The Application of Extreme Learning Machine based on Gaussian Kernel in Image Classification

Weijie LI, Yi LIN

Postgraduate student in College of Survey and Geo-Informatics, tongji university
Email: 1633289@tongji.edu.cn
Outline

- ELM Theory
- Kernel ELM (K-ELM)
- Classification Application
- Conclusion and Outlook
Study background

- During the environmental monitoring research process, it was found that machine learning can better apply remote sensing image classification, compared with SVM and ELM classification algorithms, and K-ELM has higher classification accuracy.

- For small-area research areas, Gaussian kernel ELM has more obvious and effective effects, and more realistic identification of feature information.
SLFN theory

- **Input layer**: \( n \) neurons, corresponding to the \( n \) input features
- **Hidden layer**: include \( l \) neurons
- **Output layer**: \( m \) neurons corresponding to the \( m \) output labels

For \( N \) arbitrarily determined sample \((x_i, y_i)\),

\[
x_i = \begin{bmatrix} x_{i1}, x_{i2}, \ldots, x_{in} \end{bmatrix} \quad x_i : input\ label
\]

\[
y_i = \begin{bmatrix} y_{i1}, y_{i2}, \ldots, y_{in} \end{bmatrix} \quad y_i : output\ result
\]

Input layer to hidden layer weight:

\( \omega \)

Hidden layer to output layer weight:

\( \beta \)
SLFN theory

SLFN has the aspect to improve:

- Training speed is slow. Since the gradient descent method requires multiple iterations to achieve the purpose of correcting the weights and thresholds, the training process takes a long time.

- Easy to fall into local minimum values, unable to reach the global minimum;

So Professor Huang conducted an in-depth study on the SLFN and proposed the ELM theory.
SLFN theory

SLFN to ELM

- The active function $g(\omega x + b)$ satisfies infinitely differentiable in the arbitrary interval $R \times R$, then $\omega$ and $b$ can be randomly generated from any continuous probability distribution in any interval of $R$-space.

- Compared SLFN, there is no need to adjust the $\omega$ and $b$, then the entire network only has the output weight $\beta$ is not determined. Extreme learning machine come into being

$$H_{N \times l} \beta_{l \times m} = Y_{N \times m}$$ (1)

$H$ is the hidden layer output matrix of training set ,
$Y$ is the target matrix of training set
$\beta$ is the weight from hidden layer to output layer
ELM model:

\[ H \beta = Y \]

\[ \beta = \begin{bmatrix} \beta_1^T \\ M \\ \beta_l^T \end{bmatrix}_{l \times m} \quad \quad Y = \begin{bmatrix} y_1^T \\ M \\ y_N^T \end{bmatrix}_{N \times m} \]

\[ H = \begin{bmatrix} g(\omega_1x_1 + b_1) & g(\omega_2x_1 + b_2) & \cdots & g(\omega_lx_1 + b_l) \\ g(\omega_1x_2 + b_1) & g(\omega_2x_2 + b_2) & \cdots & g(\omega_lx_2 + b_l) \\ \vdots & \vdots & \ddots & \vdots \\ g(\omega_1x_N + b_1) & g(\omega_2x_N + b_2) & \cdots & g(\omega_lx_N + b_l) \end{bmatrix}_{N \times l} \]

\[ g : \text{the active function} \]

\[ \omega : \text{the weight from input to hidden layer} \]

\[ b : \text{bias} \]

\[ x : \text{input the data label} \]
ELM theory

Solution of $\beta$:

$$H_{N \times l} \beta_{l \times m} = Y_{N \times m}$$

- When $L=N$,

$$\hat{\beta} = H^{-1}Y$$  \hspace{1cm} (3)

- When another, $H$ matrix is ill-condition, need to be solved according to the minimum norm criterion

$$\left\| H(\omega_i, x_i, b_i) \cdot \beta_i - y_i \right\| = \min \hspace{1cm} (4)$$

$$\hat{\beta} = \arg \min \min \left\| Y - H \beta \right\|_F$$  \hspace{1cm} (5)

$$\hat{\beta} = H^+Y$$  \hspace{1cm} (6)

$H^+$: Moore-Penrose Generalized Inverse of Implicit Layer Output Matrix
In order to increase the stability and generalization ability of the ELM, regularization parameters can be added to the ELM. Model is as following:

\[
L(X, Y; \beta, C) = \|H\beta - Y\|^2 + \frac{1}{C}\|\beta\|^2
\]  

(7)

At this point, the weight matrix \(\beta\) is estimated as:

\[
\beta = H^T \left(\frac{I}{C} + HH^T\right)^{-1} Y \quad \text{N}<\text{L}
\]  

(8)

\[
\beta = \left(\frac{I}{C} + HH^T\right)^{-1} H^T Y \quad \text{N}\gg\text{L}
\]  

(9)

\[
f(x) = h(x)H^T \left(\frac{I}{C} + HH^T\right)^{-1} Y
\]  

(10)
Kernel function

It can map data from low-dimensional to high dimensions, while at the same time transforming scalar product operations from high-dimensional space into low-dimensional calculations.

\[ K(x, x') = \phi(x)^T \phi(x') \]  

(11)

The classes can more easily separated in a higher-dimensional space.
Kernel function

- For the case where the number of training sample is not huge.
- If a feature mapping $h(x)$ is unknown to users, the dimensionality $L$ of the feature space (number of hidden nodes) need not be given either.
- Instead, its corresponding kernel $K(u,v)$ is given to users.

\[
f(x) = h(x)H^T \left( \frac{I}{C} + HH^T \right)^{-1}Y
\] (12)

\[
f(x) = h(x)H^T (HH^T)^{-1}Y = \begin{bmatrix} K(x,x_1) \\ \vdots \\ K(x,x_N) \end{bmatrix}^T (\frac{I}{C} + \Omega_{ELM})^{-1}Y
\] (13)
Kernel function

Kernel function other style

A powerful way to construct new kernel functions is to use simple kernel functions as basic modules. Given a legal kernel function \( k_1(x, x') \) and \( k_2(x, x') \) The following new kernel functions are also legal

\[
\begin{align*}
k(x, x') &= \phi(x)^T \phi(x') \\
k(x, x') &= c k_1(x, x') \\
k(x, x') &= k_1(x, x') + k_2(x, x') \\
k(x, x') &= k_1(x, x')k_2(x, x')
\end{align*}
\]
Gaussian kernel function

One of the kernel functions

Usually defined as the Euclidean distance between any point \( x \) in space and a certain center \( x_c \), effect is often local, the function takes a small value when \( x \) is away from \( x_c \).

\[
k \left( \| x - x_c \| \right) = \exp \left\{ - \frac{\| x - x_c \|^2}{2\sigma^2} \right\}
\]  

(15)

\( x_c \): kernel function center,

\( \sigma \): function width parameters, control local range of action

Cross validation to determine \( \sigma \)
Gaussian kernel function

An extreme learning machine model with Gaussian kernel function can be expressed as

\[
f(x) = h(x)H^T (HH^T)^{-1}Y = \begin{bmatrix} K(x,x_1) \\ \vdots \\ K(x,x_N) \end{bmatrix}^T \left( \frac{I}{C} + \Omega_{ELM} \right)^{-1} Y \tag{16}
\]

\[
k(x,x') = k_{Gauss}(x_{Gauss}, x'_{Gauss}) \tag{17}
\]

\[
G(x, x') = \exp \left( -\frac{\|x - xc\|^2}{2\sigma^2} \right) \tag{18}
\]
Classification application

- Study area

Band number: 7
Satellite: landsat-8
Study image: 554 × 591 pixels
Study region: Chao Lake, China
Image resolution: 30m × 30m

Figure 1: Original image
Classification application

- Comparison of classification results

Figure 3 ELM classification result

Figure 4 Gaussian kernel ELM classification result

\[ \sigma = 86.595 \]
## Classification application

### Classification accuracy comparison

<table>
<thead>
<tr>
<th></th>
<th>building</th>
<th>Forest</th>
<th>Arable-land</th>
<th>Bare-land</th>
<th>Water</th>
</tr>
</thead>
<tbody>
<tr>
<td>building</td>
<td>403</td>
<td>7</td>
<td>0</td>
<td>100</td>
<td>11</td>
</tr>
<tr>
<td>Forest</td>
<td>0</td>
<td>301</td>
<td>43</td>
<td>9</td>
<td>20</td>
</tr>
<tr>
<td>Arable-land</td>
<td>0</td>
<td>1</td>
<td>323</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Bare-land</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>305</td>
<td>19</td>
</tr>
<tr>
<td>Water</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>311</td>
</tr>
</tbody>
</table>

**Accuracy** = 0.8857  \( \kappa = 0.8570 \)

<table>
<thead>
<tr>
<th></th>
<th>building</th>
<th>Forest</th>
<th>Arable-land</th>
<th>Bare-land</th>
<th>Water</th>
</tr>
</thead>
<tbody>
<tr>
<td>building</td>
<td>514</td>
<td>1</td>
<td>0</td>
<td>6</td>
<td>0</td>
</tr>
<tr>
<td>Forest</td>
<td>1</td>
<td>353</td>
<td>11</td>
<td>8</td>
<td>0</td>
</tr>
<tr>
<td>Arable-land</td>
<td>23</td>
<td>1</td>
<td>300</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Bare-land</td>
<td>22</td>
<td>48</td>
<td>0</td>
<td>256</td>
<td>0</td>
</tr>
<tr>
<td>Water</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>309</td>
</tr>
</tbody>
</table>

**Accuracy** = 0.9337  \( \kappa = 0.9159 \)
Classification application

- Gaussian kernel function is suitable for the small-region

The role of the Gaussian kernel function is often local. Gaussian kernel function ELM classifier has obvious better results for small plaque region classification.

Figure 5 Original image  
Figure 6 Standard ELM result  
Figure 7 Gaussian kernel ELM result
Conclusion and outlook

- **Conclusion**

  - Gauss-K-ELM can make classification system more steady, when you add regularization, classification result will not change a lot.

  - Gaussian kernel function can improve classification accuracy, especially for small area research objects, the effect is more significant.

- **Outlook**

  - Gaussian kernel function runs too slowly. We are working on trying other kernel function for study area, such as polynomial kernel function, mixed kernel function.

  - Because the label is an important factor in classification, the next study of the kernel function will start by constructing different feature spaces.
Thank you !