Deep Learning in Storage

Champak Kumar Dutta, Subhendu Banerjee

Senior Architect
Deep Learning Applications in IT

Deep Learning Applications in Storage
Deep Learning Applications

IT - General
Image Processing Example (CNN)
The easiest way to understand a *convolution* is by thinking of it as a sliding window function applied to a matrix. It becomes quite clear looking at a visualization:

The sliding window is called a *kernel*, *filter*, or *feature detector*. Here we use a $3 \times 3$ filter, multiply its values element-wise with the original matrix, then sum them up.
Image Processing – Overview II

Stock Prediction Example (RNN and LSTM)
Stock Market Example - Process
### Person-Movie Relationship – RBM/Autoenc

<table>
<thead>
<tr>
<th></th>
<th>M1</th>
<th>M2</th>
<th>M3</th>
<th>M4</th>
<th>M5</th>
<th>M6</th>
<th>M7</th>
<th>M8</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P2</td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P3</td>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P4</td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>P6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

Person to Model, Model to Person, Model Strength

SVD (Matrix representation)

\[ A = U \Sigma V^T \]

where \( U \) is an \( m \times m \) orthogonal matrix whose columns are the eigenvectors of \( AA^T \), \( V \) is an \( n \times n \) orthogonal matrix whose columns are the eigenvectors of \( A^T A \), and \( \Sigma \) is an \( m \times n \) diagonal matrix of the form

\[
\Sigma = \begin{pmatrix}
\sigma_1 & & \\
& \ddots & \\
& & \sigma_r & 0 \\
& & & 0 & \ddots & \\
& & & & & 0 & \\
\end{pmatrix}
\]
RBM

https://deeplearning4j.org/restrictedboltzmannmachine
NLP Example

- Term Frequency, Inverse Document Frequency - tfidf
- Word Representation
  - One hot: [1,0,0,0], [0,1,0,0], [0,0,1,0], [0,0,0,1]
- Vector Representation and Cosine Similarity

<table>
<thead>
<tr>
<th></th>
<th>King</th>
<th>Queen</th>
<th>Man</th>
<th>Woman</th>
</tr>
</thead>
<tbody>
<tr>
<td>Familiarity</td>
<td>0.90</td>
<td>0.9</td>
<td>0.02</td>
<td>0.02</td>
</tr>
<tr>
<td>Wealth</td>
<td>0.90</td>
<td>0.99</td>
<td>0.5</td>
<td>0.5</td>
</tr>
<tr>
<td>Gender</td>
<td>….</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Other Attr</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Word2Vec

\[
P(w_t|h) = \text{softmax}(\text{score}(w_t, h)) \]
\[
= \frac{\exp\{\text{score}(w_t, h)\}}{\sum_{\text{Word } w' \text{ in Vocab}} \exp\{\text{score}(w', h)\}}
\]
Deep Learning Applications

IT - Storage
Prefetching
Vector Representation example

- Physical location of block
- File it belongs to
- User who owns the file
- Creation time
- Access time
Use Cases

- VM Migration
  - PreCopy and Post Copy
  - PostCopy results in network fault and copies faulted data. Also prefetches pages
  - Vector representation – Pages belonging to schedulable processes

- Tiering
  - Block movement between Tiers
  - Predicting blocks to be accessed in near future

- NFS - 4.2 has application hint for caching
  - Cache or no cache
  - No application intelligence
  - Local FS – Read ahead size
Capacity/Performance
Use cases

- Power Consumption in Data Center – Historical Power consumption Data, CPU Memory Utilization, IO/Network Workload

- Performance Modelling and Prediction inter-arrival time, and sequential-scan run-length, queue time, seek and rotational latency, transfer time, sequential/random, read/write ratio – CART (Classification and Regression Tree) model
  - Parameter selection – additive and subtractive
    - CART model - CUT points are chosen
    - RBM to get latent features - subsequent regression can find the metric
Predictive Failure
Use cases

“Recently, LSTM autoencoders and encoder-decoder frameworks have been used as reconstruction models where some form of reconstruction error is used as a measure of anomaly. The idea behind such models is: autoencoder is trained to reconstruct the normal time-series and it is assumed that such a model would do badly to reconstruct the anomalous time-series having not seen them during training.”
Miscellaneous
Parameters

- Load Balancing – some of the parameters
  - Latency
  - Response Time
  - Reject connection count

- Generalized Resource Management
  - Protocol Detection
References

http://web.cs.iastate.edu/~cs577/handouts/svd.pdf
https://www.slideshare.net/ananth/word-representation-svd-lsa-word2vec