Feature Engineering for PHM Applications

From Feature Engineering to Feature Learning

Weizhong Yan, PhD, PE
Principal Scientist
Machine Learning Lab
GE Global Research Center
What is Feature Engineering?

Feature engineering is the process of transforming raw data into features that better represent the underlying problem to the predictive models, resulting in improved model accuracy on unseen data.
- Jason Brownlee, Machine Learning Mastery

Feature engineering is manually designing what the input x's should be.
- Tomasz Malisiewicz, vision.ai Co-founder

Feature engineering is the process of using domain knowledge of the data to create features that make machine learning algorithms work better
- Wikipedia

Feature engineering is the act to inject knowledge into a machine learning model
- Anonymous
What is Feature Engineering?

The FE process includes:

- Remove unnecessary and/or redundant variables
- Modify variable data types, e.g., from categorical to numeric
- Combine some of existing variables
- Create new features
- Transform features
- ...

Predictive Modeling Pipeline

Data cleansing → Feature Engineering → Model Building → Model Deployment → Model Updating
Feature engineering is important ...

“Coming up with features is difficult, time-consuming, requires expert knowledge. “Applied machine learning” is basically feature engineering.”  
—Andrew Ng, Stanford University

“At the end of the day, some machine learning projects succeed and some fail. What makes the difference? Easily the most important factor is the features used.”  
- Pedro Domingos, University of Washington
Feature engineering is hard and time-consuming ...

“Good input features are essential for successful machine learning. Feature engineering $\approx 90\%$ of effort in industrial machine learning”

–Yoshua Bengio, University of Montreal
Feature learning alleviates some difficulties of feature engineering ...

... but finding a set of good features is still an unsolved problem

Source: dataRobot.com
Outline

- Big picture
- Feature engineering
- (Shallow) Feature learning
- Deep feature learning
Big picture

Feature Engineering
- Feature extraction
- Feature dim. reduction

Many ways to categorize the methods

Feature selection
Feature low-dim projection

Feature Learning

Shallow feature learning
- Supervised
  - Multiple kernel learning
  - Neural networks
  - Transfer learning
- Unsupervised
  - Clustering
  - Nonlinear embedding
  - Matrix factorization
  - SOM
  - Genetic programming
  - Sparse coding

Deep feature learning
- Unsupervised
  - Deep autoencoder
  - Deep RBM
  - Deep spare coding
- Supervised
  - Deep CNN
  - Deep RNN
  - Deep ELM

- Knowledge based
- Manual, labor intensive
- Domain/problem specific
- Not scalable

- Data driven
- Automated
- Generic
- Scalable
Feature Engineering (FE) (knowledge based)
Characteristics of FE

- Manual, ad hoc
- Time-consuming
- Domain/application specific (as supposed to data specific in feature learning)
- Not optimal
- Not scalable

**Domain specific:** features in one domain do not generalize to other domains

**Domains:**
- PHM
- Computer vision
- Speech recognition
- Text analytics
- Business analytics
- ....

**PHM applications:**
- Vibration analysis
- SHM
- Turbine machines
- Electrical systems
- Electronic devices
- Batteries
- ....

**Vibration analysis**
- Bearings
- Gears
- ....
FE - Feature extraction

Different Technologies
- Statistical analysis
- Signal processing
- Image processing
- Time-series analysis
- Control theory
- Information theory

Different PHM applications
- Vibration analysis
- Turbine machines
- Electrical systems
- Electronic devices
- Batteries
- SHM

Different data types
- Continuous
- Categorical
- Binary
- ...

Time dependency
- Time independent (stationary)
- Time dependent (non-stationary)

Univariate vs. multivariate

Different data sampling rate

• • •
Example: Feature extraction for vibration analysis

**Stationary signals**

- **Time domain**
  - Statistical-based
    - RMS
    - Variation
    - Skewness
    - Kurtosis
    - Crest factor
  - Model-based
    - AR model
    - HMM model
  - Signal processing
    - TSA
    - Correlation
    - Convolution
    - Fractal analysis
    - Correlation dimension

- **Frequency domain**
  - Spectral analysis
  - Envelope analysis
  - Cepstrum analysis
  - Higher order spectrum

**Non-stationary signals**

- **Time-frequency**
  - Short-time Fourier Transform (STFT)
  - Wigner-Ville distribution (WVD)
  - Empirical mode decomposition (EMD)
  - Basis pursuit
  - Spectral kurtosis
  - Cyclostationary analysis

- **Wavelets**
  - Continuous wavelet transform (CWT)
  - Discrete wavelet transform (DWT)
  - Wavelet packet transform
  - Morlet wavelet
  - Hilbert-Huang transform

Yan, W. et al, 2008
FE - Feature dim. reduction

Feature dimensionality reduction

Feature selection

Feature low-dim projection

Filter approaches

Wrapper approaches

Embedded approaches

Search methods

Evaluation criteria

Optimal
Heuristic
Random
Weight-based

Distance-based
Entropy-based
Correlation-based
Relevance-based

Optimal
Heuristic
Random
Weight-based

Accuracy as evaluation criterion

Linear

Non-linear

- PCA
- LDA
- ICA
- Projection pursuit
- Latent semantic indexing

- NPCA or KPCA
- NLDA or KLDA
- MDS
- Principal curves
- Neural networks

Yan, W. et al, 2008
(Shallow) Feature Learning (FL) (data driven)
Shallow feature learning

Including many unsupervised learning, manifold learning, and low-dim projection algorithms

- Clustering, e.g., k-means, GMM
- Matrix factorization, e.g., PCA, ICA, NMF, sparse coding
- Nonlinear embedding, e.g., isomap, LLE, Laplacian eigenmaps, etc., – manifold learning
- Neural networks, e.g., SOM, autoencoder
- Genetic programming
- Sparse coding / dictionary learning
- ...

imagination at work
Shallow feature learning
- k-means clustering

K-means clustering

Project to k cluster center
Shallow feature learning - genetic programming (GP)

GP algorithm

- Initial population
- Evaluation
- Reproduction
- Modification
- Best solution

\[ F = \left( x_1^2 + \frac{x_3}{2.5} \right) \times (x_2 - \sqrt{x_2}) \]
Shallow feature learning
- sparse coding

Natural Images

Test example

\[ x \approx 0.8 \ast \phi_{36} + 0.3 \ast \phi_{42} + 0.5 \ast \phi_{63} \]

Learned bases \((\phi_1, \ldots, \phi_{64})\): “Edges”

[0, 0, ..., 0, 0.8, 0, ..., 0, 0.3, 0, ..., 0, 0.5, ...] feature representation

Deep Feature Learning (FL) (data driven)
What is Deep Learning?

Deep learning is a part of broader family of machine learning methods that involve learning multiple levels of representations of data

Deep learning $\approx$ representation learning
- All deep learning is representation learning, but
- Not all representation learning is deep learning

Deep learning $\neq$ unsupervised learning
- Not all unsupervised learning is deep learning
- Not all deep learning is unsupervised learning
Deep learning in the news

Deep Learning

With massive amounts of computational power, machines can now recognize objects and translate speech in real time. Artificial intelligence is finally getting smart.
Deep learning in the news

IBM acquires AlchemyAPI to bring deep learning to Watson

Microsoft’s Deep Learning Project Outperforms Humans In Image Recognition

Deep Learning Machine Beats Humans in IQ Test and performs between bachelor and masters degree level

NVIDIA GTC: NVIDIA Bets Big On Deep Learning
Deep vs. shallow neural networks

Two-layer (plus input layer) neural networks are an universal approximator

Why deep?

Given the same number of non-linear (neural network) units, a deep architecture is more expressive than a shallow one (Bishop 1995)

Some functions compactly represented with k layers may require exponential size with 2 layers
... However, deep networks have challenges

- Needs labeled data (most data is not labeled)
- Scalability – does not scale well over multiple layers
  - Very slow to converge
  - “Vanishing gradients problem”: errors shrink exponentially with the number of layers

- For more: “Understanding the Difficulty of Training Deep Feed Forward Neural Networks”:
  
The deep breakthroughs


  - Stacked RBMs or AE
  - Layer-wise training with unlabeled data (unsupervised learning)
  - Fine tuning with labeled data
Going deep

googleNet (2014 imageNet competition)

# of layers = 27
Overall # of layers (independent building blocks) = 100
Total # of tunable parameters = 5MM+

Going deeper and deeper...

- **11.2** billion parameters by Google
- **15** billion parameters by Lawrence Livermore National Lab
- **160** billion parameters by Digital Reasoning
- ???
Deep learning has achieved state-of-the-art performance in different areas.

**Deep learning won all competitions**
1. IJCNN Traffic Sign Recognition Competition, 2011
2. ISBI Brain Image Segmentation Contest, 2012
3. ICDAR Chinese hand-writing recognition, 2011
4. MICCAI Mitosis detection grand challenge, 2013
Deep learning applications (products)

- IBM Watson
- Google self-driving cars
- Google Glasses
- Facebook Face recognition
- Facebook user modeling
- Microsoft natural language processing
- Apple Siri

Deep learning has not been used for PHM applications
Unsupervised vs. supervised

**Unsupervised**
- Deep auto-encoder and its variants (AE, DAE, SAE)
- Deep Restricted Boltzmann machines (RBM)
- Deep sparse coding (DSC)

**Supervised**
- Convolutional neural networks (CNN)
- Deep recurrent neural networks (RNN)
- Deep extreme learning machines (ELM)

**Hybrid:** Unsupervised pre-training + supervised fine tuning
Unsupervised deep feature learning is interesting and useful...

In most real-world applications, PHM included, labeled data is sparse (difficult to obtain), while unlabeled data is abundantly available.

H. Lee (2010)
Unsupervised feature learning did well

<table>
<thead>
<tr>
<th>Audio</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>TIMIT Phone classification</strong></td>
<td><strong>Accuracy</strong></td>
<td><strong>TIMIT Speaker identification</strong></td>
</tr>
<tr>
<td>Prior art (Clarkson et al., 1999)</td>
<td>79.6%</td>
<td>Prior art (Reynolds, 1995)</td>
</tr>
<tr>
<td>Stanford Feature learning</td>
<td><strong>80.3%</strong></td>
<td>Stanford Feature learning</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Images</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CIFAR Object classification</strong></td>
<td><strong>Accuracy</strong></td>
<td><strong>NORB Object classification</strong></td>
</tr>
<tr>
<td>Prior art (Krizhevsky, 2010)</td>
<td>78.9%</td>
<td>Prior art (Ranzato et al., 2009)</td>
</tr>
<tr>
<td>Stanford Feature learning</td>
<td><strong>81.5%</strong></td>
<td>Stanford Feature learning</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Video</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Hollywood2 Classification</strong></td>
<td><strong>Accuracy</strong></td>
<td><strong>YouTube</strong></td>
</tr>
<tr>
<td>Prior art (Laptev et al., 2004)</td>
<td>48%</td>
<td>Prior art (Liu et al., 2009)</td>
</tr>
<tr>
<td>Stanford Feature learning</td>
<td><strong>53%</strong></td>
<td>Stanford Feature learning</td>
</tr>
<tr>
<td><strong>KTH</strong></td>
<td><strong>Accuracy</strong></td>
<td><strong>UCF</strong></td>
</tr>
<tr>
<td>Prior art (Wang et al., 2010)</td>
<td>92.1%</td>
<td>Prior art (Wang et al., 2010)</td>
</tr>
<tr>
<td>Stanford Feature learning</td>
<td><strong>93.9%</strong></td>
<td>Stanford Feature learning</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Multimodal (audio/video)</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>AVLetters Lip reading</strong></td>
<td><strong>Accuracy</strong></td>
</tr>
<tr>
<td>Prior art (Zhao et al., 2009)</td>
<td>58.9%</td>
</tr>
<tr>
<td>Stanford Feature learning</td>
<td><strong>65.8%</strong></td>
</tr>
</tbody>
</table>

Other unsupervised feature learning records:
- Pedestrian detection (Yann LeCun)
- Different phone recognition task (Geoff Hinton)
- PASCAL VOC object classification (Kai Yu)

Andrew Ng., ICML 2011
Why unsupervised feature learning works – a simple explanation

- Input space
  - Motorbikes
  - “Non”-Motorbikes

- Feature space
  - “handle”
  - “wheel”

H. Lee (2010)
Auto-encoder – one of the popular DL building blocks

AE: a MLP with output being equal to input

\[ y = s_g(W' h + b_y) \]
\[ h(x) = s_f(W x + b_h) \]

\[ L(x, y) = -\sum (x_i - y_i)^2 \quad \text{OR} \quad L(x, y) = -\sum x_i \log(y_i) + (1 - x_i) \log(1 - y_i) \]
Deep AE

Unsupervised

Supervised

Vincent et al. (2010)
Denoising autoencoder (DAE)

3 different corruption processes:
1. Gaussian noise
2. Masking noise
3. Salt-and-pepper noise
Stacked DAE

2 design settings:
1. Unsupervised feature learning + standalone supervised learning
2. Deep neural network: add logistic regression on top of encoder and supervised fine tune all parameters
A deep feature learning example: Combustor anomaly detection
Gas Turbine Combustor Anomaly Detection

The business pain points

- Current rule-based engine has an insufficient detection rate (*)
- Finding a good set of features (Feature Engineering) takes significant amount of effort
- Labeled data, especially faulty data, is extremely sparse and difficult to get

(*) Source: Reliability combustion events 2008-2010, with M&D data, covering 7&9 E & F class with full-load condition.
The Data

- Single turbine (TSNxxxxxx)
- Normal (event-free) data: 3 months of data (once per minute)
- POD events: 10 events occurred over 4-month window
- 27 sensor measurements (TC readings)
- Data matrices:
  - 13,791 x 27 - normal data for feature learning
  - 300 x 27 - POD events(*)
  - 47,575 x 27 - normal data for model building & validation

(*) For POD cases, take 30 points before the POD events
Our goal is to compare learned features against handcrafted features in terms of classification performance.
Domain-driven, handcrafted features

A sample TC profile

Extracted 12 features

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>DWATT</td>
</tr>
<tr>
<td>2</td>
<td>TNH</td>
</tr>
<tr>
<td>3</td>
<td>max</td>
</tr>
<tr>
<td>4</td>
<td>mean</td>
</tr>
<tr>
<td>5</td>
<td>std</td>
</tr>
<tr>
<td>6</td>
<td>median</td>
</tr>
<tr>
<td>7</td>
<td># diff b/w positive &amp; negative TCs</td>
</tr>
<tr>
<td>8</td>
<td>zero crossing</td>
</tr>
<tr>
<td>9</td>
<td>kurtosis</td>
</tr>
<tr>
<td>10</td>
<td>skewness</td>
</tr>
<tr>
<td>11</td>
<td>max of 3-pt sum</td>
</tr>
<tr>
<td>12</td>
<td>max of 3-pt median</td>
</tr>
</tbody>
</table>
Deep feature learning

Layer 1 DAE

Layer 2 DAE

ELM classifier

12 learned features
Learned features

F1

F2

F3

F4

F5

F6

F7

F8

F9

F10

F11

F12
Classification Modeling and Results

Modeling details:
- ELM (a special type of feed-forward Neural network) as the classifier
- Unbalanced data strategy: sample weighting
- Validation method: 5-fold cross-validation (10 times of random runs)

Deep learning can learn features that give much better detection rate than manually-handcrafted features do.
Feature discovery (both FE and FL) is more important than model building, yet it is less well-studied.

Feature discovery, not model building, can be the differentiator.
Final Remarks - 2

- Traditional knowledge-driven feature engineering is hard and time-consuming, thus is insufficient.
- Feature learning, especially recently-developed deep feature learning, is data-driven, and has some potential in alleviating difficulties faced in FE.

2 directions worth pursuing:
- Integrating domain knowledge into feature learning (R)
- Tools that can automate feature discovery (D)

1 question to be answered: Is deep learning effective for PHM applications?
Thank You

Questions?

My contact information:
Dr. Weizhong Yan
Principal Scientist
Machine Learning Lab
GE Global Research Center
Email: yan@ge.com