Learning Decision Trees Recurrently through Communication

Stephan Alaniz [1 2] **Diego Marcos** [3] **Zeynep Akata** [2]

Abstract

Integrating explainability into deep neural networks without sacrificing prediction accuracy has the potential of greatly improving its value to the user. We propose to do this by exposing the steps taken through the decision-making process in a transparent manner. Instead of assigning a label to an image via deep learning in a single step, we propose to learn a few iterative binary sub-decisions, building a decision tree whose structure is given by two agents that communicate through message passing and is encoded into the memory representation of an RNN. In addition, our model assigns a semantic meaning to each binary decision in the form of attributes, providing concise, semantic and relevant rationalizations to the user.

1. Introduction

Classification decisions of deep neural networks (DNN) are often hard to interpret, hindering their practical employment in critical applications such as health-care. When explanations are critical, decision tree models are often preferred over more complex models as they are inherently interpretable through introspection. One goal of introspection is to reveal the internal thought process of the decision maker to a machine learning practitioner (Park et al., 2018). However, it has been argued that models that are optimized for offering explanations through introspection need to assume a loss in performance (Gunning & Aha, 2019). Indeed, decision trees are not competitive on most computer vision tasks as they fail to approach the accuracy of state-of-the-art neural networks.

We overcome shortcomings of classical decision trees by incorporating recent advances in multi-agent communication (Foerster et al., 2016) and by embedding the decision tree structure into the memory representation of a recurrent neural network (RNN). Our proposed model uses message-passing with discrete symbols that can be mapped to a human-understandable vocabulary, greatly improving its interpretability.

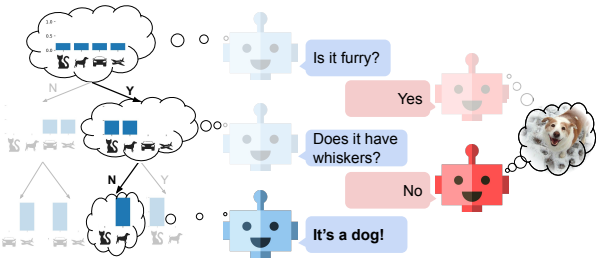

Figure 1. Our recurrent decision tree is learned by two agents communicating: *Recurrent decision tree* (left) asks questions, *attribute-based learner* (right) answers these questions with a yes/no answer which improves the classification after each step.

2. Recurrent Decision Trees via Communication: RDTC

Our Recurrent Decision Tree via Communication (RDTC) framework is a sequential interaction between two agents to solve an image-classification task through communication (see Figure 1). The first agent, i.e., the Recurrent Decision Tree (RDT), learns a decision tree that allows introspection. The second agent, i.e., Attribute-based Learner (AbL), learns attributes that provide rationales to make the communication human-understandable.

2.1. Agent to Agent Communication

For any single image x, our RDT agent tries to iteratively aggregate information into an explicit memory $\mathcal{M}$ that is sufficient to discriminate the correct class label y from the set of all classes $\mathcal{Y}$. To gather more information, the RDT agent sends a query message c_t to the AbL agent. The AbL answers the query c_t in regard to the image x with a binary response $d_t \in \{0, 1\}$ that the RDT uses to update its explicit memory $\mathcal{M}^{(t)} = \mathcal{M}^{(t-1)} \oplus (c_t, d_t)$ to improve its prediction of the class y. This constitutes one iteration t of the agent to agent communication.

Communication Protocol. Each work in the vocabulary $|A|$ corresponds to a binary attribute $a \in A$ that the AbL

[1]Max Planck Institude for Informatics [2]University of Tübingen [3]Wageningen University and Research. Correspondence to: Stephan Alaniz <salaniz@mpi-inf.mpg.de>.

Submission to the XXAI Workshop of ICML 2020, Vienna, Austria, 2020.

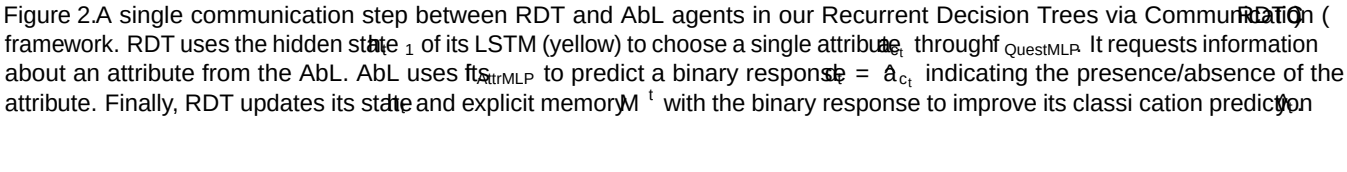

Figure 2. A single communication step between RDT and AbL agents in our Recurrent Decision Trees via Communication (RDTC) framework. RDT uses the hidden state h_{t-1} of its LSTM (yellow) to choose a single attribute a_{c_t} through f_{QuestMLP}. It requests information about an attribute from the AbL. AbL uses f_{AttrMLP} to predict a binary response $d_t = \hat{a}_{c_t}$ indicating the presence/absence of the attribute. Finally, RDT updates its state h_t and explicit memory $M^{(t)}$ with the binary response to improve its classification prediction $\hat{y}_t$.

agent predicts about a given image, e.g. black beak for a category as it guarantees the communication signals to always be ordinal. The AbL can learn to attach a human-understandable discrete during training.
meaning to the words when annotated attribute data is available. At each communication step t, the RDT chooses one attribute a_{c_t} from the vocabulary, identified by its index c_t, and requests the presence or absence of this attribute in the image. The AbL then provides this binary information d_t based on its own prediction. We deliberately limit the AbL's messages to be binary as clear yes/no answers are easier to interpret than probability values, e.g., the beak is either black or not rather than being 0.2% black.

Discrete Messages. To select a discrete index c_t, we use the Gumbel-softmax estimator (Jang et al., 2017; Maddison et al., 2017) to sample from a categorical distribution via the reparameterization trick (Kingma & Welling, 2014)

$$\text{GumbelSoftmax}(\log \alpha)_i = \frac{\exp((\log \alpha_i + g_i)/\tau)}{\sum_{j=1}^{K} \exp((\log \alpha_j + g_j)/\tau)} \tag{1}$$

where $\log \alpha$ are the unnormalized log-probabilities of the categorical distribution, τ is temperature parameter and g_i Gumbel.

On the other hand, the binary responses $d_t \in \{0, 1\}$ (presence of absence of an attribute in image x) from the observer are not required to be stochastic. Therefore, we use regular softmax (Hinton et al., 2015) with a temperature

$$\text{TempSoftmax}(\log \alpha)_i = \frac{\exp(\log \alpha_i / \tau)}{\sum_{j=1}^{K} \exp(\log \alpha_i / \tau)} \tag{2}$$

which approximates an $\arg\max$ function deterministically as τ approaches 0. Popular training strategies include annealing the parameter τ over time or augmenting the soft approximation with an $\arg\max$ function that discretizes the activation in the forward pass and a straight-through identity function in the backward pass. We resort to the second strategy as it guarantees the communication signals to always be discrete during training.

2.2. Recurrent Decision Tree (RDT) Agent

RDT consists of three parts, an explicit memory M, an LSTM (Hochreiter & Schmidhuber, 1997), and a question-decoder module Q, Question MLP as shown in Figure 2. To decide on the attribute information to request, RDT uses f_{QuestMLP} for decoding the last hidden state h_{t-1} into a categorical distribution $\log p(c_t | h_{t-1}) = f_{\text{QuestMLP}}(h_{t-1})$ where $p(c_t | h_{t-1})$ indicates the likelihood of requesting a particular attribute from the AbL. We denote the attribute index $c_t \in \{1, \dots, |A|\}$ as a sample from $p(c_t | h_{t-1})$ obtained by applying the Gumbel-softmax estimator.

After each iteration of the communication loop, RDT updates its explicit memory with the new response $M^{(t)} = M^{(t-1)}(c_t, d_t)$. Moreover, RDT updates its hidden LSTM state with the newly updated explicit memory $M^{(t)}$ and the AbL agent's binary response as follows: $h_t = \text{LSTM}(h_{t-1}; M^{(t)}; c_t; d_t)$.

At each time step the explicit memory M is used to predict the class label $\hat{y}_t = f_{\text{ClassMLP}}(M^{(t)})$.

Since the primary objective of the RDT agent is to maximize the classification performance, we minimize the cross-entropy loss of the predicted class probabilities $\hat{y}_t$ and the true class probabilities y with $L = \frac{1}{T} \sum_{t=1}^{T} L_{CE}(y, \hat{y}_t) = \frac{1}{T} \sum_{t=1}^{T} \sum_i y_i \log \hat{y}_{t,i}$. By averaging the cross-entropy loss over all T time steps, we encourage the RDT to predict the correct class in as few communication steps as possible.

At test time, we do not sample c_t, but instead always take the index with highest probability, i.e. $\arg\max$. With a deterministic choice of c_t there are exactly two distinct states for both $M^{(t)}$ and h_t given the previous time step, resulting in a binary decision tree.

Model	AWA2	aPY	CUB	MNIST	CIFAR-10	ImageNet
ResNet (He et al., 2016)	98.2 ± 0.0	85.1 ± 0.6	79.0 ± 0.2	99.4* ± 0.1	93.3 ± 0.2	73.0 ± 0.1
dNDF (Kontschieder et al., 2016)	97.6 ± 0.2	85.0 ± 0.6	73.8 ± 0.3	99.2 ± 0.1	**93.2** ± 0.1	72.6 ± 0.1
RDTC (Ours)	**98.0** ± 0.1	**85.7** ± 0.7	**78.1** ± 0.2	**99.3** ± 0.1	93.1 ± 0.1	**72.8** ± 0.1
aRDTC (Ours)	**98.1** ± 0.0	**85.3** ± 0.3	**77.9** ± 0.6	N/A	N/A	N/A

Table 1. Comparing our aRDTC (λ = 0.2) and RDTC (λ = 0) to the non-explainable ResNet and closely related dNDF. As MNIST, CIFAR-10 and ImageNet do not have attributes, aRDTC is not applicable. (std is over 5 runs, * is simple CNN)

2.3. Attribute-based Learner (AbL) Agent

The Attribute-based Learner (AbL) agent provides the RDT agent with a binary response to its request based on a set of binary attributes $\hat{\mathbf{a}}$ about the image, as shown in Figure 2. To do this, the AbL feeds its CNN image features z to f_{AttrMLP} that models a probability distribution over a set of learned binary attributes $\log p(\hat{\mathbf{a}}|z) = f_{\text{AttrMLP}}(z)$. By applying softmax with temperature, we obtain binary attributes $\hat{\mathbf{a}} \in \{0,1\}^{|A|}$, the discretization of $p(\hat{\mathbf{a}}|z)$: $\hat{\mathbf{a}} = \text{TempSoftmax}(f_{\text{AttrMLP}}(z))$. Whenever the RDT requests a particular attribute with its query c_t, the AbL simply returns the binarized attribute of the specified index to the AbL. We denote this selection operation as $d_t = \hat{\mathbf{a}}_{c_t}$.

Note that the attributes $\hat{\mathbf{a}}$ are either discovered via end-to-end learning optimizing a classification loss, i.e., we refer to this setting shortly as RDTC or they are predicted as human-interpretable concepts using the attribute loss explained in the following, i.e., we refer to this setting shortly as aRDTC.

Attribute Loss. RDTC learns splits that best separate the data. However, these are not always easy to interpret. We propose to learn attributes $\hat{\mathbf{a}}$ that align with class-level human-annotated attributes α making them interpretable. A second cross-entropy term encourages this correspondence:

$$\mathcal{L} = \frac{1}{T}\sum_{t=1}^{T}\Big[(1-\lambda)\mathcal{L}_{CE}(y,\hat{y}_t) + \lambda\mathcal{L}_{CE}(\alpha_{y,c_t}, \hat{\mathbf{a}}_{c_t})\Big] \quad (3)$$

weighted by a hyperparameter λ. Here, α_{y,c_t} corresponds to the ground-truth attribute label of class y that matches the observer's response $\hat{\mathbf{a}}_{c_t}$. The final loss ($\mathcal{L}$) encourages the network to learn attributes that agree with human-annotated attributes while optimizing for classification accuracy.

3. Experiments

We experiment on six datasets. AWA2 (Lampert et al., 2014), CUB (Wah et al., 2011), aPY (Farhadi et al., 2009) are three benchmark attribute datasets proposed by the computer vision community. MNIST (Lecun et al., 1998), CIFAR-10 (Krizhevsky, 2009) and ImageNet (Russakovsky et al., 2015) serve to evaluate our model's classification performance without attribute information.

3.1. Comparing with the State of the Art

In this section, we compare our models aRDTC and RDTC with Deep Neural Decision Forests (dNDF) (Kontschieder et al., 2016). ResNet-152 (He et al., 2016) pre-trained on ImageNet and fine-tuned on each of the datasets (except from MNIST where we use a simple CNN) including its softmax classifier serves as non-explainable deep neural network (ResNet). After this pretraining, we fix the weights of the perception module and extract the same CNN image features z for our aRDTC and RDTC, as well as dNDF. Deep Neural Decision Forest (dNDF) (Kontschieder et al., 2016) explicitly models the decision tree by mapping each inner tree node to a output neuron defining the routing probabilities of the input to the leaves through exhaustive tree traversal. While dNDF uses soft splits, in aRDTC and RDTC we use hard binary splits for each decision tree node for improved interpretability.

Results. From the results in Table 1 we observe that, our RDTC not only outperforms dNDF (e.g., 78.0% vs 73.8% on CUB), the state of the art deep decision tree model, our model exhibits improved interpretability, because we use hard instead of soft binary splits. Moreover, our model trained with the attribute loss, i.e. aRDTC, is able to give a semantic meaning to the splits, without a significant loss in accuracy. Although RDTC and aRDTC work with constrained single-bit communications to improve explainability, they succeed in maintaining the accuracy of the non-explainable state-of-the-art across all datasets.

3.2. Qualitative Results

Illustrating the decision making process helps the user get an explainable overview of the internal decision process of the whole classifier. We point to the tree branch into which a certain class (indicated by an example image from this class) falls and we present the attribute that is associated by each branch on CUB in Figure 3 where the left and right sub-tree

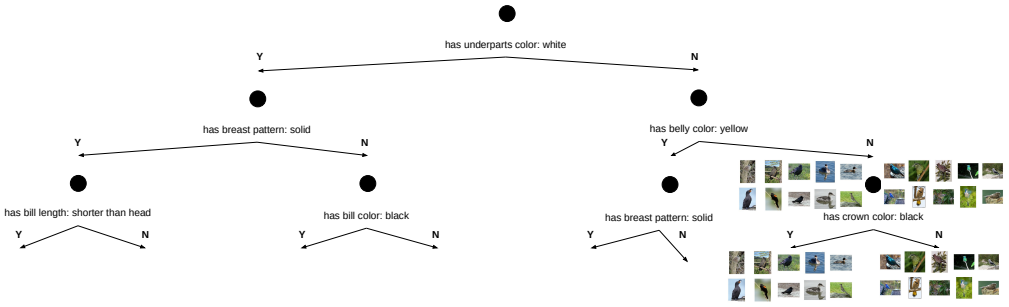

Figure 3. Our aRDTC learns explainable decisions via the decision tree and gives each decision a human-understandable meaning. Here we show the first three decisions for a subset of the 200 classes of birds in CUB where each image represents a class.

indicates that the attribute is present or absent respectively.

4. Conclusion

In this work, we propose to learn a decision tree recurrently through communication between two-agents. Our RDTC framework uses hard as opposed to soft binary splits for easier interpretation. Combining the end-to-end learning with human-annotated side information in aRDTC yields an explainable decision tree model that exposes a sequence of semantic subdecisions when classifying fine-grained image datasets while maintaining the accuracy of non-explainable deep models.

References

Farhadi, A., Endres, I., Hoiem, D., and Forsyth, D. Describing objects by their attributes. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

Foerster, J. N., Assael, Y. M., de Freitas, N., and Whiteson, S. Learning to communicate with deep multi-agent reinforcement learning. In *Advances in Neural Information Processing Systems*, 2016.

Gunning, D. and Aha, D. W. Darpa's explainable artificial intelligence program. *AI Magazine*, 40(2):44–58, 2019.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016*, 2016.

Hinton, G. E., Vinyals, O., and Dean, J. Distilling the knowledge in a neural network. *CoRR*, abs/1503.02531, 2015.

Hochreiter, S. and Schmidhuber, J. Long short-term memory. *Neural Computation*, 9(8), 1997.

Jang, E., Gu, S., and Poole, B. Categorical reparameterization with gumbel-softmax. In *ICLR*, 2017.

Kingma, D. P. and Welling, M. Auto-encoding variational bayes. In *ICLR*, 2014.

Kontschieder, P., Fiterau, M., Criminisi, A., and Bulò, S. R. Deep neural decision forests. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI*, 2016.

Krizhevsky, A. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

Lampert, C. H., Nickisch, H., and Harmeling, S. Attribute-based classification for zero-shot visual object categorization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(3), 2014.

Lecun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), 1998.

Maddison, C. J., Mnih, A., and Teh, Y. W. The concrete distribution: A continuous relaxation of discrete random variables. In *ICLR*, 2017.

Park, D. H., Hendricks, L. A., Akata, Z., Rohrbach, A., Schiele, B., Darrell, T., and Rohrbach, M. Multimodal explanations: Justifying decisions and pointing to the evidence. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., and Fei-Fei, L. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3), 2015.

Wah, C., Branson, S., Welinder, P., Perona, P., and Belongie, S. The Caltech-UCSD Birds-200-2011 Dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011.