Hyperparameters

Devon

ML4Good Germany 2024

What distinguishes a hyperparameter from a parameter?

What distinguishes a hyperparameter from a parameter?

A regular parameter is *optimized during the training process*, e.g. during gradient descent

What distinguishes a hyperparameter from a parameter?

A regular parameter is *optimized during the training process*, e.g. during gradient descent

A *hyperparameter* is something we choose *before we start training*

Another way to think about it

$$\theta_{t+1} = \theta_t - \alpha \cdot \nabla_\theta \text{Loss}(\theta_t)$$

(the equation for gradient descent)

Another way to think about it

$$\theta_{t+1} = \theta_t - \alpha \cdot \nabla_\theta \text{Loss}(\theta_t)$$

Here, parameters are what we differentiate the loss with respect to

Another way to think about it

$$\theta_{t+1} = \theta_t - \alpha \cdot \nabla_\theta \text{Loss}(\theta_t)$$

But we could not differentiate the loss w.r.t. hyperparameters like the learning rate

What hyperparameters showed up in the prerequisites?

What hyperparameters showed up in the prerequisites?

Learning rate

Number of epochs

Degree of polynomial

(there could also have been others)

How do we actually choose them?

How do we actually choose them?

Usually... search!

Different strategies we won't get into...
but a lot of it is guess and check

How do we actually choose them?

Usually... search!

Different strategies we won't get into...
but a lot of it is guess and check

(Although there is some theory here)

A note about hyperparameters...

A *hyperparameter* is something we choose
before we start training

A note about hyperparameters...

A *hyperparameter* is something we choose *before we start training*

So: every loop of a typical hyperparameter search involves a whole training run!

Why do we care about hyperparameters?

1: They often matter a lot in ML, and they frequently make the difference between success and failure

Link


epochs
dropout
batch_size
learning_rate
accuracy
10
9
8
7
6
5
4
3
2
1
0.50
0.45
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
600
550
500
450
400
350
300
250
200
150
100
50
0
0.10
0.09
0.08
0.07
0.06
0.05
0.04
0.03
0.02
0.01
0.00
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1

Note that we don’t just care about which hyperparameters get the best accuracy / loss!

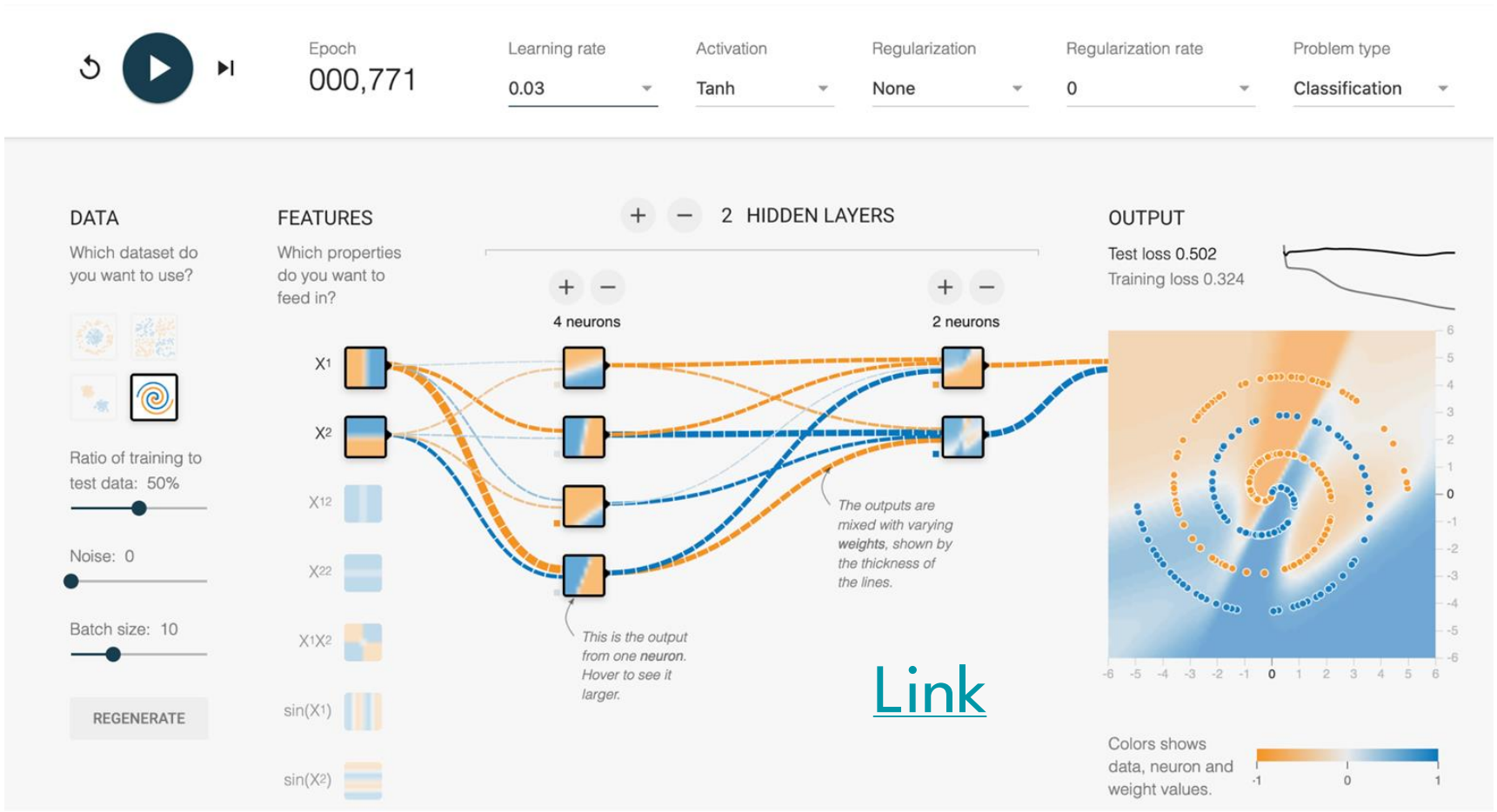
Note that we don't just care about which hyperparameters get the best accuracy / loss!

Some choices will also be more expensive to train or to run later on.

Why do we care about hyperparameters?

2: Getting them right is a significant (and expensive) part of what AI companies work on (and a significant part of their IP)

(We will come back to this)


Epoch
000,771
Learning rate
0.03
Activation
Tanh
Regularization
None
Regularization rate
0
Problem type
Classification
DATA
Which dataset do you want to use?
Ratio of training to test data: 50%
Noise: 0
Batch size: 10
REGENERATE
FEATURES
Which properties do you want to feed in?
X1
X2
X12
X22
X1X2
sin(X1)
sin(X2)
2 HIDDEN LAYERS
4 neurons
2 neurons
This is the output from one neuron. Hover to see it larger.
The outputs are mixed with varying weights, shown by the thickness of the lines.
OUTPUT
Test loss 0.502
Training loss 0.324
Colors shows data, neuron and weight values.
-1
0
1
Link

[Notebook with pair programming]

[Show solutions notebook and W&B visuals]

Why do we care about hyperparameters?

2: Getting them right is a significant (and expensive) part of what AI companies work on (and a significant part of their IP)

Scaling Laws for Neural Language Models

Jared Kaplan *
Johns Hopkins University, OpenAI
jaredk@jhu.edu

Sam McCandlish*
OpenAI
sam@openai.com

Tom Henighan
OpenAI
henighan@openai.com

Tom B. Brown
OpenAI
tom@openai.com

Benjamin Chess
OpenAI
bchess@openai.com

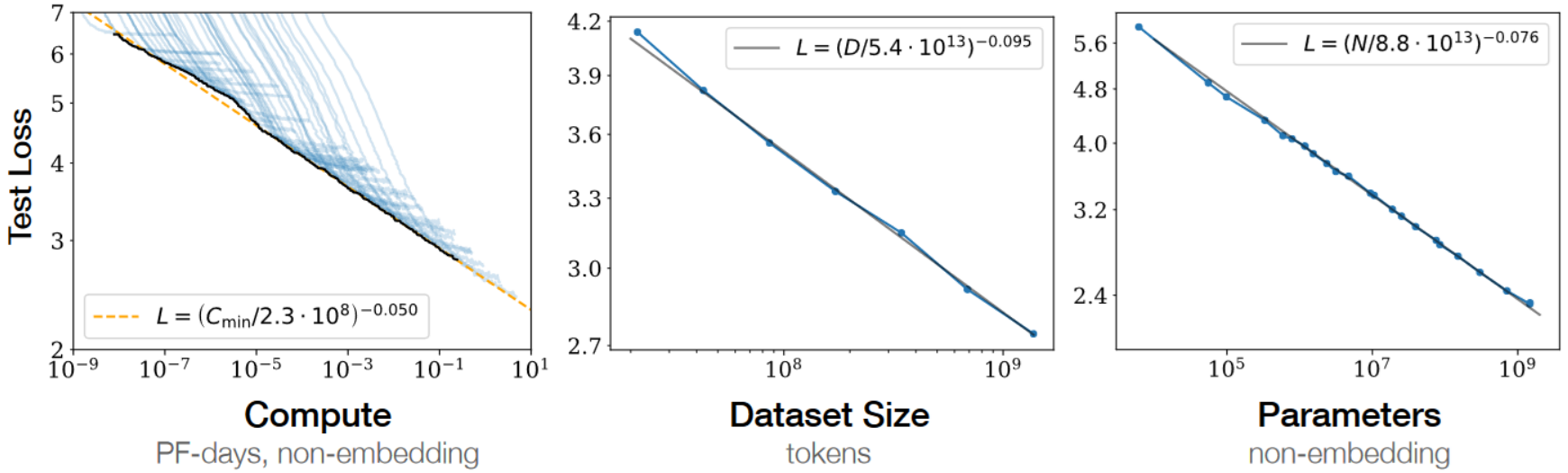
Rewon Child
OpenAI
rewon@openai.com

Scott Gray
OpenAI
scott@openai.com

Alec Radford
OpenAI
alec@openai.com

Jeffrey Wu
OpenAI
jeffwu@openai.com

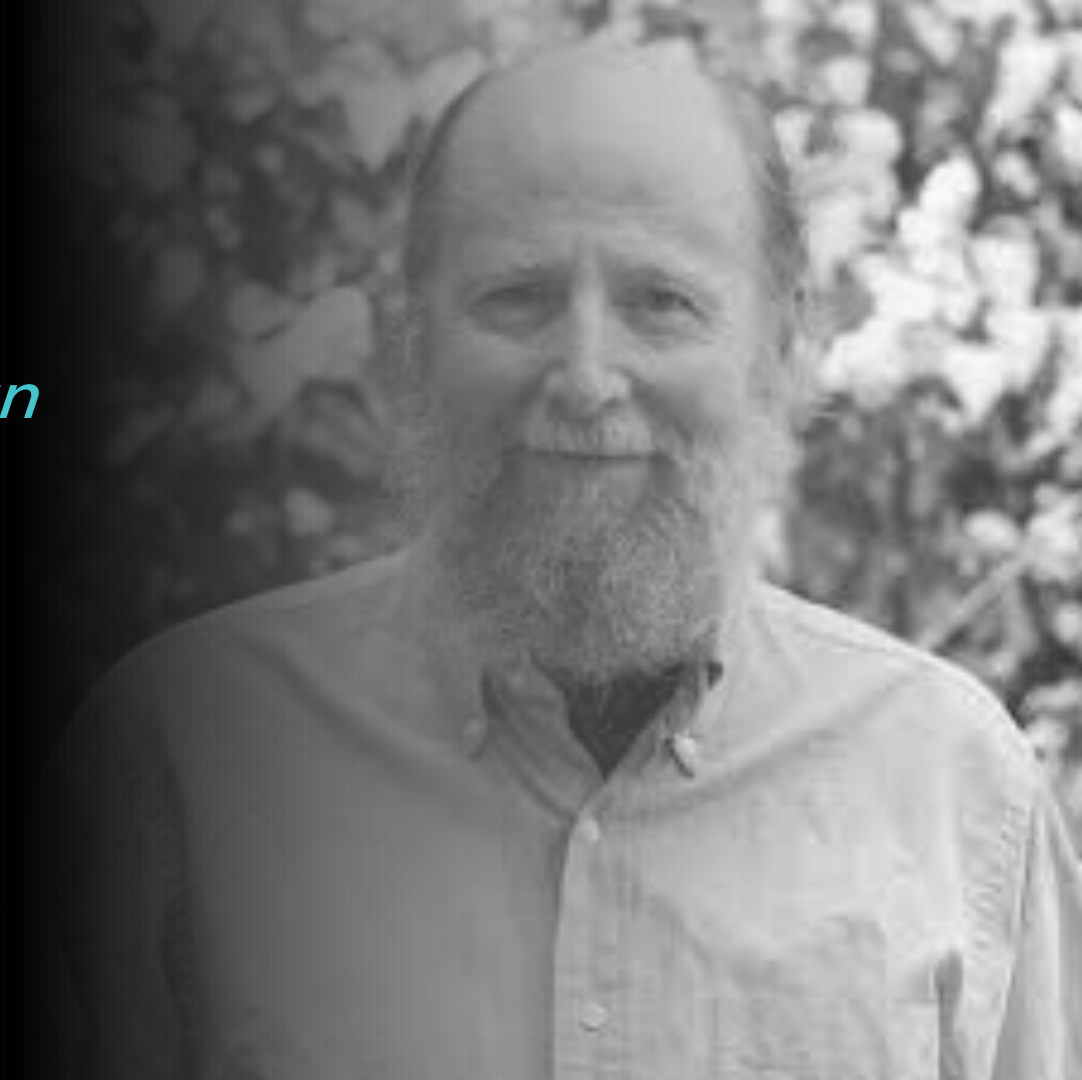
Dario Amodei
OpenAI
damodei@openai.com

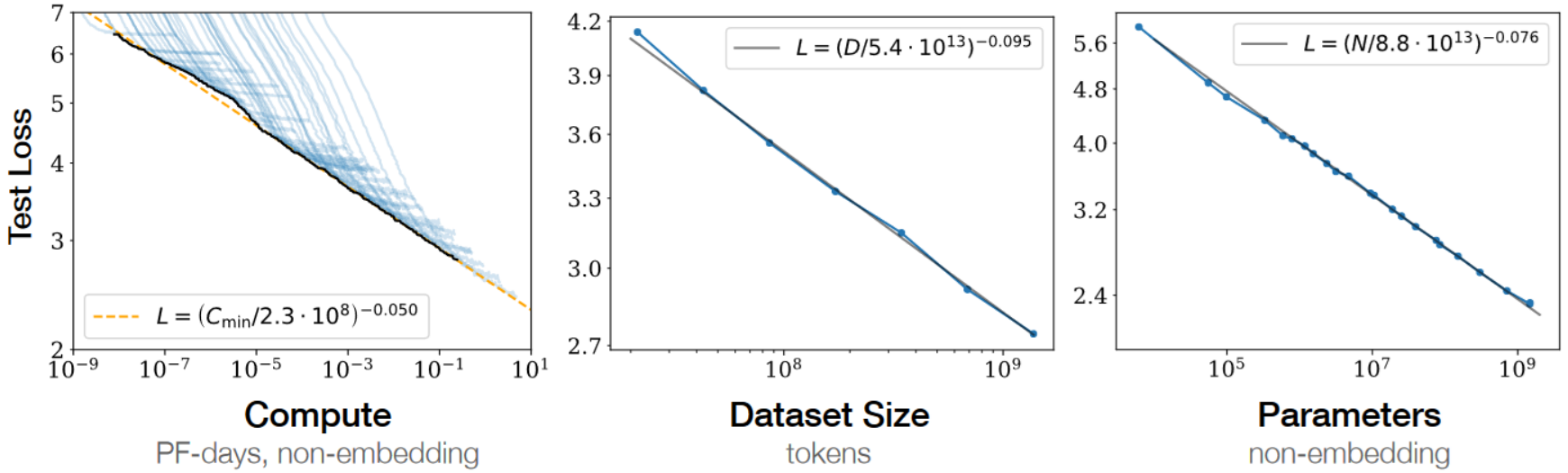


Kaplan, J., McCandlish, S., Henighan, T., Brown, T. B., Chess, B., Child, R., Gray, S., Radford, A., Wu, J., & Amodei, D. (2020). Scaling laws for neural language models [Technical report]. OpenAI.

The only thing that matters in the long run is the ***leveraging of computation.***

Rich Sutton





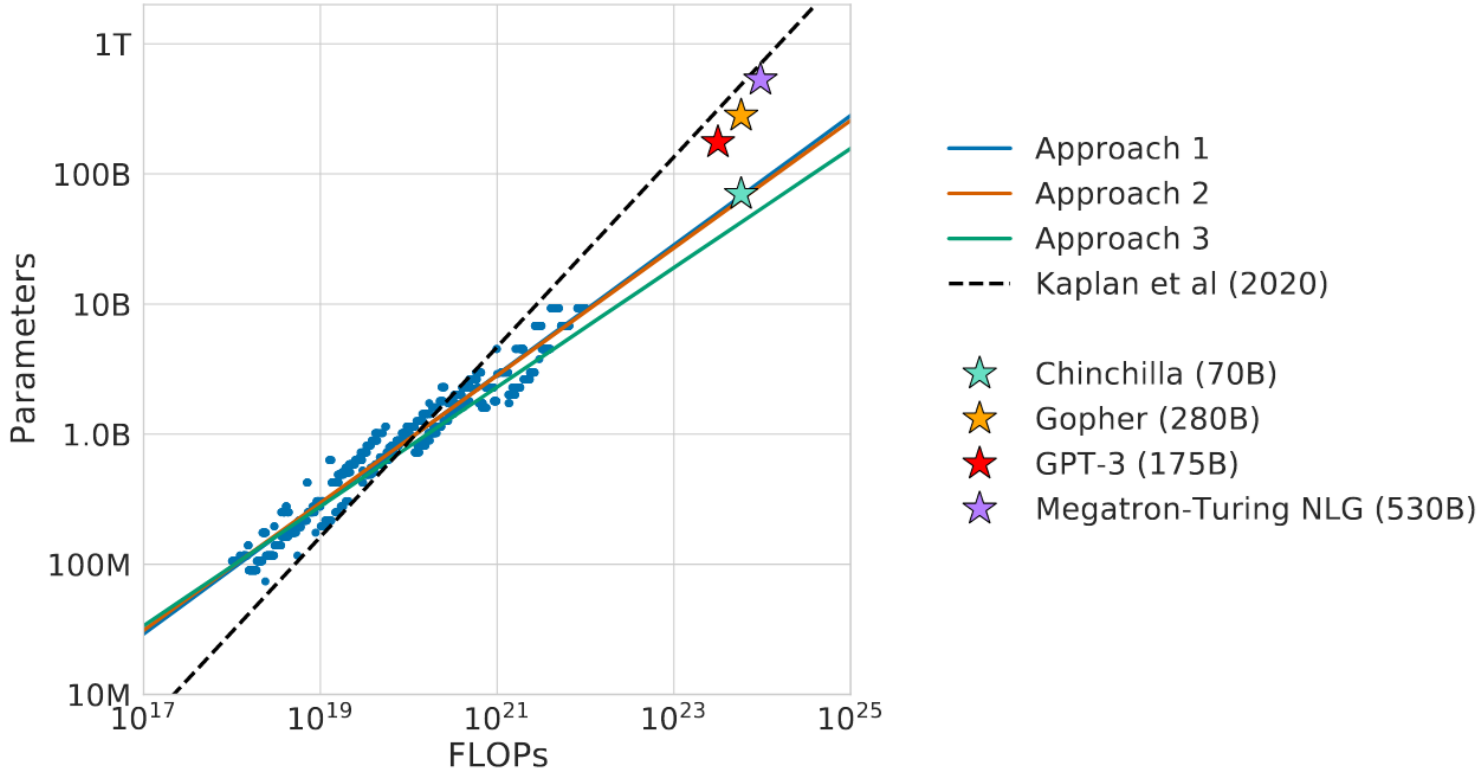
Kaplan, J., McCandlish, S., Henighan, T., Brown, T. B., Chess, B., Child, R., Gray, S., Radford, A., Wu, J., & Amodei, D. (2020). Scaling laws for neural language models [Technical report]. OpenAI.

DeepMind

Training Compute-Optimal Large Language Models

Jordan Hoffmann*, Sebastian Borgeaud*, Arthur Mensch*, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Jack W. Rae, Oriol Vinyals and Laurent Sifre*

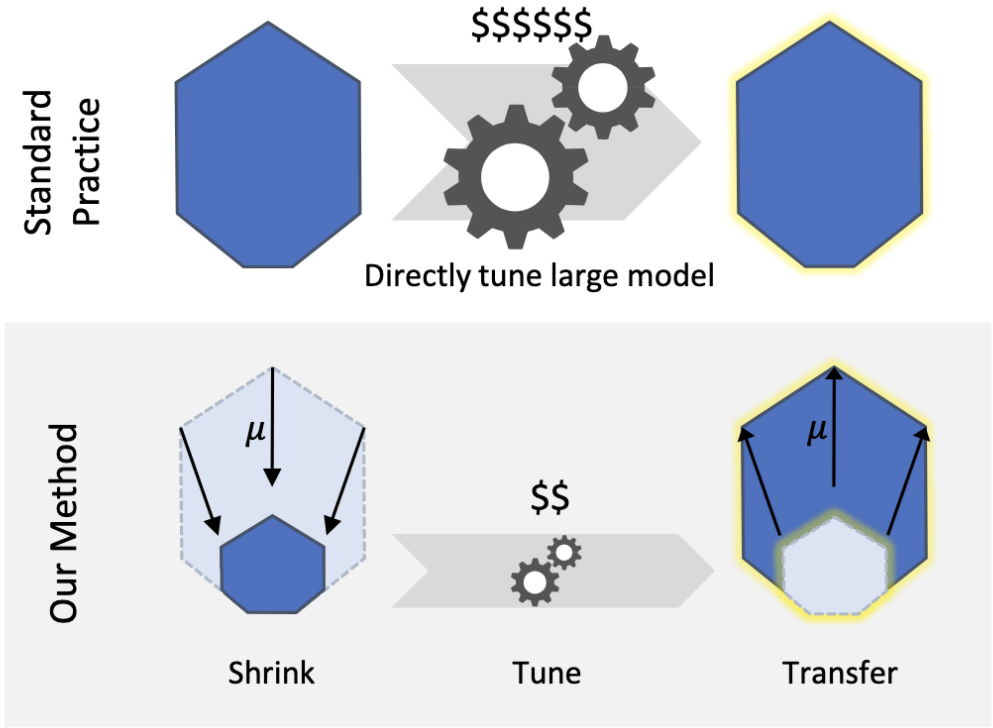
*Equal contributions



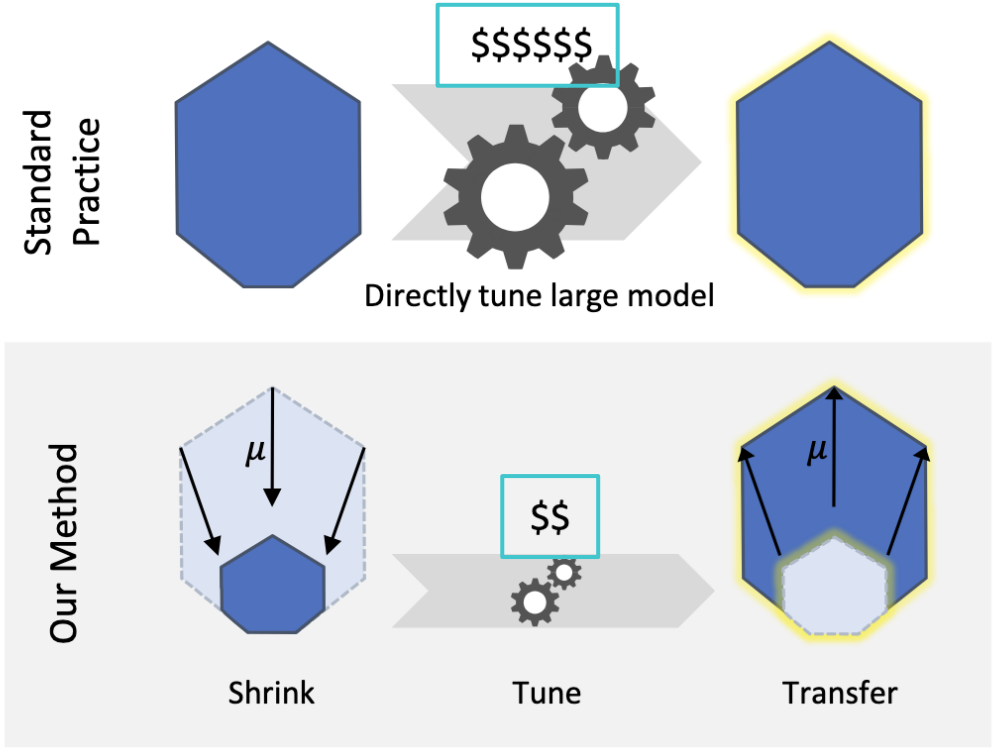
Hoffmann et al. (2022). Training compute-optimal large language models [Technical report]. DeepMind.

Tensor Programs V: Tuning Large Neural Networks via Zero-Shot Hyperparameter Transfer

Greg Yang[*,×] **Edward J. Hu**[*,×,†] **Igor Babuschkin**[°] **Szymon Sidor**[°] **Xiaodong Liu**[×]
David Farhi[°] **Nick Ryder**[°] **Jakub Pachocki**[°] **Weizhu Chen**[×] **Jianfeng Gao**[×]
[×]Microsoft Corporation [°]OpenAI



Yang et al. (2022). Tensor Programs V: Tuning large neural networks via zero-shot hyperparameter transfer [Technical report]. Microsoft, OpenAI.



Yang et al. (2022). Tensor Programs V: Tuning large neural networks via zero-shot hyperparameter transfer [Technical report]. Microsoft, OpenAI.