A Deep Learning Approach to Sign Language Recognition using Stacked Sparse Autoencoders

Dr. Pablo Rivas*
Ezequiel R.\textsuperscript{^}{\textdagger} Omar V.\textsuperscript{^}{\textdagger} Samuel G\textsuperscript{^}{\textdagger} Deep D.\textsuperscript{*}

*Marist College, School of Computer Science and Mathematics
\textsuperscript{^}{\textdagger}Nogales Institute of Technology, Research and Grad. Studies
The Problem

- American Sign Language (ASL)
The Data

- *Learning* the American Sign Language (ASL)
The Data

- *Learning* the American Sign Language (ASL)
The Data

- *Learning* the American Sign Language (ASL)
The Data

PCA visualization colored by Sign

PCA-tSNE visualization colored by Sign
## Existing Approaches

<table>
<thead>
<tr>
<th>Method</th>
<th>Class type</th>
<th># of class</th>
<th># of subj.</th>
<th>Test w/ diff.</th>
<th>Input</th>
<th>Accur. (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nagi et al. [8]</td>
<td>Gesture</td>
<td>6</td>
<td>-</td>
<td>No</td>
<td>Color</td>
<td>96</td>
</tr>
<tr>
<td>Van den Bergh et al. [14]</td>
<td>Gesture</td>
<td>6</td>
<td>-</td>
<td>No</td>
<td>Color &amp; Depth</td>
<td>99.54</td>
</tr>
<tr>
<td>Isaacs et al. [3]</td>
<td>Alphabets</td>
<td>24</td>
<td>-</td>
<td>-</td>
<td>Color</td>
<td>99.9</td>
</tr>
<tr>
<td>Pugeault et al. [10]</td>
<td>Alphabets</td>
<td>24</td>
<td>5</td>
<td>-</td>
<td>Depth</td>
<td>73</td>
</tr>
<tr>
<td>Pugeault et al. [10]</td>
<td>Alphabets</td>
<td>24</td>
<td>5</td>
<td>-</td>
<td>Color &amp; Depth</td>
<td>69</td>
</tr>
<tr>
<td>Pugeault et al. [10]</td>
<td>Alphabets</td>
<td>24</td>
<td>5</td>
<td>-</td>
<td>Color &amp; Depth</td>
<td>75</td>
</tr>
<tr>
<td>Kuznetsova et al. [6] (50/50)%</td>
<td>Alphabets</td>
<td>24</td>
<td>5</td>
<td>No</td>
<td>Depth</td>
<td>87</td>
</tr>
<tr>
<td>Kuznetsova et al. [6] (4/1)</td>
<td>Alphabets</td>
<td>24</td>
<td>5</td>
<td>Yes</td>
<td>Depth</td>
<td>57</td>
</tr>
<tr>
<td>Dong et al. [2] (50/50)%</td>
<td>Alphabets</td>
<td>24</td>
<td>5</td>
<td>No</td>
<td>Depth</td>
<td>90</td>
</tr>
<tr>
<td>Dong et al. [2] (4/1)</td>
<td>Alphabets</td>
<td>24</td>
<td>5</td>
<td>Yes</td>
<td>Depth</td>
<td>70</td>
</tr>
<tr>
<td>Ours (re-training) (50/25/25)%</td>
<td>Alph. &amp; Digit</td>
<td>31</td>
<td>5</td>
<td>No</td>
<td>Depth</td>
<td>99.99</td>
</tr>
<tr>
<td>Ours (re-training) (3/1/1)</td>
<td>Alph. &amp; Digit</td>
<td>31</td>
<td>5</td>
<td>Yes</td>
<td>Depth</td>
<td>75.18</td>
</tr>
<tr>
<td>Ours (re-training) (4/1)</td>
<td>Alph. &amp; Digit</td>
<td>31</td>
<td>5</td>
<td>Yes</td>
<td>Depth</td>
<td>78.39</td>
</tr>
<tr>
<td>Ours (fine-tuning) (3/1/1)</td>
<td>Alph. &amp; Digit</td>
<td>31</td>
<td>5</td>
<td>Yes</td>
<td>Depth</td>
<td>83.58</td>
</tr>
<tr>
<td>Ours (fine-tuning) (4/1)</td>
<td>Alph. &amp; Digit</td>
<td>31</td>
<td>5</td>
<td>Yes</td>
<td>Depth</td>
<td>85.49</td>
</tr>
</tbody>
</table>

Existing Solutions

- Learning the American Sign Language (ASL) with CNNs
The Proposal

• Learning the American Sign Language (ASL) with Auto-encoders
  – Simpler than CNN
  – More efficient than CNN (deployed)
  – Faster to train than CNN (for a similar number of layers)
  – Similar performance to a CNN
  – CNNs are not the panacea in pattern recognition on images or computer vision
The Proposal

- *Let's talk about Auto-encoders*

65536 neural units

…

decoding layer

100 neural units

…

encoding layer

training images
The Proposal

- *Let's talk about Auto-encoders*

\[
L = \frac{1}{N} \| \mathbf{x}_n - \hat{\mathbf{x}}_n \|_2^2 + \theta_w \frac{1}{2} \sum_{l=1}^{L} \| \mathbf{w}^l \|_2^2 + \theta_s \sum_{m=1}^{M} KL(\theta_\alpha \| \tilde{\alpha}_m)
\]
The Proposal

- *Let's talk about Auto-encoders*

\[
L = \frac{1}{N} \| \mathbf{x}_n - \hat{\mathbf{x}}_n \|^2_2 + \theta_w \frac{1}{2} \sum_{l=1}^{L} \| \mathbf{w}^l \|^2_2 + \theta_s \sum_{m=1}^{M} KL (\theta_\alpha \| \bar{\alpha}_m )
\]

\[
\sum_{m=1}^{M} KL (\theta_\alpha \| \bar{\alpha}_m ) = \sum_{m=1}^{M} \theta_\alpha \log \left( \frac{\theta_\alpha}{\bar{\alpha}_m} \right) + (1 - \theta_\alpha) \log \left( \frac{1 - \theta_\alpha}{1 - \bar{\alpha}_m} \right)
\]

\[
\bar{\alpha}_m = \frac{1}{N} \sum_{n=1}^{N} \psi \left( \mathbf{w}^{(l)}_m^T \mathbf{x}_n + b^{(l)}_m \right)
\]
output \( d_n \in \mathbb{R}^{31} \)

softmax activations

\( \tilde{x}_n \in \mathbb{R}^{50}_+ \)

logistic activations

\( \hat{x}_n \in \mathbb{R}^{100}_+ \)

sparse encoding layer

logistic activations

\( x_n \in \mathbb{N}^{65536}_+ \)

sparse encoding layer

input image

\[
E = \frac{1}{N} \sum_{n=1}^{N} \sum_{c \in C} \hat{d}_{cn} \ln d_{cn} + (1 - \hat{d}_{cn}) \ln(1 - d_{cn})
\]

neural network layer
## Results

- of the Auto-encoders

<table>
<thead>
<tr>
<th></th>
<th>S1</th>
<th>S2</th>
<th>S3</th>
<th>S4</th>
<th>S5</th>
<th>Avg.</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACC</td>
<td>0.9748</td>
<td>0.9923</td>
<td>0.9935</td>
<td>0.9929</td>
<td>0.9910</td>
<td>0.9889</td>
</tr>
<tr>
<td>SPC</td>
<td>0.9991</td>
<td>0.9997</td>
<td>0.9998</td>
<td>0.9998</td>
<td>0.9997</td>
<td>0.9996</td>
</tr>
<tr>
<td>MAE</td>
<td>0.1483</td>
<td>0.0640</td>
<td>0.0373</td>
<td>0.0347</td>
<td>0.0494</td>
<td>0.0667</td>
</tr>
</tbody>
</table>
Results
Results
Results
Chameleon Setup

- **Resource type:** bare metal/CHI@TACC
- **Lease:** GPU P100
- **Image:** CC-Ubuntu16.04-CUDA8
- **Libraries:**
  - cuDNN
  - libatlas-dev
Libraries: cuDNN

Connection closed by 129.114.109.140
Pablos-MacBook-Air:~ ssh rivas_perea$ ssh cc@129.114.109.140
Welcome to Ubuntu 16.04.2 LTS (GNU/Linux 4.4.0-72-generic x86_64)

* Documentation: https://help.ubuntu.com
* Management: https://landscape.canonical.com
* Support: https://ubuntu.com/advantage

Get cloud support with Ubuntu Advantage Cloud Guest:
http://www.ubuntu.com/business/services/cloud

0 packages can be updated.
0 updates are security updates.

Last login: Tue Sep 12 16:35:00 2017 from 204.210.149.122
cc@fox:~$ sudo cp cuda/include/cudnn.h /usr/local/cuda/include
cc@fox:~$ sudo cp cuda/lib64/libcudnn* /usr/local/cuda/lib64
cc@fox:~$ sudo chmod a+r /usr/local/cuda/include/cudnn.h /usr/local/cuda/lib64/libcudnn*
cc@fox:~$
Chameleon Setup

- **Resource type:** bare metal/CHI@TACC
- **Lease:** GPU P100
- **Image:** CC-Ubuntu16.04-CUDA8
- **Packages:**
  - gcc, gfortran
  - python-{numpy scipy matplotlib}
  - tensorflow
  - glances, nvidia-ml-py, screen
Conclusions

- Learning the American Sign Language (ASL) with Auto-encoders
  - Simpler than CNN
  - More efficient than CNN
  - Faster to train than CNN
  - Similar performance to a CNN
  - CNNs are not the panacea in pattern recognition on images or computer vision (no free lunch theorem)
A Deep Learning Approach to Sign Language Recognition using Stacked Sparse Autoencoders

Interested in code? Check Deep’s repo:

https://github.com/DeepDand/research