

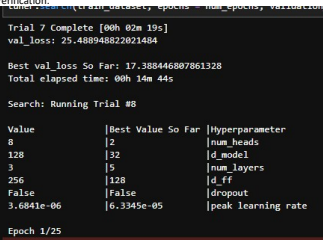
What did I do?

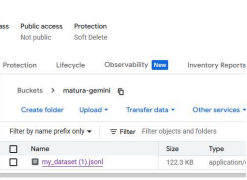
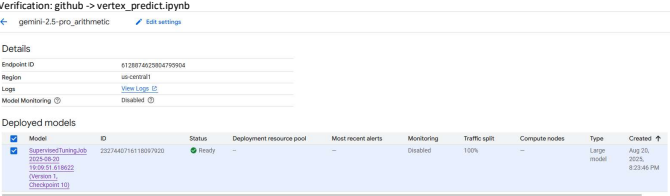
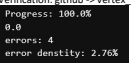

Saturday, 7 June 2025 12:51

Date	What did i do?	Time in hours
Tuesd ay, 8 April 2025	started systematic literature review, set up the overleaf document I'll be working on. Thinking about the main question of the review, I've got a general idea, but still unsure. I was able to narrow down the search on google scholar to 27 results with some keywords.	
Wedne sday, 9 April 2025	I've decided I'm going to be doing multiple searches with different keywords or aproaches(sorting by relevancy). I've picked 5 databases which I'll be searching with the keywords. THE NVIDIA GPU IS GONNA ARRIVE TOMORROW!!!	
Thursd ay, 10 April 2025	The package arrived! I'm still setting up the os system, but so far, so good. The Jetson orin Nano looks amazing.	
Friday, 11 April 2025	I've made a precise workflow for the research part of my lit. review. I also ran into some troubles with the WoS and scopus websites.	
Saturd ay, 12 April 2025	The Jetson orin nano setup is driving me crazy. When I first installed docker on it, I think I did something wrong, because a lot of errors came up. Neither I, nor the AI's I've asked have any clue on how to fix the issue. I think I might have to start over. On the other front: I decided on just using the 3 websites that did work and I picked out the 15 best sounding titles out of their top 25 results.	
Sunda y, 13 April 2025	after having looked through the titles of the different websites, only ACM and ScienceDirect had good papers. Altogether I found 18 promising titles, which I then cut down to 3 papers. Tomorrow I will read those. I had to deviate from the workflow due to the circumstances.	
Monda y, 14 April 2025		
Tuesd ay, 15 April 2025		
Wedne sday, 16 April 2025		
Thursd ay, 17 April 2025		
Friday, 18 April 2025		
Saturd ay, 19 April 2025		
Sunda y, 20 April 2025		
Monda y, 21 April 2025		
Tuesd ay, 22 April 2025		
Wedne sday, 23 April 2025		
Thursd ay, 24 April 2025		
Friday, 25 April 2025		
Saturd ay, 26 April 2025		
Sunda y, 27 April 2025		
Monda y, 28 April 2025		
Tuesd ay, 29 April 2025		
Wedne sday, 30 April 2025		
Thursd ay, 1 May 2025		
Friday, 2 May 2025	Ich habe heute angefangen das erste (auch längste) Papier zu gründlich lesen. Ich habe schon vorher einbisschen daran gelesen, aber nicht gründlich und ohne Notizen/Markierungen.	1.75
Saturd ay, 3 May 2025	weiter gelesen an der ersten Studie über XAI und dessen Kriterien.	2
Sunda y, 4 May 2025		
Monda y, 5 May 2025		
Tuesd ay, 6 May 2025		
Wedne sday, 7 May 2025		
Thursd ay, 8 May 2025	1. Studie weitergelesen und Orte welche ich gerne Zitieren würde markiert. Mit zweiter Studie über den Hawkes-Prozess angefangen. Problem: Ich bin mir nicht sicher ob ich den Hawkes prozess anwenden werde können. Obwohl sie es ein FNN nennen, lönt das Model welches sie in der Studie als Beispiel verwendet haben wie ein CNN.	1.5
Friday, 9 May 2025	1 Stunde die Studie über den Hawkes prozess probiert zu verstehen. Ich glaube aber, dass sie viel zu kompliziert ist. Ich werde einfach die Anwendung wahrscheinlich benutzen. Dannach habe ich an meinen Jetson gearbeitet und ich habe einen Setup guide benutzt, jetzt sollte mein Problem von früher wahrscheinlich behebt sein.	4
Saturd ay, 10 May 2025	Nvidia Jetson gearbeitet mit docker jetson-containers	4
Sunda	Jetson plus minus zum arbeiten gebracht. Ganzen tag sich damit beschäftigt lol.	12

y, 11 May 2025		
Monda y, 12 May 2025		
Tuesd ay, 13 May 2025		
Wedne sday, 14 May 2025		
Thursd ay, 15 May 2025		
Friday, 16 May 2025	Heute fertig mit Papiere lesen. Ich fange heute am abend/morgen mit dem Latex Dokument an.	1
Saturd ay, 17 May 2025	Erste 5 seiten des Lieraturberichtds geschrieben.	7
Sunda y, 18 May 2025	Section 3 geschrieben, aber nicht sehr hochwertig, es wird sehr wahrscheinlich bei den erklärungen von Konzepten viel korrigiert werden müssen	8
Monda y, 19 May 2025	wrote the first half of section 4. again I'm unsure in a lot of places, so there'll be a lot to correct I guess	4
Tuesd ay, 20 May 2025	didn't have a lot of time today, but I was able to finish section 4	2
Wedne sday, 21 May 2025	Turned out my questions were wrong, so I'm gonna have to redo everything, now. Gonna be quite the time crunch	5
Thursd ay, 22 May 2025	I found some good sources and formulated a vague structure of the new literature review.	2.5
Friday, 23 May 2025	I finished the introduction and first section of the second lit review.	5
Saturd ay, 24 May 2025	Wrote a page for section 3	2.5
Sunda y, 25 May 2025		
Monda y, 26 May 2025		
Tuesd ay, 27 May 2025	I finished section 3 and learned a lot actually. I somewhat have an Idea of what to write about in section 4 now.	2
Wedne sday, 28 May 2025	started section 4 but didn't really get anywhere. I need a bit more research. Also my latex document is falling apart. I should probably switch to biblatex. Also I CANNOT work today, I feel like the inner jobless me came out today. I like slept for about 10 - 11h so sleepy through out the day. I recorded, that I worked for only 3h, but in reality it was more like 6h and the rest of the day was procrastinating or distractions.	3
Thursd ay, 29 May 2025	finished the section 4 part about previously mentioned issues. Did a LOT of research	3.5
Friday, 30 May 2025	I enjoyed the good weather, didn't work tat much a part from reading and organizing all the literature. I wrote down the first approach in how to improve the model for arithmetic tasks. The goofy one with the virtual abacus thingy	3
Saturd ay, 31 May 2025	I was in the mountains today, so I didn't really do that much. Still I wrote down 4 aproaches in total now	2
Sunda y, 1 June 2025	FINISHED. Well not totally, but mainly. Dedicated the whole GODDAMN DAY to this review. I just wanna go and watch some insta reels. The takeaways section needs to be improved. But that's a problem OF THE FUTURE ME!!!	8
Monda y, 2 June 2025		
Tuesd ay, 3 June 2025	Finished up the grammatical correction of the lit. review. I also wrote an abstract and other stuff	3.5
Wedne sday, 4 June 2025	I've decided to use google colab, for the compability with the jupyter notebooks I set up for the nvidia jetson gpu earlier. I fiddled around a bit and got my first Prototype running.	5
Thursd ay, 5 June 2025	I added bootstrapping to the model. I drew a graph of expressions not in training data, showing the accuracy. This shows, that the Programm, indeed is learning something atleast. Need to do interpretation of it.	5
Friday, 6 June 2025	I made test data for numbers outside the training range. Predicted results for that test data with the bootstrapped model. I drew a outside of training range graph. Refined the previous tokenizer function.	3.5
Saturd ay, 7 June 2025	I implemented test data with more terms == longer expressions. Generalized the tokenizer function, so it works with longer expressions too. Collected bootstrapped predictions. And drew a graph showing number of terms vs deviation, with a trendline.	4
Sunda y, 8 June 2025	I evaluated the longer expressions - test data, and drew a graph for the correlation between length of expression and MAE. I also did a lot of debugging. I set about defining the a benchmark for models with different architectures, (the 3 different MSE proportionally to a base performance.) For this benchmark I had to implement pairs and stuff to reorganize the prediction results.	5.5
Monda y, 9 June 2025	Spent my day programming the generation of different models with different architectures. I think I succeeded, although I didn't really properly evaluate my results yet. The time it takes to train different models (36) is roughly an hour, this will make it diffcult for me to work in school in the near future. P.S. The padding with 0.5 worked way better, I implemented it.	7
Tuesd ay, 10 June 2025	At school I worked on my code, found some errors and changed them (neuron_steps), I couldn't squeeze in the time for a calculation today. I met up with my dad, he corrected my text for grammatical errors (though not all of I implemented), aswell as gave me an interesting article which I ended up citing additionally to prove a point in section 2. I asked Smits.	1
Wedne sday, 11 June 2025	I re-ran my calculations with an additional calculation of a benchmark for the models. I have a couple of good looking graphics for the presentation now. The cherry on top being the heatmap. At school I corrected my literature study and added some comments where needed. After some final polishing up, I'm ready to submit my results.	4
Thursd ay, 12 June 2025	I also set up a github page yesterday and restructured and re-organized it a little today (I'm still getting used to the browser-github. I previously only used it to download stuff from the internet through the terminal.) <a href="https://github.com/ArbiterStanton/maturay/tree/main">https://github.com/ArbiterStanton/maturay/tree/main</a> I also finished a rough Layout for the topics I want to discuss in the presentation. I feel like I'll speak more than 15 min Formulated a couple of questions to mr. Schneider.	2
Friday, 13 June 2025		
Saturd ay, 14	I started and finished about half of the power point. I didn't do any design yet.	2

June 2025		
Sunday, 15 June 2025		
Monday, 16 June 2025	A lil work during school on the power point.	0.5
Tuesday, 17 June 2025	Finished the Power Point. I don't have a layout though.	1.5
Wednesday, 18 June 2025	Polished up my power point and prepared for the presentation.	3
Thursday, 19 June 2025		
Friday, 20 June 2025		
Saturday, 21 June 2025	During the last couple of days I thought about the proposal made by mrs. Smits and I think she might be right. After having computed it, I'm left with a lot of interesting data to unpack. Today, I drew a couple of graphs and let her know.	2
Sunday, 22 June 2025		
Monday, 23 June 2025		
Tuesday, 24 June 2025	I implemented the suggestions provided by J.Smits and drew some graphs in order to decide wether to change my benchmark. Specifically, I will now be using seaborn for better graphs with confidence intervals.	3
Wednesday, 25 June 2025	I finalized the graphs with relative errors. And tried a new benchmark with relative errors. The results looked more promising. I also made some new entries on GitHub. I talked with mr. Schneider and stated 2 Questions.	
Thursday, 26 June 2025		
Friday, 27 June 2025	Today I explored introducing Drop-Out: It showed higher MSEs: 0.33413893 13.065571 18.018482 From <a href="https://colab.research.google.com/drive/15ek7rKQu7h8A-vx5GGL8uXua0uNCv0aPwzcITgc-f1B43ahny">https://colab.research.google.com/drive/15ek7rKQu7h8A-vx5GGL8uXua0uNCv0aPwzcITgc-f1B43ahny</a> Aswell as a heatmap with lower benchmarks, when comparing to values of a basemodel without dropout (0.3) I also started the documentation file.	1
Saturday, 28 June 2025		
Sunday, 29 June 2025	Today I set off improving the tokenizer by adding a positional encoding. I managed to code a working sinusoidal positional encoder. It showed weaker results then, without a positional encoder, which is very weird. I set off debugging and didn't find any issues, still it's difficult to believe.	6
Monday, 30 June 2025	I found the problem, there was an error when combining two arrays. I then re-drew some graphs and observed the positional encoder improving the performance, after changing the algorithm a little. I tried reducing over-fitting with L2 regularization, it didn't help and made the models perform worse.	6
Tuesday, 1 July 2025	I confirmed my suspicion that the L2 regression had the same negative effect as L1 and Drop-Out. I decided against trying different batch sizes, because in theory this shouldn't have any impact.	0.5
Wednesday, 2 July 2025		
Thursday, 3 July 2025		
Friday, 4 July 2025		
Saturday, 5 July 2025		
Sunday, 6 July 2025		
Monday, 7 July 2025		
Tuesday, 8 July 2025	I properly downloaded and setup the jupyterlab environnement locally and trained a first prototype of the recurrent Neural Networks.	3.5
Wednesday, 9 July 2025	I did some research into the functionality of LSTM and GRU RNNs. I drew the graphs just like I did earlier with the FNN. I also made some minor improvements in the code. I couldn't properly work today, because I was a little sick. (and still am)	3
Thursday, 10 July 2025	Looked over how to import variables from other .py scripts. With testing.	1
Friday, 11 July 2025	Implemented the generation of different RNN models with different amounts of neurons and layers. The Data is later also converted into usable np.array. I got as far as to drawing the heatmap, but there is a problem with the array I receive after the long calculations of the many models. I get 30 models, when really I should be expecting 28.	4
Saturday, 12 July 2025	Fixed the error. The first heatmap of RNNs has been generated. Surprisingly the SimpleRNN doesn't improve performance of arithmetical tasks.	2.5
Sunday, 13 July 2025	I generated the Heatmaps of LSTMs and researched about attentional LSTMs	3.5
Monday, 14 July 2025	I generated GRUs Heatmap overnight and later researched about how I can add attention to my LSTMs. Watched a bunch of tutorial videos on youtube and read up on some stuff too. Only research about the next attention with LSTM model.	4
Tuesday, 15 July 2025		
Wednesday, 16 July 2025	Research and first prototype. I copied and slightly modified the code from a website. (geeksforgeeks). I also researched a lot today. The odd thing is that the results of the LSTM with the attention seem to be worse actually.	4
Thursday, 17 July 2025	I reorganized and drew the heatmap, but I don't know how to move on. The heatmap was worse, as expected	
Friday, 18 July 2025	I fiddled around with the code of the attention layer, but no progress. In the evening I just wrote a lilbit about the definition of the RNN and the formulas behind it and stuff.	2.5
Saturday, 19 July 2025		

ay, 19 July 2025		
Sunday, 20 July 2025		
Monday, 21 July 2025		
Tuesday, 22 July 2025	I'm writing this at 2am so technically it's also what I did tomorrow. After watching even more tutorials and explanations online, as well as chatting with Gemini about it, I rewrote the LSTM with attention, since last time I wrote the code, I copied most of it from a website, and I wasn't able to properly understand it. Today I fully understand "everything" about this new model. Yet, unsurprisingly it showed the same scores as the ones I got from the website.	4
Wednesday, 23 July 2025	I reorganized everything and finally found a better workflow for github lol. I also discovered, that I wasn't using the validation data I defined earlier. I fixed that too. I tried out jules. Its really cool. I also did a bit of research on transformers, I'm ready to begin tomorrow.	3.5
Thursday, 24 July 2025	I did some more research on transformers, had a couple of conversations with gemini, as well as generated the first notebook, which should serve as a sort of introductory transformer model, based on the one from the first paper "attention is all you need". (I also wrote a couple of cells including the positional encoder. (the one the paper used with sin and cos.)) This is just for better understanding on my part, since it worked fairly well with RNNs. I also watched tutorials for class type objects in python, ill need that for transformers I'm sure.	2.5
Friday, 25 July 2025	I watched tutorials during the day, and then worked a bit more on the transformer at night roughly 10 - 4am. I found a very confusing error "reinvented" embedding for my project (more on that in reflection). I finished up the embedding, the positional encoding and the scaled dot product attention function inside the multithread attention class.	5
Saturday, 26 July 2025	I FINISHED the transformer (sunglasses emoji). I did A LOT today and I'll skip over a lot. In summary I coded a many to one transformer, modelled after the seq2seq one in the "attention is all you need" paper. Effectively, I just used the Encoder only. Another thing, is that I coded the Multi head attention basically from scratch, which really helped me understand the workings of a transformer. 2 big problems came up when I first fitted the model: - 1st: It takes about 100x as long to process one epoch of a transformer, then of an attentional RNN. And if you remember, it took my cpu 5.5h of computations to draw the heatmap for the attentional-RNN consisting of 330 model. - 2nd: The transformer isn't learning. It's just guessing around 0 and satisfies itself with that.	8
Sunday, 27 July 2025	Tried to install tensorflow, but to no avail.	3
Monday, 28 July 2025	Still no success	2
Tuesday, 29 July 2025	Finally was able to install tensorflow with gpu support. Kinda destroyed my jetpack in the process tho. I had some issues with firefox just not opening up. Turns out I had to install an older version of snspd to solve the issue. Afterward I was able to setup the jupyterlab and run my scripts on the jetson nano. This turned out to have 17-times the efficiency.	5
Wednesday, 30 July 2025	I looked into the issue with encoders/embeddings for previous notebooks. I experimented around a little, but like before the exact same positional encoding, which is used in the transformer(cos/sin) doesn't seem to help, the results are worse even. And the traditional embedding/encoding part isn't necessary, since I am working with number (and only a select few symbols). Basically after some investigation I came to the conclusion to leave it as is, because 1st it should theoretically be enough of an encoder as is, and 2nd because my work aimed to focus on the architecture anyways.	2
Thursday, 31 July 2025		
Friday, 1 August 2025		
Saturday, 2 August 2025		
Sunday, 3 August 2025		
Monday, 4 August 2025		
Tuesday, 5 August 2025		
##### ##### ##### #		
Thursday, 7 August 2025		
Friday, 8 August 2025		
Saturday, 9 August 2025	Final 2 Days before school: The guests left yesterday, over the course of the last week I did some work here and there on hotel Wlan, (I setup jupyterlab weird, and it requires wifi.) I implemented these features into the transformer model, as advised by websites and AIs: - Learning Rate Scheduler with linear warmup and cosine decay.(biggest change I'd say.) - AdamW optimizer - Experimented with different batchsizes - Gradient clipping (learning rate scheduler behaves a little goofy at times) - shuffle in the training data (this is so basic, I think I could/should add this to all the previous models aswell)	5
Sunday, 10 August 2025	Today I had to reinstall the correct tensorflow again because it updated, but it was much easier since I have the wheel I used the last time. Later I tackled the 0-guessing problem again: I had great success when decreasing the model complexity, this, I think, is the correct solution to the problem. (either that or increasing the dataset size.) I also tweaked and played around with the optimizer's parameters. With all that, as well as some minor adjustments, I was able to reach new mse lows.	4
Monday, 11 August 2025		
Tuesday, 12 August 2025	I played around with the variables some more, then a LLM suggested I use keras-tuner, and I did. I tried to let the program do calculations over night, but it shut itself off. -> I wrote transformer2.ipynb Was only able to run 3 trials.	4.5
##### ##### ##### #	Since I worked from school today (without access to the GPU), I decided to implement the keras-tuner for the FNNs, to test it's effectivity. I found it performed quite poorly. After 20mins of training the best model it found consisted of 1 layer and 15 neurons. That layer was very noisy and was slightly overfitting. A weird thing happened where the models with small amounts of parameters would start with very small MSEs. Edit: I watched the videos provided by my teacher on how to plan and protocol correctly.	2.5
Thursday, 14 August 2025	Today, I firstly made a plan for the remaining time and submitted it. Afterwards I worked on investigating the weird behaviour of the smaller FNNs. It turned out to be due to the glorot_unifiler weights initialization. This is weird because earlier I was encountering a similar error with the transformers (transformer0). And I confirmed that when weights are initialized at 0, this is the very lowest point of the local minimum.  Verification: github -> transformer3.ipynb and submitted MaturaPlan.png  Later I came home and tried to use my GPU to train a couple of models with keras-tuner, but the gpu keeps crashing. This is a problem. Verification:  Trial 7 is the farthest I was able to get.	3.5
Friday, 15	Today I decided to train the keras-tuner with my laptop instead because it has more RAM, and sure enough after an hour of calculations very promising hyperparameters were discovered.	6

August 2025	Later on I evaluated the benchmark of the model with the newly discovered hyperparameters. I achieved a benchmark score of 57. This is massive and by far the best model I've trained yet. The transformer itself has a pretty simple architecture with only 38k trainable parameters. After this small evaluation, the transformer seems to be performing better than other frontrunners in the generalization categories. Later on I found a logical error. The 57 number isn't reliable. More on that tomorrow... I hope Verification: github -> transformer3 and transformer4	
Saturday, 16 August 2025	Debugging. Today I was able to fix the problem. There was an issue with early_stopping, as well as with the subtraction of arrays. The benchmark was calculated correctly now. The model achieved an impressive <b>974691182974788</b> From <a href="https://colab.research.google.com/github/0194-8888/uh/tree/transformer/transformer4.ipynb">https://colab.research.google.com/github/0194-8888/uh/tree/transformer/transformer4.ipynb</a> points. I then also thought about the benchmark calculation, which raised a few doubts in my head. Verification: github -> transformer4.ipynb, FNN1_1.py <pre>(testset_pred, predsInRange(2))() print(x_test(0)) print(y_test(0)) print(y_test(1), predsInRange(0))  print(meanDiff_InRange, meanDiff_OutRange, meanDiff_LongRange, meanDiff_OutRelRange) print(baseline_deviation, baseline_out_deviation, baseline_long_deviation, baseline_relierror)  [6.94753807] [15], 0, 1, 0, 1, 1, 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 7.8 [3.884227] 0.5083455 0.487058 5.857648 8.957115983 0.48647725311281620 10.37766881935294 13.364785215218881 0.82814327266657898</pre> Oh, I also the FNN1 script with the help of an AI. I think shortening is allowed right? (-> FNN1_1.py) Here's the exact prompt I used in gemini 2.5 flash: shorten this script, only leave all the necessary calculations for: baseline_deviation, baeline_out_deviation, baseline_long_deviation, baseline_reliError, absSum	4
Sunday, 17 August 2025	This concludes my work on the transformer-workload. Today I firstly found a grasp in reality and understood the benchmark formula again. It makes sense. Afterwards I reran the keras_tuner on my laptop again. Then I had to configure my jetson to house more swap memory and trained the best model found by the tuner. It turned out to be much more complex than the other transformer in transformer4.ipynb (by a factor of 10). After evaluating the benchmark of this new transformer, there seems to be a correlation between model complexity and the models ability to generalize. But more on that in the evaluation part later. Verification: github -> transformer5.ipynb <pre>(testset_pred, predsInRange(2))() print(x_test(0)) print(y_test(0)) print(y_test(1), predsInRange(0))  print(baseline_deviation, baseline_out_deviation, baseline_long_deviation, baseline_relierror) print(meanDiff_InRange, meanDiff_OutRange, meanDiff_LongRange, meanDiff_OutRelRange)  [1.888994] [15], 0, 1, 0, 1, 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 5.1 [3.884227] 0.48647725311281620 10.37766881935294 13.364785215218881 0.82814327266657898</pre>	5
Monday, 18 August 2025	I slept terrible yesterday, so I wasn't able to work a lot / with high quality today Started the pre-trained model workload Today I was doing a lot of research. I had to study up, how to fine-tune a pre-trained LLM model for my specific task. After a while I kinda got somewhere with the gemini 2.5 pro model. I wasn't able to specify the training data. This is a big problem as I am unfamiliar with the google.genai library and there isn't a lot of documentation online. I also made an oopsie, but more on that in the Reflection section. Verification: github -> gemini2.5.ipynb <pre>In [7]: training_dataset = TuningDataset(     gcs_uri="file_name" ) print(training_dataset)  gcs_uri="file_name" examples=None</pre> ^^The error is here somewhere I think I'm not doing this the right way. PS I also encountered issues with the libraries in general not importing. [1]: client = genai.Client(http_options=HttpOptions(api_version="v1"),     api_key = "AIzaSyDcud5xv4FQ7zbK05wmKkGdf_hUcsf9e0",     )  ^^the oopsie	3.5
Tuesday, 19 August 2025	I kept on trying to get everything to work in collaboration with google cloud. Today, I made an account with a free trial, uploaded the dataset to a bucket. Kinda succeeded, the model had already started. Yesterday I was working with genai, today I tried it with vertexai, which seemed to work better (less errors, simpler for me to understand). Still a lot of research, if you want to call it that, more just looking through tutorials. Verification: github -> gemini_vertexAI.ipynb 	3
#####	I tried around in the web ui. I managed to deploy a fine-tuned model and deployed an endpoint. I don't get how to use it for predictions. Very lost at the moment. Surely tomorrow will be better. ##### ##### # Verification: github -> vertex_predict.ipynb 	2.5
Thursday, 21 August 2025	After using a different command to access the previously deployed endpoint it worked all of a sudden, even though I got a deprecated warning, I chose to ignore it. I then trained some new models and later adjusted the vertex_predict.ipynb to actually work as intended. I later predicted all the different test data, and to my surprise the model seemed to perform even better than my other models on generalization tasks. This is expected as the model already has been trained on similar data, during the general purpose logic training. A big difference was that the model, in contrast to the mathematical nature of the previous models, worked in a more categorical way, rather than a regression task. It only predicted integers, which meant it never got a thing wrong, and when it wasn't sure, it just didn't give off a guess. The error rates were analyzed and they were manageable. Verification: github -> vertex_predict.ipynb 	
Friday, 22 August 2025		
Saturday, 23 August 2025		
Sunday, 24 August 2025		
Monday, 25 August 2025		
Tuesday, 26 August 2025	Today I worked a little bit. I tried to get started on tuning a second model. This time I want to train a smaller model. I decided the gemma3 or gemma3n model should be perfect for this, because of their relatively small size, and good trainability - As far as I understand the models parameters can learn new patterns/tasks quicker/better than the ones of the big models. I found a good blog and library to do this in. I tried to use vertexai again but it sucks. Verification: Blog: <a href="https://unsloth.ai/blog/gemma3">https://unsloth.ai/blog/gemma3</a> github -> gemma_vertexAI.ipynb 	2
#####	Today, I copied the notbook of the unsloth gemini3 blog and adjusted it to my needs.	1.5

#####	Verification: github -> gemma_unsloth.ipynb	
#####		
#####		
Thursd ay, 28 August 2025	I ran the code on the nvidia gpu. It didn't work. I spent the whole time navigating around in the Jetson environment. It was very exhausting and there isn't a lot to say. Firstly I had to install the correct pytorch version. Keeping in mind all the dependencies. I also had to upgrade my jetpack version. And then install a unsloth version that worked.  Verification: The tutorial that worked for the pytorch installation. So that it will work with my gpu: <a href="https://docs.pytorch.org/TensorRT/getting_started/jetpack.html">https://docs.pytorch.org/TensorRT/getting_started/jetpack.html</a>	5
Friday, 29 August 2025		
Saturd ay, 30 August 2025		
Sunda y, 31 August 2025	Finally found some time to work some more. I was able to resolve the dependencies and built up triton from its source. After some small adjustments to the code itself (training dataset import) I encountered another Error, the biggest one I've ever seen. I think it's a problem with the pytorch compiler. I tried deactivating the compiler, it didn't work. So I'm stuck again basically.  P.S. I worked for 2 more hours, and I've managed to setup remote access to my jetson. I used ssh tunnel with an -L flag. This allows me to open jupyter lab in my laptops browser from a different network, this means I'll be able to work in school tomorrow. Verification: github -> gemma_unsloth.ipynb	7

Monday, 1 September 2025	I tackled the problem from school, and later again from home, I asked for help online, I asked my teachers, my mother and even the official discord. There they told me that the ARM architecture my Jetson is based on isn't supported. This means I can forget using unsloth, I'll switch to Huggingface with accelerate. <b>BackendCompilerFailed</b> 	4																																		
Tuesday, 2 September 2025	Verification: github -> gemma_unsloth.ipynb  Started work for the new approach - using HuggingFace. I spent most of the time researching the best methods available to do this in my architecture. I encountered some issues. I started setting up the venv with all of it's dependencies. Most of them should be installed correctly and with GPU compatibility.  Useful guide to provide an overview of how the finetuning process should look like ideally: <a href="https://medium.com/data-science-in-your-pocket/a-practical-guide-to-fine-tuning-google-gemma-2-720m-with-lora-ea02dec72ac4">https://medium.com/data-science-in-your-pocket/a-practical-guide-to-fine-tuning-google-gemma-2-720m-with-lora-ea02dec72ac4</a> Verification: github -> gemma_huggingface.ipynb	2																																		
Wednesday, 3 September 2025																																				
Thursday, 4 September 2025	Today I started working, got a head ache AND FINALLY DECIDED TO START USING JETSON-CONTAINERS. Best decision of my life. It really helps, and once you get it, it makes your life so much simpler. As it turns out Nvidia Jetson devices not only have a Aarch64 architecture, its like some super special optimized version, meaning it's super exclusive and you'll have plenty of dependency problems if you work normal, like on a x86-64 device. Basically there are internal docker containers available for different popular python packages on github: <a href="https://github.com/dusty-nv/jetson-containers">https://github.com/dusty-nv/jetson-containers</a> I was able to build on top of their containers a container to fit my needs. All of this obviously took a load of time since I'm a noob at docker and stuff, but now I slowly get it. I generated (with AI) a small script to check whether the bitsandbytes library is working properly. This test has been passed by the container. There was an additional problem where the datasets library I needed for formatting my training data wasn't available on the github. I was able to improvise with a docker commit. (sounds easy, but took me like 1.5h) Conclusion: ALL REQUIREMENTS (atleast most of them, im not entirely sure if that's all of them) HAVE BEEN INSTALLED AND THE PROGRAMM IS READY "TO BE CODED ON". <pre>from transformers import LlamaForCausalLM, LlamaTokenizer import torch import bitsandbytes asbnb  --- Bitsandbytes GPU Verification Script ---  Step 1: Checking for CUDA-enabled GPU... [Success] Success: CUDA is available. Found GPU: Orin  Step 2: Loading a model with 8-bit quantization ("load_in_8bit=True")... The "load_in_4bit" and "load_in_8bit" arguments are deprecated and will be removed in the future versions. Please use the "bnb_4bit" or "bnb_8bit" arguments instead. [Success] Success: Model loaded in 8-bit without errors. This indicates that bitsandbytes is correctly installed and communicating with the GPU.  Step 3: Verifying model properties... - Model is on device: cuda:0 [Success] Model is correctly placed on the CUDA device. - Model memory footprint: 165.54 MB  Step 4: Performing a simple inference test (forward pass)... [Success] Success: Forward pass completed without errors.  --- Verification Complete --- [Success] All checks passed! Your "bitsandbytes" installation appears to be working correctly with your GPU.</pre> Verification: github -> gemma_huggingface.ipynb, -> verifyHuggingFacePkgs.py	6																																		
Friday, 5 September 2025	I was able to fine-tune gemma3, just like I intended. Amazing! Ran into some problems: - Installing new libraries in a container (temporary solution) - Github and huggingface authentications - Correct hyperparameters and config of the Trainer - System crashed while I was ssh-ed into it. I wasn't able to recover the system and had to stop working. - High training Loss (low learning rate) All in all my task for today is completed. Verification: github -> gemma_huggingface.ipynb	7																																		
Saturday, 6 September 2025																																				
Sunday, 7 September 2025																																				
Monday, 8 September 2025																																				
Tuesday, 9 September 2025	I fixed the github authentication problem, I found out how to evaluate the model and I tweaked the hyperparameters (only the learning rate scheduler) for better results, yet I think that it isn't meant to be. The model just doesn't wanna train beyond 0.4 training loss, despite of the different learning rates I tried. I also evaluated the Error on validation data, he was much bigger than expected, the model is overfitting. <pre>print(np.mean(diffs))</pre> 1.240986 Verification: github -> gemma_huggingface_predict.ipynb	5																																		
Wednesday, 10 September 2025	Played around some more with different parameters. I defined some eval data, and tried to implement it into the model, but it didn't work for some reason. This also leads to crashes. Exact reason unclear. (probably RAM usage) but why? SFTTrainer is weird, but I have to use it because of the QLoRA. The teacher also let me know to mention that I was "considering" reproducibility of my project. Thankfully he said there are no extra points for that, that would've been a problem for me if there were. Verification: github -> gemma_huggingface.ipynb	3																																		
Thursday, 11 September 2025	I was finally able to add validation data properly. There now is also a computation of the accuracy of the model, which is the metric I used with the previous model (gemini 2.5). I did this using the compute_metrics function, which I initially generated with a new LLM I found (for coding) (MistralAI). There were a lot of errors, which I had to debug step by step. That sucked. Another problem was the amount of time it took. I just wanted to finish this up, so I could work remotely from school again tomorrow, since for this I need to sometimes restart the container because of memory overload. I should probably also mention that training this model takes 80min, compared to the previous one (without eval_data). Probably because of the inefficient compute_metrics function I used. <table><tr><td></td><td></td><td></td><td></td><td></td><td>1691/954 57:18 &lt; 21:52, 0.20 it/s. Epoch 2.17/3]</td></tr><tr><th>Step</th><th>Training Loss</th><th>Validation Loss</th><th>Accuracy</th><th>Entropy</th><th>Num Tokens</th><th>Mean Token Accuracy</th></tr><tr><td>10</td><td>9.809200</td><td>6.865591</td><td>0.000000</td><td>1.418712</td><td>1763.000000</td><td>0.359184</td></tr><tr><td>20</td><td>4.320000</td><td>2.812313</td><td>0.000000</td><td>2.478687</td><td>3520.000000</td><td>0.520952</td></tr><tr><td>30</td><td>2.039000</td><td>1.491469</td><td>0.000000</td><td>1.725600</td><td>5285.000000</td><td>0.638003</td></tr></table>						1691/954 57:18 < 21:52, 0.20 it/s. Epoch 2.17/3]	Step	Training Loss	Validation Loss	Accuracy	Entropy	Num Tokens	Mean Token Accuracy	10	9.809200	6.865591	0.000000	1.418712	1763.000000	0.359184	20	4.320000	2.812313	0.000000	2.478687	3520.000000	0.520952	30	2.039000	1.491469	0.000000	1.725600	5285.000000	0.638003	6
					1691/954 57:18 < 21:52, 0.20 it/s. Epoch 2.17/3]																															
Step	Training Loss	Validation Loss	Accuracy	Entropy	Num Tokens	Mean Token Accuracy																														
10	9.809200	6.865591	0.000000	1.418712	1763.000000	0.359184																														
20	4.320000	2.812313	0.000000	2.478687	3520.000000	0.520952																														
30	2.039000	1.491469	0.000000	1.725600	5285.000000	0.638003																														

Hypothesis for reason of the error:

- There is a previous model that is being loaded, instead of the most recent one. (or not the most recent epoch)  
All the previous runs have been deleted, there is only one left. The specific checkpoint has been specified (the most recent one).
- There is an error with generating the data, like using different hyperparameters (temperature and stuff.)  
The pytorch.generate() and the huggingface provided generator pipeline both yielded similar results. And the pipeline uses hyperparameters from the fine-tune.
- There might be an issue with the calculation during the evaluation steps  
I thoroughly checked the compute\_metrics function, and made sure the validation dataset is being used and not the train dataset. The function itself is very basic and there is no error as confirmed by multiple LLMs.

	<table><thead><tr><th>Step</th><th>Training Loss</th><th>Validation Loss</th><th>Accuracy</th><th>Entropy</th><th>Num Tokens</th><th>Mean Token Accuracy</th></tr></thead><tbody><tr><td>10</td><td>9.809200</td><td>6.865591</td><td>0.000000</td><td>1.418712</td><td>1763.000000</td><td>0.359184</td></tr><tr><td>20</td><td>4.320000</td><td>2.812313</td><td>0.000000</td><td>2.478687</td><td>3520.000000</td><td>0.520952</td></tr><tr><td>30</td><td>2.039000</td><td>1.491469</td><td>0.000000</td><td>1.725600</td><td>5285.000000</td><td>0.638003</td></tr><tr><td>40</td><td>1.216000</td><td>1.120224</td><td>0.160000</td><td>1.125232</td><td>7039.000000</td><td>0.678528</td></tr><tr><td>50</td><td>0.974200</td><td>0.948166</td><td>0.130000</td><td>0.976329</td><td>8806.000000</td><td>0.694357</td></tr><tr><td>60</td><td>0.808300</td><td>0.742455</td><td>0.240000</td><td>0.889345</td><td>10566.000000</td><td>0.748972</td></tr></tbody></table> <p>Verification: github: -&gt; <a href="#">gemma_huggingface.ipynb</a>, <a href="#">https://huggingface.co/Anton800M/output</a></p>	Step	Training Loss	Validation Loss	Accuracy	Entropy	Num Tokens	Mean Token Accuracy	10	9.809200	6.865591	0.000000	1.418712	1763.000000	0.359184	20	4.320000	2.812313	0.000000	2.478687	3520.000000	0.520952	30	2.039000	1.491469	0.000000	1.725600	5285.000000	0.638003	40	1.216000	1.120224	0.160000	1.125232	7039.000000	0.678528	50	0.974200	0.948166	0.130000	0.976329	8806.000000	0.694357	60	0.808300	0.742455	0.240000	0.889345	10566.000000	0.748972		
Step	Training Loss	Validation Loss	Accuracy	Entropy	Num Tokens	Mean Token Accuracy																																														
10	9.809200	6.865591	0.000000	1.418712	1763.000000	0.359184																																														
20	4.320000	2.812313	0.000000	2.478687	3520.000000	0.520952																																														
30	2.039000	1.491469	0.000000	1.725600	5285.000000	0.638003																																														
40	1.216000	1.120224	0.160000	1.125232	7039.000000	0.678528																																														
50	0.974200	0.948166	0.130000	0.976329	8806.000000	0.694357																																														
60	0.808300	0.742455	0.240000	0.889345	10566.000000	0.748972																																														
Friday, 12 Septe mber 2025																																																				
Saturd ay, 13 Septe mber 2025																																																				
Sunda y, 14 Septe mber 2025																																																				
Monda y, 15 Septe mber 2025																																																				
Tuesd ay, 16 Septe mber 2025	<p>After trying to evaluate my model on the data used before for gemini 2.5 I encountered a weird logical inconsistency where the accuracy calculated during the training process was much higher than if I run a test on the exact same data after the fine-tuning process(now). This was very weird and I was suspicious of a generation error on my part, but I couldn't find one.</p> <p>I tried generating using the pytorch lower-end function, aswell as the pipeline put forth on huggingface tutorials, both yielded similar results. After finetuning Gemma3 1b (a very similar, but larger SLM) I encountered a similar problem when evaluating. This time there was less of a difference between the two, which leads me to believe, that this wasn't an Error on my part after all.</p> <p>0.63</p> <p>Despite the expected accuracy of 1.00 I get something like this (0.63). This is a pretty big difference</p> <p>Verification github -&gt; matura/pretrained/gemma-huggingface-predict, big-gemma-huggingface</p> <p>(This is an entry is an accumulation of multiple, from the past couple of days)</p>	7																																																		
Wedne sday, 17 Septe mber 2025	<p>Finished up the final touches for this workload. I finished up predictions with gemma3 1B. Everything with the gemma3 1B was actually done today, I just wrote the entry for yesterday today, that's why I mentioned it.</p> <p>The results are pretty good, there are reasons behind them, everything is good.</p> <p>Verification -&gt; matura/pre-trained-transformers</p>	3																																																		
Thursd ay, 18 Septe mber 2025	<p>I was just organizing today, and I tried to debug the before mentioned problem where the models, after being finetuned, showed many times worse performance than in during training for whatever reason. I did the thing on the right as told by mr. Schneider. I can't think of any other Hypothesis for the error. AI can't help me here. I don't know what to do I'm pretty much stuck here. (I'll move on in a couple of days if the logical inaccuracy persists.)</p> <pre># The base model ID, as indicated by the error message base_model_id = "google/gemma-3-270m-it"  # Your fine-tuned adapter adapter_id = "./output"  # 1. Load the base model model = AutoModelForCausalLM.from_pretrained(     base_model_id,     dtype=torch.bfloat16,     device_map="auto",     trust_remote_code=True )  # 2. Load the Peft model and merge it with the base model model = PeftModel.from_pretrained(model, adapter_id) model = model.merge_and_unload()  # The 'model' variable now holds your fully fine-tuned model</pre> <p>Verification: The table to the right</p>	5																																																		
Friday, 19 Septe mber 2025	Only a little bit of adjusting	1																																																		
Saturd ay, 20 Septe mber 2025	Today I fixed a little error and reran the training of the gemma1b model. It wasn't a lot of work really, nothing to write home about.	1.5																																																		
Sunda y, 21 Septe mber 2025																																																				
Monda y, 22 Septe mber 2025																																																				
Tuesd ay, 23 Septe mber 2025																																																				
Wedne sday, 24 Septe mber 2025																																																				
Thursd ay, 25 Septe mber 2025																																																				
Friday, 26 Septe mber 2025																																																				
Saturd ay, 27 Septe mber 2025																																																				
Sunda y, 28 Septe mber 2025																																																				
Monda y, 29 Septe mber 2025																																																				
Tuesd ay, 30 Septe mber 2025	<p>Today I finally started working on the project again, I've been quite busy in the past few days.</p> <p>I revisited the issue of the pre-trained models giving different accuracies, then expected.</p> <p>I used one of the newest ai's for help with my problem and it suggested to add the line model.eval() line to ensure the model is in evaluation mode. I implemented this aswell as some other few adjustments. But this didn't help much.</p> <p>Afterwards I had all the evaluation data needed for pre-trained fine-tuned models. I wrote it down in an excel file. (I also started this excel file a couple of days earlier)</p> <p>Verification: <a href="#">https://kb42puyvdrk5w.ck.kimi.link</a> and github</p>	4																																																		
Wedne sday, 1 Octobe	Today I sorted out the github issue and finally committed my thingy. This took long because I'm not very familiar with github. I also got a bill from google today for 83 CHF.	2																																																		

All the previous runs have been deleted, there is only one left. The specific checkpoint has been specified (the most recent one).

- There is an error with generating the data, like using different hyperparameters (temperature and stuff.) The pytorch.generate() and the huggingface provided generator pipeline both yielded similar results. And the pipeline uses hyperparameters from the fine-tune.
- There might be an issue with the calculation during the evaluation steps
- I thoroughly checked the compute\_metrics function, and made sure the validation dataset is being used and not the train dataset. The function itself is very basic and there is no error, as confirmed by multiple LLMs.
- The model is overfitted

No, predicting on train data yielded the same (slightly better) results

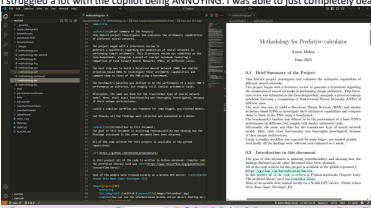
After trying out different checkpoints, all of a sudden the accuracy of the 270m model dropped from 40% to 15%

- This lead me to believe maybe it has something to do with the way the models are calculating everything, apart from the temperature and other hyperparameters, it might be because of performance optimization software calculation on GPU or GPU with work offload to CPU might give different results. Transformers is imported the same way, and is working on GPU exclusively both times.
- Ran the code on the exact same dataset used in the validation part during training, no differences.
- There is a slight difference with the calculation. In the compute metrics function the prediction is only counted if the model uses a turnseparator. This could explain why the seemingly normal gap in accuracies occurred.

After rerunning the training with the correction, the accuracy dropped a little (from 100% to 97%) at the end.

- The pre-trained model obviously has a drop out and layer norms. The issue might be that the model is not in eval mode and is dropping out a lot of valuable data, this would explain the slightly worse performance. This was not the case, because after setting model.eval() this only increased the accuracy by around half a percent for the Gemma3 270M model, nothing really changed. + This should probably be on by default anyways.
- A perfectly reasonable explanation would be, the model's weights aren't loaded in to the model properly, like it happened with FNN2.ipynb "see 4th of Oct Arbeitsprozess.

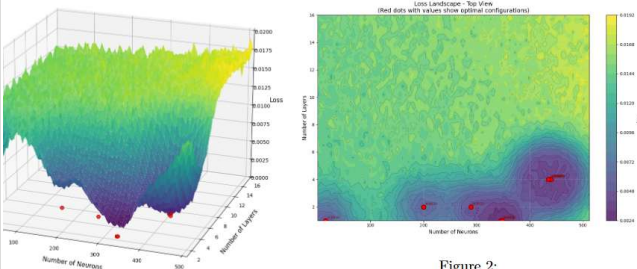
This cannot be the case because loading in the hyperparameters is completely different to loading in the weights. A model with loaded-in weights is already trained (hence the name - pre-trained model). Additionally the model is showing signs of having learned from the fine-tuning process for example in the way it responds with the trained int.0 instead of the traditional int.

2025 Thurs ay, 2 October r 2025	<p>I started working with extracting the infos from the transformers notebooks. I will be using the 2 I got by using keras tuner to optimize hyperparameters. I quickly realized that using bootstraps is basically like cheating. And not fair, because I sometimes used them, sometimes I didn't: very sketchy. I decided to define: NO bootstraps. I just shut down the billing account on google. It cost me way too much money.</p> <p>I filled the excel table with the evaluation data from the both transformers as well as RNN0.</p> <table><tr><th></th><th>RNN0:</th><th>transformer0:</th><th>transformer1:</th><th>transformer4:</th></tr><tr><td>Total Parameters:</td><td>20505</td><td>1494724</td><td>1494724</td><td>114628</td></tr><tr><td>Architecture Parameters:</td><td>8901</td><td>490241</td><td>490241</td><td>36209</td></tr><tr><td>Optimizer Parameters:</td><td>1704</td><td>596453</td><td>596453</td><td>76419</td></tr><tr><td>MAE in Range:</td><td>0.2937096</td><td>0.0301983</td><td>0.0301983</td><td>0.0448075</td></tr><tr><td>MRE in Range:</td><td>0.1297235</td><td>0.0151836</td><td>0.0151836</td><td>0.0188847</td></tr><tr><td>MRE out Range:</td><td>3.7189743</td><td>4.3057064</td><td>4.3057064</td><td>4.3702314</td></tr><tr><td>MRE out Range:</td><td>0.3856639</td><td>0.5431157</td><td>0.5431157</td><td>0.5201449</td></tr><tr><td>MAE long Expressions:</td><td>3.7682796</td><td>6.0603277</td><td>6.0603277</td><td>6.1632099</td></tr><tr><td>Benchmark score:</td><td>0.4446330</td><td>5.2696369</td><td>5.2696369</td><td>2.8852827</td></tr></table> <p>I</p> <p>Pretrained fine-tuned Models:</p> <table><tr><th></th><th>Gamma 2.2 Pro</th><th>Gamma 3.1B</th><th>Gamma 3.279H</th></tr><tr><td>Accuracy in Range:</td><td>95.17</td><td>97.87</td><td>94.22</td></tr><tr><td>Accuracy out Range:</td><td>99.51</td><td>74.27</td><td>9.67</td></tr><tr><td>Accuracy long exprs:</td><td>90</td><td>41</td><td>28</td></tr><tr><td>Parameter size:</td><td>not disclosed</td><td>1.00E+09</td><td>2.70E+08</td></tr></table> <p>Verification: github -&gt; slightly updated notebooks transformer4, transformer5, RNN0</p>		RNN0:	transformer0:	transformer1:	transformer4:	Total Parameters:	20505	1494724	1494724	114628	Architecture Parameters:	8901	490241	490241	36209	Optimizer Parameters:	1704	596453	596453	76419	MAE in Range:	0.2937096	0.0301983	0.0301983	0.0448075	MRE in Range:	0.1297235	0.0151836	0.0151836	0.0188847	MRE out Range:	3.7189743	4.3057064	4.3057064	4.3702314	MRE out Range:	0.3856639	0.5431157	0.5431157	0.5201449	MAE long Expressions:	3.7682796	6.0603277	6.0603277	6.1632099	Benchmark score:	0.4446330	5.2696369	5.2696369	2.8852827		Gamma 2.2 Pro	Gamma 3.1B	Gamma 3.279H	Accuracy in Range:	95.17	97.87	94.22	Accuracy out Range:	99.51	74.27	9.67	Accuracy long exprs:	90	41	28	Parameter size:	not disclosed	1.00E+09	2.70E+08	5
	RNN0:	transformer0:	transformer1:	transformer4:																																																																				
Total Parameters:	20505	1494724	1494724	114628																																																																				
Architecture Parameters:	8901	490241	490241	36209																																																																				
Optimizer Parameters:	1704	596453	596453	76419																																																																				
MAE in Range:	0.2937096	0.0301983	0.0301983	0.0448075																																																																				
MRE in Range:	0.1297235	0.0151836	0.0151836	0.0188847																																																																				
MRE out Range:	3.7189743	4.3057064	4.3057064	4.3702314																																																																				
MRE out Range:	0.3856639	0.5431157	0.5431157	0.5201449																																																																				
MAE long Expressions:	3.7682796	6.0603277	6.0603277	6.1632099																																																																				
Benchmark score:	0.4446330	5.2696369	5.2696369	2.8852827																																																																				
	Gamma 2.2 Pro	Gamma 3.1B	Gamma 3.279H																																																																					
Accuracy in Range:	95.17	97.87	94.22																																																																					
Accuracy out Range:	99.51	74.27	9.67																																																																					
Accuracy long exprs:	90	41	28																																																																					
Parameter size:	not disclosed	1.00E+09	2.70E+08																																																																					
Friday, 3 October r 2025	<p>I included a attention based LSTM into the excel sheet. Afterwards I got to work on FNN2, which was fairly easy to make and didn't take long at all. The keras-tuner took 3h to determine the optimal number of neurons and layers is: 2 layers each with 9 neurons. The closer I etched to this process being finished the less RAM memory did the system have left, so the final couple took very long.</p> <p>Trial 98 Complete (00h 03m 54s) val_loss: 0.031836286187171936</p> <p>Best val_loss So Far: 0.0012506949715316296 Total elapsed time: 03h 06m 18s</p> <p>Search: Running Trial #99</p> <table><tr><th>Value</th><th>Best Value So Far</th><th>Hyperparameter</th></tr><tr><td>53</td><td>9</td><td>num_neurons</td></tr><tr><td>16</td><td>2</td><td>num_layers</td></tr><tr><td>False</td><td>False</td><td>dropout/F</td></tr></table> <p>Epoch 1/100 60/60 748s 11s/step - loss: 19.3370 - val_loss: 12.8697 Epoch 2/100 60/60 22s 314ms/step - loss: 7.1973 - val_loss: 2.5593</p> <p>Verification: github -&gt; FNN/FNN2.ipynb</p>	Value	Best Value So Far	Hyperparameter	53	9	num_neurons	16	2	num_layers	False	False	dropout/F	6																																																										
Value	Best Value So Far	Hyperparameter																																																																						
53	9	num_neurons																																																																						
16	2	num_layers																																																																						
False	False	dropout/F																																																																						
Saturd ay, 4 October r 2025	<p>Today I was able to finally finish up work with the FNN2 notebook and thus the excel spreadsheet, and thus the evaluation part of my project!!!! This is veery good news. The results vary slightly from what I expected for some reason. From my hypothesis. Which since the zwischen produkt has been: smaller model with fewer tweakable parameters should perform better on this easy task. Because of the concept of overkill. This is from my lit. review. To be more precise on what I did today: I reran the keras-tuner, now leaving more ram space for it by not using the gnome environment. I also made a bit of debugging: The FNN2 first yielded much worse results after the keras-tuner, after like 15 mins I finally realised I had forgotten to train the model, after loading it with the required hyperparameters.</p> <table><tr><th></th><th>FNNs:</th></tr><tr><td>Total Parameters:</td><td>6,211</td></tr><tr><td>Architecture Parameters:</td><td>6,211</td></tr><tr><td>Optimizer Parameters:</td><td>0</td></tr><tr><td>MAE in Range:</td><td>0.0389406</td></tr><tr><td>MRE in Range:</td><td>0.0155504</td></tr><tr><td>MAE out Range:</td><td>2.2301364</td></tr><tr><td>MRE out Range:</td><td>0.2509270</td></tr><tr><td>MAE long Expressions:</td><td>6.2929916</td></tr><tr><td>Benchmark score:</td><td>6.6345832</td></tr></table> <p>Verification: github -&gt; FNN2</p>		FNNs:	Total Parameters:	6,211	Architecture Parameters:	6,211	Optimizer Parameters:	0	MAE in Range:	0.0389406	MRE in Range:	0.0155504	MAE out Range:	2.2301364	MRE out Range:	0.2509270	MAE long Expressions:	6.2929916	Benchmark score:	6.6345832	4																																																		
	FNNs:																																																																							
Total Parameters:	6,211																																																																							
Architecture Parameters:	6,211																																																																							
Optimizer Parameters:	0																																																																							
MAE in Range:	0.0389406																																																																							
MRE in Range:	0.0155504																																																																							
MAE out Range:	2.2301364																																																																							
MRE out Range:	0.2509270																																																																							
MAE long Expressions:	6.2929916																																																																							
Benchmark score:	6.6345832																																																																							
Sunda y, 5 October r 2025	<p>Today I just reorganized the github again, especially gave proper names to the files in pre-trained-transformers. I also tried using google jules to generate a readme file. This didn't work, because jules started hlucinating very drastically.</p>	2																																																																						
Monda y, 6 October r 2025	<p>Today I downloaded LaTeX, like arian and simon did. When I was writing the lit. review overleaf had notified me that the document was soon going to be too large, and that I'll need to subscribe.</p> <p>I am now using Visual Studio Code in combination with git as an editor.</p> <p>After setting everything up I was able to start the document and stuff. I decided to make two documents btw. One about the methodology and a second about the findings and theoretical background.</p> <p>I started with the introduction to the methodology.</p> <p>I struggled a lot with the copilot being ANNOYING. I was able to just completely deactivate him. I'm not dealing with that.</p>  <p>Verification: github synchronized /documentation</p>	5																																																																						
Tuesd ay, 7 October r 2025	<p>I began with the "real" writing part today. I wrote the sections: train-test, FNN. I also set up the bibliography and a whole bunch of other stuff. I was able to work with VSC synced with git between Desktop and laptop</p> <p>Oh and I fixed the problem with the FNN1 notebook.</p> <p>For more details: just check the commits.</p> <p>Verification: github -&gt; methodology.tex</p>	7																																																																						
Wedne sday, 8 October r 2025	<p>I started Today with writing the part about the benchmark, I'm still not sure wether to include theory in methodology or findings.</p> <p>I then did all of RNNs including simple RNN and attentional RNN. I rethinked Dropout a little. I implemented the stuff I wrote about RNNs earlier, but I think it's AI generated, so I'll have to re write it. I'll keep the formula probably.</p> <p>Ohhh right and I also trained a new model which I called RNN2. It is a copy from the FNN2 notebook. It just implements keras-tuner for SimplerNN, to give it a more fair chance in the excel table. It's still the worst model by far=)</p> <p>Contents</p> <table><tr><td>0.1 - Brief Summary of the Project</td><td>1</td></tr><tr><td>0.2 - Introduction to this document</td><td>1</td></tr><tr><td>1 - Feed-forward Neural Networks (FNN)</td><td>4</td></tr><tr><td>1.1 - Train and Test data</td><td>4</td></tr><tr><td>1.2 - Training a Neural Network Using Tensorflow</td><td>4</td></tr><tr><td>1.3 - FNN1 and FNN2 Notebook</td><td>6</td></tr><tr><td>1.4 - The Benchmark</td><td>6</td></tr><tr><td>1.5 - Drop-Out</td><td>7</td></tr><tr><td>2 - Recurrent Neural Network (RNN)</td><td>7</td></tr><tr><td>2.1 - Numerical Visualization of a RNN</td><td>7</td></tr><tr><td>2.2 - RNN1 and RNN2</td><td>8</td></tr><tr><td>3 - Attention and Transformers</td><td>9</td></tr><tr><td>3.1 - Attentional RNN</td><td>9</td></tr><tr><td>References</td><td>10</td></tr></table> <p>Above you can see how the Table of contents looks right now for verification.</p> <p>Verification: github -&gt; RNN2.ipynb and methodology.tex</p>	0.1 - Brief Summary of the Project	1	0.2 - Introduction to this document	1	1 - Feed-forward Neural Networks (FNN)	4	1.1 - Train and Test data	4	1.2 - Training a Neural Network Using Tensorflow	4	1.3 - FNN1 and FNN2 Notebook	6	1.4 - The Benchmark	6	1.5 - Drop-Out	7	2 - Recurrent Neural Network (RNN)	7	2.1 - Numerical Visualization of a RNN	7	2.2 - RNN1 and RNN2	8	3 - Attention and Transformers	9	3.1 - Attentional RNN	9	References	10	7																																										
0.1 - Brief Summary of the Project	1																																																																							
0.2 - Introduction to this document	1																																																																							
1 - Feed-forward Neural Networks (FNN)	4																																																																							
1.1 - Train and Test data	4																																																																							
1.2 - Training a Neural Network Using Tensorflow	4																																																																							
1.3 - FNN1 and FNN2 Notebook	6																																																																							
1.4 - The Benchmark	6																																																																							
1.5 - Drop-Out	7																																																																							
2 - Recurrent Neural Network (RNN)	7																																																																							
2.1 - Numerical Visualization of a RNN	7																																																																							
2.2 - RNN1 and RNN2	8																																																																							
3 - Attention and Transformers	9																																																																							
3.1 - Attentional RNN	9																																																																							
References	10																																																																							
Thursd ay, 9 October r 2025	<p>Today I did a lot. I reworked documentation directory. I added requirements.txt I wrote A LOT OF LATEX. I was able to finish the methodology document. Well, not finish... I finished writing everything in there. There's still stuff for me to correct and stuff.</p> <p>I also added AI_assistance directory, which contains a latex file where I list the major requests I had with AI. I didn't really have any. Only one. It's the only reason I included this directory.</p> <p>Afterwards I shared my progress with arian and he did likewise. He suggested, I should add a gittignore file. He helped me through this, but I mainly used the google jules AI for this. It was helpful, but I could've done it myself much quicker, I just wanted to use it for fun.</p> <p>So yeah, I cleaned up my repo and am currently trying to use jules again to write a readme file for the repo, after this I'll go to bed.</p> <p>Edit: this took longer than expected.</p> <p>Verification: check github commits</p>	6.5																																																																						
Friday, 10 October	<p>I started writing findings. There I wrote a whoole bunch. 2 pages to be exact with lots of research on the side.</p> <p>Yesterday, I was able to employ a great technique for explaining how a RNN architecture works: I explain it a little, I show the equations, I explain the equations. I think that should be enough.</p>	3.5																																																																						



	<div><math display="block">f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)</math><math display="block">i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)</math><math display="block">\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C)</math><math display="block">o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)</math><math display="block">C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t</math><math display="block">h_t = o_t \odot \tanh(\tilde{C}_t)</math></div> <div>Source for the equations: <a href="#">GeeksforGeeks, 2025a</a></div> <div>Some beautiful equations describing the LSTM architecture above.</div> <div>I also decided to put all the theory into the findings document</div> <div>Verification: <a href="#">github/documentation/findings</a></div>		
Saturday, 11 October r 2025	<div>Check it out!!! I finished 2 pages on the transformer architecture. What's left now are the pre-trained finetuned LLMs. This should be shorter, but the next section, is the most important one: the real findings. I already have a lot to say about my data and I have explanations for the way the benchmarks shaped themselves the way they did.</div> <div>The writing went smooth</div> <div><b>Contents</b><div><div>0.1 Abstract</div><div>0.2 Introduction</div><div>0.3 Feed Forward Neural Networks (FNNS)</div></div><div>1 RNN<div><div>1.1 Numerical Visualization of a RNN</div><div>1.2 Relevant Takeaway</div></div></div><div>2 Other Types of RNNs<div><div>2.1 Long Short-Term Memory (LSTM)</div><div>2.2 Gated Recurrent Unit (GRU)</div><div>2.3 Bidirectional LSTM with Attention</div></div></div><div>3 Transformers<div><div>3.1 Multi-Head Self-Attention</div><div>3.2 The Encoder Layer</div><div>3.3 Point-wise Feed-Forward Network</div><div>3.4 Positional Encoding</div></div></div><div>References</div></div> <div>Verification: <a href="#">github/documentation/findings</a></div>	6.5	
Sunday, 12 October r 2025	<div>I really wanted to finish the findings today, but I couldn't and I think it's for the best. I wrote a section on the pre-trained transformers, which felt kinda dry, because it was just a big block of text. Later in the day I also wrote the section on final evaluation and conclusion. The problem is I didn't really have a conclusion LOL I wrote down some notes and was able to add a latex table of the findings, but not more.</div> <div>Verification: check github commits</div>	6	
Monday, 13 October r 2025			
Tuesday, 14 October r 2025			
Wednesday, 15 October r 2025			
Thursday, 16 October r 2025			
Friday, 17 October r 2025			
Saturday, 18 October r 2025			
Sunday, 19 October r 2025	<div>I wrote a conclusion / thesis. It doesn't sound very convincing, I'll probably change it slightly later. I also wrote a hypothesis. I was able to rethink the results from a new perspective and form a sort of fresh opinion.</div> <div>During this, I thought of something I forgot to do earlier or didn't really pursue enough: positional embedding for FNN</div> <div>I basically copied the whole algorithm from the already working one in the transformers library. And copied the rest of the notebook from FNN2. The "combining" part was done with the help of AI as well as manual input of course.</div> <div>Verification: Github</div>	3.5	<div>Structure for guided discussion with marton:</div> <div><div><div>- Introduction</div><div>- Consent for recording</div><div>- Question about past experience with neural networks</div><div>- Open presentation lead by me.</div><div>- Closing remarks and stuff</div><div>- Camera off</div><div>- Show rest of the project if requested</div><div>- Give the feedback questionnaire to Marton</div></div></div>
Monday, 20 October r 2025	<div>I worked on the conclusion again and pretty much finished it up. I also added FNN3 infos to methodology and findings and Aiusage.</div> <div>I set up the date for an interview with a buddy. He fits the Zielgruppe perfectly.</div>	3	
Tuesday, 21 October r 2025	<div>I started working on the Presentation for marton today, I don't think it will be much most of the talk will be very improvised and it will be more of a discussion, than a presentation. I put together some graphs to help me explain the different architectures today. Oh, and I also wrote a thing or two about further works.</div> <div>Verification: Look my image</div> <div><div>1</div><div>2</div><div>3</div><div>4</div></div> <div>A sneak peak into the presentation.</div>	3	
Wednesday, 22 October r 2025			
Thursday, 23 October r 2025	<div>I finished up the documentation today. Well for the most part. I added some details, I ran an AI model through my pdfs and let it correct my text for grammar errors. It did a surprisingly well job, I still had to mow through the pages after it and clean everything up, but I was impressed. I also moved and rewrote the dropout section.</div> <div>I had a meeting today, in which we talked with Herr Schneider, he helped me out a lot with my questions.</div> <div>I later wrote to him about him correcting my documentation, I'm not sure if he's going to because altogether it's 27 pages. That might scare him off. But in reality there's relatively little real content in there.</div> <div><div>Fremdsprache Gröze Antoin</div><div>Web Sites</div><div><div>Findings.pdf</div><div>methodology.pdf</div></div></div> <div>Snippet of my message to Herr Schneider</div> <div>Verification: <a href="#">github: 2 commits</a></div>	6	
Friday, 24 October r 2025	<div>I finished work for the discussion with marton. We will meet tomorrow at 11 o'clock. I prepared a short powerpoint. And a questionnaire, in which I ask him all the relevant stuff, as well as also see if he understood what I taught him.</div> <div>I worked till late today, because I still had lots to do and mr.Schneider gave me some great Ideas on what to do with marton, to squeeze as much information out of him as possible.</div> <div>BTW I decided to only explain the FNN and transformer models. And skip over the RNNs entirely. All in all the presentation itself isn't long. But I want it to be this way, so that Marton gets the ability to constantly barrage me with questions.</div> <div><h2>Guided Discussion</h2><div>When you submit this form, it will not automatically collect your details like name and email address unless you provide it yourself.</div><h3>Kontrollfragen zum Thema</h3><div>1. Das Trasformer model besitzt "positional awareness". Daten die ins Model hineingefüttert werden, laufen also zuerst durch keinen</div><div>The discussion will be in german.</div></div>	5.5	



	<div><p>PROZESSBESCHREIBUNG DER BACKPROPAGATIONSDIAGONALISIERUNG</p><p>FNK2 Loss Landscape (Red dots indicate optimal configurations)</p><p>Figure 2:</p><p>Verification: check github.</p></div>	
<div><p>Monday, 10 November 2025</p></div>	<div><p>Today I only worked on the documentation. I was able to finish writing the findings section in findings as well as the conclusion, I redid the abstract a little. I also was able to implement changes suggested by my mom in methodology. After finishing everything up I feel pretty satisfied with the results.</p></div>	8
<div><p>Tuesday, 11 November 2025</p></div>	<div><p>Today I worked pretty much the whole day, only taking a break during the PE classes after lunch. What I did first is I wrote some extra stuff on the topics that I felt needed some further explanation to fulfill the categories on the right. After doing that I wrote the whole section on backpropagation, in order to make the training process more apparent.</p><p>Following this I implemented all the changes from my moms final review. I reformatted the documentation and thus finished it.</p><p>I then wrote some more details into the Prozessbeschreibung word document.</p><p>Finally I collected all files, uploaded most of them to the github repo. And sorted the rest in a different file separately.</p><p>Finally I will have submitted my project (hopefully in time.....)</p></div>	10