Analyzing Electroencephalograms Using Cloud Computing Techniques

Kathleen Ericson
Shrideep Pallickara
Charles W. Anderson

Colorado State University

December 1, 2010

Outline

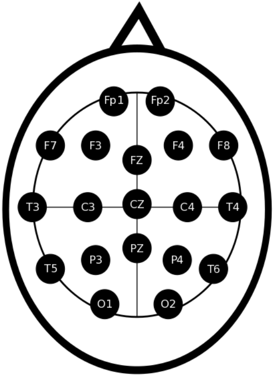
Outline

Brain Computer Interfaces (BCIs)

- Allows users who have lost voluntary motor control to interact with a computer
- BCIs work by analyzing electroencephelograms (EEGs) to interpret the users intent
- EEG signals are gathered in a non-invasive method
- Typing interface (Doug Hains, Elliott Forney)
- Weelchair (Millan)

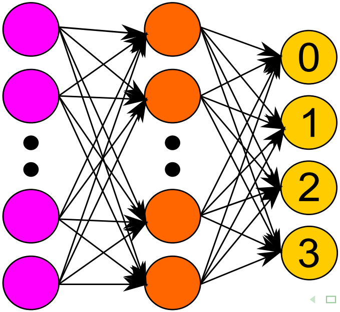
Gathering EEG data

- Non–invasive methods
  - User wears a cap which holds electrodes to the scalp
- Electrode placement followed the international 10-20 system of electrode placement



Artificial Neural Networks

- Number of input and output nodes are defined by the data
- Number of hidden units can vary
  - More hidden units can model more complex data
  - More hidden units take longer to train
- Weights are added between input and hidden and hidden and output layers



Outline

Benefits of the cloud

- Current BCI applications are limited
  - All computation happens with the user
  - Mobile BCI applications (such as a wheelchair) are tied to a laptop

Benefits of the cloud

- Current BCI applications are limited
  - All computation happens with the user
  - Mobile BCI applications (such as a wheelchair) are tied to a laptop
- A single user is classified by a single machine
  - A dedicated machine for a single user is under–utilized

Benefits of the cloud

- Current BCI applications are limited
    - All computation happens with the user
    - Mobile BCI applications (such as a wheelchair) are tied to a laptop
- A single user is classified by a single machine
    - A dedicated machine for a single user is under–utilized
- Computing capabilities are limited
    - NN complexity is limited by what can be trained on a laptop

Benefits of the cloud

- Multiple users can access the same cloud
  - Aggregation of data
    - More data leads to better trained neural networks
  - Cloud servers are separate from the users
    - Users not limited to the computational power of laptops
  - Possibility for massive scaling
    - Thousands of users can be supported simultaneously
- Complex pipelines for classification can be developed
  - Computations can be chained through MapReduce or graph-based paradigms

Outline

Approach

- R backend
  - Optimized for matrix multiplication
  - Existing code available for EEG manipulation, as well as neural network code

Approach

- R backend
  - Optimized for matrix multiplication
  - Existing code available for EEG manipulation, as well as neural network code
- Group of experts approach
  - Fits the map reduce framework – mappers classify, reducer produces expert opinion

Approach

- R backend
  - Optimized for matrix multiplication
  - Existing code available for EEG manipulation, as well as neural network code
- Group of experts approach
  - Fits the map reduce framework – mappers classify, reducer produces expert opinion
- 3 sets of experiments:
  - Baseline times in R
  - Cloud communication overhead with Snowfall
  - Cloud and bridge communication overhead with Granules and JRI

Outline

1. Background
   - BCI
   - Gathering EEG
   - Artificial Neural Networks
2. Benefits of the cloud
3. Approach
4. Frameworks
5. Network Setup
6. Results
   - Basic Tests
   - Granules Stress Tests
7. Conclusions and Future Work

Frameworks Used

- Snowfall
  - Parallel computing package for R
  - Builds on the Snow package
  - Executes sequential code on multiple machines simultaneously
  - Does not require strong parallel computing background

Frameworks Used

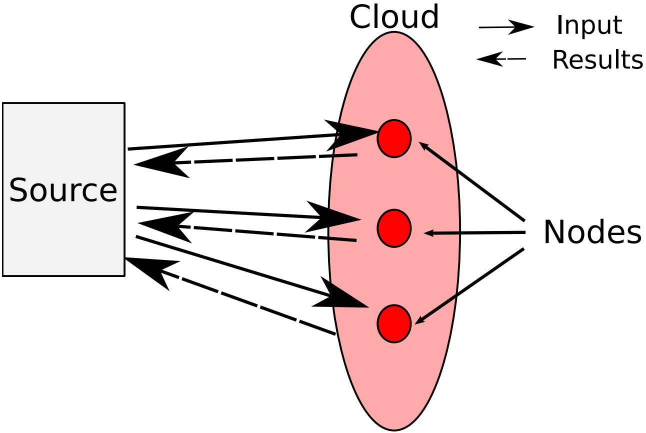
- Snowfall
  - Parallel computing package for R
  - Builds on the Snow package
  - Executes sequential code on multiple machines simultaneously
  - Does not require strong parallel computing background
- Granules
  - Lightweight cloud computing runtime
  - Java based
  - Allows user to specify run semantics – can enter a dormant state while waiting for more data to become available

Frameworks Used

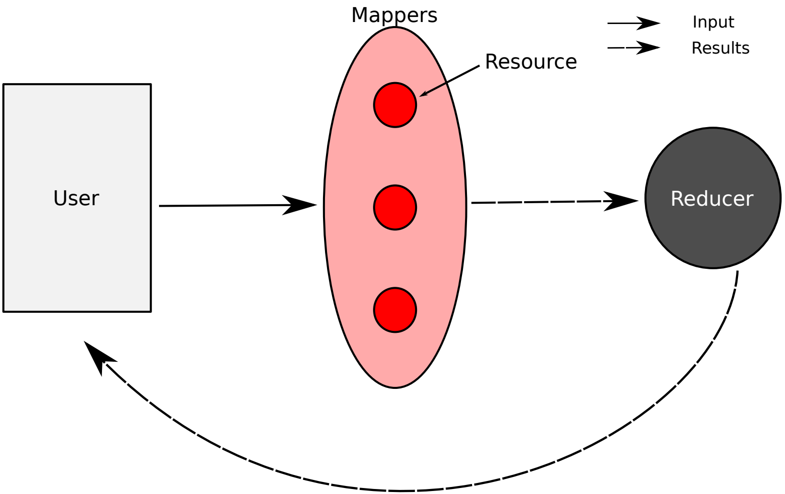
- Snowfall
  - Parallel computing package for R
  - Builds on the Snow package
  - Executes sequential code on multiple machines simultaneously
  - Does not require strong parallel computing background
- Granules
  - Lightweight cloud computing runtime
  - Java based
  - Allows user to specify run semantics – can enter a dormant state while waiting for more data to become available
- JRI
  - Java R Interface
  - Allows R computations to be run through Java
  - Communication is string–based

Outline

Snowfall Network Setup



Granules Network Setup



Outline

Baseline Results

Table: Loading a single training set (200MB) in ms

Mean(ms)	Min(ms)	Max(ms)	SD(ms)
6581.602	6439.742	6822.34	101.3716

Table: Training a neural network from 1 training set in ms

Mean(ms)	Min(ms)	Max(ms)	SD(ms)
194463.7	192433.3	197094.9	1300.87

Baseline Results

Table: Classification times with 1 neural net in ms

Stream Time	Mean(ms)	Min(ms)	Max(ms)	SD(ms)
5s	23.0432	22.17889	23.56791	0.4734237
1s	5.28194	4.909039	11.16085	0.8568976
250ms	1.710529	1.673937	1.926184	0.03777157

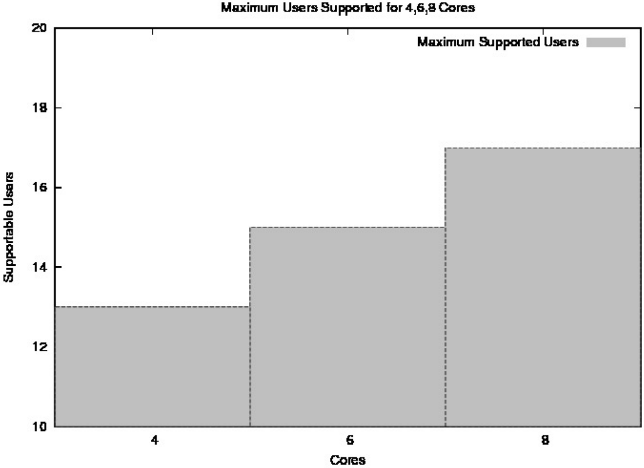
Snowfall and Granules Training Comparisons

NNs	Training Sets	Approach	Mean(ms)	Min(ms)	Max(ms)	SD(ms)
1	1	Snowfall	409860.5	403364.3	419965.7	4216.875
		Granules	313304.2	306402.0	329149.0	5141.031
3	1	Snowfall	462626.2	401475.9	483235.9	23512.64
		Granules	675968.8	610550.0	772679.0	52823.21
1	4	Snowfall	1001631	971224	1020680	17743.27
		Granules	1933540	1782531	2057664	110686.70
3	4	Snowfall	988410.4	964499	1023549	17513.37
		Granules	1964255.0	1779853	2131574	136452.10

Classification Times

Method	Stream Time	Mean(ms)	Min(ms)	Max(ms)	SD(ms)
Snowfall	5s	8884.60	8797.745	9069.47	85.82
Granules		141.69	136.42	266.63	12.75
Snowfall	1s	5825.71	5815.38	5857.98	10.09
Granules		93.16	47.51	492.68	32.13
Snowfall	250ms	2831.32	2830.38	2849.83	2.03
Granules		87.25	48.57	92.67	4.49

Maximum Supported Users on a Single Machine



Scaling to multiple machines

- Gathered statistics for classification on 5 and 10 machines
- Each machine supported 15 users
- While 17 users per 8-core machine could be supported, the network was swamped with 150 simultaneous users
  - 12MB/s

Scaling to multiple machines

- Gathered statistics for classification on 5 and 10 machines
- Each machine supported 15 users
- While 17 users per 8-core machine could be supported, the network was swamped with 150 simultaneous users
  - 12MB/s 1GB/83s

Scaling to multiple machines

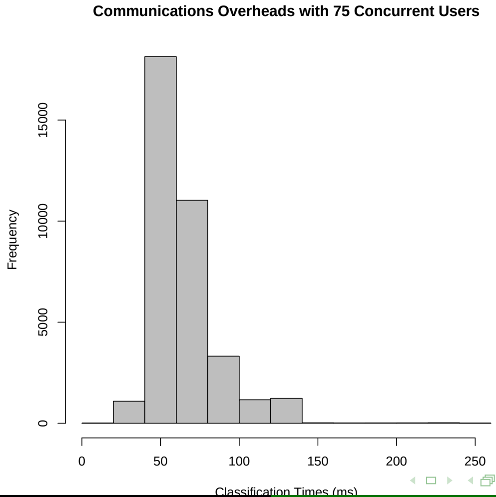
- Gathered statistics for classification on 5 and 10 machines
- Each machine supported 15 users
- While 17 users per 8-core machine could be supported, the network was swamped with 150 simultaneous users
  - 12MB/s   1GB/83s   1TB/23h

Scaling to multiple machines

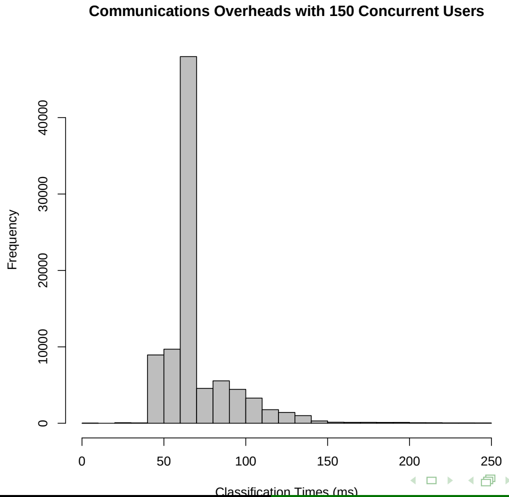
- Gathered statistics for classification on 5 and 10 machines
- Each machine supported 15 users
- While 17 users per 8-core machine could be supported, the network was swamped with 150 simultaneous users
  - 12MB/s 1GB/83s 1TB/23h

	Mean(ms)	Min(ms)	Max(ms)	SD(ms)
75 Users	64.33	21.69	268.30	20.51
150 Users	69.81	22.01	352.82	22.49

Stress Histograms – 75 users



Stress Histograms – 150 users



Outline

Conclusions

- Granules is a viable option for real-time EEG classification in the cloud
- While a pure R implementation can train a network more quickly, there is no native R support for continuous streaming data
- JRI carries a heavy overhead for communications
- Compression is needed to scale further
  - With 150 users, we are processing 1GB of EEG signals every 83 seconds
  - At this rate, over 1TB of data is processed in a day

Future Work

- Develop a byte-based Granules Bridge for R
- Implement an online learning algorithm
- Implement compression

Questions

Questions?

ericson@cs.colostate.edu