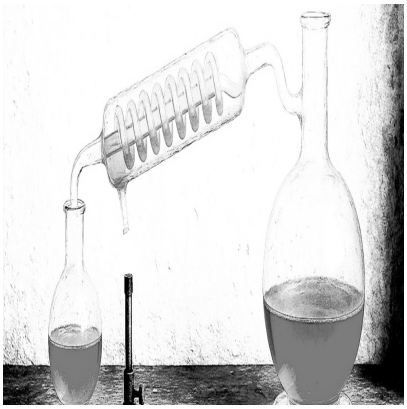


Mnemonic infidelity

Description

default watermark

Training a large language model (LLM) starts with dividing a very large corpus of texts into basic units, i.e., recurring tokens (such as symbols and parts of words), and calculating the relative positions of tokens in the texts. These relationships are processed in a neural network to create very large numerical vectors that represent patterns of associations between tokens. These vectors function as *operational surrogates* for meaning â?? representing statistical associations across contexts in the training corpus. These numerical token vectors are organised into a digital dictionary or look-up table associating tokens with vectors. Such a dictionary stays constant throughout the subsequent training and can be used across different training episodes.

Through a look-up process the stream of text tokens in the training data is converted into a stream of numerical vectors. This conversion involves further operations that take account of the position of each token in the context of the other tokens preceding it (a context window) in the source texts, as well as transformations based on semantic and positional operations that predict where different readers are likely to focus their attention [see ChatGPTâ??s refinement to this description in note below]. This stream of vectors is the input to the neural network LLM.

Training the large language model involves feeding these modified vectors into a structured network, which starts out as a matrix of numbers (weights) attached to interconnected parameters. Think of these as nodes and links in a network organised in layers. The training algorithm treats each transformed token vector in the training corpus as sequential input.

As the network encounters each token in the training data in turn, the training algorithm propagates that information through the network and makes fine adjustments to the network parameters. It carries out these operations over many iterations. The aim is to adjust the parameters such that the LLM is able to predict the next token that will follow any sequence of inputs, even sequences it has not encountered in the training data. Once the LLM is trained, the training data is discarded from the process. What we are left with is an array of floating point numbers that is the language model.

I have described here in rough terms the core operation of training a large language model. The parameters in the network are also â??fine tunedâ?• by exposure to other training data specific to a

particular domain and feedback from human operators. LLMs are also incorporated into web search and other procedures that extend their capability, but Iâ??ve restricted this explanation here to the basics. For further explanations see posts tagged LLM.

A radical multi-step transformation

Training a neural network is a radical multi-step transformation of a body of text into a table of numbers. Once the source data is selected and the operational parameters are set, the process is automatic, taking several hours or days of high-performance computer time. Once trained, an LLM functions as a highly effective machine for reconstructing patterns in text data and predicting grammatically correct and information rich language flows.

This transformation from text to numerical language model removes the possibility of reconstructing the source data from the LLM. The LLM canâ??t be searched or analysed for token frequency or other linguistic stats. It denies any of the usual functions of a text database. Such functions are sacrificed in return for the creation of an LLM as a powerful prediction engine that can generate token sequences to deliver convincing conversational exchanges, as well as linguistically-based problem solving, summarisation, language translation and computer programs.

default watermark

The LLM does not provide a condensation, reduction, compression or summarisation of the source data. An LLM may well be of similar size, or larger than the size of the source data. ChatGPT tells me that GPT-3, for example, was trained on around 570 GB of text, and its model is around 700+ GB. This negates the usefulness of metaphors of compression, extraction, summarization, or distillation if taken literally.

Distillation

Iâ??ve been looking for a suitable conceptualisation of the process. Of course, I consulted an AI for some terms. The concept of â??distillationâ?• is alluring when applied to LLMs. It suggests a process of refinement, of extracting the essential components of something, as if purifying a liquid. The result is something more refined, more useful, economical, efficient and smaller than the original.

I asked Google search â??Do LLMs distill knowledge from a text corpus?â?• The search engine used its AI to offer an emphatic overview of whatâ??s on the Web: â??Yes, Large Language Models (LLMs) do distill knowledge from a text corpus. â?¦ This learning process allows them to implicitly capture and store a vast amount of knowledge about the world, including language, grammar, syntax, and common sense.â?•

Perhaps I forced the term â??distillâ?• through my search query. But I have yet to find a book or article that describes the process of training a large language model as a â??distillation.â?• That said, â??knowledge extractionâ?• is common, with a similar connotation of a process that yields something smaller, more refined and more useful for a particular purpose than the raw material, the abundant source. The term mainly applies to reducing the size of a trained LLM by adjusting the accuracy of some of the parameters (through â??quantisationâ?•). LLM researchers and developers apply the term to techniques for reducing a trained model to something smaller so that it can run more efficiently or take up less space in computer memory. Judging by the number of publications on the topic, model

distillation is a big field in the development of large language models.

But the idea that an LLM *distills* or *extracts* from its training data implies a reduction in size, retaining only the â??essenceâ?• of the results of training. But the trained model can be as large as, or even larger than, the dataset it was trained on.

Pattern modelling

So, metaphors such as â??distillingâ?• and â??extracting contentâ?• from a source texts is less helpful than *radical transformation.*

ChatGPT advised me that the LLM training process can be understood as a kind of *statistical transformation*, or more specifically: pattern modelling, parameterisation of language space and transfer learning substate. I like its detailed descriptions of these metaphors.

- *Pattern modelling* is acquiring high-dimensional correlations between tokens across varied contexts.
- *Parameterisation of language space* suggests that â??the weights encode probabilities across an immense latent space, representing *linguistic tendencies* rather than actual text.â?•
- *Transfer learning substrate* implies that the model doesnâ??t store or recite specific documents, but is primed to generate new sequences â??in line with the statistical contours of its training.â?•

default watermark

The chatbot elaborated further that LLMs are â??functionally adaptive systems,â?• not archives, and certainly not a verbatim corpus of its training set: â??The training set acts more like a *field of influence* than a storehouse.â?• Such a framing feeds into discussions about intellectual property. At least it cannot be said that LLMs distribute their source material to users of their AI systems.

Generative flexibility

The chatbot then suggested some alternative metaphors, such as *mining* that restructures the terrain, or *cultivation*, as if the soil is the training data and the model emerges as a plant. It also suggested *crystallisation*: â??From a supersaturated solution (training data), structure emerges â?? selective, ordered, yet dependent on diffuse prior matter.â?•

The AI seemed to approve of my initial prompt on the subject: â??Calling LLM training a *radical transformation* is apt. Itâ??s not a distillation, but a statistical re-encoding â?? one that privileges generative flexibility over mnemonic fidelity.â?•

Bibliography

- Franceschelli, Giorgio, Claudia Cevenini, and Mirco Musolesi. 2024. *Training Foundation Models as Data Compression: On Information, Model Weights and Copyright Law*. arXiv.

Note

- Featured image is from ChatGPT: Please generate an image of a defunct glass lab distilling apparatus â?¦ show similar but with a spiral condenser and a Bunsen burner. Show liquid in the glass tubes.
- According to ChatGPT, my â??reference to â??readersâ?? focusâ?? overstates the intention of the model; transformers do not model human attention directly. Rather, they use a mathematically-defined *attention mechanism* that weighs the relevance of each token to others in the context window.â?•

Category

1. Artificial Intelligence

Tags

1. LLM
2. machine learning
3. metaphor
4. training

default watermark

Date Created
June 14, 2025
Author
rcoyne99