An Automatic Tool to Design CNN-UM Programs

Giovanni Pazienza

Departament d’electrònica, EALS
Universitat “Ramon Llull”, Barcelona
Outline

- Cellular Neural Network Universal Machine;
- CNN-UM programs;
- CNN-UM programs learning;
- Genetic programming;
- Results and discussion.
Cellular Neural Networks

- CNNs are a kind of Neural Networks with local connections only;
- Reduced number of weights;
- Analog computing, continuous time.
Ex. of CNN template – Hole filler

**Given**: static binary image P;

**Input**: \( U(t) = P \);

**Initial State**: \( X(0) = 1 \);

**Boundary Conditions**: \( [Y] = 0 \);

**Output**: \( Y(\infty) = \) Binary image representing P with holes filled.

\[
A = \begin{bmatrix}
  0 & 1 & 0 \\
  1 & 3 & 1 \\
  0 & 1 & 0 \\
\end{bmatrix} \quad B = \begin{bmatrix}
  0 & 0 & 0 \\
  0 & 4 & 0 \\
  0 & 0 & 0 \\
\end{bmatrix} \quad z = \begin{bmatrix}
  -1 \\
\end{bmatrix}
\]
CNN Universal Machine

- The CNN Universal Machine: an Analogic Array Computer (Roska, Chua), 1993;
- The CNN is Universal as the Turing machine (Chua, Roska, Venetianer), 1993;
CNN-UM implementation

- **ACE400**
  - Processor array/frame rate: 20x22, bin I/O, 50 000 fr/sec
  - 64x64, gray I/O, 1000 fr/sec

- **ACE4k**
  - 128x128, 50 000 fr/sec

- **ACE16k**
  - 32x32 (128x96), 10 000 fr/sec

- **XENON***
  - 176x144, 10 000 fr/sec

- **Q-Eye***
  - 1995-96
  - 1998-99
  - 2003
  - 2005/6
  - 2006/7

My acknowledgement to Prof. Roska for providing this slide.
CNN-UM programs

- A program for the CNN-UM is a sequence of templates applied to an input;

- CNN-UM programs can be represented by means of strings or trees.
**CNN-UM programs - Example**

- Out = Erosion(XOR(Thresh(Image), Hollow(Thresh(Image)))))
CNN-UM programs learning

- There is no systematic method to design CNN algorithms;
- Given a task defined by a pair of input/output images, we want to find the algorithm that performs it;

- We propose a genetic approach.
Genetic programming

- It is a genetic algorithm approach applied to a population of programs or complex structures;
- It allows to find among a population of computer programs the individual that best performs a given computational task;
- 5 basics steps: Set of terminals, Set of Functions, Fitness measure, GP parameters, Stop condition

<table>
<thead>
<tr>
<th>CNN-UM</th>
<th>CNN-UM programs</th>
<th>CNN-UM learning</th>
<th>GP</th>
<th>Results</th>
</tr>
</thead>
</table>
Steps for the GP system

- Terminals: input image(s);
- Functions: CNN templates;
- Fitness measure: Correlation or other metrics;
- GP Parameters: number of generations, size of the population, probabilities etc.;
- Stop condition: output_obtained=output_desired.
GP operators

- Crossover
- Mutation

<table>
<thead>
<tr>
<th>CNN-UM</th>
<th>CNN-UM programs</th>
<th>CNN-UM learning</th>
<th>GP</th>
<th>Results</th>
</tr>
</thead>
</table>

Results (I)

- Successful with elementary operations (XOR)
- Roughness measurement:
  - Preciado: 200 individuals, 396 generations
  - Our method: 50 individuals, 15 generations (<1%)
- Route number localization on public vehicles
  - Karacs: complexity 14, no full CNN solution
  - Our method: complexity 15, full CNN
Results (II)

- Train
- Test 1
- Test 2
Results (III)

- Train

- Test

| CNN-UM | CNN-UM programs | CNN-UM learning | GP | Results |
Bionic eyeglass (BE) project

- Led by professor Roska;
- No retinal prostheses soon available;
- The BE provides a wearable TeraOps visual computing power to advise visually impaired people in their daily life (home, street, office);
- Mobile phone camera + CNN processing and acoustic feedback;
- Problems: algorithms for the Bionic Eyeglass.
Conclusions

- Successful on several experiments: simple and complex cases;
- We always combine designer’s skills and automatic results of the machine;
- Two problems: representation and learning;
- Future works: more algorithms, better learning.
An Automatic Tool to Design CNN-UM Programs

Giovanni Pazienza

Departament d’electrònica, EALS
Universitat “Ramon Llull”, Barcelona