Using Deep Learning to Sort Down Data

David Schneider, davidsch@slac.stanford.edu

June 15, 2016

U.S. DEPARTMENT OF ENERGY
Office of Science

SLAC NATIONAL ACCELERATOR LABORATORY

Project Team

SLAC

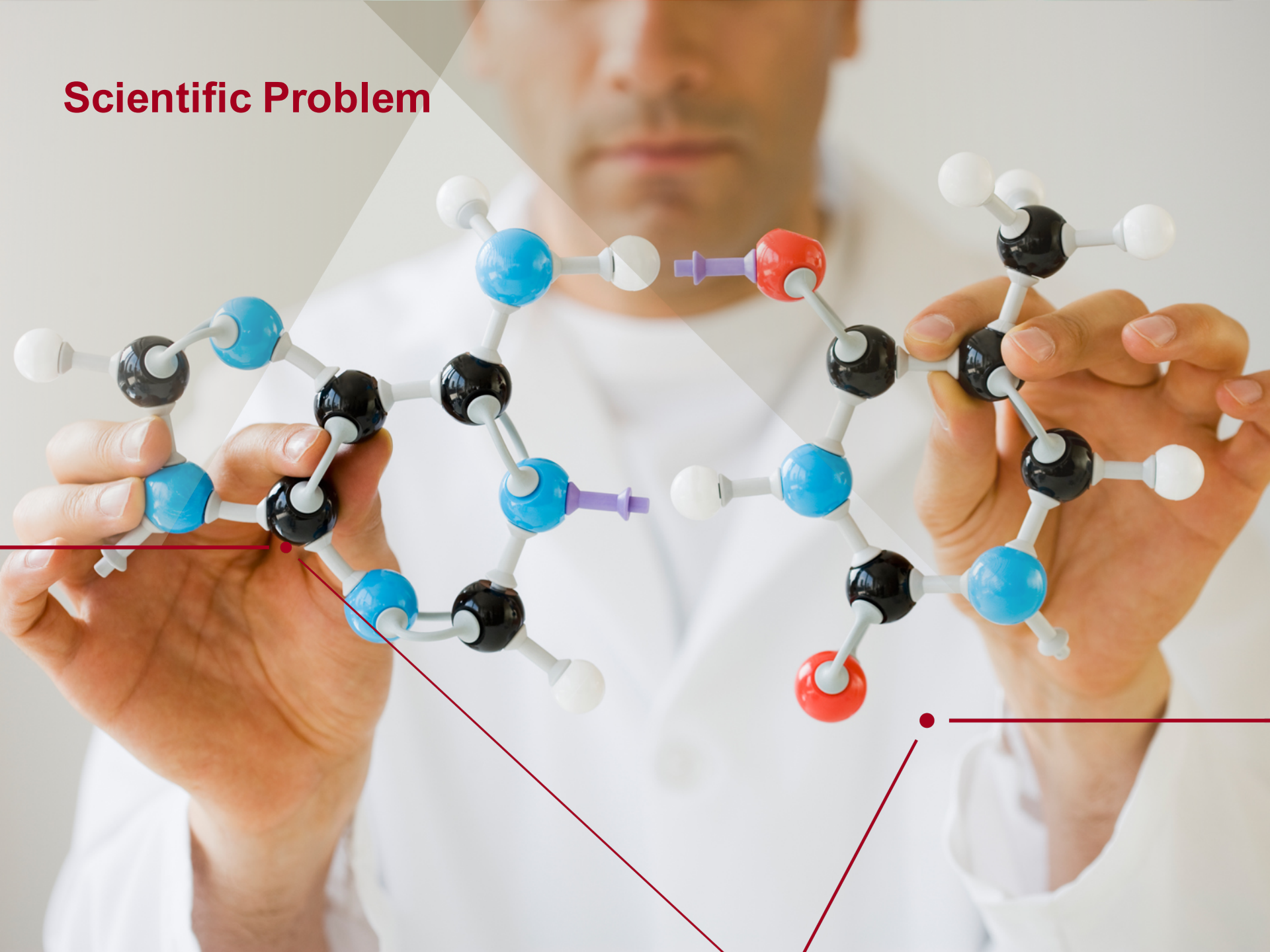
Joint work with

- Mihir Mongia – Stanford Applied Mathematics graduate student
- Ryan Coffee – LCLS staff scientist, sponsored problem, data
- Chris O' Grady – LCLS Data Analysis group
- David Schneider – LCLS Data Analysis group

Outline

- Scientific Problem
- Machine Learning
  - Then and now
- Applying Machine Learning to this problem
  - Initial sorting down results
- Guided Back Propagation
  - See what the model thinks is important
- Future work

Scientific Problem



Chemistry: Understand dynamics of a molecule

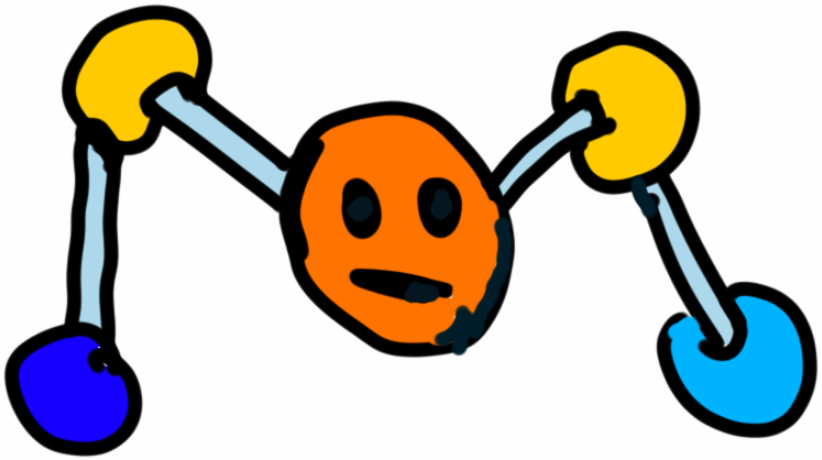
SLAC

Measure reaction to different two color shots

Chemistry: Understand dynamics of a molecule

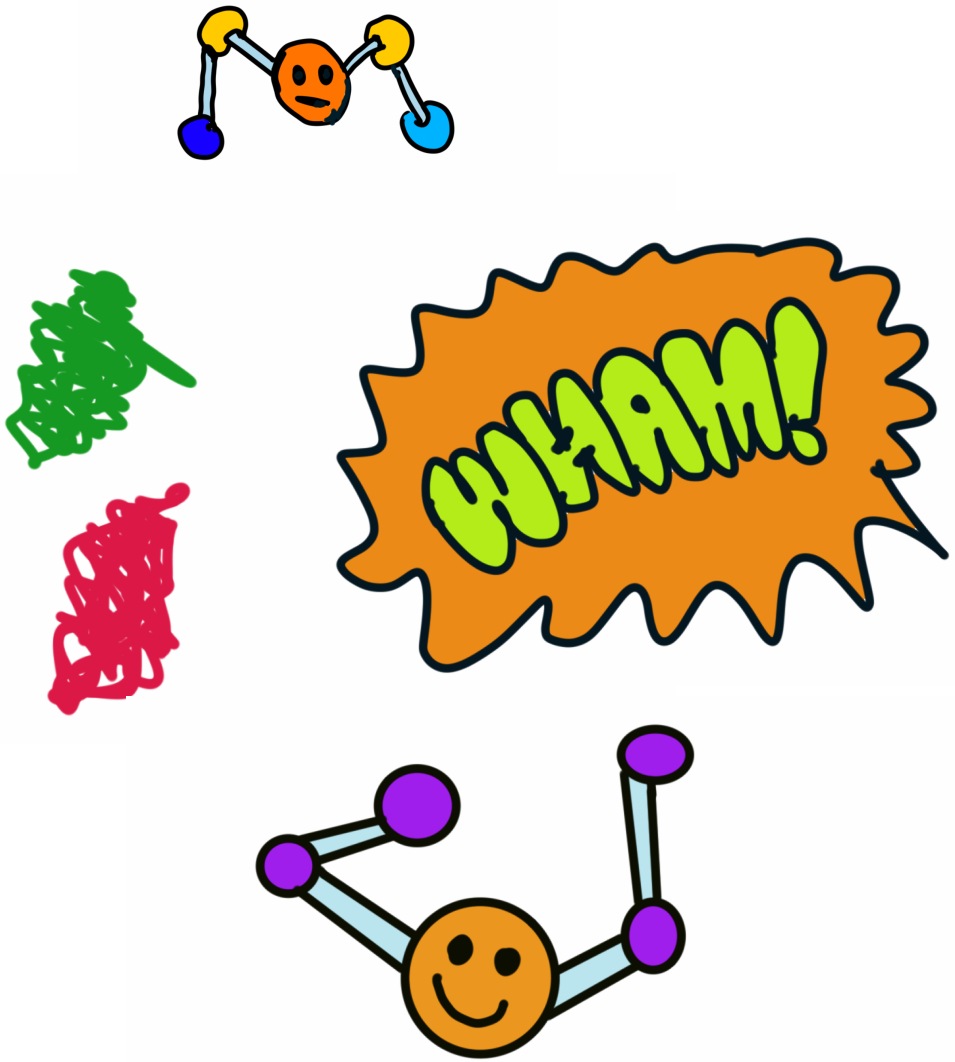
SLAC

Measure reaction to different two color shots



Molecule

Chemistry: Understand dynamics of a molecule



Sort Down to see Molecule Reactions

How good is the two color characterization?

What is signal/noise ratio for molecular reactions?

How much data was collected?

Sort down - average all shots labeled with the same characterization, hopefully see reaction of interest

Characterize Two Color Shots – 2D Image Processing

SLAC

Which lased?
Green? Red? both?

What were Exact Energies?

Xtcav diagnostic image

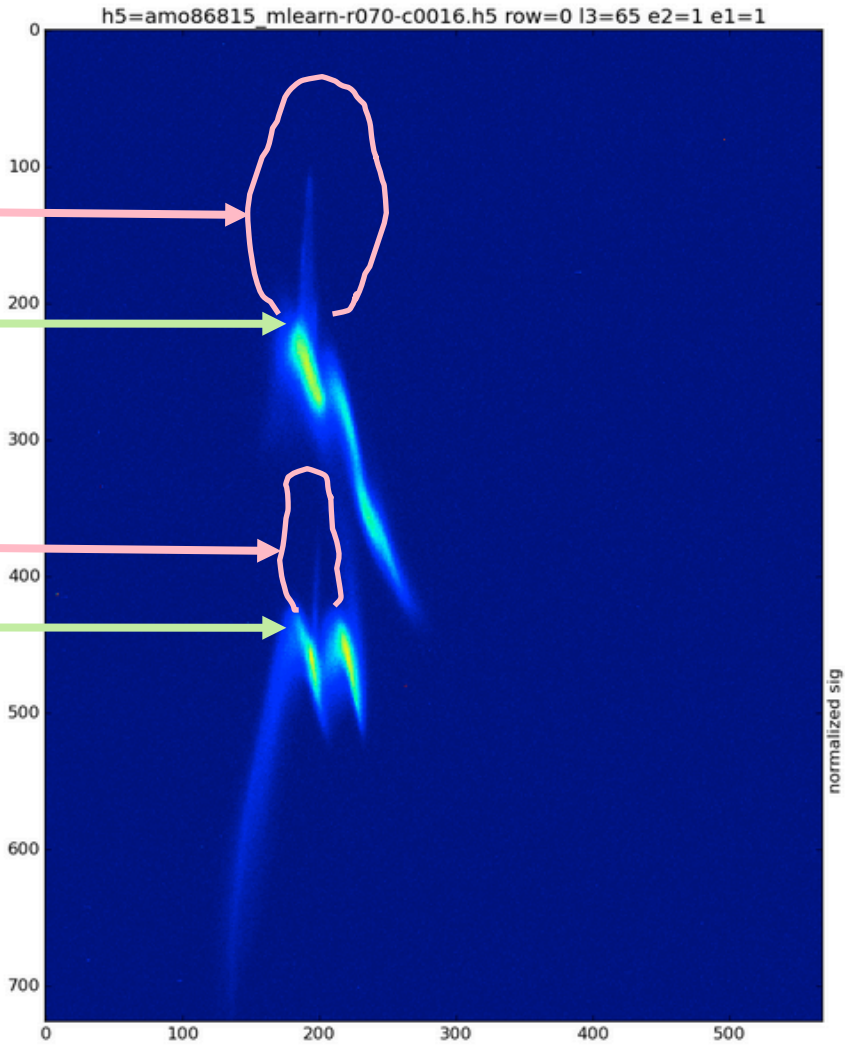
Low energy Lased, 'finger' present

Low energy value – base of finger on y axis

High energy Lased

High energy Value Base of finger on y axis

2D image processing identify fingers



Characterize Two Color Shots – 2D Image Processing

SLAC

Which lased?
Green? Red? both?

What were Exact Energies?

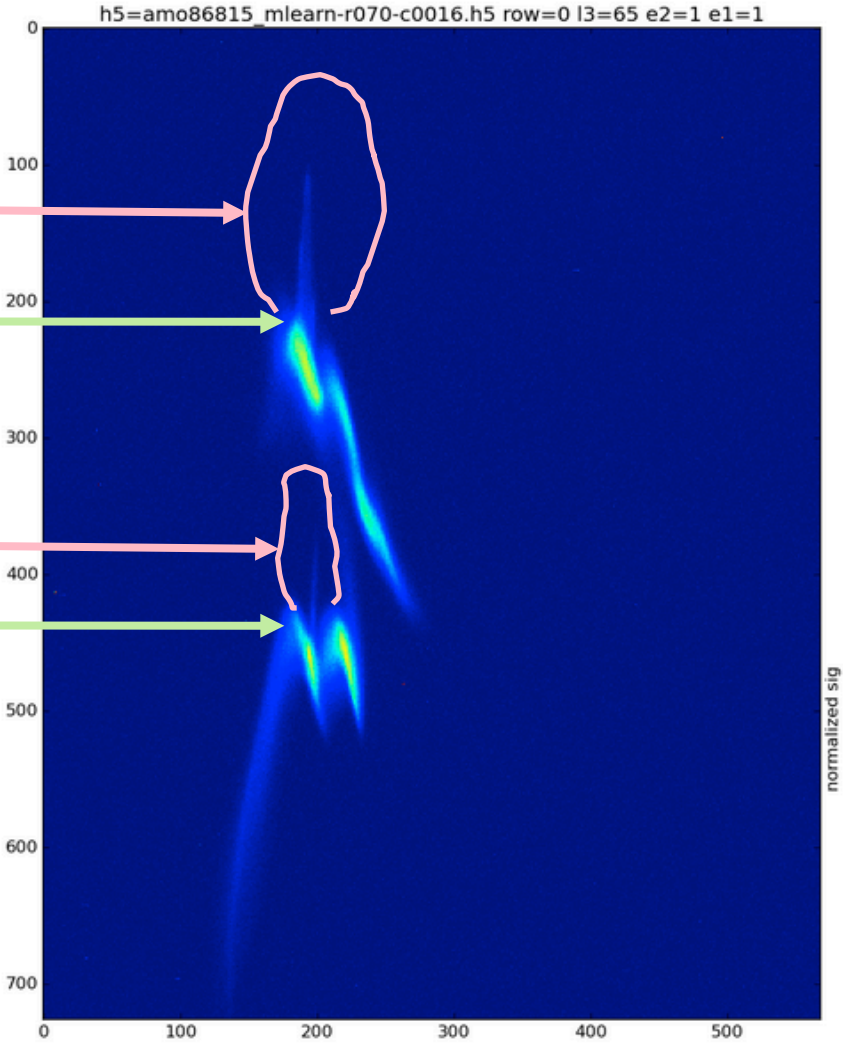
Xtcav diagnostic image

Low energy Lased, 'finger' present

Low energy value – base of finger on y axis

High energy Lased

High energy Value Base of finger on y axis

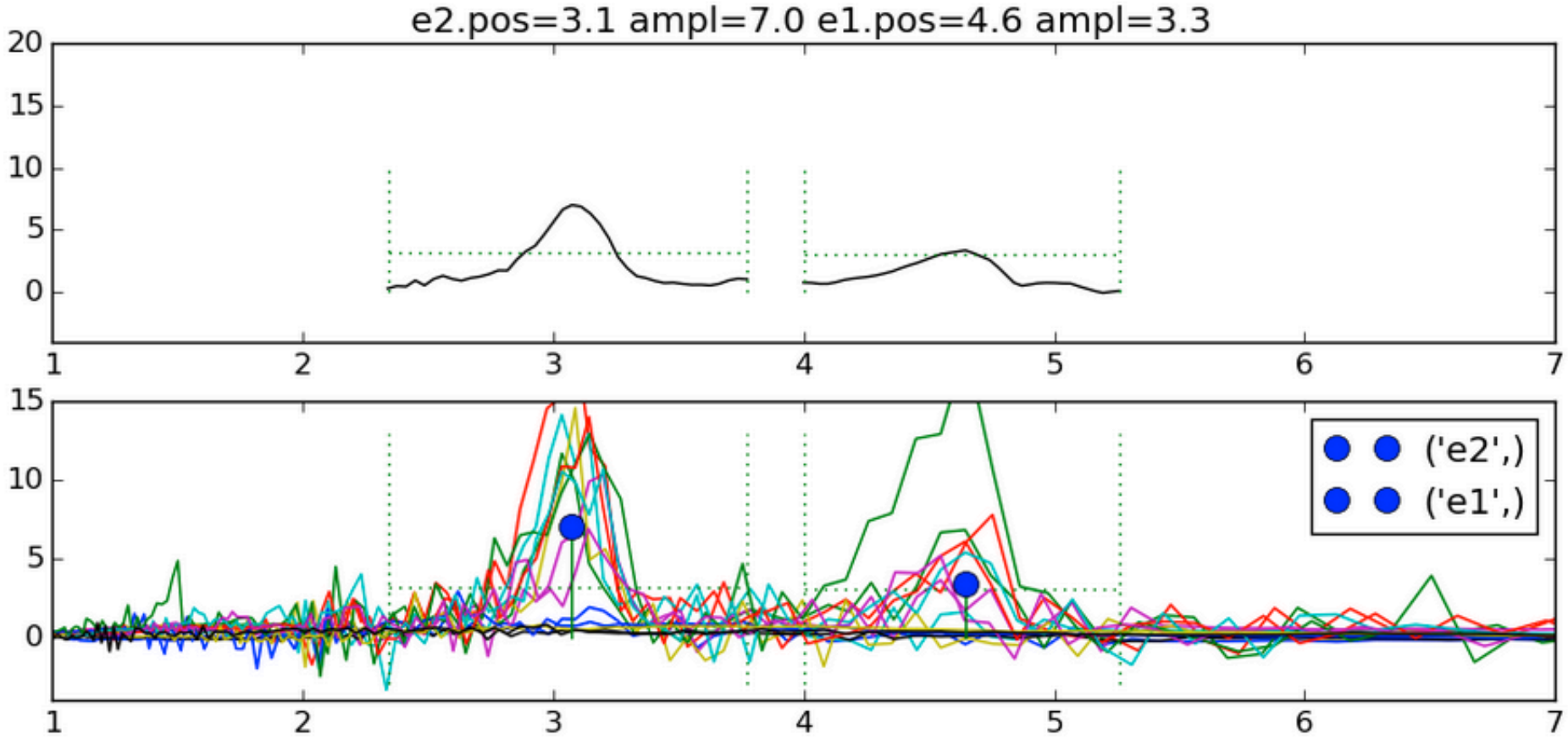


Problem: algorithm for todays Experiment won't generalize to tomorrows

Two Color characterization by 1D Peak Finding

SLAC

Detector measures lasing and energy of each color



Two Color Characterization by Machine Learning

SLAC

- Problem:
  - One Detector
  - Measure 1D peaks from two color, or
  - Molecular reactions to two color
  - Can't measure both
- Training runs: detector measures two color lasing
  - Train model on detector output
- Molecular runs: detector measures molecular reactions
  - Use model to predict two color lasing
  - Sort down detector output based on predictions

Machine Learning

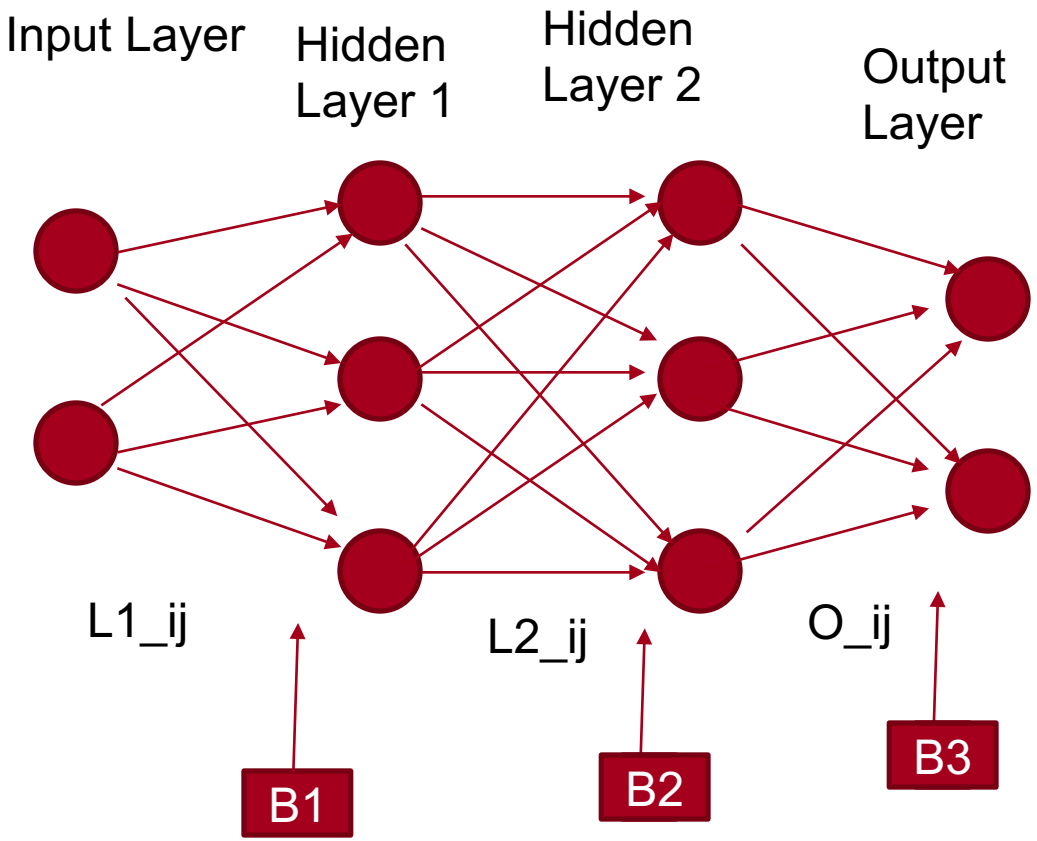
Machine Learning Background Information

SLAC

- Supervised: have labeled data
  - Train model to predict label for unseen samples
  - Classification: predict category – dog or cat
  - Regression: predict real values – home prices
  - Many kinds of models:
    - Linear Discriminate Analysis, Support Vector Machines
    - Random Forests, Neural Networks
  - Loss functions – mean squared error, soft max
  - Optimizers
    - variations of stochastic gradient descent on mini batches
  - Fit model to training data
    - minimize loss function
  - Measure performance on validation data
  - Use regularization to avoid overfitting

Example Neural Network – Fully Connected

Hidden Layers: sum all input layer nodes, add bias, apply nonlinearity



Neural Networks: Then and Now

SLAC

Past:

- Too long to train
- Machine learning used other models

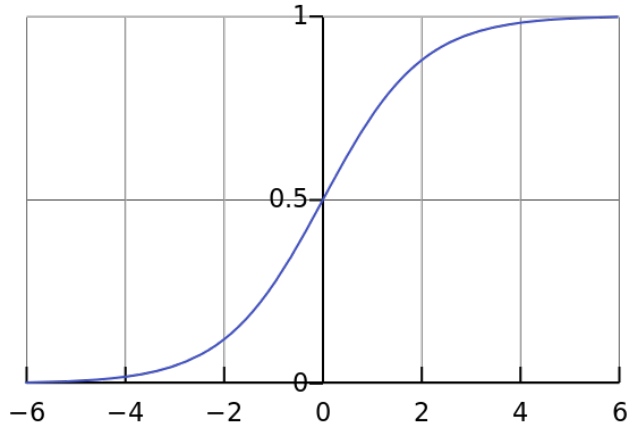
Now:

- More compute power
- GPUs
- More Data
- New Algorithms/Techniques
  - Convolutional layers
- Deep Learning: many layer neural network
- Deep Learning outperforms other models (on many problems)

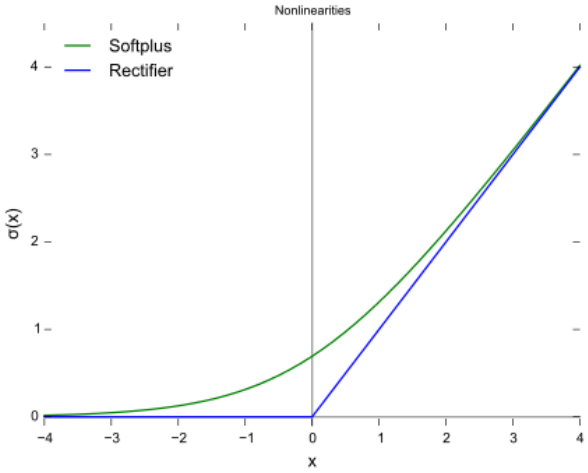
Stanford collaboration: modern ideas

SLAC

Nonlinearity – smooth, sigmoid?



Rectified linear – not smooth!



Batch Normalization:

deep network training,
Long multiplies in math :

$$I \cdot L1 * L2 * L3 * \cdots * LN$$

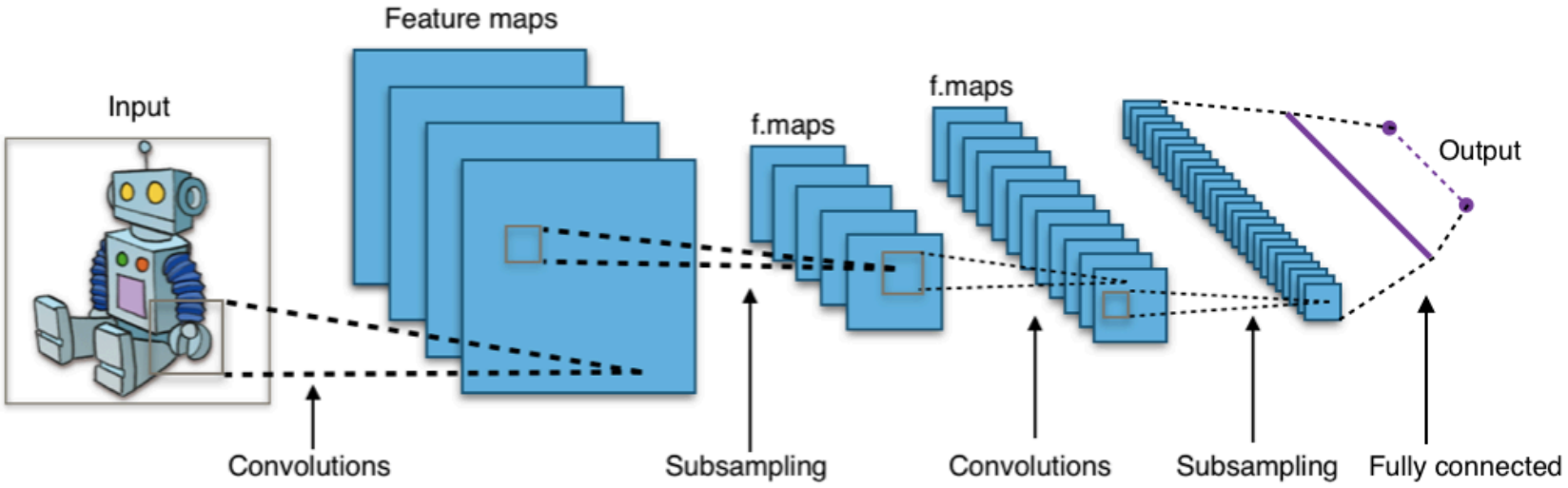
Would be easier if Lk were mean=0, std=1

Add layer before nonlinearity to learn parameters to make this so

Train much Faster!
Parameter Initialization Much easier!

Images from wikipedia: https://en.wikipedia.org/wiki/Rectifier_(neural_networks)
https://en.wikipedia.org/wiki/Logistic_function

Convolutional Neural Networks

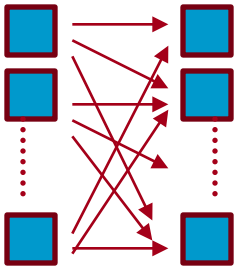


By Aphex34 - Own work, CC BY-SA 4.0, https://commons.wikimedia.org/w/index.php?curid=45679374

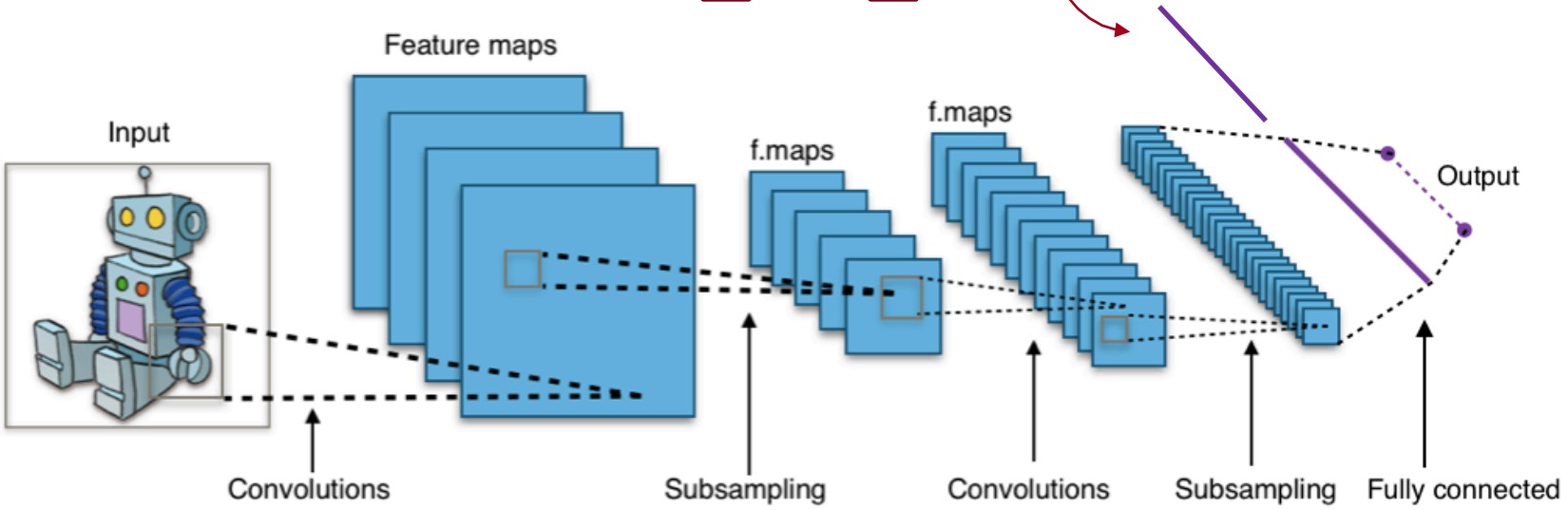
Merge Model

Our Input:
Image + feature vector
of BLD parameters
(ebeam, gasdet)

Dense Layers



Concatenate Bld
Feature vector with
Flattened Convnet
Output before final
Dense layers



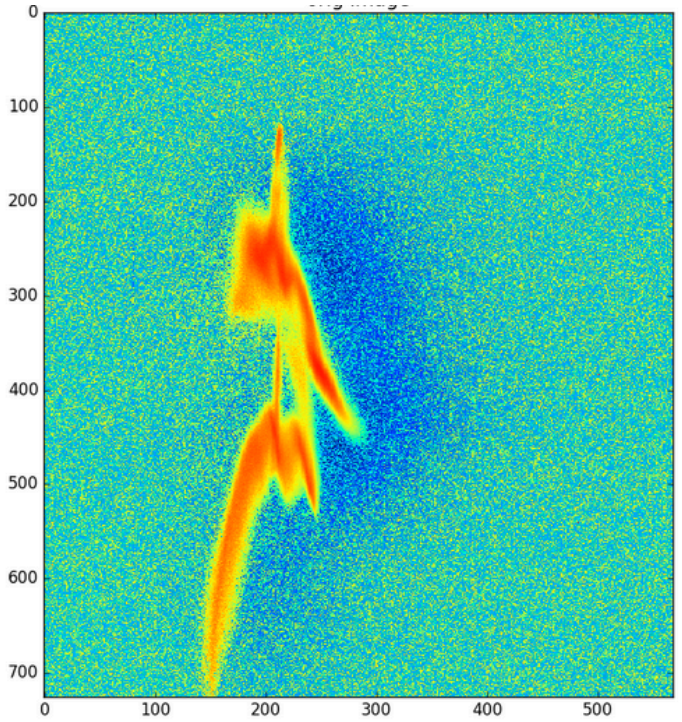
Applying Machine Learning to the our Problem

Classification Problem – which Colors Lased?

SLAC

Labels:

- 0 (neither)
- 1 (high energy),
- 2 (low energy)
- 3 (both)



Challenges

- 80 GB training data
- Images: 726 x 568 int16
- Lasing features ~ 200 ADU, beam 2000 ADU

Image Preprocessing

- Log transform, subtract mean

GPU/Deep Learning Frameworks

SLAC

GPU

- LCLS is lucky to have one nVidia Tesla K40
- 12 GB mem, supports CDNN v4
- x30 speedup (compared to 16 core 128GB Machine)

Deep Learning Frameworks (Python, GPU/CDNN)

- Tensorflow – now at r9, like it, but batch norm harder
- Theano – found it to be faster than earlier TF versions, but initial compiling is slow
- Keras – wrapper to Theano or Tensorflow
  - Really nice for typical models
  - Makes batch normalization easy
  - Issues with making merge model

Hyperparameters

- Random Initialization of model parameters
- Learning rate, add decay? Decay rate
- Which optimizer, optimizer parameters (like momentum)
- Architecture – number of layers, nodes per layer
- Covnet architecture – kernel size, pooling strides
- Which pooling function
- Which nonlinear activation function
- What kind of regularization, strength of regularization
- Loss function

Bottleneck

SLAC

First convolutional layer

Suppose you want 24 feature maps,

Minibatch of 64 images

24 x 728 x 568 x 64 x 4(float32) = 2.4 Gb

More with derivatives (backpropagation algorithm)

That's just the first layer

Often reduce batch size to handler bigger model

Final Classification Model

SLAC

Batch size: 40

7 convnet layers

- kernels range start at 12 output channels, end at 4
- kernel shapes start at (8,8) end at (3,3)
- 2 x 2 pooling for 4 layers (then image is 45 x 35)
- 1300 parameters, 6300 outputs

Merge with 16 outputs of Bld dense layer

Three dense layers: (6316 -> 40), (40->40), (40->40)

Final logits (linear map) 40->4

Cross entropy loss on softmax of logits

0.01 L2 regularization

Learning rate = 0.1, decay

Momentum = 0.85

270,000 parameters, 89% validation accuracy

Regression

SLAC

One option: train new model against numerical values

Another: use final outputs of classification model

Fit linear regressor using 120 inputs

(the 40,40,40) of last 3 layers before 4 logits

R^2=0.84 for e2, lower energy

R^2=0.67 for e1, higher energy

Quite good – model that predicts mean has R^2=0

$$R^2 = 1 - \frac{\sum(y - y_{predicted})}{\sum(y - \mu)}$$

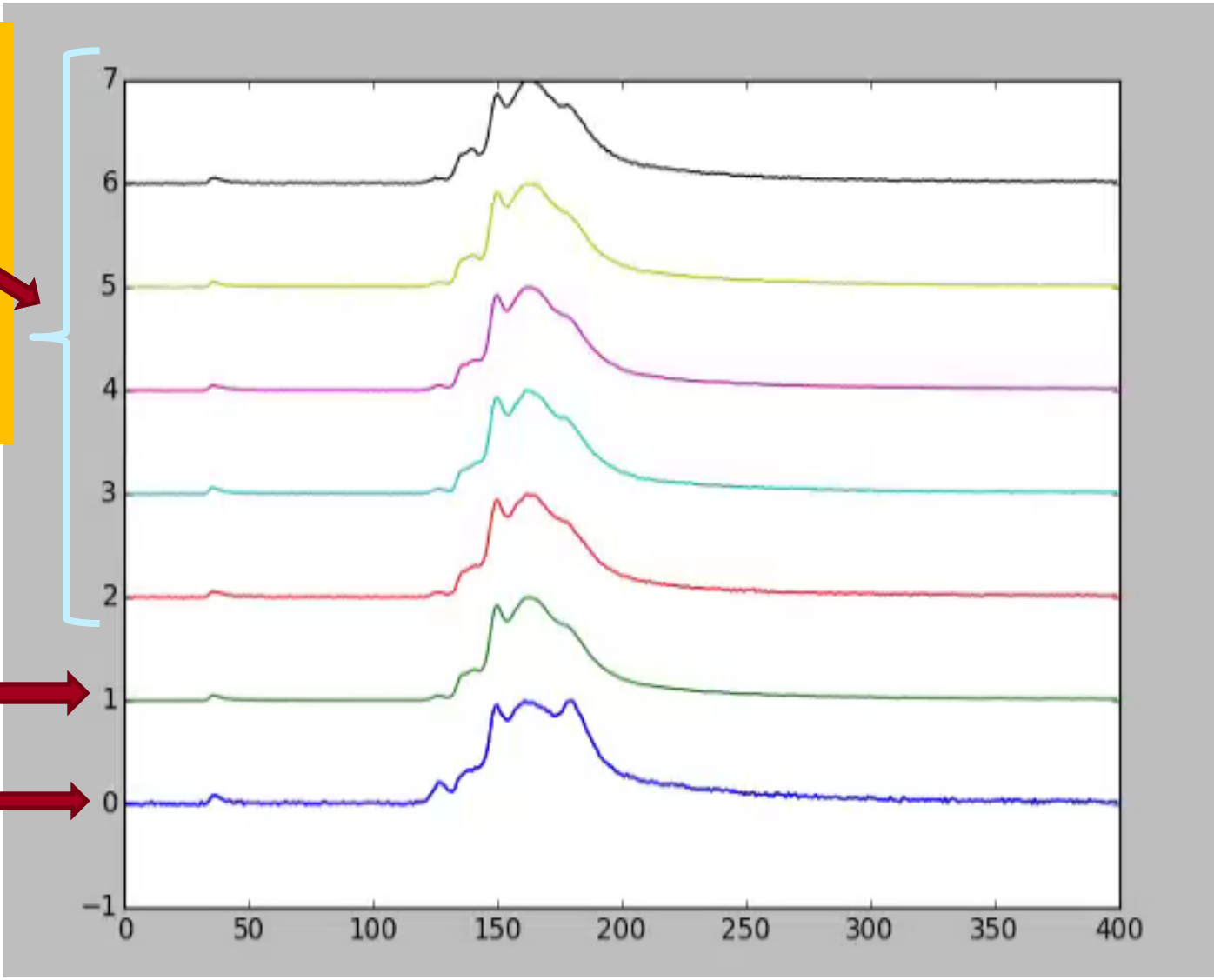
Initial Sort Down Results

SLAC

Both High and Low Energy Lased.
Fix high to (4.5-4.6) Interval of regression values, scan low over intervals in [2.9, 3.0, 3.1, 3.2, 3.3, 3.4]

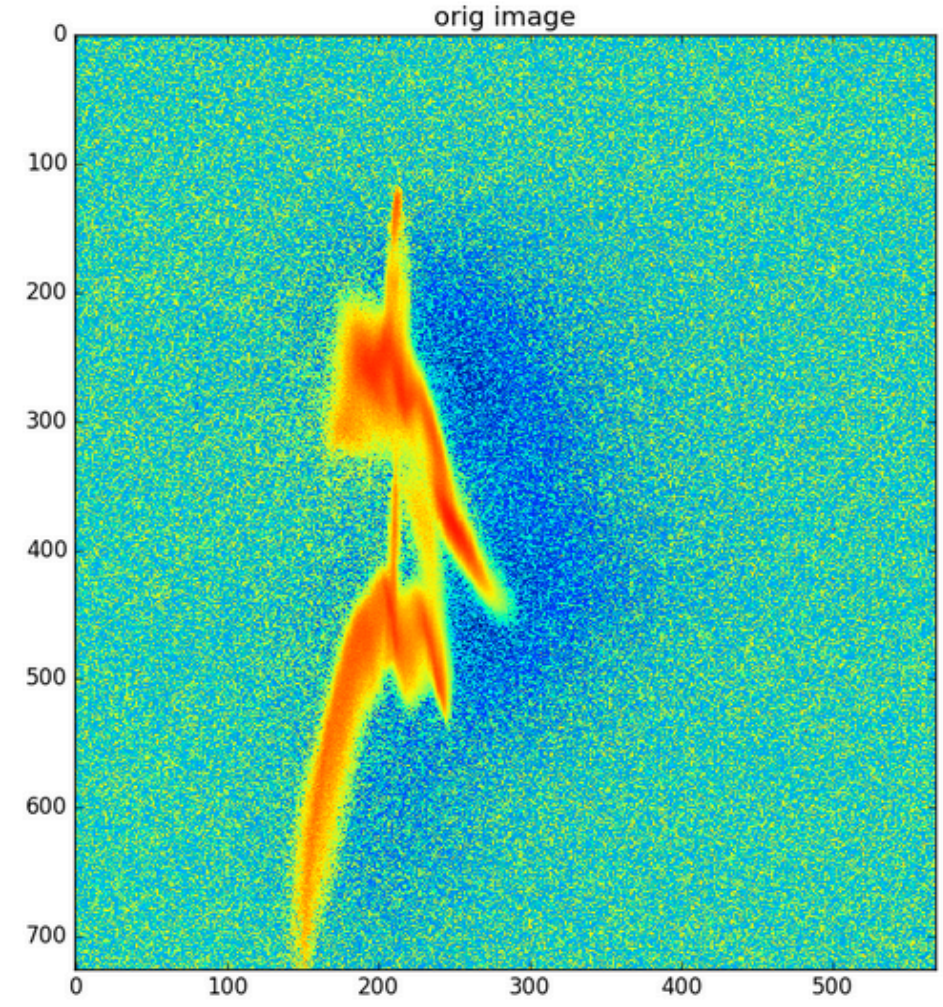
Average 5 plots above

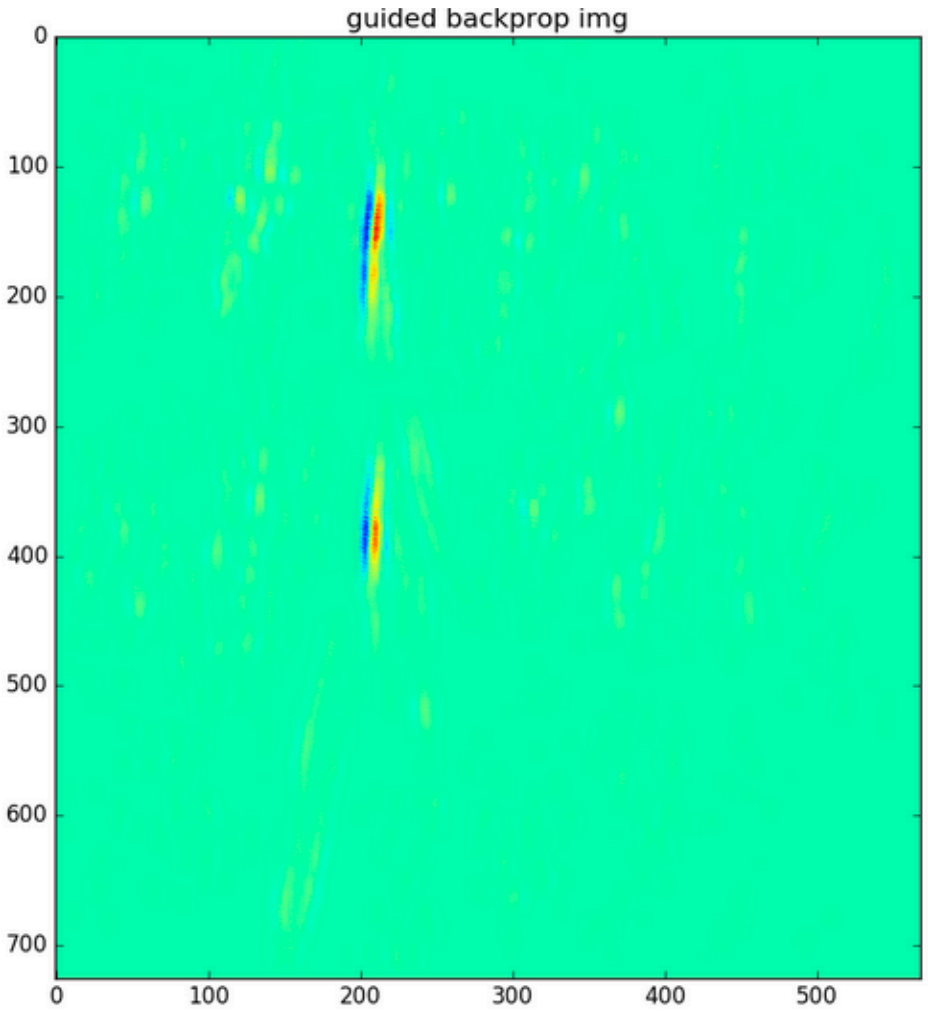
Only High Energy Lased – avg over (4.5-4.6) interval of Regression values



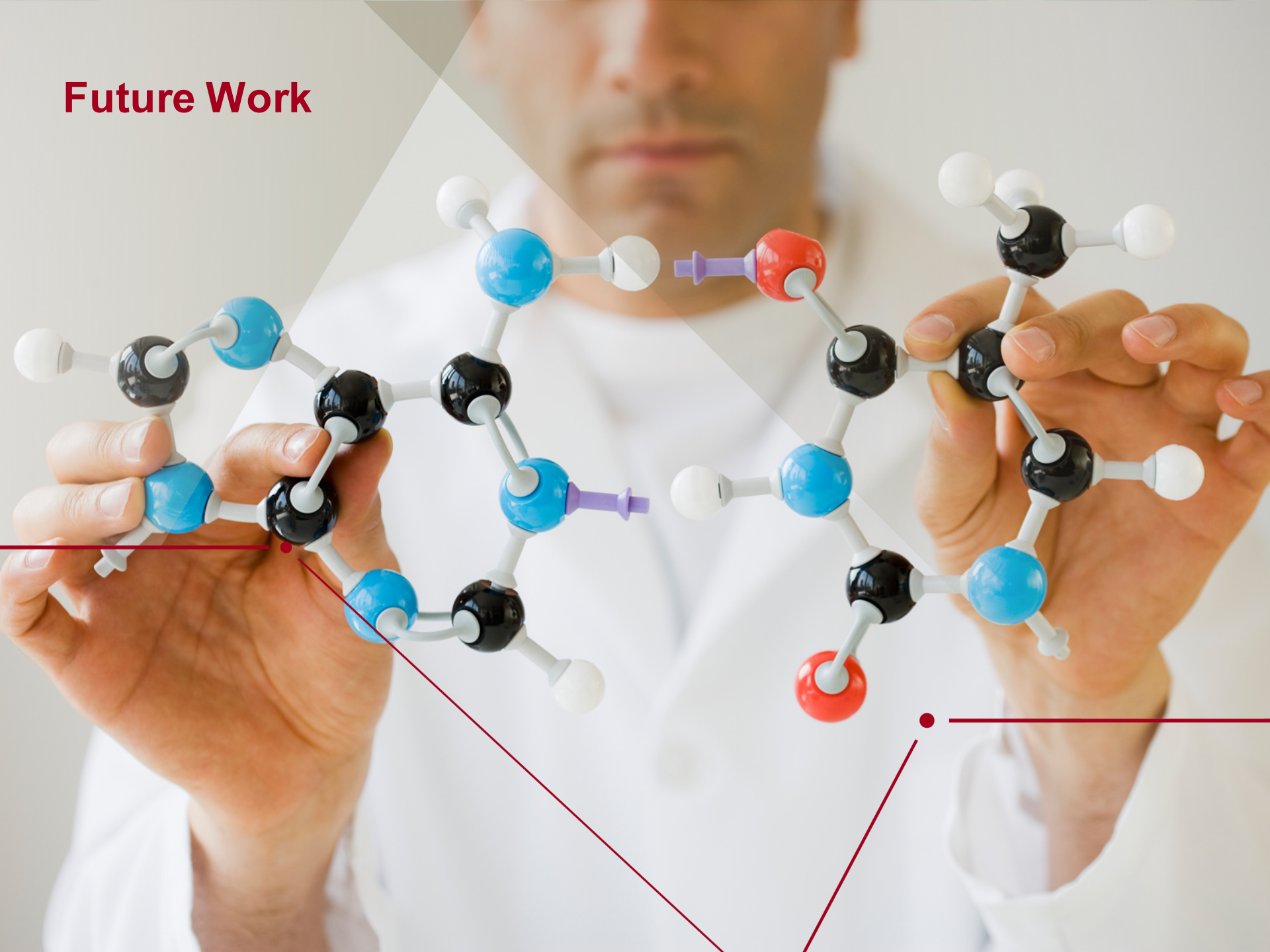
Guided Back Propagation

SLAC

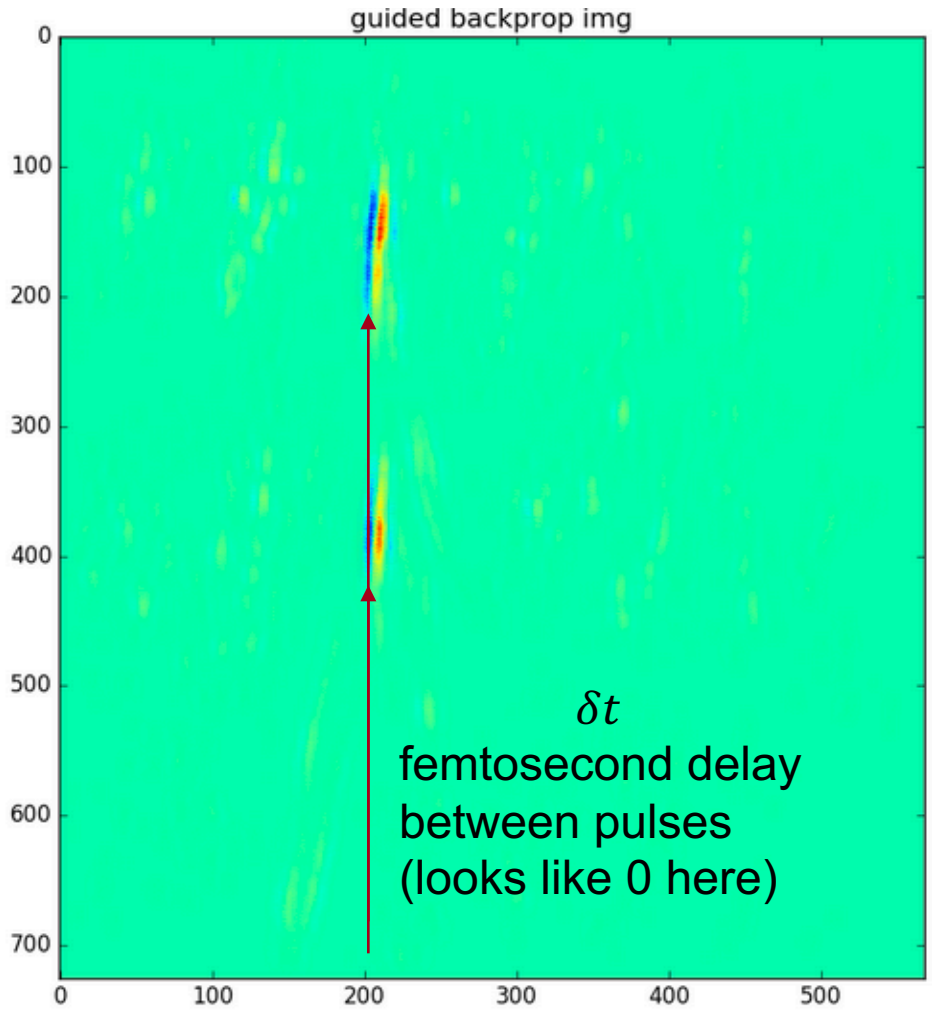




Future Work



Guided Back Propagation for Image Processing?



Additional parameter we want to be able to sort down on

Guided Back Propagation for Discovering Science?

Work with Chuck Yoon

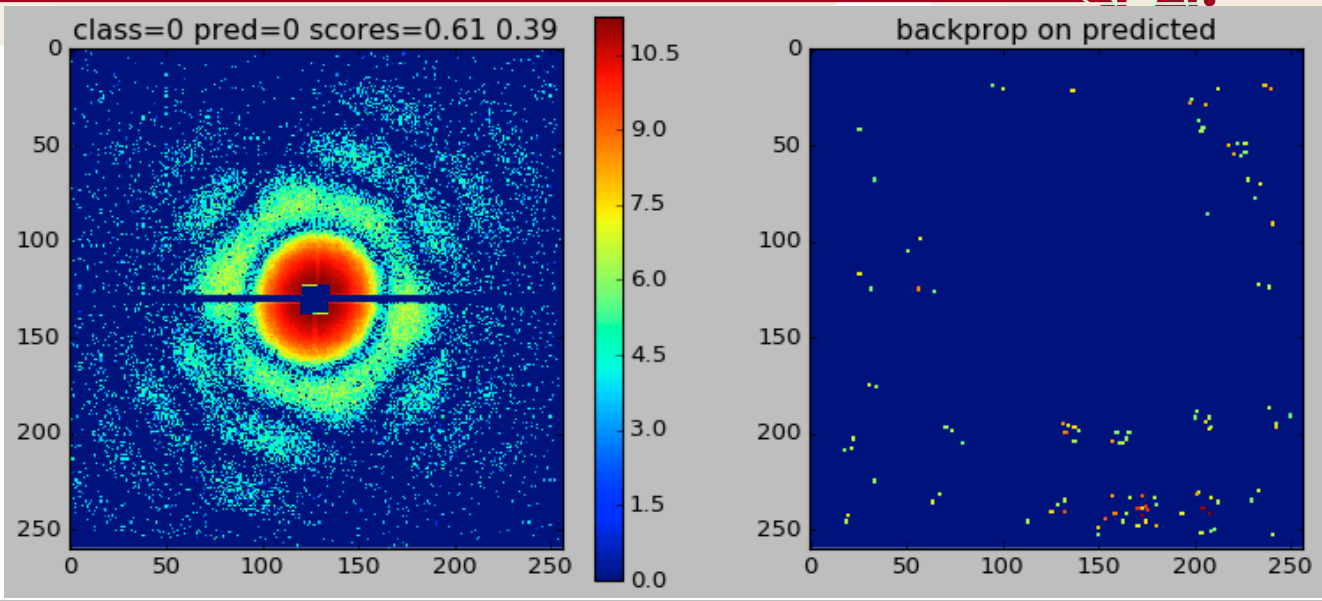
Diffraction patterns for two virus (log scaled). Science problem: understand differences

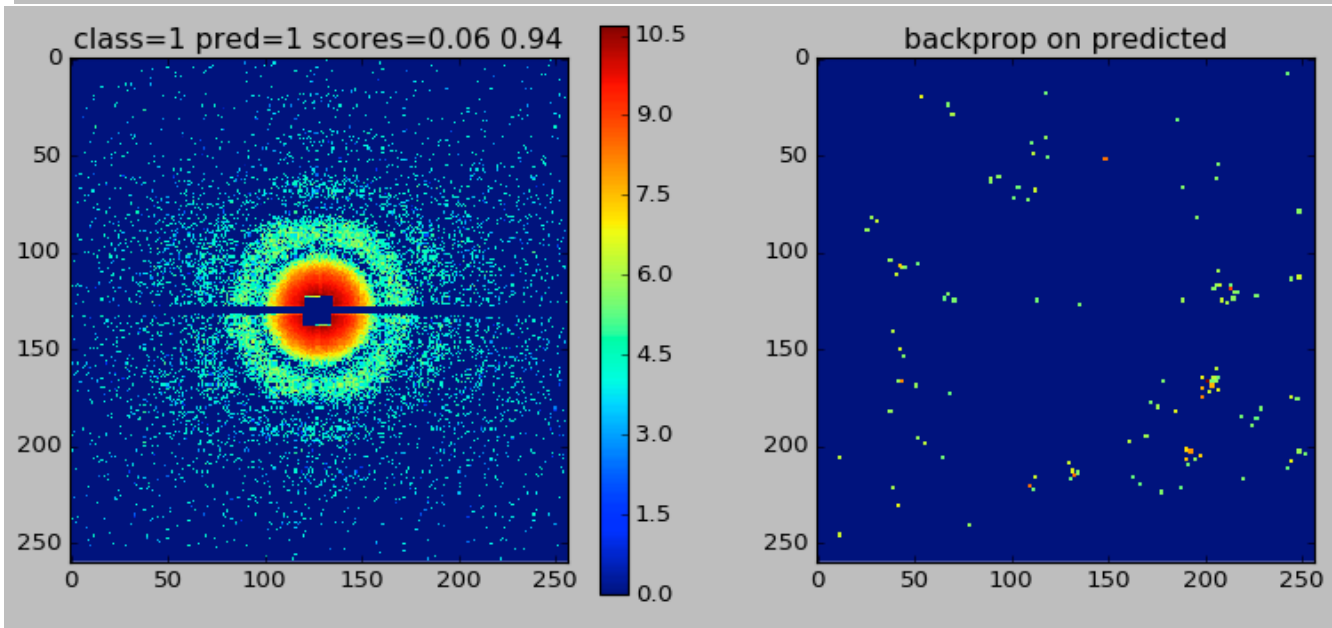
Can backprop help?

Plotting top 50% of backprop

Success here: Adapted to new problem in two days:

Trained deep model
Produced backprop





Supervised Learning: train "Splitter" from LCLS reference vs signal shots

SLAC

Input from reference shots (no lasing, dark run, background, etc.)

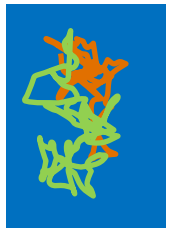
Deep Learning Splitter

Output A (signal):

A=0

Output B (ref):

B=Input



Input from signal shots (lasing, run with samples, etc.)

Deep Learning Splitter

Output A (signal)

A+B=Input

Output B (ref)