Processing of missing data by neural networks

Reproduced by: Kai Lin, Lulu Meng, Yiwen Su, Zhanhong Tian

Dec. 05 2019

In this project, the authors want to introduce a general, theoretically justified methodology for feeding neural networks with missing data.

The authors also want to model the uncertainty on missing attributes by probability density functions.
How to process missing data?
Motivation

- Collecting data is an important process for supervised learning. But sometimes data collected may have some missing fields. Discarding these data is a waste.
- Due to the great interest in deep learning in recent years, it is important to establish unified tools for practitioners to process missing data with arbitrary neural networks.
- This paper tries to figure out whether we can use data with missing value to train our model, how well the filling method performs, and whether it can be easily applied.
Background

- Neural Network
- Expectation–Maximization Algorithm
- GMM
Related Work

Generating candidates for filling missing attributes

- Use mean/mode to fill the data
- Train another model to predict the data (Neural Network/Extreme Learning Machine/K-Nearest Neighbors)
- Generative Adversarial Net (GAN)

Building Probabilistic model of incomplete data (Make assumption about the data)
e.g. MNIST Dataset

Adding Mask Randomly
Simple Solution

- Neural Network
- Extreme learning machines
- ....
Proposed Approach

Paper’s solution
Missing data representation:

Data point: \((x, J)\)

\[ S = \text{Aff}[x, J] = x + \text{span}(e^J) \]

Generalized neuron’s response:

\[
F_S(x) = \begin{cases} 
\frac{1}{\int_S F(s)ds} F(x), \text{ for } x \in S, \\
0, \text{ otherwise.}
\end{cases}
\]

\[
n(F_S) = E[n(x)|x \sim F_S] = \int n(x)F_S(x)dx.
\]
NR (Auxiliary function) Derivation

\[ NR(w) = \text{ReLU}[N(w, 1)], \]

\[ \text{ReLU}[N(m, \sigma^2)] = \sigma NR\left(\frac{m}{\sigma}\right). \]

\[ NR(w) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{w^2}{2}\right) + \frac{w}{2} (1 + \text{erf}\left(\frac{w}{\sqrt{2}}\right)); \]

\[ \text{erf}(z) = \frac{2}{\sqrt{\pi}} \int_{0}^{z} \exp(-t^2) dt. \]
Key Formula for the program:
Mixture of nondegenerate gaussians in affine space

\[
F_S^\gamma = \sum_i r_i N(m_i^S, \Sigma_i^S),
\]

\[
m_i^S = [x_J, (m_i)_{J'}], \quad \Sigma_i^S = [0_{J', J'}, (\Sigma_i)_{JJ'}],
\]

\[
r_i = \frac{q_i}{\sum_j q_j}, \quad q_i = C_{m_i, \Sigma_i, S}^\gamma \cdot p_i,
\]

\[
C_{m, \Sigma, S}^\gamma = \frac{1}{(2\pi)^{(D-|J|)/2} \prod_{l \in J'} (\gamma + \sigma_l)^{1/2}} \cdot \exp(-\frac{1}{2} \sum_{l \in J'} \frac{1}{\gamma + \sigma_l} (m_l - x_l^2)).
\]
Proposed Solution

**Missing data representation:** EM algorithm to estimate incomplete data density with the use of Gaussian Mixture Model. Then based on the data density, we can infer what the incomplete data should be. And the parameter of GMM is learnt by using neural network.

**Generalized neuron’s response:** With changing of the first layer of the network structure, the method can be easily adapted to existing network structure to deal with the problem of incomplete data.
The Output Parameters of GMM

We get these parameters from GMM as input to the convolutional layer:

1. The weights of each mixture component.
2. The mean of each mixture component.
3. The covariance of each mixture component.
4. A tensor of the specified shape filled with random normal values.

These parameters will be updated within the network structure (encoder-decoder network) to better model the data.
Network architecture

The author proposed to use encoder and decoder to solve the missing data of MNIST dataset.
Loss Function and Optimization Method

Loss Function: **mean-square error**

The loss inside the mask, outside the mask and total area can all be calculated. Since the input of model has no complete data, the loss function only based on outside the mask area.

Optimization Method: **RMSProp**
Implementation

Adaptation of a given neural network to incomplete data relies on the 2 steps:

- **Estimation of missing data density with the use of mixture of diagonal Gaussians.**

If data satisfies missing at random assumption, EM algorithm should be used to estimate incomplete data density with the use of GMM.

**general case:**

The network should learn optimal parameters of GMM with respect to its cost function.
Adaptation of a given neural network to incomplete data relies on the 2 steps:

- **Generalization of neuron’s response.**
  
  Generalizing the activation functions of all neurons in the first hidden layer of the network to process probability measures. The modification only presents on the first hidden layer.

**The paper performs several implementations:**

- Reconstruction of incomplete MNIST images
- Running RBFN on 8 examples retrieved from UCI repository
Data Summary

2 kinds of data set:

1. MNIST with a removed square patch on each image (Uniformly sample the location)

2. UCI repo two-class data sets with internally missing attributes
Experimental Results & Analysis

Evaluated in 2 types of architectures:

1. Autoencoder (AE)
   Structure: 5 hidden layers with 256, 128, 64, 128, 256 neurons (first use ReLU, rests use sigmoids) in encoder.

Results: After running for 100 epochs, it gives sharper images and lower error.
Experimental Results & Analysis

Evaluated in 2 types of architectures:

2. Radial basis function network (RBFN)

Structure: 1 hidden layer and softmax in the output layer applied with cross-entropy function

<table>
<thead>
<tr>
<th>data</th>
<th>karma</th>
<th>geom</th>
<th>k-nn</th>
<th>mice</th>
<th>mean</th>
<th>gmm</th>
<th>dropout</th>
<th>our</th>
<th>CE</th>
</tr>
</thead>
<tbody>
<tr>
<td>bands</td>
<td>0.580</td>
<td>0.571</td>
<td>0.520</td>
<td>0.544</td>
<td>0.545</td>
<td>0.577</td>
<td>0.616</td>
<td>0.598</td>
<td>0.621</td>
</tr>
<tr>
<td>kidney</td>
<td>0.995</td>
<td>0.986</td>
<td>0.992</td>
<td>0.992</td>
<td>0.985</td>
<td>0.980</td>
<td>0.983</td>
<td>0.993</td>
<td>0.996</td>
</tr>
<tr>
<td>hepatitis</td>
<td>0.665</td>
<td>0.817</td>
<td>0.825</td>
<td>0.792</td>
<td>0.825</td>
<td>0.820</td>
<td>0.780</td>
<td>0.846</td>
<td>0.843</td>
</tr>
<tr>
<td>horse</td>
<td>0.826</td>
<td>0.822</td>
<td>0.807</td>
<td>0.820</td>
<td>0.793</td>
<td>0.818</td>
<td>0.823</td>
<td>0.864</td>
<td>0.858</td>
</tr>
<tr>
<td>mammogr.</td>
<td>0.773</td>
<td>0.815</td>
<td>0.822</td>
<td>0.825</td>
<td>0.819</td>
<td>0.803</td>
<td>0.814</td>
<td>0.831</td>
<td>0.822</td>
</tr>
<tr>
<td>pima</td>
<td>0.768</td>
<td>0.766</td>
<td>0.767</td>
<td>0.769</td>
<td>0.760</td>
<td>0.742</td>
<td>0.754</td>
<td>0.747</td>
<td>0.743</td>
</tr>
<tr>
<td>winconsin</td>
<td>0.958</td>
<td>0.958</td>
<td>0.967</td>
<td>0.970</td>
<td>0.965</td>
<td>0.957</td>
<td>0.964</td>
<td>0.970</td>
<td>0.968</td>
</tr>
</tbody>
</table>

Our experiment’s result:

(training epochs 50; learning rate: 0.25; batch size: 75; hidden layer: 25; distribution: 3)

bands: Train: 0.6089 test: 0.5056
kidney: Train: 0.8539 test: 0.8625
hepatitis: Train: 0.7135 test: 0.6978
horse: Train: 0.7291 test: 0.7555
Conclusion and Future Work

- The paper presented a general approach for adapting neural networks to process incomplete data.
- The paper’s approach can be used for a wide range of networks.
- The paper gives comparable results to the other methods, some of which even require complete data in training.
Task Allocation

Yiwen Su: Revise the presentation slides, analyze the source code structure and reproduce the result of the paper (AE). Try different methods to fill in the blank in the data, including KNN, softmulte and mean.

Zhanhong Tian: Revise the presentation slides, reproduce some of the result of the paper. Finding out one wrong description of formula in the paper. Visualize the data in MNIST.

Lulu Meng: Revise the presentation slides, reproduce RBFN model binary classification experiment of the paper.

Kai Lin: Revise the presentation slides, derive the formula in the paper.
Thanks!
Any Questions?
After training with neural network, we can get a GMM, where its first component estimates a density of class 1, while the second component matches class 2, which means it can help perform better classification than estimating GMM directly by EM algorithm.
References

References


References


