Anna's questions are in italics and my answers are in normal font. I am using the https://www.politifact.com/truth-o-meter/ scale (in bold) to rank the claims. I sometimes give brief answers followed by more details.

You and/or your group have:
-been working on theorem provers for nearly two decades.

Half true.

1. My work has been mostly on metasystems and combinations of ML (machine learning - inductive thinking) systems with TPs (theorem provers - deductive thinking systems), e.g. equipping TPs with ML guidance. This goes back to my MSc and PhD theses (1998/2004 - see my web and CV). Not much on standard (deduction-only/heuristic) TPs.
   More details: My main interest is strong AI for math/science. I believe the way to get there is to combine deductive thinking (TP) with inductive thinking (ML) - see my web page. So I have been interested in theorem proving but not much in standard deduction-only TPs. My first significant ML/TP experiments/datasets are from 2003 (https://doi.org/10.1007/s10817-004-6245-1) and my first significant ML/TP metasystem - MaLARea- that has demonstrated the power of combining ML & TP in rigorous competition evaluation (http://www.tptp.org/MPTPChallenge/) is from 2006 (http://ceur-ws.org/Vol-257/05_Urban.pdf). To have a rough idea of such combinations, see our recent ML/TP system (https://doi.org/10.4230/LIPIcs.ITP.2019.34) that improves a state-of-the-art TP by 70% on a large math corpus. It has about 10MB of compiled human-written TP code and about 50-100MB of a trained ML classifier that guides the prover. The 50-100MB are compressed decision trees trained on millions of examples extracted from the proofs.

2. Strictly speaking, there was no "group of mine" until 2015 when I got a large EU (ERC) project funding (http://ai4reason.org/) to work on this and to establish my group in Prague. **More details**: Before that, I was a postdoc in the Netherlands working mainly on my own and trying to engage various (typically singleton) collaborators in the topic. The largest success was engaging Cezary Kaliszyk from Innsbruck in 2011/12. We made good progress and we both got ERC funding in 2015/16 to establish our groups and further push the topic. Since 2015, I am indeed heading a growing group of researchers, have co-established the AITP conference in 2016 which in 2019 grew to about 80 people and have been on a number of program committees for related conferences. As a student I did co-establish a loose group (ARG) of (then) students interested in general AR (automated reasoning), attending an AR seminar/lecture in Prague. But many of them have not then worked on my topics, went abroad like me, etc.

-built theorem provers using machine learning tools which allow computers to learn on their own through experience.

Mostly true.

More precisely, we mostly equip pre-existing TPs with ML components that guide the proof search - thus modifying them. And we typically build AI/TP/ML metasystems.
More details: The ML guidance is just a component of the modified TP and the TP is not built from scratch using just ML. But the ML guidance is indeed a very significant component - see the above answer. It also works the other way round: the TP produces the data on which the ML guidance is trained, i.e., the TP is in some sense used to gradually "build" the trained ML system. So what we really build are AI/TP/ML metasystems that implement a virtuous loop that iterates between (1) guided theorem proving that produces more proofs (data) and (2) learning of the guidance from the data. This virtuous loop goes back (at least) to my first MaLARea system and it is used today in several other systems.

-been exploring the potential of using neural networks

True.

More details: Neural nets are one machine learning method of many that we explore. They have been around for a long time, but a lot of progress has been made in their training and design in the last 10-15 years especially in image and text processing. It is still very much an open and active research question to what extent can (existing or new) neural network architectures learn arbitrary programs and computational/semantic tasks relevant for math and TP. And how to combine neural algorithms with other (search, deductive, computational, symbolic, semantic, evolutionary, etc.) algorithms.

-layers of computations that help machines process information in a rough approximation of our brain's neuronal connections.

Mostly true.

This is indeed the rough initial analogy. It is probably quite imprecise and the architectures might differ a lot.

The difference is that neural networks train to make good guesses while conventional machine learning approaches typically search big datasets for useful, previously verified data.

Pants on fire.

Not just neural networks but many other ML approaches train to make good guesses. ML is not much about search (TP is!) but more about learning and generalizing from data. And there is nothing like a "conventional ML approach". See e.g. https://en.wikipedia.org/wiki/Supervised_learning#Approaches_and_algorithms for an overview. Such incorrect claims are usually spread by people who have either jumped quite recently on the deep learning bandwagon and lack a broader overview of the ML field or by people who are overhyping deep learning for all sorts of personal gains (funding, media attention, bonuses, etc).

In particular when learning symbolic tasks, there is often a danger of training deep neural networks that largely memorize rather than generalizing to nontrivial algorithms.

-recently reported on new conjectures generated by a theorem prover using advanced neural networks — similar to ones used in text-generation software.

Mostly false.

1. The conjectures described in my March'20 CICM submission (http://grid01.ciirc.cvut.cz/~mptp/nn_conj20/) are not really generated by a theorem prover. Not even by a TP equipped with a neural network. They are generated *only* by the neural network. But the neural network is trained (in four different ways) on theorem proving data.
2. Some of the tasks I use neural networks for in that work are not about suggesting conjectures, but about suggesting whole pieces of reasoning and human-oriented proofs.
3. Our work in this direction goes back at least five years - see the related work in my CICM paper. In particular, we have been playing with analogy-based conjecturing since ca 2015, and with the use of linguistic neural methods that can be used to suggest statements and proofs since ca 2018.

-been partially inspired by Andrej Karpathy, who a few years ago trained a neural network to generate mathematical-looking nonsense that looked legitimate to non-experts.

True.

See the link to his blog post in my paper.
More details: It nicely demonstrated that recurrent neural networks can practically without any modifications generate texts that capture a lot of statistical dependencies even in scientific writings. On the other hand, an expert looking at the example of a generated "proof" in my recent paper will see that the neural network has still very little idea about the meaning of the statements. And this is exactly the reason why it is interesting to critically evaluate such neural architectures in formal mathematics: It allows us to judge more clearly than on unstructured texts the capabilities and limits of the chosen learning architecture. And that in turn allows us to propose improvements and new architectures. A similar virtuous loop as the one in Malarea described above, but this time still concerning human rather than AI scientists. And that is also the reason why producing critical evaluations rather than hype is a necessary attribute of true science and why uncritical hype damages science.

-trained tool to recognize proofs using tens of thousands of theorems. (These were all within the mathematical library of Mizar, an ITP.)

Mostly false.

1. We typically train systems to *invent/find* new proofs, not to recognize existing ones. Recognizing whether something is a proof is an area called proof checking or proof verification/certification. We probably could try to train such tools, but it is a more mechanical and less interesting task than the hard AI task of finding a proof of a new conjecture.
2. We have developed such AI/TP/ML systems in many contexts, not just for Mizar (where the work goes back indeed to my 2004 PhD thesis). The list of systems and their libraries where we have done this includes also HOL Light, Isabelle, HOL4 and Coq. See my and Cezary's publications and the list of publications at ai4reason.org .
3. The number of proofs we train on can easily go over a million. See e.g. https://doi.org/10.1016/j.jsc.2014.09.032 or our IJCAR'20 talk about ENIGMA: https://youtu.be/XojOEpZfH4Y .

-used the network to generate new conjectures,

Mostly true.

See above. The neural networks are trained in four different ways and some of them suggest parts of reasoning and proofs rather than conjectures.

and checked the validity of those conjectures using an ATP called E.

Mostly true.

We have done a larger but still quite initial semantic evaluation based on E and some more AI/TP tools for one of the conjecturing tasks.

-the network proposed more than 50,000 new formulas, though tens of thousands were duplicates,

Mostly true.

The network can easily produce millions of predictions - the number is just a parameter. In the initial semantic evaluation, we asked it to produce ca 400000 predictions. This deduplicates to ca 200000 predictions. Each prediction typically consists of several formulas, but some predictions contain syntactically incorrect formulas and we do not process them further. About 50000 predictions contained at least one new (syntactically correct) conjecture. In total this produced ca 52000 such conjectures deduplicated to ca 33000.

-and many others were false.

Half true.

We do not say anything like that in the paper and we have not attempted any estimate. Failure to find a proof does not mean that the conjecture is false.

More details: It is usually pretty hard to show that a conjecture is true or false. We have tried to prove all the ca 52000 but not to disprove any. We can without much trouble automatically prove 9000-10000 of those, depending on how we try. This however does not imply at all that the remaining ones are false.

Of the few that looked mathematically interesting,

Mostly false.

We never say that only a few looked mathematically interesting. This is really hard to evaluate. See e.g. our previous paper https://doi.org/10.1016/j.jsc.2014.09.032 where we do quite a lot of work to see which of the millions of automatically generated lemmas may be useful. Here we just show some interesting true and false conjectures.

-the system wasn't able to generate a convincing proof.

Half true.

We say: "It seems that we are not yet capable of proving the more interesting conjectures". We never say how many of the ones we could not prove are interesting - see above.

Side note: If an ATP finds a proof, it is always 100% convincing. While we do equip ATPs with intuition provided by ML, we never compromise the ATP's rigorous logical foundations. We can easily verify all ATP proofs with relatively simple tools. This certainty is one of the main achievements of symbolic logic and theorem proving.