Liverpool University - Stepping to Hybrid HPC. The Barkla Cluster & Cloud

Cliff Addison and Wil Mayers

UNIVERSITY OF LIVERPOOL

Overview

- Liverpool system, Barkla
  - Procurement, hardware, user buy in, Dell / Alces and AWS tie-in
- Liverpool use cases on AWS in 2018
- Lessons learned

- Future ideas – why it is cloud services and not just cloud hardware

UNIVERSITY OF LIVERPOOL

One page procurement overview

- Barkla purchased and commissioned in 2017.
- Dell collaborated with Alces and AWS to win.
- Backbone of 96 nodes with twin 6138 processors and 384GB memory.
- Strong researcher buy in – 67 of now 111 nodes.

Charles Glover Barkla
Nobel Prize in Physics, 1917
for X-ray spectroscopy

UNIVERSITY OF LIVERPOOL

Liverpool tendered for new HPC in 2017

- Essential to see tangible research impact.
- University wanted new system to align with University strategic objectives.
- Doubling of system size planned Q3 2018 (now Q3 2019)
- Additionally, tender panel wanted to get basics correct:
  - Modern core hardware and software infrastructure
  - Support for new / non-traditional HPC users (e.g. Deep Learning)
  - Make it easier to use the cluster to analyse data
  - Reduce complexity of using / managing the cluster
  - Provide cost effective basis for researcher hardware

And the winner was...

- Dell EMC / Alces with some strong collaboration and funding from Amazon Web Services (AWS).
- Winning factors from the Panel's point of view:
  - Excellent amount of hardware available.
  - Addressed all of the core and most of the highly desirable parts of the tender.
  - Brought in AWS sponsorship and financial support.
  - Presented a coherent hardware / software offering that would provide an excellent basis for the future.

DELL EMC

alcesflight

amazon web services

UNIVERSITY OF LIVERPOOL

Hardware details

- Compute nodes consist mainly of C6400 enclosures each with 4 C6420 sleds (nodes) – now 111 of these
  - Primary nodes: twin SP 6138 processors (20 cores, 2GHz), 384 GB memory, 960 GB SSD
- Supporting nodes include:
  - C4130 GPU node with quad p100s and dual E5-2650 v4 processors (quad v100 node on order)
  - Two large memory nodes: R640 servers with SP 6138 processors and 1152 GB memory
  - C6400 enclosure with 4 C6320P sleds with Xeon Phi-7230 processors, 192 GB memory
  - NFS and Lustre file systems based around PowerVault MD3060e, MD3460, and MD3420 storage shelves. (So 360 TB Lustre and 500TB of NFS storage)

Research Groups

- Main Barkla research groups (top 25 users) come from:
  - Chemistry – 3 groups bought 46 nodes
  - Engineering – 2 groups bought 13 nodes
  - Mathematical Sciences – 1 group bought 8 nodes
  - Ocean Modelling – looking to buy nodes on next expansion
  - Electrical Engineering – diverse hardware interests
- Main nodes busy most days (although some spare cores)
- Most bioinformatics work takes place on our Bull (SandyBridge) cluster and Windows Condor pool.

Software and cloud details

- System software and support via Alces Software.
  - SLURM is the job scheduler
  - Liverpool manage the scheduler,
  - Alces manage user creation, system support and maintenance
- Environment nearly identical to the Alces Flight environment available on several public clouds.
- AWS provided some research credits to experiment with bursting and other cloud related work.

Why was AWS involvement so attractive?

- Alces Flight provided a nearly seamless route onto the cloud from our cluster.

- AWS makes it easier to provide non-conventional compute to new users.

- Research credits encouraged experimentation without fear.

Expected cloud scenarios

- Cloud bursting – more cycles needed for a short period
  - typically for papers or presentations
- Specialised software environment
  - Hadoop, other Big Data
- High throughput workflows
  - Current Windows Condor pool limited to circa 8 hr jobs
- Scoping studies
  - I think I need X cores and Y GB of memory for my research
- GPU nodes for Deep Learning
- **Avoiding** large data transfers in this first instance

Cloud bursting - 1

- An existing Condor pool can be extended easily to the cloud.
  - Users just request the cloud resource on local Condor server – acts as scheduler.
  - Customise a standard AWS Linux image with necessary extra software and then save this image so is ready to go.
  - Have an in-cloud manager that deploys compute images; liaison with scheduler.
  - Spot market makes the compute even more cost-effective
    - Fits perfectly with Condor cycle stealing idea

- Test that target instances are good enough
  Micro instances may be too slow so more expensive for compute

How this can work…

- Researcher came to us in May with an urgent request to run 100,000 simulations related to a paper under review.
  - Our AWS Condor pool ideal.
  - Cost per simulation cheapest on t2.medium, but fastest on c4.large or c5.large.

Paper resubmitted on time!

- Final set up:
  - 1000 jobs with 100 simulations each, pool size of 400 (so 400 jobs at once) completed task in **7h 21m**. Serially would need about **98 days** – massive speed-up.
  - Price **$51.16**

UNIVERSITY OF LIVERPOOL

Bit more on this research

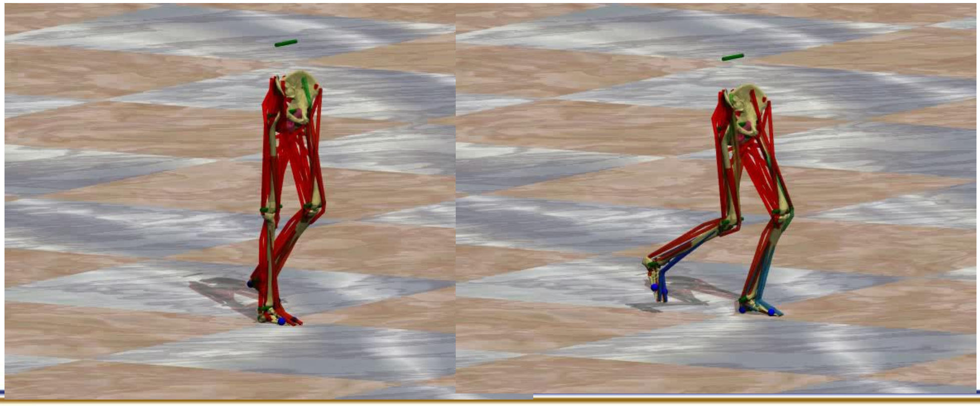
- Quick pitch for the sort of research going on in the UK.
- Bluetongue is a potentially devastating disease of cattle and sheep.
- UK outbreak in 2007 was not as severe as it could have been. Why???
- Epidemiologists modelled disease spread scenarios looking at temperature, farm density and foot and mouth disease movement restrictions.
- All had an impact on reducing the spread of this disease.

Cloud bursting - 2

- Alces Flight can spin up compute clusters with a familiar environment fairly painlessly.
- Main login node and storage are from on-demand instances so they are always there.
- Compute nodes can be spot instances and Alces Flight provides autoscaling so compute nodes do not sit idly waiting for work.
- Users can get their own private cluster or a cluster that a group can use with shared storage and login node.
- Some AWS instances now support fast MPI!

How this can work…

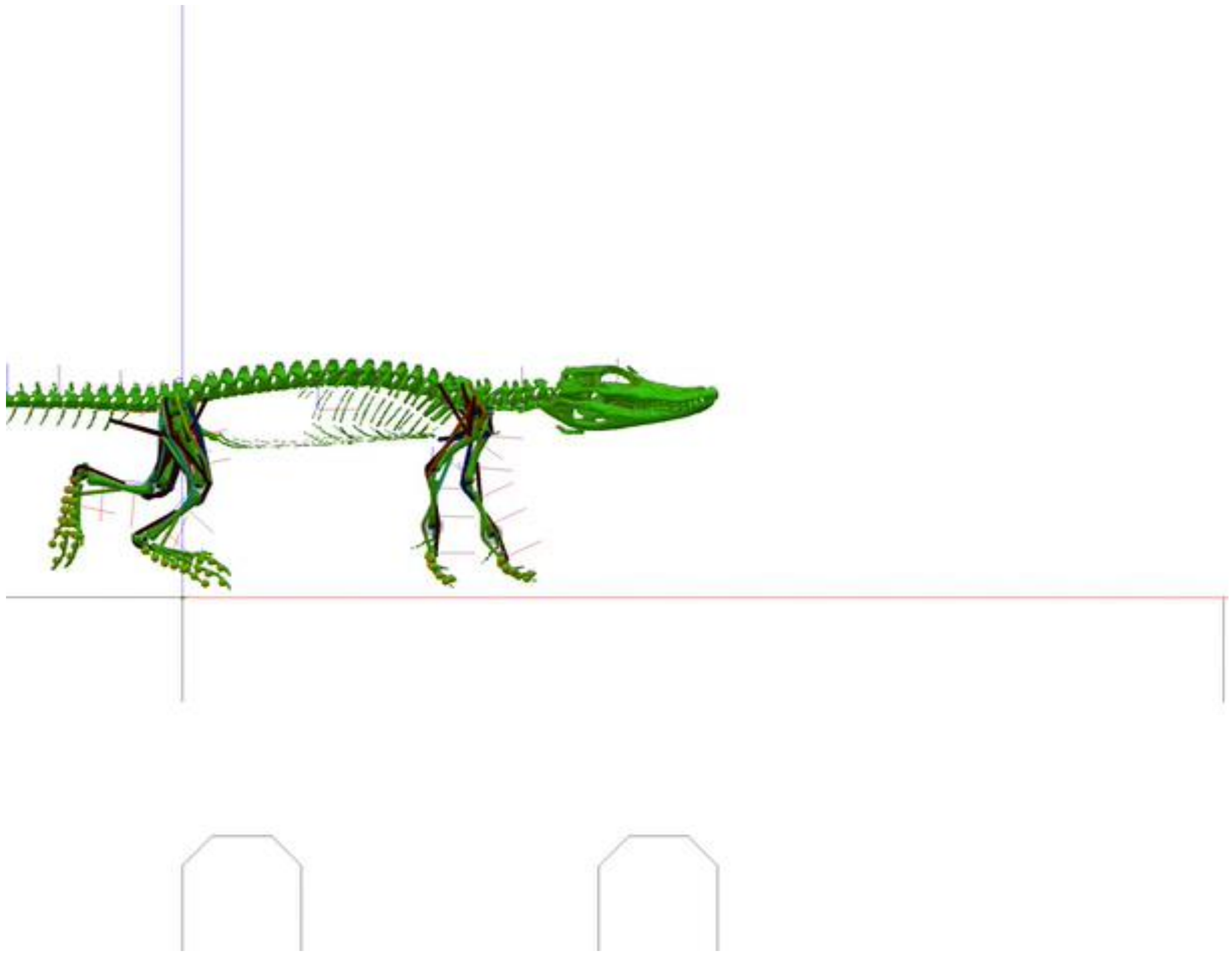
- Collaborative group at Liverpool / Manchester have a scalable code for gait analysis – starts from skeleton.
- Uses a client / server model – modest interconnect demands.
- Their demand for compute is open-ended.
- Experiment - Can they use several hundred cores over a couple of weeks to get something useful produced?



Results…

- Exposed the reality of doing experiments…
- Overjoyed at having their own cluster on the cloud
  - Ideal environment for development and testing
- Unfortunately, some bugs were revealed in testing the new animal model, so not as much useful computation took place as desired.
- Results provided pilot data and basic validation figures for a grant proposal
- Circa 6230 instance hours were used
- **But** we showed Alces Flight could bring a lot of compute to bear easily and quickly.

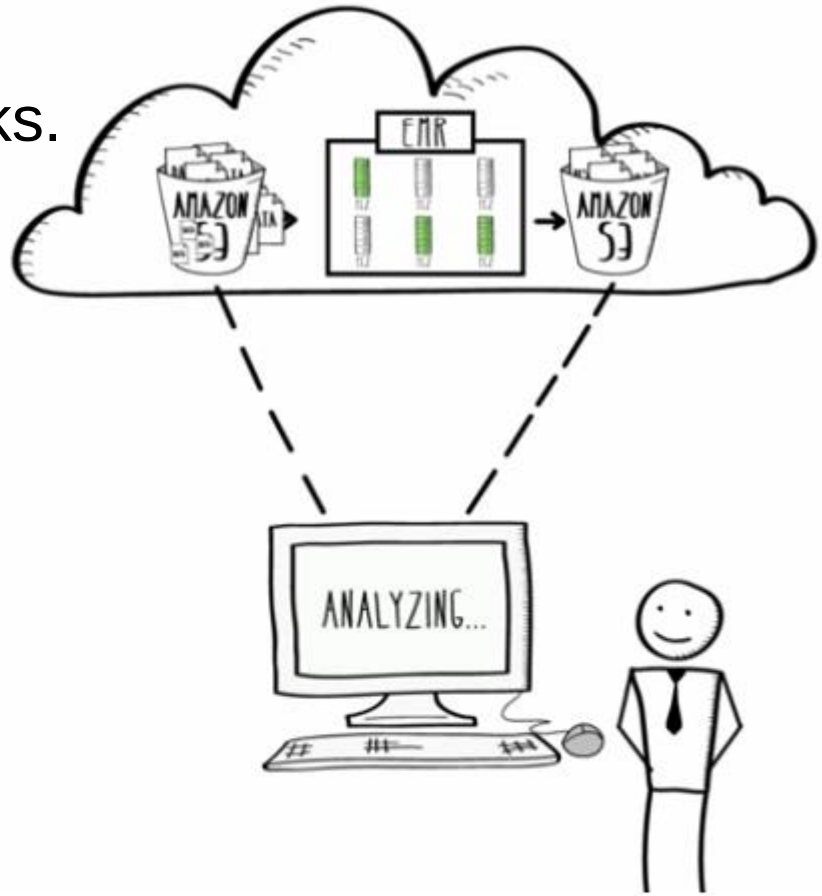
Teaching an alligator to walk has issues



UNIVERSITY OF LIVERPOOL

Specialised software - 1

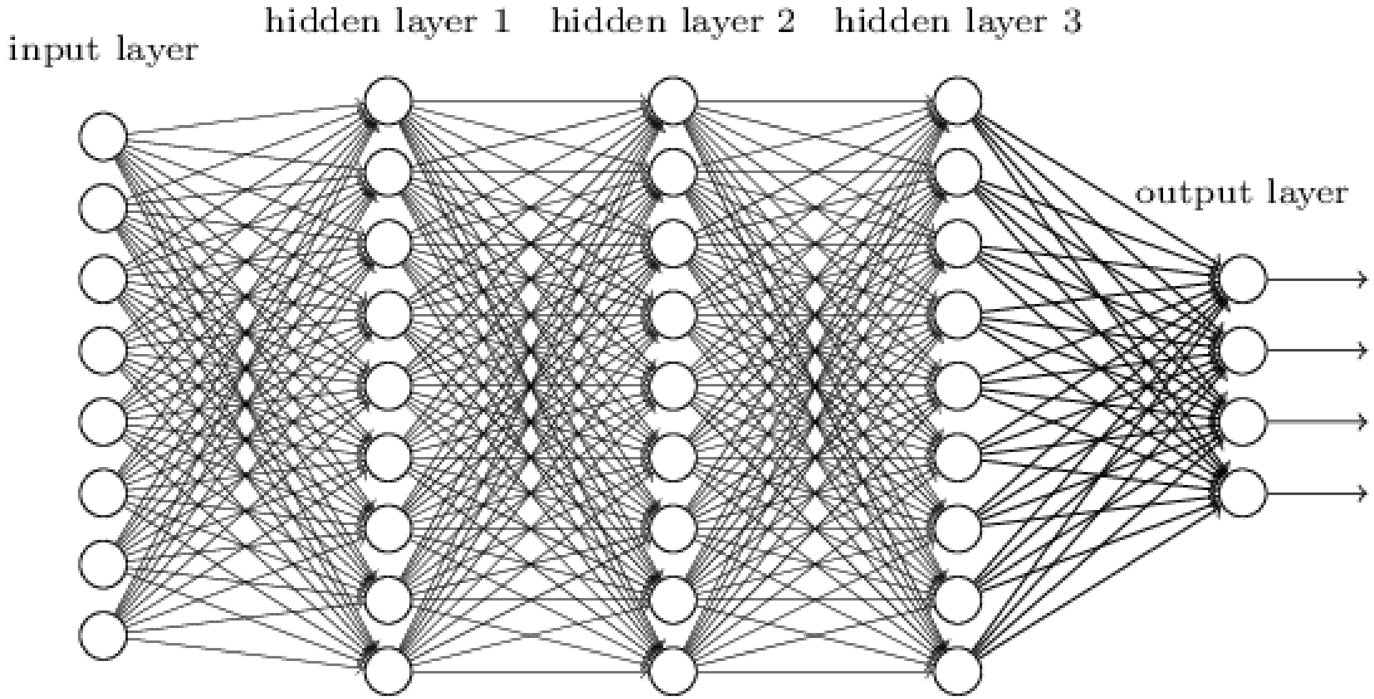
- Often requests come in for access to Big Data frameworks.
- Difficult to provide on a production cluster.
- Much easier to point them to AWS EMR (Elastic Map Reduce)
- Focus is on relevant components and problem solving not on its implementation!!



Specialised software - 2

- User requested a Windows environment on the local cluster.
- Requires annual Windows server licence
  - AWS occasional use might be cheaper!
- What is actually required?
  - Number of cores / memory / other software
- Spun up Windows server easily on EC2
- Remote console access immediately started with desktop environment.
- Quickly realised key package works on Windows desktop but not Windows server – back to vendor!

Deep neural nets - 1



See Michael Nielsen's ebook on Neural Networks

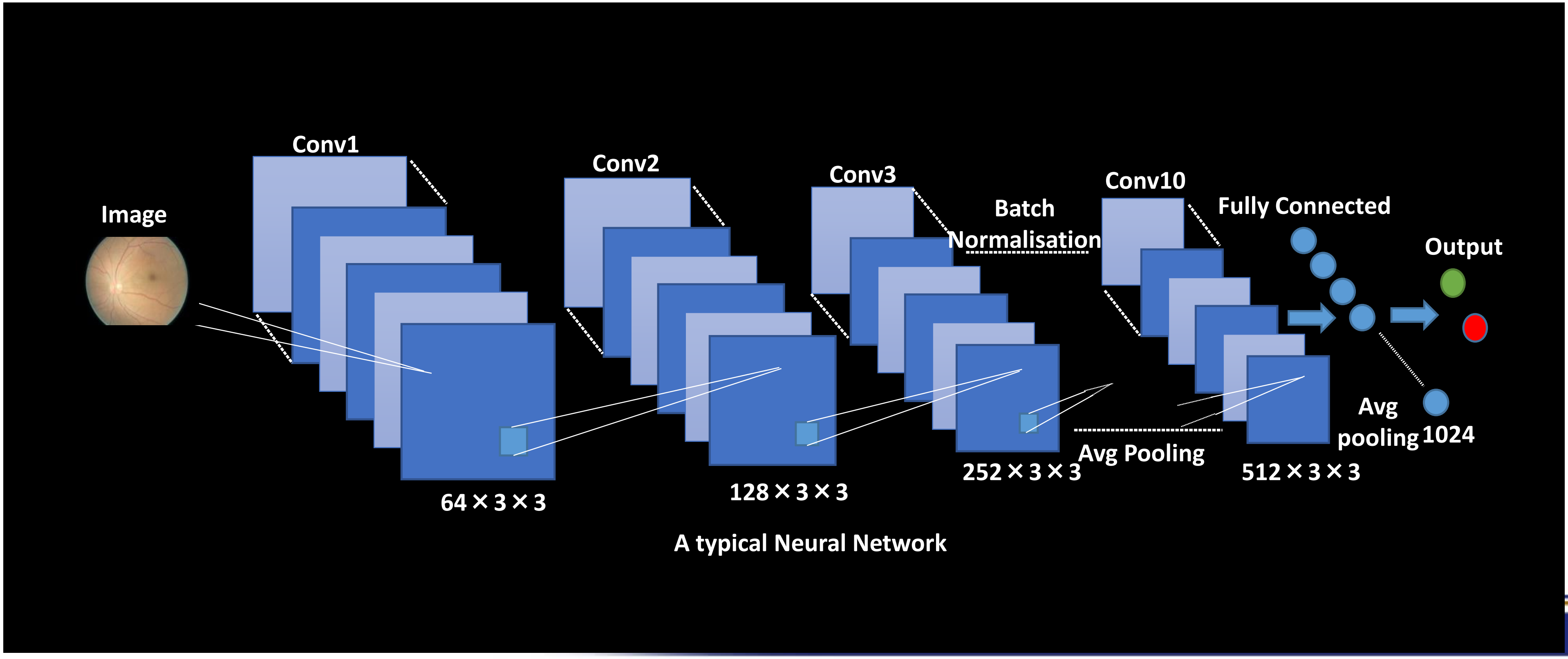
- Not just lots of layers – different layers have specific and different functions
  - Find functions and weights on edges that give "best" fit of computed outputs to actual outputs from training inputs.

Deep neural nets - 2

- There are several optimisation problems embedded in a neural net problem
  - Network definition - How define the correct operators (often with their own parameters) to act at different layers of the network.
  - What are the “best” weights on edges that connect neurons (nodes).
  - Typically obtain candidate weights by optimising across a set of input data with known outputs (training data)
  - Validate / test a “trained” network with separate data
- AWS services can help with above (later), but also there are Deep Learning EC2 instances.
- Ideal for those at development stage.

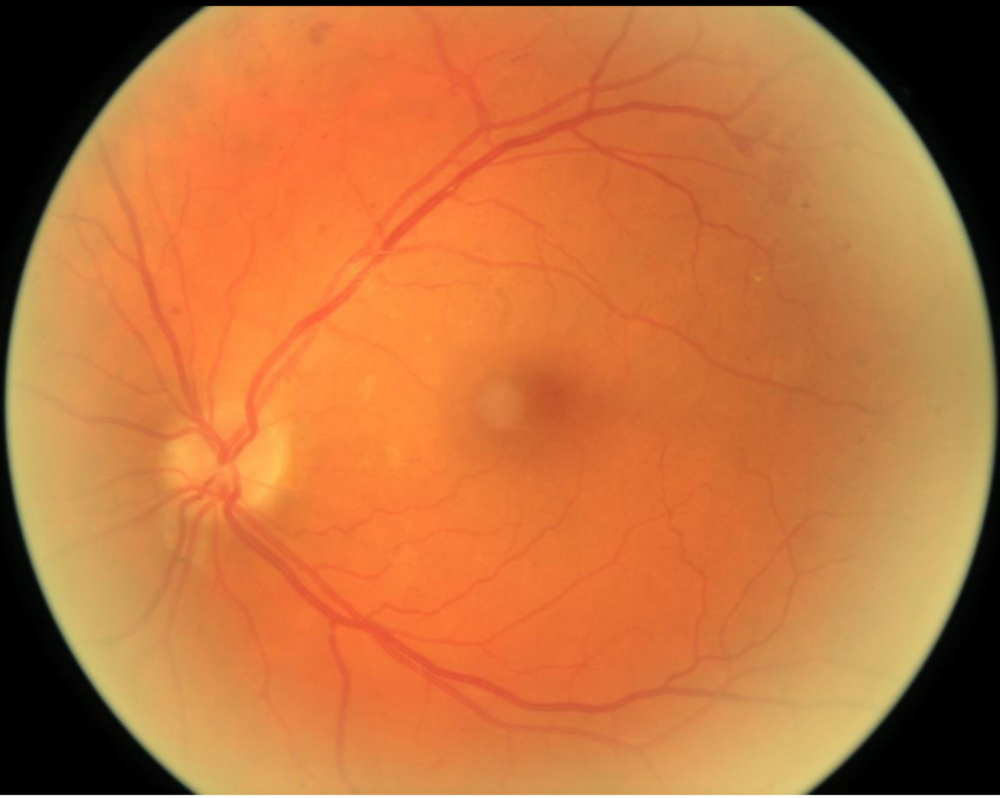
Medicine – convolutional neural net (CNN)

- Diagnose medical images for a range of eye diseases
  - GPU memory and speed are issues here – K80 – 15 minutes for 2000 images; v100 down to 2.5 minutes – **trivial set-up on AWS**



Required image detail poses problems

10M pixel raw retina image. Original size 3888 x 2592



57x pixel down-sampling required to achieve CNN ‘suitable’ image size

512 x 341 Image

Idea – convert to frequency domain via FFTs

Deep Learning @ Liverpool

- Range of maturity in developments
  - Are their network designs suitable?
  - Is there enough representative data to train and test?
    - Down-sampling can destroy required information!
  - Potential / need for augmented data?
- New / unusual user groups are appearing
- Single GPU node quickly overwhelmed;
  - Awkward situation where one node is not enough for some groups to get started – AWS really helps.
- Need to provide several levels of service
  - Workflows with coupled DL and HPC are complicated
  - New users totally flummoxed by HPC command line

Challenges - general issues

- Classical learning curve problems
  - Do people understand enough about the frameworks used to define sensible models?
  - Is there enough data of sufficiently high quality for training and validation?
  - Has enough thought gone into using the model in production (i.e. inference) and what about its life cycle (future training / modification)?
- What is the best environment for new users to start down the Machine Learning road?
  - AWS SageMaker very promising and Azure has its own ML environment – ideally start with on-premise hardware and burst?

Challenges – system side

- GPU nodes are expensive and hot
  - How keep them busy?
- Deep Learning cycles are bursty
  - How many GPUs are enough?
  - Hybrid cloud solutions a big help here
- Some DL problems involve a ***lot*** of data
  - Data storage, security, coping with hybrid cloud
  - Data ingest in a GPU often a bottleneck – IBM Newell (AC922) ?
- What environment is best for DL users?
  - SageMaker – focussed on the user's problem
  - Deep Learning EC2 instances – for the more experienced
  - Alces Flight – hybrid cloud; good for a group of users

Conclusions

- Working on AWS an eye-opening experience.
  - Some learning curve with the EC2 and with S3 storage
- Started with Alces Flight clusters and spinning specialised instances on EC2 (e.g. Condor in the cloud).
  - Just creating instances with keys for particular groups is great for small numbers of groups, but it does not scale.
- Can get major additional benefits to an on premise HPC.
- Other use cases for 2019:
  - Seamless cloud access from Barkla
  - Replication of some Barkla functionality in the cloud

BUT that misses a big cloud benefit

- Clouds can provide scalable environments that focus on the user's real underlying problem.
- Services like SageMaker and Elastic Map Reduce allow researchers to focus on their problem, not the implementation of the solution.
- When greater expertise has been gained, lower-level hybrid cloud services like Flight come into their own.
- How do we make life easier for our researchers??
  - They don't care about on-premise or cloud unless they are paying the bills.

FINALLY, there are still limitations

- There is still the capital vs. recurrent argument at most universities.
- Costs can escalate, particularly for GPU instances.
- Cloud storage can be useful, but that locks you in to a vendor.
- Need a good accounting framework so can assign budgets to groups.
- Want to hide the complexity of deploying onto a cloud from users, but want to avoid cutting own throat scenario.