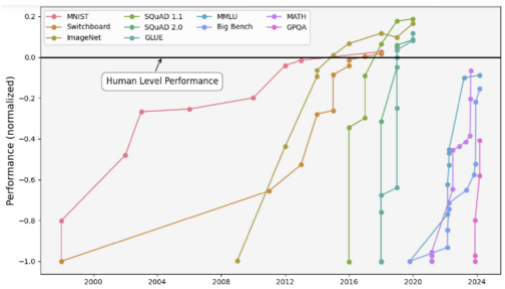
Закон нейронного масштабирования

В машинном обучении**закон нейронного масштабирования** — это эмпирический закон масштабирования, который описывает, как меняется производительность нейронной сети при увеличении или уменьшении ключевых факторов. К таким факторам обычно относятся количество параметров, размер обучающего набора данных, [1][2] и стоимость обучения. Некоторые модели также демонстрируют повышение производительности при масштабировании вывода за счет увеличения времени выполнения теста (TTC), что позволяет применять законы нейронного масштабирования не только на этапе обучения, но и на этапе развертывания. [3]



Эффективность моделей искусственного интеллекта по различным показателям с 1998 по 2024 год

Введение

В целом модель глубокого обучения можно охарактеризовать четырьмя параметрами: размером модели, размером обучающего набора данных, стоимостью обучения и показателем ошибки после обучения (например, показателем ошибки на тестовом наборе). Каждая из этих переменных может быть выражена действительным числом, которое обычно записывается как N, D, C, L (соответственно: количество параметров, размер набора данных, вычислительная сложность и потери).

Закон нейронного масштабирования — это теоретический или эмпирический статистический закон, описывающий взаимосвязь между этими параметрами. Существуют и другие параметры с другими законами масштабирования.

Размер модели

В большинстве случаев размер модели определяется количеством параметров. Однако при использовании разреженных моделей, таких как модели на основе смеси экспертов, возникает одна сложность. [4] В разреженных моделях при логическом выводе

используется лишь часть параметров. Для сравнения: большинство других типов нейронных сетей, таких как трансформерные модели, при логическом выводе используют все свои параметры.

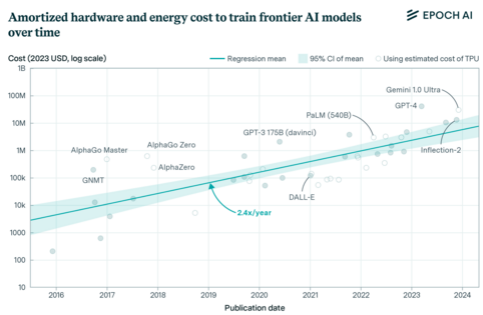
Размер обучающего набора данных

Размер обучающего набора данных обычно определяется количеством содержащихся в нем точек данных. Как правило, предпочтение отдается большим обучающим наборам данных, поскольку они предоставляют более богатый и разнообразный источник информации, на основе которого модель может обучаться. Это может привести к повышению эффективности обобщения при применении модели к новым, ранее не встречавшимся данным. [5] Однако увеличение размера обучающего набора данных также требует больше вычислительных ресурсов и времени для обучения модели.

При использовании метода «предварительное обучение, затем дообучение», применяемого для большинства больших языковых моделей, существует два типа обучающих наборов данных: *набор данных для предварительного обучения* и *набор данных для дообучения*. Их размер по-разному влияет на производительность модели. Как правило, набор данных для дообучения составляет менее 1 % от набора данных для предварительного обучения.[6]

В некоторых случаях для тонкой настройки достаточно небольшого количества высококачественных данных, а увеличение объема данных не всегда приводит к повышению производительности.[6]

Стоимость обучения



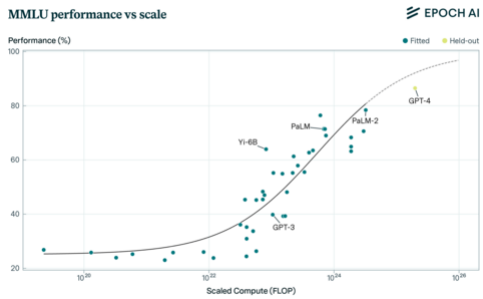
Амортизационные расходы на оборудование и электроэнергию для обучения передовых моделей ИИ с течением времени

Стоимость обучения обычно измеряется в единицах времени (сколько времени требуется для обучения модели) и вычислительных ресурсов (сколько требуется вычислительной

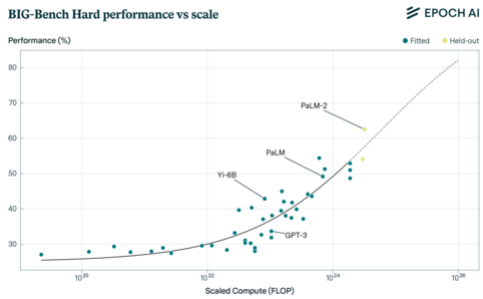
мощности и памяти). Важно отметить, что стоимость обучения можно значительно снизить за счёт эффективных алгоритмов обучения, оптимизированных программных библиотек и параллельных вычислений на специализированном оборудовании, таком как графические процессоры или тензорные процессоры.

Стоимость обучения модели нейронной сети зависит от нескольких факторов, в том числе от размера модели, размера обучающего набора данных, алгоритма обучения сложности и доступных вычислительных ресурсов. [5] В частности, удвоение размера обучающего набора данных не обязательно приведет к удвоению стоимости обучения, поскольку модель можно обучать несколько раз на одном и том же наборе данных (каждый раз это будет «эпоха»).

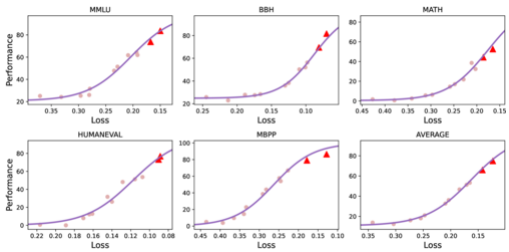
Производительность



Производительность MMLU в сравнении с масштабом искусственного интеллекта в виде сигмоиды



BIG-Bench (hard)[7] performance vs AI scale as a sigmoid



Performance on several benchmarks vs negative log-likelihood loss L (with L_0 removed), fitted as sigmoid functions[8]: Figure 2

The performance of a neural network model is evaluated based on its ability to accurately predict the output given some input data. Common metrics for evaluating model performance include:[5]

- Negative log-likelihood per token (logarithm of perplexity) for language modeling;
- Accuracy, precision, recall, and F1 score for classification tasks;
- Mean squared error (MSE) or mean absolute error (MAE) for regression tasks;
- Elo rating in a competition against other models, such as gameplay[9] or preference by a human judge.[10]

Performance can be improved by using more data, larger models, different training algorithms, regularizing the model to prevent overfitting, and early stopping using a validation set.

When the performance is a number bounded within the range of $[0, 1]$, such as accuracy, precision, etc., it often scales as a sigmoid function of cost, as seen in the figures.

Примеры

(Хестнесс, Наранг и др., 2017)

Статья 2017 года[2] является общепринятым источником информации о законах нейронного масштабирования, выведенных с помощью статистического анализа экспериментальных данных. Предыдущие работы, проведенные до 2000-х годов, на которые ссылаются авторы статьи, были либо теоретическими, либо проводились на масштабах, на несколько порядков меньших. В то время как в предыдущих работах показатель масштабирования обычно был равен $L \propto D^{-\alpha}$, с $\alpha \in \{0.5, 1, 2\}$, газета обнаружила , что $\alpha \in [0.07, 0.35]$.

Из всех факторов, которые они варьировали, только задача может изменить показатель степени αИзменение оптимизаторов, регуляризаторов и функций потерь в архитектуре изменит только коэффициент пропорциональности, но не показатель степени. Например, для одной и той же задачи одна архитектура может иметь $L = 1000D^{-0.3}$ в то время как другой мог бы $L = 500D^{-0.3}$Они также обнаружили, что для заданной архитектуры количество параметров, необходимых для достижения наименьших потерь при фиксированном размере набора данных, растет пропорционально $N \propto D^{\beta}$ для другого показателя β.

Они изучали машинный перевод с использованием рекуррентной нейронной сети ($\alpha \sim 0.13$), генеративное языковое моделирование с использованием рекуррентной нейронной сети ($\alpha \in [0.06, 0.09], \beta \approx 0.7$), классификация ImageNet с использованием ResNet ($\alpha \in [0.3, 0.5], \beta \approx 0.6$), а также распознавание речи с использованием двух

гибридных архитектур (LSTM, дополненных либо сверточными нейронными сетями, либо декодером внимания) ($\alpha \approx 0.3$).

(Хениган, Каплан и др., 2020)

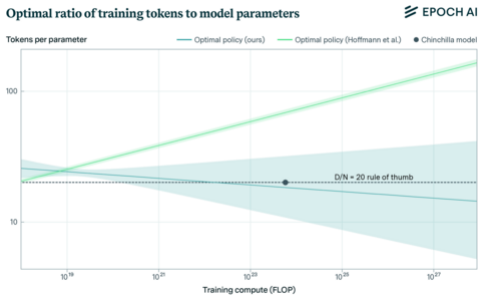
Анализ, проведенный в 2020 году [11], изучил статистические взаимосвязи между C, N, D, L Мы исследовали широкий диапазон значений и обнаружили схожие законы масштабирования в этом диапазоне. $N \in [10^3, 10^9]$, $C \in [10^{12}, 10^{21}]$, а также в различных форматах (текст, видео, изображения, преобразование текста в изображение и т. д.).[11]

В частности, были обнаружены следующие законы масштабирования (таблица 1 из [11]):

- Для каждого модального значения они фиксировали одно из двух C, N, а другой — изменяем (D изменяется при использовании $D = C/6N$), достижимая потеря при тестировании удовлетворяет следующим условиям:
  $$L = L_0 + \left(\frac{x_0}{x}\right)^\alpha$$
  где x — изменяемая переменная, и L_0, x_0, α являются параметрами, которые определяются путем статистической подгонки. Параметр α Это самое важное.
  - Когда N является варьируемой переменной, α диапазоны от 0.037 Для 0.24 в зависимости от типа модели. Это соответствует $\alpha = 0.34$ Из статьи о чешуе шиншиллы.
  - Когда C является варьируемой переменной, α диапазоны от 0.048 Для 0.19 в зависимости от типа модели. Это соответствует $\beta = 0.28$ Из статьи о чешуе шиншиллы.
- При фиксированном вычислительном бюджете оптимальное количество параметров модели неизменно составляет около
  $$N_{opt}(C) = \left(\frac{C}{5 \times 10^{-12}\text{petaFLOP-day}}\right)^{0.7} = 9.0 \times 10^{-7} C^{0.7}$$
  Параметр 9.0×10^{-7} варьируется в зависимости от модальности до 10 раз. Параметр показателя степени 0.7 варьируется от 0.64 Для 0.75 для различных модальностей. Этот показатель соответствует ≈ 0.5 Из статьи о чешуе шиншиллы.
- «Есть веские основания полагать» (но это не подтверждено статистически), что $D_{opt}(C) \propto N_{opt}(C)^{0.4} \propto C^{0.28}$ Этот показатель соответствует ≈ 0.5 Из статьи о чешуе шиншиллы.

Закон масштабирования $L = L_0 + (C_0/C)^{0.048}$ Это было подтверждено в ходе обучения GPT-3 (рис. 3.1 [12]).

Масштабирование шиншилл (Хоффманн и др., 2022)



Optimal ratio of training tokens to model parameters for Chinchilla scaling law. It shows that in general, "Chinchilla optimal" scaling is $D = 20N$, and is significantly different from (Hoffmann et al., 2022). Data analysis by Epoch AI.[13]

One particular scaling law ("Chinchilla scaling") states that, for a large language model (LLM) autoregressively trained for one epoch, with a cosine learning rate schedule, we have:[14]

$$\begin{cases} C = C_0 N D \\ L = \frac{A}{N^\alpha} + \frac{B}{D^\beta} + L_0 \end{cases}$$

where the variables are

- C is the cost of training the model, in FLOPS.
- N is the number of parameters in the model.
- D is the number of tokens in the training set.
- L is the average negative log-likelihood loss per token (nats/token), achieved by the trained LLM on the test dataset.
  - L_0 represents the loss of an ideal generative process on the test data
  - $\frac{A}{N^\alpha}$ captures the fact that a Transformer language model with N parameters underperforms the ideal generative process
  - $\frac{B}{D^\beta}$ captures the fact that the model trained on D tokens underperforms the ideal generative process

and the statistical parameters are

- $C_0 = 6$, meaning that it costs 6 FLOPs per parameter to train on one token. This is estimated by Kaplan et al.[15] Note that training cost is much higher than inference cost, as training entails both forward and backward passes, whereas inference costs 1 to 2 FLOPs per parameter to infer on one token.
- $\alpha = 0.34, \beta = 0.28, A = 406.4, B = 410.7, L_0 = 1.69$.

Although Besiroglu et al.[16] claims that the statistical estimation is slightly off, and should be $\alpha = 0.35, \beta = 0.37, A = 482.01, B = 2085.43, L_0 = 1.82$.

The statistical laws were fitted over experimental data with $N \in [7 \times 10^7, 1.6 \times 10^{10}], D \in [5 \times 10^9, 5 \times 10^{11}], C \in [10^{18}, 10^{24}]$.

Since there are 4 variables related by 2 equations, imposing 1 additional constraint and 1 additional optimization objective allows us to solve for all four variables. In particular, for any fixed C, we can uniquely solve for all 4 variables that minimizes L. This provides us with the optimal $D_{opt}(C), N_{opt}(C)$ for any fixed C:

$$N_{opt}(C) = G\left(\frac{C}{6}\right)^a, \quad D_{opt}(C) = G^{-1}\left(\frac{C}{6}\right)^b, \quad \text{where} \quad G = \left(\frac{\alpha A}{\beta B}\right)^{\frac{1}{\alpha+\beta}}, \quad a = \frac{\beta}{\alpha+\beta}, \text{and } b = \frac{\alpha}{\alpha+\beta}.$$

Plugging in the numerical values, we obtain the "Chinchilla efficient" model size and training dataset size, as well as the test loss achievable:

$$\begin{cases} N_{opt}(C) = 0.6\ C^{0.45} \\ D_{opt}(C) = 0.3\ C^{0.55} \\ L_{opt}(C) = 1070\ C^{-0.154} + 1.7 \end{cases}$$

Similarly, we may find the optimal training dataset size and training compute budget for any fixed model parameter size, and so on.

There are other estimates for "Chinchilla efficient" model size and training dataset size. The above is based on a statistical model of $L = \frac{A}{N^\alpha} + \frac{B}{D^\beta} + L_0$. One can also directly fit a statistical law for $D_{opt}(C), N_{opt}(C)$ without going through the detour, for which one obtains:

$$\begin{cases} N_{opt}(C) = 0.1\ C^{0.5} \\ D_{opt}(C) = 1.7\ C^{0.5} \end{cases}$$

or as tabulated:

$N_{opt}(C)$	C / FLOP	C / FLOPs of training Gopher	$D_{opt}(C)$
400 million	1.92e+19	1/29968	8.0 billion
1 billion	1.21e+20	1/5706	20.2 billion
10 billion	1.23e+22	1/2819	205.1 billion
67 billion	5.76e+23	1	1.5 trillion
175 billion	3.85e+24	6.7	3.7 trillion
280 billion	9.90e+24	17.2	5.9 trillion
520 billion	3.43e+25	59.5	11.0 trillion
1 trillion	1.27e+26	221.3	21.2 trillion
10 trillion	1.30e+28	22515.9	216.2 trillion

Discrepancy

The Chinchilla scaling law analysis for training transformer language models suggests that for a given training compute budget (C), to achieve the minimal pretraining loss for that budget, the number of model parameters (N) and the number of training tokens (D) should be scaled in equal proportions, $N_{opt}(C) \propto C^{0.5}, D_{opt}(C) \propto C^{0.5}$. This conclusion differs from analysis conducted by Kaplan et al.,[15] which found that N should be increased more quickly than D, $N_{opt}(C) \propto C^{0.73}, D_{opt}(C) \propto C^{0.27}$.

This discrepancy can primarily be attributed to the two studies using different methods for measuring model size. Kaplan et al.:[17]

- did not count the parameters in the token embedding layer, which when analyzed at smaller model sizes leads to biased coefficients;
- studied smaller models than the Chinchilla group, magnifying the effect;
- assumed that $L_\infty = 0$.

Secondary effects also arise due to differences in hyperparameter tuning and learning rate schedules. Kaplan et al.:[18]

- used a warmup schedule that was too long for smaller models, making them appear less efficient;
- did not fully tuning optimization hyperparameters.

Beyond Chinchilla scaling

As Chinchilla scaling has been the reference point for many large-scaling training runs, there had been a concurrent effort to go "beyond Chinchilla scaling", meaning to modify some of the training pipeline in order to obtain the same loss with less effort, or deliberately train for longer than what is "Chinchilla optimal".

Usually, the goal is to make the scaling law exponent larger, which means the same loss can be trained for much less compute. For instance, filtering data can make the scaling law exponent larger.[19]

Another strand of research studies how to deal with limited data, as according to Chinchilla scaling laws, the training dataset size for the largest language models already approaches what is available on the internet.[20] found that augmenting the dataset with a mix of "denoising objectives" constructed from the dataset improves performance.[21] studies optimal scaling when all available data is already exhausted (such as in rare languages), so one must train multiple epoches over the same dataset (whereas Chinchilla scaling requires only one epoch).

The Phi series of small language models were trained on textbook-like data generated by large language models, for which data is only limited by amount of compute available.[22]

Chinchilla optimality was defined as "optimal for training compute", whereas in actual production-quality models, there will be a lot of inference after training is complete. "Overtraining" during training means better performance during inference.[23] LLaMA models were overtrained for this reason. Subsequent studies discovered scaling laws in the overtraining regime, for dataset sizes up to 32x more than Chinchilla-optimal.[24]

Broken neural scaling laws (BNSL)

A 2022 analysis[25] found that many scaling behaviors of artificial neural networks follow a smoothly broken power law functional form:

$$y = a + \left(bx^{-c_0}\right)\prod_{i=1}^{n}\left(1+\left(\frac{x}{d_i}\right)^{1/f_i}\right)^{-c_i*f_i}$$

in which x refers to the quantity being scaled (i.e. C, N, D, number of training steps, number of inference steps, or model input size) and y refers to the *downstream* (or upstream) performance evaluation metric of interest (e.g. prediction error, cross entropy, calibration error, AUROC, BLEU score percentage, F1 score, reward, Elo rating, solve rate, or FID score) in zero-shot, prompted, or fine-tuned settings. The parameters $a, b, c_0, c_1 \ldots c_n, d_1 \ldots d_n, f_1 \ldots f_n$ are found by statistical fitting.

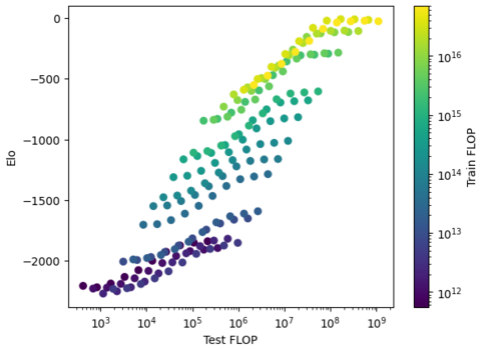
On a log–log plot, when f_i is not too large and a is subtracted out from the y-axis, this functional form looks like a series of linear segments connected by arcs; the n transitions between the segments are called "breaks", hence the name *broken neural scaling laws (BNSL)*.

The scenarios in which the scaling behaviors of artificial neural networks were found to follow this functional form include large-scale vision, language, audio, video, diffusion, generative modeling, multimodal learning, contrastive learning, AI alignment, AI capabilities, robotics, out-of-distribution (OOD) generalization, continual learning, transfer learning, uncertainty estimation / calibration, out-of-distribution detection, adversarial robustness, distillation, sparsity, retrieval, quantization, pruning, fairness, molecules, computer programming/coding, math word problems, arithmetic, emergent abilities, double descent, supervised learning, unsupervised/self-supervised learning, and reinforcement learning (single agent and multi-agent).

The architectures for which the scaling behaviors of artificial neural networks were found to follow this functional form include residual neural networks, transformers, MLPs, MLP-mixers, recurrent neural networks, convolutional neural networks, graph neural networks, U-nets, encoder-decoder (and encoder-only) (and decoder-only) models, ensembles (and non-

ensembles), MoE (mixture of experts) (and non-MoE) models, and sparse pruned (and non-sparse unpruned) models.

Inference scaling



The Elo rating of various AlphaZero agents trained to play the board game of Hex at varying train-time and test-time compute

Other than scaling up training compute, one can also scale up inference compute (or "test-time compute"[3]). As an example, the Elo rating of AlphaGo improves steadily as it is allowed to spend more time on its Monte Carlo Tree Search per play.[26]:Fig 4 For AlphaGo Zero, increasing Elo by 120 requires either 2x model size and training, or 2x test-time search.[27] Similarly, a language model for solving competition-level coding challenges, AlphaCode, consistently improved (log-linearly) in performance with more search time.[28]

For Hex, 10x training-time compute trades for 15x test-time compute.[9] For *Libratus* for heads up no-limit Texas hold 'em, and *Cicero* for *Diplomacy*, and many other abstract games of partial information, inference-time searching improves performance at a similar tradeoff ratio, for up to 100,000x effective increase in training-time compute.[27]

In 2024, the OpenAI o1 report documented that o1's performance consistently improved with both increased train-time compute and test-time compute, and gave numerous examples of test-time compute scaling in mathematics, scientific reasoning, and coding tasks.[29][30]

One method for scaling up test-time compute is **process-based supervision**, where a model generates a step-by-step reasoning chain to answer a question, and another model (either human or AI) provides a reward score on some of the intermediate steps, not just the final answer. Process-based supervision can be scaled arbitrarily by using synthetic reward score without another model, for example, by running Monte Carlo rollouts and scoring each step in the reasoning according to how likely it leads to the right answer. Another method is by **revision models**, which are models trained to solve a problem multiple times, each time revising the previous attempt.[31]

Other examples

Vision transformers

Vision transformers, similar to language transformers, exhibit scaling laws. A 2022 research trained vision transformers, with parameter counts $N \in [5 \times 10^6, 2 \times 10^9]$, on image sets of sizes $D \in [3 \times 10^7, 3 \times 10^9]$, for computing $C \in [0.2, 10^4]$ (in units of TPUv3-core-days).[32]

After training the model, it is finetuned on ImageNet training set. Let L be the error probability of the finetuned model classifying ImageNet test set. They found

$$\min_{N,D} L = 0.09 + \frac{0.26}{(C + 0.01)^{0.35}}.$$

Neural machine translation

Ghorbani, Behrooz et al.[33] studied scaling laws for neural machine translation (specifically, English as source, and German as target) in encoder-decoder Transformer models, trained until convergence on the same datasets (thus they did not fit scaling laws for computing cost C or dataset size D). They varied $N \in [10^8, 3.5 \times 10^9]$ They found three results:

- L is a scaling law function of N_E, N_D, where N_E, N_D are encoder and decoder parameter count. It is not simply a function of total parameter count $N = N_E + N_D$. The function has form $L\left(N_e, N_d\right) = \alpha\left(\frac{\bar{N}_e}{N_e}\right)^{p_e}\left(\frac{\bar{N}_d}{N_d}\right)^{p_d} + L_\infty$, where $\alpha, p_e, p_d, L_\infty, \bar{N}_e, \bar{N}_d$ are fitted parameters. They found that $N_d/N \approx 0.55$ minimizes loss if N is held fixed.
- L "saturates" (that is, it reaches L_∞) for smaller models when the training and testing datasets are "source-natural" than "target-natural". A "source-natural" data point means a pair of English-German sentences, and the model is asked to translate the English sentence into German, and the English sentence is written by a natural English writer, while the German sentence is translated from the English sentence by a machine translator.[34] To construct the two kinds of datasets, the authors collected natural English and German sentences online, then used machine translation to generate their translations.
- As models grow larger, models trained on source-original datasets can achieve low loss but bad BLEU score. In contrast, models trained on target-original datasets achieve low loss and good BLEU score in tandem (Figure 10, 11 [33]).

The authors hypothesize that source-natural datasets have uniform and dull target sentences, and so a model that is trained to predict the target sentences would quickly overfit.

[35] trained Transformers for machine translations with sizes $N \in [4 \times 10^5, 5.6 \times 10^7]$ on dataset sizes $D \in [6 \times 10^5, 6 \times 10^9]$. They found the Kaplan et al. (2020)[15] scaling law

applied to machine translation: $L(N, D) = \left[\left(\frac{N_C}{N}\right)^{\frac{\alpha_N}{\alpha_D}} + \frac{D_C}{D}\right]^{\alpha_D}$. They also found the BLEU score scaling as $BLEU \approx Ce^{-kL}$.

Transfer learning

Hernandez, Danny et al.[36] studied scaling laws for transfer learning in language models. They trained a family of Transformers in three ways:

- pretraining on English, finetuning on Python
- pretraining on an equal mix of English and Python, finetuning on Python
- training on Python

The idea is that pretraining on English should help the model achieve low loss on a test set of Python text. Suppose the model has parameter count N, and after being finetuned on D_F Python tokens, it achieves some loss L. We say that its "transferred token count" is D_T, if another model with the same N achieves the same L after training on $D_F + D_T$ Python tokens.

They found $D_T = 1.9e4(D_F)^{.18}(N)^{.38}$ for pretraining on English text, and $D_T = 2.1e5(D_F)^{.096}(N)^{.38}$ for pretraining on English and non-Python code.

Precision

Kumar et al.[37] study scaling laws for numerical precision in the training of language models. They train a family of language models with weights, activations, and KV cache in varying numerical precision in both integer and floating-point type to measure the effects on loss as a function of precision. For training, their scaling law accounts for lower precision by wrapping the effects of precision into an overall "effective parameter count" that governs loss scaling, using the parameterization $N \mapsto N_{\text{eff}}(P) = N(1 - e^{-P/\gamma})$. This illustrates how training in lower precision degrades performance by reducing the true capacity of the model in a manner that varies exponentially with bits.

For inference, they find that extreme overtraining of language models past Chinchilla-optimality can lead to models being more sensitive to quantization, a standard technique for efficient deep learning. This is demonstrated by observing that the degradation in loss due to weight quantization increases as an approximate power law in the token/parameter ratio D/N seen during pretraining, so that models pretrained on extreme token budgets can perform worse in terms of validation loss than those trained on more modest token budgets if post-training quantization is applied. Other work examining the effects of overtraining include Sardana et al.[38] and Gadre et al.[39]

Densing laws

Xiao et al.[8] considered the parameter efficiency ("density") of models over time. The idea is that over time, researchers would discover models that use their parameters more efficiently, in that models with the same performance can have fewer parameters.

A model can have an actual parameter count N, defined as the actual number of parameters in the model, and an "effective" parameter count $\hat{N}$, defined as how many parameters it would have taken a previous well-known model to reach he same performance on some benchmarks, such as MMLU. $\hat{N}$ is not measured directly, but rather by measuring the actual model performance S, then plugging it back to a previously fitted scaling law, such as the Chinchilla scaling law, to obtain what $\hat{N}$ would be required to reach that performance S, according to that previously fitted scaling laws.

A densing law states that $\ln\left(\frac{\hat{N}}{N}\right)_{max} = At + B$, where t is real-world time, measured in days.

Смотрите также

- Большая языковая модель — тип модели машинного обучения
- Базовая модель — парадигма модели искусственного интеллекта
- Общий искусственный интеллект — тип искусственного интеллекта с широким спектром возможностей
- Гипотеза Гильберта — степенной рост энтропии языка или стохастического процесса

Ссылки

1. Бахри, Ясаман; Дайер, Итан; Каплан, Джаред; Ли, Джэхун; Шарма, Уткарш (2024). «Объяснение законов нейронного масштабирования» (https://www.ncbi.nlm.nih.gov/pmc/articles/PMC11228526) . *Proceedings of the National Academy of Sciences*. **121** (27) e2311878121. arXiv:2102.06701 (https://arxiv.org/abs/2102.06701) . Bibcode:2024PNAS..12111878B (https://ui.adsabs.harvard.edu/abs/2024PNAS..12111878B) . doi:10.1073/pnas.2311878121 (https://doi.org/10.1073%2Fpnas.2311878121) . PMC 11228526 (https://www.ncbi.nlm.nih.gov/pmc/articles/PMC11228526) . PMID 38913889 (https://pubmed.ncbi.nlm.nih.gov/38913889) .

2. Гестнесс, Джоэл; Наранг, Шаран; Ардалани, Ньюша; Даймос, Грегори; Джун, Хиву; Кианинеджад, Хассан; Патвари, доктор медицинских наук Мостофа Али; Янг, Янг; Чжоу, Яньци (2017-12-01). "Масштабирование глубокого обучения предсказуемо эмпирически". arXiv:1712.00409 (https://arxiv.org/abs/1712.00409) [cs.LG (https://arxiv.org/archive/cs.LG)].

3. Кобб, Карл; Косараджу, Винит; Бавариан, Мохаммад; Чен, Марк; Джун, Хеву; Кайзер, Лукаш; Плапперт, Маттиас; Твирек, Джерри; Хилтон, Джейкоб (18 ноября 2021 г.), *Обучение верификаторов решению текстовых математических задач*, arXiv:2110.14168 (https://arxiv.org/abs/2110.14168)

4. Раджбхандари, Самьям; Ли, Цунлун; Яо, Чжэвэй; Чжан, Минцзя; Аминабади, Реза Яздани; Аван, Аммар Ахмад; Рэсли, Джефф; Хэ, Юйсюн (28 июня 2022 г.). "DeepSpeed-MoE: совершенствование логического вывода и обучения на основе смеси экспертов для масштабирования искусственного интеллекта нового поколения" (https://proceedings.mlr.press/v162/rajbhandari22a.html) . *Материалы 39-й Международной конференции по машинному обучению*. PMLR: 18332–18346. arXiv:2201.05596 (https://arxiv.org/abs/2201.05596) .

5. Гудфеллоу И., Бенджио Я., Курвиль А. (2016). Глубокое обучение. MIT Press.

6. Чжоу, Чуньтин; Лю, Пэнфэй; Сюй, Пусинь; Айер, Шрини; Сунь, Цзяо; Мао, Юнин; Ма, Сюэчжэ; Эфрат, Авиа; Ю, Пин; Ю, Лили; Чжан, Сьюзан; Гош, Гарги; Льюис, Майк; Зеттлемойер, Люк; Леви, Омер (1 мая 2023 г.). «LIMA: чем меньше, тем лучше для выравнивания». arXiv:2305.11206 (https://arxiv.org/abs/2305.11206) [cs.CL (https://arxiv.org/archive/cs.CL)].

7. "google/BIG-bench" (https://github.com/google/BIG-bench/tree/main) . Google. 2024-09-24. Получено 2024-09-25.

8. Сяо, Чаоцзюнь; Цай, Цзе; Чжао, Вэйлинь; Цзэн, Гоян; Линь, Биюань; Чжоу, Цзе; Чжэн, Чжи; Хань, Сюй; Лю, Чжиюань (2024-12-06), *Закон уплотнения больших языковых моделей*, arXiv:2412.04315 (https://arxiv.org/abs/2412.04315)

9. Джонс, Энди Л. (2021). «Масштабирование законов масштабирования с помощью настольных игр». arXiv:2104.03113 (https://arxiv.org/abs/2104.03113) [cs.LG (https://arxiv.org/archive/cs.LG)].

10. Рейтинг чат-ботов LMSYS (https://huggingface.co/spaces/lmsys/chatbot-arena-leaderboard)

11. Хениган, Том; Каплан, Джаред; Кац, Мор; Чен, Марк; Хессе, Кристофер; Джексон, Джейкоб; Хеву, Джун; Браун, Том Б.; Дхаривал, Прафулла; Манн, Крис; Рэдфорд, Алек; Рамеш, Адитья; Райдер, Ник; Зиглер, Дэниел М.; Шульман, Джон; Грей, Скотт; Халласи, Крис; Амодей, Дарио; МакКэндлиш, Сэм (27 октября 2020 г.). *Законы масштабирования для авторегрессионного генеративного моделирования*. arXiv:2010.14701 (https://arxiv.org/abs/2010.14701) . OCLC 1228442047 (https://search.worldcat.org/oclc/1228442047) .

12. Браун, Том Б.; Манн, Бенджамин; Райдер, Ник; Суббиа, Мелани; Каплан, Дж.; Дхаривал, Прафулла; Нилакантан, Арвинд; Шьям, Пранав; Састри, Гириш; Аскелл, Аманда; Агарвал, Сандхини; Герберт-Восс, Ариэль; Крюгер, Гретхен; Хениган, Т.; Чайлд, Ревон (28 мая 2020 г.). «Языковые модели способны к обучению с малым количеством примеров». arXiv:2005.14165 (https://arxiv.org/abs/2005.14165) [cs.CL (https://arxiv.org/archive/cs.CL)].

13. Бесироглу, Тамай (17 апреля 2024 г.). "Масштабирование Chinchilla: попытка репликации" (https://epochai.org/blog/chinchilla-scaling-a-replication-attempt) . *Epoch AI*. Получено 24 сентября 2024 г..

14. Хоффманн, Джордан; Боржо, Себастьян; Менш, Артур; Бучацкая, Елена; Цай, Тревор; Резерфорд, Элиза; Касас, Диего де Лас; Хендрикс, Лиза Энн; Вельбл, Йоханнес; Кларк, Эйдан; Хенниган, Том; Ноланд, Эрик; Милликан, Кэти; Дриссе, Джордж ван ден; Дамок, Богдан (29 марта 2022 г.). «Обучение больших языковых моделей с оптимальным использованием вычислительных ресурсов». arXiv:2203.15556 (https://arxiv.org/abs/2203.15556) [cs.CL (https://arxiv.org/archive/cs.CL)].

15. Kaplan, Jared; McCandlish, Sam; Henighan, Tom; Brown, Tom B.; Chess, Benjamin; Child, Rewon; Gray, Scott; Radford, Alec; Wu, Jeffrey; Amodei, Dario (2020). "Scaling Laws for Neural Language Models". *CoRR*. abs/2001.08361. arXiv:2001.08361 (https://arxiv.org/abs/2001.08361) .

16. Besiroglu, Tamay; Erdil, Ege; Barnett, Matthew; You, Josh (2024-04-15). "Chinchilla Scaling: A replication attempt". arXiv:2404.10102 (https://arxiv.org/abs/2404.10102) [cs.AI (https://arxiv.org/archive/cs.AI)].

17. Pearce, Tim; Song, Jinyeop (2024), *Reconciling Kaplan and Chinchilla Scaling Laws*, arXiv:2406.12907 (https://arxiv.org/abs/2406.12907)

18. Porian, Tomer; Wortsman, Mitchell; Jitsev, Jenia; Schmidt, Ludwig; Carmon, Yair (2024-07-25), *Resolving Discrepancies in Compute-Optimal Scaling of Language Models*, arXiv:2406.19146 (https://arxiv.org/abs/2406.19146)

19. Sorscher, Ben; Geirhos, Robert; Shekhar, Shashank; Ganguli, Surya; Morcos, Ari S. (2023-04-21). "Beyond neural scaling laws: beating power law scaling via data pruning". arXiv:2206.14486 (https://arxiv.org/abs/2206.14486) [cs.LG (https://arxiv.org/archive/cs.LG)].

20. Tay, Yi; Wei, Jason; Chung, Hyung Won; Tran, Vinh Q.; So, David R.; Shakeri, Siamak; Garcia, Xavier; Zheng, Huaixiu Steven; Rao, Jinfeng (2022-11-16). "Transcending Scaling Laws with 0.1% Extra Compute". arXiv:2210.11399 (https://arxiv.org/abs/2210.11399) [cs.CL (https://arxiv.org/archive/cs.CL)].

21. Muennighoff, Niklas; Rush, Alexander; Barak, Boaz; Le Scao, Teven; Tazi, Nouamane; Piktus, Aleksandra; Pyysalo, Sampo; Wolf, Thomas; Raffel, Colin A. (2023-12-15). "Scaling Data-Constrained Language Models" (https://proceedings.neurips.cc/paper_files/paper/2023/hash/9d89448b63ce1e2e8dc7af72c984c196-Abstract-Conference.html) . *Advances in Neural Information Processing Systems*. **36**: 50358–50376. arXiv:2305.16264 (https://arxiv.org/abs/2305.16264) .

22. Li, Yuanzhi; Bubeck, Sébastien; Eldan, Ronen; Del Giorno, Allie; Gunasekar, Suriya; Lee, Yin Tat (2023-09-11). "Textbooks Are All You Need II: phi-1.5 technical report". arXiv:2309.05463 (https://arxiv.org/abs/2309.05463) [cs.CL (https://arxiv.org/archive/cs.CL)].

23. Sardana, Nikhil; Frankle, Jonathan (2023-12-31). "Beyond Chinchilla-Optimal: Accounting for Inference in Language Model Scaling Laws". arXiv:2401.00448 (https://arxiv.org/abs/2401.00448) [cs.LG (https://arxiv.org/archive/cs.LG)].

24. Gadre, Samir Yitzhak; Smyrnis, Georgios; Shankar, Vaishaal; Gururangan, Suchin; Wortsman, Mitchell; Shao, Rulin; Mercat, Jean; Fang, Alex; Li, Jeffrey (2024-03-13). "Language models scale reliably with over-training and on downstream tasks". arXiv:2403.08540 (https://arxiv.org/abs/2403.08540) [cs.CL (https://arxiv.org/archive/cs.CL)].

25. Caballero, Ethan; Gupta, Kshitij; Rish, Irina; Krueger, David (2022). "Broken Neural Scaling Laws". arXiv:2210.14891 (https://arxiv.org/abs/2210.14891) [cs.LG (https://arxiv.org/archive/cs.LG)].

26. Silver, David; Huang, Aja; Maddison, Chris J.; Guez, Arthur; Sifre, Laurent; van den Driessche, George; Schrittwieser, Julian; Antonoglou, Ioannis; Panneershelvam, Veda; Lanctot, Marc; Dieleman, Sander; Grewe, Dominik; Nham, John; Kalchbrenner, Nal; Sutskever, Ilya (January 2016). "Mastering the game of Go with deep neural networks and tree search" (https://www.nature.com/articles/nature16961) . *Nature*. **529** (7587): 484–489. Bibcode:2016Natur.529..484S (https://ui.adsabs.harvard.edu/abs/2016Natur.529..484S) . doi:10.1038/nature16961 (https://doi.org/10.1038%2Fnature16961) . ISSN 1476-4687 (https://search.worldcat.org/issn/1476-4687) . PMID 26819042 (https://pubmed.ncbi.nlm.nih.gov/26819042) .

27. Noam, Brown (2024-09-17). *Parables on the Power of Planning in AI: From Poker to Diplomacy: Noam Brown (OpenAI)* (https://www.youtube.com/watch?v=eaAonE58sLU) (Video). Retrieved 2024-09-24 – via YouTube. Lecture at Paul G. Allen School on Thursday, May 23, 2024, 3:30 pm

28. Li, Yujia; Choi, David; Chung, Junyoung; Kushman, Nate; Schrittwieser, Julian; Leblond, Rémi; Eccles, Tom; Keeling, James; Gimeno, Felix; Dal Lago, Agustin; Hubert, Thomas; Choy, Peter; de Masson d'Autume, Cyprien; Babuschkin, Igor; Chen, Xinyun (2022-12-09). "Competition-level code generation with AlphaCode" (https://www.science.org/doi/10.1126/science.abq1158) . *Science*. **378** (6624): 1092–1097. arXiv:2203.07814 (https://arxiv.org/abs/2203.07814) . Bibcode:2022Sci...378.1092L (https://ui.adsabs.harvard.edu/abs/2022Sci...378.1092L) . doi:10.1126/science.abq1158 (https://doi.org/10.1126%2Fscience.abq1158) . ISSN 0036-8075 (https://search.worldcat.org/issn/0036-8075) . PMID 36480631 (https://pubmed.ncbi.nlm.nih.gov/36480631) .

29. Villalobos, Pablo (2023-07-28). "Trading Off Compute in Training and Inference" (https://epochai.org/blog/trading-off-compute-in-training-and-inference) . *Epoch AI*. Retrieved 2024-09-24.

30. "Learning to Reason with LLMs" (https://openai.com/index/learning-to-reason-with-llms/) . OpenAI. Retrieved 2024-09-16.

31. Snell, Charlie; Lee, Jaehoon; Xu, Kelvin; Kumar, Aviral (2024-08-06), *Scaling LLM Test-Time Compute Optimally can be More Effective than Scaling Model Parameters*, arXiv:2408.03314 (https://arxiv.org/abs/2408.03314)

32. Zhai, Xiaohua; Kolesnikov, Alexander; Houlsby, Neil; Beyer, Lucas (2022). "Scaling Vision Transformers" (https://openaccess.thecvf.com/content/CVPR2022/html/Zhai_Scaling_Vision_Transformers_CVPR_2022_paper.html) . *CVPR*: 12104–12113.

33. Ghorbani, Behrooz; Firat, Orhan; Freitag, Markus; Bapna, Ankur; Krikun, Maxim; Garcia, Xavier; Chelba, Ciprian; Cherry, Colin (2021-09-01). "Scaling Laws for Neural Machine Translation". arXiv:2109.07740 (https://arxiv.org/abs/2109.07740) [cs.LG (https://arxiv.org/archive/cs.LG)].

34. Chen, Mia Xu; Firat, Orhan; Bapna, Ankur; Johnson, Melvin; Macherey, Wolfgang; Foster, George; Jones, Llion; Schuster, Mike; Shazeer, Noam; Parmar, Niki; Vaswani, Ashish; Uszkoreit, Jakob; Kaiser, Lukasz; Chen, Zhifeng; Wu, Yonghui (July 2018). "The Best of Both Worlds: Combining Recent Advances in Neural Machine Translation" (https://aclanthology.org/P18-1008) . *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Melbourne, Australia: Association for Computational Linguistics: 76–86. arXiv:1804.09849 (https://arxiv.org/abs/1804.09849) . doi:10.18653/v1/P18-1008 (https://doi.org/10.18653%2Fv1%2FP18-1008) .

35. Gordon, Mitchell A; Duh, Kevin; Kaplan, Jared (2021). "Data and Parameter Scaling Laws for Neural Machine Translation". *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Stroudsburg, PA, USA: Association for Computational Linguistics. pp. 5915–5922. doi:10.18653/v1/2021.emnlp-main.478 (https://doi.org/10.18653%2Fv1%2F2021.emnlp-main.478) .

36. Hernandez, Danny; Kaplan, Jared; Henighan, Tom; McCandlish, Sam (2021-02-01). "Scaling Laws for Transfer". arXiv:2102.01293 (https://arxiv.org/abs/2102.01293) [cs.LG (https://arxiv.org/archive/cs.LG)].

37. Kumar, Tanishq; Ankner, Zachary; Spector, Benjamin F.; Bordelon, Blake; Muennighoff, Niklas; Paul, Mansheej; Pehlevan, Cengiz; Ré, Christopher; Raghunathan, Aditi (2024-11-30), *Scaling Laws for Precision*, arXiv:2411.04330 (https://arxiv.org/abs/2411.04330)

38. Sardana, Nikhil; Portes, Jacob; Doubov, Sasha; Frankle, Jonathan (2024-07-18), *Beyond Chinchilla-Optimal: Accounting for Inference in Language Model Scaling Laws*, arXiv:2401.00448 (https://arxiv.org/abs/2401.00448)

39. Gadre, Samir Yitzhak; Smyrnis, Georgios; Shankar, Vaishaal; Gururangan, Suchin; Wortsman, Mitchell; Shao, Rulin; Mercat, Jean; Fang, Alex; Li, Jeffrey (2024-06-14), *Language models scale reliably with over-training and on downstream tasks*, arXiv:2403.08540 (https://arxiv.org/abs/2403.08540)