An Optimization Methodology for Neural Network Weights and Architectures

Teresa B. Ludermir

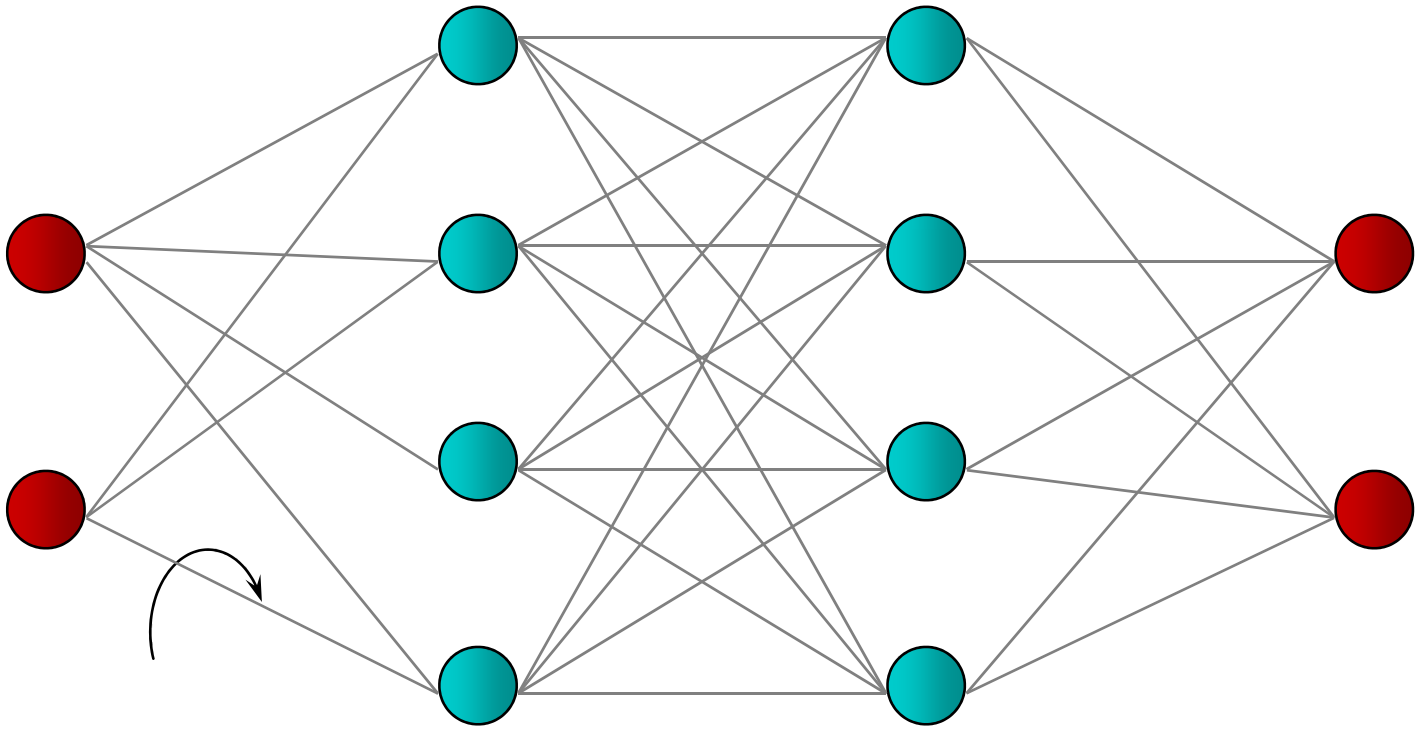
tbl@cin.ufpe.br

Outline

- **Motivation**
- **Simulated Annealing and Tabu Search**
- **Optimization Methodology**
- **Implementation Details**
- **Experiments and Results**
- **Final Remarks**

Motivation

- **Architecture design is crucial in MLP applications.**
- **Lack of connections can make the network incapable to solve a problem because there is few parameters to adjust.**
- **Too many connections can provoke overfitting.**
- **In general we try many different architectures.**
- **It is important to develop automatic processes for defining MLP architectures.**



Motivation

- There are several global optimization methods that can be used to deal with this problem.
- Ex.: genetic algorithms, simulated annealing and tabu search.
- Architecture design for MLP can be formulated as an optimization problem, where each solution represents an architecture.
- The cost measure can be a function of the training error and the network size.

Motivation

- **Most solutions represents only topological information, but not the weight values.**
- **Disadvantage: noise fitness evaluation**

- **Each solution has only the architectures but a network with a full set of weights must be used to calculate the training error for the cost function.**

- **Good option: optimizing neural network architectures and weights simultaneously.**
- **Each point in the search space is a fully specified ANN with complete weight information.**

- **Cost evaluation becomes more accurate.**

Motivation

- **Global optimization techniques are relatively inefficient in fine-tuned local search.**

- **Hybrid Training:**

- **Global technique for training the network followed by a local algorithm (Ex.: backpropagation) for the improvement of the generalization performance.**

Goal

- **Methodology for the simultaneous optimization of MLP network weights and architectures.**
- **Combines the advantages of simulated annealing and tabu search avoiding the limitations of the methods.**
- **Applies backpropagation as local search algorithm for improvement of the weights adjustments.**
- **Results from the application of the methodology to a real-world problems are presented and compared to those obtained by BP, SA and TS.**

Simulated Annealing

- **Method has the ability to escape from local minima due to the probability of accepting a new solution that increases the cost.**

- **This probability is regulated by a parameter called *temperature*, which is decreased during the optimization process.**

- **In many cases, the method may take a very long time to converge if the temperature reduction rule is too slow.**

- **However, a slow rule is often necessary, in order to allow an efficient exploration in the search space.**

Implementation Details of Simulated Annealing

– **Basic Structure of Simulated Annealing:**

- $s_0 \leftarrow$ initial solution in S
- For $i = 0$ to $I - 1$
  - Generate neighbor solution s'
  - If $f(s') \leq f(s_i)$
    - $s_{i+1} \leftarrow s'$
  - else
    - $s_{i+1} \leftarrow s'$ with probability $e^{-[f(s') - f(s_i)]/T_{i+1}}$
    - otherwise $s_{i+1} \leftarrow s_i$
- Return s_I

- S is the set of solutions, f is the real-valued cost function, I is the maximum number of epochs, and T_i is the temperature of epoch i.

Tabu search

- **Tabu search evaluates many new solutions in each iteration, instead of only one solution.**
- **The best solution (i.e., the one with lower cost) is always accepted as the current solution.**
- **This strategy makes tabu search faster than simulated annealing.**
- **It demands implementing a list containing a set of recently visited solutions (the tabu list), in order to avoid the acceptation of previously evaluated solutions.**
- **Using the tabu list for comparing new solutions to the prohibited (tabu) solutions increases the computational cost of tabu search when compared to simulated annealing.**

Implementation Details of Tabu Search

- **Basic Structure of Tabu Search:**
  - $s_0 \leftarrow$ initial solution in S
  - $s_{BSF} \leftarrow s_0$ (best solution so far)
  - Insert s_0 in the *Tabu List*
  - For $i = 0$ to $I - 1$
    - Generate a set V of neighbor solutions
    - Choose the best solution s' in V (i.e., $f(s') \leq f(s)$ for any s in V) which is not in the *Tabu List*
    - $s_{i+1} \leftarrow s'$
    - Insert s_{i+1} in the *Tabu List*
    - If $f(s_{i+1}) \leq f(s_{BSF})$
      - $s_{BSF} \leftarrow s_{i+1}$
  - Return s_{BSF}

  - The *Tabu List* stores the K most recently visited solutions.

Optimization Methodology

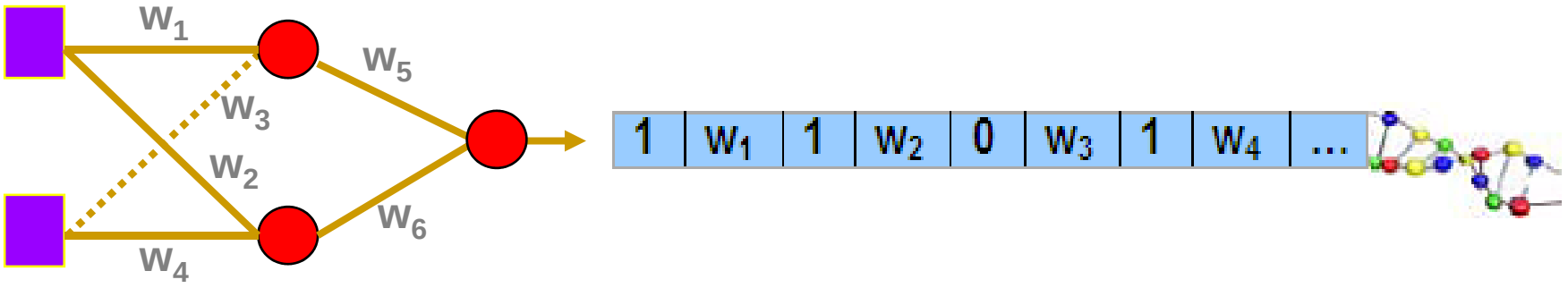
- **A set of new solutions is generated at each iteration, and the best one is selected according to the cost function, as performed by tabu search**
- **However, it is possible to accept a new solution that increases the cost since this decision is guided by a probability distribution, which is the same used by simulated annealing.**
- **During the execution of the methodology, the topology and the weights are optimized, and the best solution found so far (s_{BSF}) is stored.**
- **At the end of this process, the MLP architecture contained in s_{BSF} is kept constant, and the weights are taken as the initial ones for training with the backpropagation algorithm, in order to perform a fine-tuned local search.**

1. $s_0 \leftarrow$ initial solution
2. $T_0 \leftarrow$ initial temperature
3. Update s_{BSF} with s_0 (best solution found so far)
4. For $i = 0$ to $I_{max} - 1$
5. If $i + 1$ is not a multiple of I_T
6. $T_{i+1} \leftarrow T_i$
7. else
8. $T_{i+1} \leftarrow$ new temperature
9. If stopping criteria is satisfied
10. Stop execution
11. Generate a set of K new solutions from s_i
12. Choose the best solution s' from the set
13. If $f(s') < f(s_i)$
14. $s_{i+1} \leftarrow s'$
15. else
16. $s_{i+1} \leftarrow s'$ with probability $e^{-[f(s')-f(s_i)]/T_{i+1}}$
17. Update s_{BSF} (if $f(s_{i+1}) < f(s_{BSF})$)
18. Keep the topology contained in s_{BSF} constant and use the weights as initial ones for training with the backpropagation algorithm.

Implementation Details

- **Representation of Solutions**
  - Each MLP is specified by an array of connections.
  - Each conection is specified by two parameters:
    - the connectivity bit:
      - equal to 1 if the connection exists,
      - and 0 otherwise
    - and the connection weight (a real number).

  - Maximal network structure:
    - One-hidden-layer MLP:
      - N_1 input nodes
      - N_2 hidden nodes
      - N_3 output nodes
      - All possible feedforward connections between adjacent layers and no connection between non-adjacent layers ($N_1 N_2 + N_2 N_3$)


W1
W5
W3
W2
W6
W4
1
W1
1
W2
0
W3
1
W4
...

Implementation Details

- **Cost Function**
  - The cost function is the mean of two parameters:
    - the classification error for the training set (percentage of incorrectly classified training patterns)
    - and the percentage of connections used by the network.
  - The algorithm tries to minimize both network performance and complexity.
- **Generation Mechanism for the Neighbors**
  - The mechanism acts as follows:
    - the connectivity bits for the current solution are changed according to a given probability (in this work, 20%),
    - and a random number from an uniform distribution between –1.0 and +1.0 is added to each connection weight.
  - The mechanism changes both topology and connection weights to produce a new neighbor solution.

Implementation Details

- **Cooling Schedule**
  - **Geometric cooling rule: the new temperature is equal to the current temperature multiplied by a temperature factor.**
    - **The initial temperature is set to 1,**
    - **and the temperature factor is set to 0.9.**
    - **Temperature is decreased at each 10 epochs,**
    - **and the maximum number of epochs allowed is 1 000.**
  - **The algorithm stops if:**
    - **the GL_5 criterion defined in *Proben1* is met (based on the classification error for the validation set),**
    - **or the maximum number of 1 000 epochs is achieved.**
  - **The classification error for the validation set is measured after every tenth epoch.**

Problem Description

- **Four classification problems:**
  - **the odor recognition problem in artificial noses**
    - **the aim is to classify odors from three different vintages (years 1995, 1996 and 1997) of the same wine (Almadén, Brazil).**
  - **Diagnose diabetes of Pima Indians**
  - **Fisher's Iris data set**
  - **Thyroid data set and**
- **one prediction problem:**
  - **Mackey-Glass time series**

Problem Description

- **Data partitioning was done in the following way:**
  - **the training set had 50% of the patterns from each class,**
  - **the validation set had 25% from each class,**
  - **and the test set had 25% from each class.**

Results for MPL

Number of hidden units \ Data set	Artificial Nose	Iris	Thyroid	Diabetes	Mackey Glass
	Mean test set classification error (%)				SEP
02	33.6296	19.0598	10.2000	-	4.2146
03	-	18.2051	-	-	-
04	17.8123	7.9487	9.2704	27.8819	1.4357
05	-	6.8376	-	-	-
06	14.1185	10.6838	-	30.2951	1.8273
07	-	8.9744	-	-	-
08	11.1136	-	13.1519	28.4201	1.9045
10	6.3086	-	7.3800	27.0833	1.5804
12	8.8667	-	7.3804	27.3264	2.3831
14	11.9704	-	7.4824	28.4549	2.7860
16	-	-	10.2537	-	-

Results for the optimization approaches

- **Simulated annealing, tabu search, and the proposed methodology were implemented.**

- **Artificial Nose data set contains six input units, ten hidden units and three output units ($N_1 = 6$, $N_2 = 10$ and $N_3 = 3$, the maximum number of connections (N_{max}) is equal to 90). In Iris data set the maximal topology contains $N_1 = 4$, $N_2 = 5$, $N_3 = 3$ and $N_{max} = 35$. For the Thyroid data set the maximal topology contains $N_1 = 21$, $N_2 = 10$, $N_3 = 3$ and $N_{max} = 240$. In Diabetes data set the maximal topology contains $N_1 = 8$, $N_2 = 10$, $N_3 = 2$ and $N_{max} = 100$. In Mackey Glass experiments the maximal topology contains $N_1 = 4$, $N_2 = 4$, $N_3 = 1$ and $N_{max} = 20$.**

Data set \ Method		SA	TS	Methodology
Artificial Nose	Class. (%)	3.3689	3.2015	1.4244
	Input	5.9400	5.9667	5.8800
	Hidden	7.8067	8.0667	7.0567
	Connec.	35.3700	36.6333	29.1033
Iris	Class. (%)	12.6496	12.4786	4.6154
	Input	2.8500	2.8767	2.7100
	Hidden	2.7567	3.4867	2.6567
	Connec.	8.3433	8.3000	7.7633
Thyroid	Class. (%)	7.3813	7.3406	7.3322
	Input	20.7700	20.7700	20.3700
	Hidden	7.2267	7.4667	6.3900
	Connec.	83.7300	86.1400	71.5467
Diabetes	Class. (%)	27.1562	27.4045	25.8767
	Input	7.7600	7.7800	7.5633
	Hidden	5.2700	5.3700	4.5300
	Connec.	30.3833	30.8167	25.5067
Mackey Glass	SEP Test	2.0172	0.8670	0.6847
	Input	3.6167	3.7967	3.4567
	Hidden	1.9000	2.2700	1.8933
	Connec.	9.6300	12.0700	8.5667

Results for the optimization approaches

- **For the proposed methodology, the t test has concluded that the classification error was statistically lower, to the Iris and Artificial Nose data sets, and statistically equivalent to the obtained by the other methods in the remainder data sets.**

- **The mean number of connections for the proposed methodology was lower than all remaining approaches, for all data sets.**

Conclusions

- **Simulated Annealing and Tabu Search were used successfully for simultaneous optimization of topology and weights. Both techniques were able to find MLPs with low complexity and high generalization performance for the odor recognition problem.**

- **The proposed methodology can be used successfully for simultaneous optimization of MLP network topology and weights.**

- **The proposed methodology was originally not designed to deal with different number of hidden layers but it does work with different number of hidden layers.**

- **Others hybrid algorithms have been proposed using AG, Ant Colony and Swarm Optimization.**