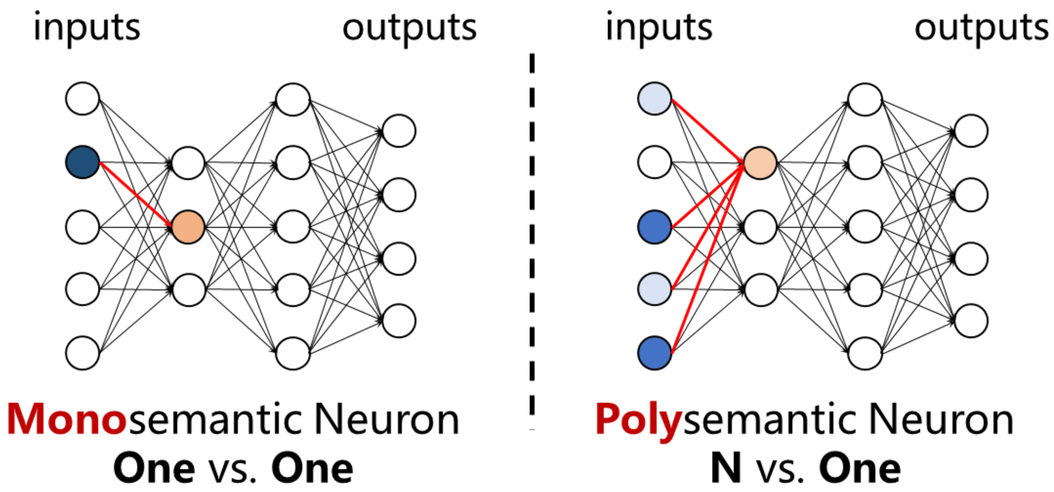
THE HONG KONG UNIVERSITY OF SCIENCE AND TECHNOLOGY

THE HONG KONG UNIVERSITY OF SCIENCE AND TECHNOLOGY (GUANGZHOU)

KDD2024 BARCELONA, SPAIN

Learning from Emergence: A Study on Proactively Inhibiting the Monosemantic Neurons of Artificial Neural Networks



—— Is **Inhibiting monosemanticity** a **new research direction** toward better performance**?**

Dr. Jiachuan WANG, Dr. Shimin DI, Prof. Lei CHEN, Prof. Charles Wang Wai Ng

Contact us: dishimin@ust.hk

The Hong Kong University of Science and Technology

The Hong Kong University of Science and Technology (Guangzhou)

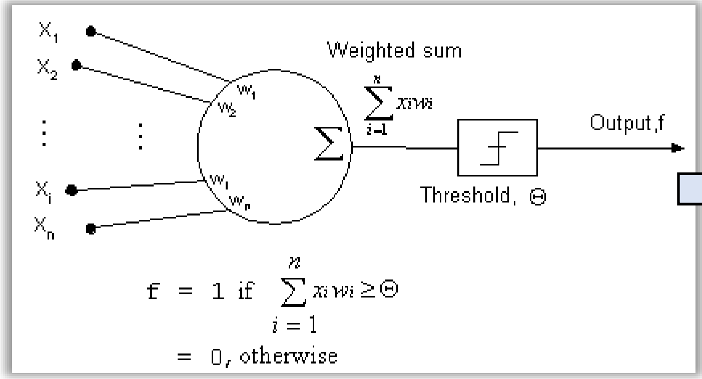
OUTLINE

- Background
- Motivation
- Method
- Experiment

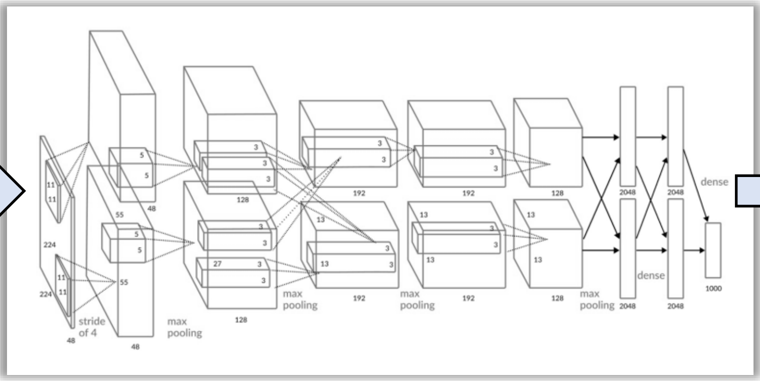
Background

Artificial Neural Networks

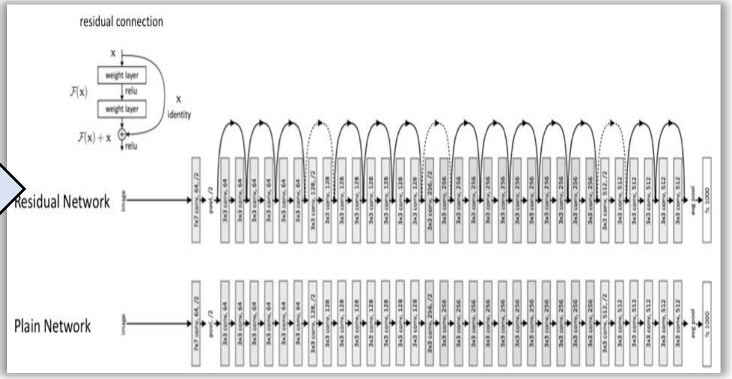
1943, Artificial Neuron



2012, AlexNet



2015, ResNet

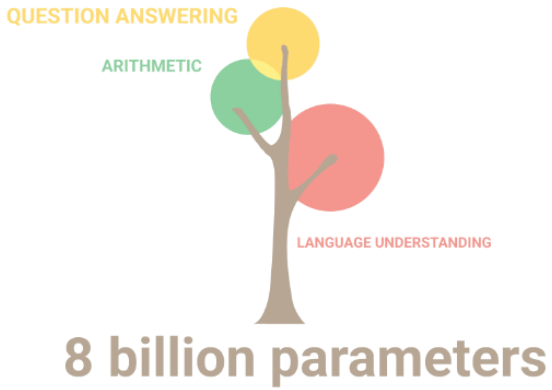


A large increase in scale!

Background

Outstanding of Large Language Models (LLM) – Emergence

- **Emergence:** just increase the scale, abilities will emerge!



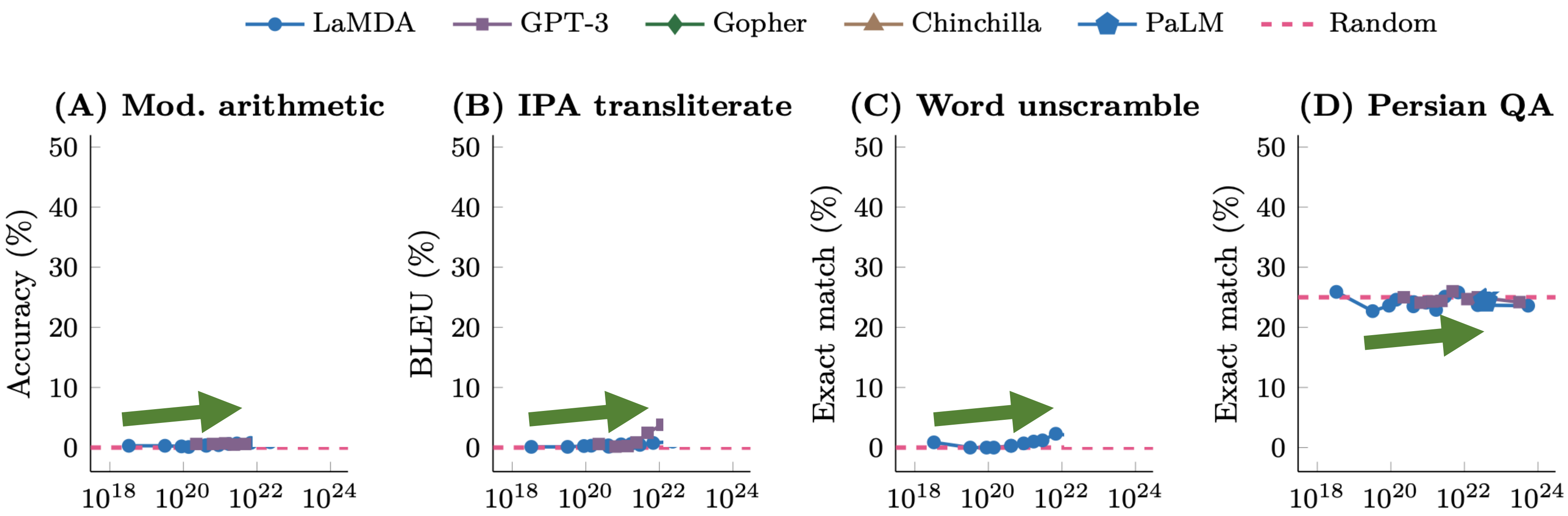
Refer to: Pathways Language Model (PaLM): Scaling to 540 Billion Parameters for Breakthrough Performance

Background

Outstanding of Large Language Models (LLM) – Emergence

□ Emergence:

the **scale** not reach a certain threshold → **gradual** improvement



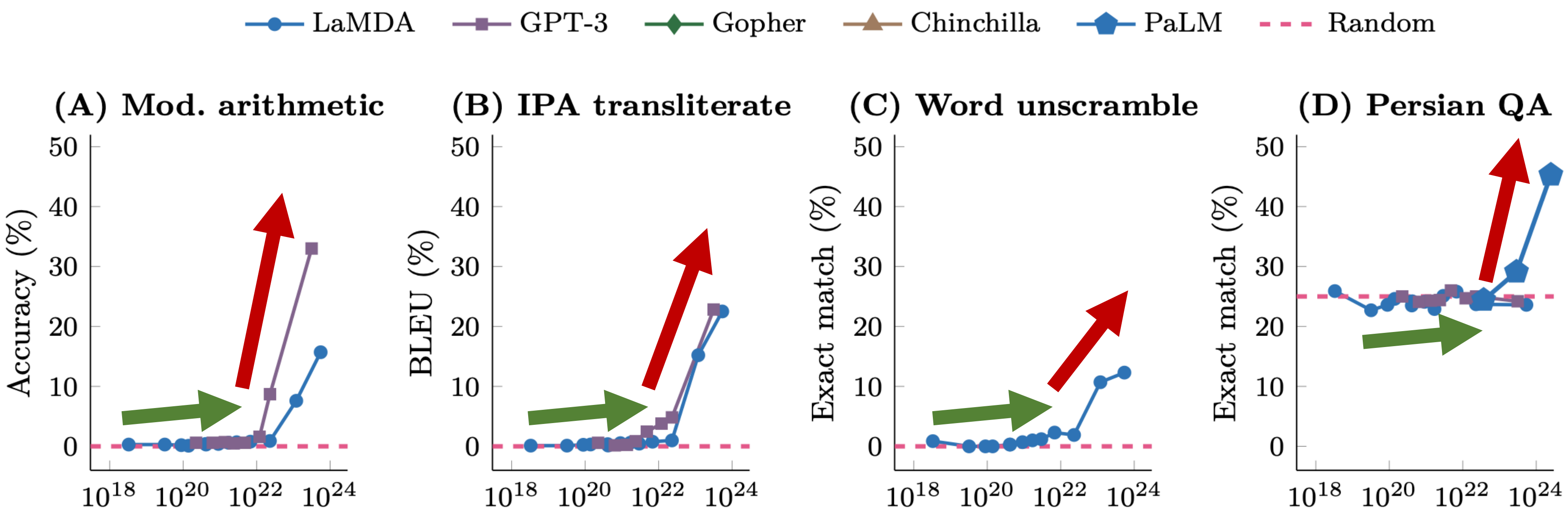
Background

Outstanding of Large Language Models (LLM) – Emergence

- Emergence:

the **scale** not reach a certain threshold → **gradual** improvement

the **scale** surpasses a certain threshold → **rapid** enhancement



Background

Outstanding of Large Language Models (LLM) – Emergence

- Emergence:

the **scale** not reach a certain threshold → **gradual** improvement

the **scale** surpasses a certain threshold → **rapid** enhancement

One interesting question:

People **increase** the model **scale** and get better results,
but **what** has changed underlying the process?

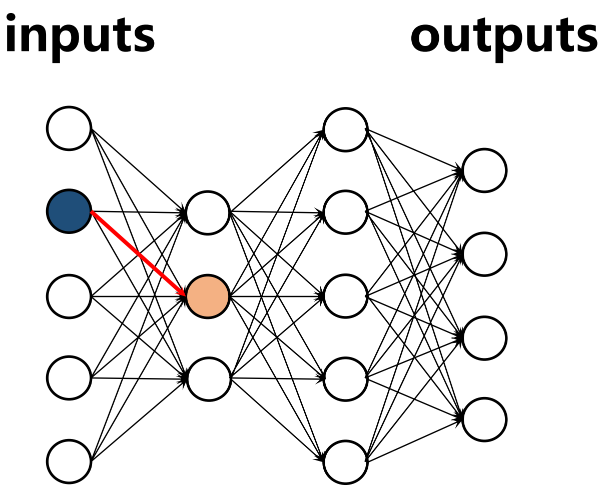
OUTLINE

- Background
- Motivation
- Method
- Experiment

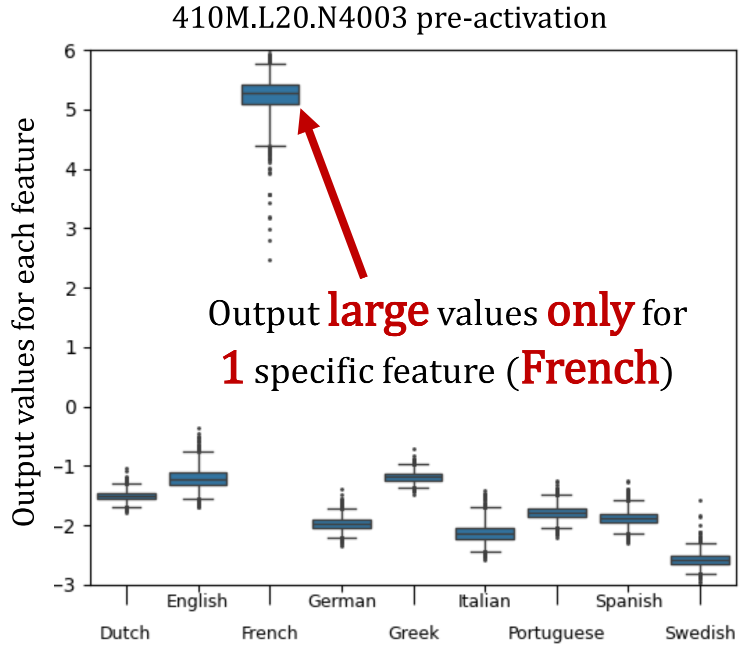
Motivation

Interpreting Emergence

- Pioneer works interpret the performance of small and large-scale models from the **correlation** between neurons and input features.



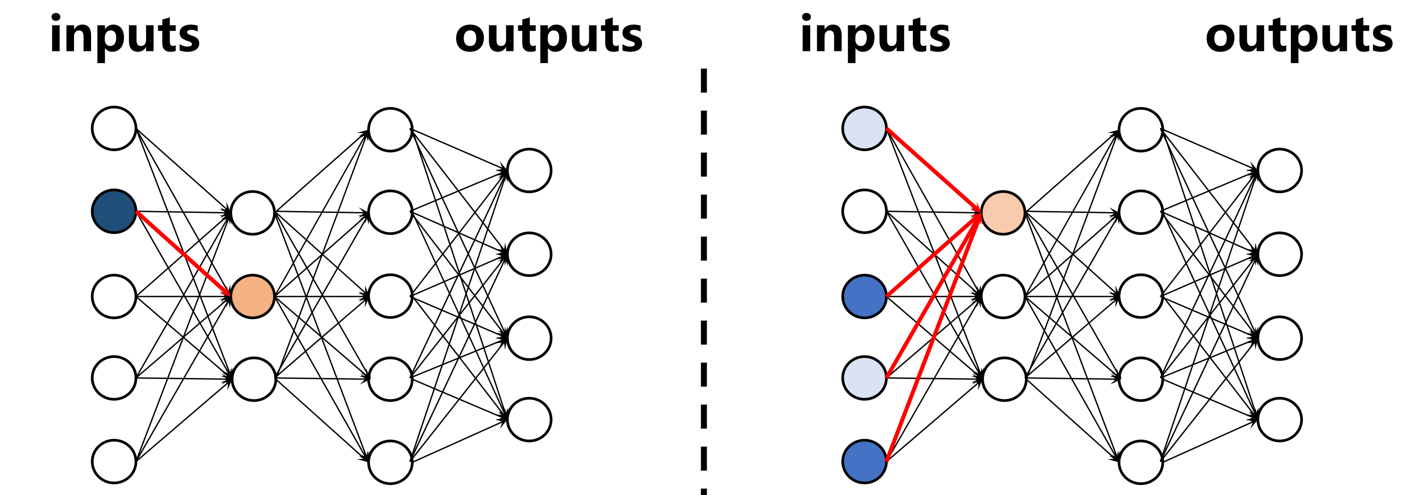
Monosemantic Neuron
One vs. One



Motivation

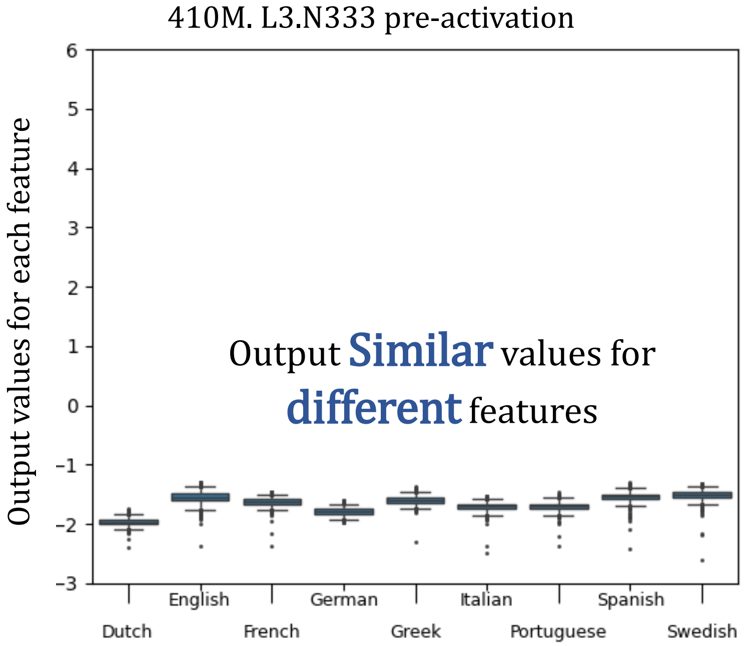
Interpreting Emergence

- Pioneer works interpret the performance of small and large-scale models from the **correlation** between neurons and input features.



Monosemantic Neuron
One vs. One

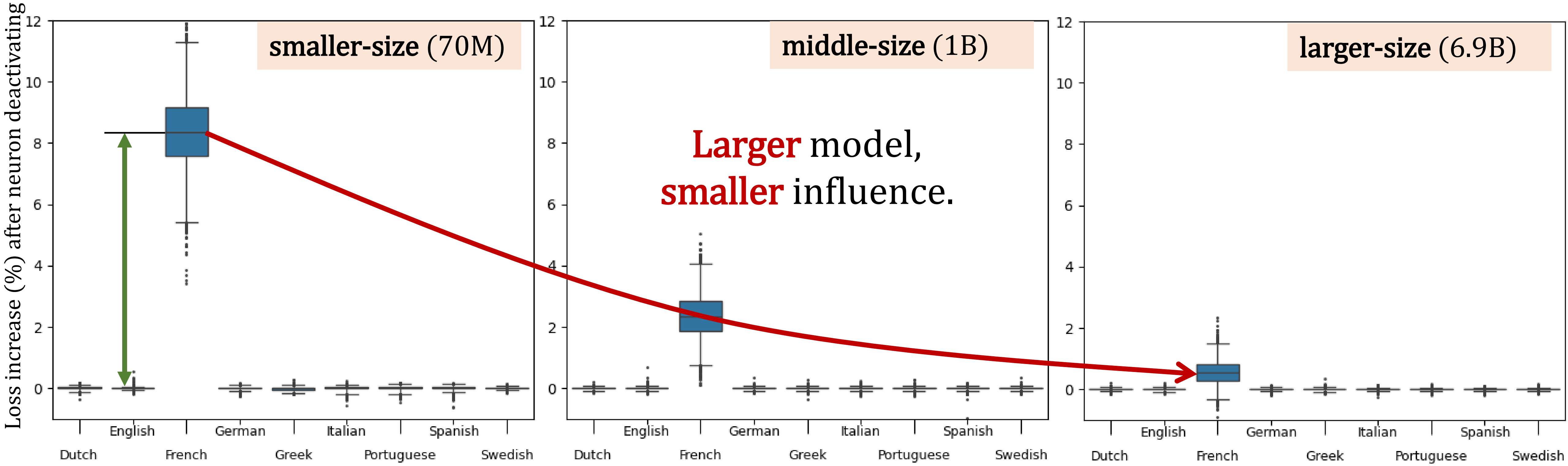
Polysemantic Neuron
N vs. One



Motivation

Motivational Experiments

- **Larger models have lower monosemanticity!**
  - Turning off monosemantic neurons, a larger model has **smaller** error increase.

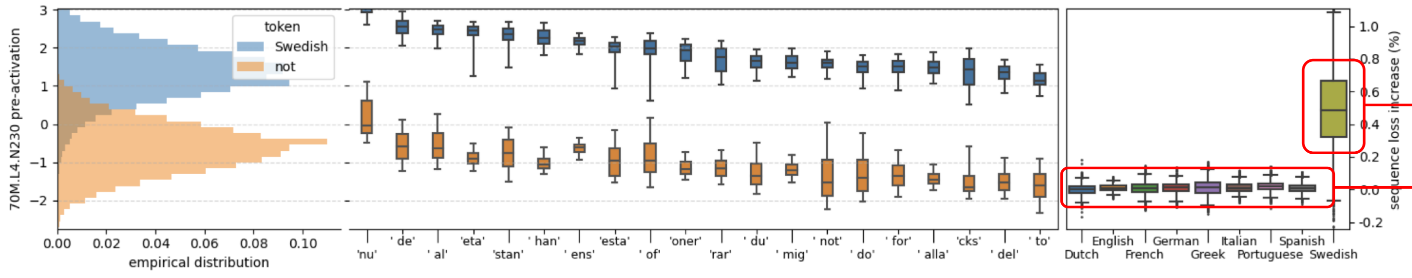


Motivation

Motivational Experiments

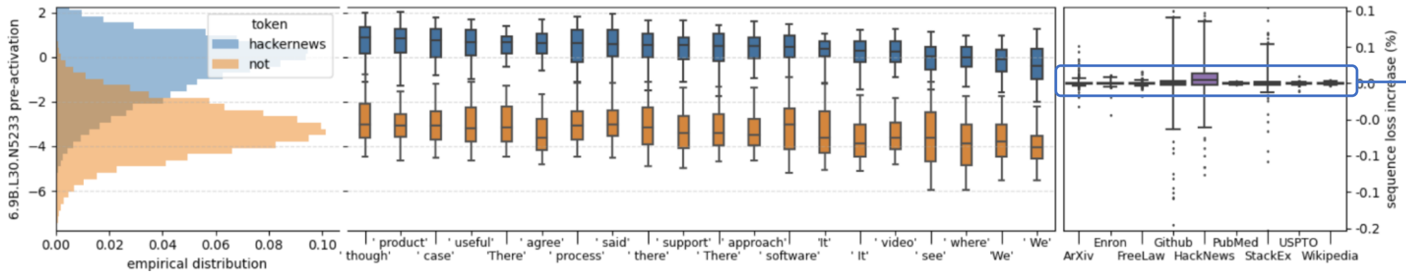
- **Larger models have lower monosemanticity!**
  - Given the corresponding/non-corresponding features, the difference in activation values of large models is smaller than that of small models

Pythia-70M



Large difference between the corresponding and non-corresponding features

Pythia-6.9B



Small difference

Motivation

Motivational Experiments

- **Larger models have lower monosemanticity!**
  - Turning off monosemantic neurons, a larger model has **smaller** error increase.
  - Given the corresponding/non-corresponding features, the difference in activation values of large models is smaller than that of small models

- **Assumption**
  - The **decrease** in **mono**semanticity may be a key factor in achieving **higher** performance as the model **scale increases**.

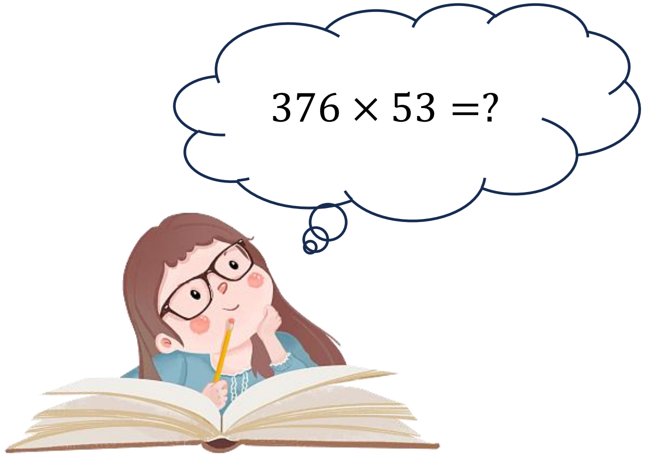
Motivation

Motivational Examples

- Assumption: The **decrease** in monosemanticity may be a key factor in achieving **higher** performance as the model **scale increases**.

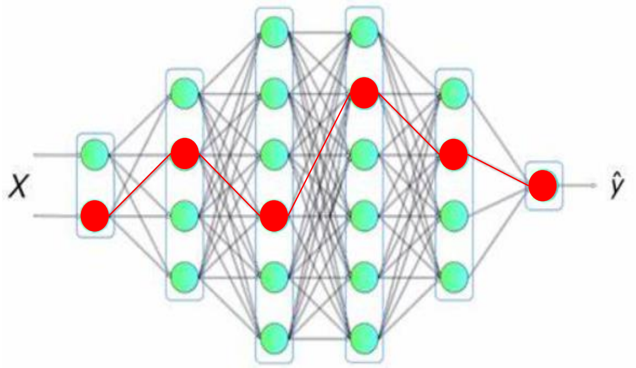
A student **memorizes** questions and answers for short-term gain.
As the amount of learning increases, understand the problem inefficiently.

Train ANNs with the observed training examples **repeatedly**.
As the amount of training increases, slowly reduce the monosemantic neurons.



$376 \times 53 = 19928$
$376 \times 53 = 19928$
$376 \times 53 = 19928$
$376 \times 53 = 19928$
$376 \times 53 = 19928$

memorize repeatedly
train repeatedly



$\Rightarrow 376 \times 53 = 19928$

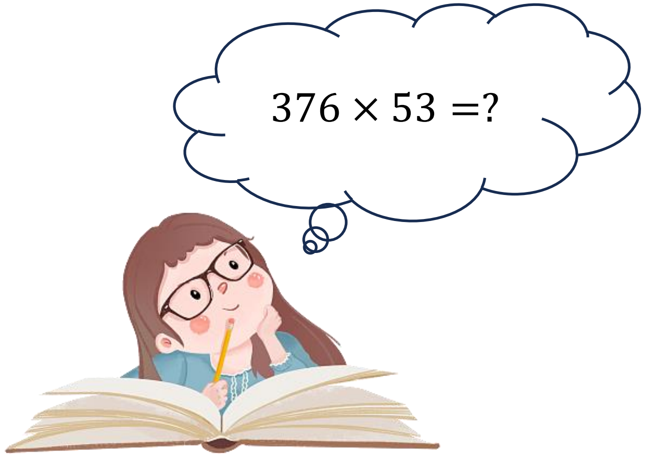
Motivation

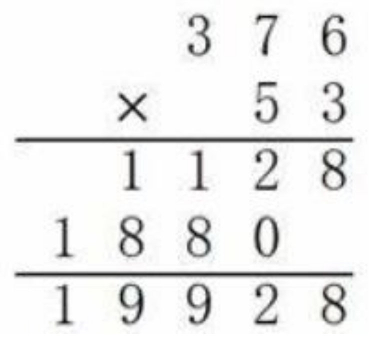
Motivational Examples

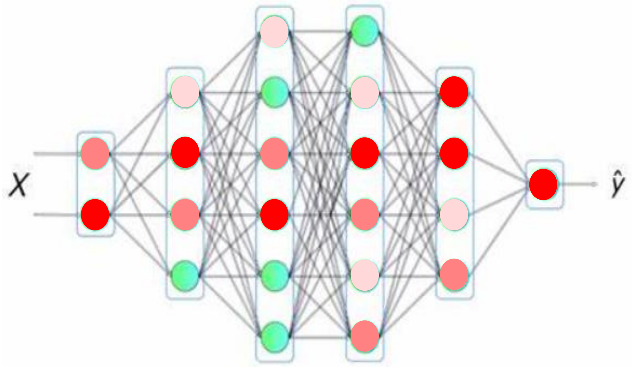
- Assumption: The **decrease** in monosemanticity may be a key factor in achieving **higher** performance as the model **scale increases**.

The student is expected to **dismantle** the problem and integrate the knowledge points, and achieve the final answer via **reasoning**.

The large model **disassembles** the training inputs, maps the features of samples to multiple neurons, integrates the neurons, and the output “**emerges**”!







$\Rightarrow 376 \times 53 = 19928$

Motivation

Motivational Experiments from Literature

- We rather conclude the current paradigm of training neural networks as a **passive** process in decreasing monosemantic neurons.

Rote Learning → One (question) to One (answer) ⟺ One (feature) to One (neuron) ← **Monosemantic** Neurons

Heuristic Learning → Understanding and Reasoning ⟺ N to One / One to N ← Polysemantic Neurons / Distributed Features

- Inspired by the emergence, we propose one question:

Can we ***proactively*** ***inhibit monosemantic neurons***

in artificial neural networks to achieve high performance?

Motivation

Technical Challenges: Monosemantic Neuron Detection

- ◻ Existing detection has limitations and high computational overhead
  - ◻ **Limitation:** require to calculate on **manually designed and labeled** feature data sets.
  - ◻ **High Computational Overhead:** Probes require training. And the calculation requires to **frequently** count the inputs to neurons and activation values from all neurons.

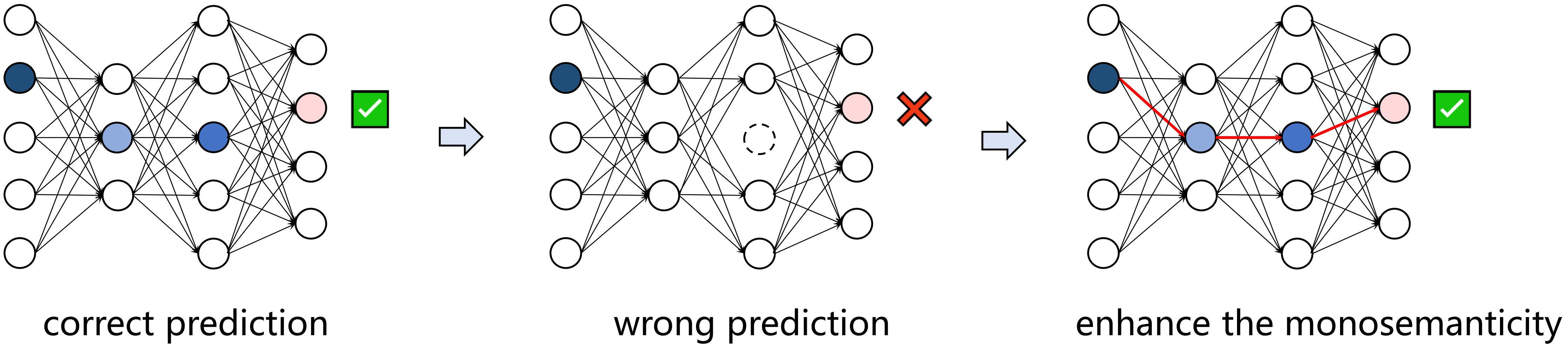
- ◻ Strictly defining monosemantic neurons is still under discussion in quantitative analysis.
  - ◻ **Generality:** Detection does not dependent on a specific data set.
  - ◻ **Efficiency:** Detect monosemantic neurons during online training.

Expected

Motivation

Technical Challenges: Monosemantic Neuron Inhibition

- Simply prohibiting the activation of monosemantic neurons will intensify the monosemanticity of artificial neural networks.



Motivation

Summary of Technical Contributions

We propose to **learn from emergence** to present a study on proactively inhibiting the monosemantic neurons of artificial neural networks.

- The Evaluation Metric for Detecting Monosemantic Neurons
  - **Data-specific evaluation** → A **quantitative** metric **does not** rely on data sets.
  - **Large** computational overhead → **Online** computation guarantee.
- The Proactive Deactivation Method to Reduce Monosemantic Neurons
  - **Hard** to deactivate → A **theoretically** supported method to suppress monosemantic neurons

OUTLINE

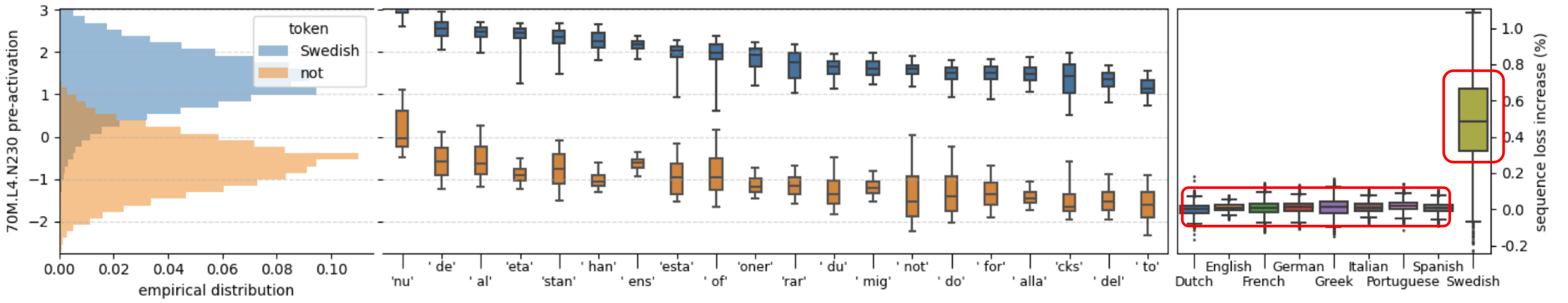
- Background
- Motivation
- **Method**
- Experiment

Method: Detection

Evaluation Measurement of Monosemantic Neurons

- Intuition: Design the metric $\phi(H)$ of evaluating monosemantic neurons from **low frequency of activation** and **high deviation of activation value**.
  - **Low frequency**: Existing work has divided hundreds of features, and the one-to-one nature determines that their activations are **sparse**.

Pythia-70M

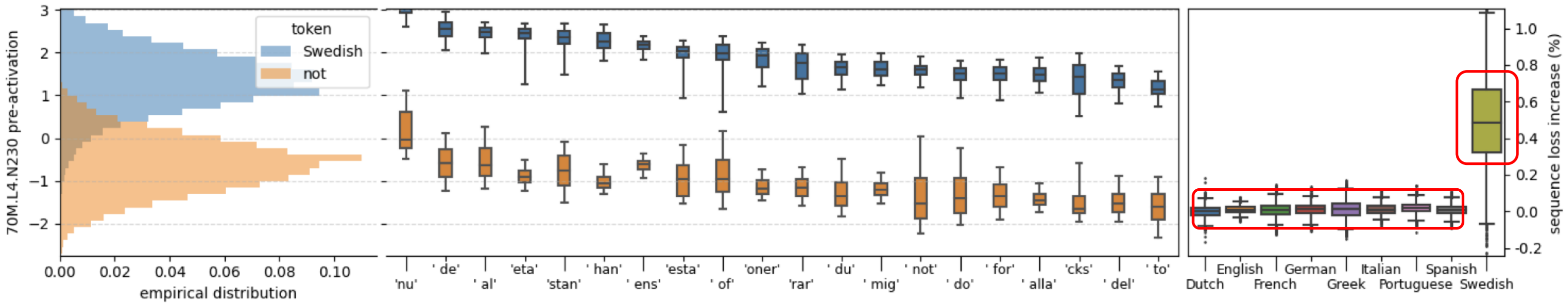


Method: Detection

Evaluation Measurement of Monosemantic Neurons

- Intuition: Design the metric $\phi(H)$ of evaluating monosemantic neurons from **low frequency of activation** and **high deviation of activation value**.
  - **High deviation**: The distribution after corresponding feature input **greatly deviates** from the overall distribution.

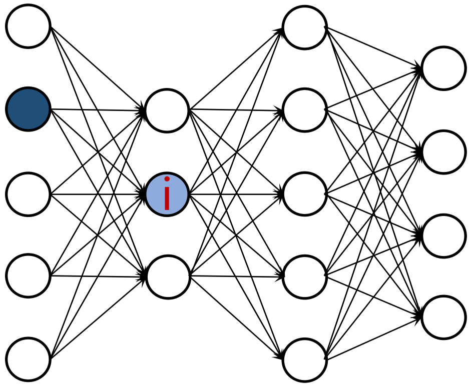
Pythia-70M



Method: Detection

Evaluation Measurement of Monosemantic Neurons

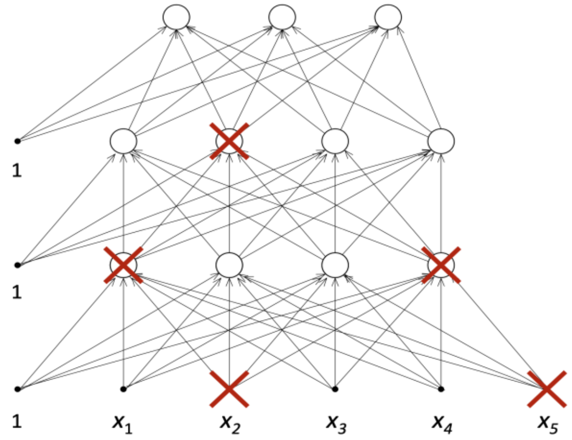
- Intuition: Design the metric $\phi(H)$ of evaluating monosemantic neurons from **low frequency of activation** and **high deviation of activation value**.
- But what is activation in our scenario? (**Another issue**)



i-th neuron at ℓ-th layer:

$$h_j^{\ell} = \sum_i w_{ij}^{\ell} z_i^{\ell-1},$$
$$z_i^{\ell} = \sigma_i^{\ell}(h_i^{\ell}),$$

Activation is a concept across **different data instances** since we need to evaluate it on different inputs, features, neurons.

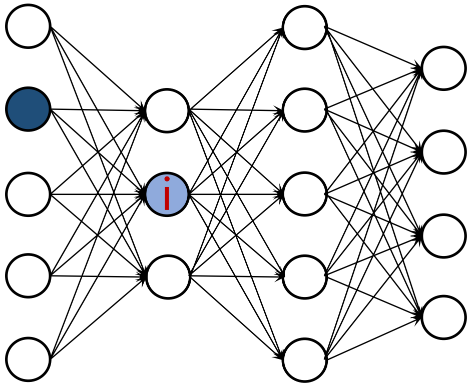


an example of dropout

Method: Detection

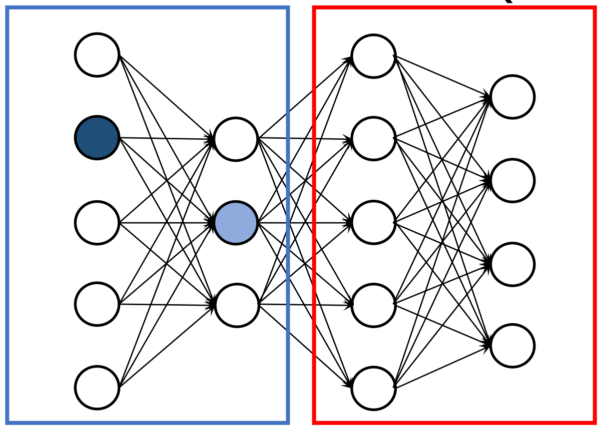
Evaluation Measurement of Monosemantic Neurons

- Intuition: Design the metric $\phi(H)$ of evaluating monosemantic neurons from **low frequency of activation** and **high deviation of activation value**.
- But what is activation in our scenario? (**Another issue**)



i-th neuron at ℓ-th layer:

$$h_j^\ell = \sum_i w_{ij}^\ell z_i^{\ell-1},$$
$$z_i^\ell = \sigma_i^\ell(h_i^\ell),$$



$\left(f_1(\boldsymbol{x})\right)_i = \boldsymbol{z}_i$ $\quad f_2(\boldsymbol{z}) = y$

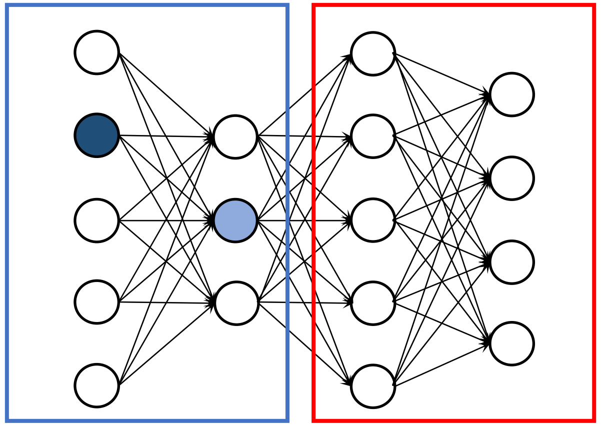
If an input $\boldsymbol{x}$ triggers a neuron z_i to output a value $\left(f_1(\boldsymbol{x})\right)_i$ that deviates **significantly** from its statistical mean $\bar{z}_i$.

Method: Detection

Evaluation Measurement of Monosemantic Neurons

- Intuition: Design the metric $\phi(H)$ of evaluating monosemantic neurons from **low frequency of activation** and **high deviation of activation value**.

- But what is activation in our scenario? (**Another issue**)



$\left(f_1(\boldsymbol{x})\right)_i = \boldsymbol{z}_i$ $\quad f_2(\boldsymbol{z}) = y$

If an input $\boldsymbol{x}$ triggers a neuron z_i to output a value $\left(f_1(\boldsymbol{x})\right)_i$ that deviates **significantly** from its statistical mean $\bar{z}_i$.

Plan A: Set a threshold τ ❌

Plan B: Pairwise comparison ❌

$$\left\|\bar{z}_i - (f_1(\mathbf{x}^{[1]}))_i\right\| < \left\|\bar{z}_i - (f_1(\mathbf{x}^{[2]}))_i\right\|$$

from different data samples

Method: Detection

Evaluation Measurement of Monosemantic Neurons

- Intuition: Design the metric $\phi(H)$ of evaluating monosemantic neurons from **low frequency of activation** and **high deviation of activation value**.
- Given i-th neuron, we denotes its historical samples given m inputs as $\{z_i^{[1]}, z_i^{[2]}, \ldots, z_i^{[m]}\}$ and new value as $z_i^{[m+1]}$, we propose metric Monosemantic Scale (MS) ϕ:

$$\phi(z_i^{[m+1]}) = \frac{\boxed{(z_i^{[m+1]} - \bar{z}_i)^2}}{S^2} \quad \textit{where} \quad \bar{z}_i = \frac{\sum_{j=1}^{m} z_i^{[j]}}{m} \quad S^2 = \frac{\sum_{j=1}^{m} (z_i^{[j]} - \bar{z}_i)^2}{m-1}$$

Can measure the **high deviation**, and $\bar{z}_i$ is mainly decided by **deactivated neurons**.

Method: Detection

Evaluation Measurement of Monosemantic Neurons

- Metric Online Computation Guarantee

LEMMA 3.2. *Denote μ_m as the value of the sample mean $\bar{z}$ given m samples, while v_m as the sample variance S^2. When the $(m+1)^{th} \sim (m+b)^{th}$ samples $z^{[m+1]}, \cdots, z^{[m+b]}$ come, one can obtain the updated values via:*

$$\mu_{m+b} = \frac{m\mu_m + b\mu'_b}{m+b}, \tag{8}$$

$$v_{m+b} = \frac{mb(\mu_m - \mu'_b)^2}{(m+b-1)(m+b)} + \frac{bv'_b + (m-1)v_m}{m+b-1}, \tag{9}$$

where $\mu'_b = \frac{\sum_{i=1}^{b} z_{[m+i]}}{b}$ and $v'_b = \frac{\sum_{i=1}^{b}(z_{[m+i]} - \mu'_b)^2}{b}$, which is of $\boxed{O(1)}$ time and memory complexity as b is a constant.

The intuition behind our theoretical guarantee:

- Define the metric on the train inputs **sequentially** allows us to calculate the metric with **incremental** computation.

Method: Detection

Evaluation Measurement of Monosemantic Neurons

- Given the set of measured MS $\{\phi(z_1^{[j]}), \phi(z_2^{[j]}), \ldots, \phi(z_n^{[j]})\}$ over neurons $\{z_1^{[j]}, z_2^{[j]}, \ldots, z_n^{[j]}\}$ for input $\mathbf{x}^{[j]}$, there are multiple ways to select neurons to inhibit. For example:
  - The maximum one
  - The largest $\log n$ neurons
  - The certain ratio (1%n, 0.1%n)

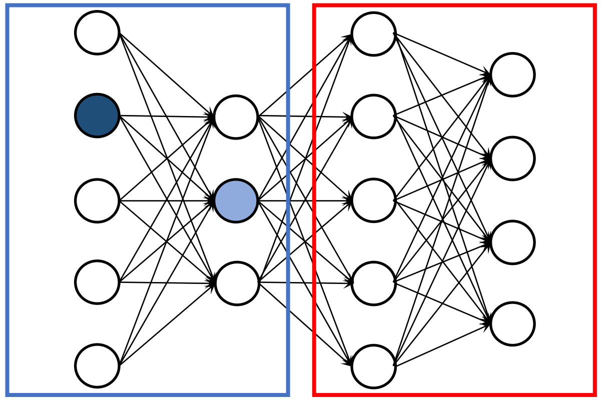
- In our paper, we firstly inhibit **the maximum one** and leave other settings as future work.

Method: Inhibition

Monosemantic Neuron Inhibition

- The goal is to **deactivate** monosemantic neurons to **reduce the monosemantic scale** of the neural networks, i.e., become more polysemantic or distributed.
  - For the identified neuron z_i as "highly monosemantic", design **deactivation strategy** to optimize the frontal model $f_1(\cdot)$ and following model $f_2(\cdot)$ so that:

Expected
- Reduce the activation degree of z_i on input X
  - reduce the reliance $\boldsymbol{x} \to \boldsymbol{z}_i$
- Reduce the dependence of output Y on z_i activation
  - reduce the reliance $\boldsymbol{z}_i \to y$



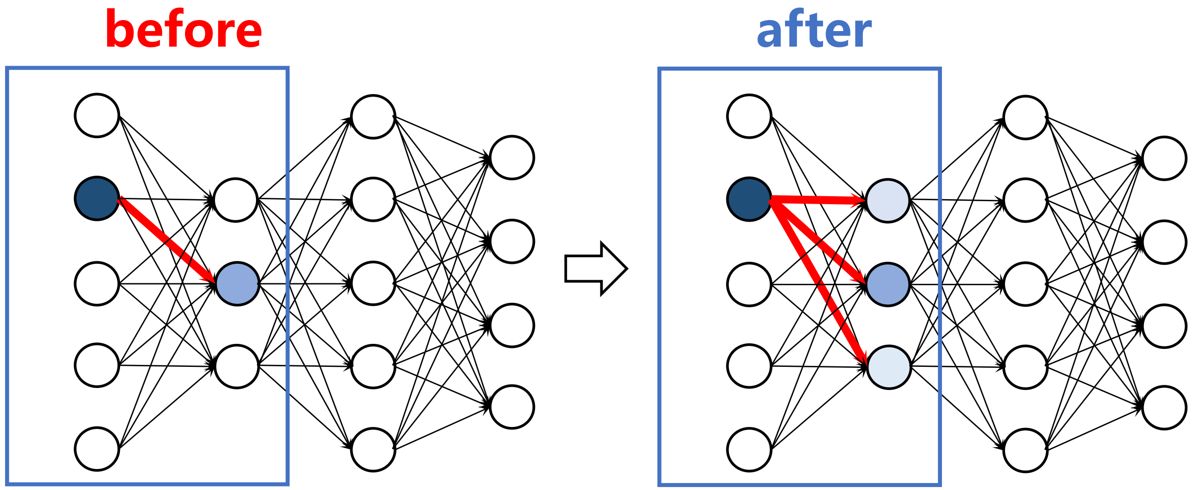
$\left(f_1(\boldsymbol{x})\right)_i = \boldsymbol{z}_i$ $\quad f_2(\boldsymbol{z}) = y$

Method: Inhibition

Monosemantic Neuron Inhibition

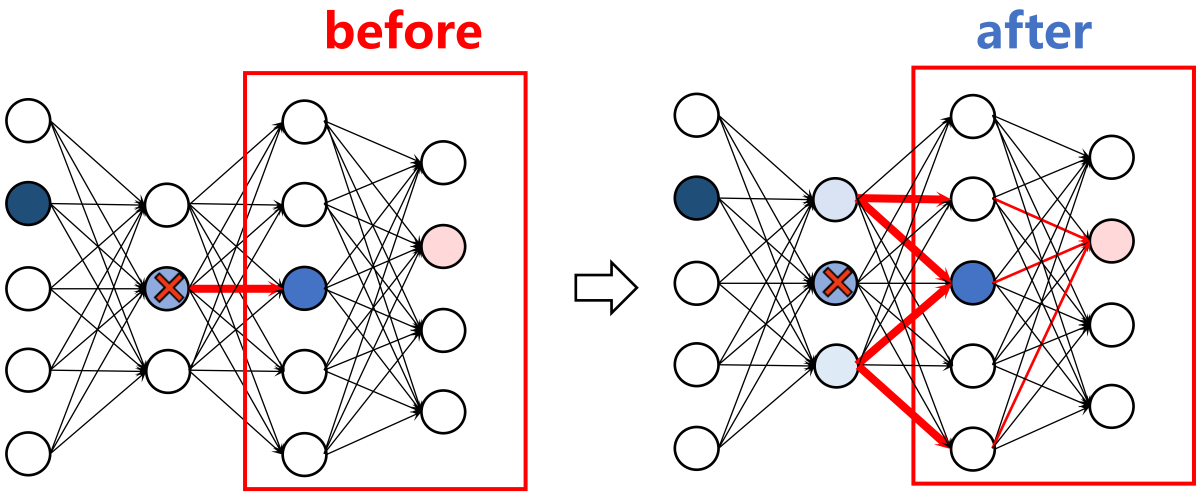
Intuitive Examples for Expected Goals

- Reduce the activation degree of z_i on input X
  - Optimize $\left(f_1(\boldsymbol{x})\right)_i = \boldsymbol{z}_i$ to $\boldsymbol{z}_i'$
- Reduce the dependence of output Y on z_i activation
  - Optimize $f_2(\boldsymbol{z}) = y$



Activate the feature as soon as it comes

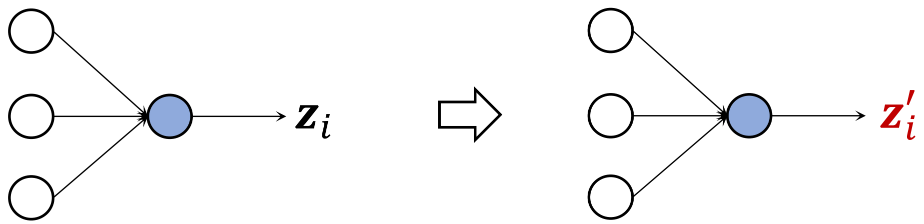
Activation becomes lower and closer to the mean



Bad output without the neuron activated

Other neurons can contribute the prediction

Method: Inhibition



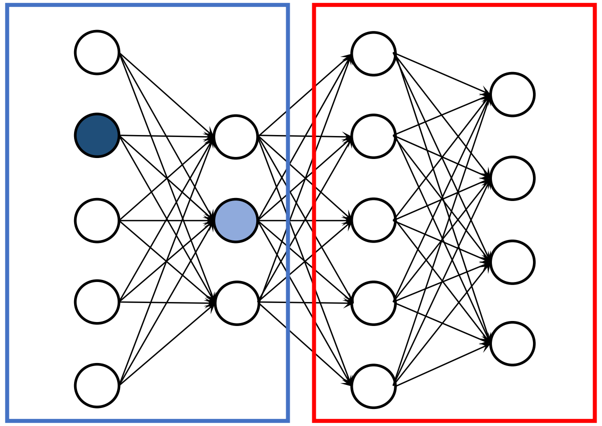
Modify the output of neurons

Monosemantic Neuron Inhibition

Naïve deactivation ways

□ Naïve (a): Deactivate by replacement

$$\mathrm{way}(a): \quad z' = \bar{z}_{ng}$$



$(f_1(\boldsymbol{x}))_i = \boldsymbol{z}_i \quad f_2(\bar{\boldsymbol{z}}_{ng}) = y$

The gradient is cut off. z_i still activated

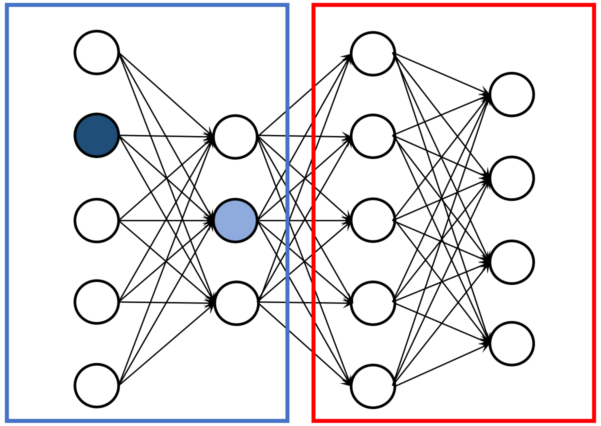
as z_i is deactivated , f_2 has to rely on other neurons

Deactivation: output value = $\bar{\boldsymbol{z}}$ instead of z

The $\cdot_{ng}$ denotes **no-gradient**, which is a scalar-tensor in coding

□ Naïve (b): Deactivate by compensation

$$\mathrm{way}(b): \quad z' = z + (\bar{z} - z)_{ng}$$



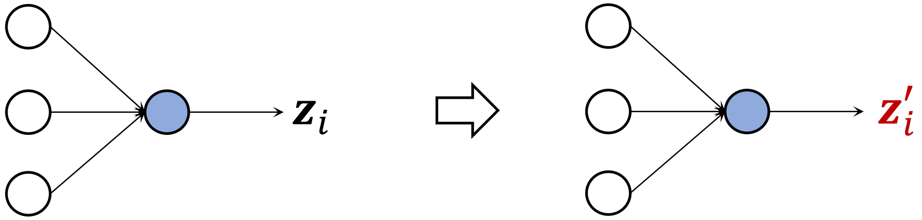
$(f_1(\boldsymbol{x}))_i = \boldsymbol{z}_i \quad f_2(\boldsymbol{z} + (\bar{\boldsymbol{z}} - \boldsymbol{z})_{ng}) = f_2(\bar{\boldsymbol{z}}) = y$

f_1 finds to get better results, z_i should be more activated

as z_i is deactivated , f_2 has to rely on other neurons

Method: Inhibition

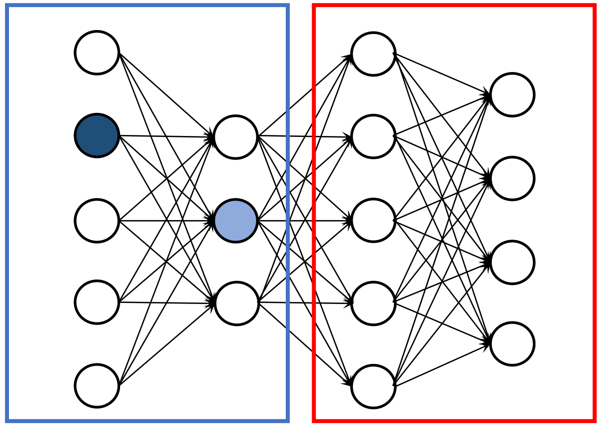
Monosemantic Neuron Inhibition



Modify the output of neurons

The proposed solution: **Reversed Deactivation**

$$z' = -z + (\bar{z} + z)_{ng} \longrightarrow \text{deactivation: } \bar{z}$$



$$\left(f_1(\boldsymbol{x})\right)_i = \boldsymbol{z}_i$$

$$f_2(-\boldsymbol{z} + (\bar{\boldsymbol{z}} + \boldsymbol{z})_{ng}) = y$$

can be optimized by gradients

will be updated to rely less on z_i as it receives a value = $\bar{z}$

(1) model find performance drops
(2) model tries to optimize the neuron z_i to intensify its activation
(3) negative direction: -> deactivation

reduce the activation degree of z_i on input X ✅

Method: Inhibition

Monosemantic Neuron Inhibition

- The theoretical guarantee on neuron inhibition

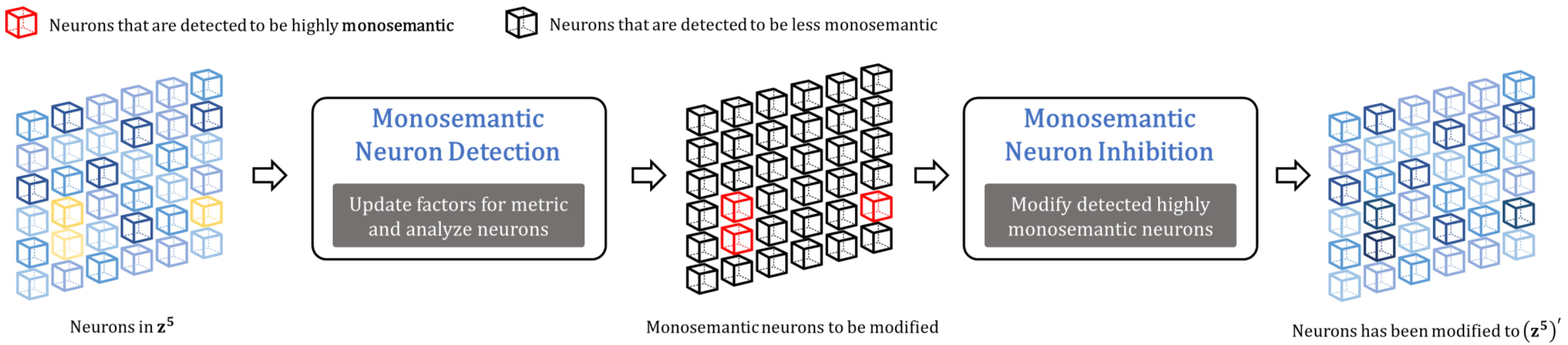
LEMMA 3.3. *Given a trained model f with 2 continuous derivatives and a Lipschitz continuous gradient, where f achieves a desired output $\mathbf{o}$ with minimal loss $\mathcal{L}(\mathbf{o})$, in which $\mathbf{o} = f(\mathbf{x}) = f_2(f_1(\mathbf{x}), \mathbf{x}) = f_2(\mathbf{z}, \mathbf{x})$ for input $\mathbf{x}$ based on its monosemantic neuron z in layer $\mathbf{z}$, suppose that $\mathcal{L}(f_2(\cdot))$ monotonically increases with $|z' - z|$ for any other value z' that replaces z. Then, with a sufficiently small learning rate l, by updating the model f with gradient descent based on the neuron processed by the RD method, the activation of z on input $\mathbf{x}$ can be inhibited.*

Please refer to our paper for details.

Method: Summary

To Inhibit Monosemantic Neurons

- First, design a **metric** to **detect** monosemantic neurons.
  We propose an efficient and flexible metric Monosemantic Scale (MS).
- Second, design **method** to **inhibit** monosemantic neurons.
  We point out problems of naïve methods and propose **Reverse Deactivation**.
- Third, a unified **framework** MEmeL.
  Flexible and lightweight to add to any neural network.



OUTLINE

- Background
- Motivation
- Method
- Experiment

Empirical Study

Experimental Setup

We hope our model MEmeL can be implemented on the top of classic/powerful neural networks to improve their performance by inhibiting Monosemantic neurons.

- **Language Task**
    - Apply MEmeL to the benchmark model **BERT** on the public dataset **GLUE**
- **Image Task**
    - Apply MEmeL to the benchmark model **Swin-Transformer** on the **ImageNet**
- **Simulation Task** (rainfall)
    - Apply MEmeL to the benchmark model **ConvGRU** on the public dataset **HKO-7**

Empirical Study

Experimental Setup

We hope our model MEmeL can be implemented on the top of classic/powerful networks to improve their performance by inhibiting monosemanticity.

Table 1: Results on GLUE Test datasets. We follow the setting of BERT to demonstrate results on 8 datasets and calculate the average score. The scores are F1 scores for QQP and MRPC, Spearman correlations for STS-B, and accuracy scores for the other tasks. All metrics are the larger the better with best results in bold font.

Model	MNLI-(M/MM)	QQP	QNLI	SST-2	CoLA	STS-B	MRPC	RTE	Average
Original	84.6/83.4	71.2	90.5	93.5	52.1	85.8	88.9	66.4	79.6
Naive (a)	84.3/83.6	**71.7**	90.6	**93.8**	52.1	85.8	88.2	66.4	79.6
Naive (b)	84.7/**84.1**	71.6	90.6	93.6	51.8	86.5	87.2	68.0	79.8
MEmeL	**84.8**/83.9	**71.7**	90.9	93.6	54.5	**86.6**	87.6	**68.2**	80.2
MEmeL-Tune	**84.8**/83.9	**71.7**	**91.2**	93.7	**55.7**	**86.6**	**89.0**	**68.2**	**80.5**

- Our MEmeL is better than the original and naïve methods
- Beyond the basic setting (deactivating **top-1** monosemantic neuron in each batch), we additionally tuning the level of inhibition to see the potential improvement can be achieved.

Table 2: Results on ImageNet-1k dataset, where 3 sizes of Swin-Transformer pretrained on ImageNet-22k are used as backbones. The metric used is top-1 accuracy, where a higher value indicates better performance. The best results are indicated in bold font.

Model	Swin-T	Swin-S	Swin-B
Size	28M	50M	88M
Original	80.9	83.2	85.1
Naive (a)	81.0	83.4	84.6
Naive (b)	81.0	83.4	85.1
MEmeL	**81.1**	83.4	85.1
MEmeL-Tune	**81.1**	**83.5**	**85.2**

Table 3: Results on HKO-7 dataset. We initially trained a ConvGRU model for 20k steps to create the base model. The metrics used are B-MSE and B-MAE, where a smaller value indicates better performance. The best results are in bold fonts. We repeated each experiment three times and reported the average scores.

Model	B-MAE	B-MSE
Original	1003.41	309.96
Naive (a)	1003.56	309.83
Naive (b)	1003.40	310.13
MEmeL	1003.25	309.94
MEmeL-Tune	**998.81**	**298.16**

Empirical Study

Experimental Setup

- We hope our model MEmeL can be implemented on the top of classic/powerful neural networks to improve their performance by inhibiting monosemantic neurons.
- We hope our model MEmeL can indeed reduce the monosemantic scale of neural networks.

Table 3: Validation experiments conducted on the Swin-B model. We record the Decrease Ratios and Update Scales of 10k neurons. The model that utilizes our Reverse Deactivation method is compared with those using two Naive methods and the original Swin-B.

Methods	Original	Naive (a)	Naive (b)	Reverse Deactivation
Average Decrease Ratio	0.003%	-0.017%	-0.044%	**0.013%**
Average Total Update Ratio	0.052%	0.118%	0.161%	**0.189%**

Compared with two naive methods, our reverse deactivation **suppresses** monosematic neurons.

Future Works

- Need to verify the effectiveness of our metric.

- Need to verify the proposition on **large language models.**

- Need to verify the effectiveness of our method on **large language models**.

Future Works

- Need to verify the effectiveness of our metric.

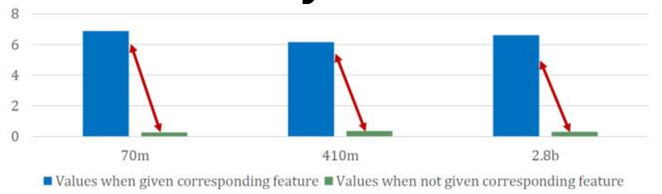


Fig. 1. Metric MS outputs values significantly different when input contains (blue) and not contains (green) the monosemantic features. Results are based on the most monosemantic 10 neurons across scales (70m to 2.8b) of pythia model, detected by sparse probing.

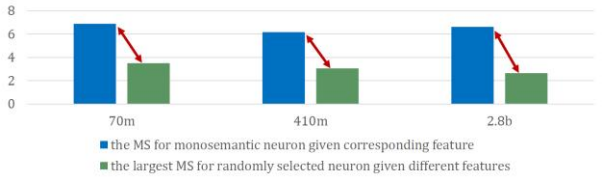


Fig. 2. Metric MS outputs much larger values for monosemantic neurons (blue) compared with randomly selected neurons (green). The settings are the same with Figure 1. For each randomly selected neuron, we records its output values given different features, and display the largest one as its relatively most sensitive feature.

Larger-scale

Full validation

New module

Following work is coming!

- Need to verify the proposition on **large language models.**

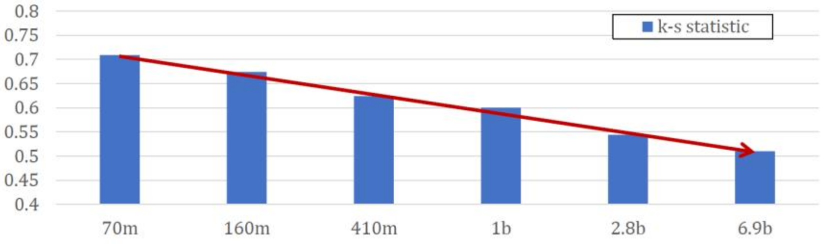
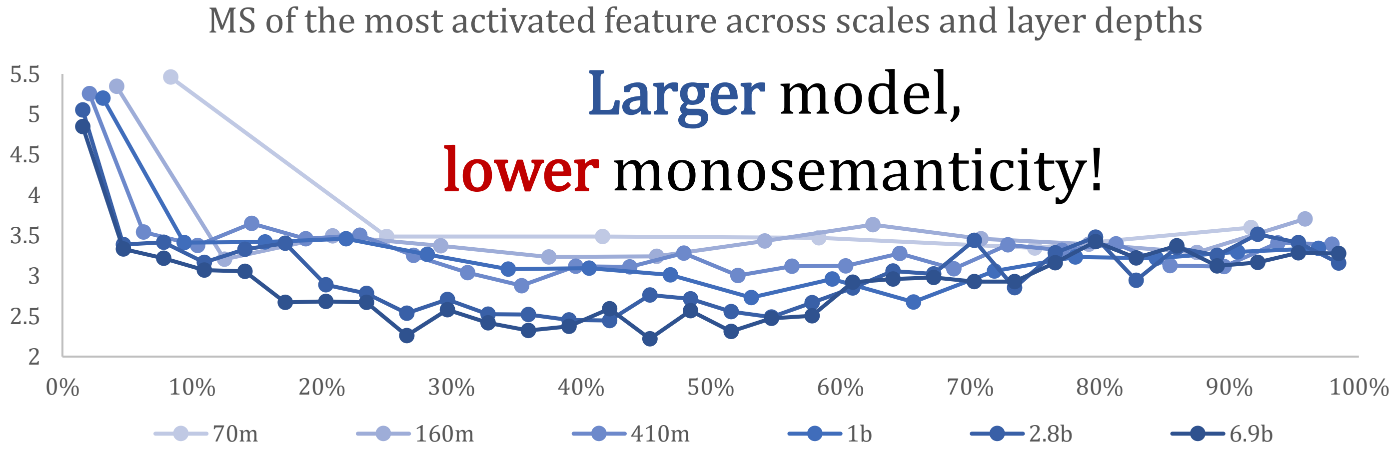


Fig. 3. Statistics of monosemanticity across scales. Randomly select 1000 neurons each scale and conduct Kolmogorov–Smirnov test for the scores of most monosemantic feature and the global scores. A lower k-s statistic refers less outstanding of the scores of most monosemantic feature, indicating a lower monosemanticity. One can observe the statistic results are negatively related with increasing scale.



Larger model,
lower monosemanticity!

- Need to verify the effectiveness of our method on **large language models**.

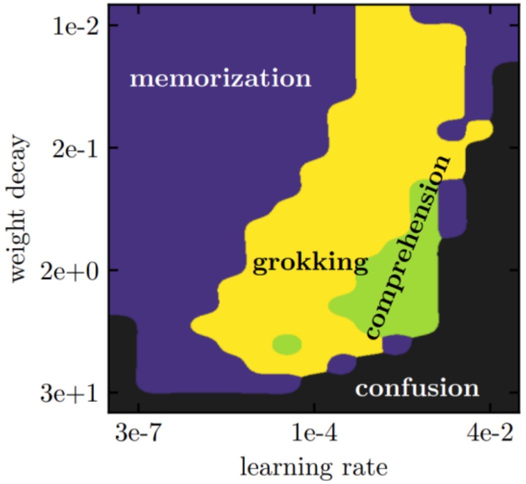
Calling for cooperation: full pretraining LLM with MEmeL.

Future Works

Possible directions

- Memorization plays a different role in different tasks
  → Inhibit or promote monosemanticity should be task oriented

Complex tasks require more polysemanticity



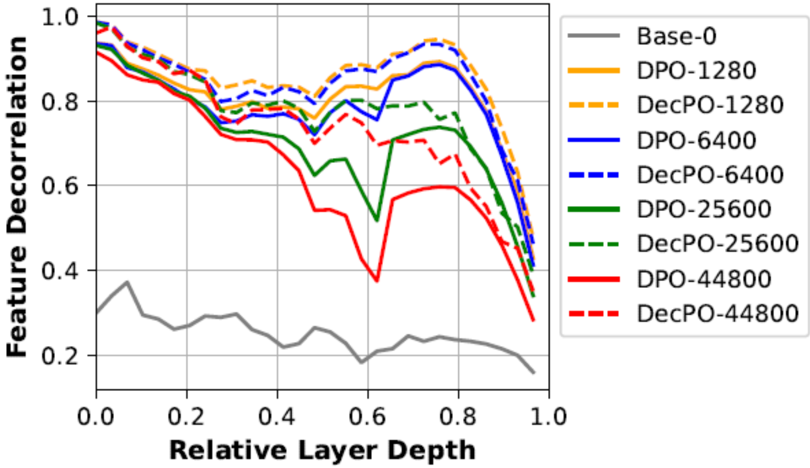
Modular Addition: expects grokking instead of memorization

4.4 Potential and Limitation of MEmeL

According to our hypothesis, MEmeL induces the model to accumulate general and abstract functionality instead of monosemanticity for a specific task, which is consistent with the goal of per-taining. Although MEmeL achieves good results during fine-tuning (demonstrated at Main Experiments in subsection 4.2), the improvement is expected to be even greater when it is applied to the pre-training phase.

Our new work also finds MEmeL is especially effective for harder tasks.

Preferences alignment require more monosemanticity



During direct preference optimization: monosemanticity is enhanced

Thank For Watching

Summary

We propose to **learn from emergence** to present a study on proactively **inhibiting the monosemantic** neurons of artificial neural networks.

- The Evaluation Metric for Detecting Monosemantic Neurons
  - **Data-specific evaluation** → A **quantitative** metric **does not** rely on datasets.
  - **Large** computational overhead → **Online** computation guarantee.
- The Proactive Deactivation Method to Reduce Monosemantic Neurons
  - **Hard** to deactivate → A **theoretically** supported method to suppress monosemantic neurons

Github **Technical Report** **Paper**