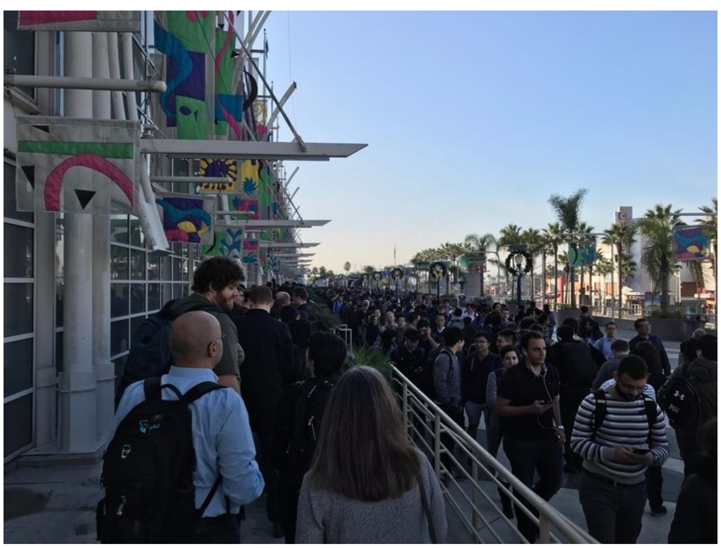
NIPS 2017 Conference Overview

Karl Ni (L41), Cory Stephenson (L41), Jeff Hetherly (L41)

NIPS is the largest artificial intelligence, machine learning, and neuroscience conference. Topics range from applied computational algorithms to theoretical proofs of performance bounds, to demonstrations of capabilities, and much more. This is the premier “data science” conference. To provide perspective, annually, Google’s DeepMind sends their entire organization from London to attend. In addition, there are technical booths, which spans two large halls, where recruiters from all organizations (but especially the large corporations) stack the hall in a lavish display to attempt to recruit top talent, sometimes offering on the spot.

It is a spectacle of money, brains, egos, and adventures. Registration takes 2.5 hours to make it to the front. Each hall is staffed with volunteers and security guards ensure that latecomers can’t enter (registrations sold out a couple weeks after being announced.) Waiting in line, there is gossip of Andrej Karpathy’s unveiling at Tesla’s party, and exchanges of which performer will be at which company’s NIPS party. Apparently Flo’Rida will be performing at Intel’s, almost to parody the parody show Silicon Valley.



General Trends

General trends that we've noticed, Bayesian Deep Learning and some movement toward computational reduction and distributed deep learning. Impressively, the highest cited type of work at NIPS was *not* deep learning. There was a move to bring it back towards neuroscience, since the original intention of NIPS was exactly that...to the chagrin of everyone there.

Terry Sejnowski was there greeting many people (and a short conversation revealed the above feelings), and remarked that this conference at over 8,000 registrants. Included at this conference for the first time ever were some competitions with a chatbot, a jeopardy style QA, a robotics-type learning to run thing, drug finding, and some others.

Basic stats, revealed some astounding numbers. For example, there were 3240 submissions this year, which the organizers bragged was 2x more than icml. In total, 156 subject areas. 12% industry. 679 for presentation, overall acceptance rate at 21%. 43% were posted on arxiv, 10% reviewers saw them. Conditioned on not posting online at arxiv, 15% were accepted. Conditioned on being posted, this jumps to 29%. 35% acceptance rate when posted online AND reviewer saw it.

Conference Day 1

December 4, 2017

Tutorial: Deep Probabilistic Modelling with Gaussian Processes

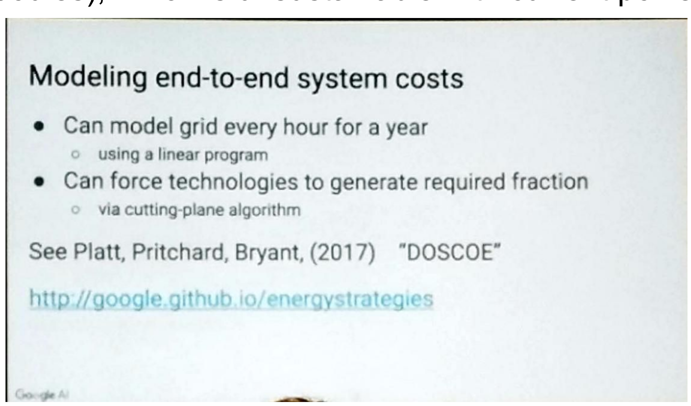
In the first part of the tutorial, Neil Lawrence gave a an overview of the various sources for uncertainty and why it matters for prediction problems. He then introduced a single layer feedforward network and described it with a probabilistic framework. This was used to motivate the use of gaussian processes as they allow for easy computation of the high dimensional integrals required to do inference with the model.

In the second part of the tutorial, he builds on this to compose these into deep gaussian processes (DGPs). There are a few advantages to doing this. For one, the derivatives of this process (if they exist) are also gaussian, and secondly, for particular covariance functions, the model is a universal approximator. However propagation of probability distributions through nonlinear functions still remains a problem, as normalization becomes intractable. Therefore, variational methods are typically used to fit DGPs.

Sidenote: Neil Lawrence doesn't like calling this AI, he instead says "algorithmic decision making"

Keynote Address: John C. Platt - Google Applied Sciences

Google is apparently concerned with energy consumption. By the year 2035, we will be using 0.2YJ (Yotta Joules), which is unsustainable with current power growth usage.

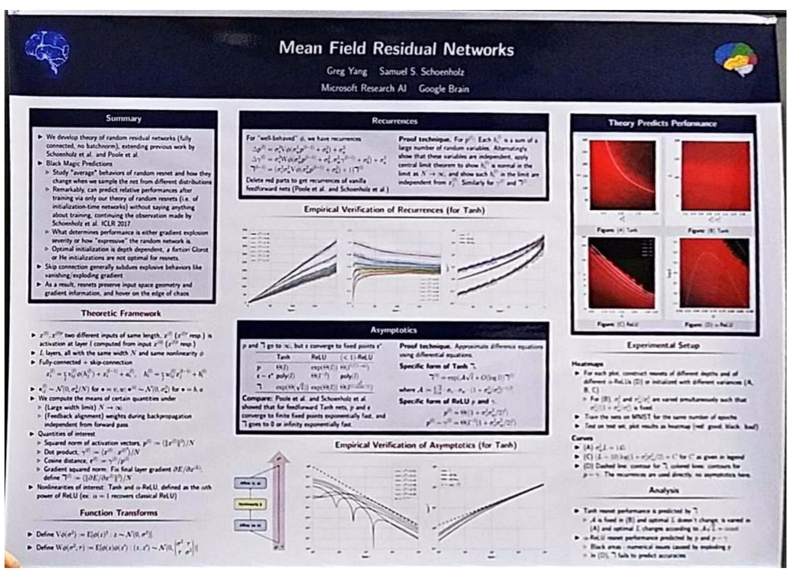


For this reason, Google is getting into the fusion business, and they're teaming up with a company by the name of TAE. Their contribution is applying machine learning to accelerate their processes. TAE brings to the table: rapid experiment (they can do one every ten minutes) Their data suggests good scaling.

One interesting note is that they refer to one of their MCMC with human preference algorithm as the "optometrist algorithm", an interesting yet descriptive name. Google is also using Tensorflow to optimize this in parallel (10 GPUs) with SGD. This is interesting. I'll restate: Google is using machine learning software packages (like Tensorflow) to optimize traditional and classical Bayesian probability processes.

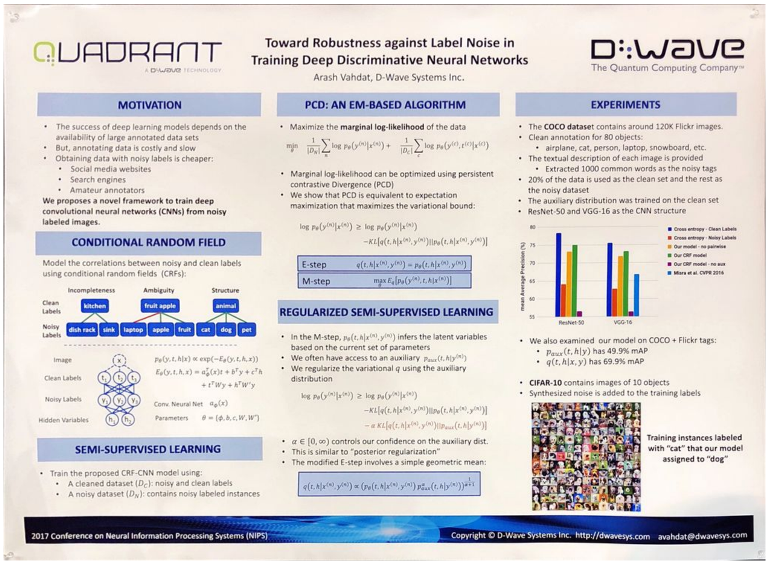
Poster Sessions:

Below are some of the posters that we were able to snap via camera. For a list and brief description of more papers, see the appendix at the end of this report.



Effectively, the speaker was saying, give me some hyperparameters, your network architecture, and he'll find you the optimal parameters for it.

D-Wave also had a poster at the sessions:



Connections on Day 1:

NVIDIA Representative: Platform enables deployment of models on hardware the size of a credit card with 7.5W power consumption profile. See http://www.nvidia.com/object/embedded-systems-dev-kits-modules.html

Of note: many Bayesian Inference, now being married to neural networks. Also, diminished number of US students in the area. It's quite a diverse crowd with the South Korean universities making a splash. On another note, I've built my own Google Deep Mind detector. Feature: are there lots of people lined up to see the poster? Must be Google...

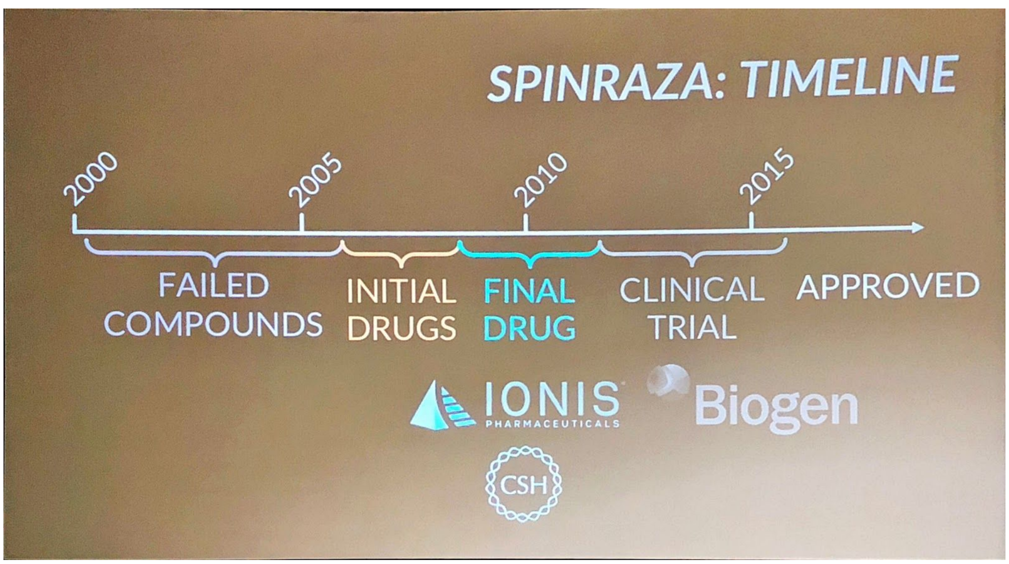
Conference Day 2

Keynote: Brendan Frey

Deep Genomics

Unethical to give placebos for drugs. Within a couple of months, Europe (which approves a lot slower), Spinraza provided a cure for SMA. Spinraza: a digital medicine, where the medicine is specified *digitally*. Spinal muscular atrophy (SMA) is boiled down to a mutation in the exon, where a C is mutated to a T. What happens is that this protein is ignored?

How is this fixed? First thing that pops into your mind is probably CRISPR CAS9, which is only a damage repair. It's a bit more loosy goosy. First these nucleotide sequences are hard to pinpoint. Secondly, using CAS9 might make changes all over the place (off-target effect). So, we can't just use CAS9. What can we do? Well, we can make a DNA sequence that way down the way, the medicine will bind to the intron and then it'll be used. This is the definition of digital medicine. The medicine is literally a nucleotide sequence that you specify.



The drug Spinraza has saved lots of lives, but it costs a lot. Now, they want to speed up the first few years (the failed compounds) and search for the good one. This is done with modularity, their A.I. Platform. With digital medicine, if we know the mechanism we're trying to achieve, and we can simulate the effects of RNA, we can design in silico the medicine. With Spinraza, they came up with 200 compounds, and Spinraza was the 3rd from the top. Afterward, they test in the wet lab, but they needed to test toxicity (is there unintended consequences with the drug.

They have a cloud laboratory. They upload a Python script, it specifies the experimental protocol. Then, robots conduct the experiments. A lot of kinks need to be worked out, but human accuracy isn't as good as people think it is.

Last night, there was a lot of talk about deep dream, and a display of deep art providing wild psychodelic medicine. Probably not what you want in genetics, where you modify to get some nightmarish Frankenstein. The idea would be to use adversarial machine learning to figure what you don't want.

Test of Time: Ali Rahimi

[Video Link]

Rahimi likened current neural network research / deep learning research to alchemy. This is the belief that there is not a lot of rigor applied to current deep learning theory. He believes there are problems in naively applying gradient descent to optimize nonlinear models. Overall, he stressed the importance of more experiments aimed at deducing causes of 'strange phenomena' when they are observed, rather than blindly chasing improvements in an evaluation metric. His talk was well received…except for [Yann LeCun].

Morning Sessions:

Diffusion Approximations for Online Principal Component Estimation and Global Convergence

Online PCA has much better time and space complexity. Oja's iteration escapes from unstable stationary points and converges. Online PCA is statistically optimal and globally convergent. Further directions: principle subspace learning, parallelizing PCA for online data

Positive-Unlabeled Learning with Non-Negative Risk Estimation

Fully binary supervised task: both positive and negative examples (PN learning). Example when negative class is not available (click-through). Unbiased PU learning: model risk of unlabeled data as mixture of positive and negative labeled data. Unbiased, PU can be better than PN. Risk of overfitting with PU learning with DNNs (negative risk at some point). Non-negative PU learning: consistent and bias decreases exponentially, can be more stable by introduction of 'max' function. Non-negative PU learning shows better performance than unbiased PU learning

Afternoon Sessions:

Upper bound the generalization error via the expectation of the mutual information

The less information extracted from training data, the less potential for overfitting. This all depends on the ingredients by learning problem. They'll get a tight upper bound....highly theoretical formulation by using a concentration inequality for the generalization error.

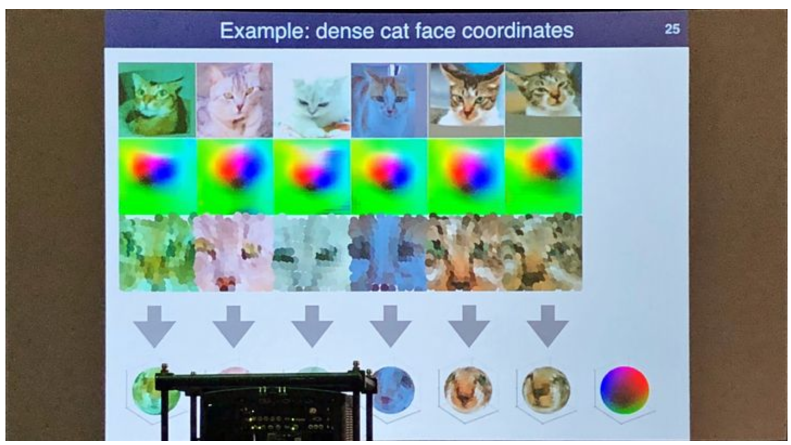
Net-Trim Convex pruning of Neural Networks.

The idea is to remove connections from deep learning. They'll get a pretty sparse connected algorithm. Kind of like stuff that Prof. Chugg at USC is looking at. The algorithm is post training, and has performance guarantees (fogettabout those Squeezenet stuff....that's all alchemy!!!). Looks like they're sparsifying layer by layer with the L-1 norm, and doing additional convex relaxation applied at each layer. Code available at https://github.com/DNNToolBox/Net-Trim-v1

Unsupervised Method of Object Detection

Idea: project image to equivariant image space, which has a correspondences to an 3-D space. He calls this space image sphere (which is really a pretty large space but just normalized). LoL, lookup "mathematical dog", which turns a dog into a sphere. Use some warping functions (or if you have the equipment for it, get more data). Avoid the degenerate case of mapping everything to a single point in the sphere with some "distinctiveness" constraints. This is all tied together with a Siamese neural network.

It seems to generalize really well. He showed a bunch of cats and they all mapped to the same places on that sphere. Then, he visualized the sphere.

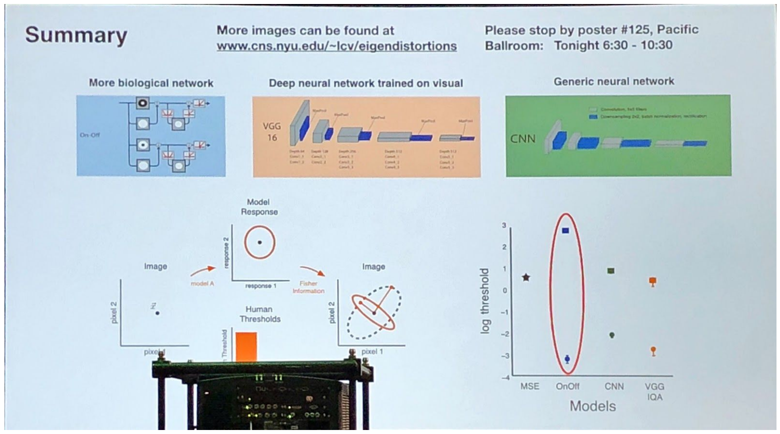


Two attractive features of this method: global consistency emerges from local correspondences (!) and no manual supervision is required.

Quantifying the Visibility of Image Distortions

For those of us who have to answer, "What are your metrics?" And also to those of us who hate the MSE. Let's use "visual physiology", and go into neuroscience. Apparently our visualization system is modeled heavily, e.g. luminance gate controls, neuronal rendering. The adversarial examples/additions kind of disprove DNN's as human visual modelers.

Instead, in this talk, the DNN models directly human sensitivity to visual distortions. To do this, there's a Ponamarenko et al, 2008 dataset on human perception. That is, they're finding a function that models the distortions themselves rather than visual predictions. Their contribution, how do these models generalize beyond database examples?



Model Interpretability
Two general purpose methods were showcased for model interpretability. The STREAK method is geared towards NN specifically and is substantially different than the popular LIME method. STREAK showed much better runtime performance and equal or better explanatory power than LIME. Worth looking into for visual tasks. SHAP is a modified and more principled version of LIME (e.i. better regularization and kernel definition). It works through additive feature attribution. Performance is slower than LIME, but better than previous Shapley methods. github.com/slundberg/shap

Connections on Day 2:

Talked with NSA R6 about meeting up for dinner in the evening. They have a set of interesting problems that we could take a look at. The next time we are out on the East Coast, we're welcomed to a meeting with the R group.

Introduced Gregory Way about some opportunities at Lab41. He seems to have a mindset about using Lab41 as a stepping stone to his dream of doing machine learning for the betterment of biomedical research. Gregory's interest is in genomics.

Talked with Carnegie Melon University students. Their problem sets are somewhat interesting, but they're very much an audio group. Too bad they're not US citizens.

Conference Day 3

Morning Sessions

Bayesian GANs
Mode collapse in GANs is reminiscent of overfitting, so a bayesian approach to combat it makes sense. Given some reasonably relaxed priors over discriminator and generator parameters, the regular GAN is recovered with MAP parameters. They claim the bayesian GAN achieves SOTA on a <1% of the labels used in other approaches.

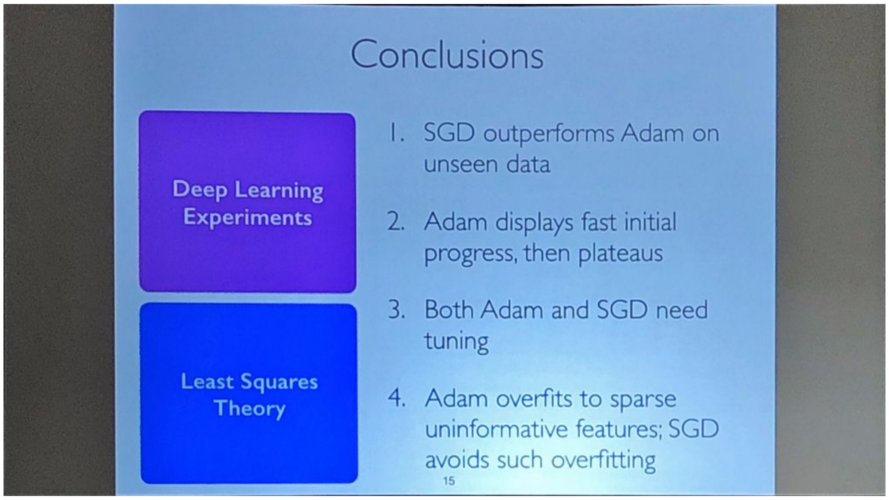
Dueling GANs
Alternating gradient descent steps between the generator and discriminator leads to instability. To combat this, they convert the min max problem in the regular gan to a max max problem. This can be done since the discriminator loss is convex, and so a dual loss can be computed.

Lipschitz Constant Bounding Generalization Error
The Lipschitz constant, I guess he's too fancy to say the slope-i-ness, of networks chosen by SGD apparently correlates with train error - test error, and stops increasing when normalized by margins. (BTW, everytime I hear "Lipschitz", I giggle.) What's fancy is that this generalization bound, which is effectively a complexity term, factors into test error. So, test error < train error + complexity term (some function of Lipschitz over margin)

Invited talk: Deep Learning for Robotics - Peter Abbeel
Peter Gave an overview of recent trends in deep learning in

Afternoon Sessions

Adaptive Gradient Optimizers? Don't Bother



From some blogs, "Few days ago, an interesting paper titled The Marginal Value of Adaptive Gradient Methods in Machine Learning (link) from UC Berkeley came out. In this paper, the authors compare adaptive optimizer (Adam, RMSprop and AdaGrad) with SGD, observing that SGD has better generalization than adaptive optimizers."

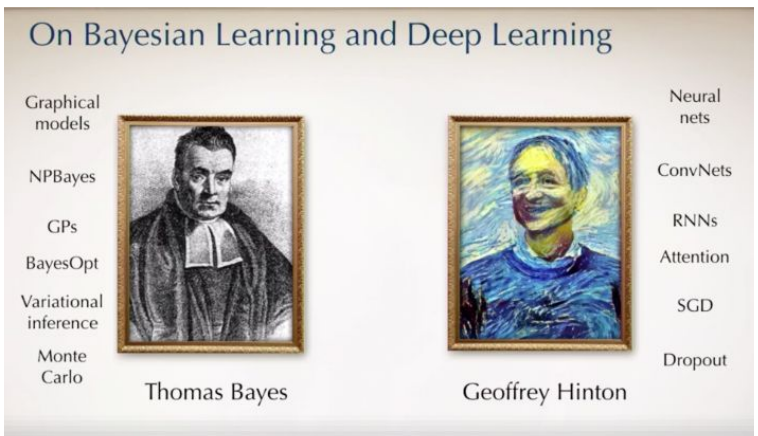
"We observe that the solutions found by adaptive methods generalize worse (often significantly worse) than SGD, even when these solutions have better training performance. These results suggest that practitioners should reconsider the use of adaptive methods to train neural networks."

Day 4

Keynote: Bayesian Deep Learning

Yee Whye Teh

This talk was concurrent with the Laboratory All-Hands. This was a broad overview of techniques (which has become quite popular) for Bayes in deep learning. On a side note, it appears that Professor Hinton has now achieved the likes of a very recognizable character from Dr. Teh's slides:



From notes that I have on the eminent Yee Whye Teh (which I had to glean from his website), he provides an overview of Bayesian methods used in deep learning on three areas:

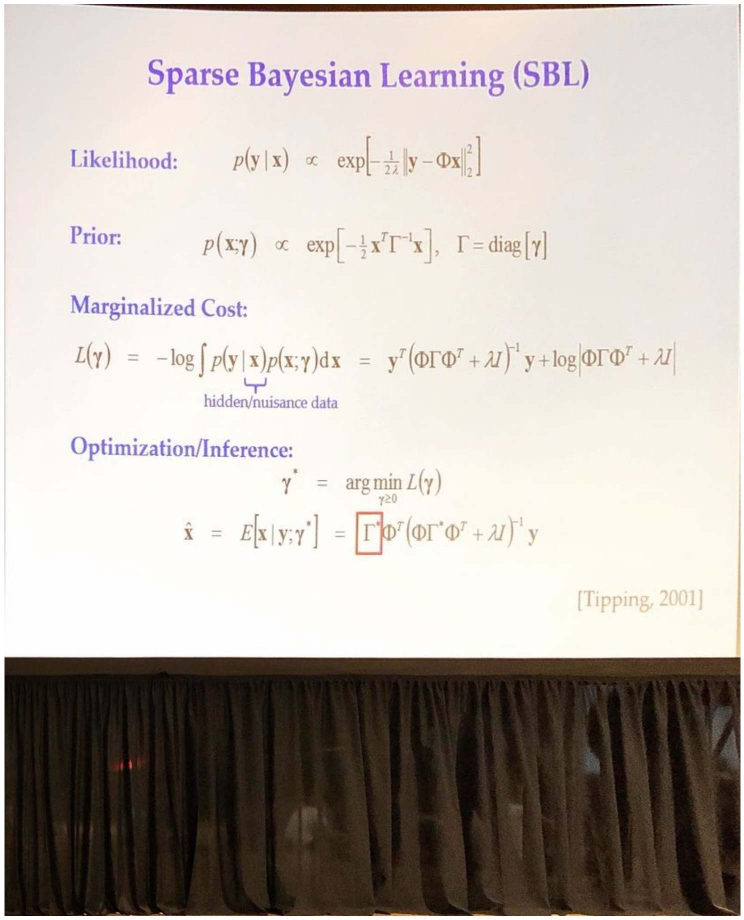
1. Distributed Deep Learning - YWT terms this as the "Posterior Server", which is aptly named. Here, you can think of the posterior probability of a function you're trying to predict as living on a parameter server, and the worker nodes are all optimize the parameters.
2. The Concrete VAE - The thought is to ingest some a priori knowledge into the autoencoder process, though I'm thinking that YWT was thinking about more broad applicability than just VAEs, here.
3. Filtered Variational Objectives (FIVO) - Well, you may have heard of ELBO when people have spoke about VAE's, but FIVO's where it's at. Apparently FIVOs have a tighter bound on the estimator of the marginal probability, and this tightness is related to the variance of the estimator, which you specify.

Online resources for this particular topic are numerous, and even here at nips, Bayesian Deep Learning is resurgent. But YWT pointed to a few toolboxes, of which I have heard of a couple only recently (though I'm sure Jeff & Cory have heard well before me.) These include Edward, Bayesflow, Pyro, and ProbTorch.

In addition, just to show the widespread adoption, there are two workshops on Friday and Saturday on this very topic. Too bad we didn't spring for workshops :-P. Then again, we're exhausted.

Afternoon Sessions

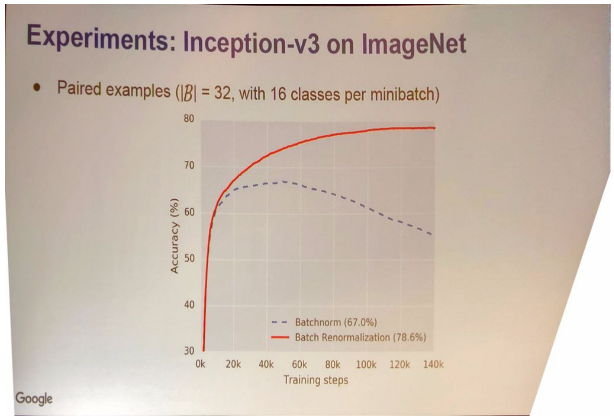
Sparse Bayesian Learning



Interesting way to optimize sparse cost functions (anything with L0 norm regularization). Using some iterative algorithms, can effectively model stuff even if correlation structure has really high off-diagonals.

Batch Re-Normalization

Spurred on by the success of batch normalization, there were some efforts to make batch re-normalization happen. Apparently batch normalization is uses a biased estimate of mean and variance, and this affects performance at inference time. They have performance curves that look like:



Could be legit….but could be that they're overfitting or doing something weird. That's a pretty weird curve.

Connections on Day 4

New York University, Neuroscience Group and Yann LeCun group lunch. Didn't get to talk to the big guy much, but certainly spoke with a few guys on computer vision methods. They're doing really exciting stuff. One of the things that I overlooked (which is also in this document) is the computer vision deep learning methods based on perceptual objective functions with biologically inspired architectures. They were telling me that they could achieve way more compression.

Dinner with Professor Keith Chugg. We have a short connection coming up in about a half hour where Professor Chugg is driving all the way up from the University of Southern California. His group has done some impressive work on FPGA's to compress neural networks into low power algorithms at the densest layers. Not sure he's thinking of a business model, but it jives with a lot of our needs to make sparse algos.

APPENDIX

Additional Papers

Day 2 - Additional Papers

Inhomogeneous Hypergraph Clustering with Applications

Graph clustering: grouping nodes. Hypergraph clustering: higher-order relationships

Safe and nested subgame solving for imperfect-information games

Approximation Bounds for Hierarchical Clustering: Average Linkage, Bisecting K-means, and Local Search

Which hierarchical algorithms are appropriate for a given task. Dasgupta's objective function. Could lead to better algorithms used in practice. New addition to Dasgupta's objective makes linkage graph invariant to partitioning. **Open question**: can we develop optimal trees that aren't binary.

Communication-Efficient Stochastic Gradient Descent, with Applications to Neural Networks

How to train very deep models? Data-parallel SGD (usual way): exchange gradient takes a long time. Exchange compressed gradients. Quantize gradients before exchange. Randomly quantize gradients. 2x speedup and as good or higher accuracy for common networks

Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results

Few labeled samples. Two predictions: student and teacher. Distort student's input. Exponential moving average of weights for teacher. Both improve each other. Works with any supervised model. Add consistency cost with exponential moving average

K-Medioids For K-Means Seeding

The idea is to place CLARANS k-medioids algorithm in between the initialization step and alternating clustering/maximization step in k-means. The improvement shows improves runtime of CLARANS and marked improvement over "vanilla" k-means (K++ initialization). There is code on Github for improved CLARANS

Online Learning with Transductive Regret
No distributional assumption is assumed in online learning. Worst-case analysis. Mixed training and test phases. Issues: performance measure, static measure not robust against modifications. "Efficient" algorithms with "guarantees"? Use weighted finite state transducers. Algorithm operates in N^2 time

Robust optimization for non-convex objectives
The idea is to train neural neural networks that are robust to many different kinds of noise (gaussian, adversarial). Given a distribution over the noise, they calculate the expectation of the loss and minimize it.

Bayesian optimization with gradients
The goal is to use gradient information to speed up bayesian optimization. To do this, they propose a new acquisition function and show that it outperforms EI. They maximize their acquisition function with SGD.

Implicit regularization in Matrix Factorization
In matrix completion, many trivial solutions exist. However finding a solution with SGD seems to result in solutions with good generalization properties, meaning there is some implicit regularization.

Constrained GLMs
GLM Regression under distance-to-set penalties is a generic algorithmic framework for GLM estimation under constraints. It uses iterative majorization-minimization (MM) and shows very nice accuracy performance on challenging constrained GLM problems.

Towards Accurate Binary Convolutional Neural Network
To deploy networks on the edge; binarization may help. For instance, it would allow us to replace floating-point operations with bitwise operations. This talk explains how to squeeze approximate full precision by several additive shifts (additional bases).

Poincare Embeddings for Learning Hierarchical Representations
Exciting talk from FAIR on learning hierarchical embeddings using a Poincare loss and special derivative. This is useful when symbolic data exhibits an underlying hierarchy (several use cases). They use an interesting loss metric: Poincare distance, which is a natural metric for hierarchy in the embeddings.

Deep Hyperspherical Learning

This talk introduced a scale-invariant convolutional architecture they call SphereConv. They claim that projecting the problem onto a hypersphere alleviates many training difficulties and leads to faster convergence with comparable accuracy to conventional convolutional networks.

Day 3 - Additional Papers

TernGrad: Ternary Gradients to Reduce Communication in Distributed Deep Learning

* Usual distributed method: data parallelism
  - communication is the bottleneck
* can be optimized by quantizing gradients into ternary set {-1,0,1}
  - can be interpreted as a form of regularization
  - shows very competitive accuracy with standard SGD with substantial savings in training times
  - specialized hardware for this ternary approximation was brought up during questions

Train longer, generalize better: closing the generalization gap in large batch training of neural networks

* thesis goes against current trend of smaller batches, small learning rate
* When training with larger batches, scale the learning rate (~sqrt(k))
* showed interesting variable: distance from initialization for weights
* they found an analogy to ultra-slow diffusion: "random walk on a random potential"
* suggests not to do early stopping?!?
* need to read this paper!

End-to-end Differentiable Proving

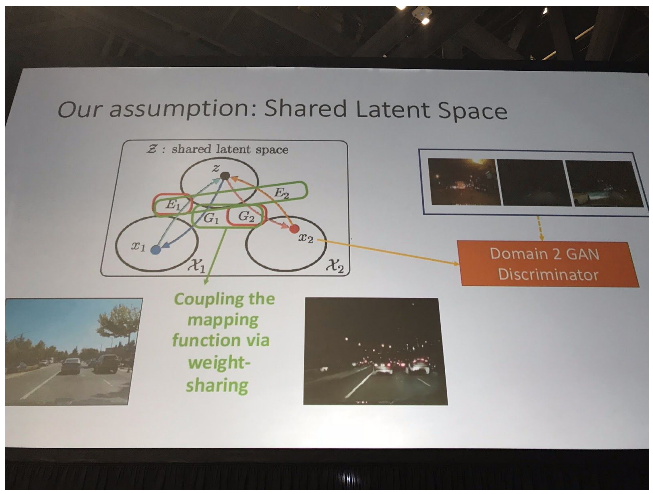
* NN for proving queries to a knowledge base
* really cool, but a bit too preliminary for practical use

Gradient descent GAN optimization is locally stable

* non-linear systems theory
* adds a "dampening term" to the discriminator loss
* Assuming continuous time gradient descent dynamics, the global equilibrium (if it exists) is stable at least locally.

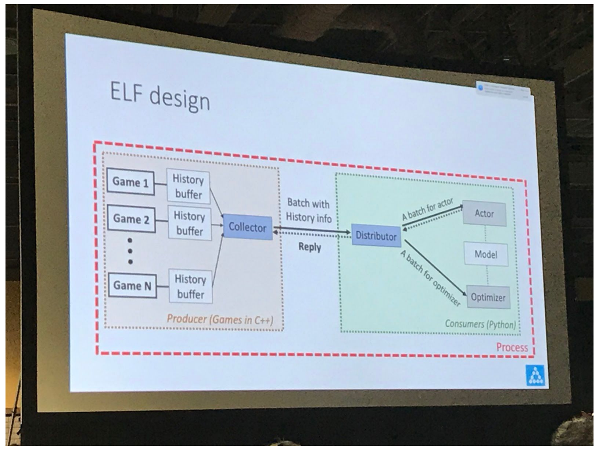
Unsupervised Image-to-Image Translation Networks

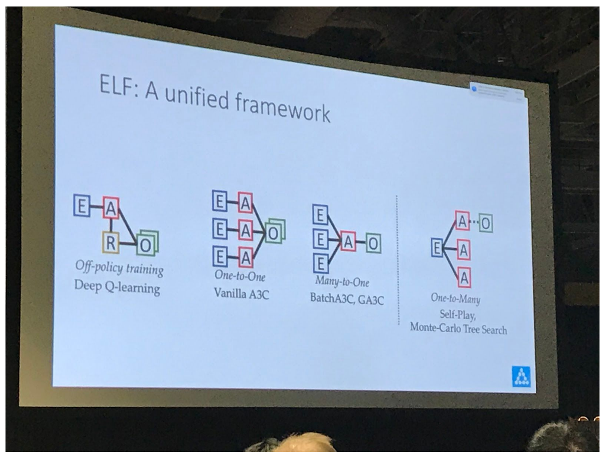
* estimating the joint distribution of images in different domains by using samples from the individual marginal distributions
* coupled-GAN framework



ELF: An Extensive, Lightweight and Flexible Research Platform for Real-time Strategy Games

* extensible framework for training any RL agent within a game with a C++ interface
* shows better performance (CPU) than OpenAI Gym
* self-play with multiple agents





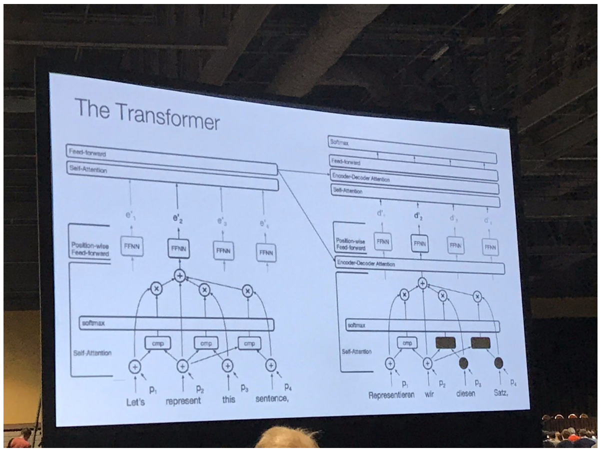
Imagination-Augmented Agents for Deep Reinforcement Learning
* agents with planning capabilities: unsupervised learning + reinforcement learning
* constructs an environment model and trains simulation policy (e.g. Monte-Carlo policy rollout)
* environment is learned through unsupervised learning, simulation policy is learned through rollout
  - this increases efficiency and performance
  - able to deal with imperfect models

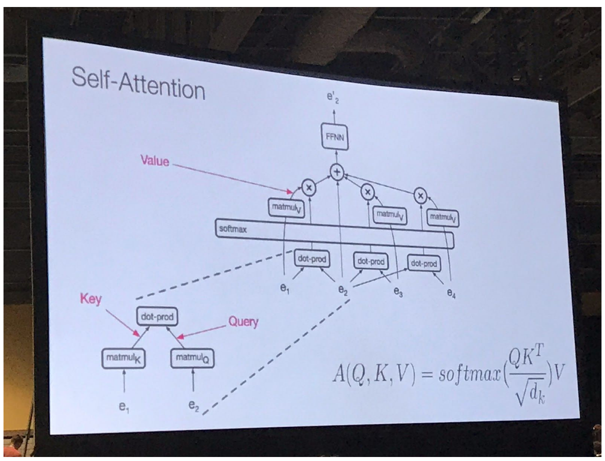
A simple neural network module for relational reasoning

* Relation Networks: simple CNN for images + LSTM for question text -> MLP -> answer

Attention is All you Need

* self-attention
  - constant 'path-length'
* the transformer





* attention is "cheap"
* faster and better performance than ConvSeq2Seq (FAIR)

Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles

* non-Bayesian method of uncertainty
* "proper scoring rule" with "adversarial training"

Inverse Reward Design

* reward design is difficult and, if done poorly, can lead to undesirable behavior
* proposal: uncertainty + risk aversion

* Which part of the reward function should be considered uncertain?
* Key idea: rewarded behavior has high true utility in the training environment
* Results show that agents tend to avoid situations it hasn't seen at training (risk aversion)
  - could be too conservative

Dynamic Safe Interruptibility for Decentralized Multi-Agent Reinforcement Learning

* interruptibility allows a supervisor (human) to interrupt the agent's behavior
* interruptions should not prevent exploration and not reinforce undesirable behavior

Learning multiple visual domains with residual adapters

* summary: share most parameters but have additional parameters conditioned on a specific task
* at training: a single multiplicative factor per layer
  - scale appropriately
* at inference: train small network to decide which task to solve
* https://arxiv.org/abs/1705.08045

Fair Clustering Through Fairlets

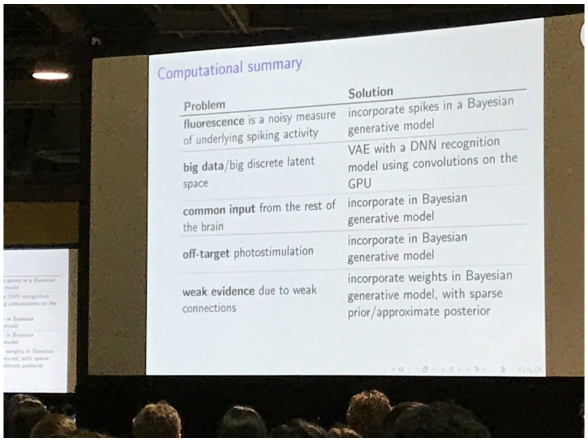
* Fairlets are a way of linking representatives from each class into locally connected objects
* them, randomly sample a representative from each fairlet and then cluster (with k-means, etc.)

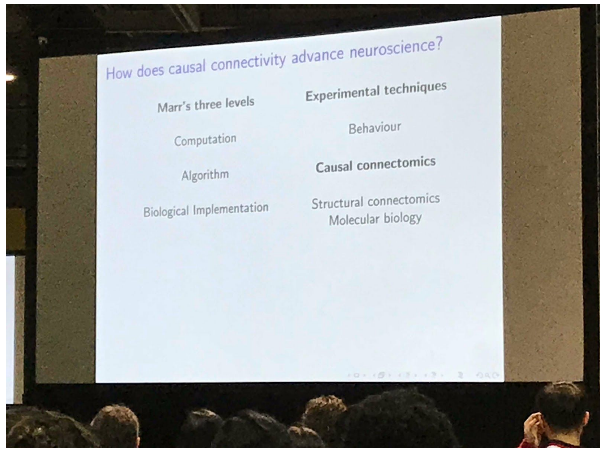
Language Style Transfer
Hard to really call it style, but they're essentially doing to text what we're doing with images. They had some fun examples.

Day 4 - Additional Papers

Model-based Bayesian Inference of Neural Activity and Connectivity from All-Optical Interrogation of a Neural Circuit

* very difficult to measure actual connectivity of neurons (laser stimulation)
  - dendrites are very spread out (spurious activations from laser)
* used a VEA to model latent variables related to fluoresces of activated neurons

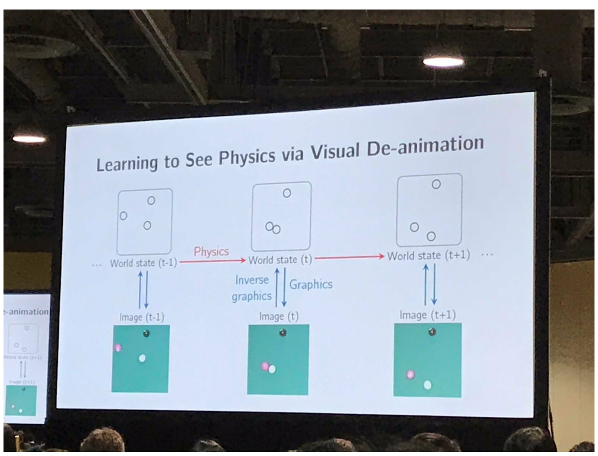
* 

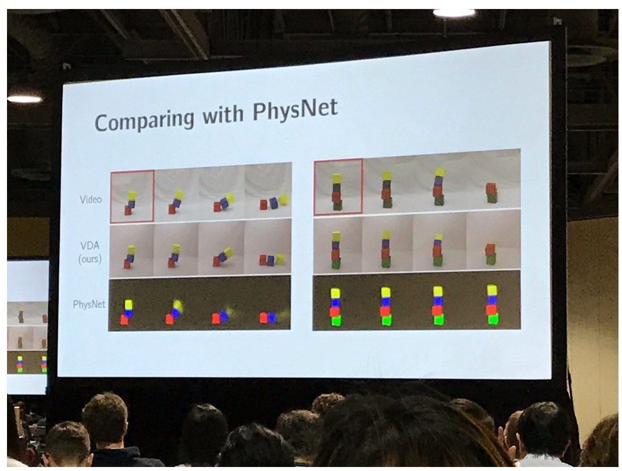


* summary: a casual algorithmic model of neuronal connectivity is an open question with promising research actively pursued

Scene Physics Acquisition via Visual De-animation

* humans can recover rich information from videos - > how can we abstract and model this (inverse graphics)
* combine the efforts of simulation engines and inference power of DNN

* 

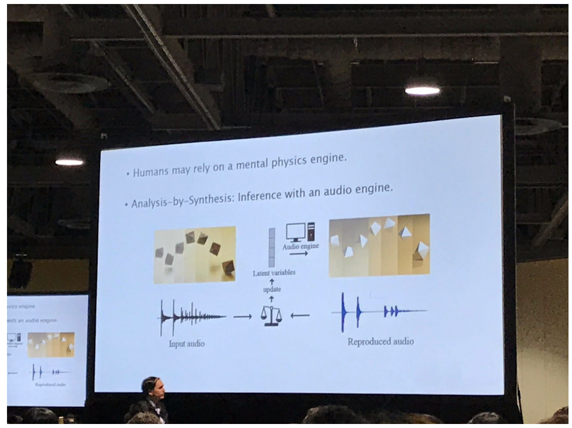


* shows better performance (computational and accuracy) than end-to-end DNN systems

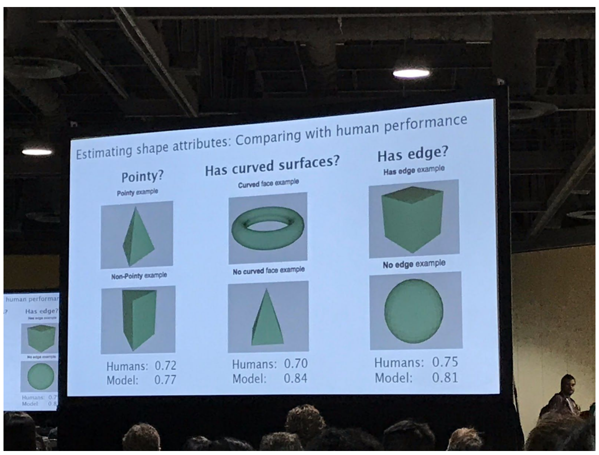
Shape and Material from Sound

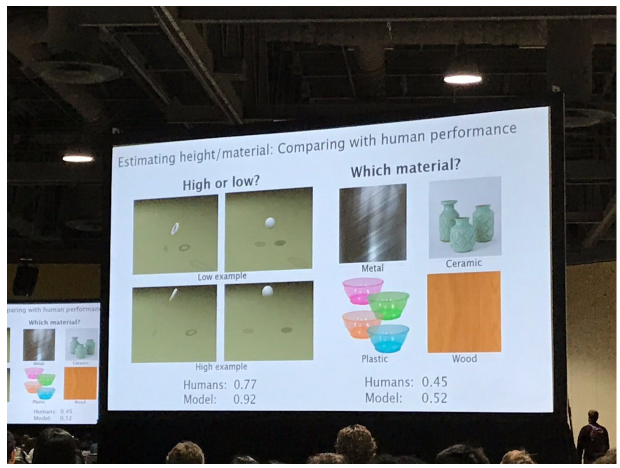
* infer the shape and material of an object from acoustic interactions

* 



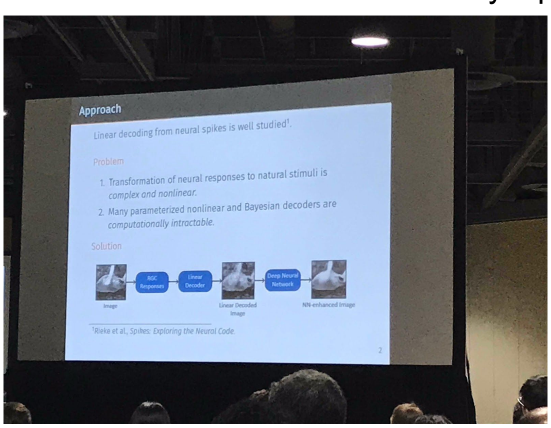
* models show superhuman performance on both shape and material inference

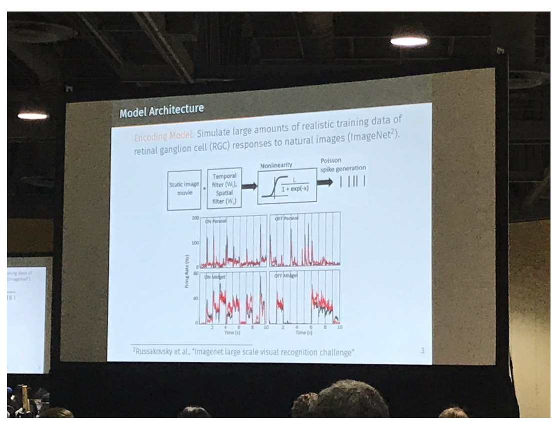
* 

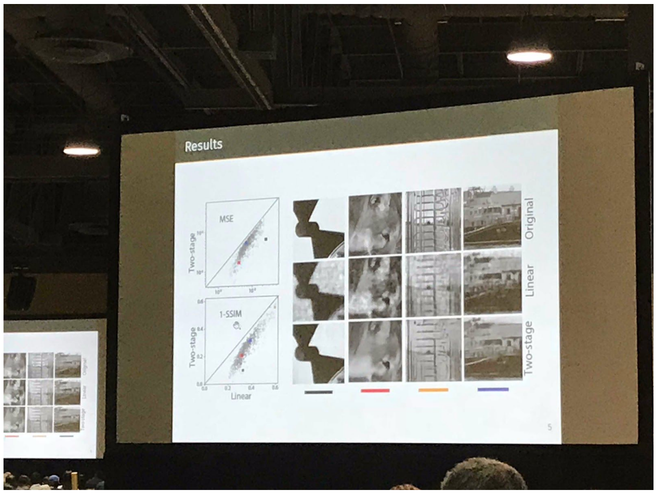


Deep Networks for Decoding Natural Images from Retinal Signals

* how does the brain encode sensory inputs?

* 



* 
Results
MSE
Two-stage
1-SSIM
Two-stage
0.0
0.2
0.4
0.6
Linear
Original
Linear
Two-stage
5