AN INTRINSIC DIMENSION PERSPECTIVE OF TRANSFORMERS FOR SEQUENTIAL MODELING

Anonymous authors
Paper under double-blind review

ABSTRACT

Transformers have gained great popularity for sequential modeling, especially in fields such as natural language processing (NLP). Recently, numerous architectures based on the Transformer framework are proposed, leading to great achievements in applications. However, the working principles behind still remain mysterious. In this work, we numerically investigate the geometrical properties of data representation learned by Transformers, via a mathematical concept called intrinsic dimension (ID), which can be viewed as the minimal number of parameters required for modeling. A series of experiments, mainly focusing on text classification tasks, backs up the following empirical claims on relationships among embedding dimension, depth, respective ID per layer and tasks performance. First, we surprisingly observe that a higher ID (of terminal features extracted by Transformers) typically implies a lower classification error rate. This is contrary to that of CNNs (or other models) performed on image classification tasks. In addition, it is shown that the ID per layer tends to decrease as the depth increases, and this reduction usually appears more significant for deeper architectures. Moreover, we give numerical evidence on geometrical structures of data representation learned by Transformers, where only the nonlinear dimension reduction can be achieved. Finally, we explore the effect of sequential lengths on the ID and tasks performance, which guarantees the validity of data reduction in training. We hope that these findings can play a guiding role in hyper-parameters selection and dimension/data reduction for Transformers on text classification and other mainstream NLP tasks.

1 INTRODUCTION

Transformers (Vaswani et al., 2017) have made a great difference in many machine learning fields, particularly leading to significant advances in natural language processing (NLP) and computer vision (CV). It has been shown that the Transformer architecture is capable of handling large-scale datasets, usually with the help of sufficiently many parameters, with bert (Devlin et al., 2018), GPT-3 (Brown et al., 2020) and bart (Lewis et al., 2019) as typical examples, and achieves impressive performance: When the Transformer is trained on enough samples, it often outperforms other competing models such as CNNs (Dosovitskiy et al., 2020). As the potential of Transformers is further tapped, a large number of variants of Transformers have emerged. For example, reformer (Kitaev et al., 2020) reduces the original computation complexity from $O(L^2)$ to $O(L \log L)$ by using locality-sensitive hashing, where L denotes the sequential length. Sparse Transformer (Child et al., 2019) introduces sparse factorizations to reduce the memory cost from $O(L^2)$ to $O(L \log L)$. Linformer (Wang et al., 2020) uses low-rank matrices to approximate the self-attention mechanism to further reduce both the computational cost and memory cost from $O(L^2)$ to $O(L \log L)$.

Despite the vigorous development of architectures, the working principles behind Transformers are still mysteries. The Transformer is often hard to train and we still know little about how it works and how the performance changes when the embedding dimension and depth increase. However, clarifying these problems is quite important because people are developing larger and deeper Transformers with additional training techniques to get better performance. Recently, there have been some preliminary works. (Xiong et al., 2020) and (Popel & Bojar, 2018) numerically illustrated the effect of tuning hyper-parameters on training Transformer. (Huang et al., 2020) explores the difficulty of optimizing the Transformer model, and proposes a new initialization method to benefit the

training of deeper Transformers. (Wang et al., 2019) also investigated on how to train huge Transformer models. (Wang et al., 2022) successfully trained a Transformer with a depth of 1000 layers. In this work, we make an initial attempt to clarify the working mechanism of the Transformer from the perspective of the ID of the Transformer representation.

Generally, people realize that the real-world data such as sounds, texts, images, etc, tends to possess some kind of low-dimensional structures. That is, only a small fraction of dimensions is required to characterize sampled data and underlying target relationships. It is reasonable to consider, for example, the dataset formed by all the $224 \times 224 \times 3$ RGB pictures labeled as dogs. There are in principle $224 \times 224 \times 256 \times 3 = 38535168$ possibilities, but the "intrinsic" number of pictures of dogs recognized by people is usually much less, where considerable similarities are common. Many algorithms and techniques in deep learning formally exploit the ubiquity of these low-dimensional data structures, such as (Hinton & Salakhutdinov, 2006) and (Gonzalez & Balajewicz, 2018). Intrinsic dimension (ID) (Amsaleg et al., 2015),(Houle et al., 2012),(Cutler, 1993) is an important mathematical tool to characterize the geometrical structure of data. It represents the minimal number of parameters required for modeling certain ground truths, hopefully capturing the low-dimensional data structures. In this work, we mainly investigate the data representation by Transformers to uncover different and interesting phenomena via the concept of ID. Our contributions can be summarized as four aspects:

- We analyze the variation of ID for data representation learned by successive Transformer blocks. It appears a *dimension reduction* phenomenon across layers, which can be strengthened by deepening the architecture.
- We show the geometrical structures of Transformers for sequential modeling: it seems that Transformers can only achieve nonlinear dimension reduction when applied to text classification tasks.
- We explore the relationship among embedding dimension (ED), intrinsic dimension (ID) of learned representation and tasks performance, which motivates a straightforward interpretation on the benefit of increasing ED from the perspective of ID.
- We investigate the effect of training dataset reduction via sequential lengths, which shows negligible influence on the IDs and tasks performance. This motivates potentially a guidance for efficiency in practical applications.

2 Related work

TwoNN method. There are lots of works on how to estimate ID of the given datasets. For example, (Fukunaga & Olsen, 1971), (Bruske & Sommer, 1998) are methods based on PCA. (Levina & Bickel, 2004) is a method based on maximum likelihood estimation (MLE). (Costa & Hero, 2004) estimated the ID by using the so-called geodesic-minimal-spanning-tree. (Kégl, 2002) utilized capacity dimension to estimate intrinsic dimension . (Facco et al., 2017) presented an estimation approach named as TwoNN, which takes advantage of the closest and second closest samples to form modeling probability distributions. Considering its computational efficiency, the TwoNN method is adopted in this work to estimate intrinsic dimensions of the representations learned by Transformers.[1]

Mathematically, the TwoNN has the following procedure. Given the dataset $\mathcal{D} = \{x_i\}_{i=1}^N$. For each data point x_i, denote its distance to the closest and second closest sample as $s_{i,1}$ and $s_{i,2}$, respectively. The TwoNN method estimate the ID by modeling the statistics

$$s_i := \frac{s_{i,2}}{s_{i,1}}$$

.

Actually we can model the statistics $s_i := \frac{s_{i,2}}{s_{i,1}}$ by a Pareto distribution (Hussain et al., 2018),(Rootzén & Tajvidi, 2006) and hence get

[1] We refer `https://github.com/ansuini/IntrinsicDimDeep` for some codes.

$$P((s_1, s_2, ..., s_N) \mid d) = d^N \prod_{i=1}^{N} s_i^{-(d+1)}$$

Here, the intrinsic dimension of the dataset $\mathcal{D}$, namely d, can be easily estimated by a common maximum likelihood method. To be more clearly, we give a pipeline of TwoNN method below.

- Ramdomly select N data point, get $\mathcal{D} = \{x_i\}_{i=1}^N$.
- For each x_i in $\mathcal{D}$, calculate the distance of closest and second closest data denoted as $s_{i,1}$ and $s_{i,2}$.
- For each x_i in $\mathcal{D}$, calculate the statistics $s_i := \frac{s_{i,2}}{s_{i,1}}$.
- Estimate d in $P((s_1, s_2, ..., s_N) \mid d) = d^N \prod_{i=1}^N s_i^{-(d+1)}$ by maximum likelihood method and the d is the estimation of ID.

Intrinsic dimension in deep learning. The performance of intrinsic dimension in deep learning architectures has been also studied recently. For example, (Pope et al., 2021) generated fake images by GAN to control the upper bound of ID in order to verify the accuracy of the ID estimation method. (Ansuini et al., 2019) analyzed variation of ID of the representations cross the layer for some classical neural networks, such as the ResNet (He et al., 2016) and VGG (Simonyan & Zisserman, 2014), on the image classification tasks. (Aghajanyan et al., 2020) studied the effect of pre-training on the intrinsic dimension for NLP tasks. However, compared to CNNs and other models applied to computer vision tasks, the related research on sequential models for NLP tasks such as Transformers are still limited despite of its great popularity. The current work aims to fill this gap.

3 RESULTS

We first introduce the experiment setup, then report our results in five aspects.

3.1 EXPERIMENT SETTING

Tasks. To explore the ID of Transformers in a convenient manner, we conduct numerical experiments on the text classification task. The reasons are as follows. First, the text classification is a representative but important task in natural language processing, and has wide applications such as spam detection (Crawford et al., 2015), (Asghar et al., 2020), text style classification (Wu et al., 2019), (Sudhakar et al., 2019), sentiment analysis (Medhat et al., 2014), (Xu et al., 2019) and so on. In addition, a great number of works, such as (Devlin et al., 2018), (Wang et al., 2020), (Shaheen et al., 2020) and so on, have shown a great success of the Transformer architecture applied to the classification task. Moreover, people usually use only a series of encoder blocks (with an additional MLP (Rosenblatt, 1961) block for classification) in the framework of Transformer when conducting the text classification, which implies the convenience for ID analysis. As a comparison, for other tasks where decoders are necessary, e.g. the text generation, one may encounter difficulties for the layer-wise ID computation and analysis.

Datasets. The experiments are performed on three datasets: IMDB (Maas et al., 2011), AG (Zhang et al., 2015) and SST2 (Socher et al., 2013). Among them, IMDB (Maas et al., 2011) is a two classification movie review dataset with 25,000 training data and 25,000 test data; AG (Zhang et al., 2015) is a news articles four classification dataset with 120,000 training data and 7,600 test data; SST2 is a two classification movie review dataset with around 7,000 training data and 2,000 test data.

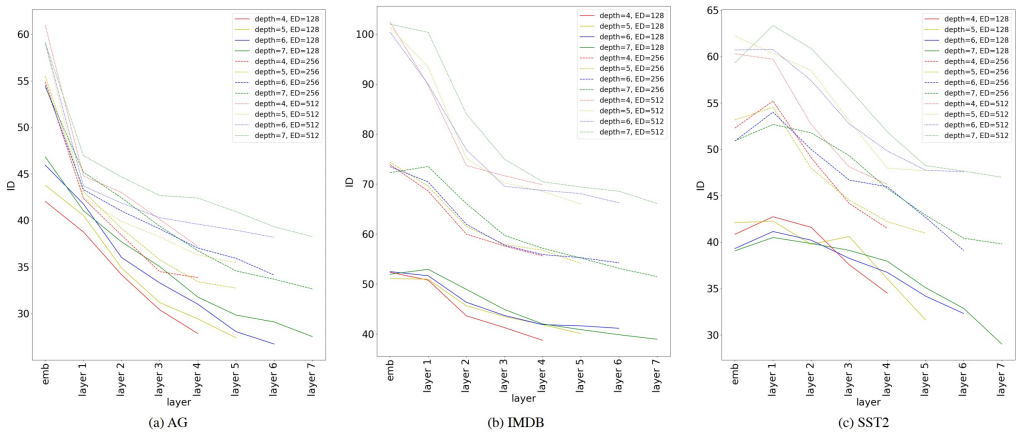
Models. We use the classic Transformer model (Vaswani et al., 2017) with successive encoder blocks to extract features and learn data representation of the full input texts. As a common practice and also for simplicity, pure MLPs (Rosenblatt, 1961) are applied for the terminal classification

layer, which maintains to a large extent the dominating effect on the final performance of the Transformer (encoders).[2]

Goals and related hyper-parameters. The present work aims to study and analyze the relationship between intrinsic dimensions of the representations learned by Transformers and the corresponding classification performance. To achieve this and conduct more comprehensive and rigorous experiments, we set to vary the following hyper-parameters:

- D: depth, i.e. the total number of layers of the Transformer.
- ED: embedding (Mikolov et al., 2013) dimension. For simplicity and by convention (following the classic work (Vaswani et al., 2017)), we set the dimension of each hidden layer to be equal.

That is, the goal is to find out their influences on the ID and classification accuracy. We unfold the analysis from the following aspects.



(a) AG (b) IMDB (c) SST2

Figure 1: The variation of ID with respect to hidden layers.

3.2 The layer-wise variation of intrinsic dimensions

We first study the variation of ID across different layers. For a Transformer model with depth D and hidden dimensions $\{n_l\}_{l=1}^{D}$, every input data is successively mapped into a n_l-dimensional vector space, $l = 1, 2, \cdots, D$. However, the hidden dimension is incapable of characterizing the inherent geometrical structures of data. Here, we use the TwoNN (Facco et al., 2017) method to compute the respective ID per layer.

Following the setting presented in Section 3.1, we perform experiments on three datasets (AG, IMDB and SST2), using the Transformer model with varied depths and embedding dimensions: $D \in \{4, 6, 8\}$, and ED $\in \{128, 256, 512\}$. The intrinsic dimension and its variation with respect to different layers are shown in Figure 1. The horizontal axis of Figure 1 represents each layer, where emb and layer i denote the embedding layer and the i-th encoder layer, respectively. The vertical axis shows the corresponding ID. Every single line in Figure 1 represents the layer-wise variation of ID under a certain hyper-parameters configuration.

From Figure 1, one can straightforwardly obtain the following observations:

- With the increase of depth, the overall intrinsic dimension appears a downward trend. Generally, the ID may only increase at the first encoder layer (see Figure 1 (c)), and then decreases through the following layers, and reaches the minimum at the last layer. This

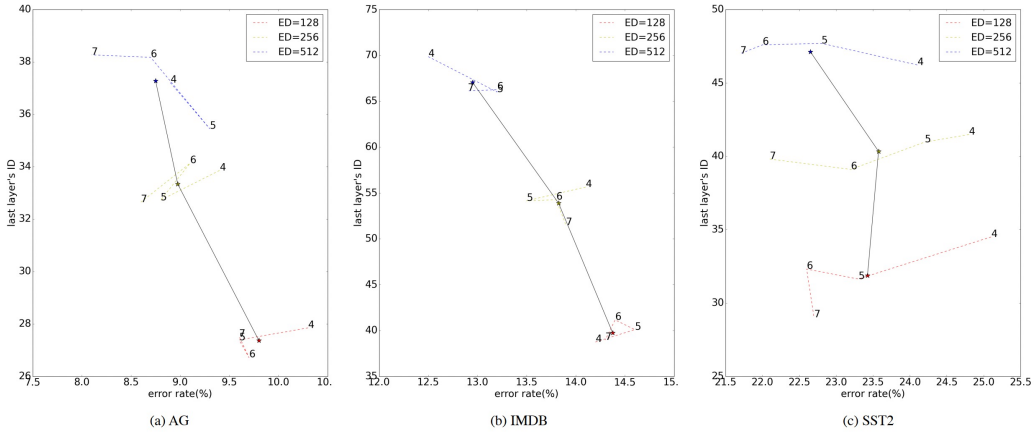
[2]We also refer `https://github.com/lyeoni/nlp-tutorial/tree/master/text-classification-transformer` for some codes.

decrease process of ID across layers can be regarded as a type of *dimension reduction* for Transformers along with the extraction of effective information for the final classification.

- When fixing the model depth on a certain dataset, we find that the intrinsic dimension basically increases with the embedding dimension. In fact, according to Figure 1, the above conclusion always holds for all depths and datasets. This result is natural at the first glance, since when increasing the embedding dimension, the (initial) intrinsic dimension generally increases as well, which leads to an increment of the terminal ID.

To scope the dimension reduction effect in a detailed manner, we can further check the variation of ratios of intrinsic dimension over embedding dimension and hidden dimensions, which is convenient and straightforward since the last two has been set to be equal. It is shown that the ratio is about $O(10^{-1})$ at the embedding layer (approximately 0.1-0.3), while it is notably reduced to $O(10^{-2})$ at the final layer.

3.3 Correlations between intrinsic dimensions and classification accuracy



(a) AG (b) IMDB (c) SST2

Figure 2: The positive correlations between the ID and classification accuracy.

Since ID characterizes the intrinsic (geometrical) structures of data distribution, and the classification performance directly depends on the final representation extracted by the last hidden layer of Transformers, it is reasonable to believe that the ID of the last hidden layer (terminal ID) is correlated with the predicted classification accuracy.

Therefore, we numerically investigate the relationship between terminal ID and corresponding classification error rate for various configurations of hyper-parameters and datasets, see Figure 2. For robustness, the training costs several independent runs using randomized initialization to ensure the convergence. The horizontal axis in Figure 2 represents the error rate of classification and vertical axis shows the ID of representation learned by the last hidden layer. The dash lines connect the results for different depths under a fixed embedding dimension, with the stars as mean values for both terminal ID and error rates.

From Figure 2, one can observe that although all experiments are performed under the same type of hypothesis space (i.e. Transformers), there are remarkable differences in term of classification performance and terminal ID along with the model size (the ED herein). As is shown in the solid lines in Figure 2, the terminal ID basically increases with the embedding dimension, while the classification error changes adversely. Interestingly, the uncovered phenomenon for Transformers applied to NLP tasks is *opposite* to that in (Ansuini et al., 2019), where the terminal ID of CNNs is positively correlated with the classification error rate in image modeling.

The underlying interpretation may be as follows. When the embedding dimension increases, according to the discussion in Section 3.2, we get a larger ID for the representation of input data, resulting in an increment of the terminal ID. Meanwhile, a higher embedding dimension, as well as hidden

dimensions, definitely extend the corresponding hypothesis space for modeling. Hence, it is reasonable to gain a better classification accuracy. Based on this phenomenon, increasing the embedding dimension helps to enhance classification performance via enlarging the intrinsic dimension.

This point may motivate a further extension for practical guidance in applications: one can imagine a principled method to utilize the terminal ID as a *posterior* "indicator" for generalization. That is, by monitoring the variation of terminal ID during training, we may achieve to guarantee better classification performance without using a completely new dataset for both validation and test. This would benefit a lot when encountering limited data in practical applications and hence deserves to explore in the future work.

Remark 1 *Our investigation shows the inherent nature of transformer models on textual tasks. For ViT models, the conclusion conclusion no longer holds as in Table 1. We notice that either the ViT embedding dimension or the ViT depth influence have little effect on ID.*

fixed depth=7	last layer's ID
ED=192	23.52
ED=288	23.07
ED=384	22.51

fixed ED=384	last layer's ID
depth=3	22.02
depth=5	22.33
depth=7	22.51

Table 1: The correlations between the ID and classification accuracy of ViT model on CIFAR-10 dataset. The left subtable shows the results under a fixed depth and the right one shows those under a fixed embedding dimension.

3.4 A PRINCIPAL COMPONENT ANALYSIS VIEWPOINT OF DATA REPRESENTATION

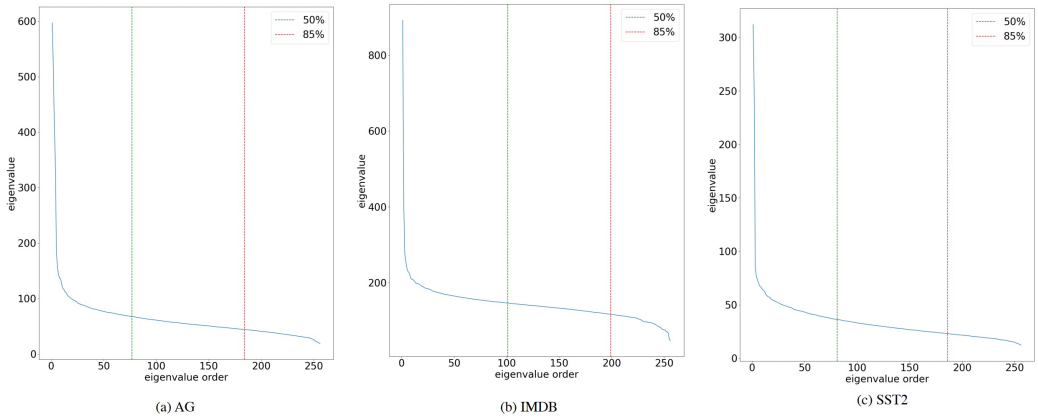


Figure 3: PCA results on data representation of the last hidden layer ($\mathrm{ED} = 256$, $\mathrm{depth} = 6$).

There are many tools to estimate the intrinsic dimension of data representation, such as a series of methods based on principal component analysis (PCA) ((Fukunaga & Olsen, 1971), (Bruske & Sommer, 1998), TwoNN (Facco et al., 2017) and so on). Due to the computational efficiency, the TwoNN algorithm is selected. One can refer to Section 2 for further details.

According to the results shown in Figure 1, we conclude that the intrinsic dimensions of Transformers are much smaller than embedding and hidden dimensions. In this section, we will further illustrate that the data representation learned by Transformers exists on low-dimensional but curved manifolds instead of flat subspaces, hence are incapable of model reduction via linear methods.

To achieve this, we perform the classic PCA (Pearson, 1901) method on the normalized covariance matrix of each layer in Transformers for varied hyper-parameters and all three datasets. Figure 3 shows the results obtained on the last hidden layer. The horizontal axis represents the order of eigenvalues of data representation in a descending sort. The vertical axis shows the value of

corresponding eigenvalues. The green and red vertical dotted lines denote the number of components required to capture 50% and 85% of the variance in data representation. Here, we call the abscissa indicated by the red line as PCA-ID, meaning the "pseudo" intrinsic dimension computed by a direct PCA method.

It is shown that the ID derived from the TwoNN method is much smaller than the PCA-ID. For example, the ID shown in Figure 1 (a) is about 34 for $\mathrm{ED} = 256$ on the AG dataset, while the PCA-ID shown in Figure 3 (a) is about 180 under the same setting, which is 5-6 times larger. Furthermore, the ratio of PCA-ID with respect to embedding dimension is about 0.7-0.9, which is much larger than that of ID (0.05-0.15). The great difference between ID inferred by TwoNN and PCA method shows the strong nonlinearity in the correlations among data samples. Based on the above results, we conclude that the space where data representation is located is totally not a linear subspace, but a certain curved manifold, which prevents people from performing the basic linear model reduction.

3.5 THE EFFECT OF INTRINSIC DIMENSION REDUCTION WITH RESPECT TO DEPTH

	depth=4			depth=8		
	emb layer's ID	last layer's ID	decrease	emb layer's ID	last layer's ID	decrease
ED=128	42.04	27.85	14.19	43.81	27.14	**16.66**
ED=256	54.91	33.86	21.05	55.09	32.44	**22.65**
ED=512	60.93	37.21	23.73	60.96	32.46	**28.50**

Table 2: The ID reduction w.r.t. depth for different embedding dimensions on the AG dataset, where "emb layer" denotes the embedding layer.

	depth=4			depth=8		
	emb layer's ID	last layer's ID	decrease	emb layer's ID	last layer's ID	decrease
ED=128	52.36	38.74	**13.62**	51.14	37.68	13.46
ED=256	73.91	55.65	18.27	72.98	47.30	**25.68**
ED=512	102.41	69.86	32.55	103.33	60.55	**42.78**

Table 3: The ID reduction w.r.t. depth for different embedding dimensions on the IMDB dataset.

	depth=4			depth=8		
	emb layer's ID	last layer's ID	decrease	emb layer's ID	last layer's ID	decrease
ED=128	40.85	34.50	6.34	41.39	30.12	**11.26**
ED=256	52.29	41.48	**10.82**	48.32	37.87	10.45
ED=512	60.30	46.22	14.08	59.8	43.27	**16.53**

Table 4: The ID reduction w.r.t. depth for different embedding dimensions on the SST2 dataset.

In Transformers as well as other neural network architectures, the model depth always plays an important role. A shallow model may have weak representation ability and poor training performance, while a quite deep model may lead to generalization issues such as overfitting (poor test performance) and requires unacceptable computation and memory cost. In this section, we further investigate the dependence of ID variation on the depth.

According to Figure 1, an ID reduction phenomenon across layers appears. To further track its effect with respect to the model depth, one can naturally focus on the gaps between IDs of embedding layers and last hidden layers. Since the relevant ID results have been already shown in Figure 1, we just summarize them in Table 2, 3 and 4.[3] It is straightforward to have the following observations:

- Fix the dataset and embedding dimension, the IDs of embedding layers almost remain the same despite the change of depth.
- Fix the dataset and embedding dimension, the IDs of last hidden layers often decrease significantly with the increase of depth.

[3]Here, the depth 6 case is not included due to space constraints.

- Combining the above two points gives that, the Transformer model shows a larger dimension reduction across layers for deeper architectures.

A direct explanation for the above phenomena is as follows. When fixing the embedding dimension, the learned representation of input data basically remains accordant, which implies similar IDs of embedding layers. Meanwhile, according to Figure 1 and the discussion in Section 3.2, the ID reduction effect may strengthen through more layers, i.e. deeper models.

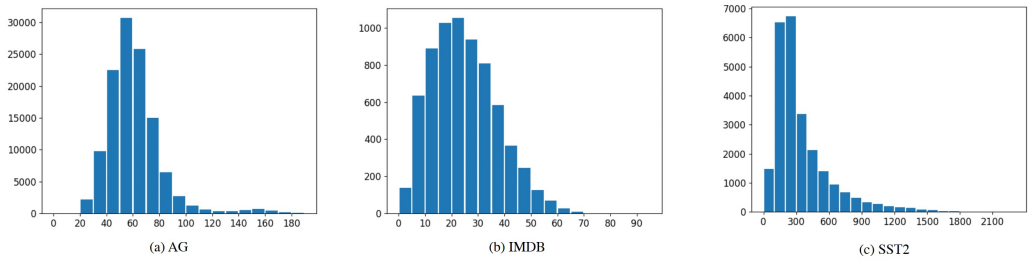


Figure 4: The length distribution of each dataset.

3.6 Instances of data reduction: influence of sequential lengths

In practice, the Transformer model usually possesses massive parameters and hence requires large dataset to guarantee reasonable training performance, resulting in enormous space and time costs. Therefore, it would be meaningful if one can reduce the size of training dataset ("*data reduction*") without significant damages on the test performance. Motivated by this, we aim to investigate the feasibility and procedure of data reduction in the training of Transformers from the viewpoint of intrinsic dimension (of data representation). It is shown that unlike other hyper-parameters such as the embedding dimension, the intrinsic dimension hardly changes with the sequential length of data samples. This motivates potential chances to achieve data reduction by appropriately discarding training samples.

As an initial attempt, we first focus on the sequential length of samples in the training dataset. Specifically, given a dataset, we first sort all the sentences used for training by their lengths (i.e. word counts, see the length distributions shown in Figure 4), and then form a "long-set" by selecting the top 80% longest sentences in the training dataset. Similarly, one can form a corresponding "short-set" by selecting the top 80% shortest sentences, while the original training dataset without any reduction is called as "full-set". In principle, we only remove extreme cases such as too short/long samples, to hopefully avoid affecting the training data and performance. Naturally, the *test datasets remain unchanged*.

		ID of embedding layer			error rate		
		full	long	short	full	long	short
AG	ED=128	43.77	46.23	44.35	9.6	10.3	10.6
	ED=256	55.49	54.07	57.21	8.8	9.5	10.1
	ED=512	58.96	59.22	59.12	9.3	9.2	9.4
IMDB	ED=128	51.10	51.17	48.29	14.6	14.7	15.3
	ED=256	74.38	74.77	71.78	13.5	13.6	14.1
	ED=512	101.34	103.70	104.15	13.2	13.2	13.9
SST2	ED=128	42.10	41.63	42.16	23.3	25.9	26.1
	ED=256	53.17	53.14	54.40	24.2	26.2	25.5
	ED=512	62.23	60.51	62.84	22.8	24.6	24.3

Table 5: The effect of sequential lengths on IDs of embedding layers and classification errors, where the depth of Transformers is fixed as 5.

We conduct experiments on these three training sub-datasets: long-set, short-set and full-set, for various configurations of hyper-parameters (mainly embedding dimensions) and datasets. For each

Transformer model trained under the above settings, we train and record the IDs of embedding layers as well as the final classification error rates (on the full test datasets) in Table 5.

There is only a key phenomenon shown in Table 5. That is, if we check the results in Table 5 row by row, both the IDs of embedding layers and classification errors do not change significantly, despite that each Transformer model is trained on different subsets of the original training dataset. This helps to verify the validity of data reduction in training, at least in the aspect of sequential lengths. It would be valuable in applications where sufficient data is unavailable, and hence worthy of exploration in the future work.

4 CONCLUSION

In this work, we propose a new perspective to understand the mechanism of Transformers, which is related to intrinsic dimensions of data representation. Many interesting phenomena are numerically uncovered to reveal the intricate relationships between intrinsic dimensions and task performance, with respect to hyper-parameters such as depths, embedding dimensions of models and sequential lengths of data. We form a series of empirical conclusions. On one hand, for the influence of hyper-parameters on intrinsic dimensions and classification tasks performance, it is shown that there are positive correlations among embedding dimensions, intrinsic dimensions and the classification accuracy. In addition, the intrinsic dimension reduction across layers exists, which can be strengthened by deepening architectures. On the other hand, for the interaction between model and data (i.e. modeling effect), we give numerical evidence that the data representation learned by Transformers lies on curved manifolds. Furthermore, the data reduction in training can be valid, which possibly motivates efficient methods to utilize data and applicable guidance for practical learning. This deserves further exploration in the future. Certainly, the outlook is not limited. We intend to extend the current research on classification to more general settings, particularly on generative tasks. Moreover, the present work focuses on the basic transformer network, and it is also necessary to further investigate the most commonly-used architectures such as typical pre-trained language models. Apart from that, it is our goal to perform quantitative analysis to complete the theoretical gaps of Transformers and intrinsic dimensions herein.

REFERENCES

Armen Aghajanyan, Luke Zettlemoyer, and Sonal Gupta. Intrinsic dimensionality explains the effectiveness of language model fine-tuning. *arXiv preprint arXiv:2012.13255*, 2020.

Laurent Amsaleg, Oussama Chelly, Teddy Furon, Stéphane Girard, Michael E Houle, Ken-ichi Kawarabayashi, and Michael Nett. Estimating local intrinsic dimensionality. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 29–38, 2015.

Alessio Ansuini, Alessandro Laio, Jakob H Macke, and Davide Zoccolan. Intrinsic dimension of data representations in deep neural networks. *Advances in Neural Information Processing Systems*, 32, 2019.

Muhammad Zubair Asghar, Asmat Ullah, Shakeel Ahmad, and Aurangzeb Khan. Opinion spam detection framework using hybrid classification scheme. *Soft Computing*, 24(5):3475–3498, 2020.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33:1877–1901, 2020.

Jörg Bruske and Gerald Sommer. Intrinsic dimensionality estimation with optimally topology preserving maps. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(5):572–575, 1998.

Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. Generating long sequences with sparse transformers. *arXiv preprint arXiv:1904.10509*, 2019.

Jose A Costa and Alfred O Hero. Geodesic entropic graphs for dimension and entropy estimation in manifold learning. *IEEE Transactions on Signal Processing*, 52(8):2210–2221, 2004.

Michael Crawford, Taghi M Khoshgoftaar, Joseph D Prusa, Aaron N Richter, and Hamzah Al Najada. Survey of review spam detection using machine learning techniques. *Journal of Big Data*, 2(1):1–24, 2015.

Colleen D Cutler. A review of the theory and estimation of fractal dimension. *Dimension Estimation and Models*, pp. 1–107, 1993.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Elena Facco, Maria d'Errico, Alex Rodriguez, and Alessandro Laio. Estimating the intrinsic dimension of datasets by a minimal neighborhood information. *Scientific Reports*, 7(1):1–8, 2017.

Keinosuke Fukunaga and David R Olsen. An algorithm for finding intrinsic dimensionality of data. *IEEE Transactions on Computers*, 100(2):176–183, 1971.

Francisco J Gonzalez and Maciej Balajewicz. Deep convolutional recurrent autoencoders for learning low-dimensional feature dynamics of fluid systems. *arXiv preprint arXiv:1808.01346*, 2018.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016.

Geoffrey E Hinton and Ruslan R Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006.

Michael E Houle, Hisashi Kashima, and Michael Nett. Generalized expansion dimension. In *2012 IEEE 12th International Conference on Data Mining Workshops*, pp. 587–594. IEEE, 2012.

Xiao Shi Huang, Felipe Perez, Jimmy Ba, and Maksims Volkovs. Improving transformer optimization through better initialization. In *International Conference on Machine Learning*, pp. 4475–4483. PMLR, 2020.

Shahzad Hussain, Sajjad Haider Bhatti, Tanvir Ahmad, Muhammad Aftab, and Muhammad Tahir. Parameter estimation of pareto distribution: some modified moment estimators. *Maejo International Journal of Science and Technology*, 12(1):11–27, 2018.

Balázs Kégl. Intrinsic dimension estimation using packing numbers. *Advances in Neural Information Processing Systems*, 15, 2002.

Nikita Kitaev, Łukasz Kaiser, and Anselm Levskaya. Reformer: The efficient transformer. *arXiv preprint arXiv:2001.04451*, 2020.

Elizaveta Levina and Peter Bickel. Maximum likelihood estimation of intrinsic dimension. *Advances in Neural Information Processing Systems*, 17, 2004.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*, 2019.

Andrew Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human language technologies*, pp. 142–150, 2011.

Walaa Medhat, Ahmed Hassan, and Hoda Korashy. Sentiment analysis algorithms and applications: A survey. *Ain Shams Engineering Journal*, 5(4):1093–1113, 2014.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

Karl Pearson. Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572, 1901.

Phillip Pope, Chen Zhu, Ahmed Abdelkader, Micah Goldblum, and Tom Goldstein. The intrinsic dimension of images and its impact on learning. *arXiv preprint arXiv:2104.08894*, 2021.

Martin Popel and Ondřej Bojar. Training tips for the transformer model. *arXiv preprint arXiv:1804.00247*, 2018.

Holger Rootzén and Nader Tajvidi. Multivariate generalized pareto distributions. *Bernoulli*, 12(5): 917–930, 2006.

Frank Rosenblatt. Principles of neurodynamics. perceptrons and the theory of brain mechanisms. Technical report, Cornell Aeronautical Lab Inc Buffalo NY, 1961.

Zein Shaheen, Gerhard Wohlgenannt, and Erwin Filtz. Large scale legal text classification using transformer models. *arXiv preprint arXiv:2010.12871*, 2020.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pp. 1631–1642, 2013.

Akhilesh Sudhakar, Bhargav Upadhyay, and Arjun Maheswaran. Transforming delete, retrieve, generate approach for controlled text style transfer. *arXiv preprint arXiv:1908.09368*, 2019.

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

Hongyu Wang, Shuming Ma, Li Dong, Shaohan Huang, Dongdong Zhang, and Furu Wei. Deepnet: Scaling transformers to 1,000 layers. *arXiv preprint arXiv:2203.00555*, 2022.

Qiang Wang, Bei Li, Tong Xiao, Jingbo Zhu, Changliang Li, Derek F Wong, and Lidia S Chao. Learning deep transformer models for machine translation. *arXiv preprint arXiv:1906.01787*, 2019.

Sinong Wang, Belinda Z Li, Madian Khabsa, Han Fang, and Hao Ma. Linformer: Self-attention with linear complexity. *arXiv preprint arXiv:2006.04768*, 2020.

Chen Wu, Xuancheng Ren, Fuli Luo, and Xu Sun. A hierarchical reinforced sequence operation method for unsupervised text style transfer. *arXiv preprint arXiv:1906.01833*, 2019.

Ruibin Xiong, Yunchang Yang, Di He, Kai Zheng, Shuxin Zheng, Chen Xing, Huishuai Zhang, Yanyan Lan, Liwei Wang, and Tieyan Liu. On layer normalization in the transformer architecture. In *International Conference on Machine Learning*, pp. 10524–10533. PMLR, 2020.

Guixian Xu, Yueting Meng, Xiaoyu Qiu, Ziheng Yu, and Xu Wu. Sentiment analysis of comment texts based on bilstm. *Ieee Access*, 7:51522–51532, 2019.

Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. *Advances in Neural Information Processing Systems*, 28, 2015.