Patents for A.I.: Hardware 1 - Software 0

by François Veltz - Algopatent

8 July 2019

Discoverability is a key factor to assess patents. Patent attorneys generally claim what can be visible, or detectable without excessive burden. Or at least they used to. Discoverability may evolve over time, for example if products or services are made available to the public (e.g. sales, publication of documentation). Regarding hardware and software, the very frontiers of discoverability are changing rapidly, noticeably with respect to *local* versus *remote* access.

A.I. for "Artificial Intelligence", or better "Augmented Intelligence", may be redefining the relations between hardware and software.

The contemporary shift towards Cloud Computing has buried numerous software inventions deep in inaccessible data centers, letting just user interfaces or APIs exposed. Reverse engineering accessible software (decompiling) is hard, or can be rendered hard (e.g. obfuscated if not hardened code). But it may also be not possible, if the code is distributed. In addition, Case Law for software regularly changes (*Bilski v. Kappos*, *Alice Corp. v. CLS*). As a result, these aspects have globally lowered interest in patenting invisible software.

By contrast with software, the tangible character of hardware (by nature) appears to be more favorable to patentability. Discoverability can remain though, but at least some chips are distributed over the world (while numerous others also remain hidden in data centers).

In patent applications directed towards computer science, patent attorneys used to consider that there exists a kind a *"duality" of hardware and software embodiments*. It is generally asserted that any software generally can be encoded into a tangible circuit. And a tangible circuit can be expressed in logical ways, i.e. by software. Such language can be found in most patent applications, where patent attorneys try to generalize and cover material and logical scope. *Yet this rule of thumb might not be always true.* This couple *"hardware versus software"* to some extent may correspond to the one of *"reality versus mathematics"*. It is known that this question is a tough one, at least a marvelous coincidence. Program, logic, mathematics, software and hardware are different words, and raise many questions. May any arbitrary program be encoded into a material hardware form? If not, what are such programs? Conversely, are there circuits (e.g. 3D, biological) that cannot be emulated in software? At least, as a matter of efficiency, it may be that a same program can be reduced to practice in a lot of different ways, more or less efficient given evaluation criteria.

Nowadays, the current emphasis set on inventions in Artificial Intelligence, and more specifically on Machine Learning (acronym ML), indicates an emerging phenomenon.

For decades, the Von Neumann architecture (schematically: inputs, processing units, memory units, outputs) has not been changed. GPUs have been *incidentally* (i.e. luckily) used in machine learning, because of parallel computing capabilities. Now that IT giants have invested or are planning to invest in artificial intelligence, new hardware is being patented, *redefining the relationship between hardware and software.*

Industry players have been using *existing* hardware, e.g. GPU for deep learning. But AI and ML probably have been ill-served by general-purpose processors so far. Some have started building *new* hardware for A.I. Without any dominant design, A.I. accelerators are often manycore designs (parallel computing) and generally focus on low-precision arithmetic, novel dataflow architectures or in-memory computing capabilities (beyond the so-called Von Neumann bottleneck).

Software evolves along new hardware. For example, CUDA allows developers to write apps to use the Nvidia GPU for parallel processing instead of a CPU. It is debated whether software is developing enough to leverage new hardware (education, number of developers, programming languages, etc), or is lagging behind.

A flood of new specialized "AI" processors has arrived indeed. Google has developed several generations of TPUs for Tensor Processing Unit (TPU), which are A.I. accelerator application-specific integrated circuits, specifically made for neural network machine learning. IBM plans to evolve to analog devices, and then fault tolerant quantum computers. For example, phase change memory, a next-gen memory material, may be the first analog device optimized for deep learning networks. Intel releases systems for image processing AI. Microsoft is preparing an AI chip for its VR/AR headset. Amazon is working on an AI chip for Alexa. AI-dedicated startup chip companies are being funded. Emerging technologies promise to further hybridize the field (e.g. stretchable electronics, bio-inspired mechanisms, 3D arrangements of circuits, or the like). Specialized AI processors may end up being merged into general purpose CPUs. On top of that, brand image and marketing also play a role. Major manufacturers have all given their AI chips cool names. The iPhone for example presents a "Bionic Chip" with a "Neural Engine" in it. Huawei has presented its "Neural Processing Unit (NPU)". Autonomous cars are said to require many of such specialized AI Chips. Recently, Google announced Stadia, a cloud gaming service relying on a specific GPU, enabling elastic computing in data centers.

The evolving relation between software and hardware is a fascinating one.

Accordingly, it seems likely that in coming years innovation around artificial intelligence will revolve increasingly around novel hardware. It remains to be seen whether the conventional software/hardware duality will remain valid. In any case, such inventions are likely to be more easily discoverable, and to be better aligned with the definitions of patent eligibility applicable in different jurisdictions, so that continued growth in the issuance of patents in this field is to be anticipated.