Module 2 - Critical Thinking Assignment: Truth Tables

Alejandro Ricciardi

Colorado State University Global

CSC410: Artificial Intelligence

Dr. Chris Whitfield

October 20, 2024

Module 2 - Critical Thinking Assignment: Truth Tables

Truth Tables (TTs) evaluate logical statements by systematically analyzing the relationship between the truth and falsehood between propositions within those statements. This essay demonstrates the use of TTs by providing two examples using three propositions and analyzing their logical relationships. It also briefly explores how TT can be used to implement Truth Table networks (TT-net), a Convolutional Neural Network (CNN) model that can be expressed in terms of TTs, and when they are combined with Deep Neural Networks (DNN) they can create a novel Neural Network Framework (NNF) called Truth Table rules (TT-rules).

Definition

A True Table evaluates all possible truth values returned by a logical expression (Sheldon). The return truth values are binary, meaning that either true or false (not true), they may be referred to as Boolean values. Boolean is a term that represents a system of algebraic notation used to represent logical propositions, usually by means of the binary digits 0 (false) and 1 (true) (Oxford Dictionary, 2005). In Boolean algebra and related mathematics fields, as well as in sciences rely on Boolean logic to show the possible outcomes of a logical expression or operation in terms of its truth or falseness that can be expressed using numbers, characters, or words. In programming languages such as C++ and C any non-zero return Boolean value is considered true; however, in the Java programming language the value can only be of the data type 'true' or 'false'. On the other hand, TTs usually use the letters 'T' for true and 'F' for false to represent truth values.

Propositional Logic

As mentioned earlier, Boolean values are used to represent logical propositions, a logical proposition also known as an atomic sentence is a sentence that can either be true or false, but not both (James, 2014). Propositional logic (also known as sentential logic or Boolean logic) is the process of forming logical statements by combining logical propositions, also known as complex sentences (Russell & Norvig, 2021). An atomic sentence is represented by a single proposition symbol, such as P, Q, R, or W_{13} that can be allocated with a true or false Boolean value. For example, $P = T$, P is true, or $P = F$, P is false, but never both. To combine the atomic sentences into a logical statement operators like AND ('$\land$'), OR ('$\lor$'), and NOT ('$\neg$') are used, as well as symbols to express implications, symbols such as '$\Rightarrow$' for '*implies*' and '$\Leftrightarrow$' for '*if and only if*'. The table below lists the five basic logical operations forming complex sentences using the operators and symbols that were just discussed.

Table 1

Basic Logical Operations

Operation	Notation	Summary of Truth Values
Negation	$\neg P$	The opposite truth value of P. True when P is false, false when P is true.
Conjunction	$P \land Q$	True only when both P and Q are true. Otherwise, it is false.
Disjunction	$P \lor Q$	False only when both P and Q are false. Otherwise, it is true.
Conditional	$P \Rightarrow Q$	False only when P is true and Q is false. Otherwise, it is true.
Biconditional	$P \Leftrightarrow Q$	True when both P and Q are either true or both false. Otherwise, it is false.

Note: From "2.2: Introduction to truth tables. Mat 1130 mathematical ideas," by Lippman (2022), modify.

Complex sentences can combine more than one operation. For example, $(W_{13} \land P_{31}) \lor W_{22} \Leftrightarrow \neg W_{21}$. Additionally, the operator follows precedence similar to the precedence arithmetic operators, it is as follows '¬', '∧', '∨', '⇒', '⇔', with '¬' having the most precedence (Russell & Norvig, 2021). Additionally, two atomic sentences P and Q are logically equivalent if they are true in the same set of models using the following notation $P \equiv Q$. A model is a specific assignment of truth values to all the atomic sentences in a logical expression. This equivalence also applies to complex sentences, and it has the following properties:

- $(P \land Q) \equiv (Q \land P)$ — commutativity of ∧

- $(P \lor Q) \equiv (Q \lor P)$ — commutativity of ∨

- $((P \land Q) \land W) \equiv (P \land (Q \land W))$ — associativity of ∧

- $((P \lor Q) \lor W) \equiv (P \lor (Q \lor W))$ — associativity of ∨

- $\neg(\neg P) \equiv P$ — double-negation elimination

- $(P \Rightarrow Q) \equiv (\neg Q \Rightarrow \neg P)$ — contraposition

- $(P \Rightarrow Q) \equiv (\neg P \lor Q)$ — implication elimination

- $(P \Leftrightarrow Q) \equiv ((P \Rightarrow Q) \land (Q \Rightarrow P))$ — biconditional elimination

- $\neg(P \land Q) \equiv (\neg P \lor \neg Q)$ — De Morgan

- $\neg(P \lor Q) \equiv (\neg P \land \neg Q)$ — De Morgan

- $(P \land (Q \lor W)) \equiv ((P \land Q) \lor (P \land W))$ — distributivity of ∧ over ∨

- $(P \lor (Q \land W)) \equiv ((P \lor Q) \land (P \lor W))$ — distributivity of ∨ over ∧

(Russell & Norvig, 2021, p.222)

Examples of Truth Tables

This section explores two sentences' logic: one involving a conditional statement with a negation and conjunction, and the other involving a biconditional statement with disjunction. The sentences in natural language are:

1- If it is sunny and I do not work today, then I will go to the beach.

2- I will pass the exam if and only if I complete all homework assignments or I study for at least 10 hours.

The first step is to identify the atomic sentences, that are part of the natural language sentences, followed by the complex sentences and the logical operator that combines them, then the last step is to form the table based on the atomic sentence, complex sentences, and logic operators. Note that atomic sentences also called atomic propositions are simple propositions that have no logical content (Lavin, n.d.). Thus, their logical value can be set as false or true to evaluate more complex propositions also called complex sentences.

Example 1

Let's start with the first sentence "If it is sunny and I do not work today, then I will go to the beach." The atomic sentences are:

o   *P:* "The weather is sunny."

o   *Q:* "I work today."

o   *R:* "I will go to the beach."

The logic operators are:

o   '¬' (not).

o   '∧' (and).

o   '⇒' (Conditional). Note that '⇒' corresponds to the terms 'implies' or "if then'.

The complex sentences are:

- $\neg Q$

- $P \land \neg Q$

- $(P \land \neg Q) \Rightarrow R$

Now, let's make the TT:

Table 2

Sentence 1 Truth Table

P	Q	R	¬Q	P ∧ ¬Q	(P ∧ ¬Q) ⇒ R
T	T	T	F	F	T
T	T	F	F	F	T
T	F	T	T	T	T
T	F	F	T	T	F
F	T	T	F	F	T
F	T	F	F	F	T
F	F	T	T	F	T
F	F	F	T	F	T

The TT is a world model that explores and evaluates all the possible truth values of the atomic and complex sentences. However, not all the table values bear relevance in proving the logical validity of sentence 1. In other words, irrelevant propositions can be ignored, no matter how many of them there are (Russell & Norvig, 2021). For example, if *R* is false, "I am not going to the beach" regardless of whether *P* and *Q* are true or false, making the proposition *R* is false irrelevant in determining the validity of sentence 1. Additionally, to prove the logical validity of sentence 1 both *(P ∧ ¬Q)* and *R* propositions need to be true and both *P* and *¬Q* propositions need to be also true. This concept is similar to coding an 'if' statement in a programming language, where two conditions combined with the logical 'and' operator must both be true for the code after the 'then' clause to execute; for instance, 'if (A && B) then print("A and B are

both true");'. Note that the proposition "print("A and B are both true");" is always true if (A and B) is true. Thus, the relevant propositions for this example are found in row three of the table:

Table 3

Sentence 1 Truth Table Row 3

P	Q	R	¬Q	P ∧ ¬Q	(P ∧ ¬Q) ⇒ R
T	F	T	T	T	T

- *P: T* — "The weather is sunny" is true.

- *Q: F* — "I work today" is false.

- *¬Q: T* — "I will go to the beach" is true.

- *R: T* — "I do not work today" is true.

- *P ∧ ¬Q: F* — "It is sunny and I do not work" is true.

- *(P ∧ ¬Q) ⇒ R: T* — "If it is sunny and I do not work today, then I will go to the beach" is true.

Therefore, the sentence "If it is sunny and I do not work today, then I will go to the beach" is logically sound.

Example 2

Now, let's explore the sentence "I will pass the exam if and only if I complete all homework assignments or I study for at least 10 hours." The atomic sentences are:

- *P:* "I complete all homework assignments."

- *Q:* "I study for at least 10 hours."

- *R:* "I will pass the exam."

The logic operators are:

- '∨' (or)

o   '⟺'. Note that '⟺' corresponds to the term "if and only if".

The complex sentences are:

o   *P ∨ Q*

o   *(P ∨ Q) ⟹ R*

Now, let's make the TT:

Table 4

Sentence 2 Truth Table

P	Q	R	P ∨ Q	R ⟺ (P ∨ Q)
T	T	T	T	T
T	T	F	T	F
T	F	T	T	T
T	F	F	T	F
F	T	T	T	T
F	T	F	T	F
F	F	T	F	F
F	F	F	F	T

As in example 1, the TT is a world model that explores and evaluates all the possible truth values of atomic and complex sentences. However, not all the table values bear relevance in proving the validity of sentence 2. Both *(P V Q)* and *R* expressions need to be true to prove that the sentence is logically valid. Additionally, only one of the atomic sentences in the proposition *(P V Q)* for the proposition to be true. Thus, the relevant propositions for this example are found in rows one, three, and five of the table:

Table 5

Sentence 2 Truth Table Row 1, 2, and 3

Row	P	Q	R	P ∨ Q	R ⇔ (P ∨ Q)
1	T	T	T	T	T
3	T	F	T	T	T
5	F	T	T	T	T

- ○ *P: T* — "I complete all homework assignments" is true.

- ○ *P: F* — "I complete all homework assignments" is false.

- ○ *Q: T* — "I study for at least 10 hours" is true.

- ○ *Q: F* — "I study for at least 10 hours" is False.

- ○ *R: T* — "I will pass the exam" is true.

- ○ *P ∨ Q: T* — "I complete all homework assignments (false) or I study for at least 10 hours (true)" is true.

- ○ *P ∨ Q: T* — "I complete all homework assignments (true) or I study for at least 10 hours (false)" is true.

- ○ *(P ∧ ¬Q) ⇒ R: T* — "I will pass the exam if and only if I complete all homework assignments or I study for at least 10 hours" is true.

Therefore, the sentence "I will pass the exam if and only if I complete all homework assignments or I study for at least 10 hours" is logically valid.

Applications

TTs have many applications in mathematics and science. A recent proposed application by Benamira et al. (2023b) suggests using them within Convolutional Neural Networks (CNNs) to create a novel CNN architecture called Truth Table net (TT-net). In traditional CNNs, researchers do not have a clear insight into how the network makes decisions, making CNNs

"black boxes." TT-net architecture will make it easier for researchers to understand and interpret how the CNN makes decisions. After training, TT-net can be analyzed and understood using Boolean decision trees, Disjunctive/Conjunctive Normal Form (DNF/CNF), or Boolean logic circuits. This will allow the researchers to map the decision-making process of the CNN. A similar proposed application of TTs by Benamira et al. (2023a) suggests using them as a framework called Truth Table rules (TT-rules) within Deep Neural Networks (DNNs). TT-rules is based on the TT-net architecture with the goal of making DNNs less of a "black box" and more interpretable by transforming the DNN-trained models into understandable rule-based systems using TTs.

Summary

Truth Tables are a useful tool to prove the logical validity of sentences using Boolean values. They are world models that explore and evaluate all the possible truth values of atomic and complex sentences. In other words, they help evaluate logical propositions by breaking down complex sentences into atomic sentences and analyzing all possible combinations of truth values within the proposition, as shown in examples 1 and 2. They have many applications in mathematics and science; proposed applications in CNNs and DDNs would involve using TTs to make the models less of a "black box" by making their decision-making processes more transparent, and interpretable.

References

Benamira, A., Guérand, T., Peyrin, T., & Soegeng, H., (2023a, September 18). *Neural network-based rule models with truth tables*. arXiv. http://arxiv.org/abs/2309.09638

Benamira, A., Guérand, T., Peyrin, T., Yap, T., & Hooi, B. (2023b, February 2). *A scalable, interpretable, verifiable & differentiable logic gate convolutional neural network architecture from truth tables*. arXiv. http://arxiv.org/abs/2208.08609

James, J. (2014). *Math 310: Logic and Truth Tables* [PDF]. Minnesota State University Moorhead. Mathematics Department. https://web.mnstate.edu/jamesju/Spr2014/Content/M310IntroLogic.pdf

Lavin, A. (n.d.). 7.2: Propositions and their Connectors. *Thinking well - A logic and critical thinking textbook 4e (Lavin)*. LibreTexts Humanities. https://human.libretexts.org/Bookshelves/Philosophy/Thinking_Well_-_A_Logic_And_Critical_Thinking_Textbook_4e_(Lavin)/07%3A_Propositional_Logic/7.02%3A_Propositions_and_their_Connectors#:~:text=Atomic%20propositions%20are%20sometimes%20longer,a%20simple%20or%20atomic%20proposition.

Lippman, D. (2022) 2.2: Introduction to Truth Tables. *Mat 1130 mathematical ideas*. Pierce College via The OpenTextBookStore. https://math.libretexts.org/Courses/Prince_Georges_Community_College/MAT_1130_Mathematical_Ideas_Mirtova_Jones_(PGCC:_Fall_2022)/02:_Logic/2.02:_Introduction_to_Truth_Tables

Oxford Dictionary (2006). *The Oxford dictionary of phrase and fable (2 ed.)*. Oxford University Press. DOI: 10.1093/acref/9780198609810.001.0001.

Russell, S. & Norvig, P. (2021). 7. Logical Agent. Artificial intelligence: *A modern approach. 4th edition*. Pearson Education, Inc. ISBN: 9780134610993; eISBN: 9780134671932.

Sheldon, R. (2022, December). *What is a truth table?* TechTarget.

https://www.techtarget.com/whatis/definition/truth-table