On the Trade-off between Fidelity and Teaching Complexity[⋆]

Brigt Arve Toppe Håvardstun[1], Cèsar Ferri[2], Jose Hernández-Orallo[2], Pekka Parviainen[1], and Jan Arne Telle[1]

[1] Department of Informatics, University of Bergen, Norway.
[2] VRAIN, Universitat Politècnica de València, Spain.
Brigt.Havardstun@student.uib.no, cferri@dsic.upv.es, jorallo@upv.es, Pekka.Parviainen@uib.no,Jan.Arne.Telle@uib.no

Abstract. In explainable AI there is usually a trade-off between the fidelity and the complexity of the explanation. In exemplar-based explanation systems, this complexity can be measured by the number or size of the examples needed for the human to grasp a concept that has sufficient fidelity with the AI system to be explained. In this paper we analyse a concept class of Boolean functions that is learned by AI systems (CNN) as bitmaps of possibly rotated and resized letters. We assume the human learner behaves like an inductive programming system with a strong prior (Karnaugh maps over Boolean functions) [1]. Our explanation procedure then behaves like a machine teaching session minimising a weighted expression of the complexity of the teaching set (the examples given to the human) and the fidelity of the taught Boolean function with respect to the original AI system. We explore the behaviour of this trade-off for varying numbers of training examples for the AI.

1 Background

The field of Explainable AI (XAI) [6] is becoming ever more critical with the growth in popularity of machine learning systems. In the field of XAI, there are multiple directions, one of them is example-based [6, 9, 11], where one aims to find examples showing how the machine learning system acts in different situations. Machine Teaching is the research area of actively selecting data sets used in teaching [14]. The goal is for the teacher to find the smallest training set —known as the *teaching* or *witness* set— that, using a learning algorithm, produces a target concept. In this work, we develop a simple proof of concept for using Machine Teaching techniques as a tool for XAI, see also [12, 13, 2]. We take the learner, who uses her learning algorithm, to be the human user, and the target concept to be (a part of) the black-box AI system that needs explanation. The machine teaching algorithm must find a small set of labelled examples that will allow the human to build her own model of the AI system and thereby arrive at an explanation of the target concept [8, 7]. We study a parameterised

[⋆] Supported by the Norwegian Research Council, project Machine Teaching for XAI.

framework where we explore the trade-off between Fidelity (squared error of the guessed model compared to the black box model) and Teaching Complexity (measured as the complexity of the set of labelled examples used as a teaching set) [5, 11]. We have the following framework:

$$T(\theta_{AI}) = \underset{S:\theta_{AI} \models S}{\operatorname{argmin}} \{\delta(S) + \mu \cdot \lambda(\theta_{AI}, \theta_M) : L_M(S) = \theta_M\} \tag{1}$$

$$L_M(S) = \underset{\theta_M:\theta_M \models S}{\operatorname{argmin}} \{\beta(\theta_M)\} \tag{2}$$

In these equations T is a teacher, aiming to teach a concept θ_{AI} to a human learner L_H, by finding a teaching set S such that $L_H(S) = \theta_{AI}$. To achieve automation and increase iteration speed a model L_M of L_H is used, and the teacher will therefore aim for $T(\theta_{AI}) = S$ **s.t.** $L_M(S) = \theta_{AI}$. We define δ as the complexity measure of a potential teaching set S, with focus on the size of the examples, as we have done in previous joint work [10]. The fidelity function λ measures how closely the guessed concept θ_M matches the concept θ_{AI}, while the factor μ allows us to balance the influence of complexity (δ) and fidelity (λ). In this work, we discuss an implementation of Equation 1 and aim to provide a Proof of Concept (PoC), showing the viability of this framework.

1.1 The AIs

For this PoC we implemented our own θ_{AI}s, and we decided their task should be to learn a Boolean function on four variables, $\phi(A, B, C, D)$, of all possible Boolean function. The input to θ_{AI} will be a bitmap containing a subset of letters from the alphabet $\Sigma = \{A, B, C, D\}$. The bitmaps representation of literals, with letters being rotated and scaled and placed randomly, within certain limits as indicated in Figures 1, 2 and 3, gives us the possibility of extensive training data for our AI. The output space of θ_{AI} is $\{0, 1\}$. For instance, with the concept $\phi = (A \wedge B) \vee (C \wedge D)$, we label an example 1 if ϕ evaluates to True, and 0 if ϕ evaluates to False. For the bitmap in Figure 3, we have $\phi(A = 1, B = 1, C = 1, D = 0) = (1 \wedge 1) \vee (1 \wedge 0) = 1$, and hence a label of 1.

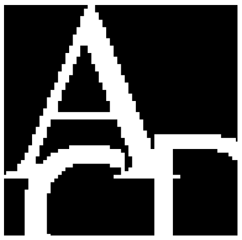

Fig. 1. Bitmap ACD, too big. Overlapping, and out of image.

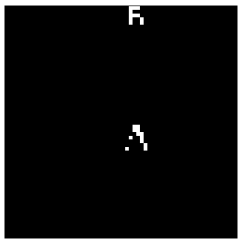

Fig. 2. Bitmap AB, too small. A is broken up, B looks like R.

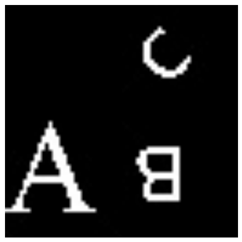

Fig. 3. Bitmap ABC, within limits, with rotation and scaling.

We implemented this AI system as a Convolutional Neural Network (CNN). The fact that CNNs are successful at their task, while at the same time not interpretable by themselves, makes CNN a good choice for our θ_{AI}. We implemented a CNN with 8 layers in Python using Keras and TensorFlow, see the master thesis [4] for details.

1.2 The model of the human

For simplicity, our model L_M of the human learner will not be given bitmaps as examples. Instead, it takes as input the letters present in each image. We thus hypothesize that a user will pay attention to the letters present in the bitmap image and disregard other information such as rotation, size and position.

The hypothesis class of L_M will consist of all Boolean functions over the 4-letter alphabet. Then, given a teaching set like $S = \{(AC, 0), (AD, 0), (BD, 0), (AB, 1), (BC, 1), (CD, 1)\}$, we must decide how L_M will act. We assume a human user would construct something like a partial truth table, in this case with 3 rows out of $2^4 = 16$ rows total filled with True, 3 rows filled with False, and 10 rows filled with Don't-Cares (x). Applying Occam's razor, we need to define the function β, to choose the Boolean function that is most simple and adheres to these constraints. A commonly accepted answer lies in using Karnaugh maps; see Figures 4 and 5 for two possible examples for this teaching set S.

CD\AB	00	01	11	10
00	x	x	1	x
01	x	0	x	0
11	1	x	x	x
10	x	1	x	0

Fig. 4. $\theta_M = (\neg A \wedge C) \vee (\neg C \wedge \neg D)$

CD\AB	00	01	11	10
00	x	x	1	x
01	x	0	x	0
11	1	x	x	x
10	x	1	x	0

Fig. 5. $\theta_M = (C \wedge D) \vee (B \wedge \neg D)$

Note we use disjunctive normal form (DNF) which mimics human reasoning. To verify a positive instance you need only to confirm one clause, whereas to confirm a negative instance you always need to check all clauses. The resource-heavy task of confirming a negative compared to a positive is somewhat similar to how humans are poor at negations [3]. For each teaching set there can be many DNFs possible, and in the spirit of K-map minimization we use the following scheme to pick the simplest. Firstly, each clause is sorted, in order, by 1) fewest variables, 2) fewest negations, 3) lexicographic order, while DNFs are sorted, in order, by 1) fewest clauses, 2) simplest clause. This defines β and gives us a unique Boolean formula in DNF form for each teaching set.

1.3 The fidelity function

When we want to compare θ_{AI} and θ_M, we need to view the former as an approximation to some Boolean function. For each subset of letters, we estimate what percentage of the bitmaps containing exactly these letters that θ_{AI} evaluates to True, rather than to False. We get values like the top row in Table 1. We observe that θ_{AI} predicts some letter groups the same and is more undecided on other letter combinations.

Symbol	$\emptyset$	A	B	C	D	AB	AC	AD	BC	BD	CD	ABC	ABD	ACD	BCD	ABCD
θ_{AI} predicts	0.00	0.00	0.00	0.95	0.00	0.99	0.02	0.00	0.63	0.02	0.91	1.00	1.00	0.04	0.74	1.00
θ_M evaluates	0	0	0	1	0	1	0	0	1	0	1	1	1	0	1	1

Table 1. Top row shows the percentage of bitmaps on letters for that column for which θ_{AI} evaluates to True. Bottom row shows the truth table of $\theta_M = (A \wedge B) \vee (C \wedge \neg A)$, and $\lambda(\theta_{AI}, \theta_M) = \frac{0.2222}{16} \approx 0.0139$ is the MSE of the difference of all 16 columns.

To evaluate how well θ_M matches θ_{AI}, we compare with the truth table for the Boolean function $\theta_M = (A \wedge B) \vee (C \wedge \neg A)$ and we get the truth table shown in Table 1. We then compare the two tables using Mean Square Error (MSE). We look at the difference between θ_{AI} and θ_M for each row and aggregate the square difference. For this θ_{AI} and θ_M, we get an error score of 0.2222. To get MSE, we divide by the 16 groups giving us a score of $\frac{0.2222}{16} \approx 0.0139$. The MSE score is the value that λ returns.

1.4 The complexity function

We have experimented with various definitions for the complexity function, to punish large and complicated teaching sets S. The chosen δ is a simple squared sum of the number of variables present in each example, plus 0.1 for the empty set (corresponding to setting no variable to True). We thus keep low the total number of variables in all examples while simultaneously putting a high cost on a single large example. Note that the δ values are typically much smaller than the λ values, so in our first set of experiments we set the multiplicative factor $\mu = 800$ when computing the aggregated score $\delta(S) + \mu \cdot \lambda(\theta_{AI}, \theta_M)$.

1.5 The teacher

The goal of the teacher is to find a teaching set explaining θ_{AI}, by iterating over potential teaching sets. For each teaching set S, we compute $L_M(S) = \theta_M$ as described earlier, and the aggregate score $\delta(S) + \mu \cdot \lambda(\theta_{AI}, \theta_M)$. During the iteration we retain the best aggregate score. For these first experiments the iteration is an exhaustive search. In the future we will work with Boolean formulas on more variables and replace this with some smart local search.

2 Experiments

Given the previous setting we performed a set of experiments using different concepts and parameters to analyse the effect of several elements to the machine teaching process and its result on explaining the behaviour of various AI models. In particular, we played with AI models trained on different sized training sets, which approximate the original Boolean function to different levels of accuracy. Depending on how well the AI is approximated by a Boolean function the trade-off parameter μ between fidelity (λ) and teaching complexity (δ) has different effects.

2.1 Fixed μ for varying AIs

We trained nine different AIs with differently sized data sets. In this first experiment, all AIs are trained with the ground truth $\phi = (A \wedge B) \vee C$ and the alphabet $\Sigma = \{A, B, C\}$. The data set sizes used in the experiment are: {10, 50, 100, 500, 1000, 2000, 5000, 10000, 50000}, accordingly we denote the different AIs: $\{AI_{10}, AI_{50}, AI_{100}, AI_{500}, AI_{1000}, AI_{2000}, AI_{5000}, AI_{10000}, AI_{50000}\}$.

In Table 2 we show several results. In the first row we see the expected result that the accuracy of the AI models wrt the original concept ϕ increases as more training examples were given to the neural network. In the next rows we show the Boolean expression that best approximates the AI, with its associated lowest possible λ-score over all Boolean functions. We see that the language of Boolean functions obtains a perfect match for the case of AI_{10} (because the underlying concept is very simple, always predicting True, which is a Boolean function) and almost perfect for AI_{10000} and AI_{50000} (because the number of training examples leads to a concept that is very close to ϕ). Note that also in other cases the most accurate Boolean function is ϕ (from AI_{2000} and up).

AIs	AI_{10}	AI_{50}	AI_{100}	AI_{500}	AI_{1000}	AI_{2000}	AI_{5000}	AI_{10000}	AI_{50000}
Accuracy AB+C	62.50	72.85	78.38	81.01	88.47	91.42	94.74	98.72	99.36
Lowest λ Boolean	Always True	A+B+C	A+B+C	AB+ AC+BC	AB+ AC+BC	AB+C	AB+C	AB+C	AB+C
Lowest λ	0	0.0730	0.0422	0.0774	0.0157	0.0248	0.0064	0.0006	0.0001
Model taught θ_M	Always True	A+B+C	A+B+C	A+C	AB+ AC+BC	AB+C	AB+C	AB+C	AB+C
Teaching Set S	{($\emptyset$,1)}	{($\emptyset$,0), (A,1), (B,1), (C,1)}	{($\emptyset$,0), (A,1), (B,1), (C,1)}	{($\emptyset$,0), (A,1), (C,1)}	{(A,0), (AB,1), (AC,1), (B,0), (BC,1), (C,0)}	{(A,0), (AB,1), (B,0), (C,1)}	{(A,0), (AB,1), (B,0), (C,1)}	{(A,0), (AB,1), (B,0), (C,1)}	{(A,0), (AB,1), (B,0), (C,1)}
$\delta(S)$	0.1	3.1	3.1	2.1	15	7	7	7	7
$\lambda(AI_x, \theta_M)$	0	0.0730	0.0422	0.0821	0.0157	0.0248	0.0064	0.0006	0.0001
$\delta + 800\lambda$	0.1	61.52	36.71	67.82	27.63	26.84	12.17	7.49	7.09

Table 2. Several AI models trained for increasing size of training examples for the concept ϕ =AB+C, shorthand for $(A \wedge B) \vee C$. We show accuracy with respect to ϕ, then the closest Boolean expression and its λ-value. Then we do teaching with $\mu = 800$, and show the metrics and values for the Boolean concept taught by the system.

Now let us look at the next few rows showing results for the teaching framework when run with the chosen parameter $\mu = 800$. First we show the Boolean concept θ_M that is actually taught by the system and note that it is almost always equal to the Boolean concept with lowest λ-value in the 2nd row. The only exception is AI_{500} where the trade-off between δ and λ favours the Boolean concept $A \vee C$ instead of $(A \wedge B) \vee (A \wedge C) \vee (B \wedge C)$ because the teaching set for the former is much simpler ($\delta = 2.1$) than the teaching set for the latter ($\delta = 15$ as can be seen under AI_{1000}). The next rows show the teaching set employed, its δ value, the λ value and the aggregate score.

There are three clear cases in the table (AI_{10}, AI_{10000} and AI_{50000}) where a simple teaching set allows the teacher to convey a concept to the learner that very closely captures the AI. But there are other cases, such as AI_{1000} and AI_{2000}, where the situation is less clear. For AI_{1000} the fidelity is not bad ($\lambda = 0.0157$) but the complexity of teaching becomes high ($\delta = 15$) so even if a sufficiently accurate concept can be taught this is at the cost of a higher effort from the learner. For AI_{2000} we see that this cost is reduced but the fidelity is worse ($\lambda = 0.0248$).

2.2 Varying μ for a single AI

In a second experiment we trained an AI model on a data set of size 350 for $\phi = (A \wedge B) \vee (C \wedge D) = AB + CD$ on 4 variables/letters. The accuracy was 78.25% and the closest Boolean function, with a fidelity value λ of 0.06, turned out to be $ABC + ABD + ACD + BCD$, which can be interpreted as "True if and only if at least 3 letters present". To investigate the trade-off between fidelity and complexity, teaching was done with varying values of μ, see left column in Table 3. We see that as μ increases more emphasis is put on fidelity at expense of complexity. Note that at $\mu = 3200$ the fidelity is as good as possible, but at the expense of a high complexity. The teaching set at $\mu = 3200$ is optimal for that optimal θ_M so increasing μ will have no effect. An option worth exploring is to take the characteristics of the human user into account when deciding on the fidelity vs complexity trade-off, e.g., having a high value of μ for an expert and a low value for a non-expert.

This second experiment also shows that it is not difficult to determine when the language used for the explanation leads to low fidelity and/or complex explanations. Actually, in this case, since the function captured by the AI does not have a clean Boolean concept, we can detect that teaching will either lead to low fidelity or complex explanation (or both). In sum, the use of the complexity of the teaching set in the trade-off is not only the right choice when doing example-based XAI but it also leads to the same insights as when the complexity of the concept is taken into account.

3 Next Steps

There are several next steps for this project, all of them focused on a better exploration of the fidelity vs complexity trade-off:

– Increase the complexity of the Boolean functions by allowing more variables

Range μ	Model taught θ_M	λ	δ	Teaching set
16	A	0.1880	1.1	{(∅, 0),(A,1)}
160-960	AC+BD	0.0881	12	{(A,0),(B,0),(C,0),(D,0),(AC,1),(BD,1)}
1120-1840	AC+BCD+AD	0.0718	30	{(A,0),(AC,1),(AD,1),(BC,0),(BD,0),(CD,0)}
1920-2400	AC+ABD+BCD	0.0656	42	{(AC,1),(AB,0),(AD,0),(BC,0),(BD,0),(CD,0), (ABD,1),(BCD,1)}
3200 - ∞	ABC+ABD+ ACD+BCD	0.0600	60	{(AB,0),(AC,0),(AD,0),(BC,0),(BD,0),(CD,0), (ABC,1),(ABD,1),(ACD,1),(BCD,1)}

Table 3. Results for a single AI model where the closest Boolean function turned out to be $ABC + ABD + ACD + BCD$. Teaching was done with varying values of μ, see left column. As μ increases more emphasis is put on lower fidelity $\lambda = \lambda(\theta_{AI}, \theta_M)$ at expense of higher teaching complexity δ.

– Use local search for the teaching set subset selector
– Focus on θ_{AI} of intermediate accuracy
– Experiment with different μ values

We also want to test the resulting teaching sets on humans, and update L_M and δ as necessary. In related work Yang et al. [13] measured the effectiveness of examples-based explanations for AI using Bayesian Teaching, aiming for high sensitivity and high specificity, and we will compare our results to theirs.

References

1. Gulwani, S., Hernández-Orallo, J., Kitzelmann, E., Muggleton, S.H., Schmid, U., Zorn, B.: Inductive programming meets the real world. Communications of the ACM **58**(11), 90–99 (2015)
2. Hernández-Orallo, J., Ferri, C.: Teaching and explanations: aligning priors between machines and humans. Human-Like Machine Intelligence pp. 171–198 (2021)
3. Hoosain, R.: The processing of negation. Journal of Verbal Learning and Verbal Behavior **12**(6), 618–626 (Dec 1973). https://doi.org/10.1016/S0022-5371(73)80041-6, https://www.sciencedirect.com/science/article/pii/S0022537173800416
4. Håvardstun, B.A.T.: Machine Teaching for Explainable AI: Proof of Concept. Master's thesis, University of Bergen, Department of Informatics (2022)
5. Lipton, P.: Contrastive explanation. Royal Institute of Philosophy Supplements **27**, 247–266 (1990)
6. Molnar, C.: Interpretable machine learning. Lulu. com (2020)
7. Ortega, A., Fierrez, J., Morales, A., Wang, Z., Ribeiro, T.: Symbolic AI for XAI: Evaluating LFIT inductive programming for fair and explainable automatic recruitment. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 78–87 (2021)
8. Pisano, G., Ciatto, G., Calegari, R., Omicini, A.: Neuro-symbolic computation for xai: Towards a unified model. In: WOA. vol. 1613, p. 101 (2020)
9. Ribeiro, M.T., Singh, S., Guestrin, C.: Anchors: High-precision model-agnostic explanations. In: Proceedings of the AAAI conference on artificial intelligence. vol. 32 (2018)

10. Telle, J.A., Hernández-Orallo, J., Ferri, C.: The teaching size: computable teachers and learners for universal languages. Machine Learning (2019), https://doi.org/10.1007/s10994-019-05821-2
11. van der Waa, J., Nieuwburg, E., Cremers, A., Neerincx, M.: Evaluating XAI: A comparison of rule-based and example-based explanations. Artificial Intelligence **291**, 103404 (2021)
12. Yang, S.C.H., Shafto, P.: Explainable artificial intelligence via bayesian teaching. In: NIPS 2017 workshop on Teaching Machines, Robots, and Humans. pp. 127–137 (2017)
13. Yang, S.C.H., Vong, W.K., Sojitra, R.B., Folke, T., Shafto, P.: Mitigating belief projection in explainable artificial intelligence via Bayesian teaching. Scientific Reports **11**(1), 9863 (Dec 2021). https://doi.org/10.1038/s41598-021-89267-4, http://www.nature.com/articles/s41598-021-89267-4
14. Zhu, X., Singla, A., Zilles, S., Rafferty, A.N.: An overview of machine teaching (2018), http://arxiv.org/abs/1801.05927