Boosted Supervised Intensional Learning Supported by Unsupervised Learning

A. C. M. Fong and G. Hong

***Abstract*—Traditionally, supervised machine learning (ML) algorithms rely heavily on large sets of annotated data. This is especially true for deep learning (DL) neural networks, which need huge annotated data sets for good performance. However, large volumes of annotated data are not always readily available. In addition, some of the best performing ML and DL algorithms lack explainability – it is often difficult even for domain experts to interpret the results. This is an important consideration especially in safety-critical applications, such as AI-assisted medical endeavors, in which a DL's failure mode is not well understood. This lack of explainability also increases the risk of malicious attacks by adversarial actors because these actions can become obscured in the decision-making process that lacks transparency. This paper describes an intensional learning approach which uses boosting to enhance prediction performance while minimizing reliance on availability of annotated data. The intensional information is derived from an unsupervised learning preprocessing step involving clustering. Preliminary evaluation on the MNIST data set has shown encouraging results. Specifically, using the proposed approach, it is now possible to achieve similar accuracy result as extensional learning alone while using only a small fraction of the original training data set.**

***Index Terms*—Intelligent computing, machine intelligence, machine learning, neural networks, intensional information, semi-supervised learning.**

I. Introduction

Machine learning (ML) is an aspect of artificial intelligence (AI) that is among the most well-known, especially among members of the general public. ML, and in particular a branch of ML known as deep learning (DL) have shown great promises in recent years. Tasks and applications that were considered difficult or even impossible a few years ago, such as autonomous vehicles [1], unmanned aerial vehicles (UAV) [2], near-human level speech processing [3], and game playing, have repeatedly made news headlines. The numerous reported success stories belie the fact that despite showing near-human or even super-human capabilities in an increasingly wide range of tasks, ML / DL lack true intelligence in a human sense and are vulnerable to possible malicious attacks.

DL algorithms run on deep neural networks (NN). Deep NN are characterized by having many (hundreds or more) hidden layers sandwiched between an input layer, which accepts input data, and an output layer, which outputs a prediction on new / unseen data once the NN has been properly trained.

NN have been around for decades, but recent advances in four areas of computing and engineering have led to an explosive growth in the use of DL NN. The first of these is in the area of advanced hardware architectures that support the large amounts of computation required for properly training DL NN. The second is the availability of big data, which is partly brought about by the democratization of content / data generation and dissemination in the age of the world wide web (WWW). Anyone with access to the internet can put content on the web and social media websites, and people have been doing so for years. The increasing popularity of IOT sensors is another source of big data.

The third factor is related to algorithmic improvements in NN training. Whereas in the past it was not feasible to effectively and efficiently train deep NN due the the vanishing gradient problem, the bottlenecks have largely been suppressed. Finally, ML and DL libraries, such as scikit-learn [4] and keras [5], have lowered the barrier to development of ML and DL models. Thus, all four factors contribute to the popularity of ML and DL in a broad sense and with successes reported in a wide range of application domains.

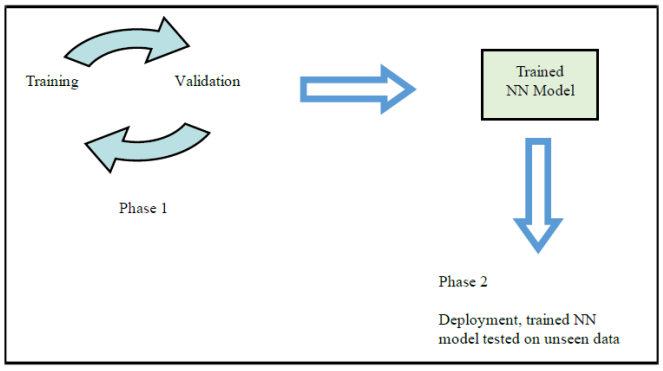

Fig. 1. Application of NN.

As with other supervised learning techniques, application of NN involves a minimum of two phases as shown in Fig. 1. During Phase 1, the NN is trained on training data of the form (input, target) pairs. Thus, for every input there is a "correct"

Manuscript received August 25, 2019; revised May 22, 2020.

The authors are with Department of Computer Science, Western Michigan University, Kalamazoo, MI 49009 USA (e-mail: acmfong@gmail.com).

doi: 10.18178/ijmlc.2021.11.2.1020

answer associated with it, and it is the learning algorithm's task to learn that. Validation is performed both to optimize all NN model parameters and to estimate the model's ability to generalize knowledge from the data. Once the NN has been properly trained, it becomes a trained model of knowledge base that is ready for deployment. Once deployed, the model will make predictions on new / unseen input data. Ultimately, a model's performance is judged upon how it performs on new / unseen data.

The training process seems nothing remarkable. It amounts to a fairly mechanical process of adjusting network weights using backpropagation in order to minimize a set loss function. Adjustment of the hyperparameters, such as network depth and number of units per layer, requires more work. Importantly, NN and especially DL NN require large volumes of training data to achieve strong predictive performance. This paper investigates the use of intensional information for NN learning [6] to reduce the reliance on such large volumes of data, which can be expensive to obtain because of the need to annotate the training data.

The rest of the paper is organized as follows. Section II outlines the general idea of intensional learning. Section III then proposes a boosted intensional learning pipeline, which uses unsupervised clustering as a preprocessing step, to provide optimal performance without heavy reliance on annotated extensional data. Section IV presents an evaluation of the learning pipeline, which demonstrates its applicability with promising results. Finally, Section V concludes the paper and suggests future research directions.

II. Learning from Intensional Information

Current training scheme for NN, as outlined in Fig. 1, is heavily reliant on data that contain extensional information. Extensional information captures the scope or extent of a concept, such as a pedestrian, a passenger car, a bicycle, fire engine, or a pickup truck. When taken to an extreme, the full extent of a concept like passenger car is to include all passenger cars that ever existed or continue to exist in the world. In the case of NN training, the data used to recognize and distinguish between, say, pickup trucks and passenger cars are typically in the form of images of these vehicles (input) with corresponding labels (target).

Therefore, when taken to extreme, current training scheme for NN would ideally involve training the network with all available images of passenger cars and trucks. This "everything under the sun, past, present, and future" approach to training NN is clearly inefficient and does not really mimic how humans learn. Clearly, a better approach is very much needed.

In fact, humans can typically learn from much smaller sample sizes to recognize and distinguish between, say, passenger cars and pickup trucks. Furthermore, humans, even young children who have only learned to recognize passenger cars and pickup trucks, can often extrapolate and quite easily identify other types of vehicles, such as minivans, vans, fire engines, and special utility vehicles (SUV). The underlying reason is that people learn to associate salient characteristics, such as a compartment riding on a set of wheels, with passenger cars and pickup trucks. These are some of the components that characterize the concept of a type of vehicles and is what intensional information is all about.

Intensional learning is intended to address the inefficiency associated with extensional learning and a deliberate attempt at injecting an element of how humans really learn. So, instead of trying to get a network to learn as many images of passenger cars and pickup trucks as possible (so that the network can self-learn the important features by passing information through its many layers), intensional learning emphasizes learning those important features that characterize a passenger car as well as features that characterize a pickup truck.

Because DL NN are often touted to have the ability to self-learn useful features obsolescing feature engineering, this might seem like taking a step backward to classical ML techniques that require a significant amount of feature engineering. However, the efficiency achievable will potentially outstrip any additional work required. It is also possible to lead to the kind of extrapolation described above: once a network model can recognize passenger cars and pickup trucks by knowing the intensional information about these vehicles, there is an elevated opportunity of recognizing also other similar types of vehicles, such as minivans. In addition, by applying unsupervised learning, such as clustering, for extracting intensional information, the amount of additional work involved can be kept to a manageable level.

In fact, by employing a light-touch approach to feature engineering, one that is driven by an unsupervised learning preprocessing step, it is a goal of this research to *simultaneously* reduce the amount of labelled extensional data required and reduce model complexity. The latter can mean shallower networks than otherwise required, which will have a positive impact on training efficiency. This can translate to faster training, less energy consumption, less expensive computer architecture, etc.

In general, simpler models can also lead to improved interpretability. Interpretable models are easier to understand and fine tune. They also potentially facilitate analysis of failure modes, so that in the future there will be increased opportunities of gaining insight into how and when high-performance ML and DL models fail. It is anticipated that such elevated understanding of ML/DL failure modes will in turn facilitate research into effective mitigating strategies.

Improved interpretability comes with another potential benefit. Prospective adversaries would find it harder to formulate malicious attacks once clarity replaces opacity. Any sabotage attempts would be more easily detected during both phases of training and deployment. During training and validation, attempts at injecting malicious data would be easily detected because intensional information that characterizes the input data can be captured. In Phase 2, once the model is deployed, the improved interpretability would establish clearer caused and effect relations between the model's predicted output and any malicious test data supplied to the model.

III. Boosted Intensional Learning with Clustering

Boosting has been found to be very effective in improving the performance of decision-tree based ML algorithms. In fact, gradient boosted machines (GBM), which are ensembles of boosted decision trees [7], [8], are among the best performing ML techniques for classifying non-perceptual data. In addition, boosting is applicable to other ensembles, and this is an aspect of the research presented in this paper.

Fig. 2 shows the pipeline of a boosted intensional learning algorithm that includes an unsupervised learning preprocessing step, which is in the form of k-means clustering. The pipeline starts with K-means clustering preprocessing using an algorithm presented in Fig. 3. It serves an important purpose, which is to organize data into similar groups to aid extract of relevant intensional information for training the NN.

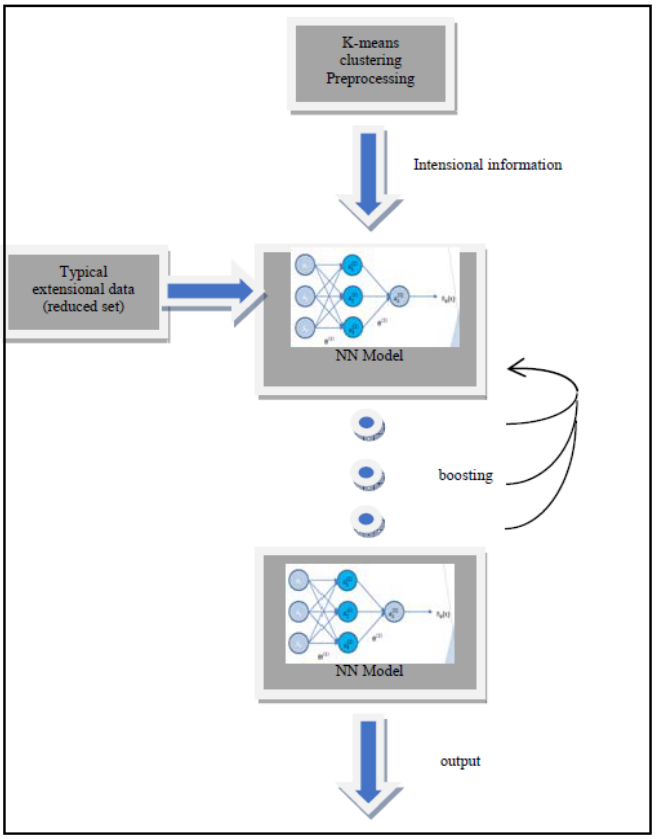

Fig. 2. Pipeline of boosted intensional learning supported by k-means clustering.

Set K
Initialization: randomly assign one of K class labels to each of the n data points

Iterate the following two sub-steps until convergence is reached

a) For each of the K clusters, compute the cluster centroid.
b) Assign each data point to the cluster with the closest centroid.

Fig. 3. K-means clustering algorithm.

In the algorithm shown in Fig. 3, convergence means no further changes are observed in class label assignments. In sub-step a), the k^{th} cluster centroid is the vector of the p feature mean for the samples in the kth cluster. In sub-step b), proximity is determined based on some popular distance measure, e.g. Euclidean distance. K-means clustering results in a flat structure of K groups of items that are semantically similar within each group.

In this research, K-means clustering serves as an important preprocessing step that extracts useful intensional information for subsequent learning. At the same time, typical annotated samples with extensional information are also presented to the NN for training. A shallow network is used with a small fraction of annotated samples that would typically be used for DL training. Boosting is applied to gradually build up the shallow NN model using an adjustable learning rate.

Finally, Phase 1 is complete when the NN is fully trained (when slight overfitting is just observable) and ready for deployment in Phase 2. In Phase 2, the deployment is per typical deployment and the trained network is evaluated on unseen test data as depicted in Fig. 1.

IV. Evaluation

An experiment has been conducted to evaluate the performance of the clustering-supported boosted intensional pipeline as described in Sec III and demonstrate its effectiveness. The MNIST data set [9], which is widely used by researchers in the field for benchmarking purposes, was used for this experiment. The data set contains a set of grayscale images of handwritten digits 0 ... 9. Each image is of a low resolution of 28 pixels × 28 pixels. The complete MNIST data set comprises 60,000 training sample images and 10,000 test sample images. To illustrate what they look like, example images are presented in Fig. 4.

The machine learning task amounts to classifying each of the low-resolution grayscale images into any of the ten digits 0 … 9. It is therefore essentially a multiclass classification problem.

Evaluation is based on comparing a typical NN trained on the full set of training data with the boosted intensional pipeline trained on a significantly reduced set of training examples. The purpose is to demonstrate that the boosted intensional approach can achieve comparable test results on a much smaller training data set compared with more conventional extensional learning with large labelled data sets.

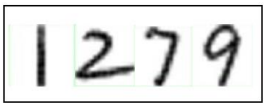

Fig. 4. Example images drawn from the MNIST data set [8]. Each grayscale image has a low resolution of 28 × 28 pixels.

A. Procedure

The evaluation procedure is as follows. First, a baseline is established by going through a standard NN training and deployment workflow as shown in Fig. 1. Second, the clustering-supported boosted intensional learning pipeline as described in Sec III is applied to train on the full set, followed by a small subset of the training data, which is varied from half of the original set to a quarter. The purpose of varying

the amount of training data size reduction is to see how this gradual change affects the classification performance.

Finally, the results from these two approaches are compared to see if there is any significant difference in performance of the two approaches. Another aspect that is also of interest is the sensitivity of the gradual reduction of the size of training data set (a half vs. a quarter of the original set) to boosted intensional learning.

B. Results

The experimental results are tabulated in Table I, which compares standard, data-intensive, extensional learning with more data-efficient clustering-supported boosted intensional learning. These are denoted methods 1 and 2, respectively, in Table I. The metric is accuracy of the predicted class, i.e. one of ten classes that correspond to ten digits.

The results for standard extensional learning are achieved using a NN architecture with two fully connected hidden layer. For boosted intensional learning, one of the hidden layer is omitted. This arrangement is, in part, to demonstrate that a simpler NN is possibly applicable to intensional learning because of the deliberate injection of intensional information into the learning process.

TABLE I: SUMMARY OF RESULTS

Method	Accuracy given Size of Training Data Set		
	Full set	½ set	¼ set
1	0.978	0.966	0.950
2	0.977	0.975	0.972

In Table I, size of the training data set refers to the number of training examples used in training. Recall that the full set contains 60,000 grayscale images. For the subsets of ½ and ¼ of the full set, 30,000 and 15,000 examples, respectively, they were randomly drawn from the full set. The results refer to test classification accuracy once the respective NN is in Phase 2 deployment mode. The full test set of 10,000 examples was used in each case. Therefore, the results reflect true performances and therefore generalization abilities of the two methods.

C. Discussion

From Table I, it is evident that while both methods perform similarly in test accuracy when trained with the full set of training data. However, Method 1, which relies totally on extensional learning, shows noticeable deterioration in test performance with decreasing size of the training data set. On the other hand, Method 2, which injects intensional information in a boosted fashion, shows much less significant deterioration (more resilience) in test performance with decreasing size of the training data set. In other words, Method 2 is much less sensitive to reduction in size of training data set. This can be a very useful trait in many real-world applications where availability of high-quality annotated data is limited.

While both the standard extensional and boosted intensional methods gave a similar accuracy performance result with the full training set, there are significant differences to consider. First, NN, and especially DL NN, are good at learning salient features on their own. So, feature engineering is minimized. However, this form of convenience comes at a cost in terms of the amount of annotated training data necessary to achieve feature extraction and generalization necessary for good classification performance.

As a form of engineering tradeoff, the boosted intensional approach requires a moderate amount of feature engineering in the terms of deliberately injecting intensional information that is fine-tuned to provide salient features for efficient supervised training. The upshot is that it is possible to achieve similar levels of performance even when the training data set is reduced to ½ or even ¼ of the full set. This is quite remarkable compared to standard extensional learning.

In the case of a relatively simple classification task involving the MNIST data set, the choice is not particularly clear. One might argue that training on the full set of MNIST is not really much of an effort. However, in the many real-world scenarios where high-quality annotated data are hard to come by, a relatively moderate amount of feature engineering can be an effective countermeasure.

It should be noted that Method 2 (boosted intensional) uses a simpler fully connected network (with one hidden layer omitted) than Method 1 (extensional), while achieving results that are fundamentally no worse (if not better) than Method 1. This highlights the positive effect intensional learning has on the expressive power (and therefore complexity) of a model. In addition, while both methods used relatively simple fully connected networks, both would likely perform better using more computationally-intensive 2D convolutional networks for image data. The reason is that 2D convolutional networks are especially useful for finding local features in images, such as those in the MNIST data set.

With reduced model complexity comes possibility of improved interpretability. Thus, intensional learning offers a further advantage in that the model's decision can become more interpretable. For example, when predicting a class that corresponds to the digit "1", it is likely that it recognizes salient features, such as a prominent vertical line and not much else. With improved interpretability of the model's decision making, the chances of detecting adversarial attacks is improved through enhanced transparency in the entire process. This is an important consideration as security of AI systems is often under the spotlight.

V. CONCLUSION

This paper has presented a comparison between standard extensional training for neural networks with a boosted intensional learning method that relies on an unsupervised preprocessing stage. Specifically, K-means clustering is selected as an unsupervised learning method to provide useful intensional information. This reduces a model's reliance on extensional (labeled) data. This is often an important consideration because in many real-world applications, availability of high-quality annotated data can be limited.

Ultimately, it represents a tradeoff between convenience (of not having to explicitly perform feature engineering) with

deep learning neural networks vs. availability of high-quality labeled data. It is also about a tradeoff involving a learning model's expressive power. These tradeoffs empower ML designers to choose and balance how they want their learning models to behave, especially when availability of good annotated data is limited.

Intensional learning offers a further advantage over extensional learning in that decision making by a trained neural network model is more interpretable. With better interpretability and transparency, the risks of adversarial attacks can be reduced. Thus, the introduction of intensional information in learning can simultaneously mitigate three major shortcomings of standard extensional NN learning, namely the needs for large amounts of high-quality annotated data, relative low levels of interpretability, and vulnerability to adversarial attacks caused by a relative lack of interpretability.

In this research, intensional information was extracted from an unsupervised clustering preprocessing step. Domain ontologies [10]-[13] which contain rich intensional domain information, can be used to drive this approach. It can be considered as either an alternative to unsupervised clustering, or as a complement that works collaboratively with other unsupervised preprocessing techniques, such as clustering. In this regard, hierarchical clustering could also be useful because it can capture real-world relations that cannot be readily represented using a flat structure achievable using K-means clustering.

Another future research direction is to investigate the amount of intensional information to inject into the learning process, relative to the amount of extensional information used in the training process. The investigation will answer questions like "Is there an optimal equilibrium?" and "Is it possible to overdo it?". Future research that aims to address these and related issues is currently underway, and the results will be reported in due course.

Conflict of Interest

The authors declare no conflict of interest.

Author Contributions

Both authors contributed equally to the research. A.C.M. Fong wrote the first draft of the paper; both authors approved the final version.

References

[1] S. Thrun, "Toward robotic cars," C*ommunications of the ACM*, vol. 53, no. 4, pp. 99–106.

[2] S. Daley. (2018). *AI Drones*. [Online]. Available: https://builtin.com/artificial-intelligence/drones-ai-companies

[3] Y. Goldberg, "A primer on neural network models for natural language processing," *Journal of Artificial Intelligence Research*, vol. 57, pp. 345–420, 2016.

[4] Scikit-Learn. [Online]. Available: https://scikit-learn.org/stable/

[5] Keras. [Online]. Available: https://keras.io/

[6] A. C. M. Fong, "Ontology-powered hybrid extensional-intensional learning," in *Proc. International Conference on Information Technology and Computer Communications (ITCC 2019)*, Singapore, August 2019.

[7] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," *Annals of Statistics*, pp. 1189-1232, 2001.

[8] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T. Y. Liu, "Lightgbm: A highly efficient gradient boosting decision tree," *Advances in Neural Information Processing Systems*, pp. 3146-3154, 2017.
MNIST data set. [Online]. Available: https://www.nist.gov/itl/iad/image-group/emnist-dataset

[9] Q. T. Tho, S. C. Hui, A. C. M. Fong, and T. H. Cao, "Automatic fuzzy ontology generation for semantic web," *IEEE Transactions on Knowledge and Data Engineering*, vol. 18, no. 6, pp. 842-856, 2006.

[10] Q. T. Tho, S. C. Hui, and A. C. M. Fong, "Automatic fuzzy ontology generation for semantic help-desk support," *IEEE Transactions on Industrial Informatics*, vol. 2, no. 3, pp. 155-164, 2006.

[11] A. C. M. Fong, B. Zhou, S. C. Hui, J. Tang, and G. Hong, "Generation of personalized ontology based on consumer emotion and behavior analysis," *IEEE Transactions on Affective Computing*, vol. 3, no. 2, pp. 152-164, 2012.

[12] S. M. Huynh, D. Parry, A. C. M. Fong, and J. Tang, "Novel RFID and ontology based home localization system for misplaced objects," *IEEE Transactions on Consumer Electronics*, vol. 60, no. 3, pp. 402-410, 2014.

A. C. M. Fong holds four degrees in electrical engineering, information systems, and computer science, from three universities: Imperial College London, University of Oxford, and University of Auckland.

He began his professional career as a research and development engineer with Motorola. Subsequently, he has held full-time faculty positions at Nanyang Technological University, Auckland University of Technology, University of Glasgow, and now Western Michigan University, Kalamazoo, MI, USA. He has published two books, 13 book sections, and circa 200 papers in international journals and conferences, including leading journals, e.g. IEEE T-KDE, IEEE T-MM, IEEE T-II. His current research interests include data analysis and applications of artificial intelligence.

Dr. Fong is a fellow of IET, senior member of IEEE, and is an engineer registered in the UK (CEng) and Europe (Eur Ing).