

# Deep Learning in the Real World

Soumith Chintala

# Approaching a Problem

# Approaching a Problem

- $\text{output} = f(\text{input})$
- what is input?
- what is output?

# Approaching a Problem

- $\text{output} = f(\text{input})$
- what is input?
- what is output?
- How much "prior" can you encode into your neural network...

# Approaching a Problem

- $\text{output} = f(\text{input})$
- what is input?
- what is output?
- How much "prior" can you encode into your neural network...
  - The more prior you encode, the lesser data it needs further

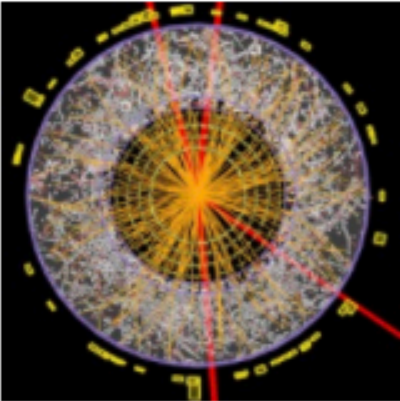


# Approaching a Problem

- $\text{output} = f(\text{input})$
- what is input?
- what is output?
- How much "prior" can you encode into your neural network...
  - The more prior you encode, the lesser data it needs further
- Choose the right loss function

# Approaching a Problem

- $\text{output} = f(\text{input})$
- A lot of times, simple = better

# Some scenarios

17 Active Competitions		
	<b>TrackML Particle Tracking Challenge</b> High Energy Physics particle tracking in CERN detectors <b>Featured</b> · 3 months to go · physics, tabular data	<b>\$25,000</b> 222 teams
	<b>Avito Demand Prediction Challenge</b> Predict demand for an online classified ad <b>Featured</b> · a month to go · tabular data, image data, text data	<b>\$25,000</b> 956 teams
	<b>CVPR 2018 WAD Video Segmentation Challenge</b> Can you segment each objects within image frames captured by vehicles? <b>Research</b> · a month to go ·	<b>\$2,500</b> 75 teams





## iMaterialist Challenge (Fashion) at FGVC5

Image classification of fashion products.

Research · 15 days to go ·

**\$2,500**  
149 teams



## iMaterialist Challenge (Furniture) at FGVC5

Image Classification of Furniture & Home Goods.

Research · 15 days to go ·

**\$2,500**  
368 teams

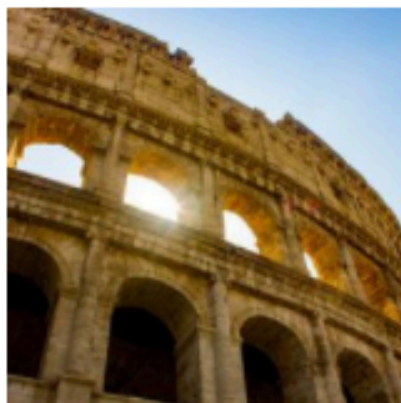


## Google Landmark Retrieval Challenge

Given an image, can you find all of the same landmarks in a dataset?

Research · 7 days to go · 🗃 image data

**\$2,500**  
183 teams



## Google Landmark Recognition Challenge

Label famous (and not-so-famous) landmarks in images

Research · 7 days to go · 🗃 image data

**\$2,500**  
422 teams



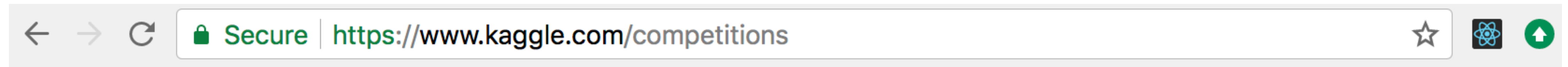
## ImageNet Object Detection Challenge

Identify and label everyday objects in images

Research · 12 years to go · 🗃 image data, object detection

**Knowledge**  
0 teams

# Some scenarios



## Titanic: Machine Learning from Disaster

Start here! Predict survival on the Titanic and get familiar with ML basics

Getting Started · 2 years to go · tutorial, tabular data, binary classification

Knowledge  
11,271 teams

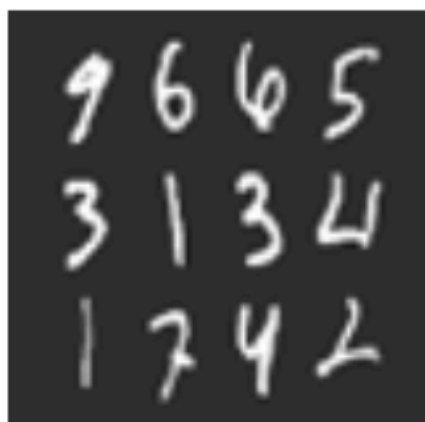


## House Prices: Advanced Regression Techniques

Predict sales prices and practice feature engineering, RFs, and gradient boosting

Getting Started · 2 years to go · tabular data, regression

Knowledge  
5,343 teams



## Digit Recognizer

Learn computer vision fundamentals with the famous MNIST data

Getting Started · 2 years to go · tabular data, image data, multiclass classification...

Knowledge  
2,454 teams



Product Feature Selection

Kudos





## Predict Future Sales

Final project for "How to win a data science competition" Coursera course

[Playground](#) · 8 months to go ·

**Kudos**  
508 teams

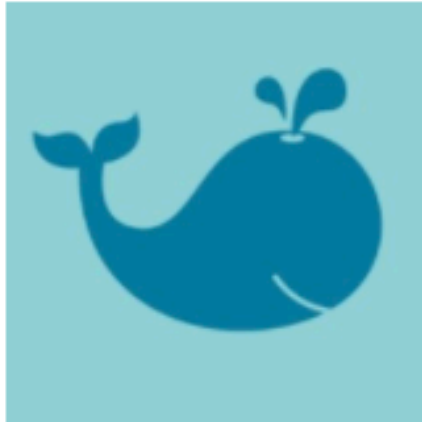


## Freesound General-Purpose Audio Tagging Challenge

Can you automatically recognize sounds from a wide range of real-world environment...

[Research](#) · 3 months to go · 🗝 sound technology

**Knowledge**  
168 teams



## Humpback Whale Identification Challenge

Can you identify a whale by the picture of its fluke?

[Playground](#) · 2 months to go · 🗝 animals, image data

**Kudos**  
277 teams



## iNaturalist Challenge at FGVC5

Long tailed classification challenge spanning 8,000 species.

[Research](#) · 20 days to go ·

**Kudos**  
44 teams



## ImageNet Object Detection from Video Challenge

Identify and label ordinary objects in videos

[Research](#) · 12 years to go · 🗝 image data, object detection

**Knowledge**  
0 teams

# Approaching a Problem

- Available Data

# Approaching a Problem

- Available Data
  - too small dataset? add unsupervised learning

# Approaching a Problem

- Available Data
  - too small dataset? add unsupervised learning
  - too noisy dataset? change loss function / regularize smartly

# Approaching a Problem

- Available Data
  - too small dataset? add unsupervised learning
  - too noisy dataset? change loss function / regularize smartly
- Type of input and output

# Approaching a Problem

- Available Data
  - too small dataset? add unsupervised learning
  - too noisy dataset? change loss function / regularize smartly
- Type of input and output
  - Images or videos? ConvNets



# Approaching a Problem

- Available Data
  - too small dataset? add unsupervised learning
  - too noisy dataset? change loss function / regularize smartly
- Type of input and output
  - Images or videos? ConvNets
  - Text? Bag of words + RNNs or ConvNets

# Approaching a Problem

- Available Data
  - too small dataset? add unsupervised learning
  - too noisy dataset? change loss function / regularize smartly
- Type of input and output
  - Images or videos? ConvNets
  - Text? Bag of words + RNNs or ConvNets
  - Feature-engineered inputs? sparse mlp

# Approaching a Problem

- Available Data
  - too small dataset? add unsupervised learning
  - too noisy dataset? change loss function / regularize smartly
- Type of input and output
  - Images or videos? ConvNets
  - Text? Bag of words + RNNs or ConvNets
  - Feature-engineered inputs? sparse mlp
- Baselines

# Approaching a Problem

- Available Data
  - too small dataset? add unsupervised learning
  - too noisy dataset? change loss function / regularize smartly
- Type of input and output
  - Images or videos? ConvNets
  - Text? Bag of words + RNNs or ConvNets
  - Feature-engineered inputs? sparse mlp
- Baselines
  - Clustering / nearest-neighbors

# Approaching a Problem

- Available Data
  - too small dataset? add unsupervised learning
  - too noisy dataset? change loss function / regularize smartly
- Type of input and output
  - Images or videos? ConvNets
  - Text? Bag of words + RNNs or ConvNets
  - Feature-engineered inputs? sparse mlp
- Baselines
  - Clustering / nearest-neighbors
  - Standard ConvNet / MLP / LSTM

# Challenges at scale

# Challenges at scale

- Engineering Challenges

# Challenges at scale

- Engineering Challenges
  - Data size, type and location



# Challenges at scale

- Engineering Challenges
  - Data size, type and location
    - Typical challenge in deep learning, no one talks about it

# Challenges at scale

- Engineering Challenges
  - Data size, type and location
    - Typical challenge in deep learning, no one talks about it
  - How to keep GPU occupied

# Challenges at scale

- Engineering Challenges
  - Data size, type and location
    - Typical challenge in deep learning, no one talks about it
  - How to keep GPU occupied
    - mini-batching

# Challenges at scale

- Engineering Challenges
  - Data size, type and location
    - Typical challenge in deep learning, no one talks about it
  - How to keep GPU occupied
    - mini-batching
    - efficient kernels

# Challenges at scale

- Engineering Challenges
  - Data size, type and location
    - Typical challenge in deep learning, no one talks about it
  - How to keep GPU occupied
    - mini-batching
    - efficient kernels
    - fusion

# Challenges at scale

- Engineering Challenges
  - Data size, type and location
    - Typical challenge in deep learning, no one talks about it
  - How to keep GPU occupied
    - mini-batching
    - efficient kernels
    - fusion
  - How to keep 10s or 100s of GPUs occupied
    - large-batch tricks

# Accurate, Large Minibatch SGD: Training ImageNet in 1 Hour

Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, Kaiming He

*(Submitted on 8 Jun 2017 (v1), last revised 30 Apr 2018 (this version, v2))*

Deep learning thrives with large neural networks and large datasets. However, larger networks and larger datasets result in longer training times that impede research and development progress. Distributed synchronous SGD offers a potential solution to this problem by dividing SGD minibatches over a pool of parallel workers. Yet to make this scheme efficient, the per-worker workload must be large, which implies nontrivial growth in the SGD minibatch size. In this paper, we empirically show that on the ImageNet dataset large minibatches cause optimization difficulties, but when these are addressed the trained networks exhibit good generalization. Specifically, we show no loss of accuracy when training with large minibatch sizes up to 8192 images. To achieve this result, we adopt a hyper-parameter-free linear scaling rule for adjusting learning rates as a function of minibatch size and develop a new warmup scheme that overcomes optimization challenges early in training. With these simple techniques, our Caffe2-based system trains ResNet-50 with a minibatch size of 8192 on 256 GPUs in one hour, while matching small minibatch accuracy. Using commodity hardware, our implementation achieves ~90% scaling efficiency when moving from 8 to 256 GPUs. Our findings enable training visual recognition models on internet-scale data with high efficiency.



# LARGE BATCH TRAINING OF CONVOLUTIONAL NETWORKS WITH LAYER-WISE ADAPTIVE RATE SCALING

**Anonymous authors**

Paper under double-blind review

## ABSTRACT

A common way to speed up training of deep convolutional networks is to add computational units. Training is then performed using data-parallel synchronous Stochastic Gradient Descent (SGD) with a mini-batch divided between computational units. With an increase in the number of nodes, the batch size grows. However, training with a large batch often results in lower model accuracy. We argue that the current recipe for large batch training (linear learning rate scaling with warm-up) does not work for many networks, e.g. for Alexnet, Googlenet,... We propose a more general training algorithm based on Layer-wise Adaptive Rate Scaling (LARS). The key idea of LARS is to stabilize training by keeping the magnitude of update proportional to the norm of weights for each layer. This is done through gradient rescaling per layer. Using LARS, we successfully trained AlexNet and ResNet-50 to a batch size of 16K.



# DON'T DECAY THE LEARNING RATE, INCREASE THE BATCH SIZE

**Samuel L. Smith\*, Pieter-Jan Kindermans\*, Chris Ying & Quoc V. Le**

Google Brain

{slsmith, pikinder, chrisying, qvl}@google.com

## ABSTRACT

It is common practice to decay the learning rate. Here we show one can usually obtain the same learning curve on both training and test sets by instead increasing the batch size during training. This procedure is successful for stochastic gradient descent (SGD), SGD with momentum, Nesterov momentum, and Adam. It reaches equivalent test accuracies after the same number of training epochs, but with fewer parameter updates, leading to greater parallelism and shorter training times. We can further reduce the number of parameter updates by increasing the learning rate  $\epsilon$  and scaling the batch size  $B \propto \epsilon$ . Finally, one can increase the momentum coefficient  $m$  and scale  $B \propto 1/(1 - m)$ , although this tends to slightly reduce the test accuracy. Crucially, our techniques allow us to repurpose existing training schedules for large batch training with no hyper-parameter tuning. We train ResNet-50 on ImageNet to 76.1% validation accuracy in under 30 minutes.

# Challenges at scale

- Engineering Challenges
  - Data size, type and location
    - Typical challenge in deep learning, no one talks about it
  - How to keep GPU occupied
    - mini-batching
    - efficient kernels
    - fusion
  - How to keep GPUs occupied
    - large-batch tricks
  - Is CPU a bottleneck?

# Production Challenges

- What is different about production?

# Production Challenges

- What is different about production?
- exporting to C++-only runtimes for use in larger projects

# Production Challenges

- What is different about production?
- exporting to C++-only runtimes for use in larger projects
- optimizing mobile systems on iPhone, Android, Qualcomm and other systems

# Production Challenges

- What is different about production?
- exporting to C++-only runtimes for use in larger projects
- optimizing mobile systems on iPhone, Android, Qualcomm and other systems
- using more efficient data layouts and performing kernel fusion to do faster inference (saving 10% of speed or memory at scale is a big win)

# Production Challenges

- What is different about production?
- exporting to C++-only runtimes for use in larger projects
- optimizing mobile systems on iPhone, Android, Qualcomm and other systems
- using more efficient data layouts and performing kernel fusion to do faster inference (saving 10% of speed or memory at scale is a big win)
- quantized inference (such as 8-bit inference)

Show me the code!!!