Reversing Multimodal Deep Learning AI Model

White Paper
(Draft Version)

Author : Yashodhan Vivek Mandke

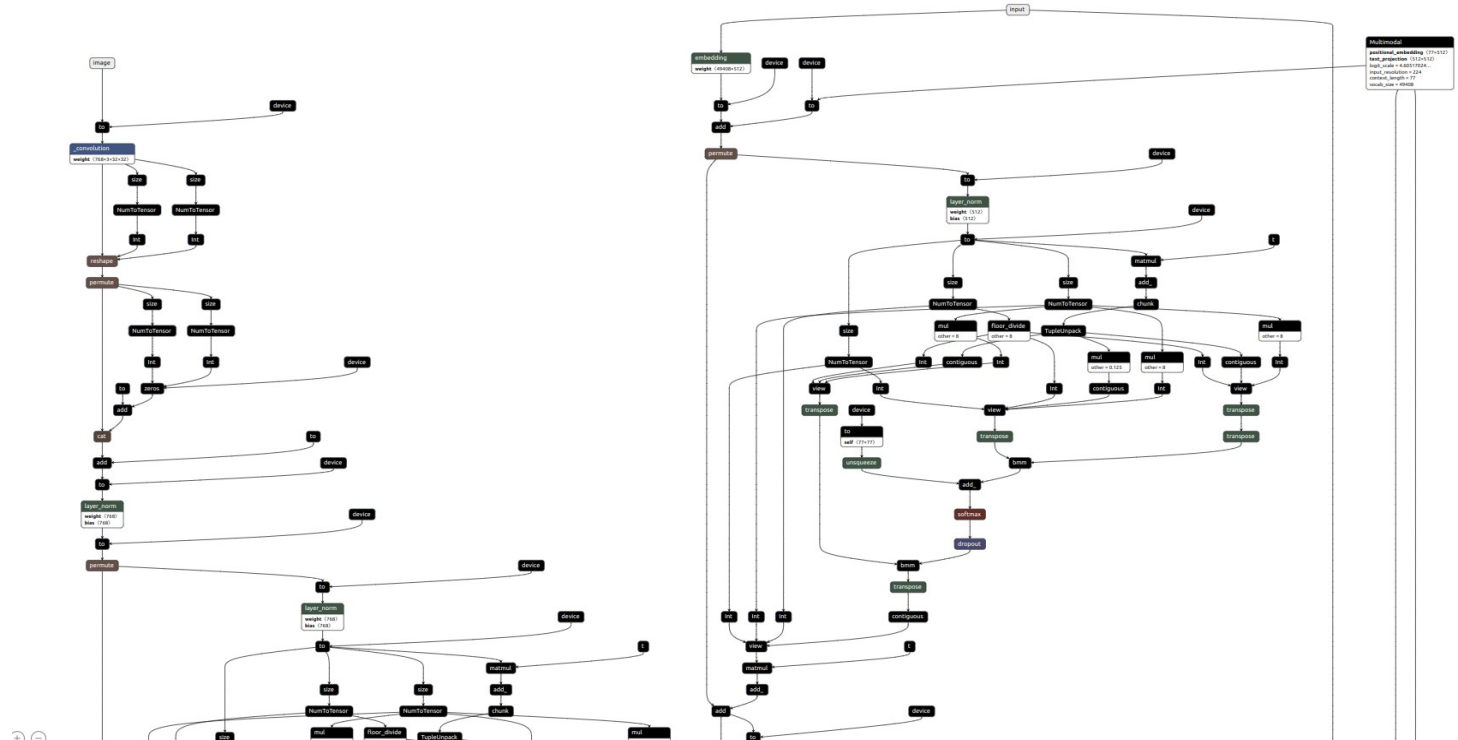
Introduction

The multimodal AI comprehends image, audio, and text inputs, and engages in seamless, end-to-end voice conversations with users. It provides real-time voice output, has comprehensive multimodal understanding, and offers flexible interactions, including the ability to handle interruptions during speech. Reversing these multimodal architecture involves reversing multiple models such as vision, language, audio etc. The model architecture once revealed to an attacker opens up fascinating way to attack it. The entire AI field governed by mathematics, the underlying math architecture can reveal some important vulnerability avenues.

Model reversing : Visual Vistas

Ξ01 Neural Fabric Architecture

The AI deep learning models consist of multiple layers of neurons called hidden layers. These neurons are mathematical structure performing linear and nonlinear mathematical operations on various data structures such as tensors, vectors, matrix etc. Based on the architecture the neural networks learn the patterns in the data. To make predictions the non linearity added to these neural networks for learning through some typical non linear functions called ReLU (Rectified Linear Unit), Sigmoid etc. Below is the entire reverse engineered architecture of the Vision model activation functions , four dimensional tensor data, weights and biases, mathematical operations and much more. This itself reveals the architecture to the malicious actor. Note the partial snippet of architecture attached here as its a large to capture in a single image.



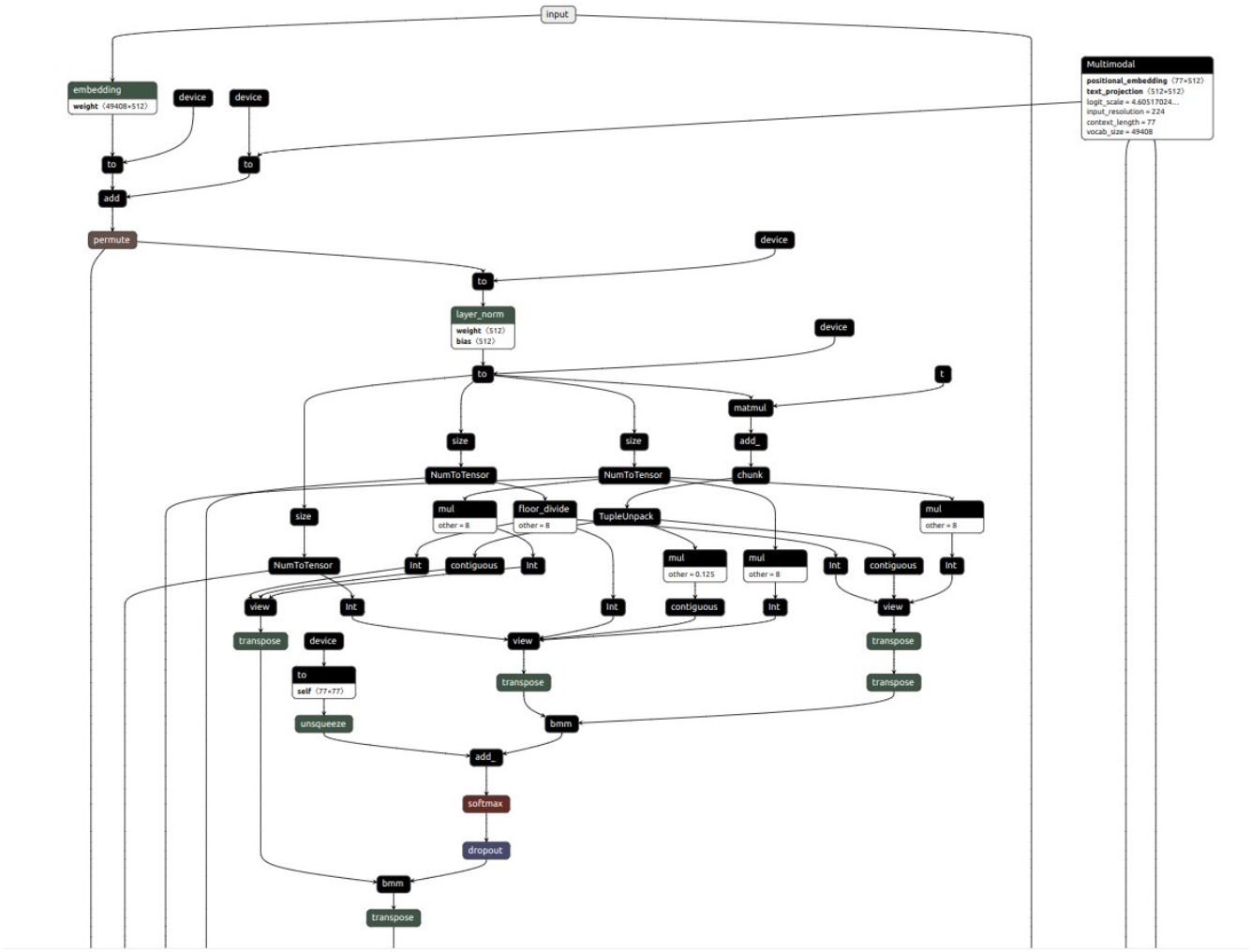


Fig.1 : Deep learning architecture reverse engineered

As we can see the above images reveal the architecture in terms of how input is entering, how it is interacting with multimodal architecture, and how further mathematical operations takes place.

⊗02 Tensor Alchemy

If we still zoom in at each layer, the mathematical information can be revealed as shown below,

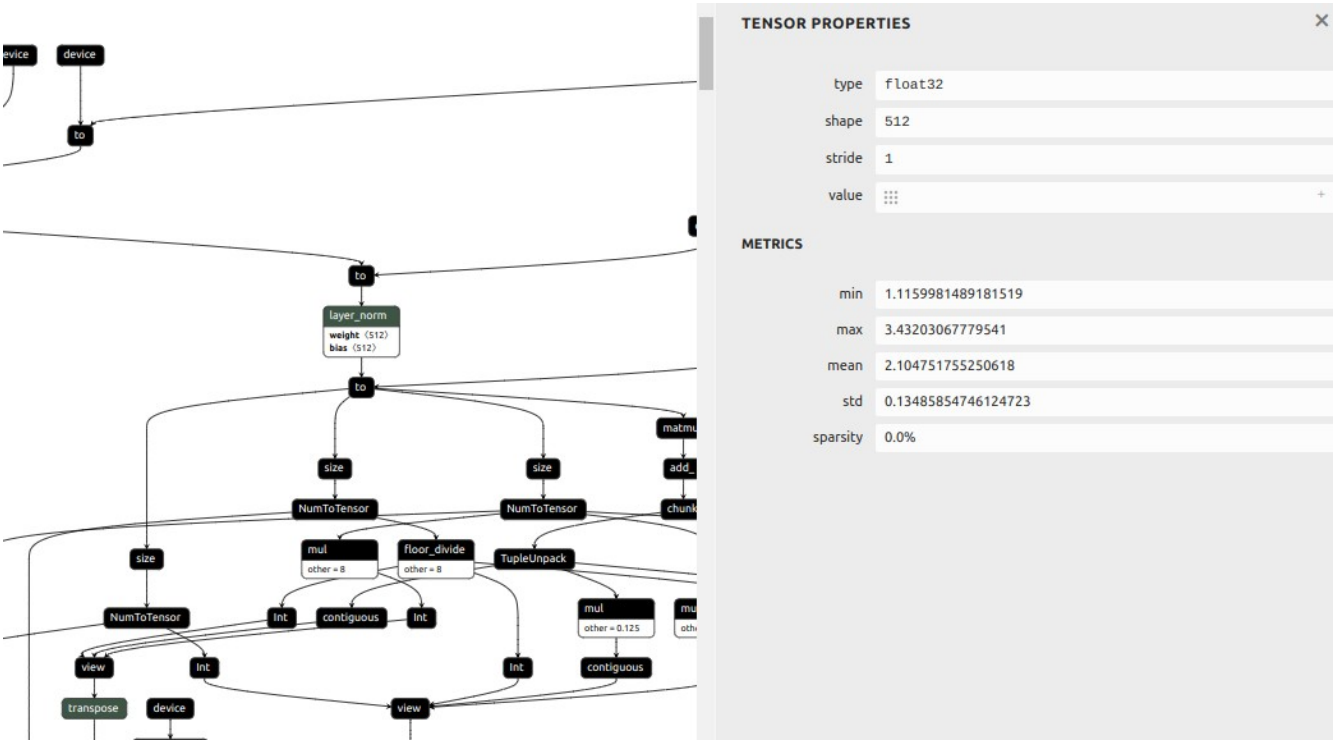


Fig.2 : Revealed tensor properties

Still deeper we go we find the values of weight tensors which attacker can use for crafting the model.

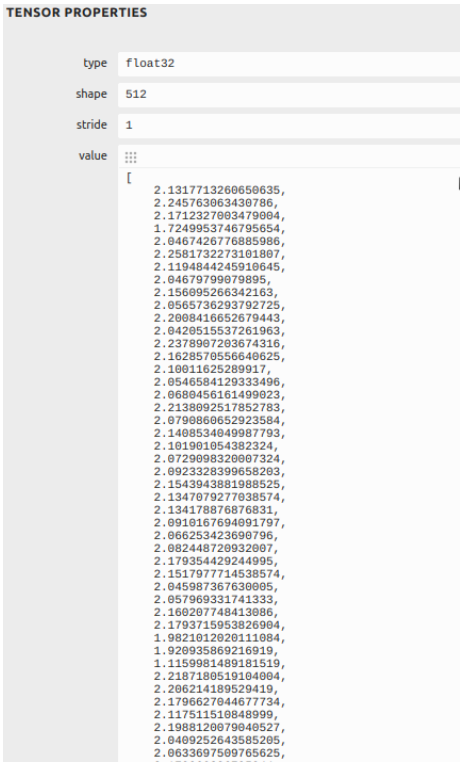


Fig.3 : Revealed tensor values of weights at one of the layer

These are critical parameters, when malicious actor tampers can make false predictions, does not converge to learn or it can be denial of AI service. Each neural network layer has weights and biases that help neural network to learn. The size of weights and biases is key idea for a malicious actor to fuzz it. The actual size is of the n-dimensional tensor which carries this data.

Ω03 Sparse Prism

If we continue going deeper and choose one layer and deep dive, it will give wealth of information such as its kernel size, strides etc. as shown below.

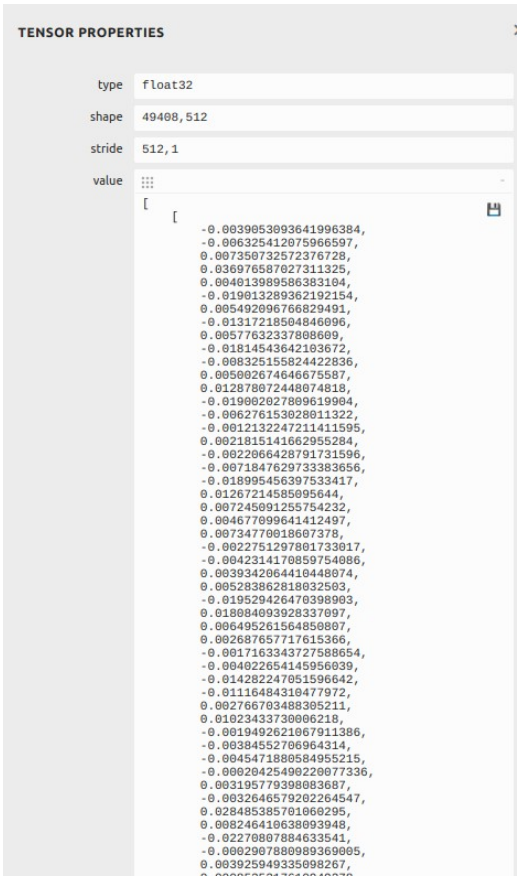


Fig.4: Revealed tensor values of weights at one of the layer

The neural network layer has weights, when reverse engineered reveal their details shown below which are of interest to an attacker. These are basically tensors which is holding the data for operation of data. The attacker can craft the exploit for the same. The same repeats for biases

When the metrics of these weights and biases as tensors explored further shows their min and max values as well as sparsity. Sparsity is critical parameter, it shows how dense is the tensor with non zero values. The below image shows sparsity shows 0.0%, meaning the fuzzing can break the performance as there are no zero values in tensor resulting in poor performance or DoS after after fuzzing.

So if the sparsity is reverse engineering to a mathematical equation we can infer something similar as below

In deep learning, sparsity in models like Neural Networks typically means having a significant number of zero weights (or close to zero) in the model parameters. This helps reduce computation and memory requirements and can also enhance interpretability.

The mathematical equation for sparsity can be defined in different ways, but a common measure is the *L0 norm* (or approximations of it) to count non-zero elements in the model weights. However, due to the non-differentiable L0 norm, the *L1 norm* is often used as a differentiable approximation to promote sparsity:

number of elements in Sparsity=Total number of elements in W$\|W\|_0$

where:

- W is the weight matrix (or tensor) of a layer or the entire model,
- $\|W\|_0$ represents the L0 norm, which counts the number of non-zero elements in W.

Alternatively, using the L1 norm:

Sparsity=n$\sum_{i=1}^{n} |W_i|$

where n is the total number of weights in W. The closer this measure is to zero, the higher the sparsity.

Another way to achieve or encourage sparsity is by using a sparsity regularization term in the loss function, often applied as a penalty:

Loss=Original Loss+$\lambda \|W\|_1$

where λ is a regularization parameter that controls the trade-off between model accuracy and sparsity.

For sparsity 0.0% we can infer as below

In terms of the equations above:

1. **L0 Norm Perspective**: The sparsity equation is That is, there are no zero weights, and every weight in W contributes to the model. Sparsity = \frac{\|W\|_0}{\text{Total number of elements in } W}
2. **L1 Norm Perspective**: Since the L1 norm does not directly measure the count of zero elements but rather the sum of absolute values, a 0% sparsity would indicate that no weights are close to or equal to zero, meaning the model is fully dense.

Likewise these vulnerabilities repeat in same architecture at different layers and for different models of multimodal architecture such as language models which can be exploited as a combination of multiple layers making huge impact on AI model performance, flexibility and learning ability. The reversed architecture images for language and audio are shown in Appendix I.

The format reverse engineered above for .pth and .pt formats which are of pytorch framework.
So in summary deep learning AI model reversing can result in some new vulnerabilities associated with mathematical data structures such as the tensor values, architectural disclosure, sparsity, min, max values.

Results : Metrics in Motion

#	**Reverse Engineering Element**	**Disclosure**	**Result**
1	Deep learning model architecture	Entire model, inputs, weights, biases, output, operation etc	Model IP revealed
2	Weights and biases	Tensors,tensor properties viz. Size, shape, stride, weight and bias values	Model training parameters revealed to an attacker and can be crafted for other data
3	Tensor Metric	Min-max values, Sparsity, Standard deviation	DoS attack, backdoor to dense network as 0% sparsity,

Appendix I : Analysis of Sound and Syntax

Below images are extensions for multimodal architecture for language and audio neural network architecture

Fig.5 : Multimodal architecture section for Language and Audio with their parameters

If we expand the above architecture, we discover the encoder-decoder architecture for language and audio with their tensor properties.

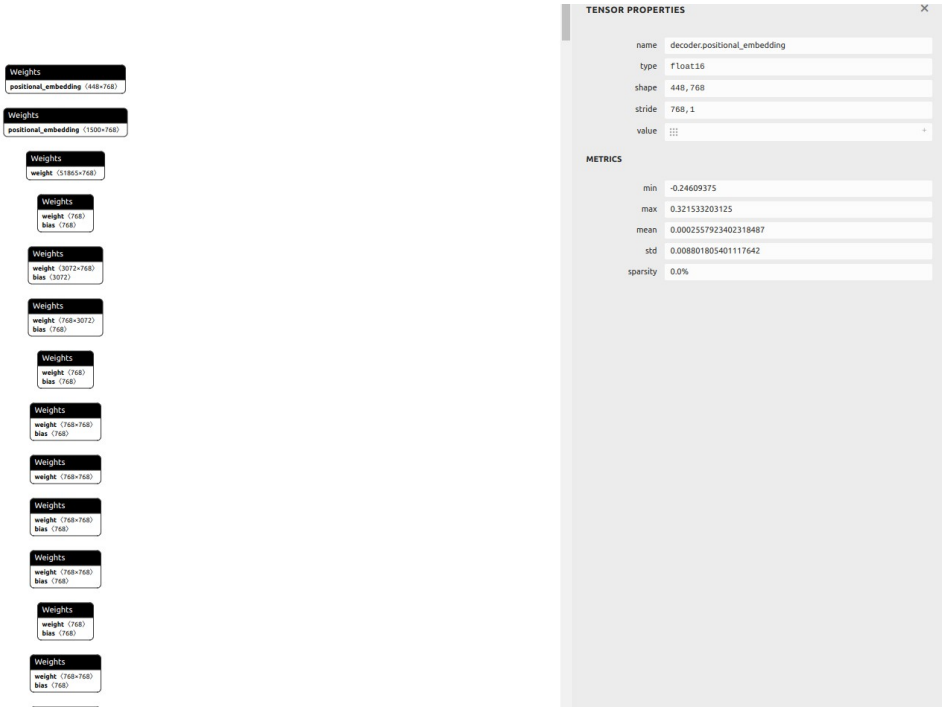


Fig. 6: Partial architecture with weights for Decoder positional embedding

Similarly the encoder architecture with its tensor properties and metrics can be visualize as below.

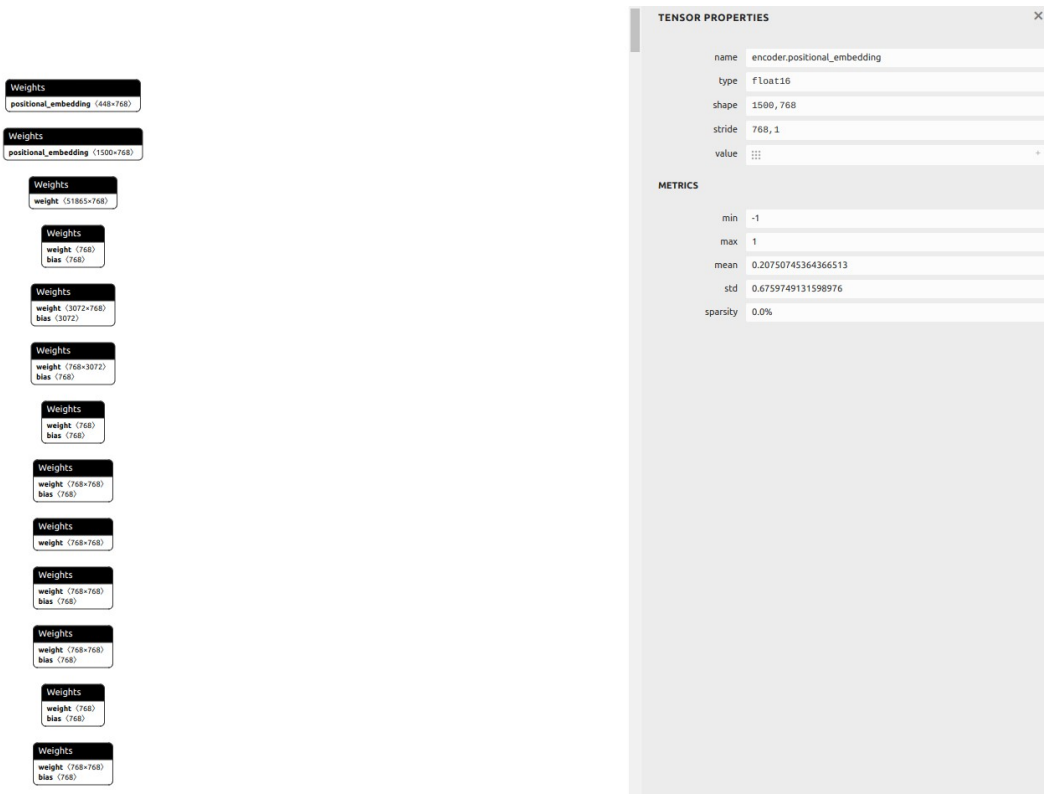


Fig.7 : Partial architecture with weights for Encoder positional embedding

References

1. Zhifei Xie, Changqiao Wu, Mini-Omni2: Towards Open-source GPT-4o with Vision, Speech and Duplex Capabilities
2. Mini-Omni2 github
3. Multimodal AI Models
4. MITRE ATLAS Matrix
5. OWASP ML Top 10
6. OWASP LLM Top 10