Understanding Deep Learning with Activation Pattern Diagrams

Francesco Craighero[1], Fabrizio Angaroni[1], Alex Graudenzi[2,†,*], Fabio Stella[1,†], and Marco Antoniotti[1,†]

[1] Department of Informatics, Systems and Communication,
University of Milan-Bicocca, Milan, Italy
[2] Institute of Molecular Bioimaging and Physiology,
Consiglio Nazionale delle Ricerche (IBFM-CNR), Segrate, Italy
[*] corresponding author: alex.graudenzi@unimib.it
† co-senior authors

Abstract. The growing demand for machine learning tools to solve hard tasks, from natural language processing to image understanding, recently shifted the attention to understand and possibly to explain the behaviour of deep learning. Deep neural networks represent today the state-of-the-art in many applications that have been shown to be solved by data-driven approaches. However, they are also well known for their complexity, which hinders the interpretation of their functioning. To address this issue, researchers have lately focused either on understanding the optimization algorithms or on extracting information from a trained model; in this context we propose the *Activation Pattern Diagram* (APD) as a new tool to analyse neural networks by mainly focusing on the input data. The APD is a graphical representation of how a dataset is learned by a neural network with piecewise linear activation functions, such as the ReLU activation. By analysing the evolution of the diagram during the training procedure, the APD sheds light on the learning process and how data influences it. Additionally, we introduce a way to plot the APD to help the visualization and interpretation of the diagram.

Keywords: Activation Patterns · Piecewise Linear Functions · Neural Networks · Visualization.

1 Introduction

Deep neural networks (DNNs) have achieved remarkably good results in a broad range of tasks, including Computer Vision [7,8,15], Natural Language Processing [2] and game playing [14]. Nevertheless, due to the complexity of these models, many phenomena are still only partially understood, such as their ability of to generalize well with over-parameterized models [17] or their fragility to adversarial attacks [16]. Moreover, the ever growing adoption of black-box models fueled the need of explainable systems, in order to gain the trust of the user and improve the confidence for safety-critical applications [3].

In order to explain neural networks, a number of techniques provide justifications for the predictions [10], such as sensitivity analysis [15]. On the other hand, other methods have been proposed to investigate the properties of DNNs, e.g., to estimate the expressiveness of the model [5,6,13], to analyse the behaviour of DNNs with different optimization techniques [18] or to characterize input data complexity [1].

The inspection of a deep neural network can be simplified by employing piecewise linear activation functions, such as the ReLU activation [4]. These activations partition the input space in linear regions to learn complex functions [11], thus properties of those regions, such as the number or the size, can be exploited to better understand the learned function [1,5,6,13,18].

In [1] we defined a novel data structure, the *Activation Pattern Diagram* (APD), that can be used to understand and visualize how data is transformed by a neural network with piecewise linear activations. Additionally, we introduced a method to estimate the input data complexity of a dataset, given the function learned by a DNN with ReLU activations. More in detail, we showed that the distribution of the input instances among the linear regions, summarised by the APD, can be used to estimate the confidence of the model in predicting the label for a given instance. Briefly, linear regions identify the transformation applied by the neural network; if many instances share the same linear region, then we expect them to be "more common", and easier, than an instance that has its own linear region. In fact, linear regions are denser around decision boundaries [18].

In order to further explore the APD properties, we aim at investigating its evolution during the training process. To this end, in the following we will:

- introduce a proof-of-concept for a novel tool to visualize the APD on a selected subset of instances, providing a new strategy to interpret DNNs;
- show preliminary results of the evolution of the APD during learning.

2 From Activation Patterns to the APD

Let $\mathcal{N}_\theta(x_0)$ be a *Deep Neural Network* with input $x_0 \in \mathbb{R}^{n_0}$ and trainable parameters θ. A layer h_l with size n_l, for $l \in 1,\dots,L$, is defined by neurons $h_{l,i} = g_{l,i} \circ f_{l,i}$, for $i \in 1,\dots,n_l$, where $f_{l,i}$ is a linear preactivation function and $g_{l,i}$ a nonlinear activation function.

Let x_l be the output of the l-th layer and the input data to the network for $l = 0$, then, we define $f_{l,i}(x_{l-1}) = W_l x_{l-1} + b_{l,i}$, where both $W_l \in \mathbb{R}^{n_{l-1}}$ and $b_{l,i} \in \mathbb{R}$ belong to the trainable parameters θ. Regarding activation functions, we will focus on ReLU activation function, i.e., $g_{l,i}(x) = \max\{0, x\}$.

Finally, we can represent the DNN $\mathcal{N}_\theta$ as a function $\mathcal{N}_\theta : \mathbb{R}^{n_0} \to \mathbb{R}^{out}$ that can be decomposed as

$$\mathcal{N}_\theta(x) = (f_{out} \circ h_L \circ \dots \circ h_1)(x), \tag{1}$$

where f_{out} is the output layer (e.g., softmax, sigmoid, ...).

Moreover, given a dataset $\mathcal{D} \subseteq \mathbb{R}^{n_0}$, we define the *activation pattern* $A_l(x_0)$ of layer l given input $x_0 \in \mathcal{D}$ as the following (binary) vector:

$$A_l(x_0) = [a_i \mid a_i = 1 \text{ if } h_{l,i}(x_{l-1}) > 0 \text{ else } a_i = 0,\ \forall i = 1, \dots, n_l]. \tag{2}$$

In order to distinguish activation patterns by the layer to which they belong, let us adjust the notation as follows:

$$A_l^*(x_0) = (l, A_l(x_0)), \forall x_0 \in \mathcal{D},\ l \in 1, \dots, L. \tag{3}$$

Then, let us define the set of activation patterns of layer l for all instances in $\mathcal{D}$ as:

$$A_l^*(\mathcal{D}) = \{A_l^*(x_0) \mid x_0 \in \mathcal{D}\},\ l \in 1, \dots, L,$$

where $|A_l^*(\mathcal{D})|$ will denote its cardinality.

Lastly, the *activation pattern diagram* (APD) of dataset $\mathcal{D}$ is a directed acyclic graph $APD_{\mathcal{N}_\theta}(\mathcal{D}) = (V, E)$, where

- V is the set of vertices defined by the activation patterns of all the layers, i.e.:
$$V = \bigcup_{l=1}^{L} A_l^*(\mathcal{D}).$$
- E is the set of edges defined by the activation of each input instance $x_0 \in \mathcal{D}$, i.e. $(A_{l-1}^*(x_0), A_l^*(x_0)) \in E$ for $l \in 2, \dots L$.

Note that the APD has the same depth of the network. In the following we will consider an extended version of the APD, in which we add a node for each predicted label and edges $(A_L^*(x_0), \mathcal{N}_\theta(x_0))$, where $\mathcal{N}_\theta(x_0)$ is the predicted label for x_0, for each input instance.

Example 1. Let us consider a network $\mathcal{N}_\theta$ with $L = 2$ and $n_1, n_2 = 2$. Given a dataset with one instance x_0, we may have $A_1^*(x_0) = (1, [0, 0])$, $A_2^*(x_0) = (2, [1, 0])$ and $N_\theta(x_0) = y_0$, i.e. y_0 is the predicted label for x_0. Then the APD is defined as:

$$V = \{(1, [0, 0]), (2, [1, 0]), y_0\}, E = \big\{\big((1, [0, 0]), (2, [1, 0])\big), \big((2, [1, 0]), y_0\big)\big\}.$$

3 APD evolution during training

In this section we will show results obtained with a neural network with $L = 3$ layers with 40 neurons each, trained on the MNIST dataset [9] with SGD and fixed learning rate at 0.001.

In figure 1 we have the loss for the training process (0.1 train/validation split of the 60 000 total instances) on the left, while on the right we have the evolution of the number of unique activation patterns, i.e. $|A_l^*(\mathcal{D})|$ for $l \in 1, \dots 3$, where $\mathcal{D}$ is the training set. We can see that the number of unique patterns at each epoch

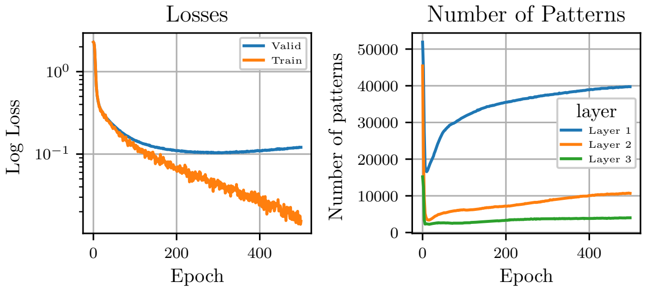


Fig. 1: (Left) Train (orange) and validation (blue) loss. The best validation is achieved at epoch 309. (Right) Number of unique activation patterns per layer, i.e. $|A^*_l(\mathcal{D})|$, with regard to the network at a given epoch. All the layers steadily increase the number of activation patterns during training. Furthermore, the first layer (blue) has 4 times the number of activation patterns of the second layer, while the third layer (green) doesn't reach more that 4 000 activation patterns in all the 500 epochs.

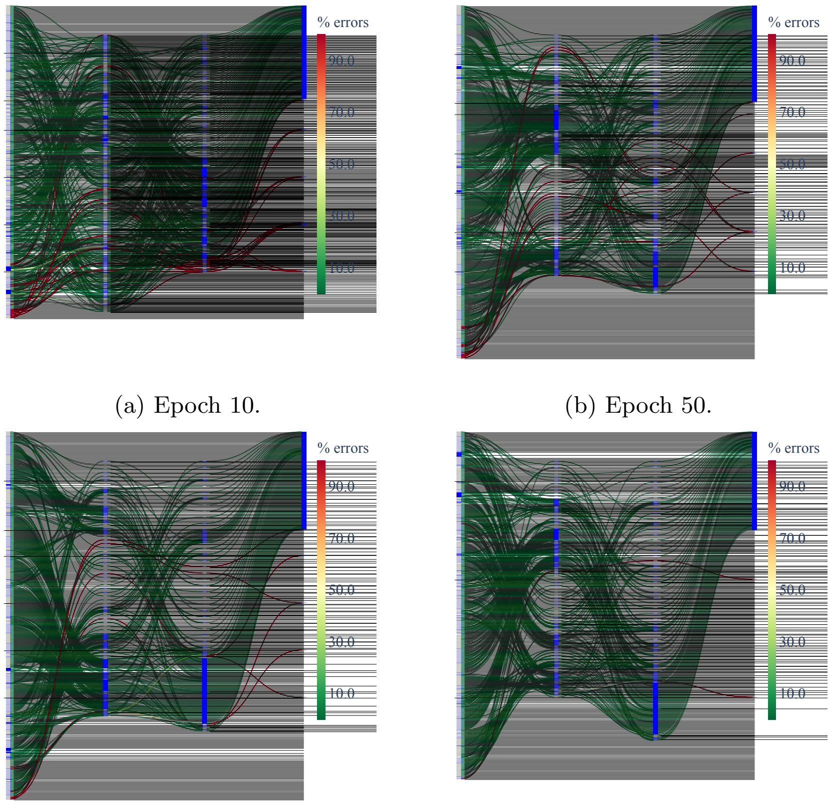
decreases with the layer's depth; moreover, the number of activation patterns of each layer is always far below the theoretical upper bound of 2^{40} possible patterns (see [6] for an explanation of this phenomenon) and less than the 54 000 training instances, thus activation patterns are shared between instances.

In figure 2 we plotted the APD obtained by performing predictions for the same 500 instances of label "1" with the learned model at epochs 10, 50, 150 and 300. The plots are Sankey diagrams from the `Plotly` library [12], where:

– the blue rectangles, from left to right, represent the activation patterns of the layers and the predicted labels, with height and color intensity proportional to the number of instances activating that pattern or predicting that label. As an example, in each APD, the upper right rectangle represent all correctly predicted labels;
– the color of the edges corresponds to the proportion of wrong instances belonging to the edge, and size proportional to the number of instances following that edge.

From figure 2 we can observe that activation patterns are shared more in deeper layers, as emerges from figure 1. As a consequence, from epoch 150 there are some clear flows of instances that share the same activation patterns. Lastly, wrongly classified instances, with regard to the chosen subset of instances, mostly belong to activation patterns that are not shared by many instances.

The trend observed in the previous figures is confirmed by figure 3. We first clustered instances based on the pattern of both the second and last layer, i.e. if x_0, x_1 belong to cluster (or "flow") C, then $A_2(x_0) = A_2(x_1)$ and $A_3(x_0) = A_3(x_1)$. Note that such clusters correspond to paths from the second to third layer in figure 2. Then, we observed the distribution of all (second row) or



(a) Epoch 10.

(b) Epoch 50.

(c) Epoch 150.

(d) Epoch 300.

Fig. 2: Four different APDs with regard to the same 500 instances of label "1" and the model learned at epochs 10, 50, 150 and 300. The APD is composed by four levels of blue nodes; the nodes of each level identify, from left to right, the activation patterns of the three layers of the network and the predicted labels. Additionally, the nodes have height and color intensity proportional to the number of instances they represent. The first layer, as expected from figure 1, has always more patterns, while in the level of predicted labels we have a tall node on top representing the mostly predicted label, i.e. "1". The edges have size proportional to the number of instances they represent, and different colors depending on the proportion of errors.

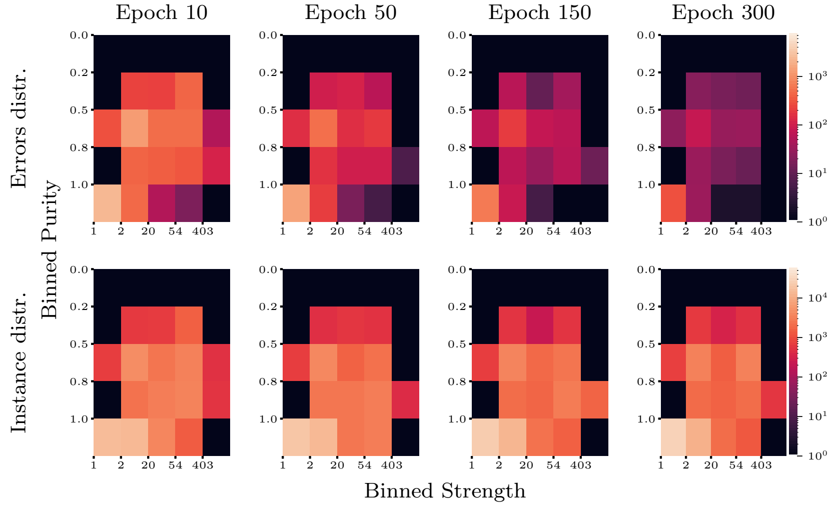


Fig. 3: Distribution of wrongly classified instances (first row) and all instances (second row) among clusters ("flows") defined by activation patterns of second and third layer. Clusters correspond to the edges in the APD between the activation patterns of the second and third layer, respectively. In the x-axis we have the strength of the clusters, with log-bins, i.e. the number of instances belonging to that cluster. In the y-axis we have the purity of the cluster, i.e. the proportion of instances belonging to the most frequently predicted label in the cluster.

wrongly classified (first row) instances among the clusters with regard to two measures: purity, i.e. the proportion of instances of the most frequently predicted class in the cluster, and strength, i.e. the cluster size. We can observe that there is a number of instances belonging to clusters with high purity and high strength, that are almost always correct from epoch 150, while wrongly classified instances usually belong to small clusters or clusters with low purity.

4 Concluding remarks

In the previous section we showed some of the possible observations resulting from the analysis of how data flows through the APD during the training process. Additionally, we introduced a novel visualization tool to plot the APD for a given set of input instances.

We are able to cluster data based on how the neural network performs the task, paving the way to further experiments with the aim of both studying how the characteristics of input data influences the learning process and providing an interpretation for the function learned by a neural network. As an example, the APD can provide a way to quickly assess when a trained DNN is straying from what it was trained on, potentially providing early warnings "on field", when it behaves in ways that were not expected or foreseen.

Among the possible future research venues, we want to investigate topological measures to quantify the information contained in the APD and experiment the influence of hyperparameters, such as the chosen architecture or optimization algorithm, on the shape of the diagram. Lastly, in our experiments we used all the neurons in each layer of the APD, but additional research may introduce new ways to identify only the relevant part of activation patterns.

Furthermore, we here introduced a visualization of the APD with the `Plotly` library [12], that might represent a new tool for the researcher or user who wants to understand the inner functioning of a DNN.

References

1. Craighero, F., Angaroni, F., Graudenzi, A., Stella, F., Antoniotti, M.: Investigating the Compositional Structure Of Deep Neural Networks. In: Proceedings of the Sixth International Conference on Machine Learning, Optimization, and Data Science. LOD 2020. Siena, Italy (2020), (preprint: https://arxiv.org/abs/2002.06967)
2. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: Pre-training of deep bidirectional transformers for language understanding. In: Burstein, J., Doran, C., Solorio, T. (eds.) Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers). pp. 4171–4186. Association for Computational Linguistics (2019). https://doi.org/10.18653/v1/n19-1423
3. Gilpin, L.H., Bau, D., Yuan, B.Z., Bajwa, A., Specter, M., Kagal, L.: Explaining Explanations: An Overview of Interpretability of Machine Learning. In: 2018 IEEE 5th International Conference on Data Science and Advanced Analytics (DSAA). pp. 80–89 (Oct 2018). https://doi.org/10.1109/DSAA.2018.00018
4. Glorot, X., Bordes, A., Bengio, Y.: Deep Sparse Rectifier Neural Networks. In: AISTATS (2011)
5. Hanin, B., Rolnick, D.: Complexity of Linear Regions in Deep Networks. In: International Conference on Machine Learning. pp. 2596–2604 (2019)
6. Hanin, B., Rolnick, D.: Deep ReLU networks have surprisingly few activation patterns. In: Wallach, H.M., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E.B., Garnett, R. (eds.) Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada. pp. 359–368 (2019)
7. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (Jun 2016)
8. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet classification with deep convolutional neural networks. Commun. ACM **60**(6), 84–90 (May 2017). https://doi.org/10.1145/3065386
9. LeCun, Y., Cortes, C., Burges, C.: MNIST handwritten digit database. ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist **2** (2010)
10. Montavon, G., Samek, W., Müller, K.R.: Methods for interpreting and understanding deep neural networks. Digital Signal Processing **73**, 1–15 (Feb 2018). https://doi.org/10.1016/j.dsp.2017.10.011

11. Montúfar, G.F., Pascanu, R., Cho, K., Bengio, Y.: On the number of linear regions of deep neural networks. In: Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N.D., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada. pp. 2924–2932 (2014)
12. Plotly Technologies Inc.: Collaborative data science. https://plot.ly (2015)
13. Raghu, M., Poole, B., Kleinberg, J.M., Ganguli, S., Sohl-Dickstein, J.: On the Expressive Power of Deep Neural Networks. In: Precup, D., Teh, Y.W. (eds.) Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017. Proceedings of Machine Learning Research, vol. 70, pp. 2847–2854. PMLR (2017)
14. Silver, D., Huang, A., Maddison, C.J., Guez, A., Sifre, L., van den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., Dieleman, S., Grewe, D., Nham, J., Kalchbrenner, N., Sutskever, I., Lillicrap, T.P., Leach, M., Kavukcuoglu, K., Graepel, T., Hassabis, D.: Mastering the game of Go with deep neural networks and tree search. Nat. **529**(7587), 484–489 (2016). https://doi.org/10.1038/nature16961
15. Simonyan, K., Zisserman, A.: Very Deep Convolutional Networks for Large-Scale Image Recognition. In: Bengio, Y., LeCun, Y. (eds.) 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings (2015)
16. Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I.J., Fergus, R.: Intriguing properties of neural networks. In: Bengio, Y., LeCun, Y. (eds.) 2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings (2014)
17. Zhang, C., Bengio, S., Hardt, M., Recht, B., Vinyals, O.: Understanding deep learning requires rethinking generalization. In: 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings. OpenReview.net (2017)
18. Zhang, X., Wu, D.: Empirical Studies on the Properties of Linear Regions in Deep Neural Networks. In: 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020. OpenReview.net (2020)