MEASURING THE INTRINSIC DIMENSION OF OBJECTIVE LANDSCAPES BY CONSTRAINING WEIGHTS ON AN ORIENTED HYPERELLIPSOID

August 12, 2022

Arturo Maiani

Abstract

This projects deals with the study of the intrinsic dimension of objective landscapes through constraints applied in the space of network parameters.
https://github.com/ArturoMaiani/DLAI-project/tree/main

1. Introduction

In this project we expand on the work of (Li et al., 2018) in which they trained DNNs whose parameters were constrained in a randomly oriented linear subspace. In this work we follow a research direction suggested by (Li et al., 2018) which consisted in exploring the use of nonlinear constraints instead of linear subspaces. The choice for such manifolds has been done in favor of an hyperellipsoid.

2. Related work

The pioneeristic work from (Li et al., 2018) has started this research direction, by training on fixed randomly oriented subspaces. (Gressmann et al., 2020) changed the subspace orientation at each SGD step, obtaining higher classification accuracy but losing the global structure of the manifold on which the weights are traveling.

3. Method

A DNN is a function of some weights $w \in \mathcal{R}^N$ but in this case we want to make sure that $w = f(\theta)$ with $\theta \in \mathcal{R}^d$ and $d << N$. In (Li et al., 2018) a linear mapping of the form $w = \theta_0 + P\theta$ has been used, where P is a $N \times d$ matrix and θ_0 is an offset. In this work an hyperellipsoid has been tested:

$$\mathbf{w} = \theta_0 + \big[P_1 \mid P_2 \mid ... \mid P_d\big]\theta$$

$$\mathbf{w} = \theta_0 + \\ + \big(P_1, P_2, ..., P_{d+1}\big) \begin{pmatrix} r_1 cos(\theta_1) \\ ... \\ r_d sin(\theta_1)sin(\theta_2)...cos(\theta_d) \\ r_{d+1} sin(\theta_1)sin(\theta_2)...sin(\theta_d) \end{pmatrix}$$

As it can be seen with an hyperellipsoid we get a free extra column for the matrix P with the same number of free parameters. This may be insignificant since $d >> 1$, but still is an interesting fact.

3.1. Clever way to choose P and σ's (not used for computational reasons)

The following reasoning was taken from the Dynamic Linear Dimensionality Reduction explained at page 4 of (Li et al., 2021). In particular a clever way to choose $\sigma_1, ..., \sigma_N$ and P would be by observing the initial steps of the optimization process of the unconstrained network, as follows. Perform T steps of e.g. SGD on the unconstrained network and collect the weight trajectory in a matrix. Then by using some numerically optimized methods such as `scipy.sparse.linalg.eighsh(A,d)` we can find the d first eigenvalues and eigenvectors of the PCA analysis for such trajectory.
Then such eigenvectors are put as columns of the matrix P such that the first one from the left corresponds to the biggest eigenvalue and the one to the far right to the smallest one:

$$P = \big[v_1 \mid v_2 \mid ... \mid v_d\big]$$

The same matrix P could be used for the linear subspace method while for the Hyperellipsoid we just need to put the radius $r_1, ..., r_N$ equal to the ordered eigenvalues $\sigma_1, ..., \sigma_N$.
As explained in section 4.1 this method could not be tested since google colab available RAM was not sufficient to

Email: Arturo Maiani <maiani.1738271@studenti.uniroma1.it>.

Deep Learning and Applied AI 2022, Sapienza University of Rome, 2nd semester a.y. 2021/2022.

store matrices of dimension $190'000 \times 190'000$. This is not much of a problem since we can still obtain nice results choosing P as a matrix whose entries are gaussian of zero mean and SD 0.1, checking that its rank is equal to d.

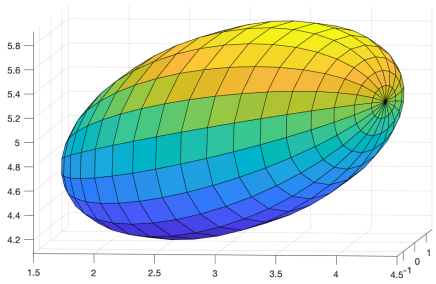

Figure 1. Example of 3-D ellipsoid

4. Experimental results

The results presented in this section are based on the classification task of the MNIST dataset. As a starting point the dataset is composed of 60'000 images for training and 10'000 images for testing. The DNN network is a MLP composed of 2 hidden layers of 200 and 200 units. The input layer has 784 neurons and the output layer has 10, since we want to classify numbers from 0 to 9.
First of all a standard model is trained: the accuracy is 98 % and has been obtained by training for 5 epochs, using the Adam optimizer, a linear scheduler for the learning rate from 10^{-2} to 10^{-4} and using dropout (20 %) after the first layer. In addition in order to avoid the "dying ReLU" problem the activation function "LeakyReLU" has been used. The training time was very fast: 13 seconds.

4.1. Linear subspace

The first constraint taken into exam is that of the linear subspace, proposed by (Li et al., 2018). The way in which the intrinsic dimension is determined is through a threshold of 90 % w.r.t. the performance of the unconstrained network, which consists of a threshold accuracy of 88%.
In addition simulations have shown that the offset vector θ_0 was not necessary. The constrained networks have been trained for two epochs with the Adam optimizer with a fixed learning rate of $5e^{-4}$.
Since matrix P is generated randomly, we can see that we got lucky for $d = 725$, obtaining a staggering 96% accuracy. In general we can conclude that for the MNIST dataset the Intrinsic Dimension falls between 700 and 750, confirming the results of (Li et al., 2018). In addition, the training time for a single epoch is around 2 minutes.

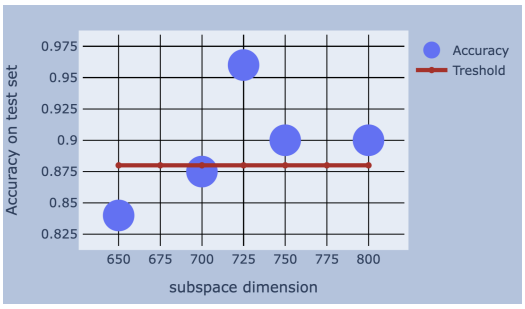

Figure 2. Test Accuracy vs Subspace Dimension

4.2. Hyperellipsoid

Such manifold is not as simple as a linear subspace, this fact makes training quite longer. The matrix P in this case has been chosen to have entries of Gaussian distribution with zero mean and 0.08 SD. In addition the radius for each direction is obtained from a uniform distribution between $[0.015, 0.3]$. The Adam optimizer with learning rate equal to $6e^{-3}$ has proven to give some results in this case.

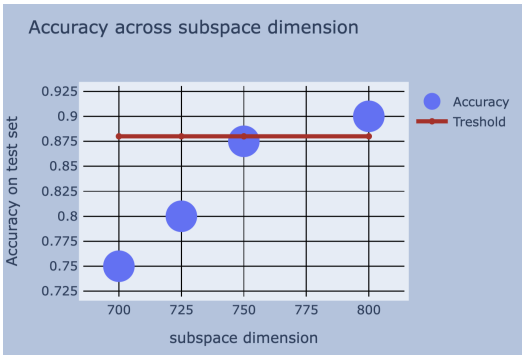

Figure 3. Test Accuracy vs Subspace Dimension

Such models have proven to be much harder to train since 1) the time for training on a single epoch becomes 3 minutes and 2) the training required more epochs. In addition the model is very sensitive to random choice of parameters, meaning that in many cases even with $d = 800$ the test accuracy was very low. In any case for $d = 750$ we got an accuracy which can be considered almost identical to the threshold, hence confirming that the intrinsic dimension is the same for a linear constraint and an hyperellipsoid.

5. Conclusions

In this work a linear constraint and a hyperellipsoid constraint have been examined in the context of the investigation of intrinsic dimensionality (ID) of a deep learning task. The results have shown that a similar ID has been found for the two, suggesting that ID could be invariant to the choice

of the constraint.

References

Gressmann, F., Eaton-Rosen, Z., and Luschi, C. Improving neural network training in low dimensional random bases. *CoRR*, abs/2011.04720, 2020. URL https://arxiv.org/abs/2011.04720.

Li, C., Farkhoor, H., Liu, R., and Yosinski, J. Measuring the intrinsic dimension of objective landscapes. *CoRR*, abs/1804.08838, 2018. URL http://arxiv.org/abs/1804.08838.

Li, T., Tan, L., Tao, Q., Liu, Y., and Huang, X. Low dimensional landscape hypothesis is true: Dnns can be trained in tiny subspaces, 2021. URL https://arxiv.org/abs/2103.11154.