Growing Pre-Trained Models for Faster Convergence

Apoorv Walia
Department of Computer Science, Rice University
apoorv.walia@rice.edu

Abstract

The goal of this research is to increase the size of small to medium-sized models while reducing computational cost. By finding an optimal initialization for a larger model using a transformation of the weights of a smaller model, we aim to achieve faster convergence. Our approach involves focusing on increasing the width of the model through the use of orthonormal matrices. While promising results were obtained with this approach, recent research has shown that a linear mapping of rows and columns of the smaller model can be used to create a larger and deeper models. Future work will involve exploring the use of orthonromal expansion for transformer models and exploiting the low rank properties of the resulting weight matrices.

Link to Code

1. Introduction

Deep neural networks have revolutionized the field of machine learning and have shown great success in a variety of applications, such as computer vision, natural language processing, and speech recognition. However, as the size and complexity of these models increase, so does the computational cost required to train them. This has led to the development of techniques for reducing the computational burden of deep neural networks, such as pruning, quantization, and compression. Another approach to reducing the compute required for training larger models is to grow smaller models to larger sizes. This involves finding an optimal initialization for the larger model using some transformation of the weights of the smaller model. In this paper, we explore methods for growing small/medium size models to larger sizes.

Increasing Width of the Model: My initial experiments involved increasing the width of the model. I did this by multiplying the weight matrices with orthonormal matrices to increase their size. The intuition behind this is that multiplication with orthonormal matrices changes the size of the matrix but preserves the transformation induced by this matrix. However, are the transformations performed by the original matrix still useful in the higher dimension space? My work shows that these transformations still remain meaningful and lead to good initialization for the larger model. This method showed promising results and reduced the computational cost required to train larger models.

Generating orthonormal matrices is a time-consuming process, but it is a one-time cost. We can generate and save these matrices for future use. In addition, we can easily chop of columns/rows of a large orthonormal matrix to create smaller orthornormal without the need of generating new ones. However, increasing the width of the model alone may not be sufficient for achieving optimal performance. It may be necessary to increase the depth of the model as well.

Increasing Depth of the Model: Recent work from MIT (Wang et al., 2023 [9]) has demonstrated a method for increasing the depth of the model by learning a linear mapping of rows and columns of the smaller model to create a larger model. This approach has been shown to reduce the computational cost required to train larger models. Chen et al., 2021 [2], however, use an interleaving technique that leverages the fact that adjacent layers learn similar representations. This approach also shows promising results. I propose a method of depth expansion based on this intuition. We could double the number of layers in the network by multiplying each layer with 2 different sets of orthonormal matrices. This will result in two unique layers that perform similar transformations. I plan to explore this approach further in future work.

2. Related Work

Several techniques have been proposed for efficient training of transformers. These include mixed precision training, large batch optimization, distributed training, and dropping layers or tokens [Hou et al., 2022 [5]]. Knowledge inheritance has also been explored, which uses knowledge distillation during pretraining to efficiently learn larger transformers [Qin et al., 2021 [7]]. Progressive training, which first trains a small transformer with few layers and

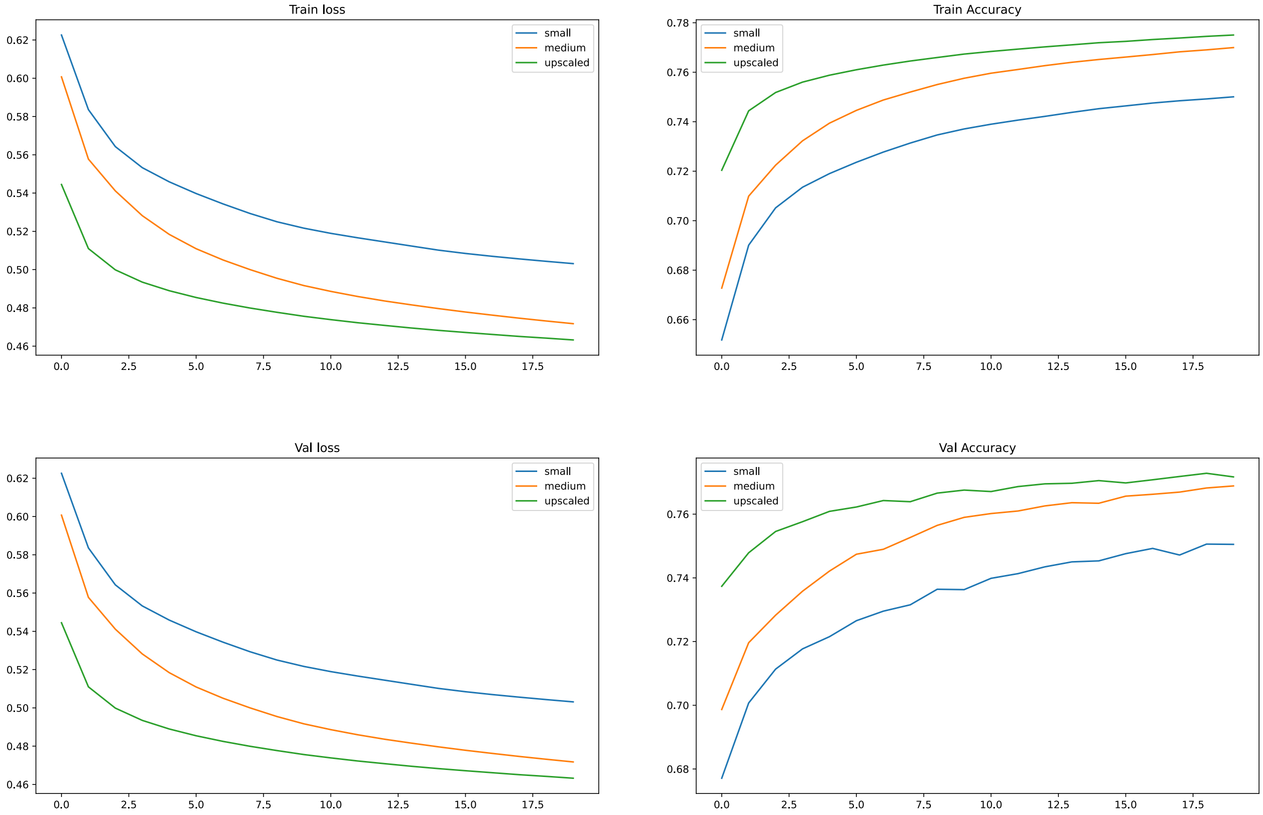
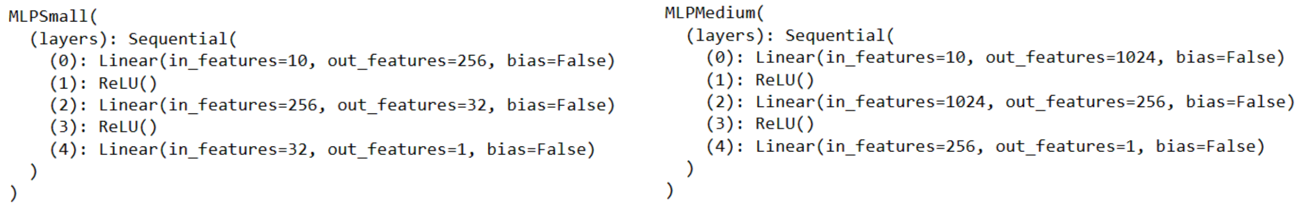


Figure 1: Growth results for Higgs Dataset

then gradually expands by stacking layers, has also been applied to accelerate transformer training [Shen et al., 2022 [8]]. Net2Net uses function-preserving transformations to grow width by copying neurons and depth by using identity layers [Chen et al., 2015 [3]], and bert2BERT extends Net2Net to transformers [Chen et al., 2021 [2]]. Work on neural network initialization has also been discussed, including controlling the norm of the parameters or replacing the normalization layers [Brock et al., 2021 [1]]. MetaInit proposes an automatic method that optimizes the norms of weight tensors to minimize the gradient quotient on minibatches of random Gaussian samples [Dauphin & Schoenholz, 2019 [4]], while GradInit learns to initialize larger networks by adjusting norm of each layer [Zhu et al., 2021 [10]]. Finally, structured matrices have been used to replace dense weight matrices for reducing training and inference computation cost. Examples include sparse and low rank matrices, Chebyshev matrices, Toeplitz matrices, Kronecker-product matrices, and butterfly matrices.

In particular, this work attempts to build on the following works. Wang et al., 2023 [9] describes an approach for accelerating transformer training by learning to grow pretrained transformers. The authors propose to learn to linearly map the parameters of a smaller model to initialize a larger model. For tractable learning, they factorize the linear transformation as a composition of (linear) width- and depth-growth operators and further employ a Kronecker factorization of these growth operators to encode architectural knowledge. Extensive experiments across both language and vision transformers demonstrate that their learned Linear Growth Operator (LiGO) can save up to 50% computational cost of training from scratch while also consistently outperforming strong baselines that also reuse smaller pretrained models to initialize larger models.

Chen et al., 2021 [2] aims to improve the efficiency of pre-training large language models by transferring knowledge from an existing smaller pre-trained model to a larger one. The method consists of two main components: parameter initialization and a two-stage pre-training process. For parameter initialization, the authors propose two strategies: Function Preserving Initialization (FPI) and Advanced Knowledge Initialization (AKI). FPI aims to make the initialized target model have the same function as the source model by duplicating and stacking the parameters of the



```
MLPSmall(
  (layers): Sequential(
    (0): Linear(in_features=10, out_features=256, bias=False)
    (1): ReLU()
    (2): Linear(in_features=256, out_features=32, bias=False)
    (3): ReLU()
    (4): Linear(in_features=32, out_features=1, bias=False)
  )
)
```

```
MLPMedium(
  (layers): Sequential(
    (0): Linear(in_features=10, out_features=1024, bias=False)
    (1): ReLU()
    (2): Linear(in_features=1024, out_features=256, bias=False)
    (3): ReLU()
    (4): Linear(in_features=256, out_features=1, bias=False)
  )
)
```

Figure 2: Model Architecture

existing smaller model. AKI expands new matrices based on not only the parameters of the same layer but also the parameters of the upper layer in the source model. After parameter initialization, a two-stage pre-training process is applied to further accelerate the training process. In the first stage, sub-structures of the target model are trained in a random manner to make the complete model converge at low cost. In the second stage, traditional full-model training is performed. The authors demonstrate through experiments that their method can save a significant amount of computational cost compared to learning from scratch and other progressive training methods while achieving similar performance on downstream tasks. Additionally, their method is shown to be generic and applicable to different types of pre-trained models.

Chen et al., 2016[3] introduces techniques for rapidly transferring the information stored in one neural net into another neural net. The main purpose is to accelerate the training of a significantly larger neural net. The authors propose a Net2Net technique that accelerates the experimentation process by instantaneously transferring the knowledge from a previous network to each new deeper or wider network. Their techniques are based on the concept of function-preserving transformations between neural network specifications. This differs from previous approaches to pre-training that altered the function represented by a neural net when adding layers to it.

3. Approach

To address the problem of increasing the width and depth of weight matrices, I propose a novel approach based on orthonormal matrix multiplication. Specifically, I expand the width of a weight matrix of size m x n to 2m x 2n by multiplying it with two orthonormal matrices L (size 2m x m) and R (size n x 2n) as follows: W' = L x W x R, where W' is the expanded weight matrix. To increase the depth of a neural network, we can apply this operation twice for each layer with a different set of orthonormal matrices, resulting in two weight matrices that perform similar transformations in the new higher dimensional space.

The key contribution of this work is to investigate the potential advantages of using this technique for weight initialization and preserving the original transformations in the higher dimensional space. I hypothesize that this approach may lead to better performance and faster convergence, as it enables the network to leverage the learned representations from the lower-dimensional space and apply them to the higher-dimensional space.

To evaluate the effectiveness of our approach, I conduct a series of experiments on various benchmark datasets using a 3 Layer fully connected model architecture. The architecture of the models used for the experiments can be seen in Figure 2. I attempt to grow a pre-trained MLPSMall to the size of MLPMedium. My results show that this approach achieves comparable or better performance than other methods, indicating the potential utility of orthonormal matrix multiplication for weight initialization.

One advantage of our approach is its simplicity and efficiency. Unlike other initialization techniques that require complex structural transformations or training, our method only requires storing two large orthonormal matrices, which can be applied to all layers of a network. This makes it easy to implement and applicable to a wide range of models.

4. Experiments and Results

Experiments were conducted using several datasets to evaluate the performance of the proposed method. The YearMSD dataset, which contains audio features and metadata for a million contemporary popular music tracks (Bertin-Mahieux et al., 2011), was used to evaluate the method's performance on music data. The HIGGS dataset, which contains simulated data for signal and background processes for the Large Hadron Collider (Baldi et al., 2014a), and the SUSY dataset, which contains simulated data for supersymmetric particles (Baldi et al., 2014b), were used to evaluate the method's performance on high-energy physics data. The Covtype dataset, which contains cartographic variables for predicting forest cover types (Blackard & Dean, 1999), was used to evaluate the method's performance on environmental data. Finally, the Delicious dataset, which contains social bookmarking data (Zubiaga & Ji, 2014), was used to evaluate the method's performance on social media data. The results of these experiments are presented in the following sections and in the appendix.

In the experiments, two architectures of different sizes, "small" and "medium," were trained from scratch. The

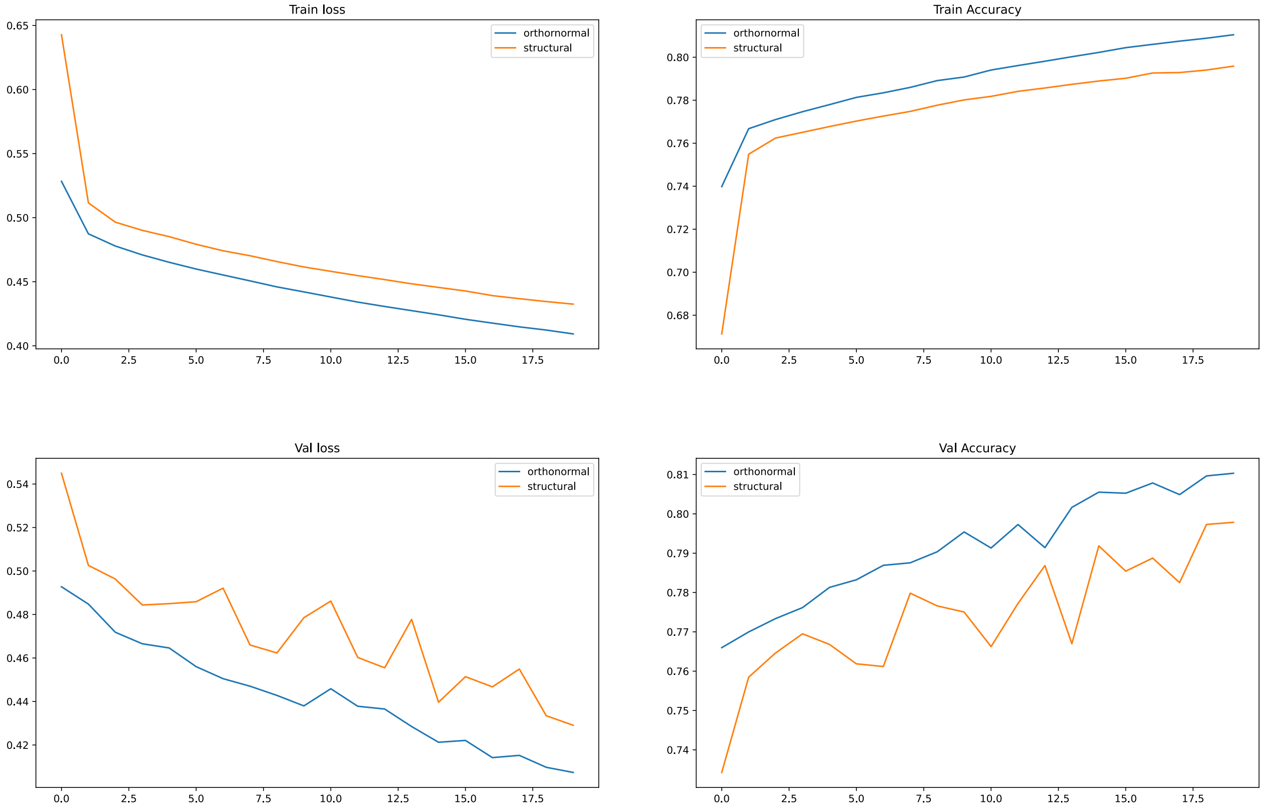


Figure 3: Comparison of Orthonormal vs Structural re-construction methods

"medium" architecture trained from scratch served as the baseline for comparison. The weights of the small model were then transformed using the techniques described above, and the convergence rate was compared to the baseline. Another experiment was conducted where the transformations were performed in a different manner. The rows/columns with the highest norms, which serve as a metric of importance, were taken from the layers of the smaller-sized model. These rows were then flattened into a single column, and the same was done for the adjacent layer, which was flattened into a row. The new layer was created as the outer product of this generated row and column. This was an attempt to recreate the method of linear combinations of rows and columns to expand the weight matrix used by Wang et al., 2023 [9]. In results presented in Figure 3, I show that weight matrices expanded by the orthonormal projection method outperform those created using linear combinations of rows and columns. However, this naive method may not be a fair comparison of the two methods.

The results of the experiments demonstrate the effectiveness of the proposed method for widening neural networks. In particular, this method led to significantly faster training times, especially for the HIGGS dataset, where we were able to achieve almost a 40% decrease in the number of iterations required for convergence. Additionally, my experiments on MLPs showed that linear combinations of rows and columns resulted in worse initialization than the orthonormal approach. One of the key benefits of our approach is its versatility; it can be applied to any model using a simple function, without the need for a re-learning process.

5. Conclusion and Future Work

In conclusion, I present a novel approach for weight matrix expansion based on orthonormal matrix multiplication, and demonstrate its effectiveness for weight initialization in neural networks. My results suggest that this technique could be a promising alternative to existing weight initialization methods, particularly for models with larger width and depth. Further research is needed to explore its full potential and investigate its theoretical underpinnings.

The findings of this study indicate that the concept of width expansion holds considerable potential. Going forward, it is imperative to explore the combined effects of both depth and width expansion on transformer networks. Additionally, it is worth exploring the utilization of low-rank properties of weight matrices to achieve faster training. The present research conducted on Multilayer Perceptrons (MLPs) has provided a basis for extending this approach to transformer architectures, where previous research indicates that the benefits may be even greater. Future investigations can continue to build on these initial findings to unlock the full potential of width expansion and related techniques in deep learning.

Furthermore, there is potential to achieve further improvements by combining the state-of-the-art techniques presented by Wang et al., 2023 [9] with the Low-Rank Approximation (LoRA) approach proposed by Hu et al., 2021 [6]. Currently, there is a lack of research on the impact of this combination, and it presents a promising avenue for future exploration. It is reasonable to anticipate that combining these approaches could lead to even greater performance gains, and further investigation into this area could be a valuable contribution to the field of deep learning.

References

[1] A. Brock, S. De, S. L. Smith, and K. Simonyan. High-performance large-scale image recognition without normalization, Feb 2021.

[2] C. Chen, Y. Yin, L. Shang, X. Jiang, Y. Qin, F. Wang, Z. Wang, X. Chen, Z. Liu, and Q. Liu. bert2bert: Towards reusable pretrained language models, 2021.

[3] T. Chen, I. Goodfellow, and J. Shlens. Net2net: Accelerating learning via knowledge transfer, 2015.

[4] Y. N. Dauphin and S. Schoenholz. Metainit: Initializing learning by learning to initialize, Jan 1970.

[5] L. Hou. Arxiv:2203.13240v1 [cs.cl] 24 mar 2022.

[6] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen. Lora: Low-rank adaptation of large language models, Oct 2021.

[7] Y. Qin, Y. Lin, J. Yi, J. Zhang, X. Han, Z. Zhang, Y. Su, Z. Liu, P. Li, M. Sun, and et al. Knowledge inheritance for pre-trained language models, Apr 2022.

[8] S. Shen, P. Walsh, K. Keutzer, J. Dodge, M. Peters, and I. Beltagy. Staged training for transformer language models, Mar 2022.

[9] P. Wang, R. Panda, L. T. Hennigen, P. Greengard, L. Karlinsky, R. Feris, D. D. Cox, Z. Wang, and Y. Kim. Learning to grow pretrained models for efficient transformer training, 2023.

[10] C. Zhu, R. Ni, Z. Xu, K. Kong, W. R. Huang, and T. Goldstein. Gradinit: Learning to initialize neural networks for stable and efficient training, Dec 2021.

Appendix

In this section, I present results of the growth operation on several data-sets.

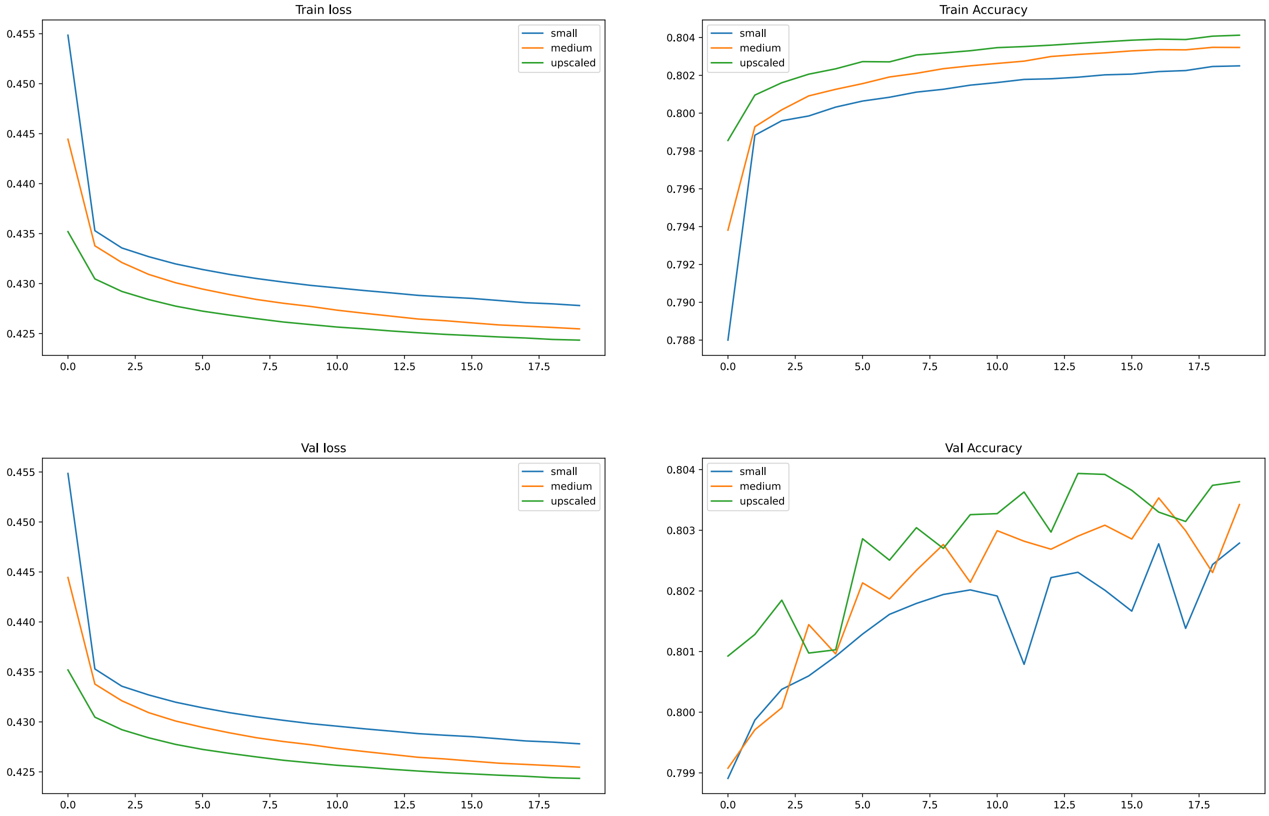


Figure 4: Growth results for SUSY Dataset

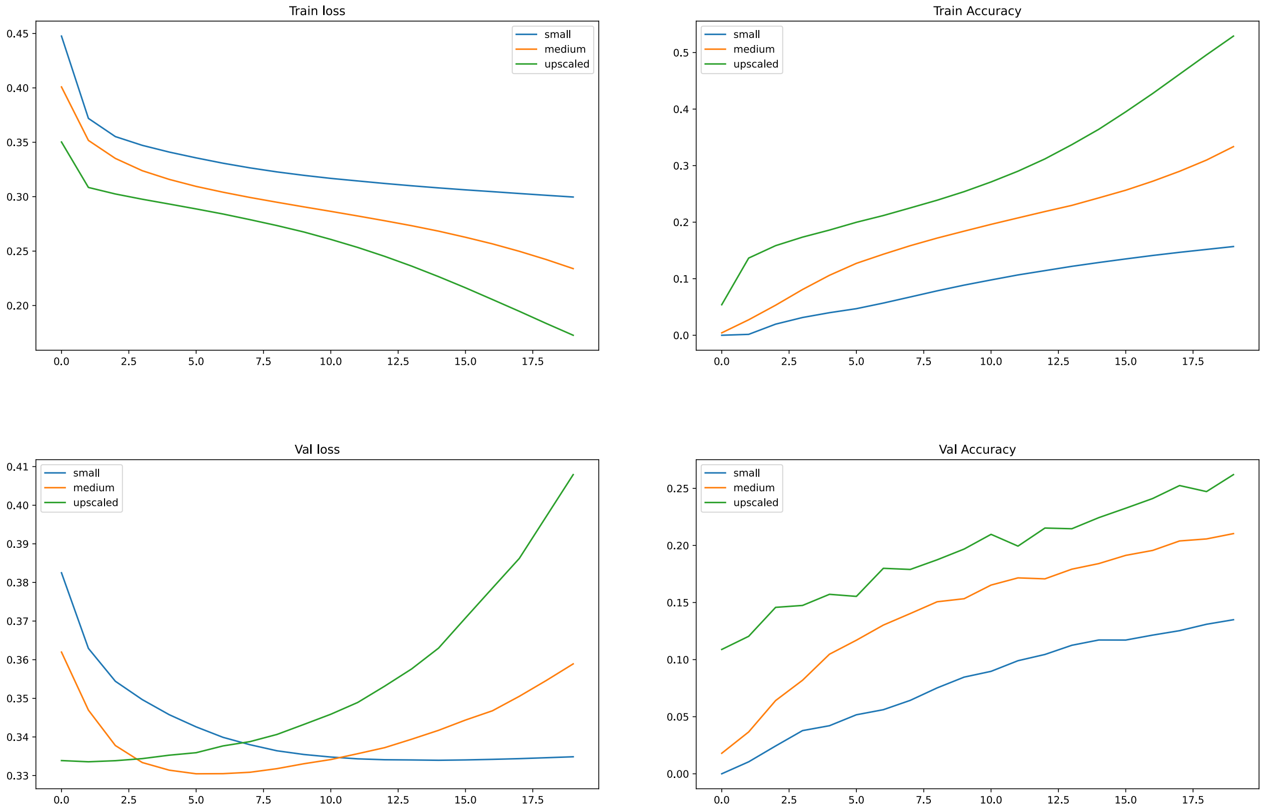


Figure 5: Growth results for Delicious Dataset

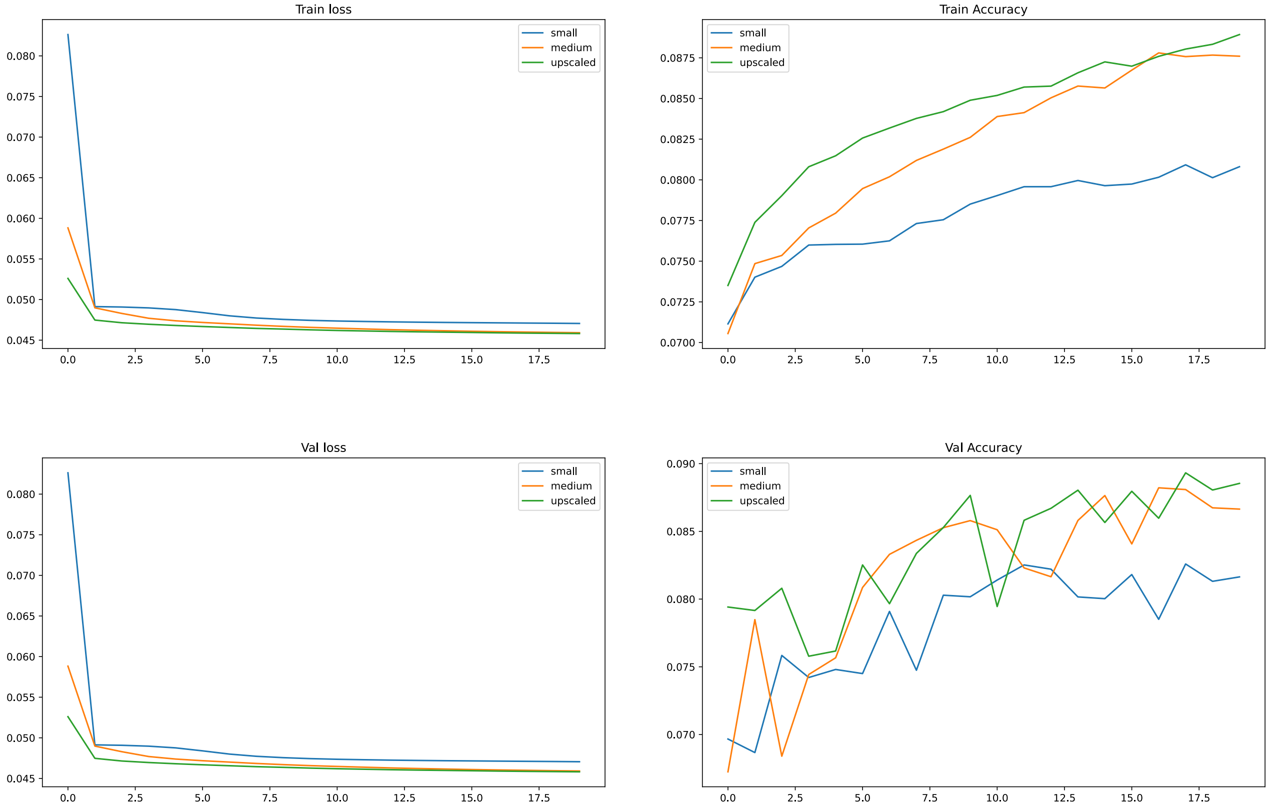


Figure 6: Growth results for YearMSD Dataset