INTERNATIONAL JOURNAL OF NOVEL RESEARCH AND DEVELOPMENT (IJNRD)
© 2026 IJNRD | Volume 11, Issue 7, July 2026 | ISSN: 2456-4184 | IJNRD.ORG

LIGHT WEIGHT INFERENCE ENGINE is an EXPERT SYSTEM and AI ----- TENSOR Flow-----

S.MOHAN
ASSISTANT PROFESSOR OF COMPUTER SCIENCE AND ENGINEERING,
V.S.B. COLLEGE OF ENGINEERING TECHNICAL CAMPUS,
KINATHUKADAVU, COIMBATORE, TAMIL NADU ,INDIA-642 109.

ABSRACT:

A light weight engine is a highly optimized software components designed to execute trained artificial intelligence (AI) model efficiency or resource constrain devices such as mobile phone and IoT hardware which the term" INFERENCE ENGINE" historically organized with rule-based expert system to process logic loops modern engine like TENSOR Flow (now rebranded by google as Lite RT) process entirely on mathematical machines learning models google dedicated solution for light weight edge based deployment in to lite RT (formally tensor Flow lite) it process AI work load locally without depending on cloud server through a distinct two-step life cycle model are typically trained in the cloud using full-scale tensor Flow. The tensor Flow lite convert -compressor. these model in to the highly efficient .if lite format , which is the executed on the edge device using the light weight tensor Flow lite interpreter the performance of optimized to achieve a small binary size and fast execution an resource constraint devices light weight engine employee specific technology

KEY WORDS; AI, IoT, lite RT, cloudra,Ml;

INTRODUCTION:

A light weight inference engine an AI is a specialized , compact frame work designed to run pre-trained machine learning and deep learning directly an edge device (like mobile phone, IoT gadgets or micro-controller) it deliveries fast , low-latency and off-line prediction without relying an heavy cloud server . the tensor Flows is a permissive open-source AI frame work developed by google . which it is wildly used for building and training complex neural network. It also directly address the need for light weight inference engine through its dedicated mobile and edge eco-system to bring artificial intelligence directly to your device tensor Flow often tensor Flow lite unlike a massive server-based training engine . this light weight inference tool is optimized for minimal latency low managing footprint and operating without an internet communication it take your bulky trained models and optimized them in to a compressed format (using flat-buffer) so they execute quickly and constraint hardware.

INFERENCE EGINE AI and EXPERT SYSTEM:

An inference engine is the "BRAIN" of an expert system which is specific type of AI . It applied logical rule the database of human knowledge to deduced new information , solve the complex problems or make decision

An expert system is designed to simulate the decision-making skill of a human specialist (like a DOCTOR or MACHANICS) it relies on two main components

1.The knowledge base : an structure re positioning of fact expert derived rules about specific domain.
2.The inference engine : the reasoning mechanism that navigate the knowledge base

The reasoning process : the inference engine process your input data (like set of symptoms) through the pre-programmed rules using two methods

FORWARD CHAINING : Start with available fact and works step-by-step to arrive at a conclusion.

BACKWARD CHAINING : Start with possible hypothesis (eg the patient has too flu) and work backward to see it the available data support it.

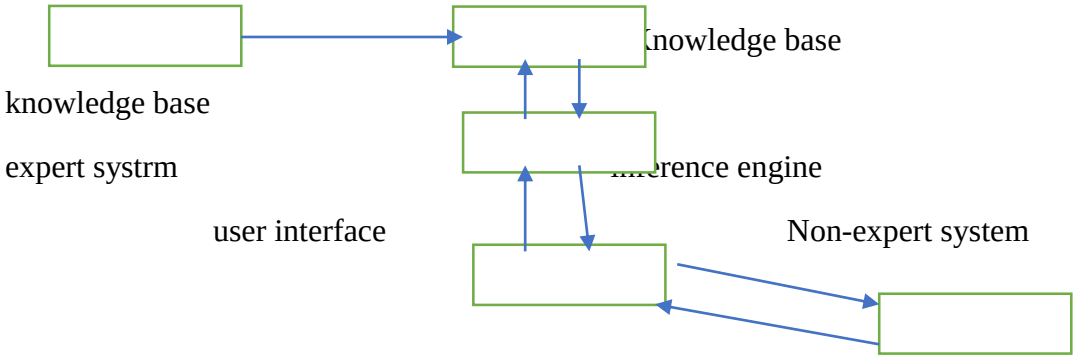
EXPERT SYSTEM vs OTHER AI:

Unlike modern machine learning model that learn pattern directly from massive data set , expert system and the inference engine rely entirely an explicit b, human provide rules . this make them highly predictable , transparent and capable of explaining exactly why they reacted a specific conclusion.

AI inference ENGINE:

In field of AI an inference engine is a software compact of an intelligence system that applied logical rules to the knowledge based to deduced new information the first inference engine were component of expert system.

EXPERT SYSTEM INFERENCE ENGINE :



The inference engine is the BRAIN of an expert system . it is a software components that apply logical rules to a knowledge base to deduced new information by processing user input against stored fact it simulate human reasoning to arrive at a diagnosis , recommendation or decisions.

INFERENCE ENGINE MEANS:

An inference engine is a core components of expert system designed to apply logical rules to knowledge base to drive new information an make decision it is the BRAIN of an expert system simulating the human ability to make inference inference engine work by using various algorithm to navigate through the knowledge base which contain domain – specific facts and rules and apply them rule to new fact to inference new fact an conclusion . these engine can operate in to different modes such as forward chaining ,when inference are made from given data to conclusion or backward chaining , where the engine start with symptoms and work backward to see if data support the hypothesis . this process allow expert system to provide explanation and justification system for each conclusion or decision made closely mimicking human expert problems solving capability .

MODERN MACHINE LEARNING :

In contemporary AI (such as LLM like chat GPT) or image generator the inference engine act as the run time environment .

THE PROCESS : it local a pre-trained neural network model (a massive directed acyclic component graph) and execute mathematically operate to (forward pass) against incurring data.

FUNCTION :

Instead of using hard – coded human logic it utilize the pattern it learned doing training to generate prediction text or image.

OPTIMIZATION :

Because generating requires internship computation . modern inference engine focus heavily an optimized hardware (such as GPU) to reduce) latency and cost.

Whether the rule based or model based intelligence system typically consists of them primary layers.

1.The knowledge based/model:
This static repository of fact , rules or learned neural network weight.
2.The working process :
Temporary data storage for the commit problem or an giving instruction
3.The inference engine :
The run time machine that instruct with the memory and knowledge base to produce an actionable answers.

LIGHT WEIGHT INFERENCE ENGINE :

A light weight inference engine is optimized low foot print software frame work designed to rule to trained machine learning or AI model efficiency an device with limited memory battery or processing power.

Because AI Training required massive computing power to “learn” the data the resulting model files are often very large and source heavy . once trained deploying the model in to the real-world requires and inference engine to process new input and generate the prediction . a light weight version strips out all the heavy training code optimized the model specifically for deployments.

SMALL BINARY SIZE inference engine:

A small binary size inference engine “refers to an AI run time whose coupled executable or library take-up very little storage space (eg a few KB to single MB) it is designed to run machine learning (ML) and large language model (LLM) directly an device with serves memory and storage .

On device AI and MOBILE :

It allows develop to AI model directly in to mobile app or desk top solution without blocking the domain local size .

IoT and EDGE Computing :

It enables smart servers , micro controller and low-power device (which might only have a few KB of RAM) to run AI locally without needing an active internet connection.

SECURITY and PRIVACY :

Because the inference engine and model an self- contained they can be deployed in highly secure environments (like medical device or critical infrastructure)

The heavy inference engine (like standard Pytorch or tensor Flow)an massive software libraries (often hundred of MB) meant for large server or GPU . a small binary size enable several critical computing.

LIGHT WEIGHT INFERENCE ENGINE means :

It is an optimized software run time despises to run pre- trained AI model with minimum memory – processing and hardware requirements it strips away the heavy , complex infra structure needed to train AI focusing strictly an making prediction and generating responses as fast as possible because they are structured .light weight engine can operate effectively and standard laptop , mobile phone and internet connected device rather than requiring massive cloud servers.

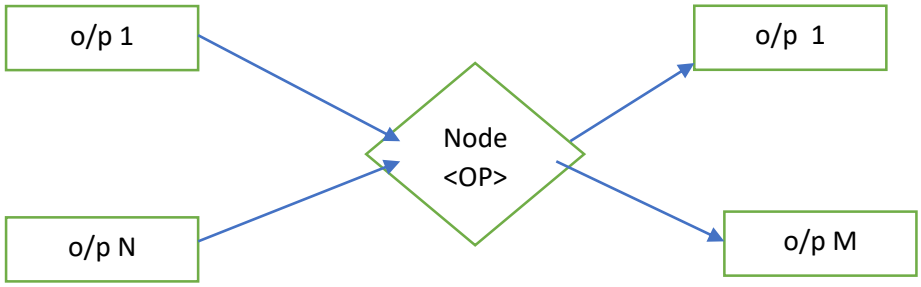
Examples :

Llama.cpp : a widely used open-source run time that allows you to run large language model and consumer – grade hardware (like laptop) using standard CPU .

ONNX run time ;

An cross – platforms engine designed to run the model efficiently across various hardware types.

TENSOR FLOW lite :

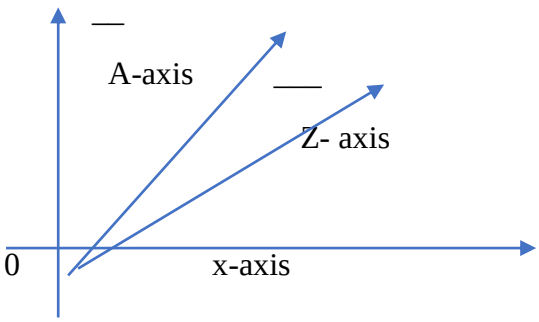


Specifically engineer for mobile and embedded device

TENSOR Flow :

The tensor Flow means is an open-source deep learning library developed by ¯deep learning application . these are several comprehensive resources available to help beginners and experience developer learn.

TENSOR Flow = TENSOR + Flow ;



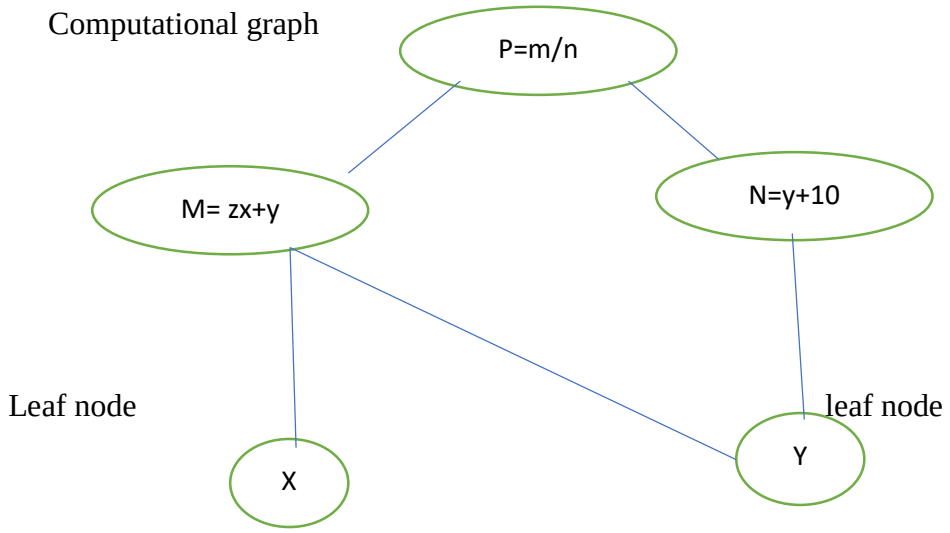
The tensor are building block of tensor Flow as all computation are done using tensor . so what exactly is an tensor . the defends Is a tensor is generating of vector and matrices to potential higher dimension internally tensor flow represents reason as n- dimensional analysis of tensor data types.

Basis of vector are associated with the connect system and can be used to represent any vector . this basis vector have determined by he represent any vector . these basis vector have a length of and hence are also know as unity vector ,these direct of these basis vector is determine by the respective co-ordinate.

For examples :

3D representation are have 3 basis vectors (X,Y,Z) so X-axis would have the direct of the X axis co-ordinate and the Y basis vector would have the direct of Y . similarly this would be the axis for Z.

FLOWS :



New comes the several fact of tensor Flow is basically underling graph cooperation frame work that uses tensor for its executes . atypical graph connects of two entities or nodes and edge nodes are also called vectors.

the edge an essentially the connection between node/ vectors through which the data flow . the node are where active computation take place . now in several the graph can be cyclic or acyclic . but in tensor Flow . it is always acyclic . it can not be start and end at the same node . lets consider a simples computational graph and explore save it attribute . these are multi way is cohesive an use tensor Flow (local as well as clouds) .

1.Anaconda
2.colab
3.Data bricks

ANACONDA :

This is graph way of using tensor Flow an a local system . we can pip installed the latest version of tensor Flow [IN]; pip installed -q tensor Flow == 2.0.0-beta 1.

COLAB :

The most council way to use tensor Flow provide by google tensor Flow is colab start the Collaboratory this represent the idea of collaborator and online laboratory it is true JUPYTER based web environment

requiring on setup as it came with all the dependency pre built without leaving worry to any sort of installation and pre dependence.

DATA BRICKS :

Another way to use tensor Flow is through the data bricks platforms . the method of installing tensor Flow an data bricks is showing following using a communicating edition account , but the source procedure can be adapted for business accept usage . we explain the fundamental deference between vector and tensor . we also covered the major different between previous and communicate version of tensor Flow . finally we want over the process of installing tensor Flow locally as in cloud environments (with data bricks).

INFERENCE ENGINE IS TRADITIONAL expert system:

These system use determine " forward and back ward chaining to deduced new information "<QUATE> .. working backward to determine the fact need to achieve it <QUATE>" as detailed by Zen van riel . the human mimic have logic based as pre defined " rule and fact to deduce new knowledge <QUATE> reach conclusion or often suggestion </QUETE>" noted by OER common>

NETWORK INFERENCE ENGINE :

the tensor Flow rules trained machine learning mode to map real-world input to output "<QUATE> to generate prediction</QUATE>.. <QUATE> it takes compact data, process it through the model and output result </QUATE>"according to the cloudra.

TENSOR Flow lite :(lite RT):

For light weight need model are converted in to compact . if lite front and locate via the lite RT engine , which optimized " ON-DEVICE inference google eco-system "<QUATE> … is no larger a previous task ; it is the universal engine for ON-DEVICE inference <QUATE>" as represented by dev.to.

CONCLUSION:

A light weight u=inference engine applies trained model to unseen data on low -resources hardware . in AI expert system traditional " inference engine use logical rules to knowledge base <QUATE> …to drive conclusion , make decision or solve problems … </QUATE> as outlined by the AI navigator .the tensor Flow serves this with lite Rt (run time) (formally TF lite) which execute model efficiently as edge device .with traditional expert system relying on explicitly programmed rules tensor Flow provide the light weight run time requirements to display deep learning models ,IoT and embedded device . this enable developer to make " real-time prediction – with moderate to local compute intensity <QUATE>. Apply learned pattern to new data </QUATE> as highlighted by CLOUDRA.

REFERENCES;

1.Manuele Rusci, Alessandro Capotondi, and Luca Benini, "Memory-driven mixed low precision quantization for enabling deep network inference on microcontrollers," Proceedings of Machine Learning and Systems, vol. 2, pp. 326 - 335, 2020.

2.Ji Lin, Wei-Ming Chen, Yujun Lin, Chuang Gan, Song Han, et al., "Mcunet: Tiny deep learning on iot devices," *Advances in Neural Information Processing Systems*, vol. 33, pp. 11711 - 11722, 2020.

3.Torben Krieger, Bernhard Klein, and Holger Fröning, "Towards hardware-specific automatic compression of neural networks," arXiv preprint arXiv:2212.07818, 2022.

4.Bichen Wu, Yanghan Wang, Peizhao Zhang, Yuandong Tian, Peter Vajda, and Kurt Keutzer, "Mixed precision quantization of convnets via differentiable neural architecture search," arXiv preprint arXiv:1812.00090, 2018.

5.Jungwook Choi, Zhuo Wang, Swagath Venkataramani, Pierce I-Jen Chuang, Vijayalakshmi Srinivasan, and Kailash Gopalakrishnan, "PACT: Parameterized clipping activation for quantized neural networks," 2018.

6.Kuan Wang, Zhijian Liu, Yujun Lin, Ji Lin, and Song Han, "Haq: Hardware-aware automated quantization with mixed precision," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 8612 - 8620.

7.Steven K Esser, Jeffrey L McKinstry, Deepika Bablani, Rathinakumar Appuswamy, and Dharmendra S Modha, "Learned step size quantization," arXiv preprint arXiv:1902.08153, 2019.

8.Alessandro Capotondi, Manuele Rusci, Marco Fariselli, and Luca Benini, "Cmix-nn: Mixed low-precision cnn library for memory-constrained edge devices," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 67, no. 5, pp. 871 - 875, 2020.

9.Liangzhen Lai, Naveen Suda, and Vikas Chandra, "Cmsis-nn: Efficient neural network kernels for arm cortex-m cpus," arXiv preprint arXiv:1801.06601, 2018.

10. Tianqi Chen, Thierry Moreau, Ziheng Jiang, Lianmin Zheng, Eddie Yan, Haichen Shen, Meghan Cowan, Leyuan Wang, Yuwei Hu, Luis Ceze, et al., "TVM: An automated End-to-End optimizing compiler for deep learning," in 13th USENIX Symposium on Operating Systems Design and Implementation (OSDI 18), 2018, pp. 578 - 594.

11. Meghan Cowan, Thierry Moreau, Tianqi Chen, James Bornholt, and Luis Ceze, "Automatic generation of high-performance quantized machine learning kernels," in *Proceedings of the 18th ACM/IEEE International Symposium on Code Generation and Optimization*, 2020, pp. 305 - 316.

12. Yaman Umuroglu and Magnus Jahre, "Towards efficient quantized neural network inference on mobile devices: work-in-progress," in Proceedings of the 2017 International Conference on Compilers, Architectures and Synthesis for Embedded Systems Companion, 2017, pp.