

Investigating diagrammatic reasoning with deep neural networks

Duo Wang, Mateja Jamnik, Pietro Liò

Department of Computer Science and Technology, University of Cambridge, UK
wd263@cam.ac.uk, {mateja.jamnik, pietro.lio}@cl.cam.ac.uk

Abstract. Diagrams in mechanised reasoning systems are typically encoded into symbolic representations that can be easily processed with rule-based expert systems. This relies on human experts to define diagram-to-symbol mapping and the set of rules to reason with the symbols. We present a new method of using Deep artificial Neural Networks (DNN) to learn continuous, vector-form representations of diagrams without any human input, and entirely from datasets of diagrammatic reasoning problems. Based on this DNN, we developed a novel reasoning system, Euler-Net, to solve syllogisms with Euler diagrams. Euler-Net takes two diagrams representing the premises in a syllogism as input, and outputs either a categorical (subset, intersection or disjoint) or diagrammatic conclusion (generating an Euler diagram representing the conclusion) to the syllogism. Euler-Net can achieve 99.5% accuracy for generating syllogism conclusions, and learns meaningful representations. We propose that our framework can be applied to other types of diagrams, especially the ones we are less sure how to formalise symbolically.

Introduction

Diagrams have been shown to be effective tools for humans to represent and reason about complex concepts [1]. Several researchers have developed automated reasoning systems for diagrams. For example, Jamnik et al [2] developed DIAMOND for automating diagrammatic proofs of arithmetic theorems. Barwise et al [3] used blocks-world to teach and reason in first order logic with Hyperproof. Stapleton et al [4] developed Edith for automated Euler diagram theorem proving. Urbas et al [5] extended Edith to spider diagrams and developed Speedith. In these systems, mechanising reasoning with diagrams usually relies on methods of encoding diagrams as symbolic representation that can be easily processed with a rule-based program. Such methods rely on human experts to define the framework of diagram-to-symbol mapping and the set of rules to reason with the symbols. In this work, we developed a method using Deep artificial Neural Networks (DNN) to learn a continuous and vector-form neural representations of diagrams without any human input rules. With this method, diagrams can be encoded into neural representations, and reasoned about with subsequent neural networks or human-defined rules.

Recently, DNNs have achieved human comparable performance in several tasks such as image recognition [6], natural language translation [7]. DNNs' success in

tasks that humans are good at can be partly attributed to their biologically inspired architecture. Deep convolutional neural networks, a type of DNN applied for supervised image-based tasks, have both strong architectural and activity pattern similarity to visual cortex in the human brain [8]. This biological similarity motivates us to apply DNN to investigate diagrammatic reasoning. We developed Euler-Net, a type of DNN that performs syllogism reasoning with Euler diagrams. The DNN takes as input a number of Euler diagrams (premises), which show set relationships between sets contained in them. Euler-Net can generate a categorical conclusion (subset, intersection or disjoint) about the relationship between the sets with 99.5% accuracy. It can also learn using Generative Adversarial Network (GAN) to generate Euler diagrams that represent the set relationships without using any additional drawing tools. This enables Euler-Net to perform full diagrammatic inference, and shows that the learnt neural representations encodes essential information of the reasoning task.

Euler-Net is developed on, but not limited to, Euler diagram syllogism tasks. It can be applied to types of diagrams that are difficult to formalise into logic symbols, and can learn feature representations that capture essential information in the diagrams and subsequently analyse it. In our future work, we will adapt Euler-Net to a broader range of diagrams, and develop it into a useful tool for the diagram research community for investigating different types of diagrammatic representations.

In the rest of this paper we first give in Section 2 some background to neural nets and Euler diagrams. In Section 3 we present our deep neural net architecture and its use in Euler-Net to reason with Euler diagrams. Next, we evaluate our system in Section 4, and discuss these results and some future directions in Section 5. Finally we conclude in Section 6.

Background

Neural networks

Artificial Neural Network (ANN) is an information processing paradigm that is inspired by biological nervous systems. A general ANN consists of layers of artificial neurons connected in a graph, most often in an acyclic directed form. A single artificial neuron has very limited computational capability. However, when many of them are connected together to form an ANN, very complex functions can be approximately represented. ANN can be optimised by back propagation algorithm [9], a way to back propagate errors from higher layers to lower layers in order to correct weights assigned to lower layer inputs.

Recently, a particular type of ANN, a convolutional neural network (CNN) has made breakthroughs in various AI tasks such as image classification [6]. CNN consists of convolutional layers that apply learnable filters on images to extract features, and pooling layers that summarise local information. A typical CNN consists of multiple stacks of convolutional and pooling layers. The layers learn to extract progressively more complex image features (from edges and contours to objects). A deep CNN can learn highly complex feature representations that are similar to neural responses in the human visual cortex [8].

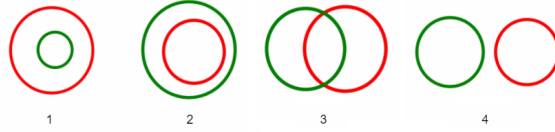


Fig. 1: Euler diagrams representing 4 possible relationships between non-empty sets A and B.

Euler diagrams

Euler diagrams [10] are a simple, yet effective diagrammatic representation for reasoning about set relationships. We will use a colour-coded modification of the Gergonne’s system of Euler diagrams [11] for its simplicity and visual clarity. In this system, minimal regions are assumed to be non-empty (i.e., the Gergonne system of Euler diagrams assumes existential import), and shading is not used. We assign a distinct colour to each contour instead of alphabet labels to denote classes. Colour coding facilitates the training of neural network by reducing the need to associate Alphabet labels with circled regions. There can be four different relationships between two sets A and B, which are: (1) $A \supset B$, (2) $A \subset B$, (3) $A \cap B \neq \emptyset$ and (4) $A \cap B = \emptyset$.

While in theory the fifth relationship $A = B$ is also possible, we do not consider it in this work because colour-coded contours will completely overlap and thereby diminish visual clarity – we leave to explore this in the future. Figure 1 illustrates how these 4 different set relationships can be represented by 4 different categories of colour-coded Euler diagrams (two sets denoted by Red and Green).

Euler diagrams are very effective in representing syllogisms. A syllogism consists of two premises that entail a conclusion. Figure 2 illustrates how the colour-coded Euler diagrams represent the syllogism “All *Green* are *Red*, all *Blue* are *Green*, therefore all *Blue* are *Red*”. In our task, we do not enforce the fixed-size contour constraint, which means that contours representing the same class can have varying sizes in different diagrams. As this Euler diagram system does not represent partial information, for certain premises there is not a single directly implied conclusion diagram, but several diagrams that are self-consistent [12] with the given premises. An example would be for premises “All B are A, some C are B”, consistent conclusions include “some C are A” and “all C are A”. We later show that neural networks can learn to reason with self-consistency and generate all conclusions that are consistent with the premises.

Neural network architecture

Euler-Net for categorical output

We built a system, called Euler-Net, which is a neural network trained to solve syllogisms represented with Euler diagrams. Euler-Net takes two Euler diagrams representing the premises of the syllogism as input, and outputs a categorical conclusion (subset, intersection or disjoint) for the syllogism. Figure 3 shows the

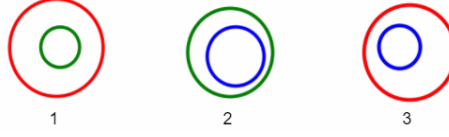


Fig. 2: Euler diagrams representing the syllogism “All *Green* are *Red*, all *Blue* are *Green*, therefore all *Blue* are *Red*”.

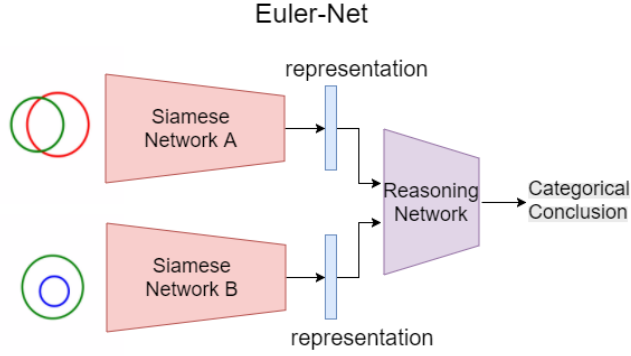


Fig. 3: Overview of the Euler-Net neural network architecture for Euler diagram syllogism reasoning.

architecture of Euler-Net. The first input diagram shows a relationship between a set *Red* and a set *Green*. The second diagram shows a relationship between sets *Green* and *Blue*. There are 4 possible categories for a categorical conclusion output from Euler-Net, as discussed in Section 2.2.

In the case of indeterminacy when no single logical conclusion can be drawn (e.g., All *Green* are *Red*, Some *Green* are *Blue*), the neural network outputs all conclusions that are consistent (not contradicting) with the premises. Neural networks are trained with pairs of premise diagrams and labels that encode correct conclusions into a binary (0 or 1) vector of length 4, with 4 positions in the vector representing each conclusion category as in Section 2.2. For example, vector [0100] encodes $Blue \subset Red$. While developed on the classical syllogisms, Euler-Net can be applied to diagram tasks with arbitrary number of contours in the diagram and arbitrary number of diagrams in the task. We also applied Euler-Net on tasks where there are 3 contours in each diagram. Namely, the first diagram contains *Red*, *Green* and *Blue* contours, while the second diagram contains *Green*, *Blue* and *Yellow* contours. The task is to infer consistent relationships between classes *Red* and *Yellow*. In Section 4 we show that for 3-contour task, the reasoning accuracy decrease is negligible.

Euler-Net is composed of two modules. The first module is a convolutional network that recognises the diagrams and encodes them into high-level neural feature representations. This network has a similar function to a visual cortex in the

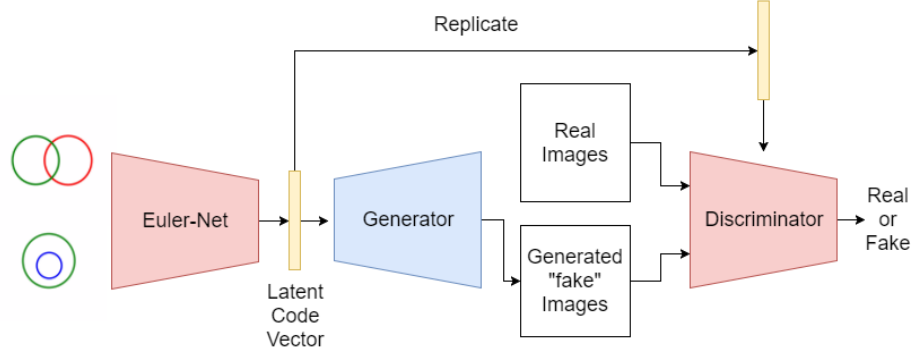


Fig. 4: Overview of diagram generation module for Euler-Net.

human brain [8], which transforms visual stimuli into neural code. The second module is a reasoning network that performs inference on the neural presentations of diagrams. This reasoning network extracts useful information from the neural representations in order to achieve accurate inference. The reasoning network consists of fully-connected layers that densely process the neural representations, and outputs the probability for each categorical conclusion. Euler-Net can be trained to minimise error rates in reasoning with standard Stochastic Gradient Descent (SGD) and a back-propagation algorithm [9]. Formally, the training objective is to minimise the loss function as in Equation 1:

$$L(D, T) = - \sum_{(d, t) \in (D, T)} \sum_i t_i \log f(d) + (1 - t_i)(1 - \log f(d)) \quad (1)$$

where D are premise diagrams, T are labels, (d, t) is a training sample of the problem set, t_i is the i^{th} element in the label vector, and $f(d)$ represents Euler-Net as a function of d .

Euler-Net for diagram generation

Instead of generating a categorical conclusion, Euler-Net can also generate diagrammatic conclusion of a syllogism, such as diagram 3 in Figure 2. This allows Euler-Net to perform complete diagrammatic inference. Diagrams can be generated from the neural representations of the syllogism problem without any human intervention or established drawing tools. This is accomplished by concatenating an image generator network to Euler-Net. Figure 4 illustrates the architecture of Euler-Net for diagram generation. This generator network uses latent neural code vector extracted from the last layer of Euler-Net to generate Euler diagrams that are consistent with the given premises. The latent code vector encodes consistent conclusions for the reasoning task. The generator network then transform this neural representation to an Euler diagram consistent with the given premises.

The generator network is trained with Generative Adversarial Network (GAN) [13] training objective, which recently became popular for generating high definition and sharp images. GAN consists of a generator network and a discriminator network that are jointly trained in a minimax game. The generator tries to generate images as real and accurate as possible, while the discriminator tries to distinguish between the generated and the correct images. The GAN training objective can be mathematically formulated as in Equation 2:

$$\min_G \max_D V(G, D) = \mathbf{E}_{x \sim p_{data}(x)} [\log D(x)] + \mathbf{E}_{z \sim p_z(z)} [\log(1 - D(G(z)))] \quad (2)$$

where G is the generator, D is the discriminator, x is a correct data sample, and z is the latent code vector. This can be viewed as a minimax game between G , which tries to minimise the objective, and D which tries to maximise it. During training, the parameters of the generator and the discriminator are updated alternatively to converge towards a dynamic equilibrium. The generator converges after 50000 iterations, and is able to generate an accurate and clear Euler diagram conclusion consistent with the given premises.

Evaluation

Euler-Net is trained with syllogism problems generated from an Euler diagram syllogism task generator. This generator firstly generates two random logical relationships for the first two premises, and then generates two Euler diagrams representing the two logical relationships with random size and position, as long as the logical relationships are not violated. Subsequently, the task generator generates consistent conclusions from a manually constructed truth table that maps any two premises to a set of consistent conclusions. For each consistent conclusion, corresponding Euler diagrams are also generated with random size and position. In total, we generated 96000 Euler syllogism reasoning problems for neural network training. For the 3-contour dataset, the diagrams and conclusions are generated in the same fashion. The truth table is larger as each premise now contains relationships between 3 classes, giving 4^3 possible cases.

During training, we divide the dataset into training set, validation set and test set with a split ratio of 8:1:1. We trained Euler-Net with the training dataset, tuned its performance with reported scores on the validation dataset, and finally, we evaluated the final performance on the test dataset. We report here the percentage accuracy which is defined as the number of syllogism problems correctly solved over the total number of problems. Euler-Net is able to achieve a nearly perfect accuracy of 99.5% on the 2-contour Euler syllogism tasks, and 99.4% on 3-contour task. In order to understand the performance results further, we separate test results for conclusive syllogism groups (a single logical conclusion) and inconclusive groups (multiple consistent conclusions). We found that Euler-Net achieves 100% accuracy for conclusive groups, which could indicate that conclusive syllogisms are relatively simpler than the inconclusive ones.

Euler-Net with added diagram generator can create high quality Euler diagrams from neural representations mostly without image artefacts. Figure 5 shows examples of Euler conclusion diagrams generated by Euler-Net for 2-contour and

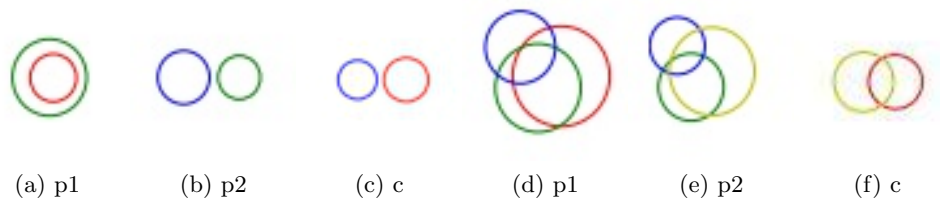


Fig. 5: Two examples of the Euler-Net generated diagrams. p1 and p2 are premises and c are Euler-Net generated conclusion diagrams.

3-contour task. This shows that Euler-Net learns neural representations that encode essential information of the syllogism reasoning problem. Such neural representations can be interpreted by a diagram generator network to create a diagrammatic conclusion.

Discussion

While we showed that DNN can perform diagrammatic reasoning and learn useful representations on the relatively simple Gergonne’s system of Euler diagram syllogism solving, DNN is not limited to a particular type of diagram. DNN provides a universal method for encoding all types of diagrams into neural codes that can be subsequently analysed. While simple diagrams like Euler diagrams can be conveniently formalised symbolically into sets of zones, labels and shadings, there are many types of diagrams that are more difficult to formalise. DNN can be applied to learn representations of such diagrams, and thus enable us to analyse aspects of those representations, such as for example, the difficulty level of the question, the amount of redundant information in the diagram, and how the core logic can be efficiently encoded with DNN in the sequence of diagrams. We will apply DNN to such more complex diagrams in our future work.

Euler-Net’s learning capacity can be increased for more complex diagrams than Euler diagrams with 3 contours by increasing architecture complexity, and including recent techniques such as residual network connections. While in theory for n -set Euler diagram, quadratic amounts of nodes are required in the network, most of the nodes can be designed as a reusable module for simple operations, similarly to how we use the CNN weights for input diagrams. We can also introduce a recent technique called “neural network attention”, which allows a neural network to reuse weights for learning simpler features such as 2-set relations – this would greatly improve scalability.

While Euler-Net achieves 99.5% accuracy, it is still not on-par with logical symbolic reasoner which is 100% accurate. However, conceptually, logical symbolic reasoners only reason with the symbols that represent diagrams, while Euler-Net reasons directly with the raw diagram input. The strength of DNN is its ability to capture feature representations for any type of diagram and its robustness to noise. In our future work we will extend Euler-Net to diagrams that are

drawn with noise or even missing parts, and where a symbolic reasoner may fail because of difficulty of transforming noisy diagrams into symbols. We will also develop Euler-Net on a full set of modern Euler-typed diagrams with shading and non-existential import, including concept diagrams and spider diagrams. We will develop Euler-Net as a neural-network-counterpart of modern diagram theorem provers like Speedith [5].

Conclusion

We developed Euler-Net, a deep neural network that can learn to perform diagrammatic reasoning about Euler diagram syllogism tasks. We illustrated that neural representations learnt by Euler-Net encode essential information about the input diagrams and the reasoning task. Euler-Net, while developed on Euler diagrams, is not constrained to a specific category of diagrams. Euler-Net can be readily adapted to other more complex diagrams where information extraction and formalisation is more difficult. We believe that deep neural networks can be developed into a useful tool for the diagram research community.

References

1. Simon, H.A., Larkin, J.H.: Why a diagram is (sometimes) worth 10,000 words. *Models of Thought* **2** (1987)
2. Jamnik, M., Bundy, A., Green, I.: On automating diagrammatic proofs of arithmetic arguments. *Journal of logic, language and information* **8**(3) (1999) 297–321
3. Barwise, J., Etchemendy, J.: *Hyperproof*. CSLI Press (1994)
4. Stapleton, G., Masthoff, J., Flower, J., Fish, A., Southern, J.: Automated theorem proving in Euler diagram systems. *Journal of Automated Reasoning* **39**(4) (2007) 431–470
5. Urbas, M., Jamnik, M., Stapleton, G.: Speedith: a reasoner for spider diagrams. *Journal of Logic, Language and Information* **24**(4) (2015) 487–540
6. 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*. (2016) 770–778
7. Wu, Y., Schuster, M., Chen, Z., Le, Q.V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., et al.: Google’s neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144* (2016)
8. Yamins, D.L., DiCarlo, J.J.: Using goal-driven deep learning models to understand sensory cortex. *Nature neuroscience* **19**(3) (2016) 356–365
9. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. *Proceedings of the IEEE* **86**(11) (1998) 2278–2324
10. Hammer, E., Shin, S.J.: Eulers visual logic. *History and Philosophy of Logic* **19**(1) (1998) 1–29
11. Gergonne, J.D.: *Essai de dialectique rationelle*. *Annales de Mathematiques pures et appliques* **7** (1817) 189–228
12. Shimojima, A.: *Semantic properties of diagrams and their cognitive potentials*. Center for the Study of Language & Information (2015)
13. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: *Advances in neural information processing systems*. (2014) 2672–2680