask (InputLayer)	(None, None)	0
token_ids (InputLayer)	(None, None)	0
gemma_backbone (GemmaBackbone)	(None, None, 2048)	2,506,172,416
token_embedding (ReversibleEmbedding)	(None, None, 256000)	524,288,000

Table 4. Gemma Model Summary information.

- Total parameters: 2,506,172,416 (9.34 GB);
- Trainable parameters: 2,506,172,416 (9.34 GB);
- Non-trainable parameters: 0 (0.00 B);

Also, in this stage we defined custom “get\_answer” function for getting well-formatted response from the Gemma model.

```
def get_answer(question: str, max_length: int = 256) -> str:
    global gemma_lm

    template = "Question:\n{question}\n\nAnswer:\n{answer}"

    prompt = template.format(
        question = question,
        answer = ""
    )

    answer = gemma_lm.generate(
        inputs = prompt,
        max_length = max_length
    )

    return answer
```

Fig. 3. Code snippet for “get\_answer” function.

### F. Inference before fine-tuning

In this phase, we defined 5 benchmark questions to evaluate the performance of our model before and after fine-tuning LoRA to assess the improvements or any other changes.

- Qn – means n<sup>th</sup> control question.
- An – means response of the model to the Qn (not full because it takes too much space).

Q1: What is a Kaggle?

A1: “Kaggle is an organization that hosts competitions, challenges, and events for data scientists, data enthusiasts, and machine learning practitioners. It’s a place to share data and ideas, learn from others, and collaborate with your peers. The best data scientists are not just good at crunching numbers. They’re also great at analyzing the numbers and finding insights that others may not have seen. Kaggle competitions are an excellent opportunity for data scientists to put their skills to the test. They’ll be challenged to find patterns, spot trends, and make sense of the data that they’re given. The competition is open to anyone, so it’s a great way for aspiring data scientists to learn from the best. And the best part? Kaggle offers prizes for the top performers in each competition – so it’s worth giving it a go!”

Q2: How to create a new notebook in Kaggle?

A2: “1) Go to your profile, then click on “My Notebooks”. 2) If the notebook you need does not exist, click on the “Create” button. 3) Select the template that best corresponds to your needs. 4) Fill the form and click on “Create”. 5) The new notebook will appear on the “My Notebooks” page.”

Q3: Can you write me code for checking that GPU is available on Kaggle Notebook?

A3: “import torch import torch.nn.parallel from torch import cuda import torch.backends.cudnn as cudnn from time import gmtime, strftime device = "cuda" if torch.cuda.is\_available() else "cpu" print(f"Using {device} device)”

Q4: How to become a grandmaster in Kaggle?

A4: “1) Get a top place. 2) Get a certificate. 3) Become a Grandmaster. 4) Enjoy your status.”

Q5: How to win a Kaggle Competition?

A5: “Kaggle Competitions are a way to learn, practice, and improve your skills. They’re also a great way to network with

other data scientists, and even compete against them. Here are 6 tips on how to win a Kaggle Competition.”

According to the initial answers to the control questions, it can be assessed that the model does not do well, but the answers are inaccurate and unstructured.

### G. LoRA fine-tuning

LoRA (Low-Rank Adaptation) is a technique employed in deep learning models to improve their interpretability and performance. Specifically, LoRA adjusts the relevance of each layer to better capture the hierarchical structure and semantic relationships within text data. [11].

The rank of the LoRA approximation controls the computational complexity and the degree of approximation in the relevance adjustment process.

Higher ranks may result in more accurate but computationally expensive adjustments, while lower ranks provide faster but less precise adjustments.

```
# Enable LoRA for Gemma 2B model
gemma_lm.backbone.enable_lora(rank=64)

# Display a summary of the Gemma 2B model after enabling LoRA
gemma_lm.summary()
```

Fig. 4. Code snippet for enabling LoRA.

Layer (type)	Output Shape	Param #
padding_mask (InputLayer)	(None, None)	0
token_ids (InputLayer)	(None, None)	0
gemma_backbone (GemmaBackbone)	(None, None, 2048)	2,527,995,904
token_embedding (ReversibleEmbedding)	(None, None, 256000)	524,288,000

Table 5. Gemma Model Summary information after enabling LoRA.

- Total parameters: 2,527,995,904 (9.42 GB);
- Trainable parameters: 21,823,488 (83.25 MB);
- Non-trainable parameters: 2,506,172,416 (9.34 GB);

During the development process, parameters were adjusted for comparatively better fine-tuning of the model.

### Model fine-tuning features:

- Sequence Length of 256 – Maximum number of tokens considered in each input sequence.
- Setting up AdamW optimizer (learning rate = 5e-5, weight\_decay = 0.01, beta\_1 = 0.9, beta\_2 = 0.99) AdamW has improved generalization and performance, particularly in scenarios where traditional Adam may struggle [12].
- Regularization based on bias and scale to prevent overfitting;
- Compiling the model using the Sparse Categorical Crossentropy as a main loss function, with pre-defined AdamW optimizer and monitoring the model performance by Sparse Categorical Accuracy metric. In sparse categorical crossentropy, the true labels are provided as integers rather than one-hot encoded vectors, making it suitable for cases where there are multiple classes and only one class is the correct label for each sample [13].

- Finally, fit the model with prepared Kaggle data, with epoch size of 10 (iterations number), and batch size of 1.

### H. Inference after fine-tuning

Q1: What is a Kaggle?

A1: “Kaggle is a platform that enables data scientists and researchers to collaborate, share and compete in the field of data science. Kaggle competitions are events where data scientists and researchers compete to generate the best predictive models using publicly available datasets.”

Q2: How to create a new notebook in Kaggle?

A2: “To create a new notebook in Kaggle follow these steps:

- 1) Log into Kaggle at <https://www.kaggle.com/>
- 2) Click on the "New Notebook" button in the top right corner.
- 3) Select the type of notebook you want to create (public, private, etc.)
- 4) Provide any additional information you may have (e.g. a title for the notebook).
- 5) Click on the "Create notebook" button.

Your new Kaggle notebook is ready! You can start adding cells, inserting code, and editing your content.”

Q3: Can you write me code for checking that GPU is available on Kaggle Notebook?

A3: “import subprocess if 'NVIDIA GPU' in subprocess.Popen(['nvidia-smi']).stdout.readline(), tabypes.StreamEtimología('none')) or 'Quadro' in”

Q4: How to become a grandmaster in Kaggle?

A4: “To become a grandmaster in Kaggle, you need to accumulate a certain number of points and rank within the top 1% of competitors worldwide.”

Q5: How to win a Kaggle Competition?

A5: “Firstly, you must register as a participant and then create a report for each competition. The Kaggle platform provides several features to help you create a winning report, including code editors, report templates, and support from the community. References: Kaggle Competitions Page”

By analyzing these answers, we can see that the answers have become more structured and clear. We got a satisfactory answer to 4 out of 5 questions, but the 3rd question had a very specific answer about checking the availability of NVIDIA GPUs, which is not an accurate answer.

### I. Fine-tuned model save and load instructions.

The final stage is to save the model for further use and implementation.

```
gemma_lm.save("gemma_kaggle_assistant.keras")

del gemma_lm

gemma_lm = keras.models.load_model("gemma_kaggle_assistant.keras", custom_objects={"GemmaCausalLM": keras_nlp.models.GemmaCausalLM})
```

Fig. 5. Code snippet for brief explanation on how to save and load the fine-tuned model.



#### IV. RESULTS AND ANALYSIS

Results of the fine-tuned model:

- Sparse Categorical Crossentropy (Loss): 0.4943;
- Sparse Categorical Accuracy: 0.7753;

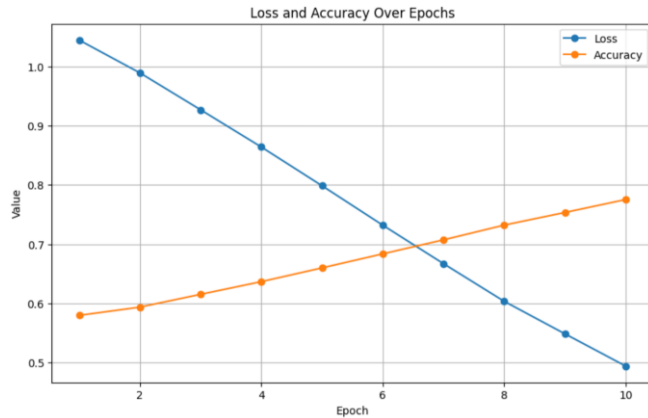


Fig. 6. Sparse Categorical Loss and Accuracy by epochs plot.

**Sparse Categorical Crossentropy (Loss):** this metric measures the difference between the predicted probability distribution and the actual probability distribution of the target classes. In our case there are instructions and responses. A lower value indicates better performance, as it suggests that the model's predictions are closer to the true labels.

**Sparse Categorical Accuracy:** this metric calculates the proportion of correctly classified samples out of the total number of samples. It is a standard formula of the model's accuracy in predicting the correct class labels. A higher value signifies better performance, indicating that the model is making accurate predictions.

In our case, the accuracy of the fine-tuned model was 0.7753 and the loss was 0.4943. We experimented with other values of epochs and batch sizes, but higher values of epochs (more than 10) overfitted the model, and the model could not answer the unknown questions correctly. Up to 10 epochs, the model performed approximately well, but above 10 the model's response had a loose structure and unrelated meaning.

Inferences on the model after fine-tuning show that the model performs reasonably well, with relatively low loss and a satisfactory level of accuracy in classifying target labels. However, further analysis and evaluation is required to fully evaluate the performance of the model and its suitability for the Kaggle assistant role.

#### V. CONCLUSION

In conclusion, the implementation of LoRA fine-tuning on the Gemma model, based on Kaggle, Python and Data Science QnA datasets, has proven to be instrumental in creating effective model for the Kaggle Assistant.

By leveraging LoRA, we were able to enhance the model's interpretability and performance, allowing it to provide more accurate recommendations and insights to users navigating Kaggle challenges. This optimization process not only

improves the user experience on the Kaggle platform but also reinforces the Gemma model's utility as a valuable resource for data scientists and enthusiasts engaged in competitive data analysis.

Moving forward, continued refinement and adaptation of the Gemma model through technique such as LoRA fine-tuning will be essential to ensure its continued effectiveness in assisting Kaggle users in their data science endeavors.

#### REFERENCES

- [1] Andrews, J. (2021). An Interview With Linus Torvalds: Open Source And Beyond - Part 2. 30 Years Of Linux. Retrieved from <https://b2a.kz/kjV>
- [2] Gemma Team and Google DeepMind (2024). Gemma: Open Models Based on Gemini Research and Technology. Retrieved from <https://b2a.kz/2ip>
- [3] Mooney, P., Chow, A. (2024). Google – AI Assistants for Data Tasks with Gemma. Kaggle. <https://kaggle.com/competitions/data-assistants-with-gemma>
- [4] Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., ... & Chen, W. (2021). Lora: Low-rank adaptation of large language models. arXiv preprint arXiv:2106.09685. Retrieved from <https://arxiv.org/abs/2106.09685>
- [5] Uslu, C. (2022). "What is Kagge?". Datacamp Blog. Retrieved from <https://www.datacamp.com/blog/what-is-kaggle>
- [6] Kaggle (2024). Kaggle Platform for DS/ML Enthusiasts. Retrieved from <https://www.kaggle.com/>
- [7] Stackexchange API 2.3. (2024). Stackoverflow data by "Kaggle" keyword. Retrieved from <https://api.stackexchange.com>
- [8] Zhalgasbayev A. (2024). Full Kaggle QnA from Stackoverflow. Retrieved from <https://www.kaggle.com/datasets/armanzhalgasbayev/full-kaggle-qna-from-stackoverflow>
- [9] Austin, J., Odena, A., Nye, M., Bosma, M., Michalewski, H., Dohan, D., Jiang, E., Cai, C., Terry, M., Le, Q., & others. (2021). Program Synthesis with Large Language Models. arXiv preprint arXiv:2108.07732. Retrieved from <https://www.kaggle.com/datasets/mpwolke/mbppjsonl>
- [10] Nichil (Kaggle username) (2024). Data Science questions and answers. Kaggle Dataset. Retrieved from <https://www.kaggle.com/datasets/nvnikhil0001/data-science-questions-and-answers>
- [11] Sallum, E., Pereira, N., Alves, M., & Santos, M. (2020). Performance optimization on LoRa networks through assigning radio parameters. In 2020 IEEE International Conference on Industrial Technology (ICIT) (pp. 304-309). Buenos Aires, Argentina. Retrieved from <https://b2a.kz/aCA>
- [12] Llusi, R., El Yacoubi, S., Fontaine, A., & Lupera, P. (2021, October). Comparison between Adam, AdaMax and Adam W optimizers to implement a Weather Forecast based on Neural Networks for the Andean city of Quito. In 2021 IEEE Fifth Ecuador Technical Chapters Meeting (ETCM) (pp. 1-6). IEEE. Retrieved from <https://b2a.kz/C3L>
- [13] Chai, X., et al. (2022). An Open-Source Package for Deep-Learning-Based Seismic Facies Classification: Benchmarking Experiments on the SEG 2020 Open Data. IEEE Transactions on Geoscience and Remote Sensing, 60, 1-19. Retrieved from <https://b2a.kz/PjG>
- [14] Zhalgasbayev, A. (2024). Fine-tuning the Gemma model for Kaggle Assistant. Kaggle Notebooks. Retrieved from <https://www.kaggle.com/code/armanzhalgasbayev/fine-tuning-the-gemma-model-for-kaggle-assistant>
- [15] Chauhan, N. & Mooney, P. (2024). Fine-tune Gemma models in Keras using LoRA. Kaggle Notebooks. Retrieved from <https://www.kaggle.com/code/nilaychauhan/fine-tune-gemma-models-in-keras-using-lora>