LLMagogy or fast multi-stage pre-training of Large Language Models with pre-selected limited datasets.

“knowing a few principles frees you from knowing many facts” R. Descartes

“the knowledge of certain principles easily compensates the lack of knowledge of certain facts” C. Helvétius

Lyubimkov Dmitri

E-mail: loftlong@gmail.com

2025

Abstract

This article proposes a novel multi-stage pre-training approach for large language models (LLMs), inspired by human learning processes. Traditional LLM pre-training involves training massive models on trillions of tokens, requiring significant computational resources and time. In contrast, the proposed method mimics human education by gradually "maturing" the model through stages, starting with small, curated datasets (e.g., textbooks) and progressively increasing model complexity and data volume. Key innovations include:

- dividing training into stages corresponding to human developmental phases,
- simulating model "growth" by incrementally increasing network depth (D) and width (W)
- preserving previously learned knowledge using a freezing coefficient (F) to stabilize earlier parameters.

This approach aims to create models that understand, rather than memorize, information, while reducing resource consumption. The method remains theoretical and requires experimental validation, but it offers a promising alternative to conventional LLM training, with potential benefits in efficiency, control, and model interpretability.

Table of contents

1. LLM pre-training now

Briefly, the process can be described as follows: a neural network of a large, final size is taken and filled with texts with trillions and trillions of tokens. As a result, a trained encoder and decoder are obtained, but this requires a lot of time, a lot of computing power, a lot of electricity. As a result, only a few companies and countries can pre-train modern large language models. In addition to this, with such a number of tokens, it is simply impossible to talk about any impact on the process and result of training in terms of choosing and presenting educational material.

At the same time, we all know very well, constantly encounter and even took part in the training of neural networks that achieve a similar result with much less expenditure of resources and time. I am talking now about a person, about us, about you personally. We definitely do not consume trillions of tokens, our reading speed is too low for this. We definitely do not spend so much energy. Even if we take into account the fact that the mechanisms of human brain training are not completely clear, we can confidently say that the number of calories that we consume in the form of food and some of which go to the brain are insignificant compared to what is spent on LLM training. Perhaps the reason is in the difference in architectures - transformers with back propagation and the brain. But let's assume that the architecture of the neural network, although important in terms of training speed or the resulting accuracy, is not fundamental and that any neural network of the right size with the right architecture will come to the desired result. And in fact, the reason is HOW we teach LLM and a person, the difference in the learning process.

How does human learning differ from LLM learning? Firstly, a person grows, his brain develops from infancy to adulthood, and LLM has its final sizes immediately. Secondly, a person can read and comprehend a limited number of texts, simply due to the speed with which he can read. LLM sees trillions of tokens many times during its learning. Thirdly, human learning is cyclical, because it is associated with his maturation, with the seasons, with some other cycles, and in each subsequent cycle a person reads other texts, using previous knowledge as a foundation for new ones. LLM also learns in epochs, but in each epoch it sees the same texts. From the point of view of LLM and the volume of its texts, a person should not be able to learn anything at all. But you were able to, at least because you are now reading and understanding this text. In order for people to be able to understand and learn texts, the science of pedagogy and special collections of texts called textbooks were invented. We can't experiment on people and determine what exactly causes them to learn from texts, but this article suggests that it has something to do with how learning occurs. And we can simulate such learning in LLM pre-training.

2. Pre-training LLM as a person

What conditions must be met in order to train LLM using the method that people train:

1. Prepare educational materials – textbooks, problem books, literary works, etc.
2. Divide the training materials into parts. Each part will correspond to its own stage of neural network training. Let's not confuse this period with the epoch. The stage refers to the corpus of texts used, and the epoch is the iteration of neural network training with the current corpus of texts
3. Figure out how to simulate neural network maturation in a transformer-based LLM, because without it, a large LLM will simply memorize a small set of educational materials
4. Think of a way to save what you have learned previously so that new knowledge is based on the old

2.1. Preparation of educational materials

Perhaps future research will reveal that for LLM based on transformers, educational materials need to be prepared differently than for people. Perhaps they need more texts. But now we are simulating human learning, so as educational materials we will take everything that people/children learn on. All books, textbooks, problem books from the age when a child begins to understand speech and his parents read books to them until finishing comprehensive school. Completion of comprehensive school is not necessarily the limit, it’s just that further our education is divided into many specializations and collecting educational materials will require significant effort.

This will simply be excessive for testing the hypothesis of the possibility of learning LLM according to the ideas of this article.

2.2. Division of educational materials into stages

Perhaps future research will reveal that for LLM based on transformers, the training materials need to be broken down into parts differently than for humans. They are not tied to the period of human maturation and our annual cycles. Perhaps the training materials will not need to be broken down into stages at all and it will be possible to train the neural network in one stage. But now we are simulating human learning, so all the training material will be broken down by years. We should have about 15 stages - from infancy to adulthood.

All educational materials will be combined by the age at which the child studies them. The first few years (preschool years) can be combined into one stage due to the insignificance of the volumes of the text corpus.

2.3. LLM growing up simulation

LLM can be represented as a sequence of transformers. Therefore, the first characteristic of the "age" of LLM will be the number of layers of transformers in it. Let's call it "depth" - D0. Each transformer has its own dimensionality, the dimension of the embedding vector at its output. This is the second characteristic of the "age" of LLM. Let's call it width - W0.

How to make an infantile LLM? We need to choose parameters D and W so small that the neural network cannot learn the texts by heart. And at the same time they should be large enough so that the neural network is able to learn the “knowledge” in these texts. The choice of parameters D and W is described in more detail in the next chapter.

Now we do a regular pre-training of a small LLM on a very small corpus of texts. This will not take long. The next step is to grow the LLM by one stage so that we can repeat its training on the next set of texts. We do this by increasing the parameters D and W to D1 and W1. The strategy for growing the parameters D and W should be determined experimentally - we can change only one of the parameters at each stage, or we can change both parameters simultaneously. The order and magnitude of the parameter changes are a matter of testing and experimentation. As a result, we will get an LLM of a slightly larger size and with a larger number of parameters. We learned some of these parameters at the previous stage, and the other part is new. We can initiate new parameters randomly or with zeros. But the author believes that the most correct way is to set their initial values so that the result of the increased LLM does not differ from the result of the LLM of the previous stage on the texts of the previous stage. In this way, we preserve the continuity of LLM generations.

2.4. Retention of knowledge gained in previous stages of LLM training

Knowledge is a set of internal LLM parameters. To “save” them between LLM generations, they need to be “frozen” a little. To do this, we introduce another freezing coefficient – F. This coefficient can take a value from 0 to 1. This coefficient is tied to the stage/generation of the trained LLM. When moving from generation 0 to generation 1, we get the coefficient F0, when moving to generation 2 – F1, and so on. It is not necessary to tie the coefficient to generations, but to have one for all training stages. This coefficient is used together with the learning rate coefficient α as another multiplier. Combined α1 = α*F0, α2 = α*F0*F1, and so on. Using the F parameter, we slow down the changes in the neural network parameters calculated at the previous stage, forcing the neural network of the current generation to use mainly the newly added parameters. Choosing the coefficient F=0 generally cancels the change in the parameters of the previous stages. The value of the coefficient F=1 disables this mechanism. Table 1 shows an example of the rate of change of parameters of different generations of LLM at F=1/2

Stage	α0	α1	α2	α3
0	-	-	-	-
1	½	-	-	-
2	¼	½	-	-

3	1/8	¼	½	-

That is, as the training stage increases, the rate of change of parameters from older LLM generations decreases all the time. And the further away the generation in which these parameters were added, the slower they will change.

2.5. Completing pre-training

By the time the last stage of pre-training is completed, we should have a model at the level of a high school student (or at the level for which we selected the training materials). It will have knowledge, a model of the world gleaned from the texts, an understanding of these texts, because this is how the training was built. Let's call this model the pre-final LLM. But the model will not know enough facts, because it simply has not seen them. Now we need to further train the LLM on the full corpus of texts (Wikipedia, reddit, stackoveflow or whatever else is included there). Similar to the transition between stages, we expand the LLM to the maximum size for D and W and set the F coefficient to 0 - i.e. we protect all previously found neural network parameters from change. Next, we train the LLM in the same way as we would have trained it before. As a result, we should get an LLM that has facts from the entire corpus of texts used in training, but at the same time is forced to rely on the knowledge and models obtained during training on a small number of selected training materials.

3. Selection of coefficients D and W at each stage of training

If the text is very simple for a person and there is a lot of time to learn it, then the person will be able to simply memorize it. If the topic is too complex and voluminous and there is not enough basic knowledge, then no matter how much a person tries to learn the educational material, he is unlikely to succeed. And only in an intermediate ratio of the volume and complexity of texts on the one hand and knowledge and time on the other hand is it possible to learn the educational material. The same should happen with neural networks - too large a network, too many parameters and a sufficient number of epochs will lead to the fact that the texts will be memorized word for word. Let me remind you that we are training LLM on a limited corpus of texts. The opposite situation is also true - too small a neural network will not be able to learn the educational material. The D and W we need lie somewhere in the interval between these extremes.

Perhaps some of us have encountered a situation in our lives when we have learned a topic or section of science so well that we can say that we have understood it, not just learned it. Perhaps you have seen how others, having encountered a logical board game or a video game or a card game for the first time, suddenly begin to play it unexpectedly well. At the same time, they simply did not have time to learn this game, to get acquainted with examples of games played by others or to read articles on strategy. The author suggests that this is also related to understanding. Understanding is the ability of the brain to build a model that will predict the future, the continuation - the next move, the next word, the reaction to an action, and so on. This model may not have a connection with real cause-and-effect relationships, its task is to correctly predict, predict the next token in the case of LLM.

Let's assume that LLMs based on transformers are capable of not only learning a corpus of texts, but also understanding them. But to do this, you need to put the neural network in a position where it could not learn the texts, but could "understand" them. The author believes that building a text model requires a smaller neural network than learning a text, which in turn is smaller than the neural network for memorizing a text. If the memorized text is just a copy of the original text, then the learned text is a zip archive of the text - an attempt to increase the amount of information per stored bit/byte. The understood text can be viewed as a picture compressed in jpeg or music in mp3. Due to the understanding of how information is structured, how tokens are related to each other, you can build a model that allows, albeit with losses, but more significantly compress the original texts. And this means that you need to look for suitable W and D by testing neural networks from a smaller dimension to a larger dimension. From a certain size, LLM will already be able to show sufficient performance. At this stage we do not need high accuracy because it will be associated with learning, and we want to achieve understanding. Until the last stage of training, we need an knowledge LLM, from which at the last stage we make a facts LLM by adding trillions and trillions of tokens to it, while freezing the parameters responsible for knowledge. Thus, forcing to combine knowledge and facts during training.

The traditional approach to LLM training is unable to lead to understanding for several reasons. First, a large number of unstructured and unprepared texts are used. The neural network simply does not have time to find too complex patterns. Imagine that a person studies not from textbooks divided into subjects and topics, but from texts chaotically distributed by topics, complexity and volume. Second, such LLMs have too many parameters, they are aimed at learning facts, but for understanding the number of parameters should be less. But if there are fewer, then memorizing facts will become impossible. And only dividing LLM pretraining into the first phase of understanding and the second phase of memorization allows us to bypass this contradiction. Perhaps the assumption that LLM based on transformers will strive to understand the text with a sufficiently small number of parameters, instead of learning it, is incorrect. But even in this case, training in several stages can still work, although with a lower probability.

I think as we conduct experiments and gain experience, we will be able to derive an empirical formula for depth (Dn) and width (Wn) depending on D(n-1) and W(n-1), stage number n, the amount of educational material in tokens, and some other parameters.

4. Selection of the F coefficient at each stage of training

Correctly selected coefficients D and W and educational material consistent with the previous stages should not significantly change the already learned parameters. In an ideal world, the coefficient F can be equal to 1 and nothing will change. In a non-ideal world, the same can be achieved by setting the coefficient F to 0. But even in an ideal world, new knowledge can clarify the previous ones and therefore should not be hindered so much. It is necessary to establish how much the parameters of the previous stages can change in total. Suppose we want the sum of the moduli (or squares) of changes in all parameters not to exceed 20% of the sums of their moduli (squares) of their values. After that, we need to select the coefficient F so close to 1 that this rule is still fulfilled.

5. Advantages, features and nuances of multi-stage LLM pre-training

The described method of LLM training has its own characteristics, obvious advantages and unexpected possible areas of application. Let's consider some of them.

5.1. Independence from neural network architecture

The only requirements for a neural network are the ability to gradually increase its size without losing results (although even the absence of losses is not necessarily possible) and the presence of a learning rate coefficient for the network so that it can manipulate changes in previously trained parameters.

5.2. PedAIgogy - AI/LLM pedagogy

The described method of training, if it works, creates a new scientific/technical discipline - AI/LLM pedagogy. The result of LLM training depends on the collected training materials and their division into stages. This could not even be dreamed of with the usual method of LLM training, since the entire body of texts has a huge volume, is collected together and is fed for training in an unstructured flow. Perhaps, it will be necessary to add teachers to the groups responsible for training and in the future, AI training will become somewhat similar to raising children.

5.3. Less resource consumption for training neural networks

At all stages except the last one, a limited text corpus is used and the neural network has an insignificant size compared to the final one. Therefore, it will require less time, computing resources and electricity for training. At the last stage, the entire text corpus is used and the model has the full size. It seems that there is no gain in time and resources and even a small loss. But the pre-final LLM is already trained enough to predict texts at the level of textbooks. This gives us hope that the last stage with training on the entire text corpus will require fewer epochs, or will give greater accuracy on the same number of epochs, or will allow training a larger LLM for the same resources.

5.4. Pre-final LLM as a basis for experts in MoE architecture

The pre-final LLM has the basic knowledge to understand a subset of texts related to one subject area, which means that it can be used to train an LLM expert. The advantage of the pre-final LLM is its compact size compared to the traditionally obtained LLM. This means that the LLM expert can be made smaller, or more experts can be trained, or the accuracy can be increased while maintaining the size.

5.5. Pre-final LLM as a gatekeeper in MoE architecture

The pre-final LLM can be further trained with general knowledge without specialization and without facts. In this case, we can obtain a gatekeeper model that can determine which expert models should be involved.

5.6. Based/Cringe detector

The pre-final LLM has only the base knowledge, pre-selected and approved. The large model learns from texts collected from the Internet and may contain unreliable, fake or cringe information. Comparison of the results of the base and full networks can be used to identify such information in order to eliminate it from training or labeling accordingly.

5.7. AI morality

The constraints on LLM imposed at the RLHF stage are similar to rules or laws. They do not affect the base model, they simply prohibit it from responding in a certain way or, on the contrary, force it to respond in a certain way. Such constraints are then bypassed using prompt engineering. Morality is the rules that a person considers his own, he responds or does things not because he was told to do so, but because he cannot do otherwise. Step-by-step pretraining of LLM with the preservation of the parameters of the previous stages can be used to set the morality of the neural network. If this morality is embedded at the initial stages, it cannot be changed or canceled without significant retraining of the model because the old parameters are used so often that disabling them or changing their values will break the model itself.

Considering what and how can be used to give LLM morality is beyond the scope of this article.

6. Testing the method

All ideas presented in the article are assumptions based by the author on his knowledge, experience, common sense and intuition. All of them are subject to experimental verification. Unfortunately, the author does not have sufficient personal computing resources or access to computing resources to check everything independently. Therefore, I will limit myself to methods by which it would be possible to check the result of training.

6.1. Pre-final LLM evaluation

The pre-final LLM is valuable in itself as a basis for further training. We used school textbooks for its training. It would be logical to use the same school tests and exams that were missing from the training set for evaluating the results.

6.2. Evaluation of the final result

The final result is an LLM and its performance should be evaluated just like any other LLM. Test just as you would test the LLMs trained in the usual way.

But in addition to the LLM itself, we need to estimate the time spent on its training. Because we assume that the pre-final LLM should reduce the number of epochs needed to train the final LLM. If this does not happen, then apparently the assumptions in this part are incorrect. To do this, we need to compare the number of epochs that were required to train two LLMs with the same accuracy using different methods.