Self-organizing neural networks for unsupervised classification of complex landscapes by polarimetric SAR data

Fabio Del Frate, Marco Del Greco, Giovanni Schiavon, Domenico Solimini
Earth Observation Laboratory, DISP, Tor Vergata University, Rome, Italy

Cosimo Putignano
GEO-K, Roma, Italy

GEO-K
Aim

To assess discrimination capability of self-organizing Neural Networks fed by polarimetric L-band data acquired on a complex landscape.
Planning

SAR data analysis

SOM algorithm development

Data classification

Ground-truth comparison

CPW comparison
Data set

- collected in summer 1991 by JPL/NASA AirSAR on Montespertoli, a rural area SW of Florence, Italian test site of MAC-Europe campaign

- complex hilly landscape (woodlands, agricultural, urban)

- L-band, polarimetric, $\theta = 50^\circ$, 16 looks, 12m x 6.6m drawn from ERA-ORA European project database (http://eraora.disp.uniroma2.it/)
Ground-truth

A: alfalfa
B: bare soil
C: mine
M: corn
OL: olivegrove
P: pasture
R: rape
S: sorghum
SF: sunflowers
U: untilled
UR: manufacts and urban
V: vineyard
W: wheat
Y: arboreus and forest
**CPW classification**


- **Polarimetric data**
- **H, A, α parameters synthesis**
- **Points location in the 3D space** (H, A, α)
- **Classify according to the position in the** (H, A, α) **space**

**Disadvantage:**
**Fixed number of subspaces**
A neural network model typically consists of computational elements or nodes linked through weights which adapt iteratively to attain an optimal performance for the classification case. The nets used as classifiers are Self-Organizing Maps (Kohonen).

The principal goal of the self-organizing map is to transform non linear statistical relationships among high-dimentional data into simple geometric relationships usually represented by regular two dimensional grid of nodes.
SOM model

\[ y_j = \sum_{i=1}^{n} w_{ji} x_i \]
\[ x = [x_1, x_2, \ldots, x_n]^T \]
\[ w_j = [w_{j1}, w_{j2}, \ldots, w_{jn}]^T \]
\[ y = [y_1, y_2, \ldots, y_m]^T \]

where \( j = 1, 2, \ldots, m \) (\( m \): total number of neurons)

Each output neuron is fully linked with the input vector
Learning algorithm

- **Competition**: a continuous input space of activation patterns is mapped onto a discrete output space of neurons by a process of competition among the neurons in the network. The neuron with weight vector $\mathbf{W}$ nearest to the input vector $\mathbf{X}$ is declared "winner".

  $i_{\text{winner}}(\mathbf{x}) = \arg\min_j ||\mathbf{x} - \mathbf{w}_j||$

- **Cooperation**: the winner neuron is linked to its neighbourhood and in this area the synaptic weight will be updated.

- **Synaptic Adaptation**: the synaptic weight vector $\mathbf{w}_j$ of winning neuron is moved toward the input vector $\mathbf{x}$. Upon repeated presentations of the training data, the synaptic weight vector tend to follow the distribution of the input vectors due to the neighborhood updating $\rightarrow$ topological ordering
**SOM model**

The SOM (Self-Organizing Map) model is a neural network model used for clustering and visualization of high-dimensional data. It organizes the input vectors onto a regular topology, such as a grid of neurons. The model has an input layer and an output layer, where the output layer neurons are arranged in a grid.

The winner neuron is determined by the input vector $x$ as follows:

$$i_{\text{winner}}(x) = \arg\min_j \| x - w_j \|$$

The weight of the winner neuron is updated as follows:

$$w_k(t+1) = w_k(t) + \eta \cdot \lambda_i \cdot (x(t) - w_k(t))$$

Where:
- $w_k(t)$ is the weight of neuron $k$ at time $t$.
- $\eta$ is the learning rate.
- $\lambda_i$ is the neighborhood function.
- $x(t)$ is the input vector at time $t$.
- $w_k(t)$ is the weight of neuron $k$ at time $t$.
How SOM works

Input x Polarimetric data → Trained SOM

Winner neuron location in the map

Winner neuron synaptic weight vector

classifier
codifier
Classification algorithm

Features:

- Codify & classify
  SOM 4x4 → 2x1
- Dichotomous
- Iterative
Designing the Algorithm

Surface scattering

- $HH < VV$
- $\varphi_{HH} - \varphi_{VV} \approx 0^\circ$

Double bounce scattering

- $HH > VV$
- $\varphi_{HH} - \varphi_{VV} \approx \pm \pi$

Volumetric scattering

- HV high
- $\varphi_{HH} - \varphi_{VV} \approx 0^\circ$

Buildings and Forest feature:
High HV values

Buildings, Forest

soil, crops, olivegrove, vineyard
Designing the Algorithm

Buildings, Forest

- HH, HV, HH-VV phase

- Buildings
  - Dense forest
  - Sparse forest

- Forest

- Rape, olive-grove, vineyard
  - HH, HV
  - HH, HH-VV phase
  - Sunflowers
  - Other crops

- soil, crops, olive-grove, vineyard
  - HH, HV

- Olive-grove, vineyard
  - HH, HV, VV

- Other crops

Designing the Algorithm
NN classification

- LD forest
- HD forest
- Urban
- Olive-grove, Vineyard
- Sunflower
- Colza
- Other crops
## Confusion matrix

**Overall Accuracy** = \( \frac{11977}{16566} \) 72%                **Kappa Coefficient** = 0.64

<table>
<thead>
<tr>
<th>Class</th>
<th>LD forest</th>
<th>HD forest</th>
<th>Urban</th>
<th>Rape</th>
<th>V&amp;OL</th>
<th>Sunflowers</th>
<th>Surface Scattering</th>
</tr>
</thead>
<tbody>
<tr>
<td>LD forest</td>
<td>54.5</td>
<td>21.1</td>
<td>3.5</td>
<td>11.0</td>
<td>7.2</td>
<td>0</td>
<td>2.2</td>
</tr>
<tr>
<td>HD forest</td>
<td>32.1</td>
<td>76.2</td>
<td>7.9</td>
<td>0.4</td>
<td>0.1</td>
<td>0</td>
<td>0.3</td>
</tr>
<tr>
<td>Urban</td>
<td>0</td>
<td>2.3</td>
<td>86.9</td>
<td>0.2</td>
<td>0</td>
<td>0</td>
<td>0.1</td>
</tr>
<tr>
<td>Colza</td>
<td>3.6</td>
<td>0.1</td>
<td>1.3</td>
<td>70.6</td>
<td>7.5</td>
<td>4.8</td>
<td>3.7</td>
</tr>
<tr>
<td>V&amp;OL</td>
<td>9.8</td>
<td>0.3</td>
<td>0</td>
<td>8.2</td>
<td>74.9</td>
<td>10.9</td>
<td>5.7</td>
</tr>
<tr>
<td>Sunflowers</td>
<td>0</td>
<td>0</td>
<td>0.4</td>
<td>9.3</td>
<td>6.1</td>
<td>59.8</td>
<td>18.5</td>
</tr>
<tr>
<td>Surface Scattering</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.5</td>
<td>4.3</td>
<td>24.5</td>
<td>69.5</td>
</tr>
</tbody>
</table>
Comparison

NN

CPW

Low Density Forest
High Density Forest
Urban
Surface Scattering
Rape, vineyard, olive-grove
### Overall Accuracy and Kappa Coefficient

- **Overall Accuracy**: \(\frac{13438}{16566} = 81\%\)
- **Kappa Coefficient**: 0.71

### Ground Truth (Percent)

<table>
<thead>
<tr>
<th>Class</th>
<th>HD Forest</th>
<th>LD Forest</th>
<th>R&amp;V&amp;OL</th>
<th>Surface Scattering</th>
<th>Urban</th>
</tr>
</thead>
<tbody>
<tr>
<td>HD forest</td>
<td>62.8</td>
<td>17.6</td>
<td>0</td>
<td>0.1</td>
<td>21.1</td>
</tr>
<tr>
<td>LD forest</td>
<td>28.5</td>
<td>57.0</td>
<td>4.1</td>
<td>0.4</td>
<td>2.7</td>
</tr>
<tr>
<td>R&amp;V&amp;OL</td>
<td>1.4</td>
<td>12.3</td>
<td>78.2</td>
<td>8.4</td>
<td>0.7</td>
</tr>
<tr>
<td>Surface Scattering</td>
<td>0</td>
<td>0</td>
<td>11.3</td>
<td>88.4</td>
<td>0</td>
</tr>
<tr>
<td>Urban</td>
<td>7.3</td>
<td>13.1</td>
<td>5.9</td>
<td>2.7</td>
<td>75.5</td>
</tr>
</tbody>
</table>
Overall Accuracy = (13887/16566) 84%  
Kappa Coefficient = 0.76

<table>
<thead>
<tr>
<th>Class</th>
<th>Ground Truth (Percent)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>LD forest</td>
</tr>
<tr>
<td>LD forest</td>
<td>54,5</td>
</tr>
<tr>
<td>HD forest</td>
<td>32,1</td>
</tr>
<tr>
<td>Urban</td>
<td>0</td>
</tr>
<tr>
<td>R&amp;V&amp;OL</td>
<td>13,4</td>
</tr>
<tr>
<td>Surface scattering</td>
<td>0</td>
</tr>
</tbody>
</table>
Conclusions

• The designed algorithm has allowed to assess the potentiality of SOM neural networks to classify polarimetric SAR data.

  The obtained overall accuracy is equal to 72% for seven classes and equal to 84% for five classes.

  The overall accuracy of NNs is 3% greater than CPW method.

  More flexibility for the number of the output classes