Electronic Design

ERNEST WORTHMAN, Contributing Editor

Deploying Edge AI: From Concept to Production

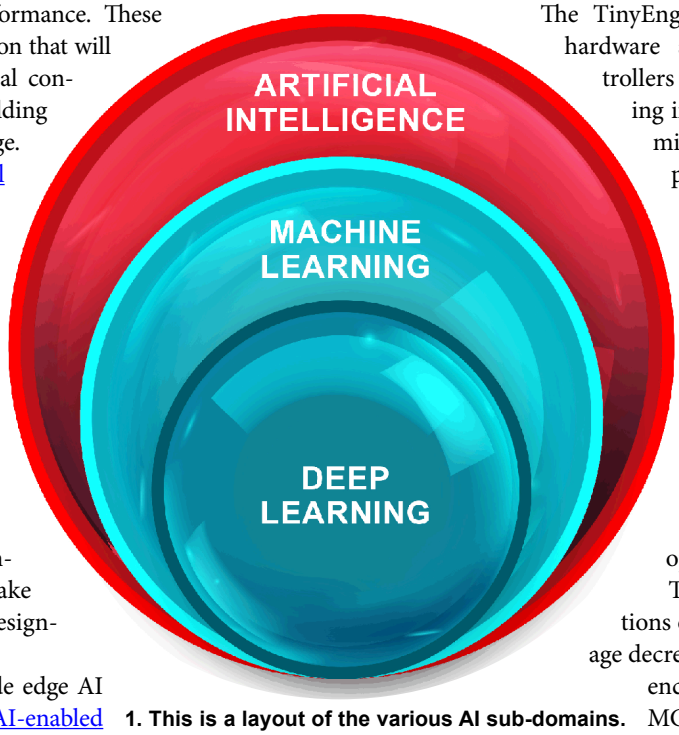
Sponsored by Texas Instruments: MCUs are opening the field for extreme edge development, unveiling a new age of possibilities and solutions — especially with the integration of neural processing units.

It seems that the edge has become the topic du jour lately, but it's not as new as most of us think. But before we get into the weeds with this paper, let's take a trip down memory lane; back to 1978 to see how TI engineers developed the first voice-synthesized DSP embedded device, the Speak and Spell, and how it led to today's advanced embedded devices.

From the Speak and Spell to the present day, we have developed advanced AI-enabled microcontrollers. These next-generation devices will revolutionize the capabilities of edge networks. We now have edge devices that offer heterogeneous compute, multimodal on-device inference, and robotics-grade real-time performance. These devices are an elegant solution that will conquer many of the critical constraints that have been holding back the evolution of the edge.

TI has developed a tool chain and a library of examples that significantly simplifies the development of edge networks using AI. This resource is the go-to tool for embedded design, whether the project demands low-power microcontrollers (MCUs) or high-performance processors. Having access to such a comprehensive toolset can take much of the stress out of designing edge AI applications.

Perhaps nothing has made edge AI possible more than the AI-enabled microcontroller. Microcontrollers are being integrated with neural processing units (NPUs). With these AI-enhanced MCUs, designers can deploy edge AI in various applications such as battery products, wireless devices, radar and LiDAR sensors, predictive-maintenance systems, intelligent sensors, cameras and autonomous vehicles, as well as a slew of other resource-constrained applications.

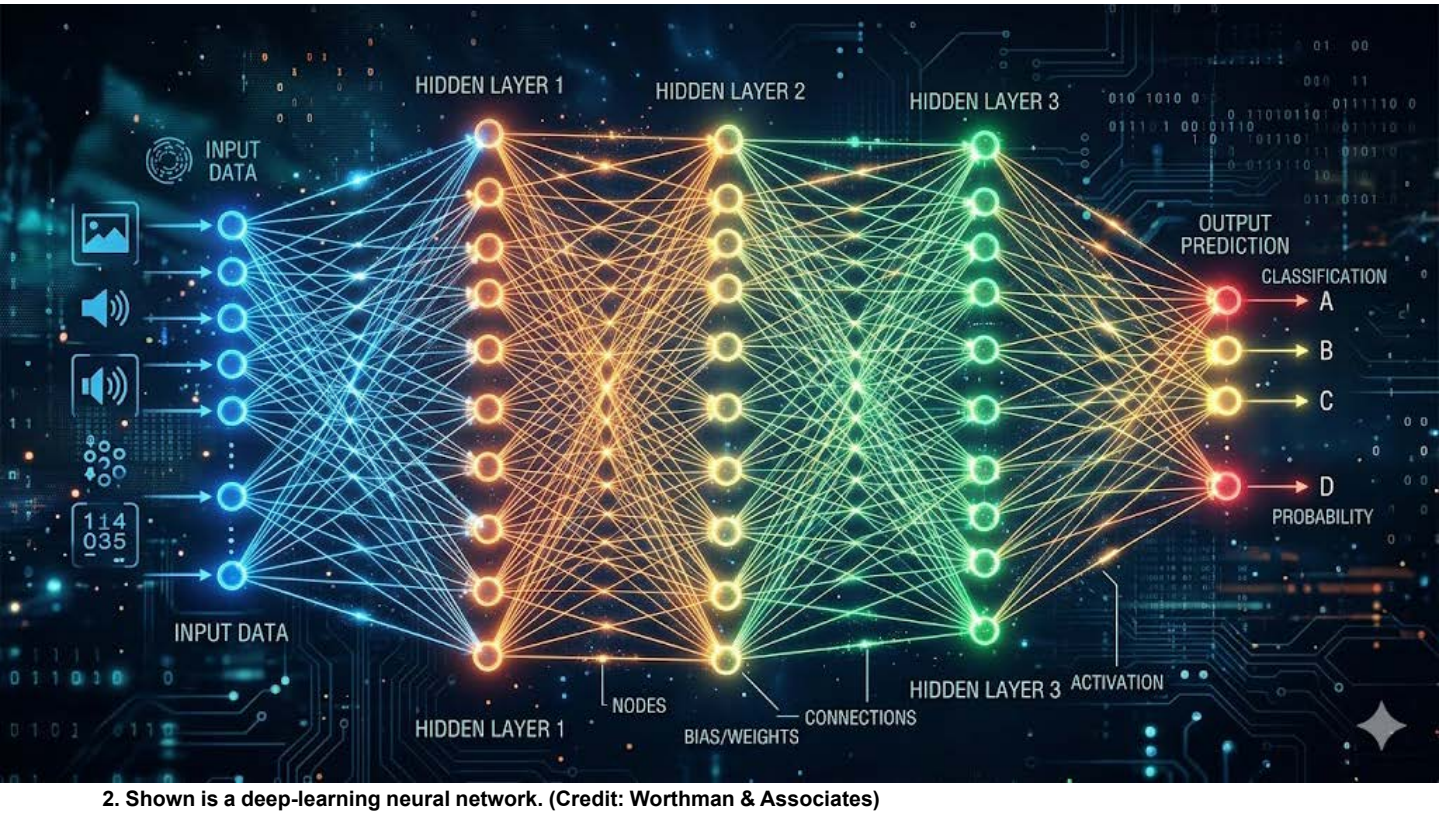


1. This is a layout of the various AI sub-domains. (Credit: TI)

NPU-Enhanced MPUs

The heart of edge network performance comes from TI's C2000 and Arm Cortex-based edge-AI accelerated and edge AI-supported MCUs. This technology is integrated in the form of the TinyEngine NPU hardware accelerator. The TinyEngine NPU is a specialized hardware accelerator for microcontrollers that enhances deep-learning inference processes to minimize latency, optimize parallel processing, reduce power consumption, and more. Much of its performance comes from its ability to process concurrent streams of sensor data by shifting deep-learning inference from the MCU to a parallel, sparsity-aware hardware engine that executes quantized neural-network operations.

This yields latency reductions of up to 90X and energy usage decreases of over 120X per inference when compared to similar MCUs without an NPU.



2. Shown is a deep-learning neural network. (Credit: Worthman & Associates)

Unwrapping the NPU

Artificial intelligence is the mothership, if you will. Within it are the subsets of machine learning (ML) and deep learning (DL). So, when we discuss edge AI, we're really talking about DL and ML, where DL is a subset of ML, which is a subset of AI *(Fig. 1)*.

It can be confusing, and further unpacking of these platforms is outside of the scope of this article. Therefore, we'll focus mainly on DL, which is the primary element giving edge AI its capabilities and performance.

Deep learning has the capability to address highly complex problems that would be too computationally intensive and resource-heavy to run at the edge. It accomplishes this by integrating a multilayered architecture that implements automated feature extraction and end-to-end training, which creates intricate, high-dimensional data. This renders it an effective instrument for tasks where conventional machine-learning methods are inadequate.

Deep learning uses multilayer neural networks, which are data models resembling the neurons of the human brain *(Fig. 2)*.

Neural networks are the backbone of edge networks because neural-network math can be done entirely in parallel. The significant advantage here is that this massive parallel multiplication uses only a fraction of the power of a standard CPU.

When designed as an NPU-enabled MCU, optimizing for functions such as zero latency, low bandwidth, and privacy/security, one has an incredible computational machine that's perfect for the constraints of modern edge networks.

The NPU and Deep Neural Networks

In the deep neural network shown in *Figure 2*, note that each node has multiple connections. Each connection carries a weight — a number that determines how strongly one node's output influences the next node's input, and a bias. Both are parameters that can be manipulated to optimize the model. This is where deep learning takes place.

Weights are numerical values assigned to the connections between neurons (nodes) in different layers of the network. These weights determine the influence of one neuron on another and are adjusted during training to minimize prediction errors.

When input data flows through the network, it's multiplied by the weights associated with each connection. This process emphasizes certain features of the input data while diminishing others, based on model design.

For example, in image recognition, weights might prioritize pixels that represent key features like the shape of an eye or the width of a nose. Properly tuned weights allow the

network to generalize unseen data, helping to make accurate predictions in real-world scenarios. The more data the DL processes in this fashion, the more accurate the model.

Bias is a learnable parameter applied to the node's weighted sum and adds flexibility to the activation function. This enables the activation function to be shifted left or right. As a result, the model can better fit the data by adjusting the threshold for activation. Bias ensures that neurons can activate even when the weighted sum of inputs is zero. With such flexibility, the network is able to model more complex patterns and decision boundaries.

There are, of course, more variables that come into play in these networks — the activation function**, for example — but space limits the discussion to just a cursory mention. However, bringing all of this together in the form of NPUs and designing them to integrate with certain MCUs offers a rather elegant solution for edge networks.

Conclusion

Going forward, the data-management ecosystem has a number of mercurial challenges ahead of it. Edge networks are emerging as a solution that, if done smartly, can help keep data flowing smoothly and unimpeded. They're able to take much of the load off the core, which will soon be unable to handle the massive volume of data coursing around the electronic superhighways.

The coupling of purpose-built MCUs with AI creates a new era of intelligent data management. This will be one of the great enablers of the emerging edge ecosystem. As the edge evolves, there will certainly be many challenges as well as opportunities. This time, standing at the edge will have a totally different meaning.

***Upon receiving a signal at its input, a neuron executes a sequence of operations on it. The neuron multiplies the weights linked to the input connections by the respective input data. This produces a weighted sum of the inputs. The neuron subsequently incorporates a bias term into the weighted sum. This weighted total, inclusive of the bias term, signifies the aggregated information that the neuron acquires from its inputs. The neuron applies its activation function to this value to generate the final output, which is transmitted to the successive layer of the neural network.*