Int. J. Nonlinear Anal. Appl.
Volume 12, Special Issue, Winter and Spring 2021, 1511-1517
ISSN: 2008-6822 (electronic)
http://dx.doi.org/10.22075/ijnaa.2021.5803

# Building the new class of artificial intelligence using nonlinear model ABZU

S Shanthini[a], A S Aarthi Mai[a,]

[a]*Department of Computer Applications PSG college of Technology Coimbatore, India*

*(Communicated by Madjid Eshaghi Gordji)*

**Abstract**

ABZU is an explainable non-linear model, that has reimagined artificial intelligence to completely change the way problems are solved. It has a new standard of interpretability that has simple visual depictions and mathematical expressions for models developed which yields high accuracy. From this, a model developed can be highly accurate and an algorithm is recognized by all. This makes more complicated predictions at an easier level and explores new features of the model. This can be achieved through leveraging the results in the form of graphs and representing them. This algorithm applied in the fields of Artificial Intelligence / Machine Learning will yield an accurate result. This improves efficiency and increases the model's reusability.

*Keywords:* Machine learning algorithm, Data science

## 1. Introduction

"Google's self-driving cars and robots get a lot of press, but the company's real future is in machine learning, the technology that enables computers to get smarter and more personal – Eric Schmidt". Machine learning algorithms are programs that can learn from data and improve from experience, without human intervention but created by humans.

Machine Learning algorithm is an evolution of the regular algorithm. It makes your programs "smarter", by allowing them to automatically learn from the data provided.

In the field of Artificial intelligence changes are only real and it gets better at each stage of development. In that case, an algorithm plays a vital role where it is considered as the heart of the model developed. It should be designed in such a way that it predicts with at most accuracy.

The primary objective of our analysis is to make an algorithm more superior and never inferior to the exciting algorithms from supervised and unsupervised learning.

*Email addresses:* shannuswami1598@gmail.com (S Shanthini), asa.mca@psgtech.ac.in ( A S Aarthi Mai )

*Received:* August 2021 *Accepted:* November 2021

## 2. Related work

An algorithm in general work for training the model and used for testing the model's accuracy. Generally, AI algorithms are classified as supervised learning and unsupervised learning, they include algorithms from Decision tree models like K-means, K-Nearest Neighbors etc., and deep learning models like Long Short-Term Memory, Convolution Neural Network etc. Supervised algorithms are descriptive but not highly accurate, whereas in case of unsupervised models they are high in accuracy but not explainable. In Supervised learning models are trained using labeled data. And these models need to find the mapping function to map the input variable ($X$) with the output variable ($Y$).

$$Y = f(X) \tag{1}$$

Supervised learning needs supervision to train the model, which is like as a student learns things in the presence of a teacher. Supervised learning can be used for two types of problems: Classification and Regression.

Unsupervised learning is another classification of algorithms in machine learning. Here patterns inferred from the unlabeled input data. The goal of unsupervised learning is to find the structure and pattern from the input data. Unsupervised learning does not need any supervision. Instead, it finds the pattern from the data by its own.

In both of the cases an algorithm takes tree based structure for classification or pattern based training for classifying data, which makes prediction in such a way that supervised learning need to be taught from the basic, from the input given and output essentials. In unsupervised learning the structure of the dataset is described so the pattern is easily identified and from that model is designed and through training it remembers the pattern and match it during the time of prediction and testing phase

## 3. Proposed work

In General, the traditional machine learning algorithms have their own advantages as well as disadvantages. To overcome the disadvantages existing in supervised and unsupervised learning, ABZU comes up with the idea of explainable non-linear model.

Abzu has reimagined artificial intelligence to completely change the way problems are solved. Introducing traceable and shared insights and a new standard of interpretability. As explainability and interpretability are crucial for Data Scientists and decision-makers to manage, trust, and understand the decisions developed by AI, especially if decisions are critical or have a social impact. At Abzu [1], we put the scientific method at the core of our workflow, and use this approach to understand data, generate hypotheses, and validate the findings against the actual observations. The hypotheses are created using QLattice [2]. This graph is neither a neural network nor a decision tree-based model. It unpacks the black box neural network and serves explainability/interpretability similar to that of a decision tree.

## 4. Implementation overview

ABZU takes in the model in the form of graphs which helps to interpret in an accurate way and give explainable model for easy interpretation. It is achieved through QLattice technology through feyn library function in python. It fits an entirely new type of solution to the problem statement. The final models were the result of a process with random initial conditions, whose evolution was shaped by the data itself.

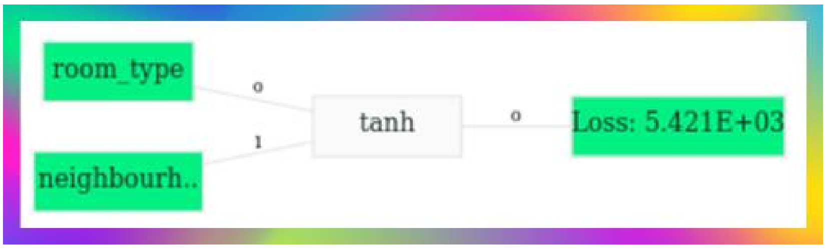


Figure 1: A simple graph of QLattice

A model should fit to the data and work upon the new data for prediction. Here the data is predicted in the form of graphs. It involves with the usual steps of data preprocessing which includes cleaning of data, removing unwanted chunks etc., then followed by building up the model, training the model, testing the model and finding its efficiency all these steps take place in QLattice does this in a different and in an easier way. The data transformations accessible in the QLattice are multiply, linear, sine, tanh and gaussian — together these covers virtually all-natural occurring dependencies. This means that a well-trained QLattice graph will extract whatever signal features when predicting the target variable. It is the model that has a simple visual depiction that let to inspect in detail, how the data is manipulated to ultimately deliver a set of predictions.

Working on this model takes in a sample dataset to test its accuracy that is so easy to understand about the efficiency of the algorithm. The input is given and in context to that data it is cleaned and followed by designing the model and then fitting it to the dataset used.

The QLattice searches among thousands of potential models for the one graph with the right set of features and interaction combinations that, in conjunction, unfolds the perfectly tweaked model to the problem. The real beauty of it all is not the high accuracy of the model. It is that way it has a simple visual depiction of the model — It can inspect in detail how data is manipulated to ultimately deliver a set of predictions. The basic workflow of QLattice how it functions to make such prediction through graphs.

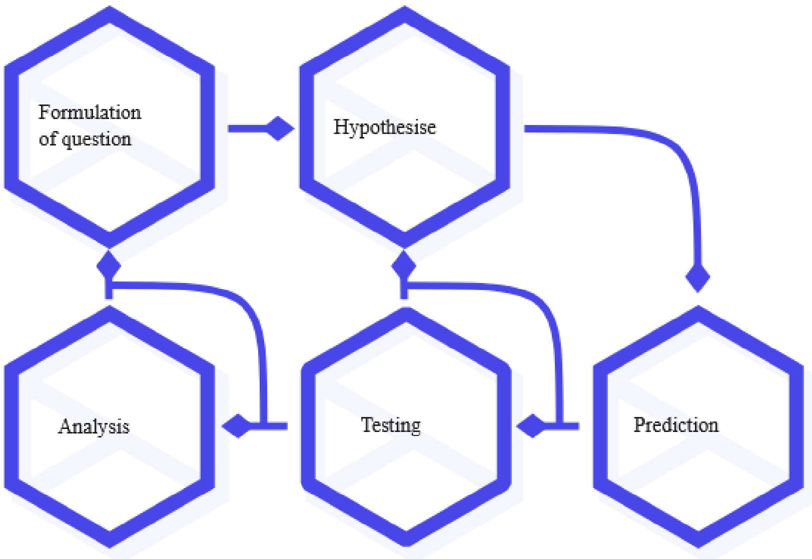


Figure 2: Basic Workflow

*A. Formulation of question*

Formulation deals with the problem statement defined for the model, designed and trained based on this QLattice works on fitting the dataset used for the problem and trains the model.

*B. Hypothesis*

QLattice generates the possible hypotheses to give different levels of prediction at different stages helpful for comparison of various obtained results.

*C. Prediction*

This determines how graph predicts from the existing data and plot what the graph tells about the model that haven't seen before. This helps for predicting the future of the model, with all the possibilities of matching it with the pattern in the trained model.

*D. Testing*

The developed model needs to be checked for its quality as well as for the efficiency of the algorithm, in that way testing phase of model uses various testcases.

*E. Analysis*

Accept or reject a hypothesis based on the graph being selected and the results that is obtained from the experiments. At this final stage, the predicted model is analyzed, and its result is used for the purpose of decision making.

***Working:***

Feyn (/faIn/) is the software development kit that we use to interact with the QLattice. It is named after Richard Feynman. Workflow typically starts after data preparation, however, it is worth mentioning that with the QLattice, don't need to do any normalization of input features, and it has an input that explicitly handles categorical variables without the need for one-hot encoding.

To tell about the QLattice the input and output variable target variable should be selected and assigned to the semantic datatype of the dataset. With this the categorical variables will be encoded automatically for QLattice. This is by design - the QLattice explores potential relationships between features and tries to come up with reduced graphs with a high degree of signal towards your question, rather than try to squeeze every drop of signal out of your features and overfit.

There are two semantic types (or s- types for short) of inputs: numerical and categorical, to distinguish these two so the model knows how to understand the inputs. Numerical variables are continuous (height, weight, age etc.). Inputs are automatically scaled using a linear transformation. Categorical variables are discrete (nationality, hair colour etc.). Inputs are automatically encoded with weights.

***Numerical type:***

Normalization or standardization is typically a required step for many machine learning algorithms. In Feyn, the numerical input type automatically takes care of normalization. It does this by setting a scale based on your minimum and maximum values, and by learning a weight and bias to your input values that will transform it into a usable range.

***Categorical type:***

This is a game changer for easily fitting in and learning from datasets that have categorical features with high cardinality.

***QGraph:***

The main part of QLattice is to produce QGraphs [3], which is a collection of all possible models connecting input to output. Once this is done, it starts working with the model training. Each model has two types of boxes. The green ones are either input or output and the pink ones are what we call interactions. An interaction takes in a value, evaluates a function at that value, then moves it to the next interaction. These models are sort of like a neural network but with fewer nodes and not the typical type of activation function on the node. Each model has a natural flow from left to right, so we feed each row of our dataset into the input, evaluate at each pink box and then produce a prediction at the end at the output. Like a neural network or any other machine learning technique, each of these models needs to be trained. The QGraph fit method needs to take the following arguments: The data the models in the QGraph should be trained on, and the loss function we want to optimize for.

the QGraph is the representation of the infinite ordered list of graphs (or paths) conceivable through the QLattice from your input features to your output feature, considering all possible combinations of interactions. Aside from controlling the max_depth of the graphs in the QGraph, one can set other conditions to these models by using the QGraph.filter function. In other words, the filter ensures that the only graphs being trained are the ones that satisfy the condition(s) imposed by the filter. It should be noted that when the filter function is called it does not modify the original QGraph, rather it returns a new QGraph object.

A graph consists of one or more of your input features, some interactions between them and a variety of functions that have been fitted to your dataset, leading to an output. In an IPython environment, you'll be able to hover over each of the interactions to get a tooltip with the internal state of it - such as the weights, biases, and encodings. The graph depiction shows you your best fitted model given your critria for complexity. The graph can be directly translated to a mathematical formula and you are good to go from there. Feyn offers a range of tools to help you dissect your graph and its dynamics. The QLattice is a generator of graphs from input to output. A QGraph is an unbounded list of graphs that have been generated from the QLattice. Fitting it is as simple as calling qgraph.fit(data). The QGraph discards the worst graphs and gives a new evolution of graphs based on what the QLattice has learnt through training.

The QLattice is a machine learning technology that allows to create hypothesis and gain a deeper understanding of the relations between features of the dataset, and how they interact. It takes an evolutionary approach through decisions to make when exploring the data.

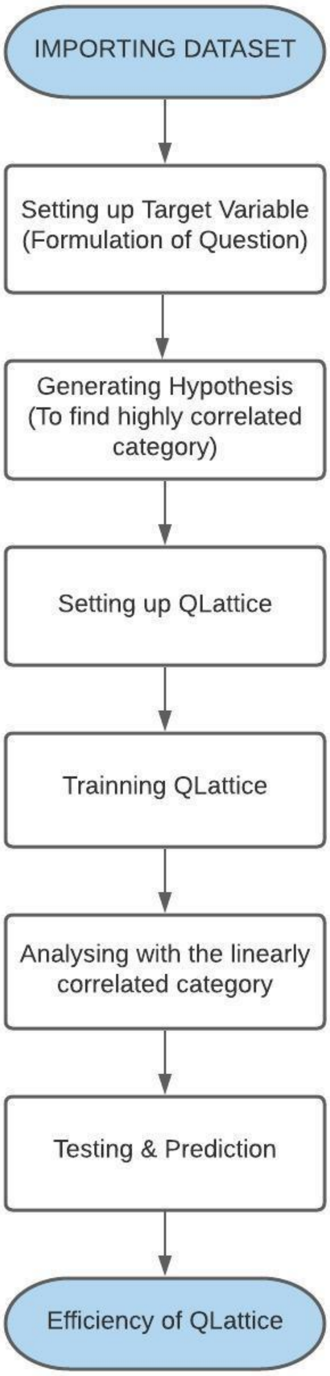


Figure 3: Working of QLattice

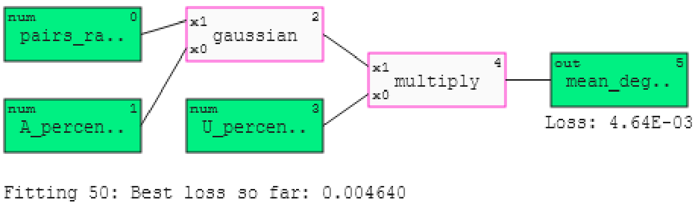


Figure 4: Training through QGraphs

## 5. Result

During fitting up of QLattice, value loss is reduced to the almost possible level. The values are passed to function that are similar to activation function in neural network like tanh, gaussian, sine etc. The accuracy of the model developed is approximately 89%. The graph learns through number of fittings it is been run. Depending on the loss rate the number of fittings can be increased. The Objective of ABZU is to give an algorithm with more efficiency and easy understandability. The prediction of result is also accurate based on the fitting and testing helps to identify the most accurate and highly correlated category of the dataset for prediction. This helps to minimize the misinterpretation of results and give high efficiency.

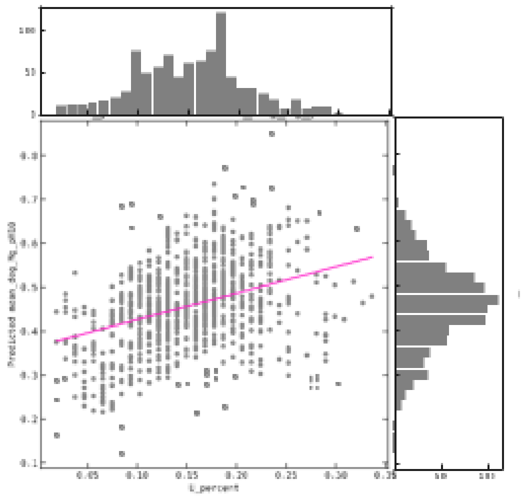

Figure 5: Result of QLattice

## 6. Conclusion

In this paper, an implementation of ABZU, through QLattice and Feyn is done. This is nothing but explainable nonlinear models. This type of models has both explainability and interpretability. It is neither a neural network nor a decision tree- based algorithm, just a combination of both for the higher level of prediction. Compared with traditional algorithms in supervised and unsupervised learning, ABZU gives more efficiency and algorithm is easy to understand as it is in the for of graphs. The accuracy of the system can be improved further by improving the number of fittings and training done on the graphs.

## References

[1] www.abzu.ai.

[2] https://medium.com/abzuai/the-qlattice-a-new-machine-learning-model-you-didnt-know-yo-needed-c2e037878cd.

[3] *Complete Guide To QLattice: A New And Transparent Machine Learning Framework*, (analyticsindiamag.com).