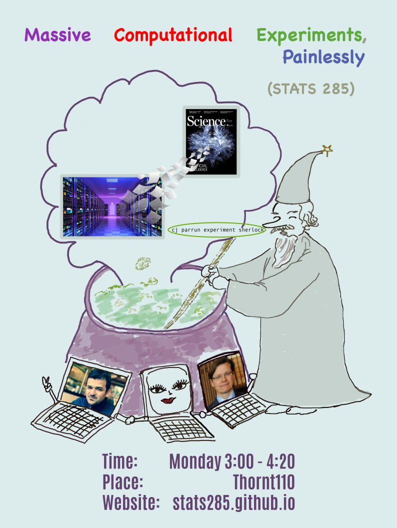
Lecture 09
Push-button Deep Learning on the Cloud

H. Monajemi/DL. Donoho

Stats285, Stanford

Nov/27/2017

Stats 285 Fall 2017



Disclaimer, 1: Google Images

This document contains images obtained by routine Google Images searches. Some of these images may perhaps be copyright. They are included here for educational noncommercial purposes and are considered to be covered by the doctrine of **Fair Use**. *In any event they are easily available from Google Images. It's not feasible to give full scholarly credit to the creators of these images. We hope they can be satisfied with the positive role they are playing in the educational process.*

Outline

Research computing demands are increasing

Shankar-Goodfellow Tweets

Ian Goodfellow replied

Shreya Shankar @sh_reya · 7h
my hope for #DeepLearning dev is that one day we can run things on our laptops within a couple hours. no tmux necessary

6 4 13

Ian Goodfellow @goodfellow_ian · 1h
Maybe better software could make the laptop->cloud connection more seamless. User doesn't invoke tmux but TensorFlow backend uses cloud

3 1 9

Tweet

seamless. User doesn't invoke tmux but TensorFlow backend uses cloud

11/25/17, 2:52 PM

1 Retweet **9** Likes

David Berthelot @D_Berthelot... · 22m
Replying to @goodfellow_ian and @sh_reya
My dream:
with tf.device('cloud123.tpu0'):
  # code

2

Marc J. Schmidt @MarcJSchm... · 41m
Replying to @goodfellow_ian and @sh_reya
exactly on what we are working with the project aetros.com :) will be in the future even more seamlessly with local datasets on remote GPU servers 💪

Research computing demands are increasing
A personal story

Dear X,
I am trying to run GPU experiments on the cluster but these are all **pending** since all the CPUs are occupied. Can you please release 4 CPUs (for the 4 GPUs)? **Otherwise, we are stuck with no GPUs.**

Research computing demands are increasing
A personal story

X

I changed the partition for the pending jobs. **Let's submit your jobs tonight** so that I can bring back those jobs tomorrow early morning.

But I need to **experiment and debug my code ...** how can I submit "my jobs tonight?"

Stats285 theme: Explosion of Computing Resources

Cloud Paradigm:

- Billions of smart devices each drive queries to cloud servers
- Millions of business relying on cloud for all needs

Symbiosis of cloud and economy is *lasting* and *disruptive*.

Cloud provides *any user* **same-day** delivery:

- Tens to hundreds of thousands of hours of CPU
- Pennies per CPU hour
- ≈ 50 cents per GPU hour

Any user can consume *1 Million CPU hours* over a few days for a few $10K's.

Cloud providers offer many services

Google Cloud Platform	Amazon Web Services[7]	Microsoft Azure[8]
Google Compute Engine	Amazon EC2	Azure Virtual Machines
Google App Engine	AWS Elastic Beanstalk	Azure Cloud Services
Google Container Engine	Amazon EC2 Container Service	Azure Container Service
Google Cloud Bigtable	Amazon DynamoDB	Azure Cosmos DB
Google BigQuery	Amazon Redshift	Microsoft Azure SQL Database
Google Cloud Functions	Amazon Lambda	Azure Functions
Google Cloud Datastore	Amazon DynamoDB	Cosmos DB
Google Storage	Amazon S3	Azure Blob Storage

source: Wikipedia

Cloud can accommodate research computing needs

Papyan's case

Why frustrated? create your own GPU cluster on the cloud

45 min later...

Hi X,

I created a cluster following homework2 in stats285. So I have computational resources now.

Cloud can accommodate research computing needs

Shankar's case

Shreya Shankar @sh_reya · 12h
my hope for #DeepLearning dev is that one day we can run things on our laptops within a couple hours. no tmux necessary

7 4 25

Hatef Monajemi @hatefmnj · 4h
@sh_reya this is a solved problem. You should show up to @stats285 lecture on Monday Nov 27 at 3:00 PM in Thornt 110 where we will teach how to do Deep Learning experiments push-button on the cloud. @stats385

1

Shreya Shankar
@sh_reya

Replying to @hatefmnj @stats285 and @stats385

did not know this is a class! 🙌

11/25/17, 5:39 PM

Computing Change is real!

- Computational Science/Computation-enabled discovery is becoming mainstream!
- PhD Students are expected to conduct **1 million CPU-hour** of computation
- Personal Laptops $\rightarrow$ Shared Clusters $\rightarrow$ Personal Clusters

Adapting to Computing Change

- Just as Climate Change demands adaptation,
- Computing Change demands adaptation:
  - **Pose** bold research **hypotheses** to settle computationally
  - **Design massive computing experiments**
  - **Adopt** painless computing **frameworks**
  - **Raise money** to pay for cloud-based computing
  - *Push a button*

How to get rid of the computing interface pain?

We need to rethink the way we do computational experiments

What does an experiment involve?

In our telling, a computational experiment involves:

1. **Precise Specification** (define metric and parameters)
2. **Execution and management** of all the jobs
3. **Harvesting** of all the data generated by all the jobs
4. **Analysis** of the data
5. **Reporting** of results.

Today we add to this list: **Building** of compute clusters

The painless computing paradigm should seamlessly integrate and automate all these tasks

Many open-source frameworks offer automation

painless (push-button) massive computing

- Building Cloud Clusters
  - **ElastiCluster** (Riccardo Murri)
  - StarCluster (MIT)
- Experiment Management Systems (Laptop $\rightarrow$ Cluster)
  - **CJ** (Yours Truly)
  - CodaLab (Percy Liang)
  - PyWren(serverless) (Eric Jonas)
- Machine Learning and Statistics
  - **PyTorch**, Tensorflow, CNTK, Theano, Keras
  - Spark, Dask

Today's focus

Push-button Massive Computational Experiments

The Cloud + ElastiCluster +

aws

python

Outline

Classical Statistics: Linear relationship

Given n realizations (x_i, y_i),

$$y_i = \beta^T x_i + \epsilon_i$$

$$x_i, \beta \in \mathbb{R}^p, \quad y_i \in \mathbb{R}, \quad i = 1, \ldots, n$$

Non-linear relationship

Given n realizations (x_i, y_i),

$$y_i = \Phi(x_i; \Omega) + \epsilon_i$$

- How to choose Φ?
- An example of a classical approach:
  - $\Phi(x; c) = \sum_\ell c_\ell K(x_\ell, x)$ with known *kernel* K
  - Find c_ℓ such that data is reproduced by the model

Neural Nets Approach

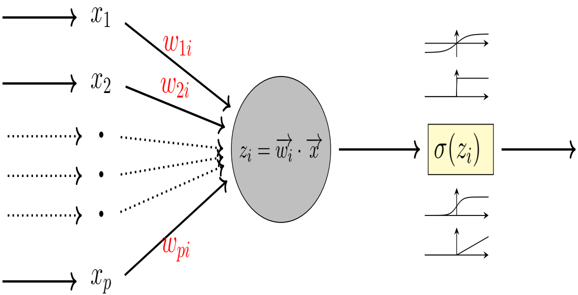
$$y_i = \Phi(x_i; \Omega) + \epsilon_i$$

$$\Phi(x; W, \beta) = \beta^T \sigma(W^T x)$$

- $W \in \mathbb{R}^{p \times d_1}$, $\beta \in \mathbb{R}^{d_1}$
- σ: A nonlinearity typically
  - *non-negative soft-thresholding (ReLU)*
  - *logistic function (sigmoid)*
- Find (W, β) such that data is reproduced (a.k.a **Training**).

Visual illustration

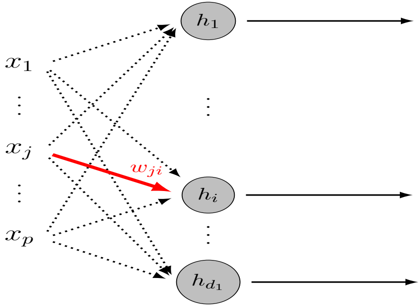
$$i\text{-th row: } \sigma(W^T x)_i \equiv h_i$$



- Perceptron, the basic block (Rosenblatt, 1957)

Visual illustration

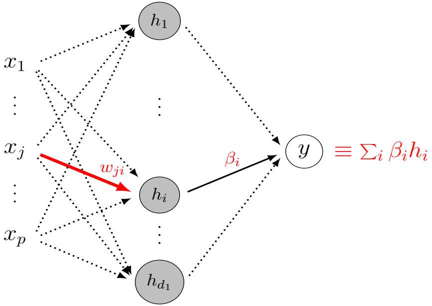
$$h_i = \sigma(W^T x)_i, \quad i = 1, \ldots, d_1$$



- Single-Layer Perceptron

Visual illustration

$$\Phi(x; W, \beta) = \beta^T \sigma(W^T x)$$



- Univariate output

Deep Neural Nets

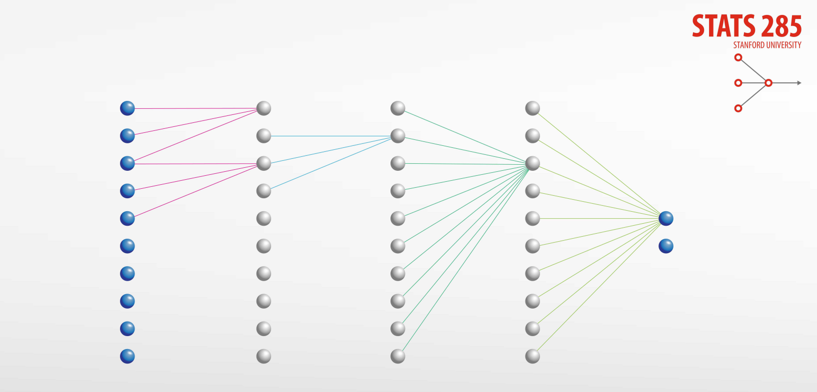
$$y_i = \Phi(x_i; \Omega) + \epsilon_i$$

$$\Phi(x; W_1, \ldots, W_L) = W_L^T \sigma\Big(W_{L-1}^T \ldots \sigma(W_1^T x)\Big)$$

- $W_\ell \in \mathbb{R}^{d_{\ell-1} \times d_\ell}, \quad d_0 = p$
- **Training**: find $(W_1 \ldots, W_L)$.
- highly over-parametrized (# unknowns $\gg n$)

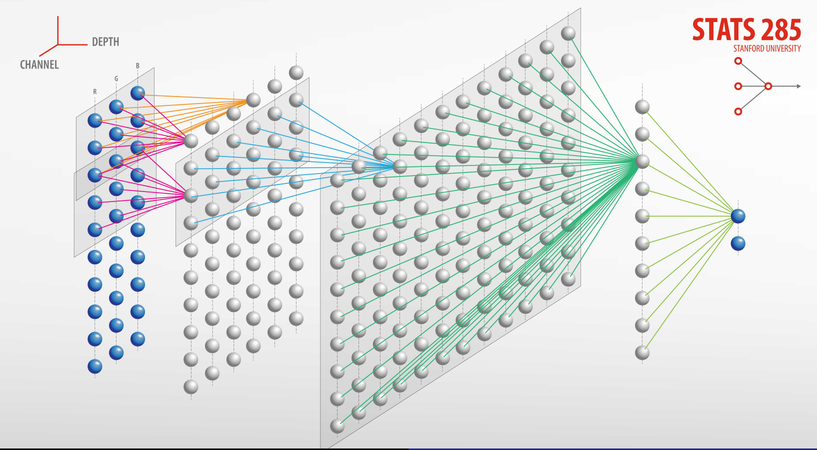
Visual illustration, Deep Nets - 2D

- Locality (convolution)
- Weight Sharing



Visual illustration, Deep Nets - 3D

- Locality (convolution)
- Weight Sharing



Back-propagation – derivation

derivation from LeCun et al. 1988

Given n training examples $(x_i, y_i) \equiv$ (input,target) and L layers

- Constrained optimization

$$\min_{W,x} \quad \sum_{i=1}^{n} \|h_i(L) - y_i\|_2$$

subject to

$$h_i(\ell) = \sigma_\ell \Big[W_\ell h_i\,(\ell - 1) \Big],$$
$$i = 1, \ldots, n, \quad \ell = 1, \ldots, L, \; h_i(0) = x_i$$

- Lagrangian formulation (Unconstrained)

$$\min_{W,x,B} \mathcal{L}(W, x, B)$$

$$\mathcal{L}(W, x, B) = \sum_{i=1}^{n} \left\{ \|h_i(L) - y_i\|_2^2 + \sum_{\ell=1}^{L} B_i(\ell)^T \left(h_i(\ell) - \sigma_\ell \Big[W_\ell h_i\,(\ell - 1) \Big] \right) \right\}$$

Back-propagation – derivation

Forward pass, $\frac{\partial \mathcal{L}}{\partial B}$

$$h_i(\ell) = \sigma_\ell \Big[\underbrace{W_\ell h_i\left(\ell - 1\right)}_{A_i(\ell)} \Big] \quad \ell = 1, \ldots, L, \quad i = 1, \ldots, n$$

Backward (adjoint) pass, $\frac{\partial \mathcal{L}}{\partial h}$, $z_\ell = [\nabla \sigma_\ell] B(\ell)$

$$z(L) = 2\nabla \sigma_L \Big[A_i(L) \Big] \left(y_i - h_i(L) \right)$$
$$z_i(\ell) = \nabla \sigma_\ell \Big[A_i(\ell) \Big] W_{\ell+1}^T z_i(\ell + 1) \quad \ell = 0, \ldots, L - 1$$

Parameter update, $W \leftarrow W + \lambda \frac{\partial \mathcal{L}}{\partial W}$

$W_\ell \leftarrow W_\ell + \lambda \sum_{i=1}^n z_i(\ell) h_i^T(\ell - 1)$

Back-prop in PyTorch

- Forward pass

```
output      = model(input)
loss        = loss_fn(outputs, target)
```

- Parameter Update

```
loss.backward()      # computes  ∑_i z_i h_i^T
optimizer.step()     # adds to  W^k
```

$\sum_i z_i h_i^T$, W^k

Deep Learning Experiments

1. ETL

```
trainloader, testloader = dl.torch.data.etl(dataset=data_set, ...)
```

2. Define model (network)

```
model = dl.torch.models.mini_alexnet()
```

3. Define loss and optimizer

```
loss_fn      = torch.nn.CrossEntropyLoss()
optimizer    = torch.optim.SGD(model.parameters(), ...)
```

4. Train

```
dl.torch.train(model, loss_fn, optimizer, trainloader, testloader, ...)
```

Understanding deep learning requires rethinking generalization (Zhang et al.)

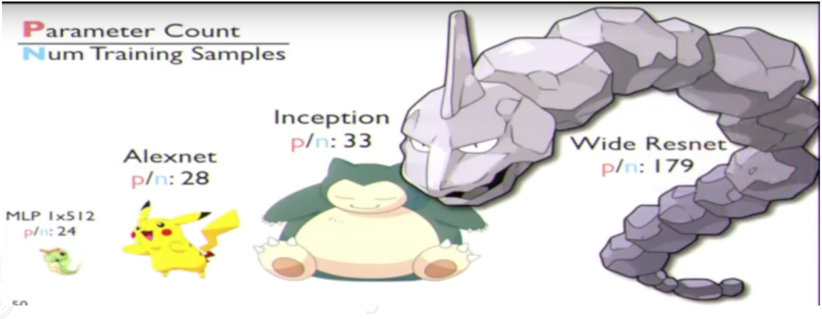
Computer Science > Learning

Understanding deep learning requires rethinking generalization

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, Oriol Vinyals

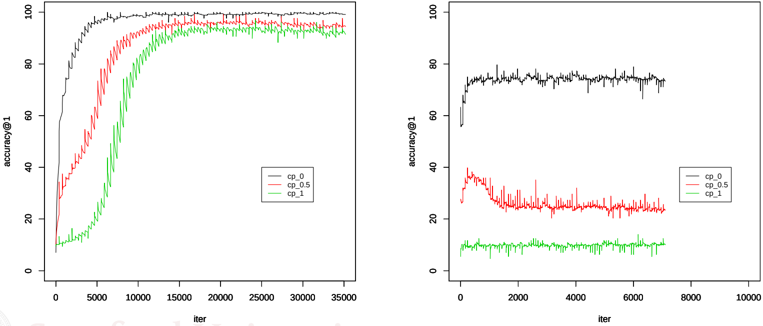
(Submitted on 10 Nov 2016 (v1), last revised 26 Feb 2017 (this version, v2))

Despite their massive size, successful deep artificial neural networks can exhibit a remarkably small difference between training and test performance. Conventional wisdom attributes small generalization error either to properties of the model family, or to the regularization techniques used during training.
Through extensive systematic experiments, we show how these traditional approaches fail to explain why large neural networks generalize well in practice. Specifically, our experiments establish that state-of-the-art convolutional networks for image



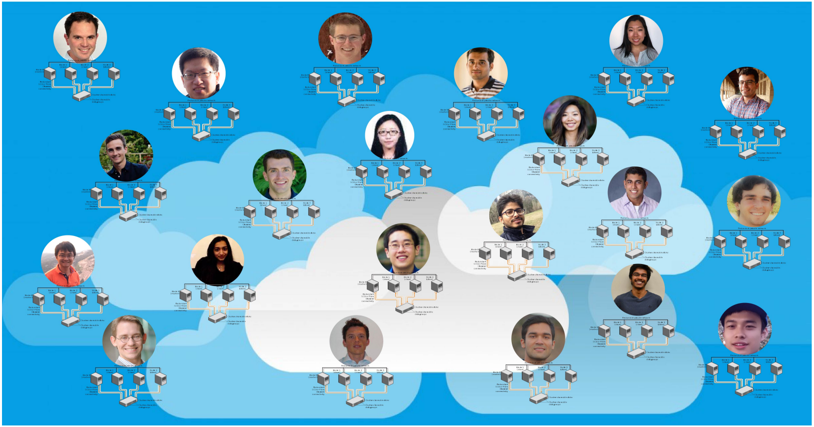
Example of Replication

90 epochs in less than 5 min on Google Compute Engine



Outline

Stats285 on the cloud!



Push-button Massive Computational Experiments

+ ElastiCluster +

Literally Push-button!

Action items for conducting experiments:

```
1) elasticluster start gce
```

```
2) cj parrun train.py gce -alloc "--gres=gpu:1"
```

Build your cluster
`elasticluster start gce`

- Setup Google Cloud Billing (*Yes, they will charge you!*)
  1. `gmail username`
  2. `project_id`
  3. `client_id`
  4. `client_secret`
- Install Docker on your machine
  - removes the pain and ir-reproducibility of software install
- Create your cluster using dockerized ElastiCluster

```
docker image pull stats285/elasticluster-gpu

docker run -it stats285/elasticluster-gpu

elasticluster start gce
```

Fire and forget!
`cj parrun train.py gce`

- add your cluster info to `CJ_install/ssh_config`

```
[gce]
 Host             35.199.171.137
 User             hatefmonajemi
 Bqs              SLURM
 Repo             /home/hatefmonajemi/CJRepo_Remote
 Python           python3.4
 Pythonlib        pytorch:cuda80:-c soumith
[gce]
```

- Fire up your jobs and track them using **ClusterJob**

```
cj parrun train.py gce -alloc "--gres=gpu:1"
```

Your assignment is online

Massive Computational Experiments, Painlessly (STATS 285)

Stanford University, Fall 2017

Assignment 02

In this assignment, we will conduct a collaborative project testing certain theoretical hypotheses in Deep Learning. In particular, each of you will build **your own personal SLURM cluster** on Google Compute Engine (GCE) using elasticluster and then run massive computational experiments using clusterjob. We then collect and analyse all the results you will generate and document our observations. Please follow the following step to setup your cluster and run experiments. This documents only contains the detail of setting up your cluster and testing that it works properly with

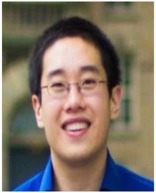
Summary

- Small computations $\rightarrow$ Massive computations
- Interactive model (copy-paste) $\rightarrow$ experiment management systems (EMS)
- Personal Laptops $\rightarrow$ Shared Clusters $\rightarrow$ Personal Clusters
- Expansion by factors of 1000's in immediate computing capacity
- Rise of frameworks takes away pain of massive computing
- Deep Learning is now a technology (Thanks to frameworks)
- Lots of opportunities for semi-empirical/semi-theoretical Deep Learning studies

acknowledgements

- People

J. Lozano



X.Y. Han

R. Murri

V. Papyan

- Google Cloud Platform Education Grants
- ElastiCluster Team