introduction to swift for tensorflow

brettkoonce.com/talks

june 5th, 2019

overview

- purpose: convince you to use swift for tensorflow
- why tensorflow + swift
- where things are going: cloud + edge
- what we bring to the table
- how to get started, recap

historical context

why tensorflow?

- gpu >> cpu: simd, cores, memory
- cambrian explosion of frameworks
- tensorflow (c++) —> keras (python)
- google: resources, research, engineering

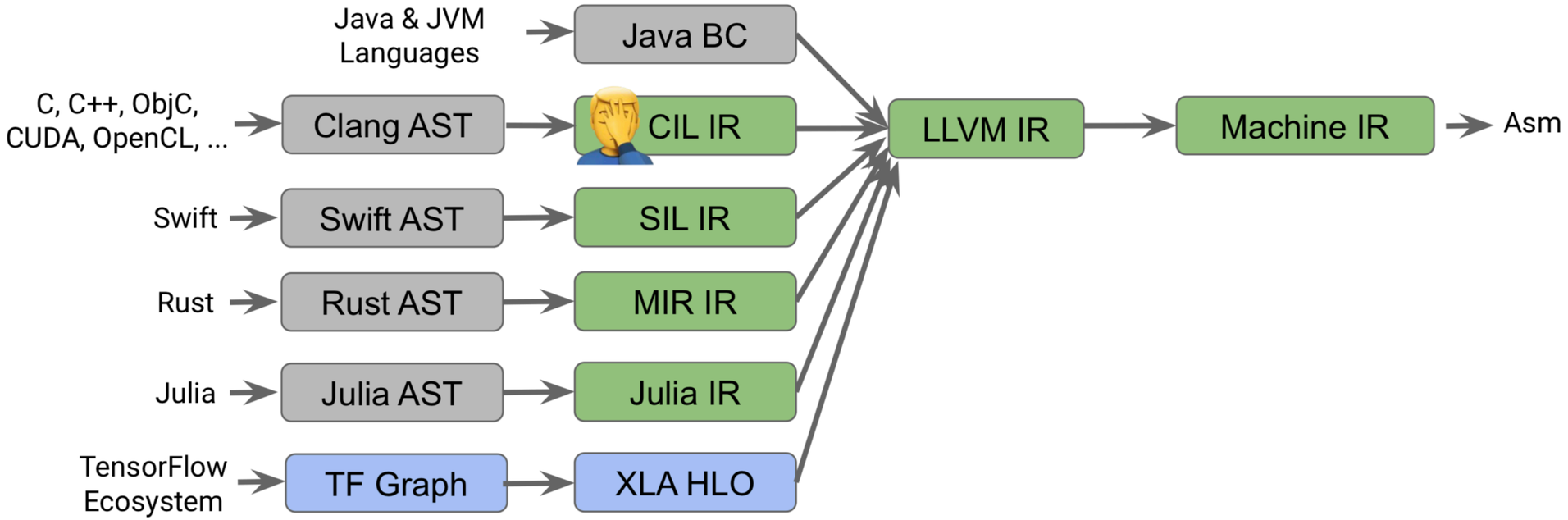
why swift?

- nextstep —> smalltalk —> obj-c —> gcc
- llvm —> memory, thread safety —> swift
- pragmatic language based on existing paradigms, work with existing code
- functional programming gateway

the next year

- pytorch —> pytorchir/jit —> cuda
- javascript —> types —> assembly
- julia, tensorflow 2 —> xla —> mlir —> device
- targets: cpu, gpu, tpu, phone, new chips

TensorFlow XLA Compiler



- Domain specific optimizations, progressive lowering

Google

"XLA Overview": https://tensorflow.org/xla/overview (video overview)

the future


LET'S GO DOWN THE HILL AND SEE IF WE CAN TRAVEL INTO THE FUTURE.
GO INTO THE FUTURE? HOW?
IT'S EASY! ALL WE HAVE TO DO IS GET GOING REAL FAST AND WE'LL TIME-WARP!
HA HA! FASTER! FASTER!
GOSH, WHAT DO YOU SUPPOSE THE FUTURE WILL BE LIKE?
WHO KNOWS? FLYING CARS AND CITIES BUILT ON CLOUDS, MAYBE!
© Universal Press Syndicate
JUST THINK OF ALL THE WEIRD THINGS WE CAN TELL PEOPLE WE SAW! OH BOY!
HEY, WE'RE AT THE BOTTOM OF THE HILL. I DIDN'T FEEL ANY TIME WARP, DID YOU?
NOPE.
BUT LOOK! IT'S TWO MINUTES LATER THAN WHEN WE STARTED! WE'RE IN THE FUTURE!!
HMM...THINGS HAVEN'T IMPROVED. I'M DISAPPOINTED.
WATTERSON 4-20

cloud computing

- large clusters, abstracted away, on demand
- tools are being commoditized
- how to use infinite cpu/gpu/data/network —> developer’s problem
- proliferation of tools to deal with problems at scale

big data/compute

- having lots of data simplifies algorithm design
- large scale reinforcement learning, simulations to generate data
- machine learning paradigm is many single-threaded jobs
- knowing the right path is key, not size

what is ai?

- rl ⊂ dl ⊂ ml ⊂ ai ⊂ 🤓 ⊂ universe
- things humans can/cannot do?
- things computers can/cannot do?
- something hard to do == ai —>
- something not hard to do == not ai

“math” on the edge

- once we solve a problem, reduce solution to the smallest possible algorithm/data
- move hardware/software as close as possible to user to reduce latency
- mobile phone will remain the ultimate platform to solve problems whenever, wherever, however user wants

what makes us unique

limits —> creativity

- we build entire worlds with tiny amounts of resources
- work with hardware directly
- create illusion of performance
- element of play, magic

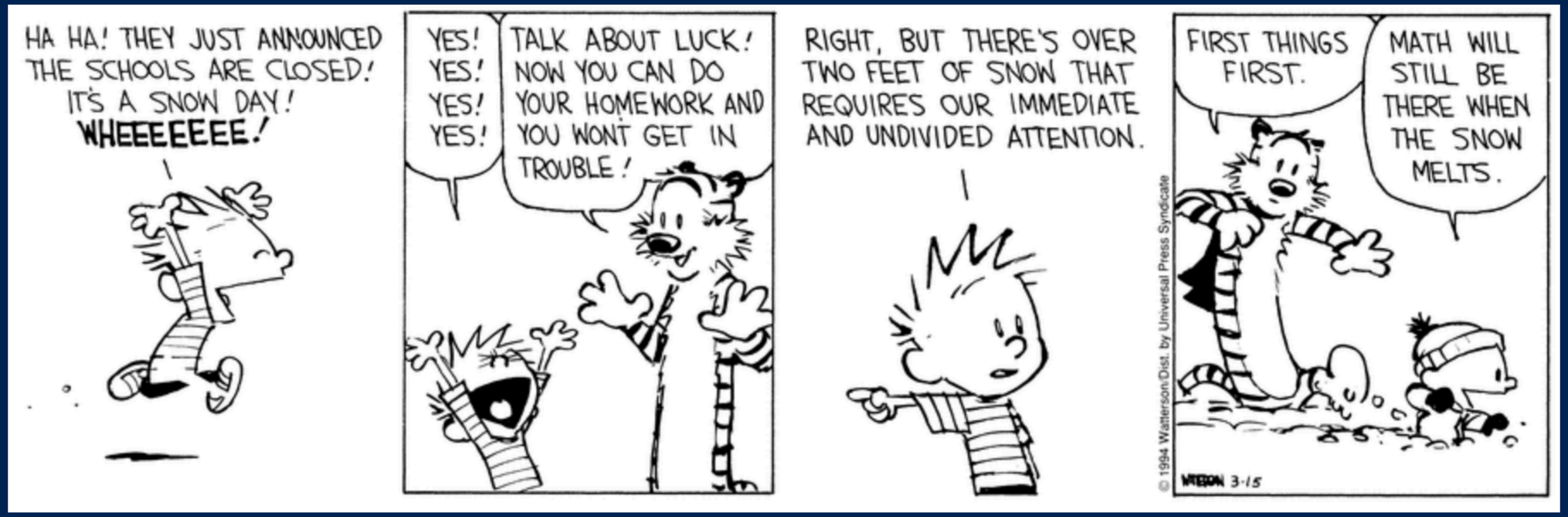
fast followers

- don’t need to reinvent wheel
- take cloud/desktop solutions, convert them to mobile
- demo: pose detection, segmentation
- think different, reexamine paradigms

tools you need

- functional programming*
- math: basic calculus/linear algebra
- intuition about users*, statistics
- willingness to experiment, patience*

how to ~~master~~ get started with deep learning in ~~21~~ 365 days



hello world

- colab + python/swift —> free gpu notebooks
- mnist demo
- concepts: code, library, gpu, output
- fast.ai 2018 notebooks (pytorch)

explore the frontier

- fast.ai 2019 notebooks: imagenette
- swift-models: mnist demo
- build s4tf from source —> make changes —> make a pr
- start reading papers, get out into real world, share your knowledge

“If you want to build a ship, don’t drum up the men to gather wood, divide the work, and give orders. Instead, teach them to yearn for the vast and endless sea.”

- **Antoine de Saint-Exupéry (sort of)**

inflection point

- deep learning is the next evolution of hardware/software
- ready-made opportunities like this are rare
- help define the future of machine learning, make world move faster

thanks for coming!