Sense Making in an IOT World: Sensor Data Analysis with Deep Learning

Natalia Vassilieva, PhD
Senior Research Manager
Deep learning proof points as of today

**Vision**
- Search & information extraction
- Security/Video surveillance
- Self-driving cars
- Robotics

**Speech**
- Interactive voice response (IVR) systems
- Voice interfaces (Mobile, Cars, Gaming, Home)
- Security (speaker identification)
- Health care
- People with disabilities

**Text**
- Search and ranking
- Sentiment analysis
- Machine translation
- Question answering

**Other**
- Recommendation engines
- Advertising
- Fraud detection
- AI challenges
- Drug discovery
- Sensor data analysis
- Diagnostic support
Why Deep Learning & Sensor Data?

Deep Learning is about …

– Huge volumes of training data (labeled and unlabeled)
– Multidimensional and complex data with non-trivial patterns (spatial or temporal)
– Replacement of manual feature engineering with unsupervised feature learning
– Cross modality feature learning

Sensor Data is about …

– Huge volumes of data (mostly unlabeled)
– Complex data with non-trivial patterns (mostly temporal)
– Variety of data representations, feature engineering is hard
– Multiple modalities

Works well for speech!

Most sensor data is time series
This talk

Does Deep Learning work for sensor data?

Do existing infrastructure and algorithms fit sensor data?

The Machine and Distributed Mesh Computing
Part I

Case Study: Sensor Data Analysis with Deep Learning
Patient activity recognition from accelerometer data

- Scripted video and accelerometer data from one sensor and 52 subjects (~20 min per subject)
- Accelerometer data: 500Hz x 4 dimensions = 12000 measurements per minute per person
- 16 classes
Data distribution

Total number of frames: ~3.35M
Approaches

Baselines
- ZeroR (majority class predictor)
- Support Vector Machines
- Decision Trees (C50 implementation)
- Shallow Neural Networks (FANN library)
- Features: manually engineered

Deep Neural Networks
- Fully-connected hidden layers (pre-trained with stacked sparse autoencoders) + softmax
- Time-delay layers
- Recurrent layers
- Deep Neural Networks and Conditional Random Fields
- Multiple meta-parameter configurations
- Features: amplitude spectrum
Approaches

Baselines
- ZeroR (majority class predictor)
- Support Vector Machines
- Decision Trees (C50 implementation)
- Shallow Neural Networks (FANN library)
- Features: manually engineered

Deep Neural Networks
- Fully-connected: stacked sparse autoencoders + softmax
- Time-delay layers
- Recurrent layers
- Deep Neural Networks and Conditional Random Fields
- Multiple meta-parameter configurations
- Features: amplitude spectrum
Growing class-separation power of deeper representations

Raw data
First level of representations
Second level of representations
Results: single person

Baseline methods, engineered features

<table>
<thead>
<tr>
<th></th>
<th>ZeroR</th>
<th>SVM (binary)</th>
<th>Shallow NN</th>
<th>c50</th>
<th>SVM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy</td>
<td>71.2</td>
<td>97.6</td>
<td>98.6</td>
<td>99.6</td>
<td>98.03</td>
</tr>
</tbody>
</table>

Deep Neural Networks, amplitude spectrum

<table>
<thead>
<tr>
<th></th>
<th>1533-200-200-16</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy</td>
<td>99.7</td>
</tr>
</tbody>
</table>
Results: 52 subjects, subject-independent

Baselines v.s. DNN

<table>
<thead>
<tr>
<th></th>
<th>ZeroR</th>
<th>c50</th>
<th>DNN</th>
<th>DNN + CRF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy</td>
<td>69.7</td>
<td>71.6</td>
<td>84.5</td>
<td>95.1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>set type</th>
<th>size</th>
</tr>
</thead>
<tbody>
<tr>
<td>train</td>
<td>2608637</td>
</tr>
<tr>
<td>cv</td>
<td>0</td>
</tr>
<tr>
<td>test</td>
<td>738180</td>
</tr>
<tr>
<td>total</td>
<td>3346817</td>
</tr>
</tbody>
</table>
Results: 52 subjects, subject-independent

Baselines v.s. DNN

<table>
<thead>
<tr>
<th></th>
<th>ZeroR</th>
<th>c50</th>
<th>DNN</th>
<th>DNN + CRF</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Accuracy</strong></td>
<td>69.7</td>
<td>71.6</td>
<td>84.5</td>
<td>95.1</td>
</tr>
</tbody>
</table>

Deep models:
- are better at classification on sensor data (generalize better)
- do not require sophisticated feature engineering
- require significant amount of iterations to converge
Part II
Today’s infrastructure and Deep Learning
## Today’s scale
Model size, data size, compute requirements

<table>
<thead>
<tr>
<th>Application</th>
<th>Model</th>
<th>Training data</th>
<th>FLOP per epoch</th>
<th>Training time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vision</td>
<td>1.7 * 10^9</td>
<td>14*10^6 images</td>
<td>6<em>1.7</em>10^9<em>14</em>10^6</td>
<td>3 days x 16000 cores</td>
</tr>
<tr>
<td></td>
<td>~6.8 GB</td>
<td>~2.5 TB (256x256)</td>
<td>~1.4*10^17</td>
<td>2 days x 16 servers x 4 GPUs</td>
</tr>
<tr>
<td></td>
<td></td>
<td>~10 TB (512x512)</td>
<td></td>
<td>8 hours x 36 servers x 4 GPUs</td>
</tr>
<tr>
<td>Speech</td>
<td>60 * 10^6</td>
<td>100K hours of audio</td>
<td>6<em>60</em>10^6<em>34</em>10^9</td>
<td>days x 8 GPUs</td>
</tr>
<tr>
<td></td>
<td>~240 MB</td>
<td>~34*10^9 frames</td>
<td>~1.2*10^19</td>
<td></td>
</tr>
<tr>
<td>Text</td>
<td>6.5 * 10^6</td>
<td>856*10^6 words</td>
<td>6<em>6.5</em>10^6<em>856</em>10^6</td>
<td>4 weeks</td>
</tr>
<tr>
<td></td>
<td>~260 MB</td>
<td></td>
<td>~3.3*10^16</td>
<td></td>
</tr>
<tr>
<td>Signals</td>
<td>1.2 * 10^6</td>
<td>3*10^6 frames</td>
<td>6<em>1.2</em>3<em>10^6</em>3*10^6</td>
<td>days</td>
</tr>
<tr>
<td></td>
<td>~4.8 MB</td>
<td></td>
<td>6.5*10^13</td>
<td></td>
</tr>
</tbody>
</table>
Challenges of DNN training
Slow and expensive

- Very large number of parameters (>10^6), huge (unlabeled) data sets for training (10^6 - 10^9)
- Computationally expensive: requires O(model size * data size) FLOPs per epoch
- Needs many iterations (and epochs) to converge
- Needs frequent synchronization to converge fast

Compute requirements today:
10^{13} – 10^{19} FLOPs per epoch
1 epoch per hour: ~10x TFLOPS SP

Today’s hardware:
NVIDIA Titan X: 7 TFLOPS SP, 12 GB memory
NVIDIA Tesla M40: 7 TFLOPS SP, 12 GB memory
NVIDIA Tesla K40: 4.29 TFLOPS SP, 12 GB memory
NVIDIA Tesla K80: 5.6 TFLOPS SP, 24 GB memory
INTEL Xeon Phi: 2.4 TFLOPS SP
Scalability of DNN training for time series
Hard to scale

- Google Brain: 1000 machines (16000 CPUs) x 3 days
- COTS HPC systems: 16 machines x 4 GPUs x 2 days
- Deep Image by Baidu: 36 machines x 4 GPUs x ~8 hours
- Deep Speech by Baidu: 8 GPUs x ~weeks
- Deep Speech 2 by Baidu: 8 or 16 GPUs x 3 to 5 days

Limited scalability of training for speech/time-series data!

J. Dean et. al, Large Scale Distributed Deep Networks
Types of artificial neural networks
Topology to fit data characteristics

Images:
Locally Connected Convolutional

Speech, time series, sequences:
Fully Connected, Recurrent
Today’s hardware (scale-out or scale-up)

CPU/GPU cluster

- GPU
- PCIe
- CPU
- Memory

InfiniBand

Multi-socket large memory machine

- CPU
- Memory
- NUMA node 1
- NUMA node 2
- QPI link
- CPU
- Memory
- NUMA node 3
- NUMA node 4
- QPI link

InfiniBand: ~12 GB/s
PCIe: ~16 GB/s
QPI link: ~12.8 GB/s per direction
Part III
The Machine and Distributed Mesh Computing
Processor-centric computing  Memory-Driven Computing

Memory + Fabric

Hewlett Packard Enterprise
I/O

 Dram

 Copper
Processor-centric computing  Memory-Driven Computing
The Machine will be ported to different scales and form-factors

NVM = Non-volatile memory
## The evolution of the IoT

<table>
<thead>
<tr>
<th>Gen 0</th>
<th>Yesteryears</th>
<th>Gen 1</th>
<th>Today</th>
<th>Gen 2</th>
<th>Tomorrow</th>
<th>Gen 3</th>
<th>The future</th>
</tr>
</thead>
<tbody>
<tr>
<td>Things on a network</td>
<td>Still works well for small, local, custom systems with strict performance needs</td>
<td>The Cloud-centric IoT</td>
<td>Good choice for low-cost “things” where data can easily be moved, with few ramifications</td>
<td>Edge analytics</td>
<td>Ideal for “things” producing large volumes of data that are difficult, costly or sensitive to move</td>
<td>Distributed Mesh Computing</td>
<td>Multi-party “things” autonomously collaborate with privacy intact</td>
</tr>
</tbody>
</table>
Tomorrow: Deep Learning and Edge Analytics

**Edge Node**
- Gets trained model
- Uses the model in real-time
- Collects data
- Sends *some* data to center

**Center**
- Collects all data
- Trains model
- Sends model to edge nodes
The Future: Deep Learning and Distributed Mesh Computing

Edge Node
Participate in Training
Uses the model in real-time
Collects data
Sends some data in mesh

The Mesh
Distributed training
Sends model as needed
Summary

Does Deep Learning work for sensor data?
Yes, we have proof points

Do existing infrastructure and algorithms fit sensor data?
No, training deep models for sensor data is slow and expensive today

The Machine and Distributed Mesh Computing
We believe this changes everything
Thank you!

Natalia Vassilieva
nvassilieva@hpe.com

To learn more about Hewlett Packard Labs, visit: http://www.labs.hpe.com
To learn more, visit www.hpe.com/themachine