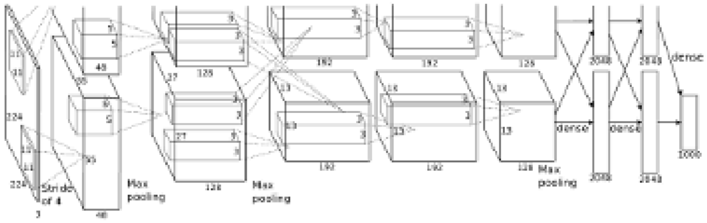
James Skelley

Technology Law - IP Prosecution - Transactions

MAIN UPDATES LAWMUX KNOWLEDGEBASE SOFTWARE FAQ CONTACT SUBSCRIBE

“Can I patent my Deep Learning [X]?”

February 18, 2019 SKELJ Uncategorized 0



From ” ImageNet Classification with Deep Convolutional Neural Networks” Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Advances in Neural Information Processing Systems 25 (NIPS 2012)

I’ve been getting this question a lot from folks lately (mostly after ultimate games for some reason), so I thought I’d summarize my response here for purposes of economy.

To recap the context: After much time and suffering, you’ve finally trained some combination of layers using, e.g., via TensorFlow, Torch, Caffe, etc. to your satisfaction. You’ve run the system through a variety of production-grade test sets and are satisfied that it’s ready to go live. At this point, could anything you’ve done be amenable to patent protection?

The short, incomplete answer is “yes.” Much of your work probably *could* be protected via a patent if you *wanted* to acquire such protection. There’s nothing “magical” about deep learning as compared to other development situations involving some

RECENT UPDATES

DMCA Study

Skelley Snapshots

CPC Based Deal Search (i.e., not a COVID post)

NAVIGATION

Main

Updates

LawMux Knowledgebase

Software

FAQ

Contact

Subscribe

automation – say, automated car production or a cooking process involving an egg beater. You can't patent Jacobian matrices ("abstract", see below) or an egg beater (it's prior art), but the process in context may be subject to claiming. Similarly, in deep learning the application context, your training methodology, your testing methodology, the finished product – all of these *may* be opportunities for protection at some level.

From a legal perspective, the most readily apparent challenges will be 35 USC 101 ("abstractness") and 35 USC 112 ("written description/enablement"). For the former, your claims will need to be sufficiently concrete so as to avoid being "wholly directed" to an abstract concept. For the latter, you'll need to go into sufficient detail so that it's clear the claims correspond to your invention and can be readily replicated. Incidentally, the former has already received much attention in the software context, but the latter has actually been subject to increased scrutiny in the courts as well recently. The USPTO is even issuing new **guidance** to the examiners regarding 112.

That said, whether patent protection is prudent or useful for you is a different matter. As evidenced by adversarial inputs, overfitting, etc. neural nets can be powerful but fickle creatures. That fickleness, coupled with the 101/112 requirements can make design-arounds "relatively" straightforward. "Oh, you claimed a pixel mapping to a convolutional layer up front? I'll just pre-process the data to avoid that initial edge mapping and then continue with the rest of your claim." That's a silly hypo, but you get the idea. A design-around may require some creativity and suffering, but deep-learning's fluidity usually lends itself to multiple approaches. Clever claim drafting can avoid some of this, but in such a rapidly moving technology it's hard to anticipate everything.

Also, depending upon the business posture, it may make more sense to keep the details a trade secret. Even if you go with a patent you'll be faced with the perennial machine learning dilemma of claiming your training or your testing setup. Do the former and you'll likely catch who you want, *but* their infringement may be difficult to detect. Do the latter and they'll be easier to detect, *but* you may be suing customers and arguing indirect/secondary infringement against your real target, which is at best awkward and at worst unsuccessful (and also still awkward). Obviously, you can do both, but the complementarity issue remains.

As always, this isn't legal advice (I'm speaking much too generically – any real-world situation is going to involve infinitely more considerations), but the idea that deep learning is somehow "magically" different from other machine learning setups in the IP sense should be dispelled. I think a lot of smart engineers intuitively suspect that the rote aspects of DL can't be patented – and they're right (in view of 101, 112, and prior art). But if you've done something creative *around* the core methodology (like a chef applying an egg beater to make bacon in a novel way [. . . use your imagination]), then the creative addition in combination and context could very well include worthwhile coverage.

PREVIOUS ARTICLE **NEXT ARTICLE**

No content on this website is intended, or should be construed, as legal advice. Rather, the content provided here merely serves to notify a wide audience of various issues' general character. If you have a legal question contact an attorney.