UVA CS 6316: Machine Learning : 2019 Fall
Course Project: Deep2Reproduce @
https://github.com/qiyanjun/deep2reproduce/tree/master/2019Fall

Towards Understanding Learning Representations: To What Extent Do Different Neural Networks Learn the Same Representation

Authors: Liwei Wang, Lunjia Hu, Jiayuan Gu, Yue Wu, Zhiqiang Hu, Kun He, John Hopcroft

Reproduced by : Team Falcon
Members:
Akhila Ranga (ar8aq)
Debarati Sarkar (ds5yb)
Mohit Sudhakar (ms5sw)
Shivani Dewan (sd4dw)

Motivation

- The success of deep neural networks is widely attributed to learning good representations.
- Despite being highly intuitive, there is lack of theoretical and and systematic approach with which one can quantitatively characterize what representations deep neural networks are learning.
- In this paper, authors try to get a better understanding of these representations. They develop a theory that gives a complete characterization of the structure of neuron activation subspace matches.

Background

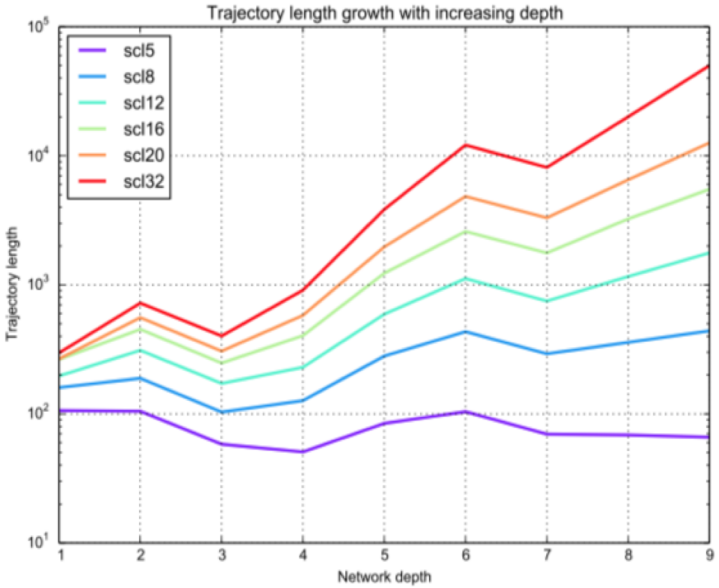
- Dauphin et al., [2014], observed that training the same neural network from different initializations frequently gives similar performance. This raises the question if differently initialized networks learn similar representations as well.
- However, there is a lack of theory and systematic analysis categorizing the representations learned by deep neural networks.
- The authors analyze if representations learned by deep neural networks are similar when trained from different initializations.

Related Work

1. Li et al.[2016] breaks down the concept of similarity into one-to-one , one-to-many and many-to-many mappings. They applied a sparse weighted Lasso model to study one-to-one mappings and figured that the entire correspondence can be decoupled to a series of correspondence between smaller neuron clusters. Spectral clustering algorithm was applied to find many-to-many mappings.

1. Raghu et al. [2017] - The activation vector shows the neuron's responses over a finite set of inputs which act as representation of a single neuron. Further, the representation of a neuron cluster is denoted by the subspace spanned by activation vectors of neurons in the cluster.

An Intuitive Figure Showing WHY Claim



- This figure shows how the trajectory (output of the network as the input sweeps along a one dimensional path) changes with different initializations scales as a trajectory is propagated through a convolutional architecture for CIFAR - 10 with ReLU activations [Raghu et al., 2017].
- This sets the background of the current paper where we study how similar are two neural networks based on different initializations.

Claim / Target Task

- To study the similarity between representations learned by two networks with identical architecture but trained from different initializations and give a complete analysis for the structure of matches.
- To propose efficient algorithms for finding the maximum match and the simple matches in the neuron subspace match model.
- To demonstrate that representations learned by most convolutional layers exhibit low similarity in terms of subspace match.

Proposed Solution

- The paper proposes a solution to find out how similar are the representations learned by two networks.
- The activation vector shows the neuron’s responses over a finite set of inputs.
- The subspace spanned by activation vectors of neurons in the cluster acts as the representation of a cluster of neurons
- The similarity between neurons is modeled as the matches of subspaces spanned by activation vectors of neurons.
- The core concepts for similarity are maximum match and simple match which describe the whole similarity and the minimal units of similarity between sets of neurons in two networks respectively, collectively giving a complete characterization. Efficient algorithms are proposed to find maximum match and simple matches.

Implementation

DEFINITIONS:

Say $\mathcal{X}$ and $\mathcal{Y}$ are the set of neurons in the same layer of two networks with identical architecture but trained from different initializations. Let epsilon $\epsilon \in [0, 1)$.

Activation vector of neuron x over d inputs is $z_x := (z_x(a_1), z_x(a_2), \cdots, z_x(a_d))$, where $z_x(a_1)$ is output of neuron x over input a_1

span(z_X) is the representation of subset of neurons $X \subseteq \mathcal{X}$, measured by the subspace spanned by the activation vectors of the neurons therein

$$\text{span}(\mathbf{z}_X) := \{ \sum_{\mathbf{z}_x \in \mathbf{z}_X} \lambda_{\mathbf{z}_x} \mathbf{z}_x : \forall \lambda_{\mathbf{z}_x} \in \mathbb{R} \}.$$

Implementation

ϵ-approximate match - If $X \subseteq \mathcal{X}$ and $Y \subseteq \mathcal{Y}$ then (X, Y) is an ϵ-approximate match if,

$\forall x \in X, \text{dist}(z_x, \text{span}(z_Y)) \leq \epsilon|z_x|$ and $\forall y \in Y, \text{dist}(z_y, \text{span}(z_X)) \leq \epsilon|z_y|$

1. Algorithm 1 - to Find the maximum match:
Initialize the maximum match (X^*, Y^*) to be $X^* = \mathcal{X}$, $Y^* = \mathcal{Y}$.
If there is $x \in X^*$ such that x cannot be linearly expressed by zY^* (i.e., $\text{span}(z_{Y^*})$) within error ϵ, we remove x from X^*. The same is applicable for some $y \in Y^*$.
X^* and Y^* are repeatedly updated in this way until no such x, y can be found.

2. Algorithm 2 - to output v-minimum match for a neuron v.
The algorithm starts from (X_v,Y_v) being the maximum match and iteratively finds a smaller (X_v,Y_v) keeping $v \in X_v \cup Y_v$ until further reducing the size of (X_v,Y_v) would have to violate $v \in X_v \cup Y_v$.

Implementation

1. The maximum matching similarity is introduced to measure the overall similarity between sets of neurons. The maximum matching similarity s under error e is defined as

$$s(e) = \frac{|X^*| + |Y^*|}{|X| + |Y|}$$

1. The minimal matching similarity is also calculated using the same measure.

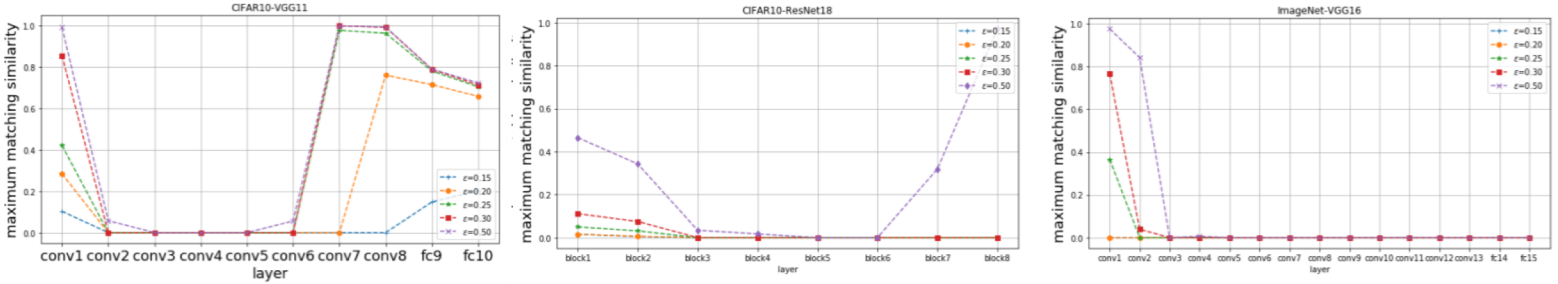
Data Summary

- Experiments in the paper are conducted on architectures of VGG and ResNet on the dataset CIFAR10 and ImageNet.
- The neurons are activated by ReLU.
- In these experiments multiple networks are initialized with different random seeds.
- We have run experiments on combination of VGG16 and CIFAR10 on 500 random samples.
- Random seeds of 0 and 1 are used.

Experimental Results - Paper

1. Maximum Match

- The similarity values show little variance among different pairs, which indicates that this metric reveals a general property of network pairs.
- Layers close to the output sometimes exhibit high similarity.
- There is also relatively high similarity of layers close to the input.

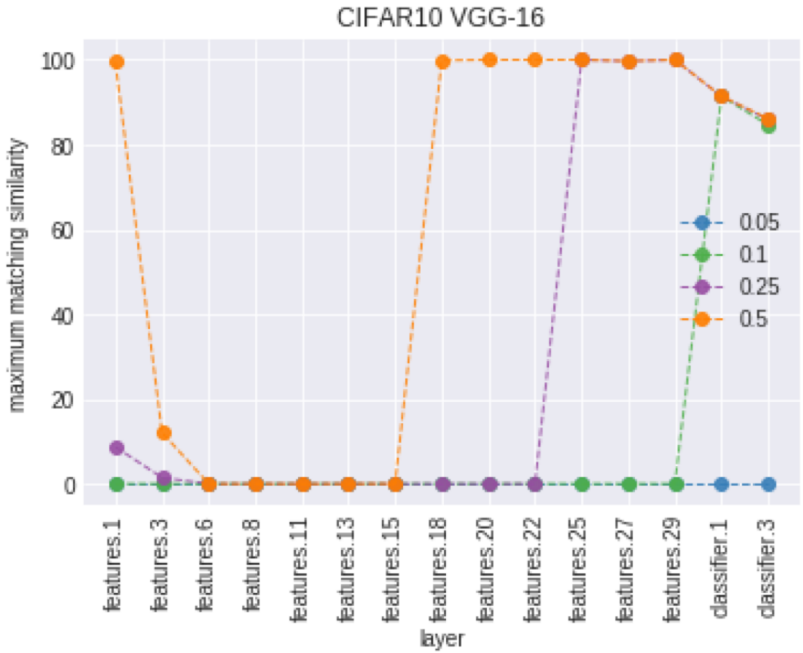


(a) CIFAR10-VGG11 (a) CIFAR10-ResNet18 (b) ImageNet-VGG16

Figure 3: Maximum matching similarities of all the layers of different architectures under various ϵ.

Experimental Results - Reproduced

1. Maximum Match

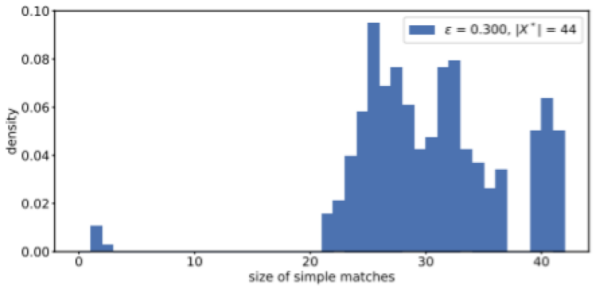


- We trained the VGG16 model with whole CIFAR10 dataset with two different initializations - Random seeds 0 and 1.
- We then took 500 samples for which we extracted layer wise information from trained data for both initializations.
- Then calculated maximum match for these 500 samples.

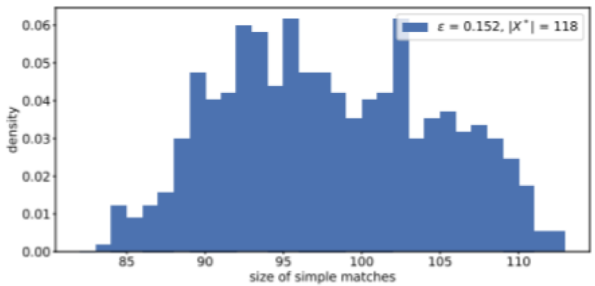
Experimental Results - Paper

2. Simple Match

- The final result is the collection of all the simple matches found, which is used to estimate the distribution.
- The layers close to input and output do not show similarity in local manner.



(a) Layer close to input



(b) Layer close to output

Figure 2: The distribution of the sizes of minimal matches of layers close to input and output respectively

Experimental Results - Reproduced

2. Simple Match

Average Similarity for last output layer = 3.19%

- The layers close to output do not show similarity in local manner.
- As we can see, the simple match at a neuron level show very low similarity over all neurons, which matches the result in the paper.

Experimental Analysis

1. Maximum Match

- The observation shows that different CNNs (with the same architecture) may learn different intermediate patterns.
- High similarity of layers close to the input is due to their alignment to the output.
    1. The output vector of two networks must be well aligned because they both achieve high accuracy.
    2. it is necessary that the layers before output are similar because if not, after a linear transformation, the output vectors will not be similar.
- High similarity of layers close to the input is due to their alignment to the same input data as well as the low-dimension nature of the low level layers. Moreover, it is much easier to have high similarity in low dimensional space than in high dimensional space.

Experimental Analysis

2. Simple Match

- While the layers close to output are similar overall, it seems that they do not show similarity in a local manner. There are very few simple matches with small sizes. It is also an evidence that such similarity is the result of its alignment to the output, rather than intrinsic similar representations.
- The layer close to input shows lower similarity in the finer structure. Again, there are few simple matches with small sizes.
- In sum, almost no single neuron (or a small set of neurons) learn similar representations, even in layers close to input or output.

Conclusion and Future Work

In this paper, we probe into the similarity between the representations that are learned by two neural networks which have the same architecture but they are trained with different initializations. Based on the results, we conclude that the representations learned by convolutional layers are not as similar as previously expected.

Two important questions are raised by this result:

- Do two networks learn completely different representations from different initializations?
- Is subspace match a good metric for measuring similarity of representations and if not, then what could be a good metric for the same?

Contribution

- Akhila - Initial feature extraction and matches on local machine, Added explanation on Jupyter notebook, Jupyter notebook coding, Slides, Plotted results
- Shivani - Initial training and model exploration on local machine, Trained the data on Jupyter notebook, Plotted results, Slides, Organized meetings
- Debarati - Added explanation on Jupyter notebook, Slides, Compiled initial findings of training and feature extraction for different initializations on google collab
- Mohit - Training and prediction for different initializations on Rivanna, Coded and obtained results for max match similarity on Google Collab, Updated slides.

References

1. Yann N Dauphin, Razvan Pascanu, Caglar Gulcehre, Kyunghyun Cho, Surya Ganguli, and Yoshua Bengio. Identifying and attacking the saddle point problem in high-dimensional non-convex optimization. In Advances in neural information processing systems, pages 2933–2941, 2014.
2. J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In CVPR09, 2009.
3. 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, pages 770–778, 2016.
4. Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. Cifar-10 (canadian institute for advanced research). URL http://www.cs.toronto.edu/~kriz/cifar.html.
5. Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems, pages 1097–1105, 2012.
6. Yixuan Li, Jason Yosinski, Jeff Clune, Hod Lipson, and John Hopcroft. Convergent learning: Do different neural networks learn the same representations? In International Conference on Learning Representation (ICLR ’16), 2016.