AI & Big Data Expo 2019, Santa Clara

main questions:

how is AI used in context of a company? is it always ML?

how do people come up with enterprise business ideas? what are some interesting examples?

when will we reach the AI winter/hit a wall with this innovation?

General notes: most people coming to these conference are execs looking to use AI in their business more efficiently, who don't really understand the technical nitty-gritty, but want to stay up with the current buzzword trends. AI, generally, is defined as getting a computer to execute a task without explicit programming it to do that task (top down learning via bottom up instruction).

Conversational AI

NLP is harder than vision because images are inherently numeric, and each pixel is independent (parallel). With language, you have to create numerical patterns from text, and this theory of how letters relate to each other is where linguistics come into play. You can find this analogy in the human brain, where kids know images instinctively, but language is a process that took millennia to evolve as a society, and kids have to be taught it. It's also non-deterministic—one sentence can have several meanings depending on more factors than just the textual input, whereas on image is correctly classified only one way.

Useful when it comes to product recommendations because even a small increase in ROI at industrial scale can be seen as an excellent move. For example, Levi's would always hit a 3rd party API to get recommendations based on a user's browser data, but that took 24 hours to update, and so you'd update their recommendations *next time* they visited the page, but you wanted it to be update in real time, and so they decided to build out their own ML infrastructure for this.

When they decided to do this, they had to submit docs outlining predicted costs, timelines, architectural planning, ROI, and more—and passing through these administrative hoops takes time, too. When they unveiled how they did it in the end, they had to use 10 services, from DynamoDB (part of AWS) to Redis and more—it was a complex architecture for a seemingly simple output. All of this work is very sophisticated, involving advanced mathematics even, but the question of whether using your life to optimise Jean recommendations is "meaningful" is still an open one.

There's a lot more nuance going into these things than they let on. They take the optimal recommendations, and hardcode a black tee in if all the recs are white simply for colour contrast to draw the user's eye to the recommended items section so that they're more likely to shop more. That said, getting some cool sweatpants recommended to me on Instagram that make me happy when I wear them *is* contributing genuine value to society. And all of this complex architecture is just what someone at *Levi's* is working on—imagine the technical sophistication of the work done by Spotify/Netflix/Amazon!

Jobs said "You have to start with the consumer experience and them move backwards to the technology", and so I'm right in that my problem based investigation is correct, as opposed to forcing ML down the throat of anything that will listen.

AI for social good

Turing is a company founded to recruit technical talent from around the world. Hiring is riddled with biases, and algorithmic matching can remove some of that (weighing how you want the algorithm to match is biased in itself). You have to survey problem spaces, *and then identify world-changing problems that will align with economic incentives*. Sourcing good talent from around the world hits both criteria.

There is a phrase called "paving the cow path" which mocks how the lead cow of the pack walks a winding, inefficient path, but all the other cows just blindly follow, and then the farmer paves the path thinking it to be the correct, optimal path back to the barn. This reminds me of how Tomasi says that as humans, we're very good at optimising narrow objective functions, but not good at integrating these local solutions into a coherent, functional solution to a complex, multivariate problem in the form of a global solution. This is also why choosing a utility function—choosing *what* you want your model to optimise for, is just as important as the robustness of the model itself.

AI is also used widely in banking and financial services—lots of tech companies are moving into fintech, like Uber with their credit card. Business facing AI is being created that scrapes records of a particular user's past interactions, and outputs a probability of them completing a transaction, which informs how much the business should invest in that user.

On going to grad school to work in industry

For almost all companies, even AI positions at companies like GreyParrot or Cruise, you don't need a PhD—only for research positions at FAIR, Google Brain, IBM Research, etc., and there's only around 200 of those jobs in existence in the first place, so think very carefully about grad school—why you want to go or not. People in industry generally don't really respect PhDs as much as you might think. Only do one if you're intent on being an academic, or are dying to explore a certain topic, he says.

Enterprise AI

Defined loosely as using Ai (usually ML) to improve productivity internally or improve operational efficiency of a large business. In fact, many big companies use AI without knowing it—they pay for an external tool (like Salesforce CRM tool) that usually involves a lot of heavy duty ML on the software's server. Examples are google minimising data centre energy expenditure, Ocado using robots to structure their warehouses, Rolls Royce monitoring the plane engines they sell in-situ to look for faults, or automating customer service/support using NLP/voice. In the future, having the data being collected to where it will be computed on will be key because moving vast amounts of data from place to place, especially out of a cloud environment, is a pain in the ass.

Chevron (oil/gas) data scientist talk—Day to day of a data scientist:

30-40ML engineers at Chevron, surprisingly, they comb through and look at the data collected by sensors fit onto things like oil pipes, to predict when they will fail to avoid being sued for environmental damages by the government, compared to data scientists at FANG, who interpre results of A/B tests for new features. It's important to know that data science can demand a lot of domain-specific knowledge about the data you're analysing to get at meaningful insights (eg oil physics/fluid mechanics to meaningfully interpret chevron sensor metrics), which is why some believe that there is no future for data scientists, and it'll just be a skill that professionals in any industry go on to learn and pick up, much like Excel.

Enterprise Vision

Retail—replacing barcodes, theft prevention (last year alone, $14B was lost to cashiers sweet-hearting people they know!), automated delivery, personalised service, and more. Offering AI to enterprises so they can create these services is what an enterprise AI company would do.

Industry—worker safety to check when standards are being followed, inspection—currently literal people go walking around planes as they land to check for dents, watching video tape of a factory/plant to look for mechanical failures gets boring so people get bad and QA is a huge application.

A corollary is that we often have unreasonably high expectations of technology. Technical solutions don't need to be *perfect*, just better than existing best solutions, which they are. We often forgive humans for having a 10% error rate, but when a computer has a 1% error rate, we rage about how it's a bad replacement.

AI in retail more—another use case is Walmart recommending you a specific ice-cream flavour if the one you requested is out of stock. And all the features you see in any of these shopping/industrial sites is heavily tracked in terms of metrics—they A/B test everything, and only ship the changes that are proven to be effective.

Interesting to note that recommendations also vary with culture. For example, Levi's does an analysis to find that Brazilians treat their brand as more aspirational than Londoners, and so change the AI algorithms in that region to make recommendations accordingly. They also change the recommendations people get with geography, weather, and more, to really get their best shot at convincing you to buy something you might not otherwise. And there are, of course, teams of people trying to reverse-engineer these algorithms to get better product placement for their items.

In enterprise AI, most companies' products are very technical and niche—you'd only ever encounter the problem firsthand by working in the company/industry for a long time and witnessing these problems. You'd need to have done a lot of ML engineering to know the kind of problems ML engineers have and this is part of the reason that the Bay Area is so far ahead—it's to technically talented that they can spot and fix niche developer problems and really optimise workflows. You need to find a way to reconcile the difference between climate engineering-esque moonshots and Infoworks-esque companies that help with large scale data migration to the cloud for big companies. Entrepreneurship is really about the latter, not the former.

There are data teams at every company that are in charge of interpreting user data to inform decisions made by the company. For example, GoPro collects data on what sort of pictures users take, and use that to inform what they should invest money into improving, which involve image classification.

We're ending the era where simply appending .ai to your domain name adds $10M to your valuation. All of this hype came from a 2012 ImageNet paper which was breakthrough, and all other fields of AI benefitted similarly, coupled with improvements in cloud data storage to train models, and GPUs to make ML at scale feasible—a combination of factors that led to big companies identifying this insight, which then caused the whole industry to blow up with interest.

Conversation with Peter, the ML engineer:

Government contracting:

He worked at a company in Virginia that did a lot of government contracting. The government would send them images it was interested in, and they'd process them and tell the government whether they were doctored in any way or not. The way it works is that the government puts out hundreds of "SBIRs" or "calls to action", listing problems it has, and then private companies, including startups as well as giants like Boeing, Lockheed Martin, vie for those spots. The acquisition by the bigger company came about because the bigger company had contracted with the startup before, and it was impressed by their performance, as well as the fact that they were geographically next door meant that executives cross pollinated a lot. When they do this service, server costs can quickly ramp up because they are moving petabytes of data around (mainly files to feed into CLI to train their ML models).

Day to day:

Does a literature review of current updates on ML/vision mainly by scanning through Twitter to see papers that people recommended (ArXiv is far too messy and scattered). Then spends around 2/3 of his time cleaning data that the data team handed to him to train his models—doing things like writing scripts to flip images that are upside down, etc. This is important because the models are only as good as the variety and quantity of the data used to train them.

Then he spends some time building the models—writing code in his editor by important TensorFlow or PyTorch, which is literally model = init.model.neuralneut(12), then model.add(conv2D(32)) and things like that to add layers of nodes to the neural network, with all of the mathematical heavy lifting and linear algebra abstracted away by the library. The crucial insight is that linear algebra is to the day-to-day of a ML engineer what studying operating systems is to a software engineer. It's useful to have studied—and a good understanding separates the good from the great, but it's not important to have to MVP things and prototype. You could totally build out a model to classify dogs using a few lines of code and not understanding any linear algebra at all, just like you can build out a functional web app without knowing how to implement an HTTP request.

All the training of models (the process of feeding the training data in and adjusting the matrices/functions at the beginning—backpropogation) is automated so that you just go to the command line and type model.train(data) and the computer handles the rest.

The actual math of a perceptron (vanilla neural network) is that you take in the image data (binary/RGB values) and then multiply it by a matrix, reducing the size from, say, 4x4 to, say, 2x2, and continue until you're at a certain value. The first time your output value will be garbage, but then the ML framework will go back and adjust the initial matrix's weights such that the output is now 10% closer to the correct output you expected to have, and you repeat with a large, varied data set to get a robust model, which, when you pass a new image into, will be classified correctly. Each "layer"—that is, each matrix you multiply by/function, is at a higher layer of abstraction, whereby you go from 128 nodes all the way down to 4, but the 128 are lines and circle, and the 4 are eyes and arms (composed of many lines and circles).

DoD employee notes:

He was sent to the conference and had it paid for him, and he has to go back and present all his findings and what he learned since the government is too stingy to send multiple delgates.

Why do people work for the government? Primarily because of 1) stability 2) prestige, on a local level and 3) allure of working on classified projects.

You can apply for internships with the DoD to get your security clearance (poly, background check, etc.) which can be an asset in both reapplying and applying to other companies (some teams at MSFT, for example, demand clearance).

Interesting to note that most of his work was classified—he couldn't talk about anything. And it isn't as if they ramp you up—once he got his clearance, he was given all the classified information at once, hours after signing the NDA (but there are in fact layers of clearance, of course). There is, however, lots of red-tap and you've got to know the right people internally to push projects through, and so in that sense a data scientist for the DoD is somewhat like working at big companies where it's the same.

That said, the cool part is that much of the tech that they use really does seem like it's straight out of sci-fi—he said working there was like "being told, okay, now you see how the world *really works*) so it's likely that all the conspiracy theorist-esque concepts of how powerful and conniving the government is have some degree of truth to them.

InfoWorks

what it does

moving legacy data onto cloud infrastructure is hard. big cloud providers don't make it easy, and so you need to hire lots of data engineers who need to write lots of code to move data from legacy hardware onto the cloud. this is only a problem for large companies, as the difficulty of migrating data from, say, a SQL database, becomes exponentially harder with the amount of data you have. this company makes a platform that allows non-engineers to use a GUI to move whatever data they want onto their cloud provider of choice. they therefore sell to big businesses that are already using cloud providers, not to the cloud providers or consumers themselves. this is an example of enterprise AI, since this migration, and other services they offer, make use of AI/ML to operate a business-facing service.

how they came up with it?

founder helped set up Google Cloud, and witnessed the deficiencies in the operations of a large cloud provider firsthand, and saw the opportunity to make a business to help legacy business migrate to the cloud at scale in an easy-to-use way.

Pachyderm

what it does

"Git for data science". essentially a version control system for machine learning/data engineering models (code) as well as the training data associated with it. Right now, each company has a different way to build ML models/do data analytics. you can't use GitHub because data science involves lots of training files and thus lots of data—GitHub has a file/size limit. so using Pachyderm, you can get access to versions of all your data (done through diffing changes as opposed to snapshots, like most companies currently do). if someone fucks up a model and it ships, you can now just see all the versions of it, merge a change, and ship it to production, whereas before this Git capability wouldn't work with data science/ML. there is also lots of ML involved in making this offering, which is a business-facing offering, so this is classified as an enterprise AI company. so enterprise AI is not necessarily AI *in* enterprise (like Netflix recommendations), but instead it's using AI to make an offering to an enterprise solve their problem.

Northwest Towers

what it does

in rural areas, cellular communication is weak. and big telecom companies like AT&T don't have much of a financial incentive to fix it since they don't have many customers in those areas. however, there are a lot of big business like oil/gas drilling, mining, etc. that operate in rural areas and need high bandwidth network infrastructure for their IoT needs and for general communication. this company orders all the separate hardware, assembles it into whatever is needed for the networking needs of their client (eg 200Mb/s in a certain mine of a certain shape at a certain time) and installs all the hardware infrastructure to allow that data communication—most of which comes from sensor data and video feed (eg. video feed of a mine precision drill being operated remotely/ temperature data in an oil reservoir). these guys sell the hardware/infrastructure/installation as opposed to allow others to "rent" it like the big telecoms (where paying per megabyte is infeasible when you're using terabytes of data per day). another use case is in harbours, where the local dock owners have to park the ships by law, not the ship drivers, around the world—you need good network connection between the ship and the harbour, which these guys can install.

Dataforest

what it does

basically Notitia. crowdsources training data for AI models from the public. if you're the first to make a request for certain data (say, squirrel pictures) and then that data is used again in the future, you make a passive income ("royalties") for it, despite being a client. they crowdsource the labelling as well, and have an internal team validate it. started the company at university in turkey, and is moving to the Bay Area as it's beginning to take off.

how he came up with it

interested in CS at startups, though about which field is taking off/attracting the most investment, bet on AI only growing in the future, and so brainstormed what companies you could make for AI/ data science.

Qlik

what it does

data visualisation in RAM, fundamental technological innovation because the ability to run these programs in RAM fundamentally didn't exist until a few years ago. you can interact with the data in real time as a consequence, as opposed to printing out long sheets of pre-pared pdfs with lots of charts that you had made beforehand.

how he came up with it

he was a consultant working with a swedish company that had a special situation where they needed to see data insights and manipulate data in real time, and the founders had a technical background and so decided to implement it in memory.

Cambridge Quantum Computing

what it does

hardware—using quantum entanglement to generate truly random numbers (provably so), that are sold to any company that uses encryption (governments/finance mostly)

software—

compiler; you learn the QC language they wrote "ticket" and then you can write programs to use quantum computers in the same way that you use normal classical programming languages and run programs on a QC that way

chemistry; a chemical modelling software that is able to make more sophisticated and complex models for things like drug modelling/chemical testing, since it's an inherently quantum system it can simulate other quantum systems

spoke to Mark Jackson—PhD in string theory under Briane Greene, did a few post-docs, and then transferred to industry, living in Berkeley—reach out to ask him about working in a research-based environment in industry as well as slowly finessing an internship to work at this company

how he came up with it

worked under Stephen Hawking, who told him personally that quantum physics will change computing forever, and that it must be paid careful attention to, and he established a company to look into it (research-based)