2024 | WHITE PAPER

AI IN INDUSTRY

Fitting AI solutions on connected platforms in real-world industrial applications

Dirk van den Heuvel | Embedded systems architect

TOPIC

CONTENTS

For more information go www.topicembedded.com or contact us via +31 (0)499 33 69 70 | contact@topic.nl

INTRODUCTION

Artificial Intelligence is often presented as the magic wand that solves our problems seamlessly. However, there is no such thing as a free lunch. When bringing the benefits of AI to industrial challenges, you need to consider the preconditions to facilitate AI-based algorithm processing. As an example, many industrial applications cannot rely on the availability of large remote data centres with infinite computational power. Therefore, it is important that the deployment of AI algorithms in regular industrial settings meet typical industrial requirements like limited processing- and electrical power, fast real-time response, and a suitable mechanical fit. Also, safety, reliability, robustness, serviceability, and other requirements must be considered. This often also implies that the latest silicon technologies need to find their way into environments where the typically implementation choice is driven by proven technology.

In this paper TOPICs experiences and insights are shared with deployment of AI solutions in real-world applications, addressing hardware, software, and firmware aspects. The pre-conditions and consequences of AI based solution are discussed from a deployment perspective and cover the connectivity, processing platforms, board design consequences as well as system integration.

As a strong player in the domain of medical and industrial applications in Europe and North America, TOPIC has been involved with AI applications for nearly a decade. It is a natural and welcome addition to the mathematical algorithm developments on which many of the realized projects by TOPIC are based. When capturing phenomena in formulas becomes too complicated, artificial intelligence can often offer a solution. However, an overlooked aspect is the fact that AI algorithms need to be trained and that training is one of the key success factors of an algorithm, but can be costly in terms of time and budget.

The distinct difference between mathematical and AI algorithm design is the evolution of the quality of the produced result. In case of mathematical designs, there is no evolution: it is what it is. You may come with different insights, but you need to enhance the model to incorporate this. In case of an AI algorithm, you determine and tune the basic algorithm to the specific application. Then you start training with available data. The quality of the algorithm is determined by the set of unique applied examples and weighted results. Over time, your algorithm becomes better trained and will keep on advancing in quality as long as you feedback data to the training model and update the deployment model. Therefore, a decision to choose AI over mathematical models is a balance between spent effort, quality objectives and complexity of the problem.

TO AI OR NOT TO AI: NOT A TRIVIAL QUESTION

A nice trade-off example was presented by a customer of ours. They are a leading high-quality mattrasses manufacturer and can configure pocket spring mattrasses with variable tension springs to optimize the sleeping comfort of the customers. It sounded like a typical AI problem: not a single person is the same, there are a lot of different springs in the mattress with many input parameters as well as possible outcomes. A special calibration mattrass was designed to characterize a person. Apart from technicalities reading each springs tension when measuring a person, the algorithm training was taking place with a too limited initial test set. When exploring the AI algorithm characteristics, the conclusion was, that you need over 1000 unique test persons to characterize the calibration data set. You also need the same amount of mattrasses produced to validate if the selected mattrass was the right mattrass for that customer. The cost of the training turned out to be very expensive. In the meantime, a TOPIC mathematician figured out a way to characterize the calibration mattrass in a different, purely mathematical way. The years of experience of factory personnel with mattrasses design and production resulted in a promising solution with a feasible result. But it is not AI and therefore the marketing value was too low. A different approach was pursuit by the customer.

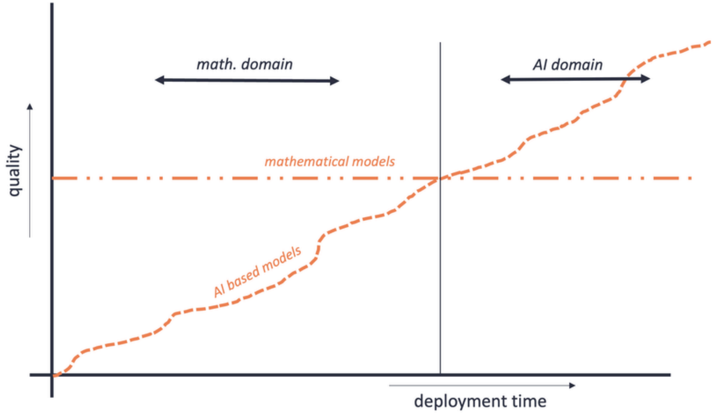


Figure 1: Mathematic modelling versus AI-based application development

WISH: AI IN A GLANCE

One of TOPIC's first AI applications was WISH (Workflow Information System for Hospitals). The goal is to predict the end time of routine procedures executed in operation theaters (OR), like knee and cataract surgery. To enable this, smart sensors are distributed over the OR, sensing e.g. mains current fluctuations due to the use of specific equipment and use of a smart instrument table, sensing change in weight and shadows on the table when changing instruments. The algorithm was trained to recognize patterns in the readings and correlate these to the expected pattern. From this a prediction of the end time of the procedure can be established. As a result, the logistics around the OR can be optimized, reducing uncertainty in patient preparation and reduce operational cost as the theater is used more effectively. The AI algorithm is running on a dedicated x86 based server with GPU support used for training, deployment, and execution. The server can be hosted in the cloud or locally in the hospital. The quality of the prediction is clearly influenced by the number of procedures executed.The outcome of the end time predictions can also trigger ethical questions. How do you deal with surgeons who structurally need more time to complete a procedure? Is that time spent on a specific part of the procedure? Does the quality of the procedure correlate to the time spent on the procedure? This kind of data becomes available and quantifiable. The example also shows that algorithms for recognizing patterns are not just mathematical, but more organic: there is not an absolute right-or-wrong answer.

Figure 2, on the next page, illustrates a brief summary of a typical AI development flow illustrating the different steps that were made when implementing the WISH concept. The five steps are iterative and passed multiple times. The first step is to gather the required data and get them into a form that you can feed to your algorithm. We learned that the mains currents needs sampling with significantly high dynamic range to identify the usage of specific devices. To distinguish the usage of a plasma knife or switching on a LED is quite a challenge. Here, the real learning takes place.

Also, the discovery that a solar panel can act as a carrier plate as well as a sensor on the instrument table was quite rewarding. Apart from this, many more parameters are being collected as input for the algorithm. The implementation of the algorithm is making use of an existing suitable model architecture and specific domain expertise from the field to tune the model. Using sensor readings, either artificially obtained or acquired from the field, are then used to train the model. After this initial training, the model is then deployed in the real application.

In the WISH context, the end-time prediction model was deployed on the same server where the training was taking place. The AI model used to detect objects on the instrument table was deployed on an edge device in the table itself, partially relying on sensor data supplied by the server. It was a form of "training-on-the-job", where feedback was immediately looped-back to the training module, enhancing the algorithm as it goes. Every now and then you need to revisit the model architecture itself for an optimization task. However, this implies that you must retrain the model again with all the stored examples.

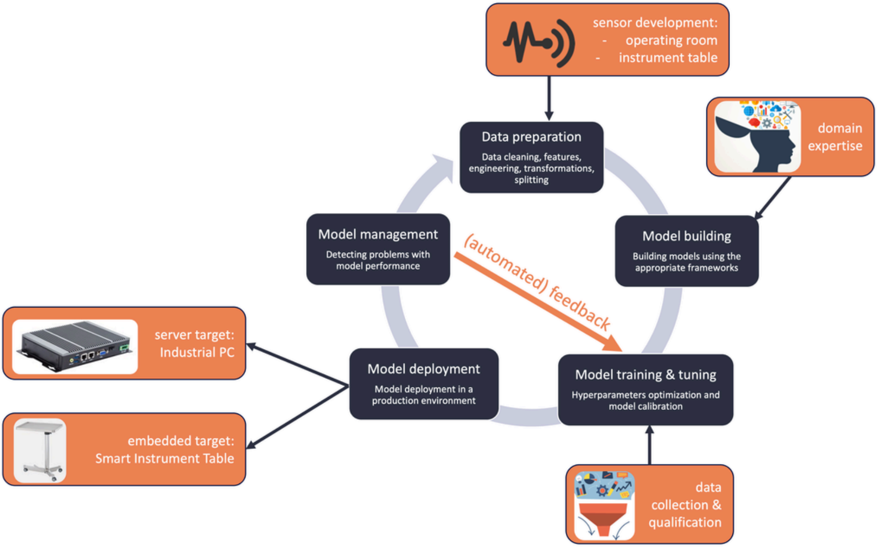

sensor development:
- operating room
- instrument table
domain expertise
Data preparation
Data cleaning, features, engineering, transformations, splitting
Model management
Detecting problems with model performance
Model building
Building models using the appropriate frameworks
(automated) feedback
server target: Industrial PC
Model deployment
Model deployment in a production environment
Model training & tuning
Hyperparameters optimization and model calibration
embedded target: Smart Instrument Table
data collection & qualification

Figure 2: Typical AI implementation flow

ENABLING AI ON THE EDGE

With the advances of silicon technology, you see dedicated devices being released that allow deployment of AI algorithms also on the edge without involvement of connected servers. The dedicated neural network infrastructure on these devices is optimized to meet the specific needs for AI algorithms. You always see a controlling processor next to the neural network processing unit to manage the applied data and algorithmic steps. This can be an external processor or that the accelerator is part of an SOC. The latter is highly preferred as edge devices require to be relatively small and power efficient.

Next to the physical neural network implementation, there is always an eco-system with a processor platform to map trained models onto the neural network. As a premier partner of AMD Embedded, formerly known as Xilinx, TOPIC saw the first instantiation of a neural network appear as a convolutional neural network (CNN) with limited performance, limited complexity and limited data resolution on FPGA fabric. However, this enabled already quite a few interesting applications. Especially, vision applications for recognizing multiple objects simultaneously at high speeds were key indicators that AI was becoming mature as a technology and suitable for use in real application on the edge.

On FPGA devices, the neural networks were denoted as DPU, Deep-learning Processing Units. Google/Corel introduced the TPU (Tensor Processing Unit), followed by many more attempts to bring AI algorithms from the cloud to the edge. As an example, TOPIC is at this moment exploring a newly released AI-focused SOC by SiMa.ai.

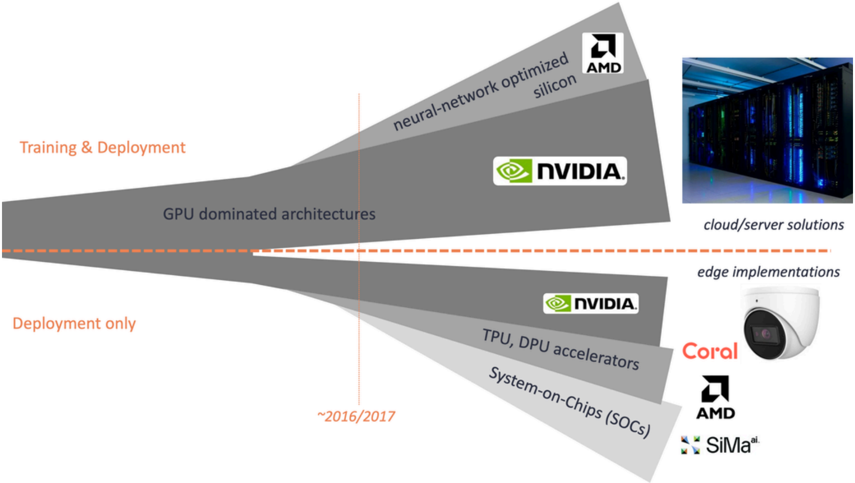


Figure 3: AI edge deployment evolution

Given the amount of data required to execute an AI algorithm, the move from the cloud to the edge for deployment is very useful to reduce the amount of communication traffic. A project TOPIC executed for a leading technology company, involved 4x 4K MIPI camera's observing rooms in a 360 angle. This leads to an uncompressed video pixel stream of around 50Gbps. The application was realized using system-on-chip (SOC) technology, incorporating FPGA fabric to interface real-time with the 4 MIPI video streams and detect/isolate regions-of-interest (ROI) in the derived images. The regions-of-interest were streamed by the incorporated multi-core processor into the AI engine. The AI engine was realized as part of the FPGA fabric. The output of the AI engine was then communicated with a cloud server via a WiFi connection. A typical characteristic of such an edge device is that it is limited in available size, available power and communication bandwidth. The solution uses an AMD Zynq Ultrascale+ SOC and was suitable for demonstrating the feasibility of functionality within the given power and space budget.

AI DEPLOYMENT FLOW

A dominating factor in deploying a trained model on a target device is the tool flow you need to translate or compile the model to the neural network on the device. It is very common to train your AI model using double precision floating point accuracy. However, typical edge implementations are using integer-based processing, even limited to 4 bits. Translating a trained AI model from floats into 4 bits integer numbers gives loss of quality. However, all neural network vendors provide tool flows that support engineers in this decimation process and limit the loss of quality as much as possible with quantified numbers. A typical flow, based on AMD Vitis AI, is illustrated in figure 4, highlighting key optimization steps. Other providers of neural network solutions provide similar implementation flows.

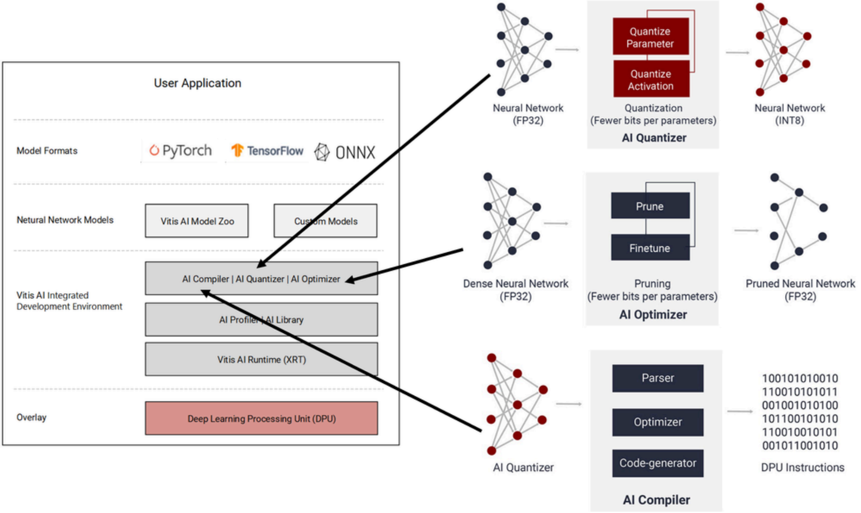


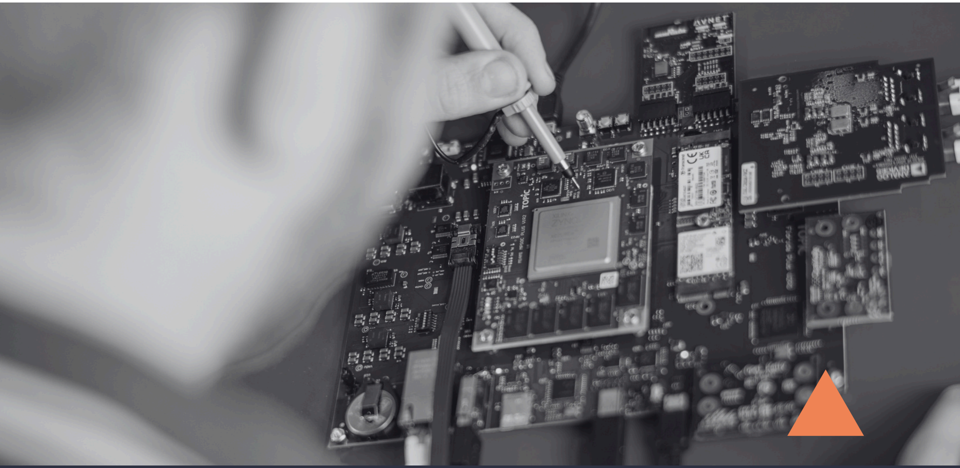
Figure 4: Example AI deployment flow

First, you match the developed AI model with a model from the Model Zoo. These are pre-defined networks with specific characteristics. But you can always configure your own based on your experience. Using that input, the trained model is matched with the selected model. In the second step, you need to match the trained model data representation with the implemented neural network data format.

Typically, the tooling supports this in three steps:

- First, the optimizer tries to reduce the neural network complexity. This is referred to as pruning. This will cost accuracy, but as a programmer, you can determine if the loss is acceptable.
- Secondly, the data representation is reduced to whatever is needed. Most common at this moment is a reduction to 8-bit integers. There are quite some developments towards higher resolutions on the short term. In this step, also interaction with the engineer is required to monitor and steer the accuracy loss.
- The last step is the actual compiling of the pruned and quantified model on the targeted network.

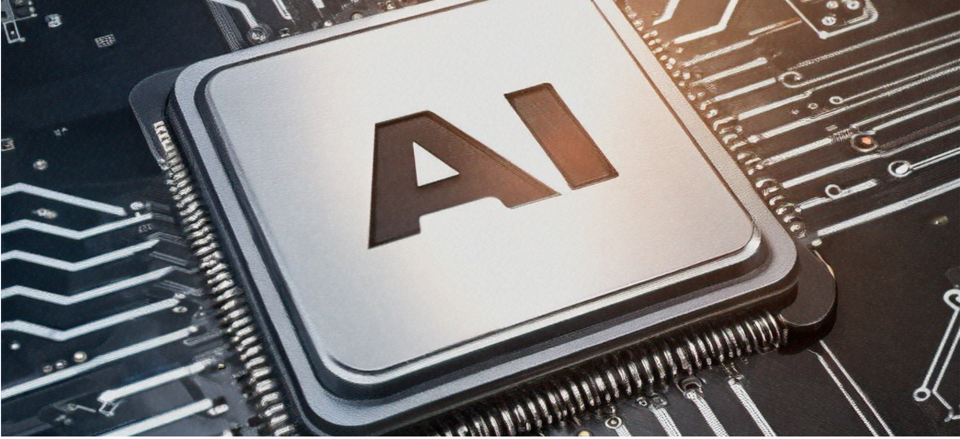
This network needs to be driven by data, controlled by the connected processor. A profiler is then in place to help analyze the execution pipeline over time. You can also have support to time the data production and consumption side to optimize performance further. After this step you have a deployed AI engine that you can integrate in your (software) application running bare-metal, Linux or a qualified RTOS.



AI @ WORK

Not just the deployment flow keeps advancing. With the ever-decreasing silicon technology geometries, now even at 3 nm, logic densities keep on increasing and enable cost-effective, relatively low-power programmable neural network architectures on silicon devices as part of system-on-chips. This also enables neural network nodes with higher precision. Like all systems around, the complexity of a neural node, the number of neural nodes and clock rate of the network are highly determining the performance figures of the neural network. For this reason, you see various flavors of neural network processors appearing, addressing a variety of applications around. This gives way to many more applications that were previously not feasible. An interesting domain form agricultural applications. AI has many advantages here over traditional programming methods as a property of nature is, that nothing is the same. As an example, one of TOPICs customers requires to inspect thousands of eggs every day for cracks using optical inspection. If an egg with a crack is detected, you have limited time to remove an egg with a crack before it is being further processed. The eggs are inspected with more than 10 high-resolution/high-speed cameras and multiple times per egg. An AI algorithm processes snapshots of a particular egg from multiple angles in a limited amount of time. This sounds like a trivial exercise, but the amount of data to process is massive, the AI algorithm complex, the available space in the machine limited and the optical path complicated. However, AI is here the right way to go, superior to the previous sound-based crack detection method. The sheer volume of eggs passing the system is a great way for training and improving the algorithm with every egg passing the algorithm.

This is an example that is representative for many agricultural sorting and/or processing problems. TOPIC has been involved with quite a few projects in this domain where AI was driving the solution. This involves cutting leaves from plants in greenhouses to re-planting seedings into larger pots. But many more applications can be identified. Especially when you can combine this with autonomous moving robots (AMR), a solution that can be resolved perfectly using a single system-on-chip (SOC), when you choose the right one.



CONCLUSION : IS AI A FIT ON THE EDGE?

Like all technology developments, AI is not the holy grail. AI forms a part of the solution. However, AI has become a mature technology over a reasonably short period of time, also as an edge deployment solution. The examples mentioned in this paper illustrate this clearly. They demonstrate successful deployment of AI based solutions on the edge as a fusion of sensor data acquisition, pre-processing, neural network execution and output control.

Care must be taken that it is not suitable for all problems: complexity and cost for training need to be accounted for. Heterogeneous processing solutions are required as edge solutions, embedding the AI implementation in end-application. With the increasing amount of silicon platform solutions, the question is not if an AI is a fit, but which configuration is the best fit. Also the tools supporting AI deployment are maturing rapidly, supporting the process of bringing AI to the edge.

As such, the conclusion is that AI is a fit as reliable implementation method embedded devices. Feel free to share thoughts, ideas, questions and others remarks by contacting us.

ABOUT TOPIC EMBEDDED SYSTEMS

“We make the world a little better, healthier and smarter every day”. Our mission statement reflects exactly what we do: developing innovative systems for our customers. The way we do that, is by combining our customers domain specific know-how with our expertise in hardware and software development. This results in the most optimal product for our customers. TOPIC has a strong background of more than 28 years in developing systems, which can contain embedded-, application- and cloud software, FPGA code and PCB designs. We help customers in different domains such as medical, imaging, machine control & safety. With over 130 employees, we are a strong and established company with our headquarters in Best, the Netherlands. TOPIC has an ISO13485 (medical) certified Quality Management System and adopted the Agile way-of-working for optimal interaction with the customer.

Premier Partnership with AMD | TOPIC is one of the few AMD Premier Adaptive Computing Partners in the world. Our partnership with AMD started in 2009 and since than we have been working closely together over the last years.

Materiaalweg 4, 5681 RJ Best, The Netherlands

+31 (0)499 33 69 79

contact@topic.nl

www.topicembedded.com

linkedin.com/company/topic-embedded-systems