# Machine Learning working group

Scalable Tools Workshop 2017

Lake Tahoe

# Intro

- Introduction by David Levinthal
- How/Where can we collect it? What type of data?
  - Collecting data from clusters, multiple data collectors and sinks
  - How to handle time-series? -> rNNs (Levinthal)
- Most try and error but intuition helps
  - There are approaches to help doing try and error
- Large correctly tagged training sets required (for supervised learning)
  - Army tank recognition training set at cloudy and sunny days

# Application for HPC data

- Requires model and tagged training set
    - Be careful about the inputs!
    - Might be biased or missing characteristics
- Workflow:
    - Create a reasonable and big training set
    - Design the network (mostly) by try and error
        - Multiple data sources might require multiple network types put together
    - Hyperbolic parameter sweeps for convergence and stabilization
    - Domain specific knowledge required

# Potential ML at Livermore

- Task: How to schedule jobs that they don‘t influence others
  - Predicting job runtime by looking at job scripts
  - Predicting load on filesystem for job
  - Problem: Same job script but runtimes differ
    - different inputs but not mentioned in job scripts
- Task of mesh refinement/relaxation
  - Can change over time
- Maybe tasks can be solved with
  - FNN (fully connected NN)
  - CNN+RNN

# Input data

- Time-series probably RRNs
  - CNN for first item in sequence and then RNN?
- Do we get all the data? -> Security issues, multiple sources
- How to tag the data for training sets
  - Users rate their jobs from 1 to 10?
  - Most difficult task!
- Objective functions are different for performance data than for image recognition, …
  - Sometimes only parameter testing -> no ML required
  - Can we compare same kernels in different applications?
  - How often do we train (nightly?)
  - Does overfitting matter?
- Domain specific knowledge needed to create the model