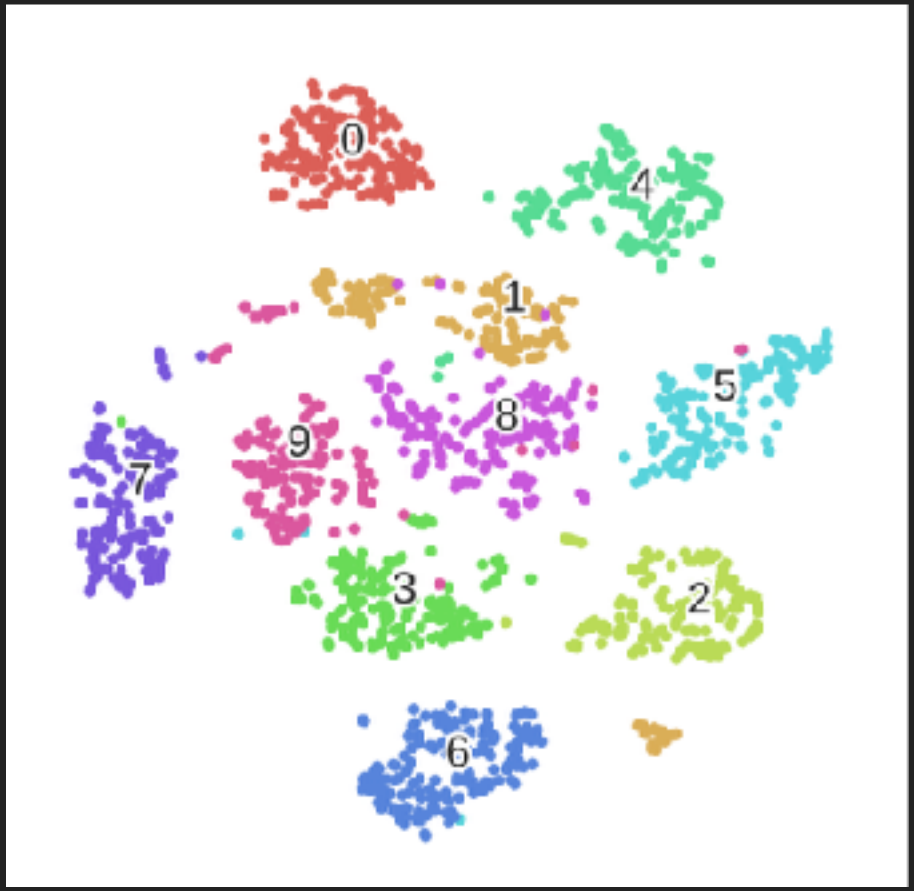
MACHINE LEARNING

AS AN EXTENSION OF LINEAR REGRESSION

- 2D Linear regression seeks to minimize a cost function (mean squared error) given a simple linear model Y=Wx.
- Many real world data scenarios involve MUCH higher dimensional spaces
    - Curse of dimensionality - higher dimensional spaces don’t behave like 3D
    - Can the machine pick up patterns that would elude us?

TYPES OF “LEARNING”

https://github.com/oreillymedia/t-SNE-tutorial



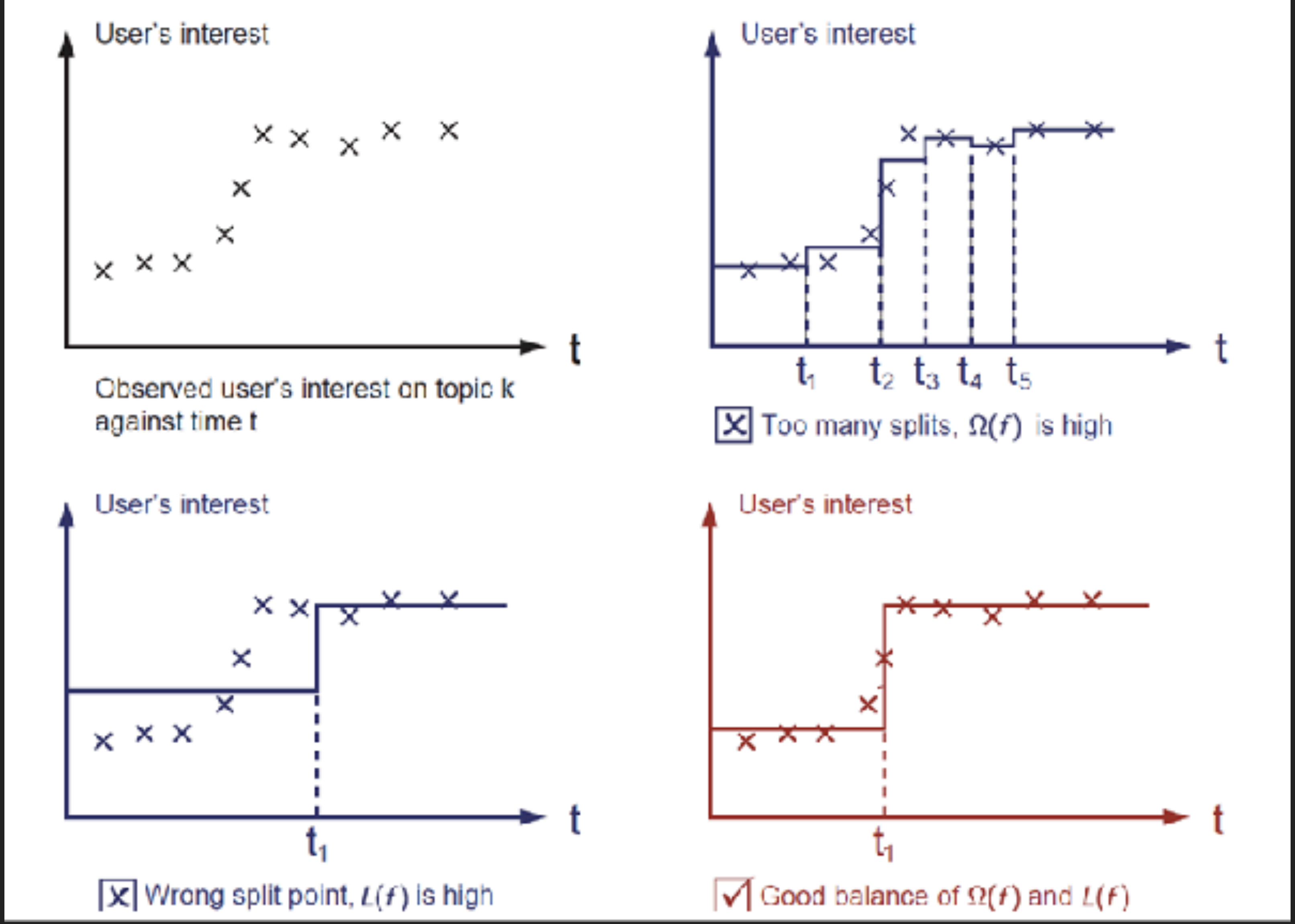
- Unsupervised Learning
  - Cluster recognition, kNN
  - PCA, T-SNE - dimension reduction
- Supervised Learning
  - Involve a “training” set where data samples have a known label, category or target - input -> output
  - MANY algorithms and techniques

SUPERVISED LEARNING

- Algorithms can be thought of as various ways of chopping up multidimensional space, for example in classification problems.
- Can we "learn" the rules of a signal that result in an outcome? Then when the program sees a similar (but different) input, it can give the correct response.
- Classic problem space - computer vision, recognizing speech, "expert" systems, pattern matching, face recognition.

GRADIENT BOOSTING

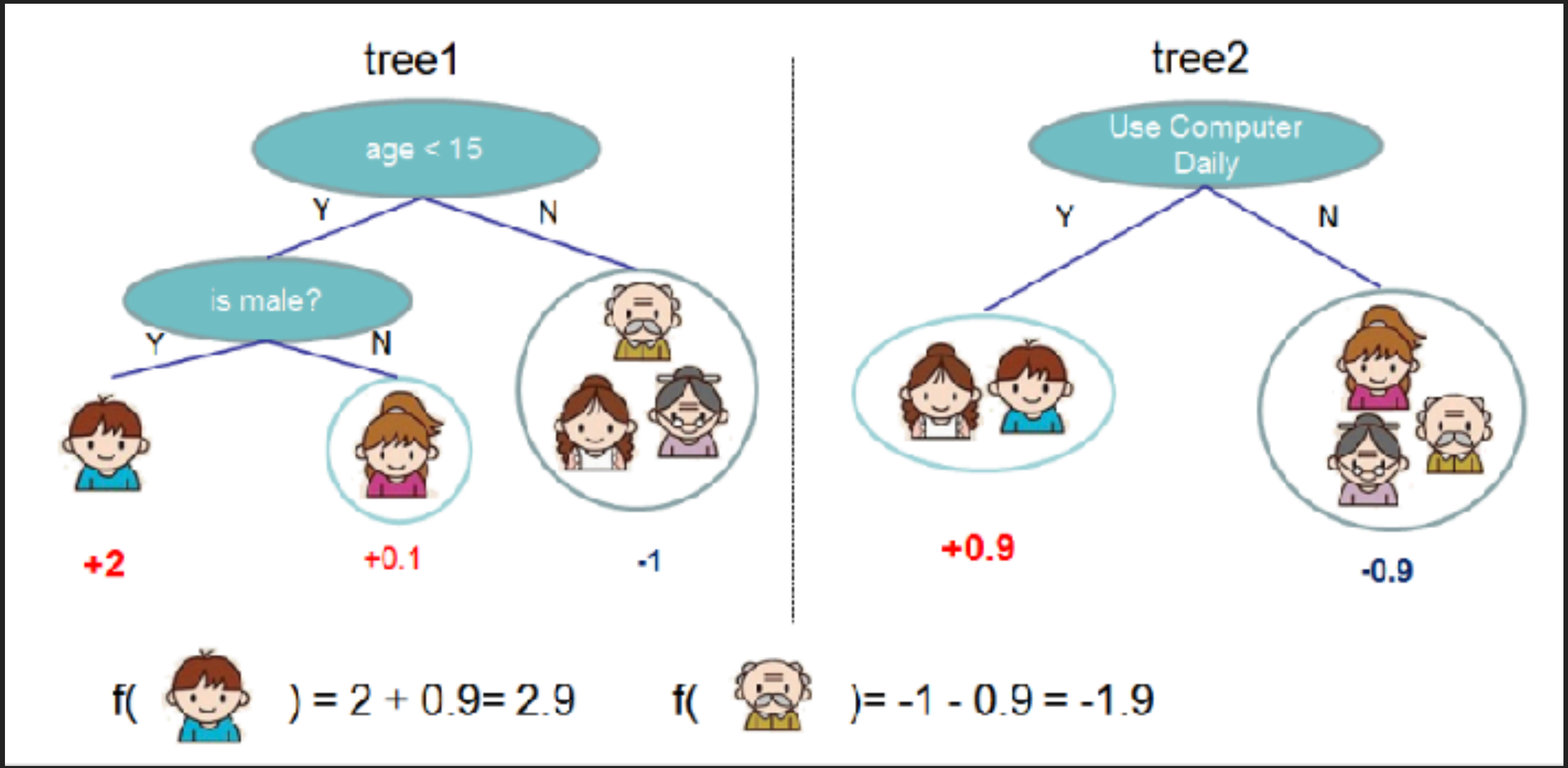
- Regularization
- Add penalty to the loss function based on model complexity



http://xgboost.readthedocs.io/en/latest/model.html

GRADIENT BOOSTING

- Builds up an ensemble of decision trees step by step, each step trying to improve the loss function while keeping the model complexity down (regularization)



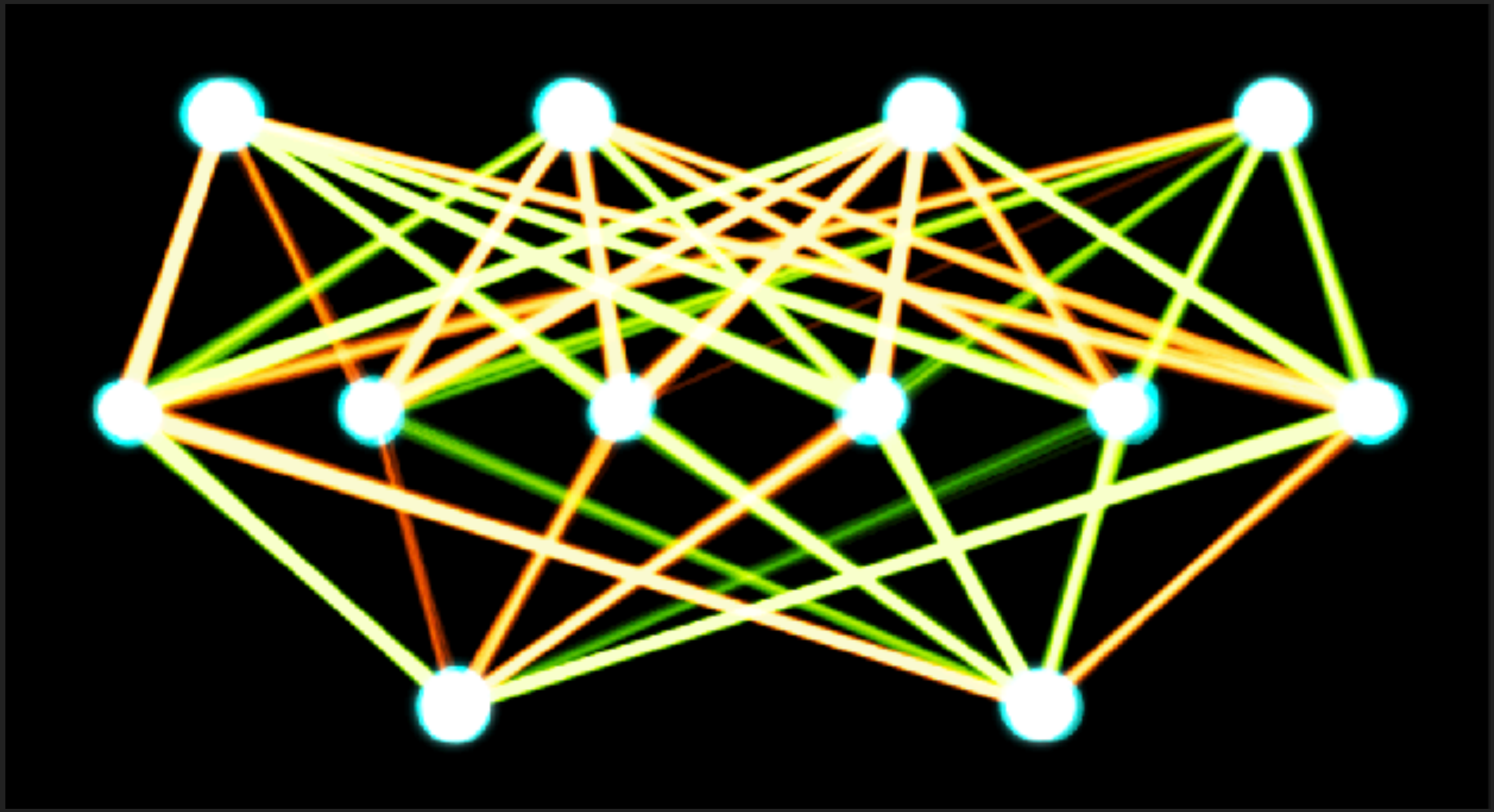
http://xgboost.readthedocs.io/en/latest/model.html

GRADIENT BOOSTING

- Kaggle Higgs Boson Challenge
  - XGBoost used to produce the winning model in challenge to classify tau + tau -> Higgs boson decay events versus background
  - Use of machine learning to detect small signals in background noise
- Boosted trees also successful at many other classification problems

NEURAL NETWORKS

- History - From perceptrons to deep and recurrent nets
- Advancements in training algorithms



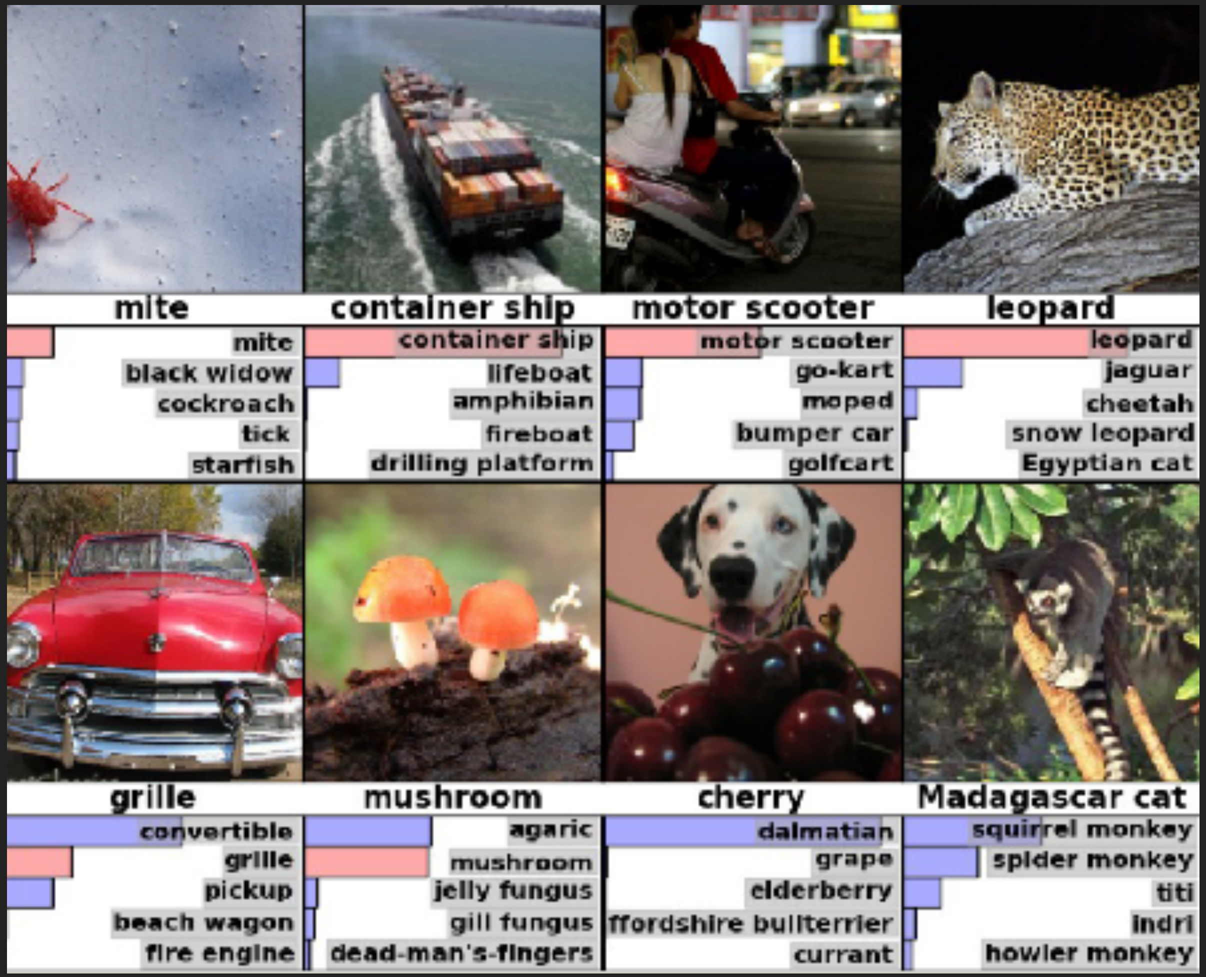
By Akritasa (Own work) [CC BY-SA 4.0 (http://creativecommons.org/licenses/by-sa/4.0)], via Wikimedia Commons

AI AND THE GAME OF GO – NEURAL NETWORKS

- Until 2015, computer programs could only achieve an “amateur dan” rating; a decent human could reliably beat them. Go is MUCH harder to program than chess.
- In 2017 Google’s AlphaGo program beat the #1 player in the world in a tournament setting. This is a result of advancements in the training of neural networks.
- NN trained with 30 million moves from human games, then multiple NNs paired against one another to train and improve.

DEEP NEURAL NETS

- Computer vision
  - Convolutional nets - inspired by similar structures to our own vision.
  - ImageNet Challenge
    - 1.2 million images
    - 1000 classes
    - Best NNs can get ~85% accuracy



https://chsasank.github.io/deep-learning-crash-course-2.html

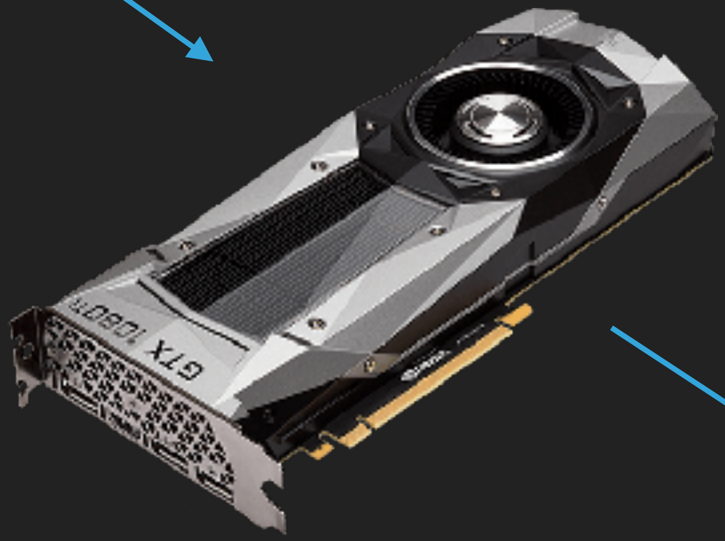
DEEP NEURAL NETS

- Speech generation direct from text
  - https://google.github.io/tacotron/
  - “The buses aren't the problem, they actually provide a solution.”
  - “The buses aren't the PROBLEM, they actually provide a SOLUTION.”

WHAT MAKES IT POSSIBLE?

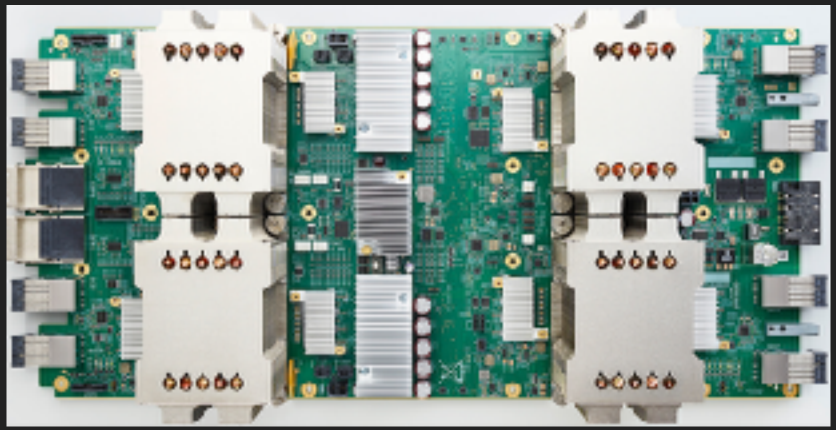
- Computational resources now allow training of machine learning models of impressive complexity and capability

CPU



GPU

TPU



SOFTWARE TOOLS

- General Tools
  - Python - scikit.learn
    - Optimizers, unsupervised and supervised learning, classification, simple NN
    - Tools in many other languages: R, Java, Julia, C++, Mathematica
- Neural Networks
  - Keras, TensorFlow, Theano, DeepLearning4J

QUESTIONS … AND DEMO