Helder Filipe Rebelo de Pinho

Family Name: Pinho

First Name: Helder

University/Department: Instituto Engenharia do Porto

Email: i970326@dei.isep.ipp.pt

Do you like to present a contributed talk: yes

Title of your contribution: Title of your contribution: Analysis of RBS data by Artificial Neural Networks: a systematic approach

Name and email of your advisor: Prof. Armando Vieira ::: armandoviera@mail.telepac.pt

Do you like to apply to a grant (Yes/No)? yes

Method of payment: not confirm

Analysis of RBS data by Artificial Neural Networks: a systematic approach

Armando Vieira and Helder Pinho, Physics Dept, ISEP, R. S. Tomé, 4200 Porto, Portugal.

Rutherford backscattering (RBS) is a non-destructive, fully quantitative, technique for accurately determining the compositional depth profile of thin films. The inverse RBS problem, which is to determine from the data the corresponding sample structure, is however in general ill-posed. Skilled analysts use their knowledge and experience to recognize recurring features in the data and relate them to features in the sample structure. This is then followed by a detailed quantitative analysis. Artificial Neural Network (ANN) have already been sucessufuly applied to data analysis of implantations of Ge in Si, and Er in saphire among others. In this work we show the first results of using neural networks to a more general problem, namely implantations of *any element* in *any substract* under *any experimental conditions*. This is a very hard problem for a ANN where we used housands of constructed spectra of samples for which the structure is known. We used a efficient algorithm to extract features from the 512 channel spectra, thus reducing drastically the dimensionality of the data.

The ANN learns how to interpret the spectrum of a given sample, without any knowledge of the physics involved. The ANN was then applied to experimental data from samples of unknown structure. The quantitative results obtained were compared with those given by traditional analysis methods, and are excellent. The major advantage of ANNs over those other methods is that, after the time-consuming training phase, the analysis is instantaneous, which opens the door to automated on-line data analysis. Furthermore, the ANN was able to distinguish two different classes of data which are experimentally difficult to analyze. This opens the door to automated on-line optimization of the experimental conditions.

State of work

I have used a Multilayer Feedforward Preceptron trained by backpropagation with a training set consisting of 4000 simulated spectra obtained with the NDF (Nuno Data Furnace) code. Each spectrum contains the number of recoiled particles that reached the counter within each energy range. The objective is to determine the depth and dose of the implanted substance with a good enough accuracy.

Initially several tests were performed in order to determine the best network architecture.

The best by number of examples:

Architecture	Training error	Test error	Number of Data	nrejTEST
13 30 10 2	0.031471	0.033695	2000	395
13 30 20 2	0.031245	0.037008	3000	622
13 40 20 2	0.032005	0.032537	4000	795
13 30 20 2	0.029814	0.032206	5000	1027

We pretend 20 percentage reject examples.
More details in annex file netchooseII.txt

The decision:

13 30 20 2 NET
3000 examples
0.031245 TrainError / TrainE.PredictingMean
0.037008 TestError / TestErrorPredictingMean
622 reject on train
72 reject on test
timecorrect:::0.02, this means the difference between the avg1 and avg2 that determines the call of function correct.
0.587366 (TrainError / TrainE.PredictingMean) correct1 when we call the correct
0.128657 (TrainError / TrainE.PredictingMean) correct2 when we call the correct
0.051306 (TrainError / TrainE.PredictingMean) correct3 when we call the correct
correctceiling::2.75 this is the hard we put in determination of god examples.

The results between the nets are very close, all chooses are god.
I prefer this because 3000 examples make the net much more light than 5000.

We rejected some bad examples (20%, without losing generalising), they showed some features that give a height negative performance and in correct function we put them out of train with considerable improvements.

In nests tables you can see the deferent's results between one to tree corrects invocations.
(1 calling function, 2 and 3 calling function, by rows)

In 2000

Architecture	Training error	Test error	Number of Data	nrejTEST
13 30 10 2	0.088286	0.084612	2000	157
13 30 10 2	0.045499	0.045148	2000	289
13 30 10 2	0.031471	0.033695	2000	395

In 3000

Architecture	Training error	Test error	Number of Data	nrejTEST
13 30 20 2	0.111099	0.154219	3000	248
13 30 20 2	0.044430	0.059427	3000	423
13 30 20 2	0.031245	0.037008	3000	622

In 4000

Architecture	Training error	Test error	Number of Data	nrejTEST
13 40 20 2	0.096979	0.133867	4000	315
13 40 20 2	0.056084	0.061786	4000	574
13 40 20 2	0.032005	0.032537	4000	795

In 5000

Architecture	Training error	Test error	Number of Data	nrejTEST
13 30 20 2	0.098537	0.098903	5000	403
13 30 20 2	0.049759	0.053663	5000	741
13 30 20 2	0.029814	0.032206	5000	1027

More details in annex file corr1vs2vs3.txt

Calling the correct function in several times we get best results, in the first calling function the net assimilate the rejections and in the second time will reject with more accuracy.

In future

In this moment we are testing the results and confirming the results obtained which will be compare with those given by traditional analysis methods.
After this confirmation the experimental work is finish.