RE-IMPLEMENTATION: IDENTIFYING LEARNING RULES FROM NEURAL NETWORK OBSERVABLES

Jiawen Hu, Chuang Du, Zhaoxing Guo & Jiawei Qi
School of Electronics and Computer Science
University of Southampton, UK
{jh3n20, cd5g20, zg2u20 & jq1u20}@soton.ac.uk

ABSTRACT

The paper(Nayebi et al., 2020) introduced a virtual experiment to infer the learning algorithm which modifies the parameters of a neural network, from the trajectory it induces. A number of different deep artificial neural network models are trained on ImageNet tasks with four learning algorithms. While learning proceeds, the authors collect aggregate statistics. These aggregate quantities are then used as predictors to train simple classifiers, which attempt to discriminate which of the four learning rules trained the model. We divide the authors' work into four parts and verify the results separately: quantifying separation, gini feature importance analysis, trajectory subsampling, noise robustness. Our implementation could be found here :Identifying-Learning-Rules-From-Neural-Network-Observables. We have partially successfully reproduced the relevant experiments in the original paper and confirmed some of the conclusions of the original paper.

1 INTRODUCTION

The paper uses TensorFlow version 1.13.1 to conduct all model training experiments. The model function code and the names of the model layers from where they generated statistics are provided in Github. A brief tutorial is given in the Google Colab notebook on the use of many parts of this codebase, including analyzing the dataset, visualizing saved classifier results, and training classifiers on the dataset. Majority of the reproducibility experiments was implemented basing on the tutorial in the notebook, with some parts of the original code for reproducibility.

2 THE REPRODUCIBILITY

2.1 QUANTIFYING SEPARATION

2.1.1 IMPLEMENTATION

In this part, the task is to reproduce the confusion matrices. The matrices from the original paper shows that the learning rules, which are Adam, SGDM, Information Alignment (IA) and Feedback Alignment (FA), can be separated by the observable measures. The observable measures here refer to checking the features which are the weights, activations and layer-wise activity change. These features are analogized respectively to synaptic strengths in the brain, post-synaptic firing rates and paired measurements that involve observing the change in post-synaptic activity with respect to changes induced by pre-synaptic input.
Because the hardware condition cannot be met, we use the trained classifiers, the SVM, the Conv1D MLP and the Random Forest, to separate all four learning rules on ten category-balanced 75%/25% train/test splits of the data set - ImageNet, SimCLR, WSN and CIFAR-10 - using all of the observable statistics as the tutorial instructed. Choosing 'mean_accuracy' as the element of the confusion matrix. Reloading the learning rules which were pickled, we tested the accuracy on the test set. With the accuracy computed, we can plot the confusion matrix for each learning rules. Figure 1 shows the confusion matrices on the test set. From left to right, the graph presents the

matrix of the Conv1D MLP, Random Forest and SVM.

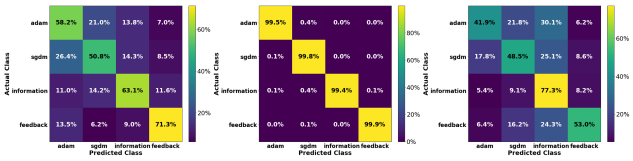

Figure 1: The confuse matrices

2.1.2 Result & Analysis

From Figure 1, we can see that the learning algorithm can be well recognized by classifier Random Forest than Conv1D MLP and SVM and it is clear that classifier Random Forest can hardly mistake one learning rules from the other. As for classifier Conv1D MLP and SVM, they are more likely to confuse Adam & SGDM than IA & FA. Choosing the proper features, which are weights, activations and layer-wise activity change, and training with four kinds of tasks, we verified the quantifying separation successfully.

2.2 Gini Feature Importance Analysis

2.2.1 implementation

After confirming the observable measurement that can be used to separate the learning rule, researchers focus on further analysis into the drivers of learning rule discriminability. Hence, the author considers a additional sets of controls the scale of observable statistic converted by the measurementwhich used for importance Analysis. By implement the statistic for RF classifier training ,the Gini Feature important extracted from the training process which represent the significance of observable statistic in learning rule separation.

In this parts, the task is to reproduce the Gini Feature Importance, and review the correctness of the result. The reproduction process can be divided into three steps: 1). Separately training the randomforest classifier for each types of statistic (weights, activation and layer-wise activity change), and obtain the Gini impurity feature importance from learning trajectory. 2). Sum the Gini impurity feature of each type of statistic extracted from 21 learning trajectories(one learning trajectories correspond to 5 epoch of the 100 epoch in model training). 3). Plot the processed Gini impurity feature importance.

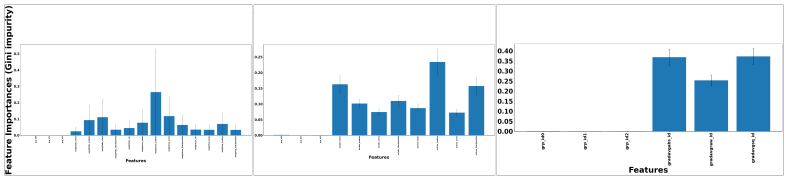

Figure 2: The reproduction exists slightly difference with the original

2.2.2 Result & Analysis

Figure 2 is drawn precisely according to the method stated by the authors. The reproduced graph indicated a similar result as the original image. For weight measurement, its median statistic contributed most important in separation; for averaged layer-wise activity changes, the absolute value the same as square. However, for activation, the reproduction shows the median statistic has more significance than the norm, which differs from the original result that the norm statistic more significance than the median.Since the plotting code provided by the authors does not sum the statistics for each category of the 21 learning trajectories, the lack of clarity in the summation method may be a factor that leads to the bias.

2.3 TRAJECTORY SUBSAMPLING

2.3.1 IMPLEMENTATION

An experimental often collects data throughout learning at regularly spaced intervals(Holtmaat et al., 2005). In this section, we attempt to reproduce the "trajectory subsampling" experiments mentioned in the original paper to further validate the conclusions regarding the performance of each measurement at different intervals in the training classifier when the trajectories are undersampled.
We designed the code with reference to the sampling scheme and sampling ratios and sampled the dataset provided in the original paper accordingly.Our implementation includes:

1. Sample sets are formed for three subsampling solutions according to every 5 epochs (original sample epoch interval), every 15 epochs, and every 25 epochs: "dense solution": a total of 21 trajectory samples, i.e. the full trajectory; "intermediate solution": a total of 7 trajectory samples; "sparse solution": a total of 5 trajectory samples.
2. A fixed number of subsampling operations are performed on the sample sets of these three solutions, and the random forest classifier is trained using the subsampled dataset to get the test accuracy variation for all observations of the corresponding solution and for each of the three measurable observations only.
3. A visualisation of the experiment results is made.

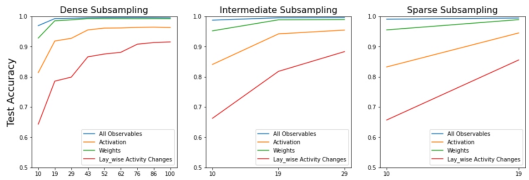

Figure 3: The Reproduced results of Trajectory Subsampling

2.3.2 RESULT & ANALYSIS

The results is shown in Figure 3, where the X-axis scale indicates the number of samplesas a percentage of the 21 samples in the full trajectory for each training data.
The Trajectory Subsampling Experiment is originally designed to explore the robustness of the prediction results in the case of undersampling of training data. It is also to see whether the three measurements of network weights, activation, and layer-wise activity change would show significantly different performance in experiments on classifying learning rules used to train neural networks due to the undersampling of the data, thus adding new evidence for measurements suitable for identifying learning rules to the data.
Comparing our results with those of the original paper does lead to the same conclusion about the higher robustness of measurements that are far apart in time, as well as showing that the network weights are overall the most robust to undersampling of the trajectory. Regarding the original conclusion that the activations can lead to achieve comparable performance with enough frequency of samples, we did not see the corresponding results in our experiments. We speculate that it is possible that we did not take into account the possibility of different subsampling starting positions when sampling, and that this randomness may be the factor that can affect the variation in performance of activations in the experiment.

2.4 NOISE ROBUSTNESS

2.4.1 IMPLEMENTATION

This part is to test the robustness of each observation (weights of the layer, activation from the layer, and layer-wise activity change of a given layer's outputs relative to its inputs) under the influence of unit undersampling and noise. Based on the simple tutorial code given by the author, we make a slight modification. First make a deep copy of the original data, and add a series of gaussian noises with a mean value of zero and variances of [0,0.5,1,5,10]. Finally train the random forest model and obtain the accuracy of the test set of each observation under different noise conditions. Under

the filtering of various conditions that ResNet-18 architecture trained on supervised ImageNet and self-supervised SimCLR with either the FA or IA learning rules, using a batch size of 256, model seed of None, and dataset randomization seed of None given in the paper, there are only 360 items we could use here, but we need to unit subsample at the rates of $3x10^{-3}\%$,$2.08x10^{-2}\%$ and so on. We think it is rather impossible to implement it. So here we only implement experiments under different noises.

2.4.2 Result & Analysis

And the result as the Figure 4 shows.It can be seen from the figure that, just like the conclusion mentioned in the original paper, the activation observable is the most robust across all noise (regardless of classifier). Under the influence of different noises, the accuracy of the test set remains high and stable. And the other two observables will be affected by noise, and the accuracy of the test set will drop significantly.

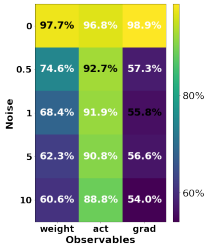

Figure 4: Accuracy of three observable with different noises

3 Conclusion

We performed some virtual experiments involved in the original paper based on the dataset composed of learning trajectories trained under different combinations of learning rules, models, and tasks. Generally speaking, most of the original experiments are reproduced, and obtained the same conclusion with the paper. Random forest can better distinguish various learning rules than Conv1D MLP and SVM only on the basis of aggregate statistics of the weights, activations , or layer-wise activity changes, and the classifier trained by subsampling of the learning trajectory with a further interval will be more robust and aggregate statistics across units of the network's activation patterns are most robust to unit undersampling and measurement noise.

References

Anthony J.G.D. Holtmaat, Joshua T. Trachtenberg, Linda Wilbrecht, Gordon M. Shepherd, Xiaoqun Zhang, Graham W. Knott, and Karel Svoboda. Transient and persistent dendritic spines in the neocortex in vivo. *Neuron*, 45(2):279–291, 2005. ISSN 0896-6273. doi: https://doi.org/10.1016/j.neuron.2005.01.003. URL `https://www.sciencedirect.com/science/article/pii/S0896627305000048`.

Aran Nayebi, Sanjana Srivastava, Surya Ganguli, and Daniel LK Yamins. Identifying learning rules from neural network observables. In *The 34th Conference on Neural Information Processing Systems (NeurIPS 2020)*, 2020. URL `https://arxiv.org/abs/2010.11765`.