OPTIMISING NEURAL NETWORK ARCHITECTURES FOR COMPENSATOR DESIGN

John H Goodband*, Olivier C Haas*, John A Mills+

** Control Theory and Applications Centre, Coventry*
University, Priory Street, Coventry, U.K.
Tel. +44 24 76 888972, Fax: +44 24 76 888052,
e-mail: ctac@coventry.ac.uk
+ Radiotherapy Physics, Walsgrave Hospital,
Clifford Bridge Road, Coventry, UK
Tel. +44 1203 602020 Ext. 7087,
e-mail: jam@walsgrve.demon.co.uk

Abstract: This paper reports on investigations into optimising neural network (NN) design for predicting complex 3-dimensional compensator profiles for intensity modulated radiation therapy (IMRT) treatment. The first part of the paper describes the model used to represent compensator dimensions. The second part describes the methods used to obtain the optimal NN architecture. Results show that all three methods produce NNs capable of zero validation error using a nearest integer error criterion. The degree of accuracy obtained is within clinically accepted bounds and NNs offer a faster means for calculating compensator dimensions than existing algorithms. *Copyright © 2005 IFAC*

Keywords: Compensators, Modelling, Neural Networks, Prediction Methods.

1. INTRODUCTION

The aim of intensity modulated radiation therapy (IMRT) is to deliver a prescribed high dose of radiation to a tumour whilst simultaneously minimising the dose to healthy tissue surrounding the tumour. The use of compensating materials in IMRT is well documented (Haas, 2003; Meyer, 2002; Webb, 2001). Compared with more complex designs such as multi-leaf collimators (MLCs), 3-D compensators are cheap, robust, and quality assessments are easily made (Webb, 2001; Bakai, 2001). Webb comments that the high cost of MLCs (both in terms of hardware and staff training) will preclude the use of IMRT in many centres unless simpler, cheaper methods are utilised (Webb, 2002).

A major problem associated with designing patient specific compensators is that of relating a desired dose distribution (in the form of a matrix of intensities, usually converted into monitor units) to the physical dimensions required to produce this distribution. Forward prediction models (i.e. those calculating the fluence distribution attributable to a particular attenuator profile) are available which provide a high degree of accuracy, especially those using Monte-Carlo techniques (Rogers *et al*, 1995). According to (Harmon *et al*, 1998) the inverse problem of predicting the correct profile given a matrix of intensities desired for a treatment plan is less tractable, since there are a large (theoretically infinite) number of profiles which would produce similar intensity maps. Several methods have been postulated. The general problem is that of speed versus accuracy. Rapid inverse calculation methods rely on simplifications which introduce inaccuracies in the attenuator dimensions. More accurate algorithms are computationally much slower. A comparison between a deconvolution algorithm and a systems identification approach using least squares is made in (Meyer, 2001) demonstrating the long computation times required for the former. Although the least-squares method presented in (Meyer, 2001) is very much faster, reservations are expressed in (Goodband *et al*, 2003) regarding its capability to generalise. For a full discussion of available methods, refer to (Webb, 2001).

Some studies have investigated the use of neural networks (NNs) in radiation therapy treatment planning. The training time may sometimes be long, but, once trained, NNs carry out complex calculations in a fraction of the time taken by inverse algorithms. However, the correct design of NN is crucial if good generalisation (i.e. response to unseen data) is to be achieved. Some of the areas discussed have been classification (Leszcynski *et al*, 1999; Willoughby *et al*, 1996), beam-orientation (Knowles *et al*, 1998), planning target volume prediction (Kaspari *et al*, 1997) and the design of intensity-modulated fields from portal image data (Gulliford *et al*, 2002). A novel approach using NNs to predict compensator dimensions was introduced in (Goodband *et al*, 2004a). Initial investigations showed that a NN can be trained to accurately predict simple 2-dimensional (2-D) compensator profiles using data calculated with an algorithm. In general, the standard backpropagation (BP) algorithm with momentum is used for training multi-layer perceptrons (MLPs). There are other, more efficient algorithms available which work well with NNs of moderate size. The work presented in this paper extends the concept presented in (Goodband *et al*, 2004b) to more complex 3-dimensional (3-D) profiles. By optimising the architecture for a MLP using the Levenberg-Marquart training algorithm and early stopping, the resultant thresholded error is zero. The accuracy of the NN is therefore dependent only on that of the algorithm used to produce training data. Compared with existing methods using sequential calculations, NNs are at least one order faster in computational time.

2. MODELLING

2.1 Modelling X-ray Attenuation

In order to train the NNs introduced in section 3 it is necessary to provide training data. Good training practice is to make the number of data pairs greater than the number of parameters being optimised (Hagan, 1996). For the present NN architectures, it would be impractical to attempt this by experimental means, due to the high cost involved both in terms of time and resources. X-ray fluence profiles are therefore calculated using an algorithm. A similar method is used in (Gulliford *et al*, 2002). A major advantage in using algorithmically generated data is that it is noise free. The algorithm produces a maximum dose error of 4.55%. Details of the algorithm used can be found in (Goodband *et al*, 2004a). It will be noted that more complex algorithms could be used to reduce this error, but without loss of generality, the present method is used to demonstrate the principle. The present study is based on the use of mercury as the attenuator, as part of a wider investigation into the use of liquid-metals as compensators, but the same concept can be extended to any material acceptable for radiation therapy purposes.

2.2 Modelling the Compensator Profile

Elements of compensating material are modelled as discrete, identical units (Figure 1). Each element represents 6.7mm length and 1mm height of the compensator. A 3-D profile is built up using a series of parallel 2-D ‘slices’, 6.7mm apart. For treatment purposes, the compensator is mounted on a linear accelerator (linac) tray at a distance of 67cm from the beam source and 33cm from the isocentre of the prescribed treatment volume i.e. the centre of the tumour (Figure 2).

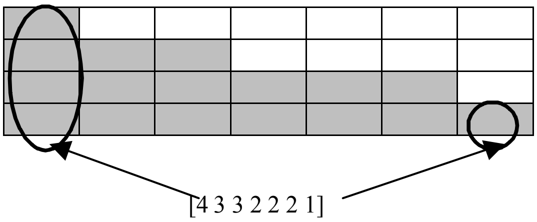

Figure 1.A schematic 2-D vector representation of compensator dimensions.

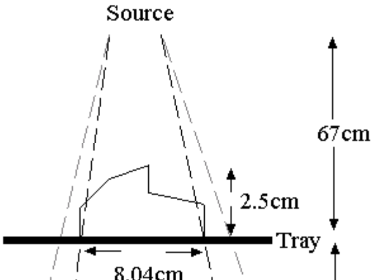

Figure 2. Geometry of attenuation set-up (shown in cross-section: not to scale).

6.7mm at the base of the compensator projects to 10mm length at the treatment isocentre, which is a clinically acceptable resolution. Each vector representing a slice of the compensator contains 12 elements, the total length representing 80.4mm width on the linac tray. In the present study each increment is equivalent to 1mm depth of mercury. 25mm of mercury attenuates 80.4% of the dose from a 6MeV X-ray beam. Each element of the compensator vector has an integer range [0, 25], representing 25 depth increments and a resolution of 1mm. 10 depth levels are used for validation purposes in (Webb, 2002). Although these dimensions are associated with a device designed to fit between the linac head and tray (Goodband *et al,* 2004a), the method can be extended to any dimensions required by altering the size and/or number of the depth increments. The corresponding dose distribution vectors similarly have 12 elements, representing a total width of 120mm at the dose isocentre. Each dose vector element has a continuous range (0, 1], 1 representing a 100% dose from a beam i.e. no attenuating material present. Zero dose is theoretically unattainable as it would be attributable to an infinite depth of attenuating material. The penumbra effect caused by beam divergence (Hendee and Ibbott, 1996) is not modelled since the compensator dimensions are small enough for it to be ignored.

3. NEURAL NETWORK DESIGN

3.1 Network Architecture

The optimal architecture for a NN designed to reproduce a functional mapping may not be immediately obvious. The selection criterion of ‘Occam’s razor’ (Haykin, 1999) is implemented in order to ensure that the NN architecture is not too large for the problem it is designed to solve. This states that the simplest network should be used to solve the input-output relationship. Implicitly this makes the function as smooth as possible. The relationship between attenuator depth and dose distribution is non-linear, but monotonically decreasing i.e. the deeper the attenuating material, the lower the dose delivered. A MLP with one hidden layer is capable of describing any functional mapping of this type (Hertz, 1991) and is therefore used throughout. The most commonly used non-linear activation functions for MLPs are the logsig function

$$y = \frac{1}{1+e^{-u}} \tag{1}$$

and tansig or hyperbolic tangent function

$$y = \frac{e^{u} - e^{-u}}{e^{u} + e^{-u}}. \tag{2}$$

where y is the output produced by an input of u applied to a neuron. Since no correlation exists between training vector elements, the number of input and output neurons is dictated by the size of the training vectors. However, the number and type of hidden neurons required cannot be ascertained *a priori* (Ozturk, 2001), but instead is optimised during the network training.

3.2 Training Algorithm

Although very popular for training feed-forward networks, the BP algorithm (Rumelhart and McClelland, 1986) has been criticised for its slow convergence (Masters, 1993). BP was used in (Goodband *et al*, 2004b) and although reasonable results were obtained, the time taken to train even relatively small networks was found to be very long. The training algorithm used for the present study is the Levenberg-Marquart. This is recognised as a fast method for training small to medium sized NNs, and is a modification of the Gauss-Newton method. It uses the following weight update:

$$\Delta\mathbf{w}_n = -\left[\mathbf{J}_n^T\mathbf{J}_n + \lambda_n\mathbf{I}\right]^{-1}\mathbf{J}_n^T(\mathbf{t}-\mathbf{y}_n) \tag{3}$$

where $\Delta\mathbf{w}_n$ is the *n*th weight update, **J** is the Jacobian matrix containing first derivatives of network errors with respect to weights and biases, λ is an adjustable scalar parameter, **t** is a target vector and **y** the network output. λ is decreased with a reduction in the energy function

$$E_n = (\mathbf{t}-\mathbf{y}_n)^T(\mathbf{t}-\mathbf{y}_n) \tag{4}$$

and increased otherwise. This allows the algorithm to take advantage of the speed of the Gauss-Newton approximation to a second-order method near an error minimum. The decrease factor for the present study is 0.1 and the increase factor 10. A disadvantage of the method is the amount of memory required in computing **J** at each iteration, which restricts its use to networks with no more than a few thousand free parameters. For the present study, this requires a NN to be trained using 2-D slices of the 3-D data. Because of the obvious spatial relationship between compensator dimensions and dose distribution, this is a reasonable simplification, although it does ignore the effect of particle scattering (Hendee and Ibbott, 1996) in a direction orthogonal to the slices.

3.3 Network Training Data

A set of training data is produced covering a range of possible compensator profiles ranging from flat to graded and asymmetric. In all, 1435 different slices are generated. The fluence distribution attributable to each profile is calculated using the algorithm described in (Goodband *et al*, 2004a). Each pair of fluence and dimension vectors is used respectively as the target and input vector for training the NN. Each set of fluence vectors is normalised with zero mean and values in the range [-1, 1]. This accelerates training by preventing weight oscillation. Two sets of dimension vectors are used, one rescaled to lie in the range [0.25, 0.75] by adding 12.5 to each element then dividing by 50, the second set rescaled to [-0.25, 0.25] by subtracting 12.5 and dividing by 50. This allows either logsig, or tansig neurons, respectively, to be utilised on the output layer, thereby avoiding the requirement for an additional linear layer and the associated increase in training time (Haykin, 1999; Goodband *et al*, 2004b).

3.4 Generalisation

Care must be taken to ensure that good generalisation is built into a NN, so that when presented with a previously unseen input it will produce an appropriate output. A NN trained to a very small error may have effectively 'memorised' training data without learning the input-output relationship. Early stopping is therefore implemented using validation data not used during the network training. All training is carried out using 75% of a random sample of the total data whilst the remaining 25% is used only for validation purposes. If the validation mean-squared error (MSE) increases for more than 5 consecutive epochs, training ceases.

3.5 Optimisation

Initially, a small number of architectures are trained to assess the relative merits of using logsig or tansig activation functions using 100 data pairs. In all cases tansig hidden neurons with logsig output neurons return the lowest MSE. Thereafter, all architectures use this combination.

Although genetic algorithms (GAs) have been successfully implemented for optimising NN architectures (Reeves and Steele, 1991; Wu *et al*, 2001; Ozturk, 2003), it is known that GAs cannot be guaranteed to converge to the global minimum of an energy function. They can also be slow, since an unrepresentative training result may cause the genetic search to move in an inappropriate direction. The present study examines a deterministic approach based on increasing the size of data sets and growing the NN architecture incrementally. Training is carried out for a sequence of NN architectures commencing with 1 hidden neuron and 100 data pairs chosen at random from the total set. The optimal number of training pairs depends on the size of NN architecture and the nature of the functional relationship and cannot be established with certainty before training commences. NN theory gives generalised worst-case lower bounds for data size, but these predictions are often impractical to implement (Haykin, 1999). 20 training runs are made for each data set with re-initialisation of random weights for each run. This is consistent with usual ensemble-averaging in NNs (Haykin, 1999) allowing training to commence from different points in the weight space, thereby increasing the chance of at least one of the training runs converging to the global minimum. The number of data pairs is increased incrementally in 100s to 1400 with 20 training runs being carried out for each data set. Three different methods for optimising the NN architecture are compared. In each case, after the 20 training runs are executed for all sizes of data set, for each architecture:

1) The minimum MSE is found.
2) The smallest mean for each set of training runs is found.
3) The smallest mean of each set of training runs excluding the worst single mse from each set is found. This eliminates some of the variance introduced through poor results obtained by training routines which gets trapped in local minima.

A discussion of these different techniques can be found in (Fahlman, 1989). In each case, if there is an improvement on the previous architecture an additional neuron is added and the process is repeated. This is continued until 2 consecutive architectures show no improvement on the previous one, then the training is stopped. This is an arbitrary choice, based on the training time available. Any output vector from an optimally trained network is first rescaled by multiplying by 50 and adding 12.5. Each element is then rounded to its nearest integer (NI) value, corresponding to the resolution used during training. The final error performance is based on the rescaled and NI output. Although a smaller NN architecture could be achieved by using the NI output as an error measurement during training, in the interests of empirical risk minimisation (Haykin, 1999) this method was not adopted.

Training and validation is carried out using the Matlab®6.5 Neural Network toolbox on a Pentium 4 2.0MHz processor and 261 MB RAM.

4. RESULTS AND DISCUSSION

A plot of MSE vs number of hidden neurons for the various architectures is shown in Figure 3 and a plot of data set size vs number of hidden neurons in Figure 4. Method 1 gives an optimal architecture with 17 hidden neurons, achieving a minimum validation MSE of $7.901\times10^{-7} \equiv 4.44\%$ after rescaling and zero NI error, using 1200 data pairs, 9 run stops at 19 hidden neurons for this method.

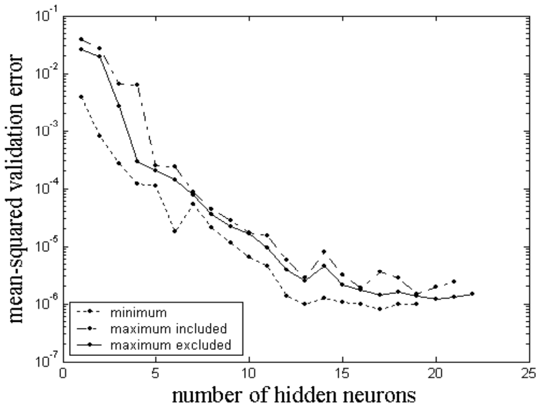

Figure 3. Plot of validation MSE vs number of hidden neurons in NN architecture for methods 1, 2 and 3.

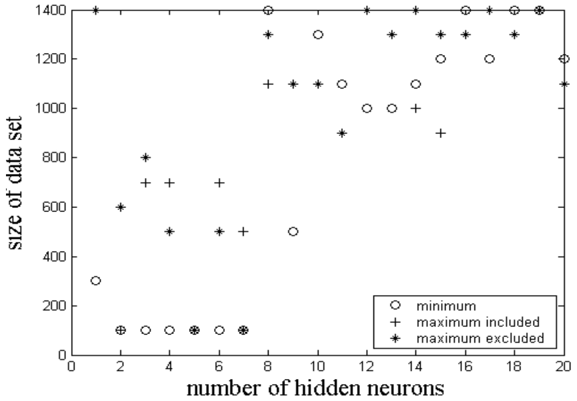

Figure 4. Plot of data set size vs number of hidden neurons for methods 1, 2 and 3.

Method 2 gives a best average of 1.458×10^{-6} using 900 for training with 300 for validation. The training 19 hidden neurons and 1050/350 data pairs. This includes a best minimum of $9.263\times10^{-7}\equiv4.81\%$ MSE after rescaling and zero NI error. Training terminates at 21 hidden neurons. Method 3 predicts an optimal architecture of 20 hidden neurons with an average MSE of 1.204×10^{-6} using 825/275 data pairs and best minimum of $8.935\times10^{-7}\equiv4.73\%$ MSE after rescaling and zero NI error. Training terminates at 22 hidden neurons.

From Figure 3 it can be seen that method 3 produces the smoothest curve of the 3 methods. Eliminating worst results gives a more representative view of the capabilities of an architecture and reduces variance. It is felt that although this produces the longest training time, it is likely to give the optimal NN architecture in general. However, since all 3 methods allow an NN to be trained to zero error using the NI error criterion, the minimum MSE method is clearly the most efficient, and allows a smaller NN architecture to be implemented. Training for a NN with 17 hidden neurons to zero NI validation error takes ~15 minutes and is obtained on average in 20% of training runs.

Figure 4 shows the size of data sets producing the best results using each method. Interestingly, methods 1 and 3 produce the best 7 hidden neuron result using only 100 data pairs i.e. 75 for training, 25 for validation. The number of free parameters for an architecture of this size is $12\times7+7\times12+7+12=187$. Accepted NN theory states that the number of data pairs used in training should exceed the number of free parameters to give reasonable results. For 8 hidden neurons = 212 free parameters, best results are obtained by data sets > 1000 in size, which is what would normally be expected. The results show that it may be possible to use small experimentally obtained data sets to produce reasonable, although not optimal, training results.

5. CONCLUSIONS AND FURTHER WORK

The paper has presented on-going work in the design of a NN to predict compensator profiles for IMRT. Three different methods for optimising NN architectures have been compared. A general recommendation for the third method, which excludes maximum training MSE values has been made, while acknowledging that the method using only best MSE results from training runs is suitable for this specific problem. A NN representing $12\times12=144\text{ cm}^2$ pixels at the prescribed target volume mapping into a 25 depth increment 3-D compensator can therefore be constructed using 144 input neurons, $17\times12=204$ hidden neurons and 144 output neurons.

It has been shown that by training a NN in sections, a more efficient training algorithm can be utilised which allows convergence to a very small mean-squared validation error and zero error after rounding rescaled outputs to nearest integer values. Any error in predicting compensator dimensions is therefore attributable only to the forward algorithm used for calculating fluence distributions. The algorithm used produces a maximum error of 4.55%, which is considered clinically acceptable. Compared with existing sequential methods for calculating compensator dimensions, NNs are at least one order of magnitude faster.

Further work will include integrating a trained NN into a beam-optimisation routine using combinations of compensator designs and investigating beam-divergence to predict penumbra effects for larger compensator dimensions. The use of a more accurate dose distribution calculation, perhaps using Monte-Carlo generated data, will also be assessed.

ACKNOWLEDGEMENTS

This work is sponsored by the Engineering and Physical Sciences Research Council, Industrial Case Training award no. 02303507.

REFERENCES

Bakai A, Laub W U, Nűsslin F (2001). Compensators for IMRT – An Investigation in Quality Assurance, *Z Medical Physics* **11**, 15-22.

Fahlman S E (1988). Fast-Learning Variations on Back Propagation: An Empirical Study, *Proceedings of the 1988 Connectionist Models Summer School, Pittsburg,* 38-51.

Goodband J H, Haas O C L, Mills J A (2003). Attenuation of 6MV X-Rays using mercury compensators, *Proceedings of 16th ICSE,* 188-193.

Goodband J H, Haas O C L, Mills J A (2004a). A reusable attenuation device for intensity modulated radiation therapy, *Proceedings of Controlo 2004,* 636-641.

Goodband J H, Haas O C L, Mills J A (2004b). Neural Network Approaches to Predicting Attenuator Profiles for Radiation Therapy, *Proceedings of Control 2004.*

Gulliford S, Corne D, Rowbottom C, Webb S (2002), Generating compensation designs for tangential breast irradiation with artificial neural networks, *Phys. Med. Biology* **47**(2), 277-288.

Haas O C L (2003). An Intelligent Oncology Workstation for the 21st Century, *Nowotwory Journal of oncology,* Volume **53**(4), 389-397.

Hagan M T, Demuth H B, Beale M (1996). *Neural Network Design*, PWS Publishing.

Harmon J F, Bova F, Meeks S (1998). Inverse Radiosurgery Treatment Planning Through Deconvolution and Constrained Optimisation, *Medical Physics,* **25**(10), 1850-1857.

Haykin S (1999). *Neural Networks: A Comprehensive Foundation*, 2nd ed, Prentice-Hall Inc.

Hendee W R, Ibbott G S (1996). *Radiation Therapy Physics*, 2nd ed, Mosby.

Hertz J, Krogh A, Palmer R G (1991). *Introduction to the Theory of Neural Computation,* Addison-Wesley.

Kaspari N, Michaelis B, Gademann G (1997). Defining Planning Target Volume in radiotherapy for Glioblastoma Mutliforme by using Artificial Neural Networks, *Proceedings of IEEE* 1997, 14-18.

Knowles J, Corne D, Bishop M (1998). Evolutionary Training of Artificial Neural Networks for Radiotherapy Treatment of Cancers, *1998-IEEE-ICEC Proceedings*-Cat No.98 TH 8360, 398-403.

Leszcynski K, Provost D, Bissett R, Cosby S, Boyko S (1999). Computer-Assisted Decision Making in Portal Verification-Optimisation of the Neural NetworkApproach, *Int. Journal of Radiation Oncology Biology Physics* **45**(1), 215-225.

Masters T (1993). *Practical neural network recipes in C++,* Academic Press, Boston.

Meyer J, Burnham K J, Haas O C L, Mills J A, Parvin E M (2002). Application of a least-squares parameter estimation approach for 2-D spatial modelling of compensators for IMRT, *Transactions of the IMC* **24**(5), 369-386.

Meyer J (2001). *PhD Thesis,* Coventry University.

Ozturk, N (2001). Use of genetic algorithm to design optimal neural network structure, Engineering Computations, **20**(7), 979-997.

Reeves C R, Steele N C (1991). Genetic Algorithms and the Design of Artificial Neural Networks, *IEEE 'MICROARCH'*, **6**, 15-20.

Rogers D W O, Faddengon B A, Ding G X, Ma C M, We J, Mackie R (1995), BEAM: A Monte Carlo code to simulate radiotherapy treatment units, *Medical Physics* **22**, 503-524.

Rumelhardt D E, McClelland J L (1986). *Parallel Distributed Processing*, MIT Press, Cambridge, Mass.

Webb S (2002). IMRT using only jaws and a mask, *Phys. Med. Biology* **47**, 257-275.

Webb S (2001). *Intensity-Modulated Radiation Therapy,* Series in Medical Physics, Institute of Physics Publishing, Bristol and Philadelphia.

Willoughby T R, Starkschall G, Janjan N A, Rosen I I (1996). Evaluation and Scoring of Radiotherapy Treatment Plans using an Artificial Neural Network, *Int. Journal of Radiation Oncology Biology Physics,* **34**(4), 923-930.

Wu J-X, Zhou Z-H, Chen Z-Q (2001). Ensemble of GA based Selective Neural Network Ensembles, *Proceedings of the 8th International conference on Neural Information Processing,* **3**, 1477-1482.