Supporting Information for

A Law of Data Separation in Deep Learning

Hangfeng He and Weijie J. Su

Weijie J. Su.
E-mail: suw@wharton.upenn.edu

This PDF file includes:
- Supporting text
- Figs. S1 to S11
- SI References
Supporting Information Text

In this section, we first detail the general setup for all experiments, then briefly highlight some distinctive settings for figures in the main text, after that mention the argument for robustness, and finally show some additional results. More details can be found in our code.*

General Setup. In this subsection, we detail the general setup that is used for the experiments throughout the paper.

Network architecture. In our experiments, we mainly consider feedforward neural networks (FNNs). Unless otherwise stated, the corresponding hidden size is 100. For simplicity, we use neural networks with 4, 8, 20 layers as representatives for shallow, mid, and deep neural networks. By default, batch normalization (1) is inserted after fully connected layers and before the ReLU nonlinear activation functions.

Dataset. Our experiments are mainly conducted on Fashion-MNIST. Unless otherwise stated, we resize the original images to 10 × 10 pixels. By default, we randomly sample a total of 1000 training examples evenly distributed over the 10 classes.

Optimization methodology. In our experiments, three popular optimization methods are considered: SGD, SGD with momentum, and Adam. The weight decay is set to 5 × 10−4, and the momentum term is set to 0.9 for SGD with momentum. The neural networks are trained for 600 epochs with a batch size of 128. The initial learning rate is annealed by a factor of 10 at 1/3 and 2/3 of the 600 epochs. As for SGD and SGD with momentum, we pick the model resulting in the best equi-separation law in the last epoch among the seven learning rates: 0.001, 0.003, 0.01, 0.03, 0.1, 0.3, and 1.0. Similarity, we pick the model resulting in the best equi-separation law in the last epoch for Adam among the five learning rates: 3 × 10−5, 1 × 10−4, 3 × 10−4, 1 × 10−3, 3 × 10−3. Unless otherwise stated, Adam is adopted in the experiments.

Detailed Experimental Settings. In this subsection, we show detailed experimental settings for the figures in the main text.

Optimization. As shown in Fig. 1, for FNNs with different depths, three different optimization methods are considered: SGD, SGD with momentum, and Adam.

Visualization. As shown in Fig. 1, we visualize the features of different layers in the 8-layer FNNs trained on Fashion-MNIST. In particular, principal component analysis (PCA) is used to visualize the features in a two-dimensional plane.

Training epoch. As shown in Fig. 2, we show the separability of features among layers in different training epochs. In this part, we simply consider the 20-layer FNNs trained on Fashion-MNIST. The equi-separation law doesn’t emerge at the beginning, and gradually emerges during training. After that the decay ratio is decreased along the training process until the network converges. Furthermore, we conducted an analysis of the last-layer features in the aforementioned experiments. As illustrated in Fig. S1, it is evident that the separation fuzziness of the last-layer features does not reach convergence until after 400 epochs. Conversely, the equi-separation law manifests at epoch 100, as demonstrated in Fig. 2, with a Pearson correlation coefficient of −0.996. This observation indicates that the equi-separation law emerges prior to the occurrence of neural collapse.

As shown in Fig. 3, we further consider the impact of the following factors on the equi-separation law: dataset, learning rate, and class distribution.

Dataset. As shown in Fig. 3, we experiment with CIFAR-10. We use the second channel of the images and resize the original images to 10 × 10. Similar to Fashion-MNIST, we randomly sample a total of 1000 training examples evenly distributed over the 10 classes.

Learning rate. As shown in Fig. 3, we compare the 9-layer FNNs trained using SGD with different leaning rates on Fashion-MNIST.

Imbalanced data. As shown in Fig. 3, we experiment with imbalanced data. Specifically, we consider 5 majority classes with 500 examples in each class, and 5 minority classes with 100 examples in each class. The law of equi-separation also emerges in neural networks trained with imbalanced Fashion-MNIST data, though its half-life is significantly larger than that in the balanced case. This might be caused by the collapse of minority classifiers as shown in (2).

Convolutional neural networks (CNNs). As shown in Fig. 4, we experiment with two canonical CNNs, AlexNet and VGG (3). We use the PyTorch implementation of both models.1 We experiment with both Fashion-MNIST and CIFAR-10, and the original images are resized to 32 × 32 pixels. Given that AlexNet and VGG are designed for ImageNet images with 224 × 224 pixels, we make some small modifications to handle images with 32 × 32 pixels. As for AlexNet, we change the original convolution filters to 5 × 5 convolution filters with padding 2. The color channel number is multiplied by 32 at the beginning. As for VGG, we change the original convolution filters to 5 × 5 convolution filters with padding 2, and the last average pooling layer before fully connected layers is replaced by an average pooling layer over a 1 × 1 pixel window. The color channel number is multiplied by 16 at the beginning.

---

*Our code is publicly available at GitHub (https://github.com/HornHehhf/Equi-Separation).

1More details are in https://github.com/pytorch/vision/tree/main/torchvision/models.
Moreover, we examined VGG employing four distinct configurations\(^\dagger\) on the Fashion-MNIST and CIFAR-10 datasets. As demonstrated in Fig. S2, an approximate equi-separation law is also observed across different VGG configurations. It is important to note that the equi-separation law can be further refined by modifying convolution filter size and input channel number, as illustrated in Fig. 4.

To gain further insights into CNNs, we explored a variant of VGG11 (referred to as pure CNNs) by retaining all convolutional layers, the final fully connected layer, and removing all pooling layers along with the initial two fully connected layers. Additionally, we ensured an equal number of channels in each layer for our analysis. In this simplified network, convolutional layers serve to extract features, which are subsequently input into the final fully connected layer for classification. As presented in Fig. S3, the equi-separation law is present in pure CNNs for various datasets with an appropriate number of channels; however, the separation fuzziness of the last-layer features is considerably higher than that of FNNs and canonical CNNs (e.g., AlexNet and VGG). We further investigated the influence of other factors in CNNs, such as depth, kernel size, pooling layer, and image size. Our findings indicate that the equi-separation law is indeed present in CNNs but necessitates a delicate balance among these factors. This suggests that an alternative measure, rather than separation fuzziness (\(D\) in Eq. 1), may be required to describe the information flow in CNNs, given the unique structure of features following convolutional layers.

**Depth.** As shown in Fig. 5, besides Fashion-MNIST and CIFAR-10, we also consider MNIST to illustrate the optimal depth for different datasets. Similar to Fashion-MNIST, we resize the original MNIST images to \(10 \times 10\) pixels, and randomly sample a total of 1000 training examples evenly distributed over the 10 classes.

As shown in Fig. 6, we consider FNNs with different widths and shapes.

**Width.** As shown in Fig. 6, we consider FNNs with three different widths. All layers have the same width in each setting. For simplicity, we consider the 8-layer FNNs trained on Fashion-MNIST.

**Shape.** As shown in Fig. 6, we consider three different shapes of FNNs: Narrow-Wide, Wide-Narrow, and Mixed. As for Narrow-Wide, the FNNs have narrow hidden layers at the beginning, and then have wide hidden layers in the later layers. Similarity, we use Wide-Narrow to denote FNNs that start with wide hidden layers followed by narrow hidden layers. In the case of Mixed, the FNNs have more complicated patterns of the widths of hidden layers. For simplicity, we only consider 8-layer FNNs trained on Fashion-MNIST. The corresponding widths of hidden layers for Narrow-Wide, Wide-Narrow, and Mixed considered in our experiments are as follows: \((100, 100, 100, 100, 1000, 1000, 1000), (1000, 1000, 1000, 1000, 100, 100, 100),\) and \((100, 500, 500, 2500, 2500, 500, 500)\).

**Residual neural networks (ResNets).** As shown in Fig. 7, we consider three different types of ResNets: ResNets with 2-layer building blocks, ResNets with 3-layer bottleneck blocks, and ResNets with mixed blocks. For 3-layer blocks, the expansion rate is set to 1 instead of 4, and the original \(1 \times 1\) and \(3 \times 3\) convolution filters are replaced by \(5 \times 5\) convolution filters. For all types of ResNets, the channel number is set to 8, and the color channel number is also multiplied by 8 at the beginning. In this part, the Fashion-MNIST and CIFAR-10 images are resized to \(32 \times 32\) pixels instead of \(10 \times 10\) pixels.

**Dense Convolutional Network (DenseNet).** As shown in Fig. 8, we investigate the law when applied to DenseNet (4), namely DenseNet161.\(^\ddagger\) In this part, the images of Fashion-MNIST and CIFAR-10 are resized to \(32 \times 32\) pixels, i.e., \((3, 32, 32)\) for CIFAR-10 and \((32, 32)\) for Fashion-MNIST.

**Out-of-sample performance.** As shown in Fig. 9, given 20-layer FNNs, instead of the standard training, we also consider the following two-stage frozen training procedure: 1) freeze the last 10 layers and train the first 10 layers; 2) freeze the first 10 layers and train the last 10 layers. When we conduct the frozen and standard training on CIFAR-10, we find that the out-of-sample performance of standard training (test accuracy: 23.85\%) is better than that of frozen training (test accuracy: 19.67\%), even though the two training procedures have similar training losses (standard: 0.0021; frozen: 0.0019) and training accuracies (standard: 100\%; frozen: 100\%). This is consistent with the equi-separation law: the neural networks trained using standard training have much clearer equi-separation law compared to the neural networks trained using frozen training. Specifically, the corresponding Pearson correlation coefficients for standard and frozen training are \(-0.997\) and \(-0.971\), respectively.

**Additional Results.** In this subsection, we provide some additional results for the equi-separation law.

**Image size.** As depicted in Fig. S4, we examine FNNs on images with a resolution of \(32 \times 32\) pixels, specifically \((3, 32, 32)\) for CIFAR-10 and \((32, 32)\) for Fashion-MNIST. It is important to note that Fashion-MNIST, despite its original size of (28, 28), is frequently resized to \((32, 32)\) in the same manner as MNIST. In this part, we set the width as 1000 instead of 100 for FNNs, given that the images with \(32 \times 32\) pixel resolution possess a higher dimension in comparison to those with \(10 \times 10\) pixel resolution.

**Sample size.** As shown in Fig. S5, we experiment with different number of randomly sampled training examples. For each setting, we have the same number of examples for each of the 10 classes. For simplicity, we only consider the 8-layer FNNs trained on Fashion-MNIST. The law of equi-separation emerges in neural networks trained with different number of training examples, though the half-life can be larger for larger datasets due to the increased optimization difficulty.

\(^\dagger\) More details can be found at: \https://github.com/kuangliu/pytorch-cifar/tree/master/models.  
\(^\ddagger\) Further details can be found at \https://github.com/kuangliu/pytorch-cifar/tree/master/models.
**Class-wise data separation.** As shown in Fig. S6, we further consider separation fuzziness for each class, i.e., $SS_k^w SS_k^+$, where $SS_k^w$ indicates the within-class covariance for Class $k$. For simplicity, we only consider the 8-layer FNNs trained on Fashion-MNIST. The equi-separation law emerges in each class. Even though the equi-separation law can be a little noisy for some classes, the noise is reduced when we consider the separation fuzziness for all classes.

**Training dynamics.** As shown in Fig. S7, we consider the convergence rates of different layers (without the last-layer classifier) with respect to the relative improvement $\frac{D_{l+1}}{D_l}$ of separability of features. For simplicity, we consider the 8-layer FNNs trained on CIFAR-10 in this part. The relative improvement of bottom layers converges earlier compared to those of top layers.

**Test data.** As shown in Fig. S8, we further show how features separate across layers in the test data. For simplicity, we only consider the 8-layer FNNs trained on Fashion-MNIST here. A fuzzy version of the equi-separation law also exists in the test data.

**BERT.** As shown in Fig. S9, we experiment with BERT features. We use the pretrained case-sensitive BERT-base PyTorch implementation (5), and the common hyperparameters for BERT. Specifically, we fine-tuned the pretrained BERT model on a binary sentiment classification task (SST-2) (6), where we randomly sampled 500 sentences for each class. Given a sequence of token-level BERT features, two most popular approaches are used to get the sentence-level features at each layer: 1) using the features of the first token (i.e., the [CLS] token); 2) averaging the features among all tokens in the sentence.

**Batch normalization.** As shown in Fig. S10, it is difficult to optimize a neural network without batch normalization, let alone achieve the law of equi-separation. Even when the network is well-optimized, the law is often not clear.

**Pretraining.** As shown in Fig. S11, we consider the impact of the pretraining on the equi-separation law. Specifically, we first pretrained the 20-layer FNNs on Fashion-MNIST as shown in Fig. S11 (Depth=20). After that we fix the first 10 layers of the pretrained model and train additional 5 layers as in Fig. S11 (Pretraining), which is quite different from training 15-layer FNNs from scratch as in Fig. S11 (Depth=15). At the same time, the equi-separation law also emerges in the additional 5 layers in the pretraining setting.

---

Fig. S1. Separation fuzziness of last-layer features throughout the training process. The $x$ axis represents the epoch number, while the $y$ axis indicates the separation fuzziness ($D$ in Eq. 1). The $x$ axis commences from approximately 100 epochs due to the exceedingly high initial separation fuzziness, which would otherwise mask discernible trends.

---

The input layer is not considered here since the inputs of the first token do not take other tokens into account.
Fig. S2. Emergence of a fuzzy equi-separation law in VGG networks with varied depths on Fashion-MNIST and CIFAR-10.
**Fig. S3.** Equi-separation law emerges in pure CNNs with appropriate numbers of channels on Fashion-MNIST (64 channels) and CIFAR-10 (16 channels).
Fig. S4. Equi-separation law emerges in FNNs across varied depths on 32 × 32 pixel images of Fashion-MNIST and CIFAR-10.
Fig. S5. The impact of the sample size on the law of equi-separation.
Fig. S6. The law of equi-separation emerges for each class.
Fig. S7. Convergence rates of different layers (without the last-layer classifier) with respect to the relative improvement $\frac{D_{l+1}}{D_l}$ of separability of features. The x-axis represents the training epoch, while the y-axis indicates the relative improvement $\frac{D_{l+1}}{D_l}$. 
Fig. S8. A fuzzy version of the equi-separation law also emerges in the test data.
Fig. S9. The equi-separation law does not exist in BERT features. Given a sequence of token-level BERT features, two most popular approaches are used to get the sentence-level features at each layer: 1) using the features of the first token (i.e., the [CLS] token); 2) averaging the features among all tokens in the sentence (denoted as BERT-AVG).
Fig. S10. The equi-separation law is not clear on Fashion-MNIST for FNNs trained without batch normalization.
Fig. S11. The impact of pretraining on the equi-separation law. We first pretrained the 20-layer FNNs as shown in Fig. S11 (Depth=20). After that we fix the first 10 layers of the pretrained model and train additional 5 layers as in Fig. S11 (Pretraining), which is quite different from training the 15-layer neural networks from scratch as in Fig. S11 (Depth=15).
References

1. S Ioffe, C Szegedy, Batch normalization: Accelerating deep network training by reducing internal covariate shift in *Proceedings of the 32nd International Conference on Machine Learning*. (PMLR), Vol. 37, pp. 448–456 (2015).

2. C Fang, H He, Q Long, WJ Su, Exploring deep neural networks via layer-peeled model: Minority collapse in imbalanced training. *Proc. Natl. Acad. Sci.* **118**, e2103091118 (2021).

3. K Simonyan, A Zisserman, Very deep convolutional networks for large-scale image recognition in *International Conference on Learning Representations*. (2015).

4. G Huang, Z Liu, L Van Der Maaten, KQ Weinberger, Densely connected convolutional networks in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. pp. 4700–4708 (2017).

5. T Wolf, et al., Transformers: State-of-the-art natural language processing in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. pp. 38–45 (2020).

6. R Socher, et al., 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).