MULTIPLE MULTIVARIATE REGRESSION AND GLOBAL SEQUENCE OPTIMIZATION: AN APPLICATION TO LARGE SCALE MODELS OF RADIATION INTENSITY.

To Appear in: *Signal Processing* , vol. 64 issue 3.

H. Zaragoza, P. Gallinari,
LIP6, *Université Pierre et Marie Curie*,
4, place Jussieu F-75252 PARIS cedex 05 (France).
{Hugo.Zaragoza, Patrick.Gallinari}@lip6.fr

R. Curtelin, F. Leglaye
SNECMA - Villaroche.
7750 Moissy Cramayel (France).

ABSTRACT

We investigate the strengths and weaknesses of several neural network architectures for a large-scale thermodynamical application in which sequences of measurements from gas columns must be integrated to construct the columns‘ spectral radiation intensity profiles. This is a problem of interest for the aeronautical industry. The approaches proposed for its solution can be applied to a wide range of signal problems. Physical models often make use of a number of fitted functions as a simplified parametric base to approximate a high-dimensional nonlinear (and usually computationally intractable) function. Realistically models of radiation contain thousands of fitted functions. The use of Neural Networks in applications of this scale are rare, and most effective conjunctions techniques rely on cross-validation methods or involve other heavy computational overhead that are impracticable when a very large number of models need to be trained. We have employed here two different approaches: multiple multivariate regression, and global sequence minimization. The first approach shows that the integration of several nonlinear regression models into a single neural network may improve both generalization performance and speed of computation. For the former we propose a method of optimization by which we specialize our models globally, on typical sequences of input signals. We show how this does not degrade the over-all accuracy but, rather, allows us to specialize our models.

1 INTRODUCTION

The motivation behind the research presented in this paper was the development of a system for the fast and accurate computation of *intensity of radiation* profiles, observed through a column of absorbing gas. This is a problem of interest for the aeronautical industry, and is applied to a wide range of domains such as infrared monitoring or turboreactor design. There exists a method for the precise resolution of this thermodynamical problem [1], but unfortunately it is too computationally intensive, and cannot be used for interactive simulations or real-time applications. A number of approximated physical models exist and have been implemented to compute relatively accurate profiles in reasonable computational time [1][2].

To implement such approximated models, however, it is necessary to determine a large number of application-dependant parameters, to which models are extremely sensible. This is usually an expensive and lengthy task, and demands a great experience on the part of model constructors. Part of the difficulty is the nonlinear nature of the problem: physicists must carefully transform and partition data so that local linear regression models work.

In this implementation we utilize nonlinear multi-layer perceptrons (MLPs) to build accurate models which are constructed automatically, without the lengthy hand-tuning of traditional models. We will see that, while the automatically constructed MLP models cannot compete in accuracy with hand-crafted local linear models, they can be made sufficiently accurate and offer an alternative for applications in which the cost of hand-crafting them is prohibitive. This is not possible with traditional linear models. Because of the large number of regressors needed, this is in fact not possible for traditional neural network techniques either.

In the process of developing the thermodynamical application, we have obtained several results which we believe will be of general interest to developers of neural network applications. For example, a typical problem of nonlinear regression is their large number of parameters needed, in comparison to linear models. We show that it is possible to construct MLP architectures with equal or even less parameters than the original linear models. Specifically, we show how the integration of several correlated regressors into a single neural network may bring a drastic reduction in the number of network parameters while improving its generalization performance and reducing the computational costs of its utilization and development.

In the standard model, a large number of linear regressors (several thousands, in our application) are constructed to approximate independently small regions of the signal space. These regions correspond to sequences of points (segment descriptions) in typical gas columns. Regressors are then combined into a complex nonlinear model. In the second part of our work we propose a method of global sequence optimization, which goes beyond the standard thermodynamical model. We set out to train our models globally, taking into account the entire sequence of points in a column, with respect to the researched profile. This allows us to improve the generalization performance on a family of input sequences, biasing or «specializing» the model in a natural and easy to implement manner.

In Section 2 we give a brief overview of the thermodynamical model in which the application is based. We will see that the main problem that needs to be solved is fitting a large number of functions which form a parametric base for the global solution of the problem, in order to integrate a function of these parameters over the signal sequence. In Section 3 we discuss how ordinary least square regression and neural networks (NNs) may be used to fit these functions. We will see that a straight-forward substitution of linear models by NNs brings a drastic improvement in accuracy, but at a very high computational cost. In Section 4 we discuss a method to reduce this cost while improving the generalization capacity of the NN model. This method consists in using a single NN to approximate simultaneously a family of parameter functions. Finally, in Section 5, we propose a two-stage optimization procedure which directly optimizes the global sequence instead of fitting independently the different parameter functions. This improves generalization and has the special interest of providing a subtle and easily implementable way to specialize the models after normal training is completed.

2 THE PHYSICAL PROBLEM

Below, we briefly introduce the physical problem, without going into details; these are not important for the comprehension of this work and are beyond the scope of the paper. We have provided a more thorough description of the physical model in the Appendix. For a detailed description the reader may refer to [1] and [2].

Spectral radiation profiles offer a concise representation of a radiation source over a certain region of the spectrum. The radiation models allow to compute these profiles in any point of the space, given the characteristics (temperature, pressure, molecular composition, etc.) of the media between the point of observation and the source. Temperature and pressure gradients between these two points, as well as the molecular composition of the medium, will cause certain spectral regions to be filtered while others will be amplified. Profiles can be furthermore integrated to compute the total radiation intensity at a point in the space. Entire radiation images can be constructed in this manner. These images will inform us of the distribution of radiation intensity around the source.

Spectral radiation profiles of homogeneous (constant molecular composition) and isothermal (constant temperature and pressure) media can be computed in a straight forward manner. This is done by integration of the *spectral transmissivity* gradient of the column. For this application, however, we are concerned with atmospheric medium with strong temperature and pressure gradients (such situations are common, for example, near turboreactors). We are interested in computing the spectral radiation profile of a specific point at some distance of a known energy source.

The line between the sources and the point of interest will cross different pressure and temperature zones with varying molecular composition. We will call this line a «gas column». Gas columns are in fact discretised into a sequence of column segments. Temperature, pressure and molecular composition are constant within segments. A gas column is therefore formalized into a discrete sequence of measurements (Figure 1). The spectral intensity of radiation (I) of such a column is then the integral of the inter-segment transmissivity gradients (τ_{s-1}-τ_s), weighted by the vacuum intensity of each segment I_s^o:

$$I = \sum_{s = 1, S} (\tau_{s-1} - \tau_i) I_s^o + \text{const} \tag{1}$$

where I is the spectral intensity of radiation, *s* is a segment index (from the source, *s*=1, to the observer, *s*=S), and τ_s is the segment‘s transmissivity.

The difficult task is then the determination of the spectral transmissivity τ_s of each segment. These are influenced by the characteristics of the column from the source to the segment‘s location. There exists a very precise model to determine spectral transmissivity, named the Line-by-Line model (LBL), but this model is too computationally intensive for real-world applications. A number of simplified models have been developed and are of common use [1]. The correlated fictitious gases method (CKFG-N) is the most adequate of these for nonhomogeneous gases and has already been applied with success in a number of applications [2]. It is this method that we have set out to improve with the use of neural networks.

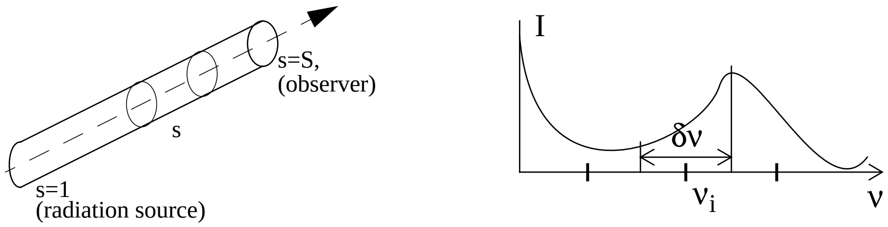

FIGURE 1. Column and spectrum discretisation. On the left we see a gas column discretised in segments, indexed by *s*. All measurements are considered constant within each segment. On the right we see the spectrum discretised into spectrum windows of width δν and index *i*. All computations for each spectral window are carried out independently.

The CKFG-N model approximates the spectral transmissivity of a segment (τ_s) with a high dimensional parametric function of the form:

$$\tau_s = \prod_{j = 1, N} \left[\sum_{m = 1, M} A_m \exp\left(\sum_{i = 1, s} f_{jm}(t_i, p_i)\, l_i c_i \right) \right] \tag{2}$$

where l_i and c_i denote length and molecular composition of segment *i*, *s* is the segment for which transmissivity is computed, A_m are constants of integration, j is an index on different parameter families and f_{jm} are the *parameter functions* (a justification of this form of parametrisation and a more detailed explanation of the different terms is presented in the Appendix). Note that for the computation of the transmissivity at a given segment *s*, all the sequences of measurements up to this segment need to be taken into account. This means that $N \cdot M \cdot s$ parameters need to be estimated to compute the transmissivity at one segment. Since equation (1) integrates over the whole sequence (*s*=1..S), we see that (1) and (2) construct a recursive computation over all subsets of the signal starting at the energy source.

Except for the parameter functions, all terms in (2) are constant and known for a given gas column. Parameter functions are two-dimensional functions (their value for a given segment depends only on the segment’s temperature and pressure) for which there is no analytical expression; they must be estimated from LBL data. Although parameter functions are low-dimensional, they are strongly non-linear, and the data available for their construction is scarce; it is therefore difficult to build good regressors. The accuracy of the parameter functions is the only determinant factor of the final accuracy of the CKFG-N model (all other parameters in (1) and (2) being fixed). Obtaining good regressors is thus critical.

The number of parameter functions that need to be determined, as well as the number of times these need to be evaluated, impose practical constrains on the kind of techniques applicable to the regression. In a typical application such as ours, the values of N and M in equation (2) are 5 and 7 respectively. This means that 35 parameter functions f_{jm} need to be fitted for each spectral point. Typically about 100 spectral points are needed to characterize the spectral intensity of a single column, which means that a total of 3500 parameter functions need to be fitted for the complete CKFG-N model.

This large number of functions constrains indeed the possibilities of experimentation and forbids the use of training parameters determined by cross-validation, repeated experimentation, etc. A final constraint is introduced by the necessity of fast computations. Hundreds of sequences need to be processed, a typical sequence containing several hundred measurements. This means that several millions of parameter function values are needed for a single run. This imposes a practical constraint on the speed of the computation of these values, and therefore on the complexity (i.e. the number of parameters) of the regressors.

3 INDEPENDENT REGRESSION FRAMEWORK

We dispose of a set of construction points obtained by the LBL model from which the transmissivity parameter functions are to be fitted. Formally, we dispose of a set of points $(\mathbf{x}_s , \mathbf{y}_s)_{s=1..S}$ which assign, to each segment description $\mathbf{x}_s=(x_1^s, x_2^s, . . . , x_p^s)$, the values of the parameter functions at the segment, $\mathbf{y}_s=(y_1^s, y_2^s, . . . , y_N^s)$. NN terminology denotes $\mathbf{x}$ and $\mathbf{y}$ as the input and desired patterns respectively, while in statistics they are called predictor and response measurement vectors.

A simple way to construct the parameter functions would be to use ordinary least squares regression. A separate regression would be performed for each parameter function as follows:

$$y_j(\mathbf{x}) = \sum_{i=1}^{p} x_i \cdot w_{ij} \tag{3}$$

The nonlinearity of the dependence on temperature and pressure may be introduced in the regression by directly providing nonlinear terms as variables. Physical knowledge allows us to choose the appropriate nonlinear transformations of the variables (in our case, a polynomial development on the known measurements) [2].

Similarly, a standard neural network (NN) approach would be to build a small network to predict each parameter function separately. We have used multi-layer perceptrons (MLPs) in order to approximate these functions nonlinearly. As a first approach, we have replaced each of the linear regressors (thirty-five per spectral point) by an MLP with a sigmoidal hidden layer and a single linear output unit:

$$y_j(\mathbf{x}) = \sum_{h=1}^{H} \varphi\left(\sum_{i=1}^{p} x_i \cdot w^j_{ih}\right) \cdot w_{hj} \tag{4}$$

where H is the number of hidden neurons, φ is the sigmoidal function and $w^j{}_{ih}$ and w_{hj} are the input-to-hidden and hidden-to-output weights of the jth parameter function (see Figure 3).

This type of NN is widely used in function approximation and nonlinear regression, for several reasons: it scales well to high-dimensional problems, the choices of architecture are few and not as critical as for other models (i.e. the number of hidden units), and there exist extensive knowledge on their design.

Inputs and outputs were normalized to zero mean and variance of one, since this speeded up training. The number of hidden units was varied to determine empirically a reasonable network size. We recall that a large number of different models must be trained; each model may indeed require a different number of hidden cells. We found that, while results varied from network to network, networks with more than four hidden units and up to ten units yielded similar (mean) results. We present results with 5 hidden neurons.

The weights of the networks were learned with an improved implementation of the scaled conjugated gradient (SCG) method [3], a second order learning algorithm which that offers the great advantage of setting automatically all training parameters. Our particular SCG algorithm, furthermore, regularizes implicitly the network to obtain low-curvature solutions whenever possible [4] (this reduces the ovefitting effect, yielding better generalization and rendering less critical the choice of the number of hidden units). The only parameters which were left to be determined empricially were (besides the number of hidden units) the maximal training error and maximum training epochs allowed, used to stop the training automatically. These parameters were easily determined after some preliminary experimentation.

Since NNs may approximate any function with arbitrary accuracy, patterns are usually separated into a training set. In this application, however, we have chosen to deal differently with validation.

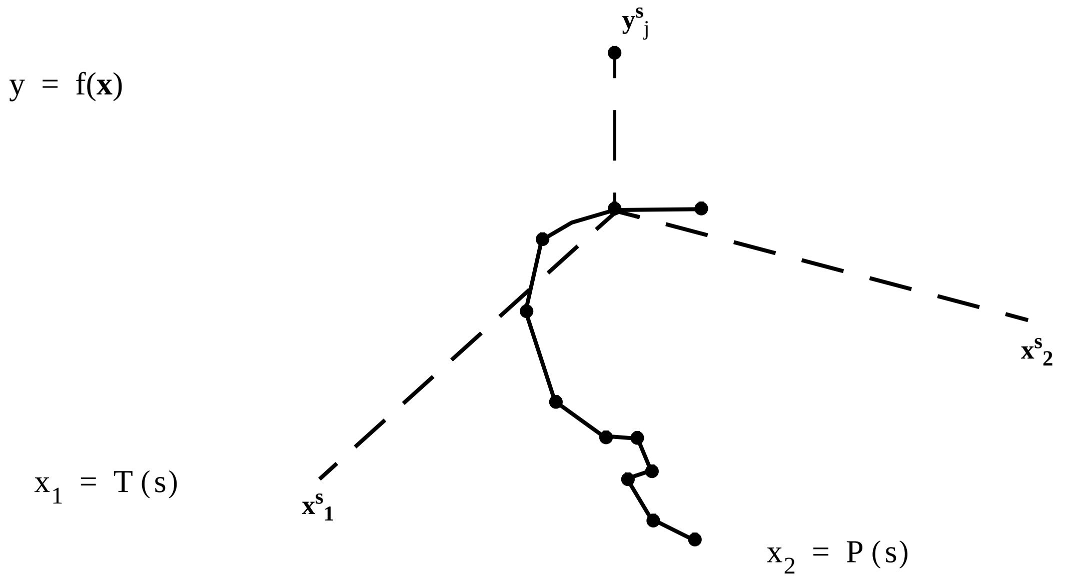

FIGURE 2. A two-dimensional parameter function: We give here a simplified two-dimensional representation of the problem (using temperature (T) and pressure (P) as the only two variables). The points with horizontal projections indicate the LBL training data. The surface has been constructed by regression on these points. The continuous line represents a gas column: each point is a two-dimensional representation of a segment. For each segment *s* , we can then obtain its parameter value (y^s_j), for each parameter-function *j*, from the constructed surface.

The number of points available for fitting the data was small and it was not possible to partition it into reasonable sized train and validation sets. We disposed however of a set of *validation columns* of known radiation intensity but unknown transmissivities (in other words, we knew the value of I for a set of column description vectors x^n of unknown parameter function vectors y^n). We used these validation columns to test the generalization capabilities of our models. This means that, for a given network processing a set of previously unseen vectors, we shall not be interested in the mean squared error of the NN‘s output, but on the absolute error between the LBL (true) intensity of radiation and the intensity obtained from the fitted parameter functions. Note that training and test criteria are different, which makes more difficult the selection of optimal models.

Model:	P-train	P-mix	P-high	P-low
LIN*	0.54	0.02	0.03	0.001
LIN	2.12	0.41	0.52	0.05
NN	0.01	0.10	0.13	0.01
i-NN-35	0.01	0.09	0.06	0.05
i-NN-10	0.04	0.03	0.07	0.05
i-NN-5	0.36	0.01	0.40	0.09

Table 1: Linear and NN Implementations. Absolute intensity of radiation error for the construction column P-train and three validation columns.

Model:	# of params.	hidden units	mean val. err.
LIN*	490(*)	0	0.02
LIN	245	0	0.33
NN	1400	175	0.08
i-NN-35	1470	35	0.07
i-NN-10	630	10	0.05
i-NN-5	210	5	0.17

Table 2: Model Complexity and Mean Validation Error.

Rows two and three of Table 1 present the performances of the linear model (LIN) and the NN model described above, in terms of the absolute intensity of radiation error for the training column (P-train) and for three validation columns: two dealing with low and high pressure situations respectively (P-low and P-high) and another one containing both low and high pressures (P-mix). We present for comparison the results of a hand-crafted model (LIN*) specially built to deal with columns of the type used for validation. This model consists of a piecewise linear regression for which the location of the training points has been carefully chosen and relative weights on these points have been determined empirically to bias the behaviour of the model.

We see that MLPs perform significantly better than the linear model, in training as well as in generalization. With respect to the hand-crafted model, however, generalization of the neural network is not as good. This results must be weighted against the enormous increase of computational cost brought by the MLP models. In the first two rows of Table 2 we show the number of parameters and hidden units, as well as the mean validation error, for the linear and NN models. Leaving aside the cost of computing the non-linear activation functions of the hidden units, the number of parameters of the regression function are increased sixfold (the linear model requires 240 parameters, the NN requires 1400). This number is (at best) linearly related to the computational time needed to compute a solution.

It seemed necessary at this point to reduce the complexity of the model. There exist several techniques to reduce the size of a NN, such as variable selection, regularization and weight pruning [5]. We have experimented extensively several of such methods, but it has been very difficult to apply them successfully to our application for several reasons, the most important being the difficulty, when dealing simultaneously with thousands of different regressors, of determining automatically the different hyper-parameters that these methods demand [6]. In the next section we discuss the approach we adopted to successfully reduce the size of the network

4 MULTIPLE FUNCTION REGRESSION

Several ideas lead naturally to the implementation of multiple function regressors for our application. The combination of regressors, when correlated, may improve generalization. In the case of linear regression it can be shown that a weighted sum of a set of regressors gives a better (or at worst equal) solution than the independent regressors. Appropriate weights may be estimated from data [7]. There do not exist to our knowledge similar results in the case of non-linear function approximation. However, a number of empirical investigations exist in the case of classification tasks [8].

Another interest of combining several regressors into a single neural network architecture is the implicit increase in the speed of computation, since most hidden units and links will be shared and thus the effective number of parameters reduced. This will also speed up the learning phase.

We have constructed a single NN to estimate the thirty-five parameter functions needed for the computation of the intensity of radiation at a single spectral point.To this end we utilized a fully connected MLP with one hidden layer and thirty-five linear output units. Unlike the independent NNs described in the previous section, the function realized by this network is of the form:

$$y_j(\mathbf{x}) = \sum_{h=1}^{H} \varphi(\sum_{i=1}^{p} x_i \cdot w_{ih}) \cdot w_{hj} \tag{5}$$

The difference with (4) is that the input-to-hidden weights w_{ih} are the same for all j (see Figure 3). This is equivalent in fact to linear regression of the type presented in (3), on a set of variables obtained from an adaptive non-linear transformation of the original predictor variables. We hope that function correlations will be captured at the hidden layer (which is now connected to all parameter functions) and used during the training phase to make up for the reduced number of parameters.

In Table 1 (last three rows) we present the performance of such an integrated architecture, for MLPs with 35, 10 and 5 hidden units. Results should be contrasted with Table 2, were we present the number of parameters, hidden units and the mean validation error of each topology. The largest model, i-NN-35, has about the same number of parameters as the NN described in Section 3, but one fifth the number of hidden units. The accuracy obtained in the training column is similar to the one obtained by the simple NN scheme, and the mean error of generalization is slightly lower. The second integrated model, i-NN-10, has about one half the number of parameters than the previous two architectures, and it is not capable of learning as accurately the training functions (the error for the P-train column is higher), but reduces the mean validation error by a factor of 1.7. We see here the double benefit of constraining the network while making use of the correlation of the parameter functions: we have almost doubled the accuracy of our network, using one half of the number of parameters. With respect to the linear network, i-NN-10 utilizes three times more parameters and reduces by seven the mean validation error. Our last integrated model i-NN-5, shows that we may build systems with less parameters than the linear model, and still obtain better results. This contradicts the common intuition that we must «pay the price» of a large increase in parameters if we use non-linear regression. While this is generally the case, it is not so when dealing with large families of correlated functions.

FIGURE 3. Three models of parameter estimation: the independent linear models (top left), which has no hidden units, the independent MLP models (right) which have a separate hidden layer for each output (parameter function), and the integrated MLP model (bottom left), which has a single hidden layer shared by all models.

5 TWO-STAGE SEQUENCE OPTIMIZATION

Whether we employed linear or non-linear regression methods, the models of sections 3 and 4 were concerned with the approximation of the parameter functions, without regard for the role each function plays in the spectral radiation model. However, we can see from (1) and (2) that not all the parameter functions are equally important for the intensity calculations. Depending on the column characteristics some parameters will have negligible effect. Conversely, the accuracy of some parameters will be crucial.

Parameters are in fact the values that the parameter functions take for every segment. A column, being a sequence of segments, constitutes a sequence of points in the parameter's functional domain (see Figure 2). Therefore, we see that the accuracy of the approximations over certain regions of the parameter-functions‘ domain become unimportant, (or, conversely, crucial) depending on the column characteristics.

The models described so far did not take this into account. In effect, the previous regressors were trained by minimizing the mean square error of the parameter function approximations. Taking the mean of the error as the value minimized implies that all points are attributed equal importance (or, more specifically, an importance proportional to the local density of training points). This is particularly dangerous in the case of integrated regressors since shared weights and units may be dominated by functions which may be difficult to approximate but have negligible effect on the ulterior intensity of radiation calculations.

Physicists aware of these facts have traditionally biased by hand their regressors by carefully choosing the distribution of the training points as well as weighting the relative importance of training points, and by subdividing the regions of regression with piece-wise linear regressors. This is necessary since the value optimized for fitting the base functions is the mean square error of their approximations and not the intensity of radiation error obtained *a posteriori* with the parameter functions. Such an heuristic approach leads to very accurate systems, but necessitate of an expensive period of experimentation and calibration which is specific to the application and must be repeated for every new model

To further explore the capabilities of NNs in this application we have gone one step forward and we have sought to integrate the regression of the parameter functions and the computation of the intensity of radiation. We want to make our regression models very accurate over crucial regions, while giving up accuracy if necessary over unimportant regions. These regions are of course unknown: they depend on the characteristics of the particular sequence (column) treated. But they can be found by derivation of the computational chain that leads from the parameter to intensity (eqs. (1) and (2)), for any given column. In fact, these derivation can be carried out in parallel with the learning of the regressors. In this manner, the relative accuracy of the different regions will be controlled by their impact on the accuracy of the final computation of intensity profiles.

This can be done simply, redefining the optimized cost function (normally the mean square regression error) as the quadratic error of the intensity of radiation (the exact formulation of such cost function, as well as the minimization procedure, are described in the Appendix).

Formally, the function minimized by the regressors presented in the previous sections (the mean square function) may be written as:

$$C_j = \frac{1}{2}\sum_i \left(y_j^i - \hat{y}_j^i\right)^2 \tag{6}$$

where C_j is the cost function minimized by the *j*th regressor and y_j^i and $\hat{y}_j^i$ denote respectively the values and the approximated values of parameter function *j* at the point in space defined by segment *i*. In the case of the integrated models, the cost function minimized is the mean of all costs functions C_j minimized.

Note that this cost function does not take into account the order of the segments in the sequence. Once the regressors are found, they are plugged in the computational chain defined by (1) and (2). We propose to minimize directly the (quadratic) error on the computed radiation intensity, with a cost function of the form:

$$C' = \frac{1}{2}\sum_k \left[I\left(\zeta^k\right) - \hat{I}\left(\zeta^k\right)\right]^2 \tag{7}$$

where $I(\zeta^k)$ denotes the spectral radiation intensity of column *k*. Of course radiation intensity depends on the characteristics of each column. In order to minimize (6) all we needed was a representative set of segment descriptions, and their LBL parameter values (see Figure 2). The notion of sequence was unimportant, and therefore any column could be used as long as its segments were well distributed over the entire measure space. Such columns are artificial «construction» columns (like P-train) but are very unlike real gas columns (such as P-min, P-low or P-high). On the other hand, the regressors obtained by minimization with cost function (7) will be extremely dependant on the particular column used.

We see from (1) and (2) that all values in the optimization are constant except for the parameter functions, which are approximated by the models. To solve for C' in (7), we need to compute its derivatives with respect to these approximated parameter functions which are, in fact, the output values of the regression models described in the previous sections. We may therefore use the same MLP architectures to optimize C', with the back-propagation implementation of the gradient descent method; only the derivative of the cost function needs to be redefined (the details of the derivation are presented in the Appendix).

This kind of optimization presents however serious difficulties, since there is only a local guarantee of convergence and the computational cost of a single back-propagation pass is extraordinary. To assure a faster and more subtle convergence of the direct optimization, we chose a two stage learning process in which parameter functions are first approximated separately to a reasonable degree, and only then global sequence optimization is carried out.

The first step of the optimization process is identical to the one described in the previous section. An integrated MLP is trained on the sequences with cost function (6). Note that for such a cost function, the order in which the measurements are presented in the training phase is irrelevant. The second part of the optimization is carried out, on the same model, with cost function (7). This cost function takes into account the entire computational chain, and therefore the order of presentation (that is, the sequence formed by the ordered segment measurements) is crucial to the computation. We must use, therefore, a «typical» gas column to train the model, instead of the arbitrary sequence of training points used in the first optimization step. This second step is in fact similar to the process of hand-tuning the models undertaken by physicists. For example, it is known that certain pressure configurations will make radiation intensity only sensible to the first or last segments of a gas column. For applications dealing in such pressure configurations, models are traditionally biased by hand to be most accurate in the pertinent pressures, to the decrement of other less crucial pressures. The second optimization stage proposed here implements this type of specialization, with the advantage of doing so automatically, simply by establishing a set of typical columns. Its disadvantage is that only a relatively limited amount of «physical expertise» can be injected into our models. As we have previously pointed out, automatically built models cannot compete in accuracy with carefully crafted models, but offer a good compromise for applications for which there is no readily available physical knowledge or its implementation cost is too expensive.

In Table 3 we present the results of this optimization scheme, on the previously discussed i-NN-10 model. The first stage consists on the optimization of parameter functions with a cost function similar to C, as described in Section 3; the results obtained are the same as those of Table 1. We then retrained the MLPs optimizing directly the intensity of radiation of a real gas column (P-Mix), with cost function C'. We present in the second column of values the errors produced by this network on the optimized column (P-Mix), and on the two other validation columns (P-High and P-Low) as well as on the construction column (P-Train). We see that the error on the gas column optimized has been greatly reduced. More important, we see that the error on the other two validation columns has not been increased by this optimization but, on the contrary, it has been slightly reduced. We can therefore expect very accurate predictions on columns similar to those for which the model has been specialized (P-Mix, in this example), without deterioration of the accuracy on other type of columns. Finally, we note that the error on the (local) training column has increased as expected.

	Intensity of Radiation Error	
	1st stage	2nd stage
P-Mix	0.034	*0.005*
P-High	0.066	0.057
P-Low	0.057	0.042
P-Train	*0.024*	0.056

Table 3: Global optimization of the intensity of radiation. Intensity of Radiation errors computed after independent model optimization (1st stage) and after simultaneous model optimization (2nd stage). Italics indicate the column used for training at each stage.

6 CONCLUSIONS

We have presented an effective model of spectral luminance computation based on neural networks. This model transforms a sequence of column measurements into a spectral radiation intensity profile. The main characteristics of the problem is that several thousands of regressors have to be simultaneously estimated. This forbids the use of traditional estimation methods such as cross-validation or regularization. We have shown how to make the computations tractable by taking benefit from the dependence of the different functions to be approximated. The model proposed is much more accurate than its linear counterpart, but it is not as accurate as hand-crafted local linear models. It offers a good compromise between accuracy and implementation cost. For a similar number of parameters, the neural network model halves the average validation error of the linear model, while more complex (but still computationally feasible) models reduce this error by an order of magnitude.

Our model can be extended to the optimization of global sequences, that is, global optimization of all regressor with respect to typical gas columns. This provides the advantage of selectively improving the accuracy of the model on sequences of segments that play a more important role in the application. Our two-stage approach to global optimization permits to specialize in this manner different models on different regions of the input space, after normal training is completed and without deterioration on the overall performance.

Acknowledgments

This material is based on work supported in part by the French aeronautical society SNECMA.

REFERENCES

[1] R. M. Goody and Y. L. Young, Atmospheric Radiation, Oxford Univ. Press. New York, NY, 1989.

[2] Ph. Rivière, A. Soufia and J. Taine, "Correlated-k and fictitious gas methods for H2O near 2.7µm", J. Quant. Spetrosc. Radiat. Transfer, Vol. 48, No. 2, 1992, pp. 187-203.

[3] M.F. Moller, Efficient Training of Feed-Forward Neural Networks, Ph. D. thesis, DAIMI, Arhus University, 1993.

[4] SN2.8 : A Connactionnist Simulator, Appendix R, Nerutistique S.A., Paris, 1993.

[5] B. D. Ripley , Pattern Recognition and Neural Networks, Cambridge Univ. Press. New York, NY, 1996.

[6] H. Zaragoza, "Lessons from a large-scale neural network thermodynamical application: variable reduction, multiple-function approximation and global optimization", IBP Technical Rapport, LAFORIA-IBP (University of Paris 6, France) (in press).

[7] L. Breiman and J.H. Friedman, "Predicting Multivariate Responses in Multiple Linear Regression", Royal Statistical Society, (in press).

[8] R. Caruana, "Learning Many Related Tasks at the Same Time With Backpropagation", Advances in Neural Information Systems, Vol. 7, 1994, pp. 664-657.

APPENDIX : Global Sequence Minimization for the CKFG-N Model.

An heterogeneous column may be discretised in a sequence of homogeneous segments defined by their temperature T_i, pressure P_i, length l_i and molecular composition C_i. The spectral intensity of radiation of such a column, I^v, is defined as the discretised integral of the gradient of the segment‘s mean spectral transmissivity $\tau^v{}_i$ weighted by its vacuum transmissivity I :

$$I = \text{const} + \sum_{i = 1, N} (\tau_{i-1} - \tau_i) I°_i \tag{8}$$

The CKFG-N model formulates the mean spectral transmissivities t_i as the product of N uncorrelated «fictitious» transmissivities, τ_{ji}:

$$\tau_i = \prod_{j = 1, N} \tau_{ji}$$

which are computed in a parametrical form:

$$\tau_{ji} = \sum_{m = 1, 7} A_m \exp\left(\sum_{i' = 1, i} K^*_{ijm}\right) \tag{9}$$

where A_m are constants of integration and K* are the fictitious absorption coefficient functions:

$$K^*_{ijm} = k^*_{ijm}(T_i, P_i) \cdot l_i \cdot C_i \tag{10}$$

Note that, for a given column, T_i, P_i, l_i C_i are constant and the only values undetermined are the two-dimensional k* functions (referred in this paper as the *parameter functions*) which must be fitted from LBL data. This form of parameterization reduces the dimension of the regression; the k* parameter functions are relatively smooth, independent and low-dimensional (although highly nonlinear).

It is clear that optimizing independently each k* function produces a global nonlinear optimization of the whole computational chain, since the rest of terms are constant for a given gas column. But we observe that, depending on the characteristics of the column, certain k* will dominate the calculations, since each of these functions is weighted differently depending on the segment‘s characteristics (10), the characteristics of its fictitious class (9), and the local transmissivity gradient (8).

We are interested in optimizing directly the intensity of radiation (8) instead of minimizing independently the k* functions. We may do this by redefining the minimized cost function as shown in equations (6) and (7). We can derive (7) with respect to each of the k* functions, and use the gradient descent algorithm to optimize this new cost function. These derivatives are provided for completion:

$$\frac{dC}{d\hat{k}_{ijm}} = \frac{dC}{d\hat{I}} \cdot \frac{d}{d\hat{k}_{ijm}}\hat{I} = -(I - \hat{I}) \cdot \frac{d}{d\hat{k}_{ijm}}\hat{I}$$

$$\frac{\partial I}{\partial k_{ijm}} = \left[\sum_{i' = i, N-1} \frac{\partial \tau_{i'}}{\partial k_{ijm}}(I_{i'+1} - I_{i'})\right] - \frac{\partial \tau_N}{\partial k_{ijm}} I_N$$

$$\frac{\partial \tau_{i'}}{\partial k_{ijm}} = \left(\prod_{j' \neq j} \tau_{j'i'}\right) \frac{\partial \tau_{ji'}}{\partial k_{ijm}}$$

$$\frac{\partial \tau_{ji'}}{\partial k_{ijm}} = A_m \exp\left(\sum_{\iota = 1, i'} k_{\iota jm} l_\iota C_\iota\right) l_i C_i \ .$$

TITLE: MULTIPLE MULTIVARIATE REGRESSION AND GLOBAL SEQUENCE OPTIMIZATION: AN APPLICATION TO LARGE SCALE MODELS OF RADIATION INTENSITY.

RUNNING HEAD: MULTIPLE MULTIVARIATE REGRESSION

NUMBER OF MANUSCRIT PAGES: 18

ABSTRACT :

We investigate the strengths and weaknesses of several neural network architectures for a large-scale thermodynamical application in which sequences of measurements from gas columns must be integrated to construct the columns‘ spectral radiation intensity profiles. This is a problem of interest for the aeronautical industry. The approaches proposed for its solution can be applied to a wide range of signal problems. Physical models often make use of a number of fitted functions as a simplified parametric base to approximate a high-dimensional nonlinear (and usually computationally intractable) function. Realistically models of radiation contain thousands of fitted functions. The use of Neural Networks in applications of this scale are rare, and most effective conjunctions techniques rely on cross-validation methods or involve other heavy computational overhead that are impracticable when a very large number of models need to be trained. We have employed here two different approaches: multiple multivariate regression, and global sequence minimization. The first approach shows that the integration of several nonlinear regression models into a single neural network may improve both generalization performance and speed of computation. For the former we propose a method of optimization by which we specialize our models globally, on typical sequences of input signals. We show how this does not degrade the over-all accuracy but, rather, allows us to specialize our models.

COMPLETE ADDRESSES FOR ALL AUTHORS:

H. Zaragoza, P. Gallinari,
LIP6, *Université Pierre et Marie Curie,*
4, place Jussieu F-75252 PARIS cedex 05 (France).
{zaragoza,gallinari}@laforia.ibp.fr

R. Curtelin, F. Leglaye
SNECMA - Villaroche.
7750 Moissy Cramayel (France).

MAILING ADDRESS FOR GALLEY PROOFS:

Hugo Zaragoza,
LIP6, U.P.M,C Boite 169
4, place Jussieu
75252 PARIS Cedex 05
France

LIST OF FIGURE CAPTIONS:

FIGURE 1 :

Column and spectrum discretisation. On the left we see a gas column discretised in segments, indexed by *s*. All measurements are considered constant within each segment. On the right we see the spectrum discretised into spectrum windows of width $\delta\nu$ and index *i*. All computations for each spectral window are carried out independently.

FIGURE 2:

A two-dimensional parameter function: We give here a simplified two-dimensional representation of the problem (using temperature (T) and pressure (P) as the only two variables). The points with horizontal projections indicate the LBL training data. The surface has been constructed by regression on these points. The continuous line represents a gas column: each point is a two-dimensional representation of a segment. For each segment *s* , we can then obtain its parameter value (y^s_j), for each parameter-function *j*, from the constructed surface.

FIGURE 3:

Three models of parameter estimation: the independent linear models (top left), which has no hidden units, the independent MLP models (right) which have a separate hidden layer for each output (parameter function), and the integrated MLP model (bottom left), which has a single hidden layer shared by all models.

LIST OF TABLE CAPTIONS:

Table 1: Linear and NN Implementations. Absolute intensity of radiation error for the construction column P-train and three validation columns.

Table 2: Model Complexity and Mean Validation Error.

Table 3: Global optimization of the intensity of radiation. Intensity of Radiation errors computed after independent model optimization (1st stage) and after simultaneous model optimization (2nd stage). Italics indicate the column used for training at each stage.

LIST OF SYMBOLS USED IN THE ARTICLE: Only standard symbols were used.

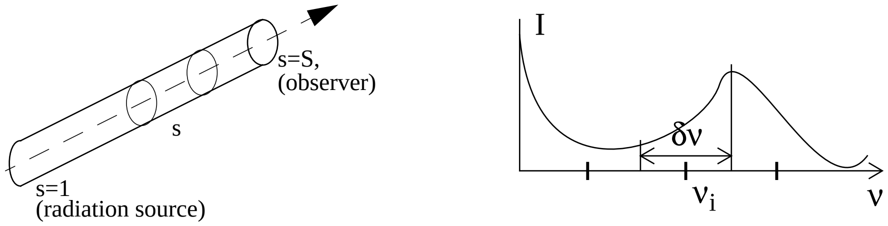

Figure 1

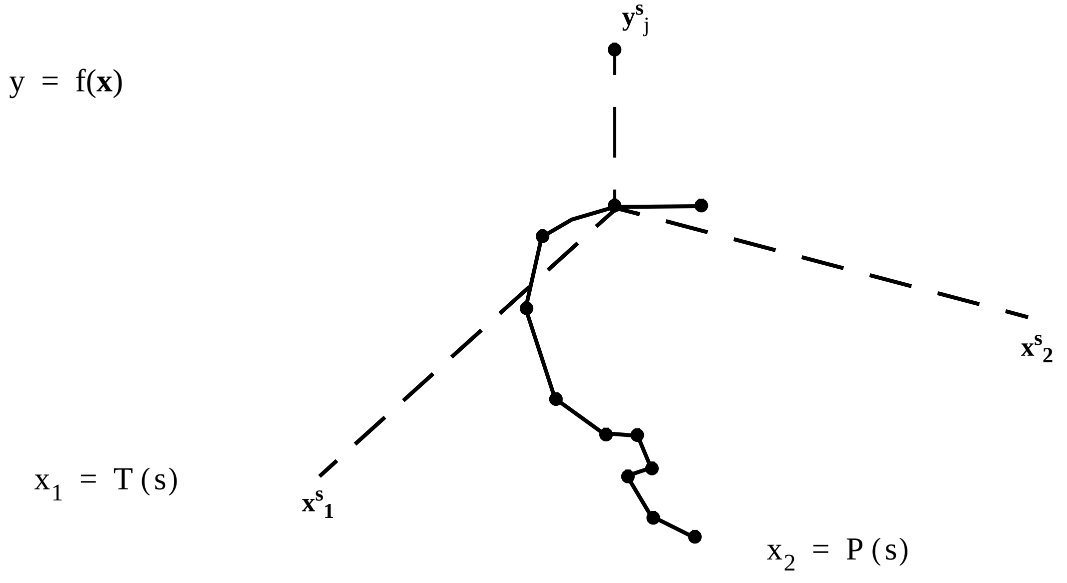
$\mathbf{y}^{\mathbf{s}}{}_{j}$
$y = f(\mathbf{x})$
$\mathbf{x}^{\mathbf{s}}{}_{\mathbf{2}}$
$x_1 = T(s)$
$\mathbf{x}^{\mathbf{s}}{}_{\mathbf{1}}$
$x_2 = P(s)$

Figure 2

Figure 3

Model:	P-train	P-mix	P-high	P-low
LIN*	0.54	0.02	0.03	0.001
LIN	2.12	0.41	0.52	0.05
NN	0.01	0.10	0.13	0.01
i-NN-35	0.01	0.09	0.06	0.05
i-NN-10	0.04	0.03	0.07	0.05
i-NN-5	0.36	0.01	0.40	0.09

Table 1

Model:	# of params.	hidden units	mean val. err.
LIN*	490(*)	0	0.02
LIN	245	0	0.33
NN	1400	175	0.08
i-NN-35	1470	35	0.07
i-NN-10	630	10	0.05
i-NN-5	210	5	0.17

Table 2

	Intensity of Radiation Error	
	1st stage	2nd stage
P-Mix	0.034	*0.005*
P-High	0.066	0.057
P-Low	0.057	0.042
P-Train	*0.024*	0.056

Table 3