Tracing and Manipulating Intermediate Values in Neural Math Problem Solvers

Yuta Matsumoto[1] **Benjamin Heinzerling**[2] **Masashi Yoshikawa**[1] **Kentaro Inui**[1,2]
[1]Tohoku University / Japan [2]RIKEN / Japan
yuta.matsumoto.q8@dc.tohoku.ac.jp, benjamin.heinzerling@riken.jp,
yoshikawa@tohoku.ac.jp, inui@ecei.tohoku.ac.jp,

Abstract

How language models process complex input that requires multiple steps of inference is not well understood. Previous research has shown that information about intermediate values of these inputs can be extracted from the activations of the models, but it is unclear where that information is encoded and whether that information is indeed used during inference. We introduce a method for analyzing how a Transformer model processes these inputs by focusing on simple arithmetic problems and their intermediate values. To trace where information about intermediate values is encoded, we measure the correlation between intermediate values and the activations of the model using principal component analysis (PCA). Then, we perform a causal intervention by manipulating model weights. This intervention shows that the weights identified via tracing are not merely correlated with intermediate values, but causally related to model predictions. Our findings show that the model has a locality to certain intermediate values, and this is useful for enhancing the interpretability of the models.

1 Introduction

Recent language models (LMs) can solve complex input such as math word problems (Saxton et al., 2019; Geva et al., 2020). To obtain the correct output from such complex (latent structured) inputs, it is necessary for multiple steps of inference via intermediate values. However, how LMs process their inputs and capture latent structure is still not well understood. In previous studies, Linzen et al. (2016) and Tran et al. (2018) showed that the neural models can capture some implicit hierarchical structure, but it is unclear where that information is encoded. Shibata et al. (2020) observed that in LMs trained with Dyck language and showed some activations are highly correlated with the depth of their syntactic tree. However, even if such features can be extracted, there is no guarantee that it is used by the model (Elazar et al., 2021; Lovering et al., 2021). Given these considerations, to better understand LM predictions for latent structured inputs, it is necessary to: (a) To find where information about intermediate values of the latent structured inputs is encoded. (b) To evaluate the impact of the features when the model makes predictions.

In this work, we introduce a method for analyzing the relationship between internal representations in Transformer (Vaswani et al., 2017)-based models and intermediate values of latent structured inputs by using simple math problems. We choose them as a formal language because their intermediate values of the latent (tree) structure are clear and continuous, and it is easy to investigate their relationship to the internal representation of the model. The intermediate value of $(154 - 38) - (290 - 67)$ can be clearly defined as 154, 290, $154 - 38 = 116$, and so on. we take up a Transformer model trained to solve math equations. An overview of our experiments is shown in Fig. 1. First, we search which directions of internal representations are highly correlated with intermediate values in equations by PCA (**tracing**) to find where the information about intermediate values is encoded. We find some directions correlate very well with the intermediate values. Second, we observe how the model prediction changed when we manipulate the weights along its direction (**manipulation**) to conduct a causal intervention. The result of this experiment suggests that some directions of them are indeed used by the model.

These two results show that a Transformer model has a locality to certain intermediate values, and it could help enhance the interpretability of the models. Our contributions are as follows: (a)We show that intermediate values of equations are encoded in particular directions in internal representation. (b)We show that some features representing intermediate values are used during inference.

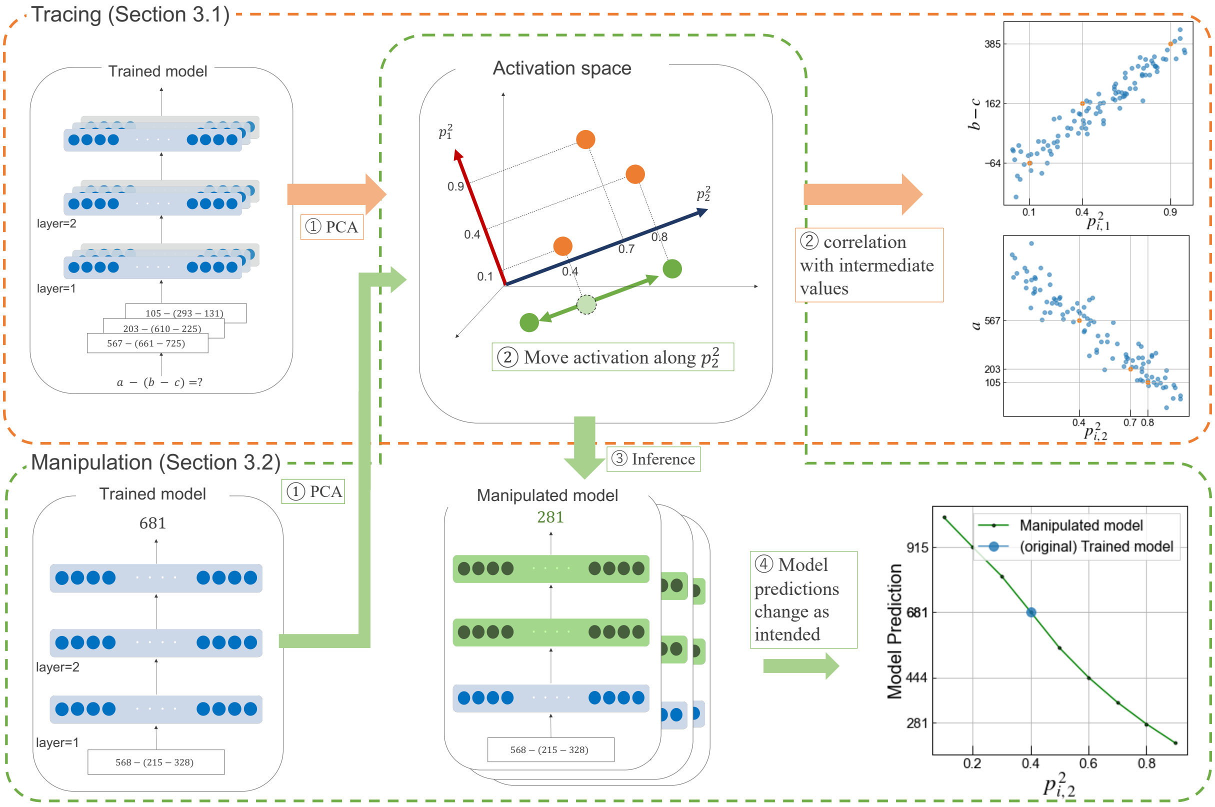


Figure 1: An overview of our methods. We find which directions obtained by PCA are correlated with intermediate values of the equations and how the model prediction changes when the weights of their directions are manipulated.

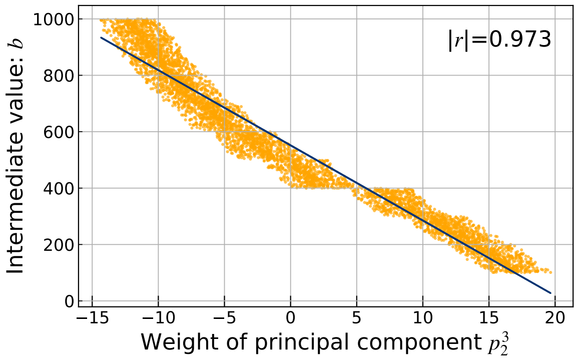


Figure 2: The relationship between $p_{i,2}^3$ and $R_i^j = b_i$ in the equation $a - (b - c)$. The correlation is very high.

2 Related Work

Intermediate values. Previous work has examined the representation of intermediate values in neural models. Linzen et al. (2016), Bowman et al. (2015) and Tran et al. (2018) found that LMs capture implicit hierarchical structures to some extent, e.g., when performing logical inference over formal languages. Closest to this work are Shibata et al. (2020), who trained LMs on the Dyck language and observed hidden units that are highly correlated with nesting depth. In contrast to their work, we analyze representations of more complex inputs, i.e., equations, and also manipulate these representations to understand the impact of correlated activations on model predictions.

Numeracy Geva et al. (2020) have shown that they can reach the state-of-the-art performance of numerical reasoning by using large pre-trained LM. Several studies have shown that a Transformer model can solve more complex problems such as linear algebra and elementary mathematics to some extent (Charton, 2021; Saxton et al., 2019). Based on their findings, we use simple mathematical equations as problems that can be solved by a Transformer model in this study.

3 Experiments

We conduct two types of experiments. First we trace the representation of intermediate values in the model. As a result we find directions in activation space that are highly correlation with intermediate values. Then we manipulate activations along these directions and observe if model predictions change as expected.

As neural math problem solving model, we train

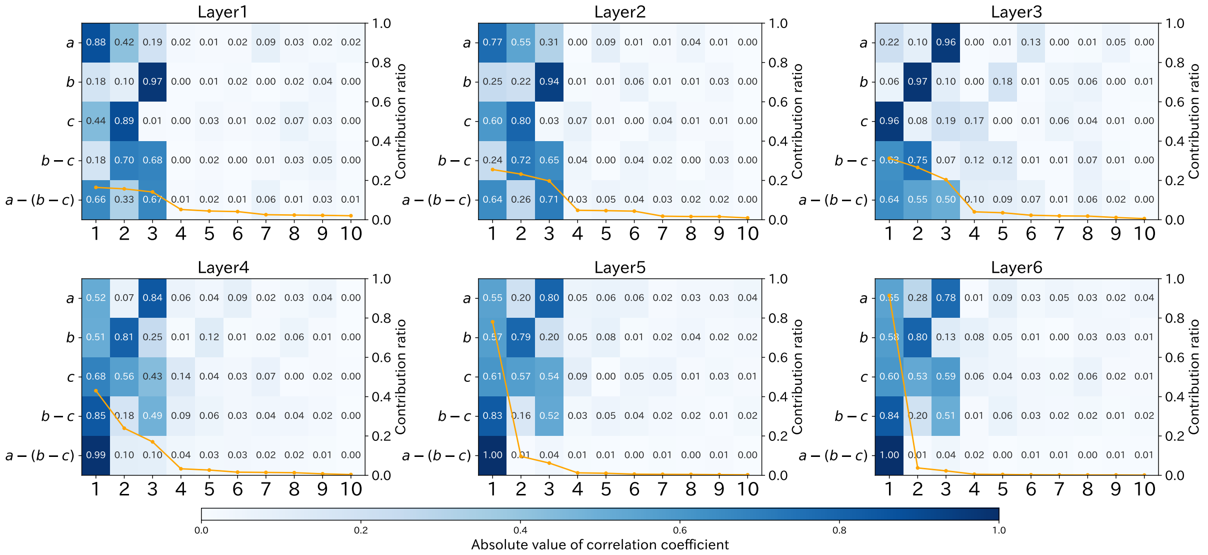


Figure 3: Correlations between each principal component and intermediate values, for all layers. Each cell represents the absolute value of the correlation coefficient between the weights of k-th principal component (column) and the intermediate values (row). The orange line shows the contribution ratio of each principal component.

a 6-layer Transformer using the settings by Sajjad et al. (2021) on synthetic data. We generate 200k equations involving up to five steps of addition or subtraction of integers between 1 and 1000, e.g., $(154 - 38) - (290 - 67)$. Following Geva et al. (2020) inputs are split into digits, e.g., “123” is tokenized into 1, ##2, ##3. Model predictions are obtained via linear regression on the final layer's [CLS] token representation. After training on 190k equations we evaluate the model on 10k equations and obtain a regression score of $R^2 = 0.9988$, i.e., the model solves the equations almost perfectly.

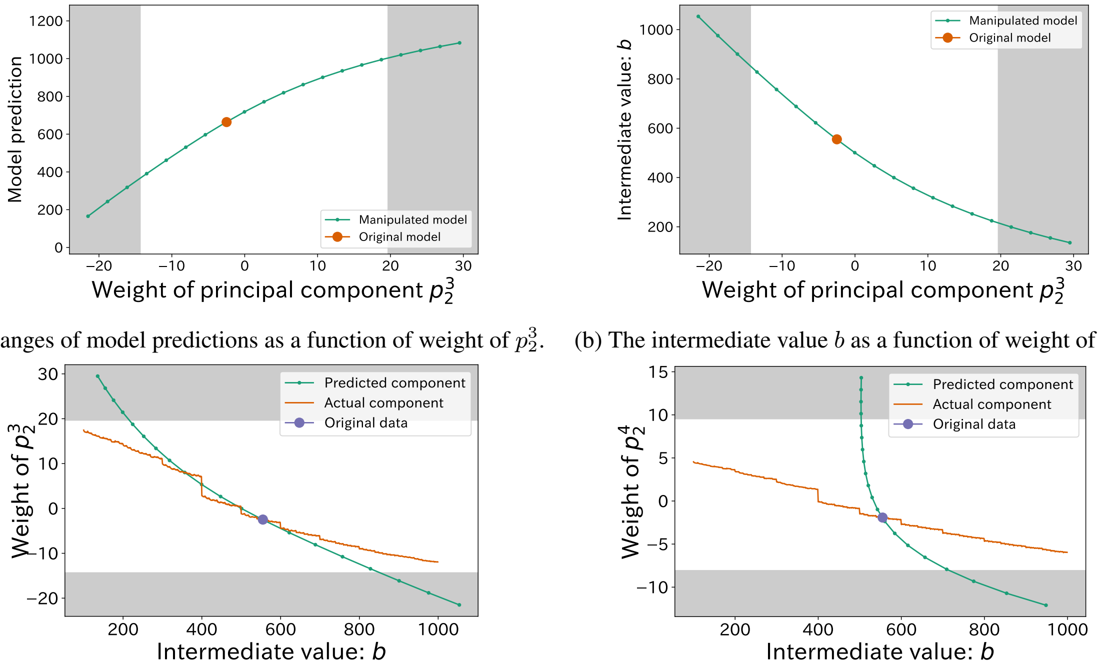
3.1 Tracing intermediate values

Method. We describe our method for tracing the representation of intermediate values in model activations. First, we reduce the dimensionality of the activations at each model layer. Let h^l_j be the layer activations of the j-th word in a hidden layer l. Given an input of length n, we concatenate all token representations in layer l, obtaining the layer representation $H^l = h^l_1 \oplus h^l_2 \oplus \cdots \oplus h^l_n$ and fit a PCA to obtain the top 10 principal components $p^l_k, k \in [1, ..., 10]$. Applying this PCA to instance i yields the 10-dimensional representation $p^l_{i,k}$. Our hypothesis is that the intermediate values are encoded by one or more of the principal components. Intuitively, we assume that a principle component encodes an intermediate value if the magnitude of model activation in this direction correlates with the magnitude of the intermediate values. To test this hypothesis, we measure the correlation $\mathbf{corr}(R^j_i, p^l_{i,k})$ between the value of the intermediate values R^j_i and the magnitude of principal component k in the representation $p^l_{i,k}$. Finally, we obtain **most-correlated direction** $\hat{p}^l_k(R^j) := \mathbf{argmax}_k(\mathbf{corr}(R^j_i, p^l_{i,k}))$. If this correlation is high, we conclude that the intermediate value is encoded in that direction.

Results. We trace intermediate values for the equation pattern $a - (b - c)$. For example, Fig. 2 shows a strong correlation of 0.973 between the intermediate value b and its most-correlated direction p^3_2. After measuring the correlation of each intermediate value and each of the top 10 principal components, we plot all correlations in Fig. 3. Overall, most-correlated directions show high correlations with intermediate values with moderate contribution ratio up to the 3rd layer, which we take as evidence that the model encodes intermediate values along these directions.

3.2 Manipulating intermediate values

Method. So far, we found correlations between intermediate values and directions in activation space. However, such correlations do not necessarily mean that these directions determine model predictions. To test if the directions we found actually influence model predictions, we perform causal interventions by *manipulating* activations. Concretely, we manipulate activations along principal



(a) Changes of model predictions as a function of weight of p_2^3.

(b) The intermediate value b as a function of weight of p_2^3.

(c) Predicted and actual weights of most-correlated direction $\hat{p}_2^3(b)$ as a function of the intermediate value b.

(d) Predicted and actual weights of most-correlated direction $\hat{p}_2^4(b)$ as a function of the intermediate value b.

Figure 4: The results of manipulation. The weights of the shaded areas do not appear in the dataset.

components and observe changes in model predictions, as shown in Fig. 1. Formally, we transform layer representation H^l (see §3.1) into $H^{l\prime}$, by increasing or decreasing its projection onto the principal component p_k^l by a factor of r:

$$H^{l\prime} \leftarrow H^l + (r-1)\left({p_k^l}^\top H^l\right) p_k^l \qquad (1)$$

Intuitively, increasing r moves H^l along p_k^l.

If a most-correlated direction $\hat{p}_k^l(R^j)$ indeed encodes the intermediate value R^j, it should be possible to manipulate activations in a way that corresponds to changing R^j. For example, if the model prediction given the input $43-(50-20)$ changes from the 13 to 19, this difference is consistent with changing the first input term from 43 to 49. By manipulation factors r of a particular most-correlated direction, observing model predictions, and calculating corresponding intermediate values, we obtain data for fitting a function from intermediate values to manipulation factors r. That is, we learn to manipulate activations in a way that corresponds to changing a particular intermediate value. To assess the fidelity of this manipulation, we change input terms and compare *actual* activation changes along the most-correlated direction $\hat{p}_k^l(R^j)$ to the factor r *predicted* by our fitted function.

Results. Using the input $617-(555-602)$ and the intermediate value $b=555$ as example, we find its most-correlated direction $\hat{p}_2^3(b)$, as described in §3.1. By manipulating activations along p_2^3, model predictions change from the original 664 to results ranging from ca. 200 to 1000, as shown in Fig. 4(a). Calculating intermediate values b that are consistent with these model predictions, we obtain Fig. 4(b). By axis inversion we obtain a function from b to *predicted* manipulation factors r for component $p_{i,2}^3$. We compare these *predicted* component weights to the *actual* component weights observed under changed inputs $\{(617-(i-602))|(100 \leq i < 1000)\}$ (Fig. 4(c)). Predicted and the actual weights of the most-correlated direction agree well (corr. 0.986, R^2 score 0.687), which we take as evidence that $\hat{p}_2^3(b)$ encodes the intermediate value b and determines model predictions accordingly. Conversely, manipulation identifies most-correlated directions that are correlated but less used in prediction. The most-correlated direction $\hat{p}_2^4(b)$ has a high correlation of 0.81 with b, but predicted component weights show much less agreement with actual weights (corr. 0.802, R^2 score -1.06×10^4, Fig. 4(d)).

In conclusion, this case study showed how manipulations in activation space can find a causal connection to intermediate values.

4 Acknowledgments

This work was supported by JST CREST Grant Number JPMJCR20D2, Japan.

References

Samuel R. Bowman, Christopher D. Manning, and Christopher Potts. 2015. Tree-structured composition in neural networks without tree-structured architectures. In *Proceedings of the 2015th International Conference on Cognitive Computation: Integrating Neural and Symbolic Approaches - Volume 1583*, COCO'15, page 37–42, Aachen, DEU. CEUR-WS.org.

François Charton. 2021. Linear algebra with transformers.

Yanai Elazar, Shauli Ravfogel, Alon Jacovi, and Yoav Goldberg. 2021. Amnesic probing: Behavioral explanation with amnesic counterfactuals. *Transactions of the Association for Computational Linguistics*, 9:160–175.

Mor Geva, Ankit Gupta, and Jonathan Berant. 2020. Injecting numerical reasoning skills into language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 946–958, Online. Association for Computational Linguistics.

Tal Linzen, Emmanuel Dupoux, and Yoav Goldberg. 2016. Assessing the ability of LSTMs to learn syntax-sensitive dependencies. *Transactions of the Association for Computational Linguistics*, 4:521–535.

Charles Lovering, Rohan Jha, Tal Linzen, and Ellie Pavlick. 2021. Predicting inductive biases of pre-trained models. In *International Conference on Learning Representations*.

Hassan Sajjad, Fahim Dalvi, Nadir Durrani, and Preslav Nakov. 2021. On the effect of dropping layers of pre-trained transformer models.

David Saxton, Edward Grefenstette, Felix Hill, and Pushmeet Kohli. 2019. Analysing mathematical reasoning abilities of neural models. In *International Conference on Learning Representations*.

Chihiro Shibata, Kei Uchiumi, and Daichi Mochihashi. 2020. How LSTM encodes syntax: Exploring context vectors and semi-quantization on natural text. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 4033–4043, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Ke Tran, Arianna Bisazza, and Christof Monz. 2018. The importance of being recurrent for modeling hierarchical structure. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4731–4736, Brussels, Belgium. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.