Using Genetic Algorithms for an Optical Thin-Film Learning Model

Xiaodong Li and Martin Purvis
Computer and Information Science
University of Otago
Dunedin, New Zealand
xiaodong@otago.ac.nz, purvis@otago.ac.nz

Abstract

A novel connectionist architecture based on an optical thin-film multilayer model (OTFM) is described. The architecture is explored as an alternative to the widely used neuron-inspired models, with the thin-film thicknesses serving as adjustable 'weights' for the computation. The use of genetic algorithms for training the thin-film model, along with experimental results on the parity problem and the iris data classification are presented.

1 Introduction

The genetic algorithms were first developed by Holland in 1970 to mimic the processes of natural evolution and have been shown since to be useful in a variety of search problems[1]. In this paper we explore the use of genetic algorithms as the key optimization procedure in the design of an alternative connectionist learning model inspired by the technology of optical thin-film multilayers and first reported in [2,3]. The proposed thin-film model has shown the capability of carrying out computational tasks that are often handled by conventional neural network models. We initially describe the architecture of the model and how the genetic algorithm is applied to it, then we provide two examples to illustrate its ability to perform certain computational tasks.

2 The Optical Thin-Film Model

When a light beam is incident on a single thin-film, it undergoes multiple reflections and refractions, the various component beams of which interfere with each other to produce an overall reflectance and transmittance governed by the refractive index and thickness of the film and the wavelength of the incident light. When several thin-films of differing materials are deposited on top of each other to form a thin-film multilayer (Figure 1), there are multiple reflections and transmissions at each boundary, and it is possible to derive a recursive algebraic expression for the overall reflection coefficient of the resulting multiple beam interference, which we show below.

Suppose an additional thin-film layer with complex refractive index $\tilde{n}_N = n_N + ik_N$ and thickness d_N is brought up to the existing multilayer structure shown in Figure 1. (The imaginary component of the complex refractive index accounts for any absorption of the thin-film material.) It can be shown [4] that the combined complex reflection coefficient, r_N, of the resulting N layer system comprising the structure and the added layer is

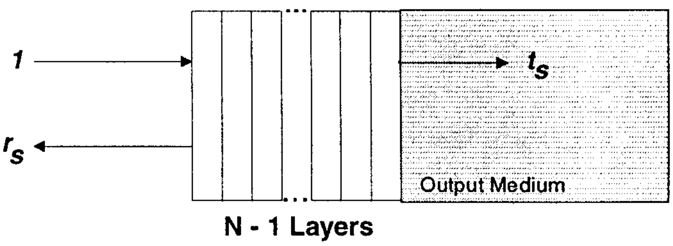

Figure 1. N - 1 thin-films deposited on a substrate.

$$r_N = \frac{-f_N(1 - \varepsilon_N^2) + (\varepsilon_N^2 - f_N^2)r_{N-1}}{(1 - f_N^2\varepsilon_N^2) + f_N(1 - \varepsilon_N^2)r_{N-1}} \tag{1}$$

where

$$f_N = \frac{(\tilde{n}_N - 1)}{(\tilde{n}_N + 1)} \quad , \qquad \varepsilon_N = \exp(i2\pi\tilde{n}_N d_N/\lambda)$$

The optical reflectance, which is what is typically measured, is obtained by taking the square of the absolute magnitude of the reflection coefficient in (1). Almost any desired spectral characteristic of the reflection can be obtained by choosing a sufficient number of layers with appropriate values of n and d for each thin-film layer.

For our computational purposes we view the reflection coefficient as the output. Input is encoded into the thin-film system by adding small, scaled offsets to the refractive indices of given layers. Typically, the *i-th* input value is scaled and added as an offset to the base index n_{Bi} of the *i-th* thin-film layer of a multilayer structure so that $n_i = n_{Bi} + n_{\Delta i}$. Training involves adjusting the individual layer thicknesses and then calculating the multilayer reflection coefficient r (or, alternatively, the reflectance R) for each optical wavelength for which output is specified. The resulting spectral output $r(\lambda)$ (or $R(\lambda)$) is compared to a target output $r_T(\lambda)$ (or $R_T(\lambda)$) for each wavelength, and a merit function M is used to determine the error for the given configuration:

$$M = \sum_{\lambda} |r(\lambda) - r_T(\lambda)|^2 \quad , \quad or \quad M = \sum_{\lambda} |R(\lambda) - R_T(\lambda)|^2 \qquad \textbf{(2)}$$

where $R = |r|^2$. Training is continued until a configuration of layer thicknesses is found that produces a satisfactory value of the merit function.

3 GA for Learning

Calculation of the reflection coefficients (as simulated on a computer) is computationally expensive for thin-film structures with many-layers. Instead of exhaustive search through the thin-film thicknesses space, using genetic algorithms may provide a more efficient way of training. Generally two components in genetic algorithms are problem dependent: the problem encoding and the evaluation function. Consider the thin-film model where a set of optimal thicknesses needs to be found to minimize the merit function Eq.(2). These thicknesses are represented by a binary string and the resulting reflection coefficient at each wavelength is evaluated by the merit functions Eq.(2).

The GA search for an optimal set of layer thicknesses can be defined by the following procedure:

Start: generate initial population of binary bits which represent sets of thickness values. For each set of thicknesses, the system evaluates its merit.

Loop through following steps:

1) Select parents whose genetic make-up contributes to fitter offspring.
2) Produce offspring, using suitable genetic operators (reproduction, crossover and mutation).
3) Evaluate the performance of these offspring.
4) Replace certain parents with new offspring.

The process is repeated until the population has converged or a number of iterations has been reached.

4 Experimental Results

Experiments were conducted using code based on the SGA presented by Goldberg[1,5]. Two examples, 16 four-bit parity and iris data classification, which have been used to demonstrate the properties of conventional feed-forward neural network models, are used for the thin-film model training employing GA search.

4.1 16 four-bit Parity

In this parity problem, the target output is reflectance value of 0.0 if the parity of the four bits of binary numbers(0 or 1) is even, and the target output is 1.0 if the parity is odd. The model was specified to have 25 layers with values of the base refractive index n_{Bi} for each layer alternating between 1.2 for the odd-number layers and 1.6 for the even-number layers. Each input is a 4-bit binary number and encoded as small values to be added to n_{B1}, n_{B2}, n_{B3}, and n_{B4} respectively.

With population size of 300, a chromosome length of 160 as the binary string which is interpreted as 25 thickness values, crossover probability 0.6 and mutation probability 0.04, we obtained a training result after 16 generations, as shown in Table 1. With an acceptance threshold set to 0.5, the resulting reflectance matches the targets.

Table 1. Training result for solving 4-bit parity problem.

Inputs				Target Reflectance	Output Reflectance
0	0	0	0	0.0	0.037
0	0	0	1	1.0	0.750
0	0	1	0	1.0	0.723
0	0	1	1	0.0	0.005
0	1	0	0	1.0	0.677
0	1	0	1	0.0	0.181
0	1	1	0	0.0	0.322
0	1	1	1	1.0	0.693
1	0	0	0	1.0	0.586
1	0	0	1	0.0	0.091
1	0	1	0	0.0	0.082
1	0	1	1	1.0	0.629
1	1	0	0	0.0	0.026
1	1	0	1	1.0	0.689
1	1	1	0	1.0	0.648
1	1	1	1	0.0	0.049

4.2 Iris data classification

The iris data set [6] has been frequently used as an example for discriminant analysis of real data. The data set was collected for three species of iris (setosa, versicolor and viginica) and comprised four measurements (petal length, petal width, sepal length, and sepal width) for 50 samples of each species. One of the species is thought to have arisen as a hybrid species and displays a mixture of features of the other two, making discrimination among the species by means of the measured properties somewhat difficult [7]. One class setosa is linearly separable from the other two, while the versicolor and viginica are not linearly separable from each other.

For each iris data example used, e.g. {5.1, 3.5, 1.4, 0.2}, it is necessary to scale the values appropriately for encoding into the thin-film system. A thin-film stack of only four layers was selected with values of the base refractive index for each n_{Bi} alternating between 1.2 and 2.5. The training was carried out for the optical wavelength values: 5.0, 5.3, and 5.6 μm, and the target reflectance coefficient values used are shown in Table 2.

Table 2. Target reflection coefficients

λ	Class 1 r_{RE}	Class 1 r_{IM}	Class 2 r_{RE}	Class 2 r_{IM}	Class 3 r_{RE}	Class 3 r_{IM}
5.0	0.2	0.2	-0.5	0.1	0.1	-0.5
5.3	0.3	0.3	-0.4	0.3	0.3	-0.4
5.6	0.4	0.4	-0.3	0.4	0.4	-0.3

The thin-film stack was then trained, using 120 examples (40 randomly chosen from each of the three classes). Chromosome length 60 was chosen since only 4 layer thicknessses are needed to map into a binary string. After 6 generations, with initial population size 300, crossover probability 0.6 and mutation probability 0.033, the training had the effect of moving the response reflection coefficient points as close as possible to the 3 target points for each optical wavelength value in an effort to separate the collection of output reflection coefficient values as much as possible. Although data from setosa were

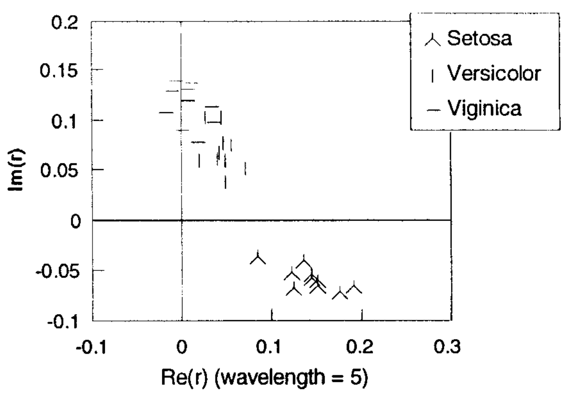

Figure 2. Test results on 30 iris data samples, λ = 5.0μm.

completely separated from the other two classes, there were a couple of points from versicolor and viginica that always overlapped into each other's regions.

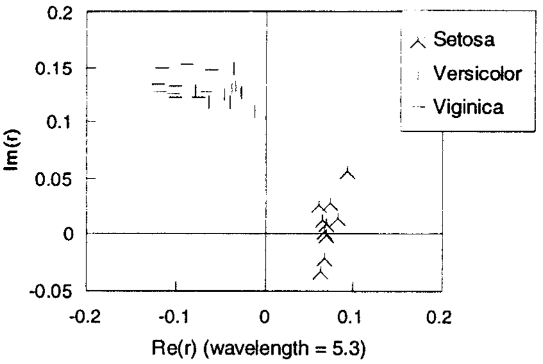

Figure 3. Test results on 30 iris samples, λ = 5.3μm.

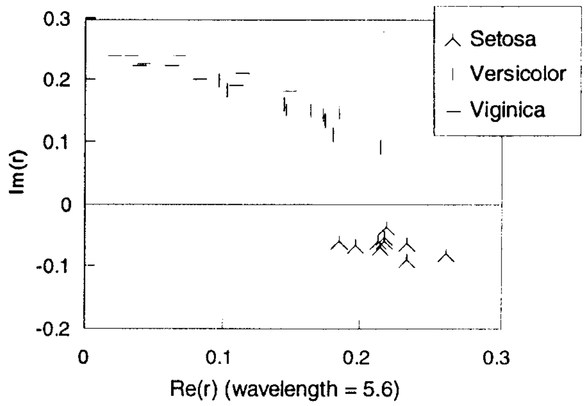

Figure 4. Test results on 30 iris data samples, λ = 5.6μm.

After training, a "winner-take-all" approach [8] was used to set points in the output range to be used for discrimination of samples (during test runs) into the three species. This was accomplished by finding the average values of all the output points at each optical wavelength and for each training class. Thus each of the three species will have a set of characteristic values for output reflection coefficients.

When the test example is entered into the system, it is a matter of calculating the output reflection coefficient for the test example (at each wavelength) and then using the merit function evaluation of Eq.(2) to determine to which species class the example output is closest. Output for thirty novel examples is shown in Figures 2, 3, and 4. Of the thirty examples only 3 were misclassified by the thin-film system. When the system was given all 150 samples, 8 were classified incorrectly.

5 Conclusions

Using genetic algorithms has been demonstrated as an efficient training method for the proposed optical thin-film model (OTFM), which has shown the learning capability that are typical of conventional neural network architectures. Results of other experiments using different search algorithms can be referred to articles [2,3], and a more comprehensive description of the model will be published in the near future.

References

[1] Goldberg, D.E., *Genetic algorithms in search, optimization, and machine learning.* Reading, MA: Addison-Wesley, 1989.

[2] Purvis, M.K. and Xiaodong, L., "*Connectionist Computations Based on an Optical Thin-Film Model*", Proceedings of the First New Zealand International Two Stream Conference on Artificial Neural Networks and Expert Systems, IEEE Computer Society Press, Los Alamitos, California,pp.130-133, 1993.

[3] Purvis, M.K. and Li, X., "*Connectionist Learning Using Optical Thin-Film Model*", Proceedings of the 2nd New Zealand International Two-Stream Conference on Artificial Neural Networks and Expert Systems, IEEE Computer Society Press, Los Alamitos, California,pp.63-66, 1995.

[4] Case, W. E., "New Synthesis Method for Optical Thin-film Coatings", *Applied Optics*, 22:24, pp. 4111-4117, 1983.

[5] Smith, R.E., Goldberg, D.E. and Earickson J.A., "SGA-C: A C-language Implementation of a Simple Genetic Algorithm", TCGA Report No. 91002, The University of Alabama, 1994.

[6] Anderson, E., "The Irises of the Gaspe Peninsula", *Bulletin of the American Iris Society,* vol 59, pp. 2-5.

[7] Ripley, B. D., "Statistical Aspects of Neural Networks" in *Networks and Chaos -- Statistical and Probabilistic Aspects*, Chapman and Hall, London, pp. 40-123, 1993.

[8] Hart, A., "Using Neural Networks for Classification Tasks -- Some Experiments on Datasets and Practical Advice", *J. Opl. Res. Soc.*, vol. 43, pp. 215-226, 1992.