Linking Labels to Neural Subnetworks

Marcus Gullstrand[1,2]
Supervisors: Maria Riveiro[1,3] and Florian Westphal[1,4]

I. INTRODUCTION

Neural networks, in all their different forms, have shown great performance in a variety of tasks in recent years. Within the last few years, models have become larger and more sophisticated and their capabilities have exceeded our expectations [1], [2], [3]. Large neural network models, or neural network models in general, come at a price for their capabilities: They are hard to understand and interpret. Normally, neural networks are opaque models and suffer from the black-box problem [4]. In high-stakes domains such as medicine, criminal justice, banking, or critical industrial applications, there is an inherent need to being able to explain the reasoning or the process that led to the final recommendations. Thus, if we need to assess the limitations and capabilities of neural network models, we need to find a way to overcome their opaque nature.

To assess what a neural network has learnt and the limitations of the network, one strategy is to be able to dissect the network and look at the structures or data-paths that might emerge inside of it. For each label that a neural network learns, an inner structure emerges such that it can solve the machine learning task. Common metrics such as accuracy, mean absolute error or intersection over union do not reflect these learnt inner structures. Approaches have been developed to fully understand what is happening inside a neural network with varying degrees of success [4]. These works have, e.g., looked at only one instance of an architecture [5], layer wise comparison for different learnt representations [6], or finding human-friendly concepts within already trained models [7]. However, it would be good to have other methods to view, detect and evaluate inner structures that we ourselves can view and interpret well.

In this project, we will approach the black-box problem by developing a way to uncover the inner workings of neural networks. We explore the idea of subnetwork analysis on multiple trained instances of the same network architecture but initialized differently. We define a subnetwork as a partial neural network inside a neural network for a specific label. This subnetwork encapsulates all related concepts to the label, e.g., a label for cats in the model should have a subnetwork that includes the concept of a tail, fur or paws, etc. We attempt to find and extract subnetworks from these neural network instances and compare the convergence for a given label. Studying multiple instances of a neural network architecture allows us to gain deeper insights into how subnetworks capture concepts that relate to labels, and how models learn when instantiated differently. This broader understanding helps us to generalise our findings beyond specific instances of neural networks. Thus, the overall research question is

- How can we map parts of the network to a particular label?

[1]Department of Computing, Jönköping University, Gjuterigatan 5, Jönköping, Sweden.
[2]marcus.gullstrand@ju.se
[3]maria.riveiro@ju.se
[4]florian.westphal@ju.se

II. RELATED WORK

In this section, we briefly review several works related to creating and finding subnetworks. We also summarize works that aim to perform analysis multiple instances of the same architecture.

The "Lottery ticket hypothesis" was recently coined as an idea as to why we want to train large neural network models [8]. When training large neural network models, some weight configurations, called *winning tickets*, will learn faster. These *winning tickets* (or subnetworks) are kept while other small-magnitude weights are pruned. After pruning, the remaining weights are reset to their previously untrained values, and the network is retrained once again. This process continues until performance drops significantly or some other metrics are met. In the end, you will have a very pruned network consisting mostly of *winning tickets*. The focus here is on pruning neural networks by large amounts, but no more investigations are made as to why these *winning tickets* are learning better or if they represent learnt concepts better.

A common approach for analysing neural network is temporary weight pruning, i.e., looking at what effect a set of weights have when they are removed from the network. Wang et al. [9] analyse the working process of the model to understand how the model achieves its decision. They propose a framework for interpreting neural networks by analysing the CDRPs (Critical Data Routing Paths) identified by the proposed distillation guided routing method. Their framework analyse already trained networks during inference on testing and validation sets to detect weights of higher significance. These weights are then set to zero, and performance should drop significantly if a CDPR is found. This work focuses on preventing adversarial attacks on neural networks, but they state that future work should explore the underlying principle of the emergence of these CDRPs, such as subnetworks. Csordás et al. [10] also explore the training of binary masks but on the neural network to get the neurons that make up a specific subtask in

the dataset. They define and look at two different neuron-types, "P_specialize" and "P_reuse," that either specialize themselves or are reused over multiple inner functions. This approach performs well, and has the upside of also showing to which degree neurons are shared between different groups of neurons, which they also call *circuits*. They noticed a problem that similar functions in the network are learned multiple times and could potentially be merged.

Another common way to analyse neural networks is by following the path the data takes. Fiacco et al. [11] identifies pathways within neural networks by looking at what happens when they feed data into the neural network. They construct an activation matrix where each column represents an individual neuron, each row represents one data instance, and the values in the cells are the activation for that specific neuron and data point. They use "Linear probes" which are a series of trained logistic regression models that are trained to map a neural representation to a given linguistic phenomenon. They use logistic regression because previous neural network methods have had the problem of trying to interpret a complex model with another complex model. The strength of this method is that we are able to look at only the interesting parts, and that we, to some extent, can draw a line between neurons in the network and a learnt concept.

There have been few works analysing multiple neural networks with similar network architectures. Kornblith et al. [6] test a new way of looking for similarities or substructures in both single and between multiple deep neural networks. They introduce a similarity index called Centered Kernel Alignment (CKA) which compares activation similarities between layers when data is fed into one or multiple trained models. The drawback of this method is that it only compares the same layer in multiple model instances, and it does not compare larger subnetworks.

The detection of larger subnetworks is in itself a hard problem that often comes with computational difficulties. While methods exists to detect different aspects of subnetworks, there is still room to systematically test combinations of methods to achieve better results [4].

III. METHOD

In this project, we aim at finding subnetworks because they encapsulate multiple concepts that make up the label; having all concepts related to a label isolated from the remaining parts of the neural network is a stepping stone to making neural networks less opaque. To investigate if subnetworks exist within a model, we primarily use quantitative research in the form of experiments.

We first train multiple neural networks on known problems with clear knowledge of the desired outcome. Each model should solve the machine learning task satisfactorily before we start to analyse and look for subnetworks. This could mean that, e.g., for a classification problem, each neural network should have a high accuracy before we analyse it.

We aim to detect subnetworks within all models, tying learnt labels to detected subnetworks with currently available methods such as those mentioned by Kornblith et al. [6]. We take note of observable similarities between the networks; this is to establish the possibility of overlapping convergence of concepts between network instances. Exploration of new methods will also be tested.

To evaluate the scope of a subnetwork inside a neural network, we will isolate the subnetwork for a single label to know how much performance we can expect from that particular subnetwork and label and to what degree parts of the network that are not identified as part of the subnetwork affect performance. This by itself is a key insight, especially when looking at multiple trained model instances of the same architecture. Do, for example, some models crystallize their learnt concepts into more distinguishable subnetworks, or do most models always require the entire network?

Finally, we will evaluate the possibility of combining the subnetworks by extracting a subnetwork from each model instance for a single label. In theory, we should be able to have the relations between neurons such that we find an absolute representation of the learnt labels for the models' architecture.

In this project, we carry out research in collaboration with two companies with different challenges related to explainable AI, classification, and object detection. The companies supply real-world scenarios, data, and expert knowledge. Thus, we look at how well a subnetwork can be mapped to a particular label in a real-world scenario with real data.

REFERENCES

[1] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, "Language models are few-shot learners," vol. 33, pp. 1877–1901, Curran Associates, Inc., 2020.

[2] C. Wang, S. Chen, Y. Wu, Z. Zhang, L. Zhou, S. Liu, Z. Chen, Y. Liu, H. Wang, J. Li, L. He, S. Zhao, and F. Wei, "Neural codec language models are zero-shot text to speech synthesizers," 1 2023.

[3] Z. Zou, K. Chen, Z. Shi, Y. Guo, and J. Ye, "Object detection in 20 years: A survey," *Proceedings of the IEEE*, vol. 111, pp. 257–276, 2023.

[4] T. Räuker, A. Ho, S. Casper, and D. Hadfield-Menell, "Toward transparent ai: A survey on interpreting the inner structures of deep neural networks," 7 2022.

[5] D. Bau, B. Zhou, A. Khosla, A. Oliva, and A. Torralba, "Network dissection: Quantifying interpretability of deep visual representations," 4 2017.

[6] S. Kornblith, M. Norouzi, H. Lee, and G. Hinton, "Similarity of neural network representations revisited," 2019.

[7] B. Kim, M. Wattenberg, J. Gilmer, C. Cai, J. Wexler, F. Viegas, and R. Sayres, "Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav)," 11 2017.

[8] J. Frankle and M. Carbin, "The lottery ticket hypothesis: Finding sparse, trainable neural networks," *International Conference on Learning Representations*, 2019.

[9] Y. Wang, H. Su, B. Zhang, and X. Hu, "Interpret neural networks by identifying critical data routing paths," pp. 8906–8914, IEEE Computer Society, 12 2018.

[10] R. Csordás, S. van Steenkiste, and J. Schmidhuber, "Are neural nets modular? inspecting functional modularity through differentiable weight masks," 10 2020.

[11] J. Fiacco, S. Choudhary, and C. Rose, "Deep neural model inspection and comparison via functional neuron pathways," pp. 5754–5764, Association for Computational Linguistics, 2019.