A Review of Unsupervised Feature Learning and Deep Learning for Time Series

MARTIN JOHAN LANGKVIST
AMY LOUTFI
LARS KARLSSON
Introduction

• Most real world data has a temporal component
  • Natural processes (weather, sound waves)
  • Man-made (stock-market, sensors)
• Traditional modeling approaches (Autoregressive models, Hidden Markov models) cannot be used to model high-dimensional, noisy real world data
• For complex data, develop robust features to capture relevant information
• Hard to develop domain-specific features
  • Expensive
  • Time-consuming
  • Requires expertise of data
• Unsupervised feature learning
  • Uses unlabeled data – easy to obtain
  • Can be stacked to learn complex structures
Properties of Time Series

- Sampled data points taken from continuous process over time
- Noisy and High Dimensional
  - Use dimensionality reduction, wavelet analysis or filtering
  - Loss of information
- Not enough information
  - Financial data – small aspect of complex system
- Non-stationarity
  - Mean, Variance, Frequency change over time
- Invariance
  - Features need to be invariant to translations in time
AR, MA and ARIMA models

- **Autoregressive (AR) Model**
  - Current values can be explained as a function of past values
    \[ x_t = \Phi_1 x_{t-1} + \Phi_2 x_{t-2} + \cdots + \Phi_p x_{t-p} + w_t \]

- **Moving Average (MA) Model**
  - Current values can be explained as a function of white noise
    \[ x_t = w_t + \Theta_1 w_{t-1} + \Theta_2 w_{t-2} + \cdots + \Theta_q w_{t-q} \]

- **(Autoregressive Integrated Moving Average) ARIMA Model**
  - AR, MA with seasonal component
Restricted Boltzmann Machine

• Probabilistic model between input units (visible), and latent units (hidden)
• Used to model static data
• Learn representation in forward pass, reconstruct input in backward pass
• Learning Function:

\[
\frac{\partial \log P(x)}{\partial W_{ij}} \approx \langle x_i h_j \rangle_{data} - \langle x_i h_j \rangle_{recon}
\]

• Can be stacked to form Deep Belief Network (DBN)
• Feature learning, Dimensionality Reduction
Contrastive Divergence Optimization

- KL Divergence:

\[
D_{KL}(p(x) \parallel q(x)) = P(x) \log \frac{p(x)}{q(x)}
\]

- Forward pass: Update the weights of all hidden nodes in parallel.
- Backward pass: Reconstruct the input vector with the same weights used for hidden nodes.
- Compare the input to the reconstructed input based on KL divergence.
- Reconstruct the input vector again and keep repeating for all the input data and for multiple epochs. This is repeated until the system is in equilibrium distribution.
Conditional RBM

- Extension of RBM that models multi-variate time series data
- Autoregressive weights that model short term temporal structures
- Connections between current hidden and past visible units
- Probability of activation:

\[
P(h_j|x) = \sigma \left( b_j + \sum_i W_{ij}x_i + \sum_k \sum_i B_{ijk}x_i(t-k) \right)
\]

\[
P(x_i|h) = \sigma \left( c_i + \sum_j W_{ij}h_j + \sum_k \sum_i A_{ijk}x_i(t-k) \right)
\]
Gated RBM

- Extension of RBM that models transition between two input vectors
- Energy function:
  \[ E(y, z; x) = -\sum_{ij} W_{ij} x_i y_j z_k - \sum_k b_k z_k - \sum_j c_j y_j \]
- Probability of activation:
  \[ P(z_k = 1|x, y) = \sigma(\sum_{ij} W_{ijk} x_i y_j + b_k) \]
- Large number of parameters due to weight tensor
- Impractical for large images
Autoencoder

- Learns efficient encodings of data unsupervised
- Initially used for Dimensionality Reduction
- Input is concatenation of current and past frames
- Cost function:

\[
J(\theta) = \frac{1}{2N} \sum_{n} \sum_{i} (x_i^{(n)} - \hat{x}_i^{(n)})^2 + \frac{\lambda}{2} \sum_{l} \sum_{i} \sum_{j} (W_{ij}^l)^2 + \beta \sum_{l} \sum_{j} KL(\rho || \rho')
\]

- Regularization terms prevent learning 1-to-1 mappings
Recurrent Neural Network

- Used to model sequential data
- Models short time dependency with hidden to hidden connections
- Trained iteratively with back propagation through time
- Very deep network with shared parameters
- Can suffer from vanishing gradients over long sequences
- Not good at finding long term dependencies
  - Alternative is LSTM
Convolution and pooling

- Hidden units not fully connected to inputs
- Convolution used with RBM (convRBM) and Autoencoder (convAE)
- Time Delay Neural Network (TDNN): Convolutions on overlapping time windows
- Pooling: Combine nearby values through max, average etc.
  - Invariant to local distortions, reduce dimensionality
- Space-Time DBN:
  - ConvRBM with spatial pooling layer and temporal pooling layer
  - Invariant features for spatio-temporal data
Hidden Markov Model

- Markov chain: Sequence of events in which the probability of each event depends only on the state attained in the previous event
- Two probability distributions:
  - Transition Distribution – Probability of going from hidden state to next one
  - Observation Distribution – Relation between observed values and hidden states
- Limited representational capacity in hidden states
  - Need $2^N$ hidden states to model $N$ bits of information
- Used in Speech Recognition with Gaussian Mixture Models for discretization
Unsupervised Learning Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Temporal relation</th>
<th>Memory</th>
<th>Typical input size</th>
<th>Generative</th>
</tr>
</thead>
<tbody>
<tr>
<td>RBM</td>
<td>-</td>
<td>-</td>
<td>10-1000</td>
<td>✓</td>
</tr>
<tr>
<td>AE</td>
<td>-</td>
<td>-</td>
<td>10-1000</td>
<td>-</td>
</tr>
<tr>
<td>RNN</td>
<td>✓</td>
<td>1-100</td>
<td>50-1000</td>
<td>✓</td>
</tr>
<tr>
<td>cRBM</td>
<td>✓</td>
<td>2-5</td>
<td>50</td>
<td>✓</td>
</tr>
<tr>
<td>TDNN</td>
<td>✓</td>
<td>2-5</td>
<td>5-50</td>
<td>-</td>
</tr>
<tr>
<td>ANN</td>
<td>-</td>
<td>-</td>
<td>10-1000</td>
<td>-</td>
</tr>
<tr>
<td>GRBM</td>
<td>✓</td>
<td>2</td>
<td>&lt;64x64</td>
<td>✓</td>
</tr>
<tr>
<td>ConvGRBM</td>
<td>✓</td>
<td>2</td>
<td>&gt;64x64</td>
<td>✓</td>
</tr>
<tr>
<td>ConvRBM</td>
<td>-</td>
<td>-</td>
<td>&gt;64x64</td>
<td>✓</td>
</tr>
<tr>
<td>ConvAE</td>
<td>-</td>
<td>-</td>
<td>&gt;64x64</td>
<td>-</td>
</tr>
<tr>
<td>ST-DBN</td>
<td>✓</td>
<td>2-6</td>
<td>10x10</td>
<td>✓</td>
</tr>
</tbody>
</table>
Classical Time Series Problems: Videos

- Series of images over time (Spatio-temporal data)
- Model Transitions between images
  - Use Gated RBM
  - Does not scale to larger images
- Convolutional Gated RBM with max-pooling
  - Reduces number of parameters
  - Allows larger input sizes
  - Handles affine transformations
- ST DBN: Convolutional RBM with spatial pooling layer and temporal pooling layer
  - Action Recognition, Video Denoising
- Most models still not good at learning longer time dependencies
Stock Market Prediction

- Highly complex, difficult to predict
- Trends are non-linear, uncertain and non-stationary
- Efficient Market Hypothesis (EMH): Stock prices follow random walk
- Accuracy can be improved using information extracted from news, social media
- ANN, Recurrent TDNN, RNN
## Summary

<table>
<thead>
<tr>
<th>Problem</th>
<th>Multivariate</th>
<th>Raw data</th>
<th>Frequency rich</th>
<th>Common features</th>
<th>Common method</th>
<th>Benchmark set</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stock prediction</td>
<td>-</td>
<td>✓</td>
<td>-</td>
<td>-</td>
<td>ANN</td>
<td>DJIA</td>
</tr>
<tr>
<td>Video</td>
<td>✓</td>
<td>✓</td>
<td>-</td>
<td>SIFT, HOG</td>
<td>ConvRBM</td>
<td>KTH</td>
</tr>
<tr>
<td>Speech Recognition</td>
<td>-</td>
<td>✔️</td>
<td>✓</td>
<td>MFCC</td>
<td>RBM, RNN</td>
<td>TIMIT</td>
</tr>
<tr>
<td>Music recognition</td>
<td>✓</td>
<td>-</td>
<td>✓</td>
<td>Chroma, MFCC</td>
<td>ConvRBM</td>
<td>GTZAN</td>
</tr>
<tr>
<td>Motion capture</td>
<td>✓</td>
<td>✓</td>
<td>-</td>
<td>-</td>
<td>eRBM</td>
<td>CMU</td>
</tr>
<tr>
<td>E-nose</td>
<td>✓</td>
<td>✓</td>
<td>-</td>
<td>Many</td>
<td>TDNN</td>
<td>-</td>
</tr>
<tr>
<td>Physiological data</td>
<td>✓</td>
<td>✔️</td>
<td>✓</td>
<td>Many, spectrogram</td>
<td>RBM, AE</td>
<td>PhysioNET</td>
</tr>
</tbody>
</table>
Conclusion

- Modeling time series is challenging:
  - High Dimensionality, Non-Linear relationships
  - Long term dependencies in Multivariate signals
  - Use models that use temporal pooling or sequences of hidden unit activations
- Choice of model is highly dependent on data
  - Unsupervised feature learning to find useful features
  - Applying to time series still a challenge
- Deep Learning over shallow approaches
- Model averaging to capture both short and long term dependencies