Scaling laws for neural language models

Birshert Alexey 171

Introduction

Motivation of the research

How well language models perform based on ...?

Background and methods

Experiment setup

Language models are trained on WebText2, tokenized using byte-pair encoding. Loss function - cross entropy loss. Main model - decoder-only Transformer.

Key findings

Notation

- L - cross entropy loss, averaged over tokens in context
- N - number of model parameters excluding embeddings
- D - dataset size in tokens
- C - amount of compute

Key findings of research

Performance depends strongly on scale, weakly on model shape

Model performance depends strongly on N, D and C. Model performance very weakly on architectural hyperparameters such as depth, width.

Key findings of research

Smooth power laws

Performance has a power-law relationship with each of the three scale factors N, D, C when not bottlenecked by the other two.

Key findings of research

Universality of overfitting

Performance improves predictably as long as we scale up N and D in tandem, but enters a regime of diminishing returns if either N or D is held fixed while the other increases.

Key findings of research

Universality of training

Training curves follow predictable power-laws whose parameters are roughly independent of the model size.

Key findings of research

Transfer improves with test performance

Transfer to a different distribution incurs a constant penalty but otherwise improves roughly in line with performance on the training set.

Key findings of research

Sample efficiency

Large models are more sample-efficient than small models, reaching the same level of performance with fewer optimization steps and using fewer data points.

Key findings of research

Convergence is inefficient

When working within a fixed compute budget C but without any other restrictions on the model size N or available data D, we attain optimal performance by training very large models and stopping significantly short of convergence.

Key findings of research

Optimal batch size

Optimal batch size - balance between speed and compute efficiency. The ideal batch size for training these models is roughly a power of the loss only.

Summary of scaling laws

Notation

- L - cross entropy loss, averaged over tokens in context
- N - number of model parameters excluding embeddings
- D - dataset size in tokens
- C - amount of compute

Summary of scaling laws

For models with a limited number of parameters, trained to convergence on sufficiently large datasets:

$$L(N) = \left(\frac{N_c}{N}\right)^{\alpha_N}$$

$$\alpha_N \sim 0.076; \quad N_c \sim 8.8 \times 10^{13}$$

Summary of scaling laws

For large models trained with a limited dataset with early stopping:

$$L(D) = \left(\frac{D_c}{D}\right)^{\alpha_D}$$

$$\alpha_D \sim 0.095; \quad D_c \sim 5.4 \times 10^{13}$$

Summary of scaling laws

When training with a limited amount of compute, a sufficiently large dataset, an optimally-sized model, and a sufficiently small batch size:

$$L(C_{min}) = \left(\frac{C_c^{min}}{C_{min}}\right)^{\alpha_C^{min}}$$

$$\alpha_C^{min} \sim 0.050; \quad C_c^{min} \sim 3.1 \times 10^8$$

Summary of scaling laws

Equation that governs the simultaneous L dependence on N and D and governs the degree of overfitting:

$$L(N, D) = \left[\left(\frac{N_c}{N} \right)^{\frac{\alpha_N}{\alpha_D}} + \left(\frac{D_c}{D} \right) \right]^{\alpha_D}$$

$$D \propto N^{\frac{\alpha_N}{\alpha_D}} \sim N^{0.74}$$

Summary of scaling laws

The critical batch size, which determines the speed/efficiency balance, also roughly obeys a power law in L:

$$B_{crit}(L) = \frac{B_*}{L^{1/\alpha_B}}$$

$$\alpha_B \sim 0.21; \quad B_* \sim 2 \times 10^8$$

Summary of scaling laws

When training a given model for a finite number of parameter update steps S in the infinite data limit, after an initial transient period, the learning curves can be accurately fit by:

$$L(N, S) = \left(\frac{N_c}{N}\right)^{\alpha_N} + \left(\frac{S_c}{S_{min}(S)}\right)^{\alpha_S}$$

$$\alpha_S \sim 0.76; \quad S_c \sim 2.1 \times 10^3$$

$S_{min}(S)$ - minimum possible number of optimization steps for L

Summary of scaling laws

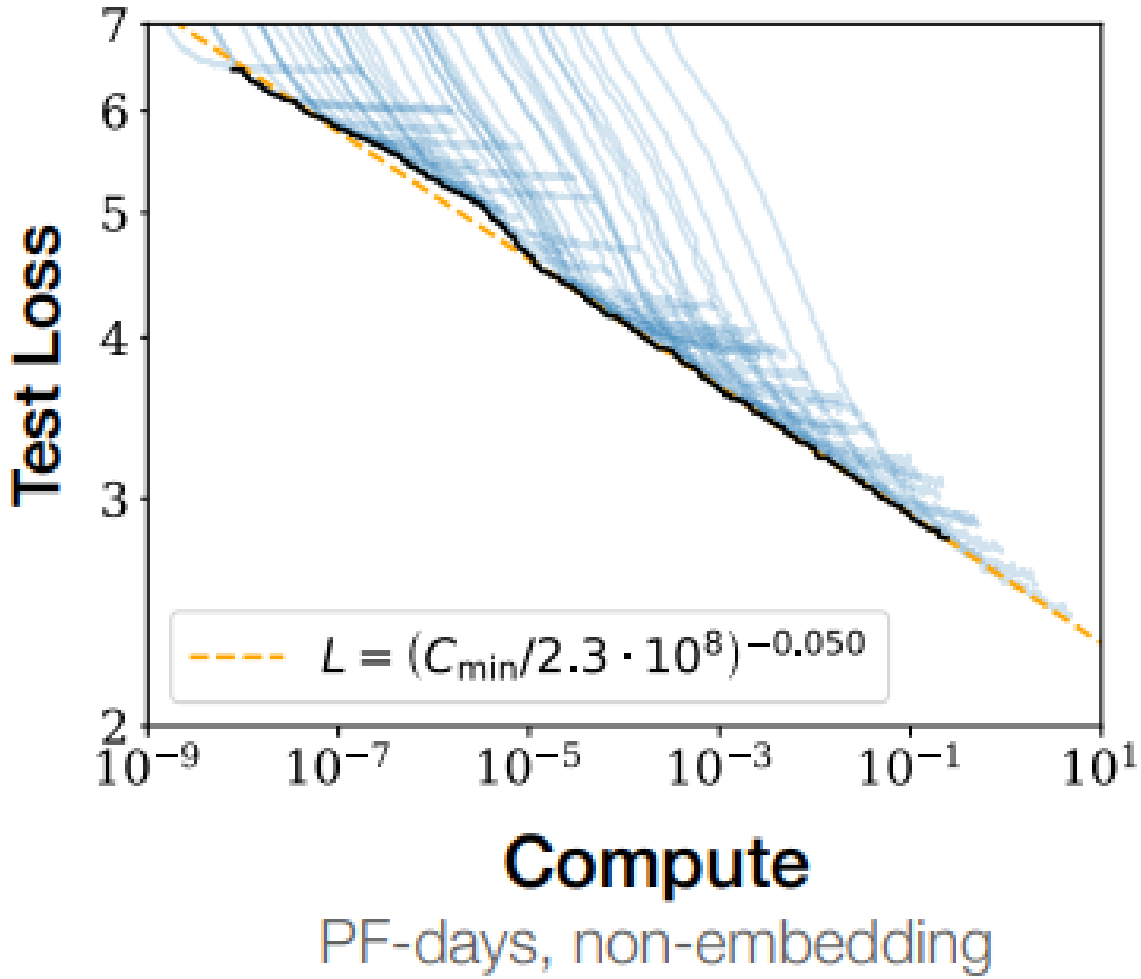
When training within a fixed compute budget C, but with no other constraints, optimal model size N, optimal batch size B, optimal number of steps S, and dataset size D should grow as:

$$N \propto C^{\alpha_C^{min}/\alpha_N}, \quad B \propto C^{\alpha_C^{min}/\alpha_B}, \quad S \propto C^{\alpha_C^{min}/\alpha_S}, \quad D = B \times S$$

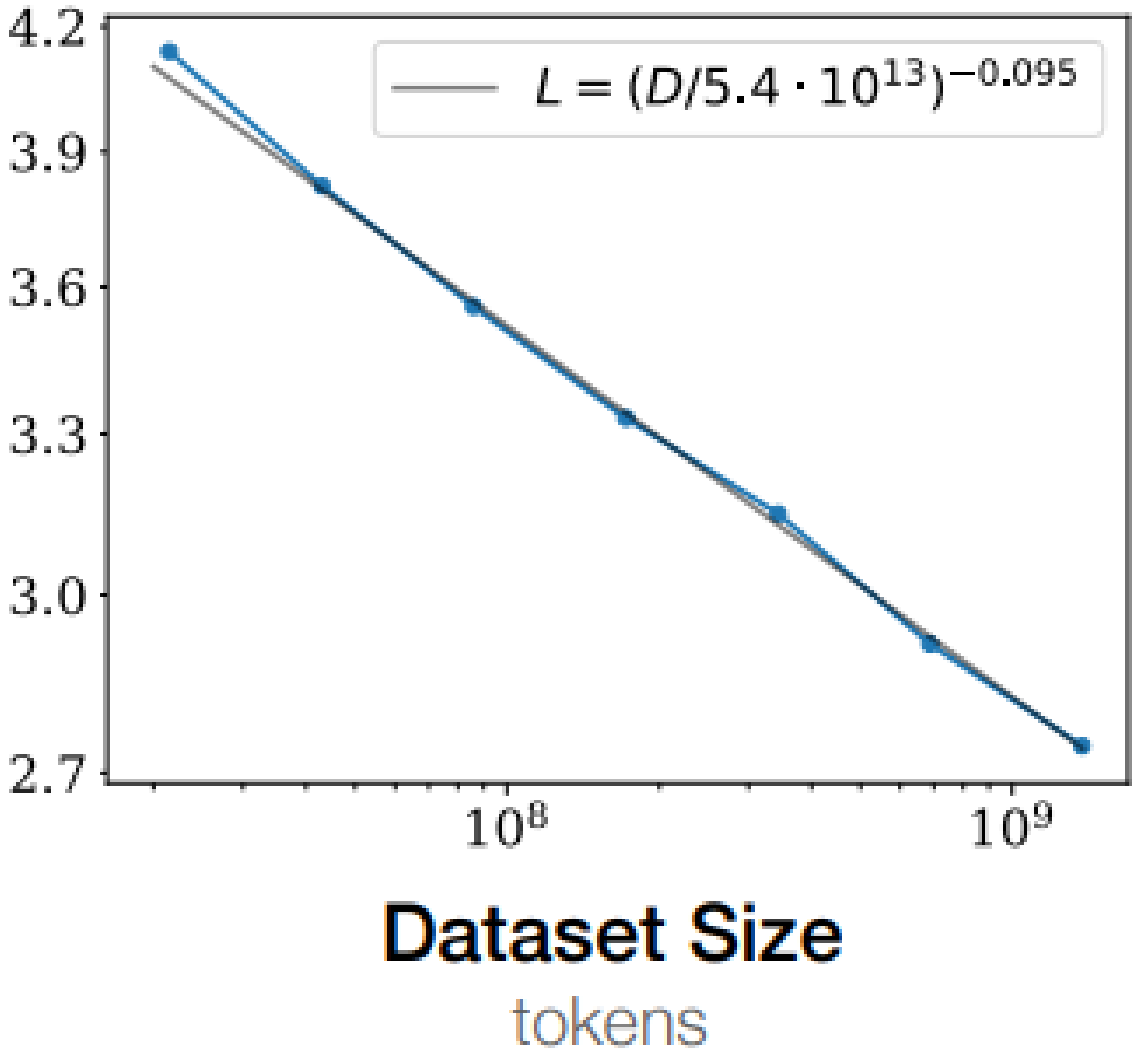
$$\alpha_C^{min} = \frac{1}{\frac{1}{\alpha_N} + \frac{1}{\alpha_B} + \frac{1}{\alpha_S}}$$

Some interesting visualization

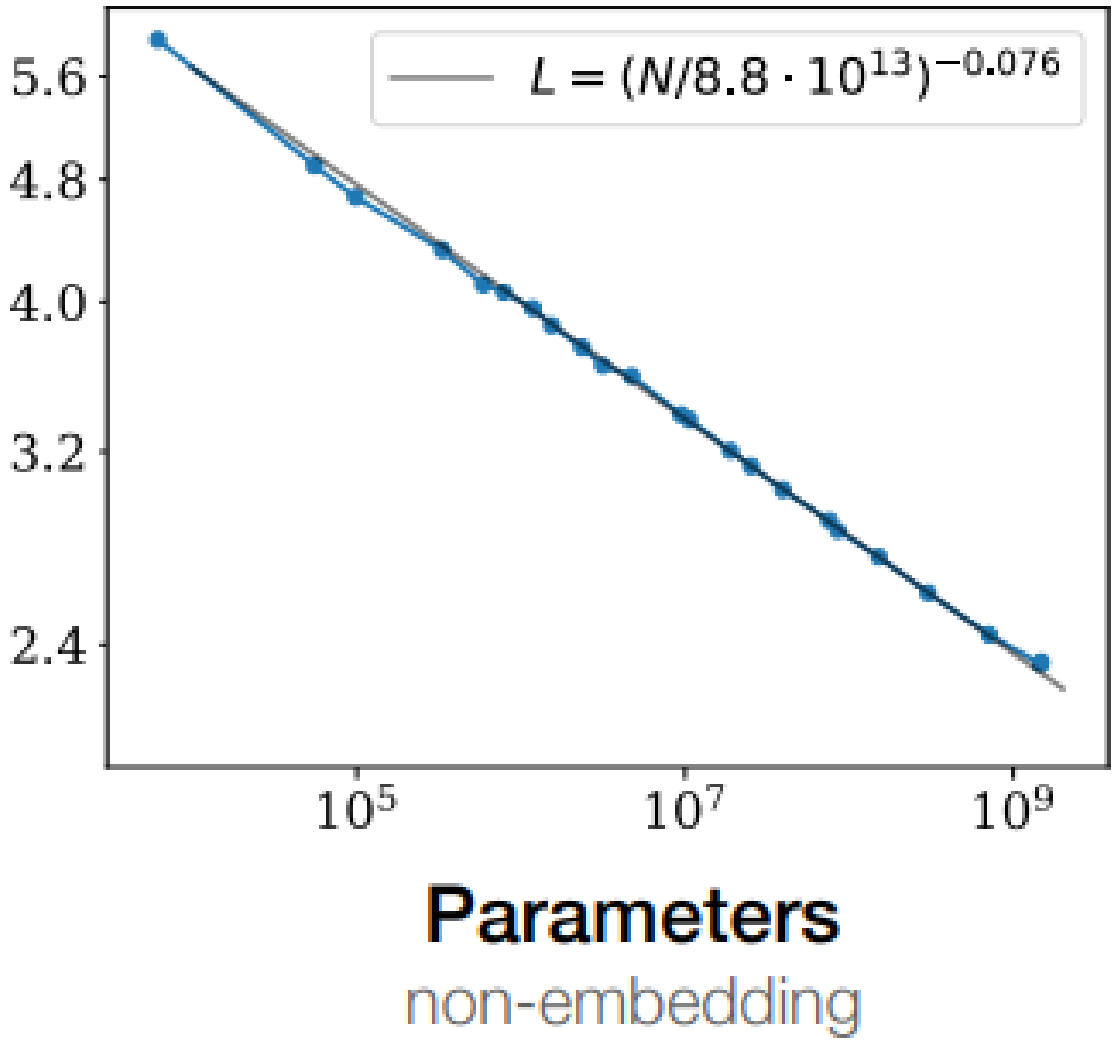
Loss vs C


Test Loss
7
6
5
4
3
2
10^{-9}
10^{-7}
10^{-5}
10^{-3}
10^{-1}
10^{1}
$L = (C_{\text{min}}/2.3 \cdot 10^{8})^{-0.050}$
Compute
PF-days, non-embedding

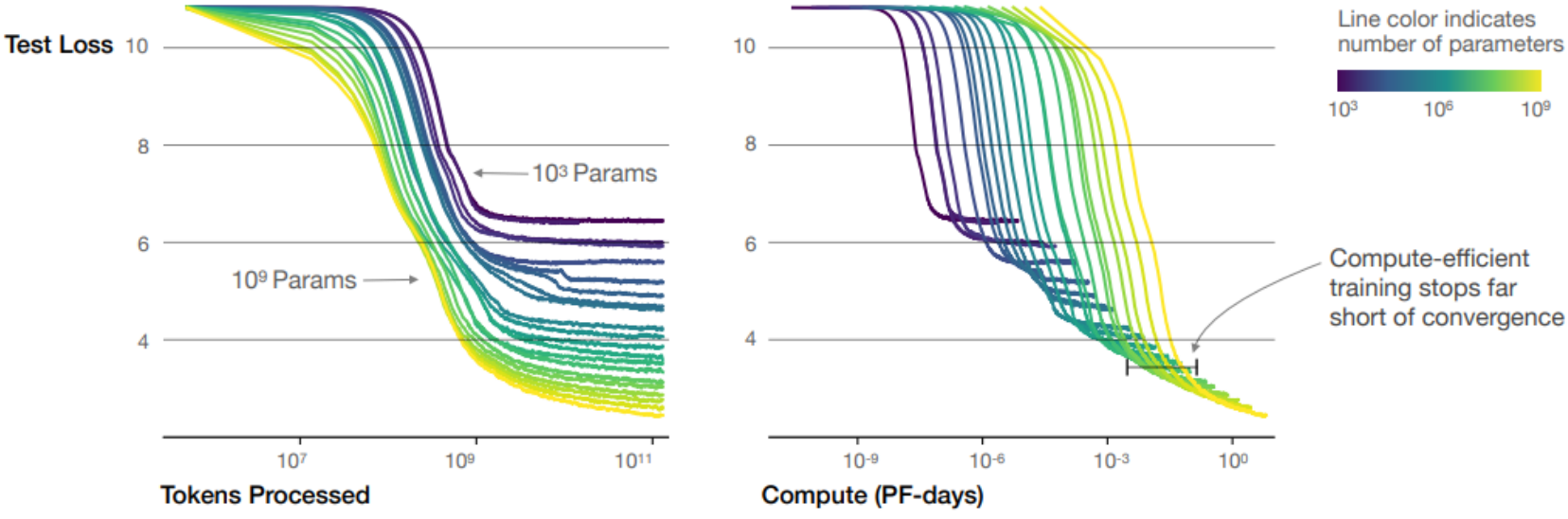
Loss vs D


$L = (D/5.4 \cdot 10^{13})^{-0.095}$
4.2
3.9
3.6
3.3
3.0
2.7
10^8
10^9
Dataset Size
tokens

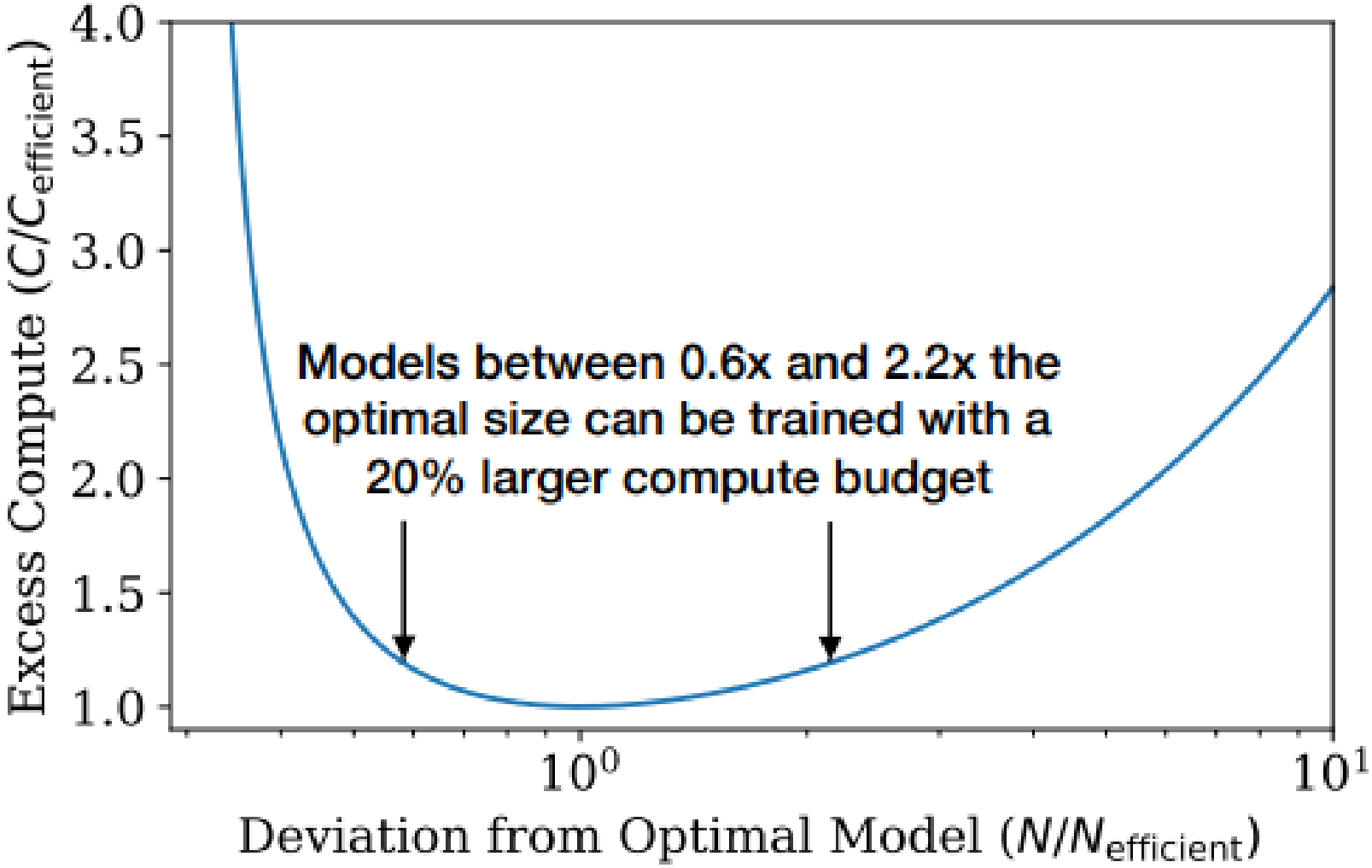
Loss vs N


$L = (N/8.8 \cdot 10^{13})^{-0.076}$
5.6
4.8
4.0
3.2
2.4
10^5
10^7
10^9
Parameters
non-embedding

Sample efficiency



Optimal model size



Questions

Questions

- Запишите любые три ключевых вывода, которые можно сделать из исследования.
- Запишите функцию зависимости ошибки на тестовой выборке от размера модели и размера датасета.
- Допустим мы знаем, что оптимальный размер модели, которая не переобучается на датасете размера D_1 равен N_1. Запишите оценку на размер модели на датасете размера D_2, если мы не хотим переобучаться?

References

References

1. Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, Dario Amodei. Scaling Laws for Neural Language Models. In ArXiv 2020