THE HONG KONG
UNIVERSITY OF SCIENCE
AND TECHNOLOGY

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

Dr. Shimin DI

Joint Work with Dr. Jiachuan WANG, Prof. Lei CHEN, Prof. Charles Wang Wai Ng

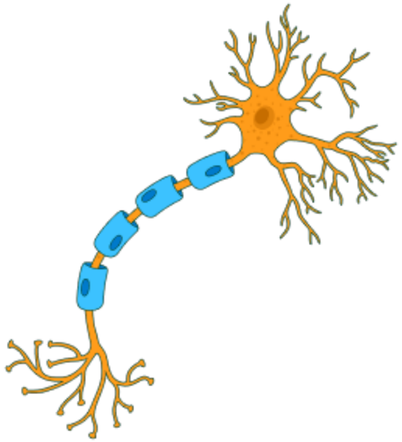
Contact us: dishimin@ust.hk

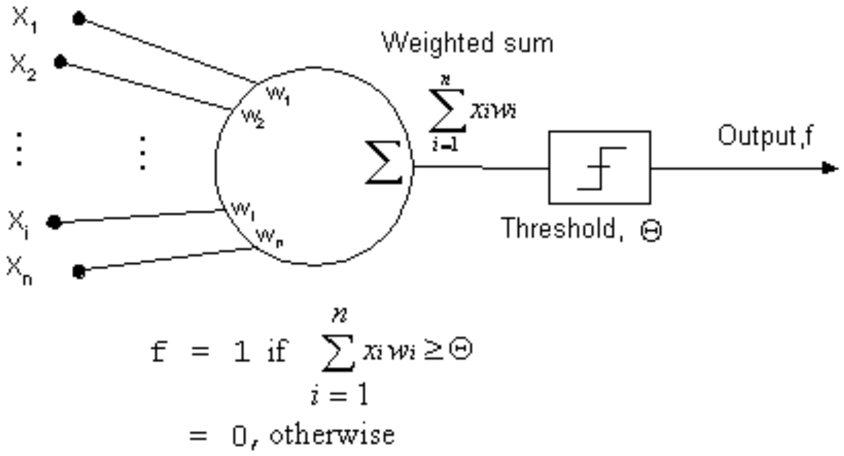
Department of Computer Science and Engineering
The Hong Kong University of Science and Technology

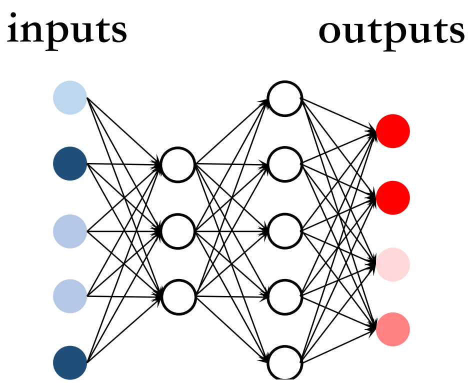
Background

Artificial Neural Networks

- In 1943, Warren McCulloch and Walter Pitts presented their model of artificial neurons, considered the first artificial intelligence.
  - The term "artificial intelligence" was coined on 1956 by John McCarthy.







i-th neuron at ℓ-th layer:

$$h_j^\ell = \sum_i w_{ij}^\ell z_i^{\ell-1}, \quad \text{inputs}$$

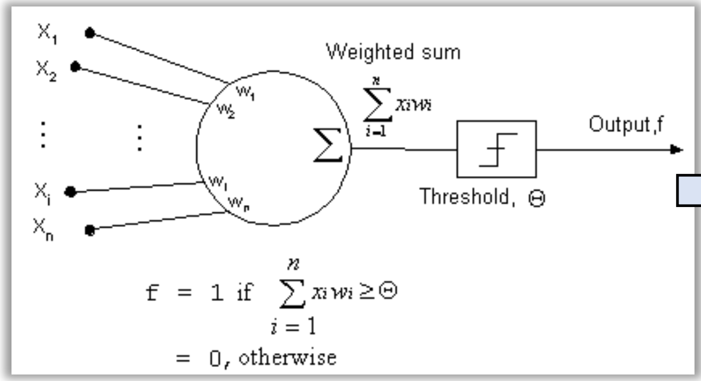
$$z_i^\ell = \sigma_i^\ell(h_i^\ell), \quad \text{activated values}$$

Background

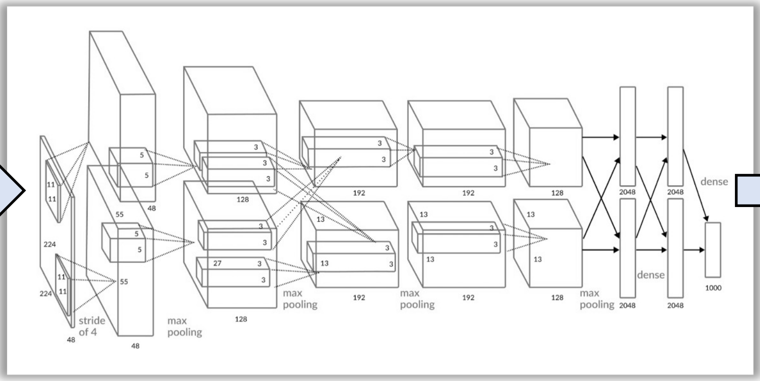
Artificial Neural Networks

- Development in Recent Years

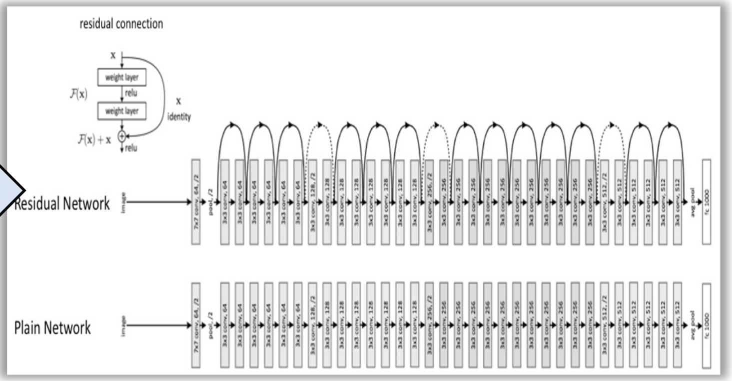
1943, Artificial Neuron



2012, AlexNet



2015, ResNet

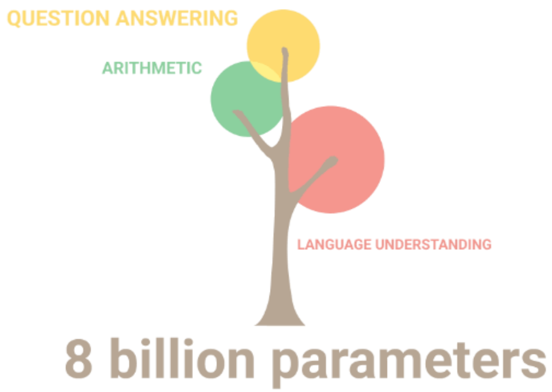


Milestones in the development of artificial neural networks are accompanied by a large increase in scale.

Background

Emergence from Large Language Models

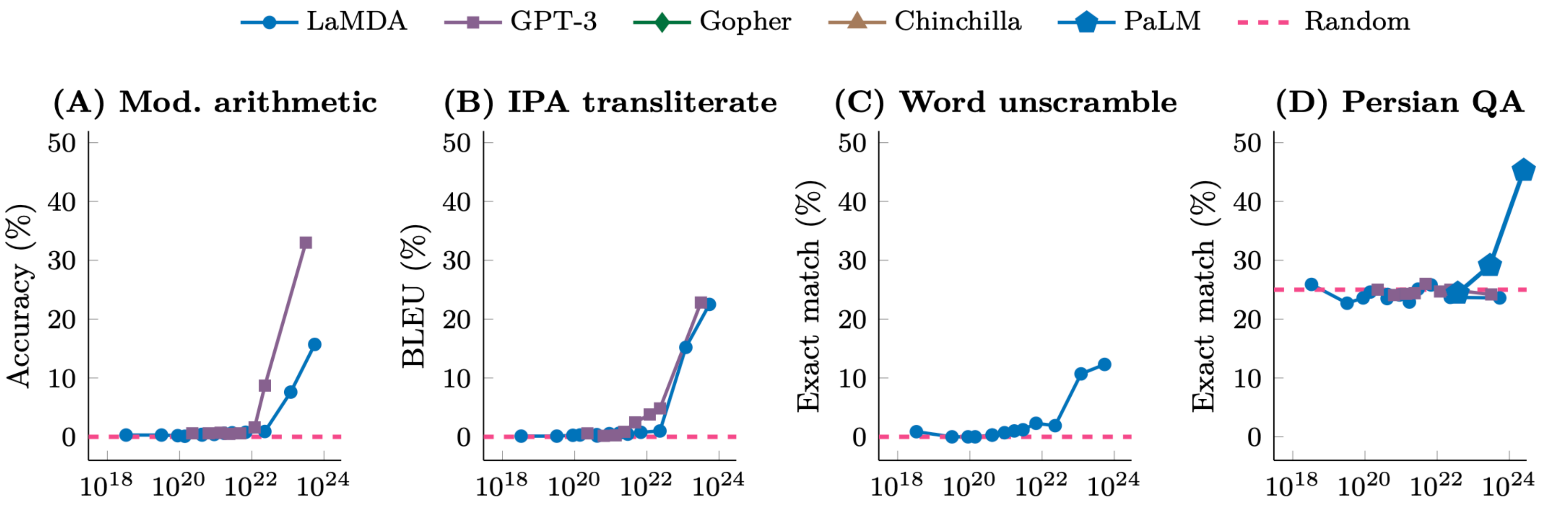
- **Emergence** is the gradual improvement of model performance before the scale reaching a certain threshold, followed by a rapid enhancement once the threshold is surpassed.



Background

Emergence from Large Language Models

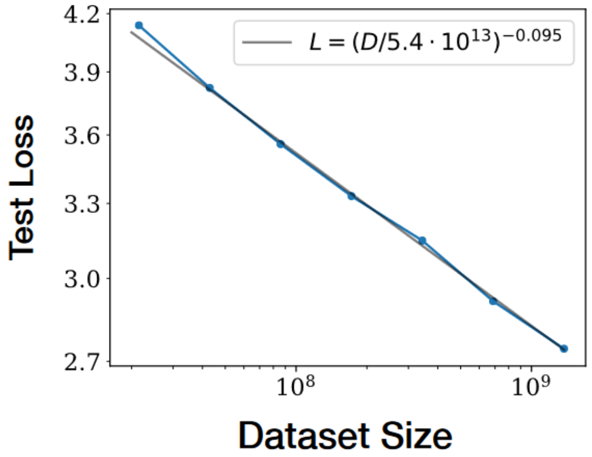
- **Emergence** is the gradual improvement of model performance before the scale reaching a certain threshold, followed by a rapid enhancement once the threshold is surpassed.

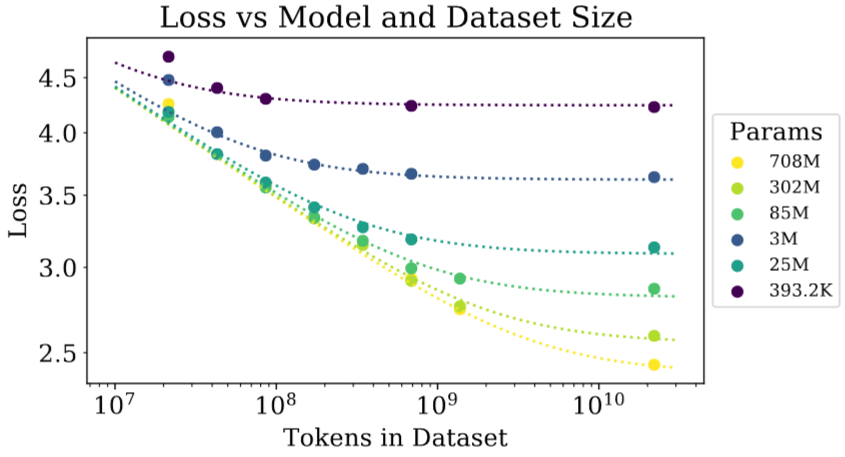


Background

Emergence from Large Language Models

- Increasing evidence suggests that the surprises may not arise from new module and architecture designs, but rather from the underlying nature of scale changes.





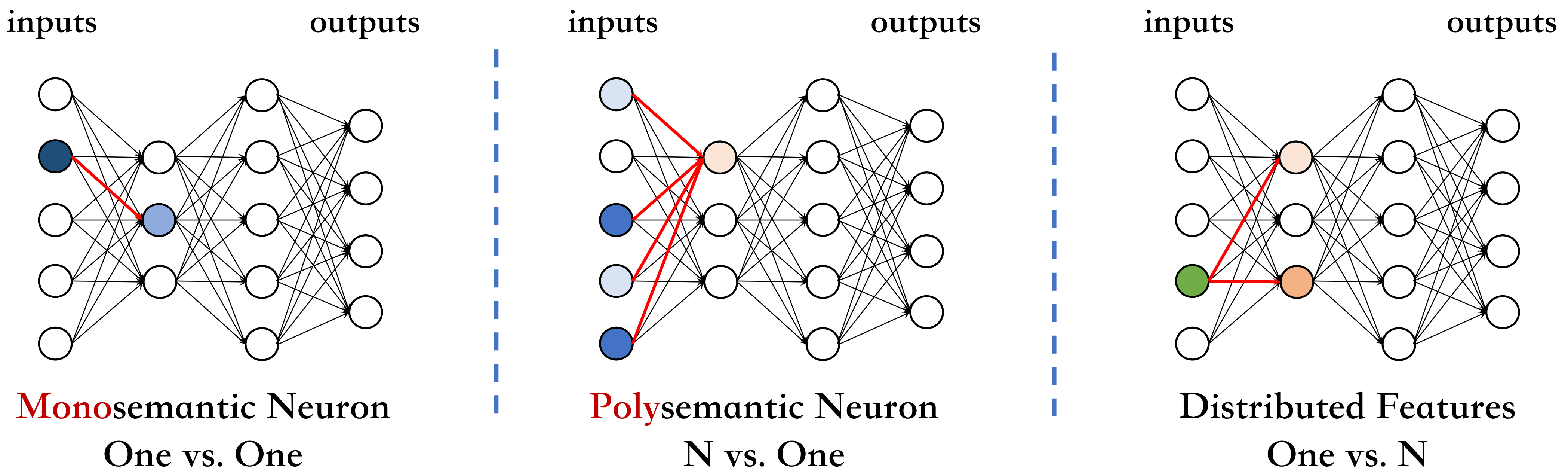
One interesting question: *People increase the model scale and get better results, but **what** has changed underlying the process?*

- Kaplan et al., Scaling Laws for Neural Language Models. 2020
- Xu et al., CVALUES: Measuring the Values of Chinese Large Language Models from Safety to Responsibility. 2023

Motivation

Interpreting Emergence

- Some pioneer works try to interpret the performance of small and large-scale models from the correlation between neurons and input features.



Motivation

Motivational Experiments from Literature

- From literature, we observe that **large models have low monosemanticity.**
  - 1st Observation: Given the specific feature, when turning off monosemantic neurons, the error of a large model drops smaller than that of a small model.

depends on the model size—in the 70M parameter model (≈ 12k neurons), ablating a single neuron causes an average loss increase of 8% per French sequence, while in the 6.9B model (≈524k neurons), ablating one neuron results in only a 0.2% increase in loss.

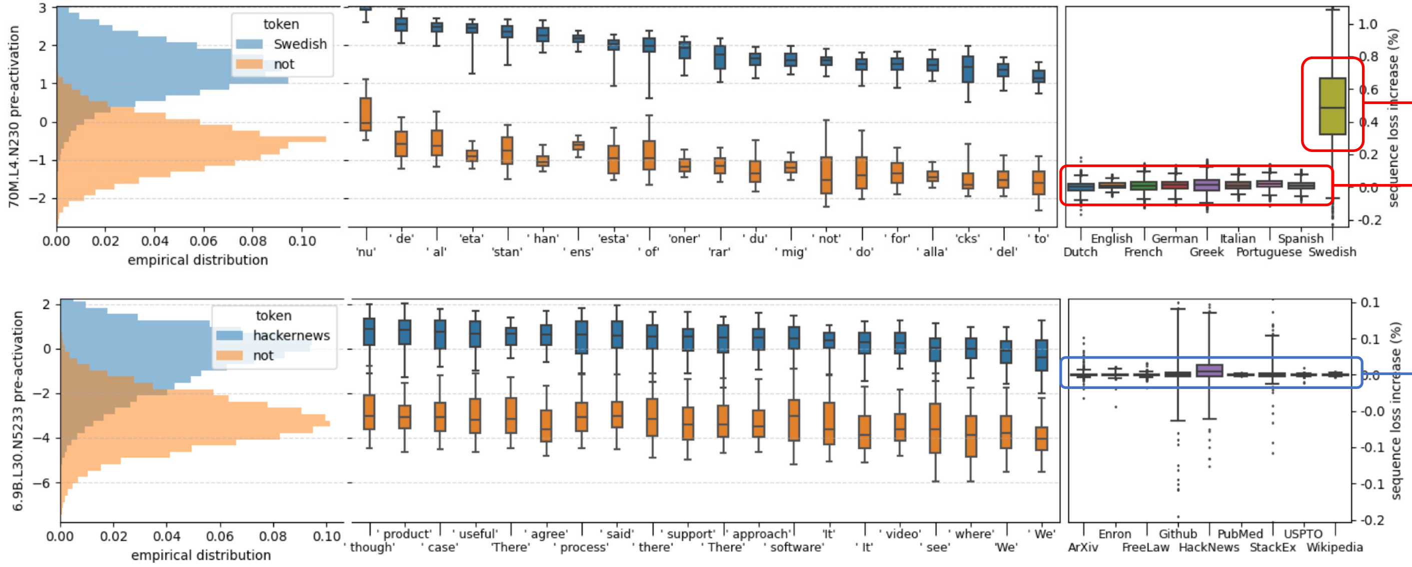
Motivation

Motivational Experiments from Literature

- From literature, we observe that **large models have low monosemanticity.**
  - 2nd Observation: Given the corresponding/non-corresponding features, the difference in activation values of large models is smaller than that of small models.

Pythia-70M

Pythia-6.9B



Large difference between the corresponding and non-corresponding features

Small difference

Motivation

Summarized Motivations from Literature

- From literature, we observe that **large models have low monosemanticity.**
  - 1st Observation: Given the specific feature, when turning off monosemantic neurons, the error of a large model drops smaller than that of a small model.
  - 2nd Observation: Given the corresponding/non-corresponding features, the difference in activation values of large models is smaller than that of small models.
- Motivated by existing works, we propose an assumption:

*the **decrease** of monosemantic neurons may be a key factor in achieving **higher** performance as the model **scale increases**.*

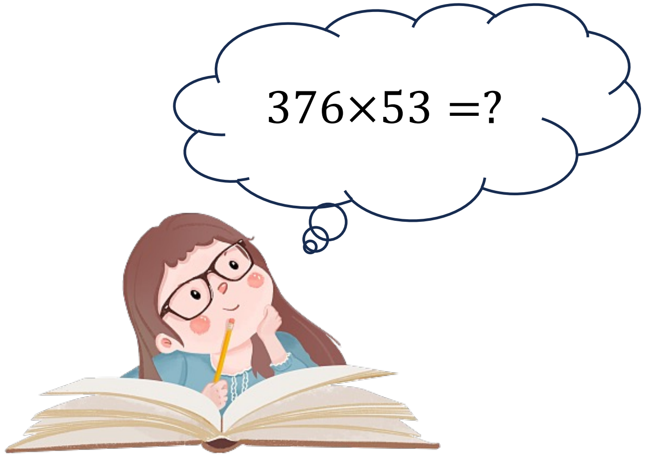
Motivation

Motivational Examples

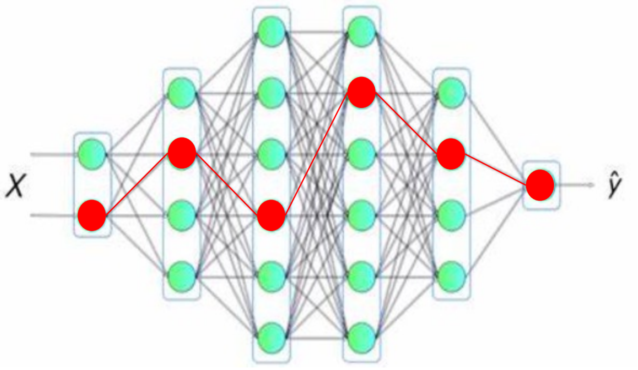
- Assumption: The decrease of monosemantic neurons may be a key factor in achieving **higher** performance as the model scale increases.

  - 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.



376×53 = 19928
376×53 = 19928
376×53 = 19928
376×53 = 19928
376×53 = 19928
memorize repeatedly
train repeatedly

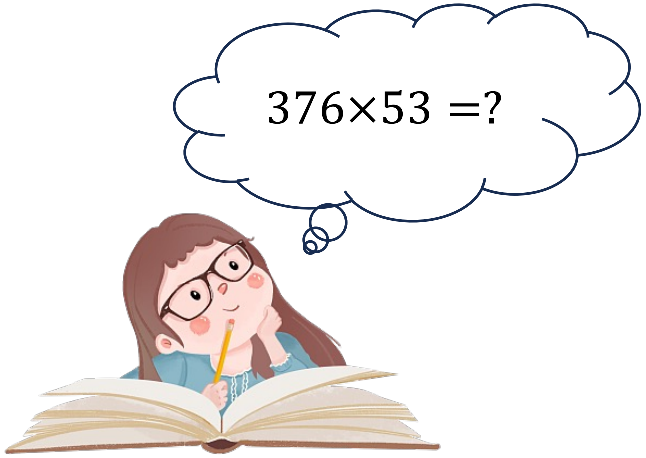


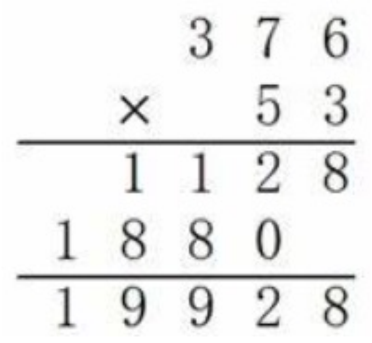
⇨ 376×53 = 19928

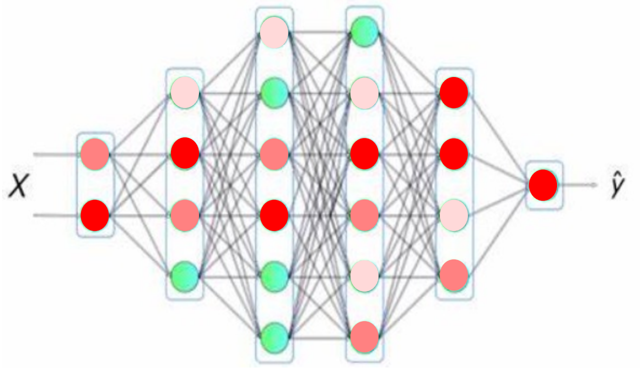
Motivation

Motivational Examples

- Assumption: The decrease of monosemantic neurons 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 weights the output.

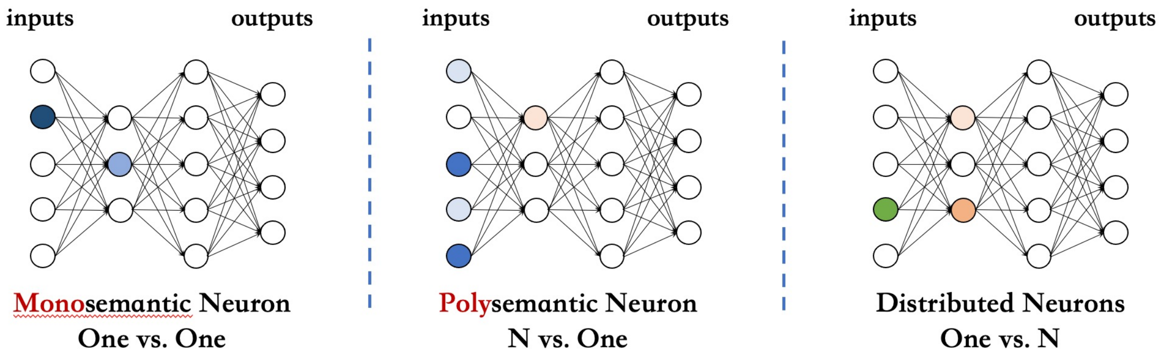






$376 \times 53 = 19928$

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

Motivational Experiments from Literature

- Inspired by the emergence, we propose one question:

*Can we **proactively inhibit monosemantic neurons** in artificial neural networks to achieve high performance?*

- Unfortunately, it is a non-trivial task to proactively inhibit monosemantic neurons from the perspectives of **monosemantic neurons detection** and **inhibition**.

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.

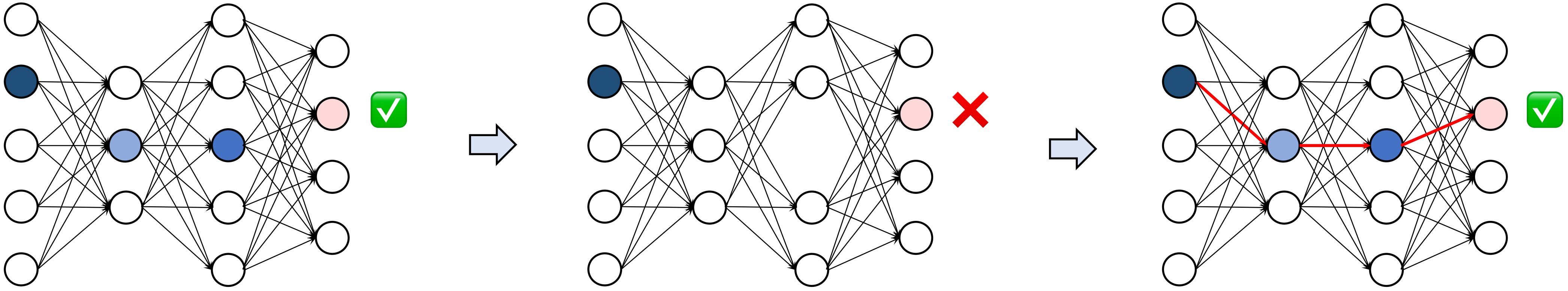
- Strictly defining monosemantic neurons is still under discussion in quantitative analysis.
  - Generality: Detection does not dependent on a specific data set.
  - Efficiency: Detect monosemantic neurons during online training.

Expected

Motivation

Technical Challenges: Monosemantic Neuron Inhibition

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



correct prediction   wrong prediction   enhance the monosemanticity

Motivation

Summary on Technical Contributions

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

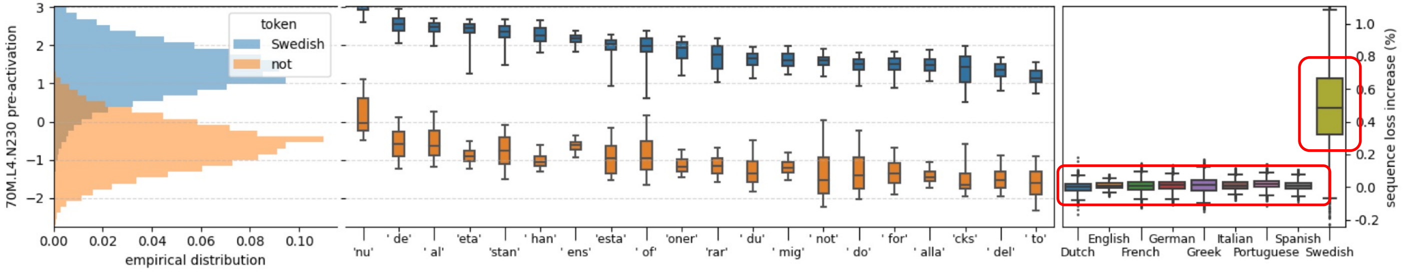
- The Evaluation Metric for Detecting Monosemantic Neurons
  - Data-specific evaluation → A quantitative metric **does not** relies on data sets.
  - Large computational overhead → Online computation guarantee.
- The Proactive Deactivation Method to Reduce Monosemantic Neurons
  - Hard to deactivate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

A Study on Proactively Inhibiting the Monosemantic Neurons

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

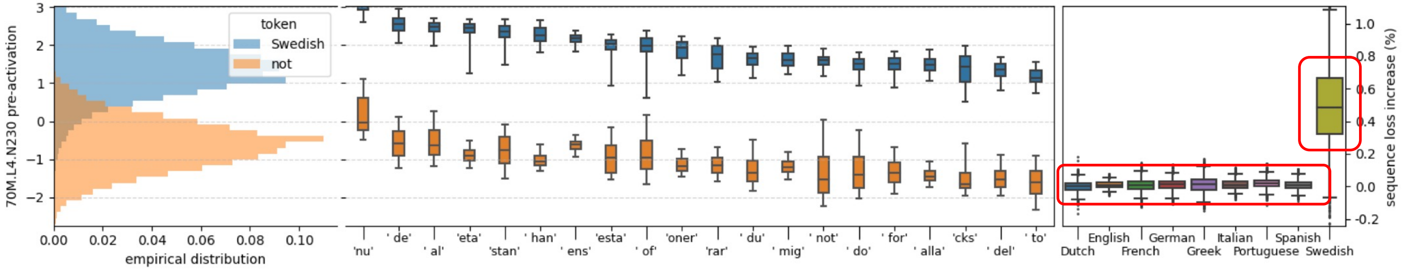


A Study on Proactively Inhibiting the Monosemantic Neurons

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.

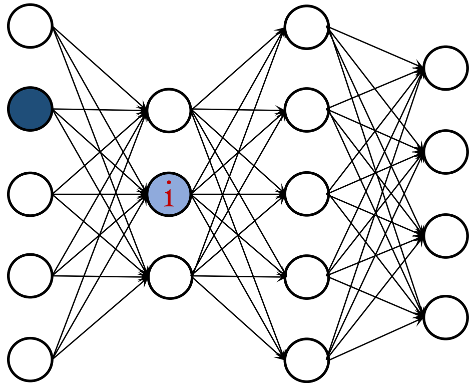
Pythia-70M



A Study on Proactively Inhibiting the Monosemantic Neurons

Evaluation Measurement of Monosemantic Neurons

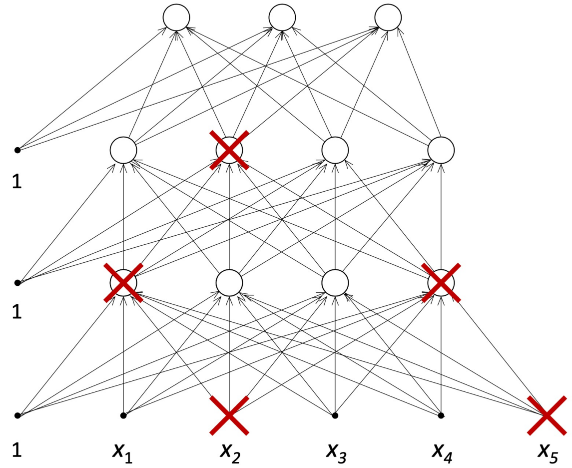
- Intuition: Design the metric $\phi(H)$ of evaluating monosemantic neurons from low frequency of activation and high deviation of activation value.
- But what is activation in our scenario? (Another issue)



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.

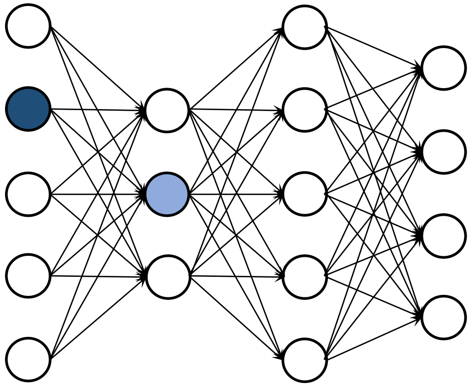


an example of dropout

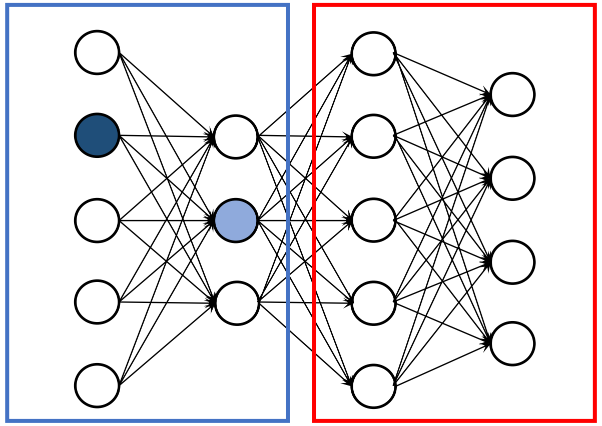
A Study on Proactively Inhibiting the Monosemantic Neurons

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

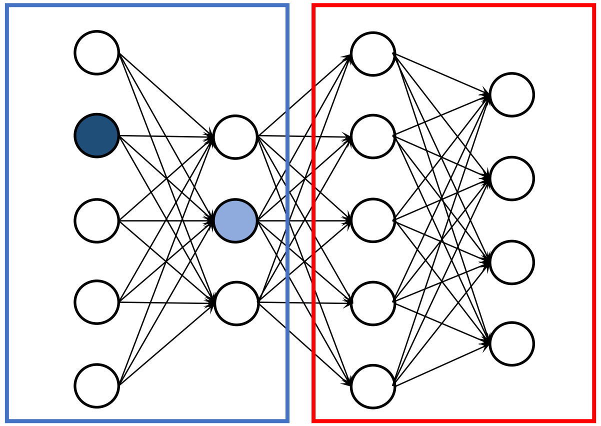


$(f_1(\boldsymbol{x}))_i = \boldsymbol{z}_i$ $\quad f_2(\boldsymbol{z}) = y$

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

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?



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

If an input $\boldsymbol{x}$ triggers a neuron z_i to output a value $(f_1(\boldsymbol{x}))_i$ 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

A Study on Proactively Inhibiting the Monosemantic Neurons

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 under m inputs $\{x^{[1]}, x^{[2]}, \dots, x^{[m]}\}$ as $\{z_i^{[1]}, z_i^{[2]}, \dots, z_i^{[m]}\}$ and new value under $x^{[m+1]}$ as $z_i^{[m+1]}$. The proposed monosemanic scale evaluation $\phi(z_i)$:

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

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

A Study on Proactively Inhibiting the Monosemantic Neurons

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.*

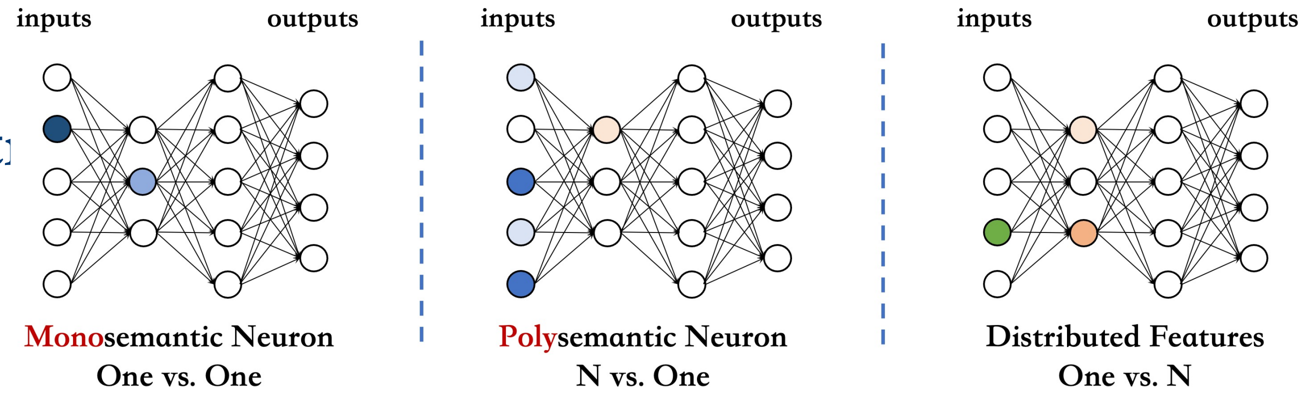
Intuition behind our theoretical guarantee:

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

Evaluation Measurement of Monosemantic Neurons

- Given a series of measured monosemantic scales $\{\phi(z_1^{[j]}), \phi(z_2^{[j]}), \ldots, \phi(z_n^{[j]})\}$, there are multiple ways to filtering those monosemantic neurons:
  - The maximum one
  - The largest $\log n$ neurons
  - The maximum one in every batch
  - The certain ratio (1%n, 0.1%n)
  - Sampling from the distribution $\phi(\cdot)$

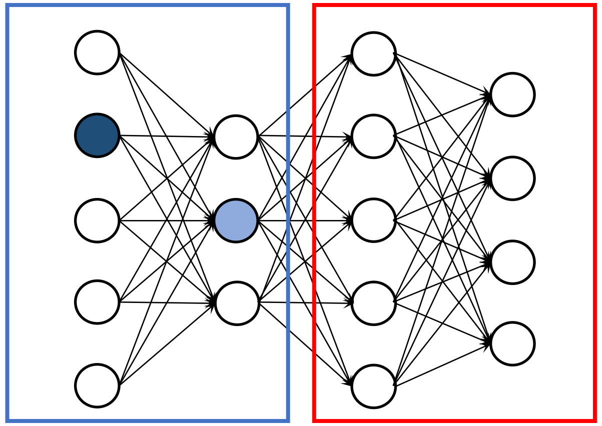
A Study on Proactively Inhibit



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 correlation $\boldsymbol{x} \rightarrow \boldsymbol{z}_i$
- Reduce the dependence of output Y on z_i activation
  - reduce the correlation $\boldsymbol{z}_i \rightarrow y$



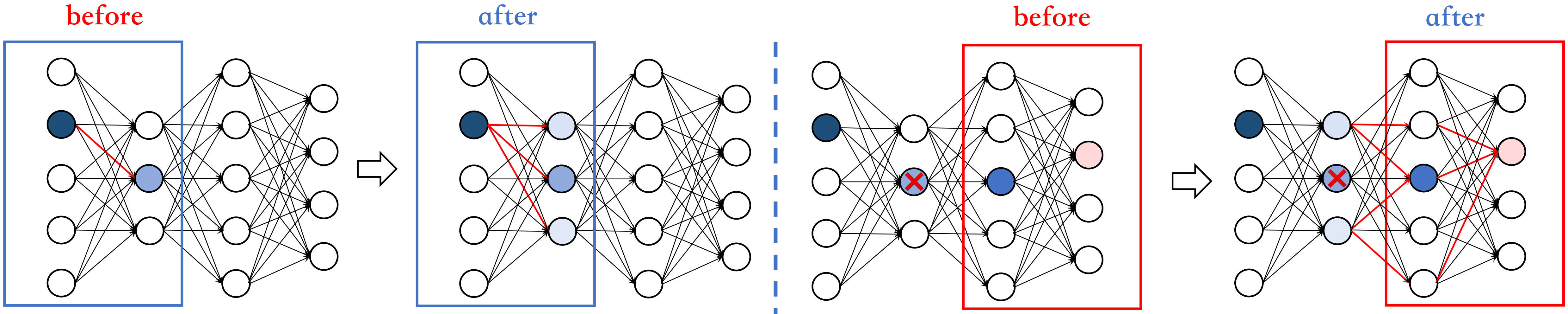
$\left(f_1(\boldsymbol{x})\right)_i = \boldsymbol{z}_i$ $\quad f_2(\boldsymbol{z}) = y$

A Study on Proactively Inhibiting the Monosemantic Neurons

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

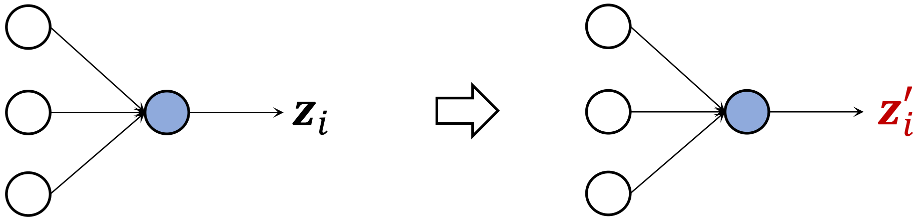
Activation becomes lower and closer to the mean

Bad output without the neuron activated

Other neurons can contribute the prediction

A Study on Proactively Inhibiting the Monosemantic Neurons

Monosemantic Neuron Inhibition



Modify the output of neurons

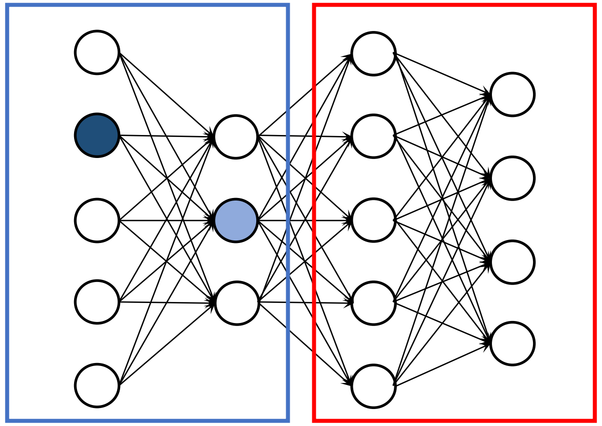
Naïve deactivation ways

- Naïve (a): Deactivate the neuron directly
- Naïve (b): Deactivate the neuron directly

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

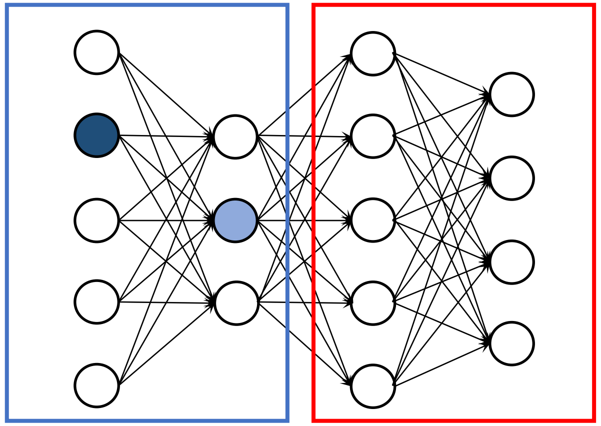
deactivation: $\bar{z}$

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



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

but $\boldsymbol{z}_i$ still activated — will not rely on $\boldsymbol{z}_i$



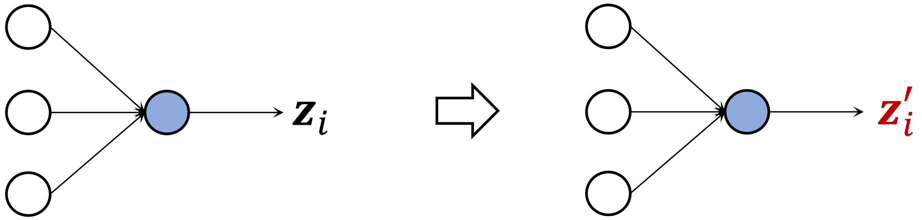
$\left(f_1(\boldsymbol{x})\right)_i = \boldsymbol{z}_i \quad f_2(\boldsymbol{z} + (\bar{\boldsymbol{z}} - \boldsymbol{z})_{ng}) = y$

Even to be more activated — will not rely on $\boldsymbol{z}_i$

A Study on Proactively Inhibiting the Monosemantic Neurons

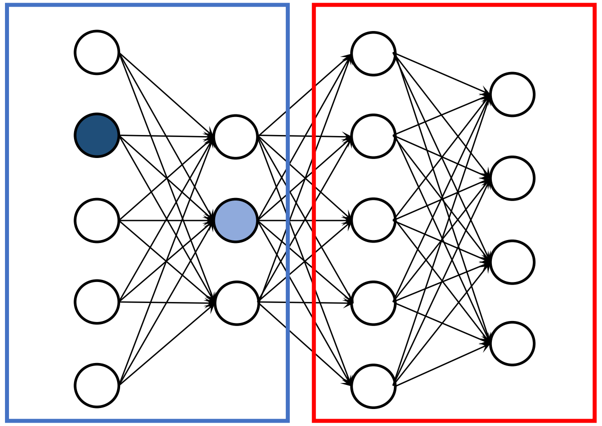
Monosemantic Neuron Inhibition

- ☐ The proposed solution: Reversed Deactivation



Modify the output of neurons

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



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

$$f_2(\underbrace{-\boldsymbol{z}}_{\text{can be optimized by gradients}} + (\bar{\boldsymbol{z}} + \boldsymbol{z})_{ng}) = y$$

will not rely on $\boldsymbol{z}_i$ due to $\bar{\boldsymbol{z}}$

(1) model find performance drops
(2) model tries to optimize the neuron i to increase its weight
(3) negative direction: -> decrease weight

⇩

reduce the activation degree of $\boldsymbol{z}_i$ on input $\boldsymbol{X}$ ✅

A Study on Proactively Inhibiting the Monosemantic Neurons

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.*

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

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.

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	66.4	80.0
MEmeL-Tune	**84.8**/83.9	**71.7**	**91.2**	93.7	**55.7**	**86.6**	**89.0**	**68.1**	**80.5**

- Only Top-1 monosemantic neuron is deactivated in each batch
- MEmeL (reverse deactivation) is better than others

Table 2: The experimental results of Swin-Transformer on the ImageNet data and ConvGRU on the data HKO-7. For results on ImageNet-1k dataset, 3 Swin-Transformers pretrained on ImageNet-22k are used as backbones. The metric used is top-1 accuracy, where a higher value indicates better performance. For results on HKO-7 dataset, we initially train 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.

Model	Swin-T	Swin-S	Swin-B	B-MAE	B-MSE
Size	28M	50M	88M		
Original	80.9	83.2	85.1	1003.41	309.96
Naive (a)	81.0	83.4	84.6	1003.56	309.83
Naive (b)	81.0	83.4	85.1	1003.40	310.13
MEmeL	**81.1**	83.4	85.1	1003.25	209.94
MEmeL-Tune	**81.1**	**83.5**	**85.2**	**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.

Summary

Shortcomings

- Need to verify the effectiveness of our method on large language models.
- Need to monitor whether the training process of modern neural networks (e.g., CNNs, RNNs) on different public benchmark data sets changes from high to low.
- Need to prove that our method is significantly faster and more effective in terms of inhibiting monosemnatic neurons, and then verify the superiority of proactive inhibition over passive method.

However, extending this research to very large-scale datasets is appealing yet impossible for research departments due to limited resources. We are delighted to share the co-authorship and await collaboration from any AI company/group.