10/18/19

Computer Architecture & Machine Learning: A tale of two computing paradigms

* PolyArch Research Group
* CS 33: Computer Organization – architecture + OS + low-level prog)
* CS251a, CS259
* CS 33: Eggert (lectures, depth), Reinmann (clear, easier grade)
* Architecture
* Pre-1964: machine (w/ mass) – hardware, (no mass) – software
* Physics/ Manufacturing -> Devices (Transistors) -> circuit -> component -> hardware org
* Algorithm -> application -> programming language -> compiler -> operating system
* Require an interface between hardware and software (architecture – bridge between both)
  + Different machines with same software – x86
* Ingredients of architecture
  + Memory: store values (state, variables, etc.)
  + Instructions one state to another
  + Program: set of instructions
  + Execution: when do excute each instruction
* Von Neuman Execution – common
  + Instructions executed sequentially – defined by a program counter
* ISA: hardware – division of works
  + Intel P6 – one chip with one professor
  + Fetch, decode, execture. writeback
* Trends
  + General purpose CPU -> deep learning
  + Machine Learning
    - Use of data to train a function
      * Define the form of a function
        + Linear function
      * Observation the output from a function to the correct one
        + Backpropagating
    - Layer: one of the volumes
    - Neuron: one element of the volume
      * Linear function of inputs
    - Synapse: connection neurons in different layers
      * values the slopes of the data
* Training – regression (layers of it)
* Neuron: multiply the inputs by weights -> transfer function -> net input
* Deep learning:: linear algebra – input layer -> metrics multiplying
* Google TPU Processor ~2014 -> software company making hardware
  + Speech recognition -> matrix multiply
    - DDR3 DRAM -> weight fetcher -> matrix -> accumulator
* Deep Learning
  + Accelerator DianNao (2014) – inspiring
  + Explosion of researchs
  + Co-optimize for deep learning ~ domain specific architecture +startup
* General
  + 2014 – Intel Haswell
  + 2019 – Intel Ice Lake -> same architecture
* Paradox: general – stagnating, machine learning – thriving
* No longer technology free ride
  + Scaling general harder than scaling linear-algebra
* Transistors -> more efficient – complication increases exponentially
  + Unable to run professors – thermal limit
  + Performance -> stagnating
* Conclusion: using ML architectures to build better general-purpose processors
  + Co-optimize- computational patterns

PPT – Seminar 10/23/19

Tony Nowatzki – Computer Architecture & Machine Learning

* CS33, CS251a, CS259

Computer Architecture – bridge between hardware and software

- Ingredients of architecture: memory, instructions, program, execution model

- Von Neuman execution: run executions sequentially – defined by counter (branch)

ISA: compile codes into machine language instructions

- Software: pull from memory to memory – instructions clear

- Hardware: layout of modules (fetch, decode, execute, writeback)

-> Fetch (program counter) -> address instruction (cache) -> decode from register files -> writeback

- ISA: abstract hardware 0> software stack simpler (Von Neumann ISA)

-- Process pipline

Trends

- General purpose CPU vs Deep Learning

-- Function that learn (complication) -> use data to train

--- Approach 1: define form of function – linear function ~ stacking

--- Approach 2: backpropagation (estimation result -> tuning the error)

Matric Computation – linear function of inputs (from other neurons)

-> change input layer by layer -> output -> regression = training (layers)

-> Each neuron: multiply accumulate – non-linear function

~Linear algebra: input layer -> hidden layer w/ weighted edge

Specialized Architecture

- Google TPU: including a matrix multiply unit – from general: bottlenecked from execution stage (operation result)

->Processor: Deep Learning – specialized CPU -> fast linear algebra computation

-> Machine learning processor: co-optimize for deep learning -> domain specific architecture (specific hardware with more efficient coding)

Stagnation of modern general purpose CPU compare of thriving of machine learning proc.

-> no longer technology free ride

->scaling general purpose is harder than linear-algebra architecture

General: computation: FP + INT ~9% of power -> power overhead reduces multicore calc

~ Extract parallelism out of single thread – instruction level -> difficult

Machine learning: scaling -> maybe use it to improve general processing CPU