Semi-supervised Learning

ZHAO YANG MIN

Fudan Univ.

April 24, 2017

Content

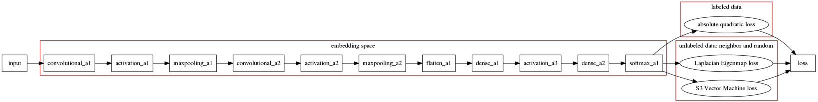
1. Deep Learning
2. DRBM
3. S3VM
4. Ladder Network
5. Label Spreading

Semi-supervised Deep Learning

SSDL takes the neural network as a mapping function and maps data(labeled or unlabeled) to feature space

$$f(x) \rightarrow y$$

Simple Model



The feature mapping(neural network) is the same for labeled and unlabeled data. But because of the absence of label, loss function would be different.

$$\sum_{i=1}^{M} l(f(x_i), y_i) + \lambda \sum_{i,j=1}^{M+U} L(f(x_i), f(x_j), W_{i,j})$$

For labeled data

The fisrt term is labeled data loss:

$$\sum_{i=1}^{M} l(f(x_i), y_i)$$

At first I used hinge loss:

$$\mathcal{L}(y, y_t) = \max(0, 1 - y_t \cdot y)$$

But later I found absolute quadratic loss maybe better:

$$\mathcal{L}(y, y_t) = \max(0, 1 - |y_t \cdot y|)^2$$

table, page 5

For unlabeled data

The paper proposed 3 embedding algorithms:

1. Multidimensional scaling

$$L(f(x_i), f(x_j), W_{i,j}) = (||f_i - f_j|| - W_{ij})^2$$

2. ISOMAP
3. Laplacian Eigenmaps

$$\sum_{i,j} L(f(x_i), f(x_j), W_{i,j}) = \sum_{i,j} W_{ij}||f_i - f_j||^2$$

4. Siamese Networks

$$L(f_i, f_j, W_{ij}) = \max(0, m - ||f_i - f_j||_2)^2, \text{ if } W_{ij} = 0$$

W_{ij} indicates the neighbor relationship.

Most Computation

W_{ij}

Different W_{ij} would take very different time.

1. k-N
   slow
2. pixel
   fast

This is because k-N involves computing distance and ordering:

$$\sqrt{(\mathbf{x_1} - \mathbf{x_2})^T(\mathbf{x_1} - \mathbf{x_2})}$$

Difficulty of Implementation

1. Loss
   It is difficult to represent W_{ij} in tensorflow. Code trick ...
2. Batch
   This algorithm cannot perform batch computation in tensorflow, though its loss is written as:

$$\sum_{i=1}^{M} l(f(x_i), y_i) + \lambda \sum_{i,j=1}^{M+U} L(f(x_i), f(x_j), W_{i,j})$$

3. Weight Reusing
   Define layers outside function in Keras.

Improve: Add S3VM loss

The loss described in paper:

$$L_{\text{supervised}} + \alpha L_{\text{manifold}}$$

But we can add loss based on cluster to compare:

$$L_{\text{supervised}} + \alpha L_{\text{manifold}} + \beta L_{\text{cluster}}$$

s.t. $\beta + \alpha = 1$.
I used S3VM loss here:

$$\sum_{\text{labeled}} \max(0, 1 - y_i f(x_i)) + \lambda_1 + ||h||^2_{\mathcal{HK}} + \lambda_2 \sum_{\text{unlabeled}} \max(0, 1 - |f(x_i)|)$$

Preference: table, page 4

Careful points

One-hot label ...... nan problem
Distance:
good LE:

$$\sum_{ij} L(f_i, f_j, W_{ij}) = \sum_{ij} W_{ij}||f_i - f_j||^2$$

bad LE:

$$\sum_{ij} L(f_i, f_j, W_{ij}) = \sum_{ij} W_{ij}||f_i - f_j||$$

good SN:

$$L(f_i, f_j, W_{ij}) = \max(0, m - ||f_i - f_j||_2)^2, \text{ if } W_{ij} = 0$$

bad SN:
$||f_i - f_j||_2 = (f_i - f_j)^T(f_i - f_j)$

Improve

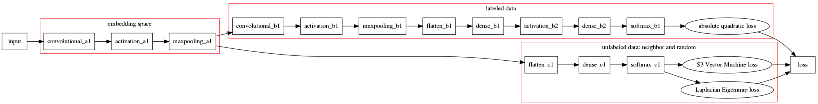
1. Change Embedding Algorithm
   Use Laplacian Eigenmaps instead of Siamese Networks. Table, page 5
2. Absolute Quadratic Loass

   $$\mathcal{L}(y, y_t) = \max(0, 1 - |y_t \cdot y|)^2$$

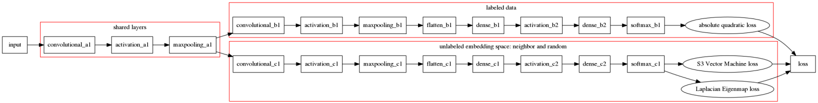
   table, page 5
3. Neighbor Radius
   R = 1,2,3: table, page 5
4. Auxiliary EmbedCNN

Improve

NN Auxiliary:



CNN Auxiliary:



S3VM

Memory Problem
Cannot find Multi-label Semi-supervised SVM

DRBM

Structure and loss:

$$\mathcal{L}_{\text{semi-sup}}(\mathcal{D}_{\text{train}}, \mathcal{D}_{\text{unlab}}) = \mathcal{L}_{\text{TYPE}}(\mathcal{D}_{\text{train}}) + \beta \mathcal{L}_{\text{unsup}}(\mathcal{D}_{\text{unlab}})$$

$$= -\sum_{i=1}^{|\mathcal{D}_{\text{train}}|} \log p(y_i|\mathbf{x_i}) - \beta \sum_{i=1}^{|\mathcal{D}_{\text{unlab}}|} \log p(\mathbf{x_i})$$

or

$$= -\sum_{i=1}^{|\mathcal{D}_{\text{train}}|} \log p(y_i|\mathbf{x_i}) - \alpha \sum_{i=1}^{|\mathcal{D}_{\text{train}}|} \log p(y_i, \mathbf{x_i}) - \beta \sum_{i=1}^{|\mathcal{D}_{\text{unlab}}|} \log p(\mathbf{x_i})$$

DRBM - Implementation

Turn the gradient for U, d zero:
code, line 322

Other Algorithms

1. Label Spreading
2. Ladder Network