Doing Deep Learning

Questions about Python/Linux from lecture on Piazza?

“Any appearance of support for Windows is purely coincidental and may end at any time without notice”

Raise your hand when I say things that confuse everyone and don’t notice

A brief history of GPUs

- Computer graphics require an unnatural amount of linear algebra computations
- CPUs weren’t fast enough, so GPUs came about (ISAs, ASICs)
    - GPUs vs graphics cards
- The death of Moore’s Law and the rise of ASICs; GPGPU
- Deep learning is basically all linear algebra, so GPUs

A brief history of GPUs

- Computer graphics require an unnatural amount of linear algebra computations
- CPUs weren’t fast enough, so GPUs came about (ISAs, ASICs)
  - GPUs vs graphics cards
- The death of Moore’s Law and the rise of ASICs; GPGPU
- Deep learning is basically all linear algebra, so GPUs
- Dedicated Neural Net ASICs are a thing, you can use Google’s (TPUs) on GCP if you want. Most new phones have them too. No desktop ones are readily available though.

GPU Manufacturers

- Intel
    - Integrated only
- Misc mobile manufacturers
- AMD
    - Lowkey given up to focus on Ryzen and the server CPU market, for good reasons
- NVidia
    - Winner by default
    - Also partially pioneered GPGPU, and deep learning before they were big

APIs

- OpenGL/Vulkan ; DirectX
- OpenCL ; CUDA
- Metal 2
- Performance and power consumption changes in Vulkan and DX12

CUDA

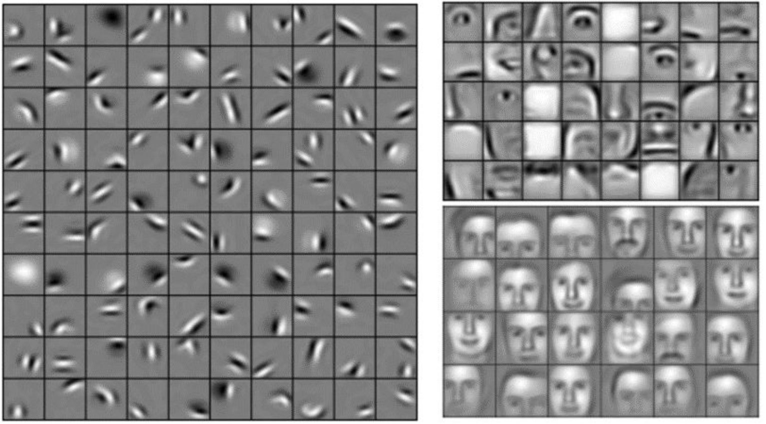
- So deep learning is basically all done on CUDA
- What's CUDA
- cuDNN?
- Installing? Drivers? Etc?
- AMD GPUs formally banned from being used
- Terrible OpenCL layers for some things
  - If it wasn't for Apple, openCL would be dead, and now they're stopping using it, so there's that

Deep Learning Libraries

- Keras
    - TensorFlow
        - Relatively difficult to read/write
        - Market share killer
    - Performance impact
- PyTorch
- Caffe 2
- MX Net
- Theano
- CTNK
- Tensorboard
- The role of open source
- Building from source, AVX

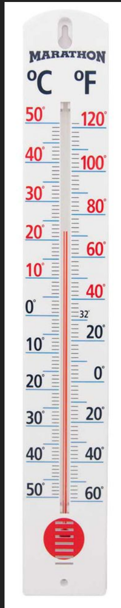
Debugging

- Visualize EVERYTHING
  - Tensorboard/ Seaborn (matplotlib)
  - Skim print of net bias and weight values
- Validate your data with a comical level of neuroticism
- Look at real inferencing cases of the trained model on images
- Training/testing split
- Modular code
- Look at, and do analytics of, your weights and biases
  - if the mean weight is 0, that’s instructive



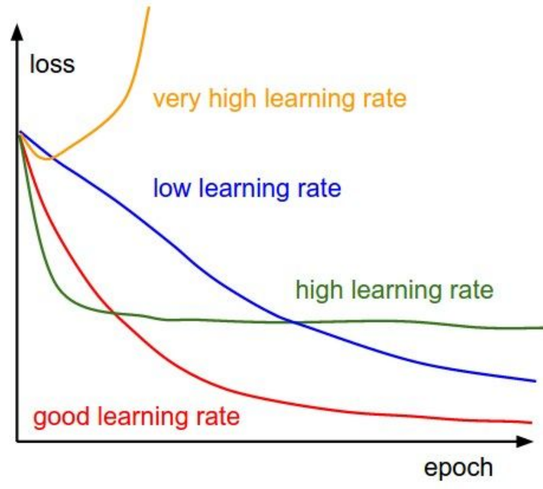
General Parameter Choices

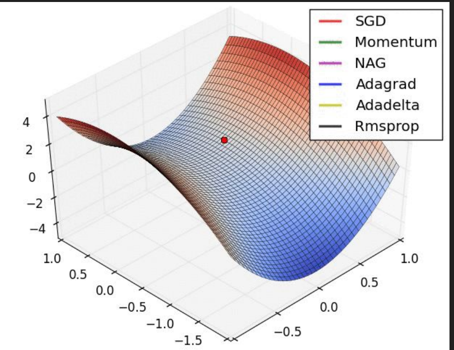
- Less capacity is generally easier to optimize
- Choosing number of filters
- Choosing filter sizes
- Choosing number of convolutional layers
- Choosing number of fully connected layers
  - https://www.youtube.com/watch?v=He4t7Zekob0



Optimization

- Adam et. al.
- Dropout
- Regularization
- Learning rate
- These will change accuracy during training
- Keras has good defaults
- Try to scale input values to be between less than .001 and 1
- Leaky ReLU
- Remove all extra data possible





Optimizing Hyperparameters

- Dark art
- Look for boundary conditions
- Genetic Algorithms (grid search)
- Trial and error
- Figure out limits of what it can train to or know what values other people can achieve
- Holding the RNG seed constant is usually asking for trouble