Detection of Malignant Melanoma using Neural Classifiers

Mads Hintz-Madsen, Lars Kai Hansen, and Jan Larsen
Dept. of Mathematical Modelling, Technical University of Denmark, DK-2800 Lyngby, Denmark
email: hintz, lkhansen, jlarsen@ei.dtu.dk

Eric Olesen, and Krzysztof T. Drzewiecki
Dept. of Reconstructive Surgery S, The National Hospital of Denmark, DK-2100 Copenhagen, Denmark

Abstract

In this paper we propose a method for design of feed-forward neural classifiers based on regularization and adaptive architectures, and we apply the scheme to the problem of detecting malignant melanoma. Using features acquired from color photographs describing color and texture properties of skin tumors, we are able to detect 76.0±7.8% of melanoma cases in a test set.

1 Introduction

The incidence of malignant melanoma, the most lethal of skin cancers, has risen rapidly during the last 50 years. Fortunately patients can be saved from this life-threatening cancer, if it is detected at an early stage. Thus, in recent years, there has been an increased interest in schemes for automatic and early detection of melanoma. Digital imaging may assist and improve the possibility of such early detections. A recent review of digital imaging in this field was recently published in "Skin Research and Technology" [1]. In this paper we apply a method for design of feed-forward neural classifiers based on regularization and adaptive architectures to the problem of detecting melanoma. Statistical measurements describing color and texture properties are extracted from color photographs of skin tumors and used for classifying tumors.

Key features of the design approach is efficient regularized Gauss-Newton training [2], SoftMax [3] normalization of outputs enabling a probabilistic interpretation of the outputs, architecture optimization and evaluation with Optimal Brain Damage (OBD) [4], and an asymptotic algebraic test error estimate [5].

2 Neural Classifier

Let us assume that we have a training set, D, consisting of q input-output pairs

$$D = \{(\mathbf{x}^\mu, y^\mu) | \mu = 1, ..., q\}, \tag{1}$$

where $\mathbf{x}$ is an input vector consisting of n_I elements and y is the corresponding class label. In this presentation we will assume that the class label is of the definite form $y = 1, ..., n_O$, with n_O being the number of classes. An alternative soft target assignment might be relevant in some practical contexts where the target could be, e.g., an estimate of class probabilities for the given input.

We aim to model the conditional probability distribution

$$p(i|\mathbf{x}), \qquad i = 1, \ldots, n_O. \tag{2}$$

To represent these distributions we choose the following network architecture:

$$h_j(\mathbf{x}^\mu) = \tanh\left(\sum_{k=0}^{n_I} w_{jk} x_k^\mu\right), \qquad \phi_i(\mathbf{x}^\mu) = \sum_{j=0}^{n_H} W_{ij} h_j(\mathbf{x}^\mu), \tag{3}$$

with n_I input units, n_H hidden units, n_O output units, and parameters $\mathbf{u} = (\mathbf{w}, \mathbf{W})$, where w_{j0} and W_{i0} are thresholds. To ensure that the outputs, $\phi_i(\mathbf{x}^\mu)$, can be interpreted as probabilities, we use the normalized exponential transformation known as SoftMax [3]:

$$\hat{p}(i|\mathbf{x}^\mu) \equiv \frac{\exp(\phi_i(\mathbf{x}^\mu))}{\sum_{i'=1}^{n_O} \exp(\phi_{i'}(\mathbf{x}^\mu))}, \tag{4}$$

where $\hat{p}(i|\mathbf{x}^\mu)$ is the estimated probability, that $\mathbf{x}^\mu$ belongs to class i.

The likelihood of the parameters, $\mathbf{u}$, can now be expressed as

$$P(D|\mathbf{u}) = \prod_{\mu=1}^{q} \prod_{i=1}^{n_O} \hat{p}(i|\mathbf{x}^\mu)^{\delta_{i,y^\mu}}, \tag{5}$$

where δ_{i,y^μ} is the Kronecker delta.

Training is based on the minimization of the negative log-likelihood

$$E(\mathbf{u}) = -\frac{1}{q} \log P(D|\mathbf{u}) = -\frac{1}{q} \sum_{\mu=1}^{q} \sum_{i=1}^{n_O} \delta_{i,y^\mu} \left[\phi_i(\mathbf{x}^\mu) - \log\left(\sum_{i'=1}^{n_O} \exp(\phi_{i'}(\mathbf{x}^\mu))\right)\right]. \tag{6}$$

In order to eliminate overfitting and ensure numerical stability, we augment the cost function by a regularization term, e.g. a simple weight decay,

$$C(\mathbf{u}) = E(\mathbf{u}) + \frac{1}{2}\mathbf{u}^{\mathrm{T}}\mathbf{R}\mathbf{u}, \tag{7}$$

where $\mathbf{R}$ is a positive definite diagonal matrix with elements $2\frac{\alpha_j}{q}$.

Using matrix/vector notation the Gauss-Newton paradigm of updating the weights can now be computed as [2]

$$\mathbf{u}^{\mathrm{new}} = \mathbf{u} - \eta\,(\mathbf{H} + \mathbf{R})^{-1}\left[\frac{\partial E}{\partial \mathbf{u}} + \mathbf{R}\mathbf{u}\right], \tag{8}$$

where $\mathbf{Ru}$ and $\mathbf{R}$ are the first and second derivatives of the regularization term respectively, $\mathbf{H}$ is the Hessian matrix containing the second derivatives of (6) w.r.t. the weights and η is a parameter, that may be used to ensure a decrease in the cost function, e.g., by line search.

In order to reduce and optimize a networks architecture, we recommend to apply a pruning scheme such as *Optimal Brain Damage* (OBD) [4]. The aim of OBD is to estimate the importance of the weights for the training error and rank the weights according to their importance. If the importance is estimated using a second order expansion of the training error around its minimum, the *saliency* for a weight u_i is [6]

$$s_i = \left(\mathbf{R}_{ii} + \frac{1}{2}\mathbf{H}_{ii}\right) u_i^2, \tag{9}$$

where the Hessian $\mathbf{H}_{ii}$ and $\mathbf{R}_{ii}$ is the i'th diagonal element of $\mathbf{H}$ and $\mathbf{R}$ respectively.

By repeatedly removing weights with the smallest saliencies and retraining the resulting network, a nested family of networks is obtained. To select the network with the best generalization ability,

	Fully connected ANN			Pruned ANN		
Conf. mat.	Benign nevi	Dyspl. nevi	Melanoma	Benign nevi	Dyspl. nevi	Melanoma
Benign nevi†	42.4±9.1%	29.0±6.2%	22.0±10.1%	47.6±9.3%	27.5±7.6%	18.5±8.8%
Dyspl. nevi†	24.3±5.2%	49.5±6.9%	5.5±3.7%	22.4±6.0%	53.5±5.8%	5.5±4.4%
Melanoma†	33.3±6.7%	21.5±6.7%	72.5±7.6%	30.0±8.4%	19.0±3.9%	76.0±7.8%

Table 1: Confusion matrix for the test set using fully connected (103 weights) and pruned networks (41-62 weights). Note that the classifier performs best for the critical melanoma class. † indicates the ANN output classes.

we can use an estimate of the test error. Using statistical assumptions, see e.g. [7] and [5], the following estimate of the test error of a network $\mathbf{u}$ estimated on a training set D can be derived [5],

$$\langle \widehat{E_{\text{test}}} \rangle = E_{\text{train}}(\mathbf{u}(D)) + \frac{N_{\text{eff}}}{q}, \tag{10}$$

where $E_{\text{train}}(\mathbf{u}(D))$ is the training error of the model. The effective number of weights is given by $N_{\text{eff}} = \text{Tr}[\mathbf{H}(\mathbf{H}+\mathbf{R})^{-1}]$, where $\mathbf{R}$ is the second derivative of the regularization term in (7).

3 Results

From a collection of color photographs of skin tumors at The National Hospital of Denmark 21 statistical measurements describing color and texture properties have been acquired for each tumor and are used for classification into three groups: Benign nevi (non-cancer), dysplastic nevi (non-cancer, but increased risk of developing cancer) and melanoma. A biopsy has been performed on all skin tumors used in this study, thus giving us the "correct" diagnosis for each tumor. This dataset is considered to be difficult as all tumors were labelled "suspect" by doctors prior to the biopsy, and thus excludes obvious cases of non-cancer tumors.

A total of 180 images with an even split of the three classes were used for training and 60 images were used for testing. Ten feed-forward networks with 21 inputs, 4 hidden units and 3 output units were trained[1] and subsequently pruned; hence resulting in ten nested families of pruned networks. The estimate of the test error for each family was used to select the network with the lowest estimated generalization error. Two weight decays were used: $\alpha_w = 0.1$ and $\alpha_W = 1$ for the weights in the input and output layer respectively. In figure 1 a typical pruning scenario is shown. Table 1 shows the confusion matrix for the test set classified[2] with the fully connected networks and the selected "optimal" networks. The mean and standard deviation of the ten runs are reported. Overall the fully connected networks classified 89.6±1.6% of the training set and 54.6±4.1% of the test set correctly, while the results for the pruned networks are 86.7±3.6% for the training set and 58.9±2.1% for the test set. Thus the pruning has increased the generalization ability of the networks. An important effect of the pruning approach is the selection of input features, that are salient for the classification; thus providing us with information that can be used in clinical dermatology. Of the ten pruned networks, four didn't use input 15 and three didn't use input 16. This is an indication that these two color features (blue color mean and red color variance) are unimportant for the classification task.

For comparison a standard *k-Nearest-Neighbor*[3] (k-N-N) classification was performed. The training error may be computed from the training set by including each training pattern in the majority vote. A *leave-one-out* "validation" error on the training set may be computed by excluding each

[1] Training was stopped when the 2-norm of the gradient vector was below 10^{-5}.

[2] Following Bayes decision theory, the network output with the highest probability determines the class label.

[3] Within k-N-N a pattern is classified according to a majority vote among its k nearest neighbors using the simple Euclidean metric.

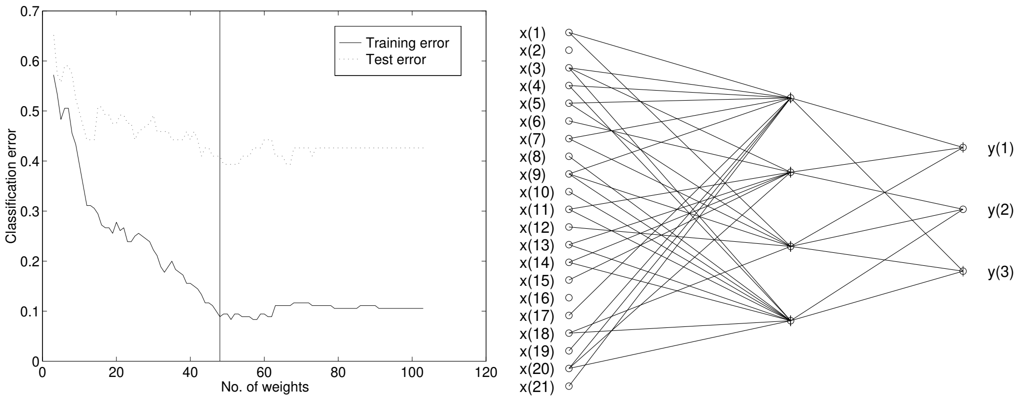

Figure 1: *Left*: Example of a pruning session. The vertical line indicates the "optimal" network selected by the estimated test error. *Right*: Example of a pruned network (from 103 to 48 weights). Note that two inputs are not used.

training pattern from the vote. Finally, the test patterns may be classified by voting among the k nearest neighbors found among the training patterns. Using the validation error we found that $k = 6$ was optimal for this data set. The 6-N-N scheme classified 74.7% of the training set and 57.3% of the test set correctly. Thus the performance of the optimized K-N-N scheme fall in between the pruned and fully connected networks.

4 Conclusion

We have applied a method for design of feed-forward neural classifiers to the problem of detecting melanoma. The approach was shown to be able to design compact neural network models with a generalization ability better than non-pruned network as well as giving us an indication of which input features are unimportant for the classification task; thus enabling us to reevaluate the acquired statistical measurements.

References

[1] W.V. Stoecker, R.H. Moss, F. Ercal, and S.E. Umbaugh. Nondermatoscopic digital imaging of pigmented lesion. *Skin Research and Technology* **1**, pages 7–16, 1995.

[2] C.M. Bishop. *Neural Networks for Pattern Recognition*. Clarendon Press, Oxford, UK, 1995.

[3] J.S. Bridle. *Probabilistic Interpretation of Feedforward Classification Network Outputs with Relationships to Statistical Pattern Recognition*, volume 68 of *Neurocomputing - Algorithms, Architectures and Applications*, pages 227–236. Springer-Verlag, Berlin, 1990.

[4] Y. Le Cun, J. Denker, and S. Solla. Optimal Brain Damage. *Advances in Neural Information Processing Systems*, 2:598–605, 1990.

[5] S. Amari and N. Murata. Statistical Theory of Learning Curves under Entropic Loss Criterion. *Neural Computation*, 5:140–153, 1993.

[6] C. Svarer, L.K. Hansen, and J. Larsen. On Design and Evaluation of Tapped-Delay Neural Network Architectures. *The 1993 IEEE International Conference on Neural Networks*, pages 46–51, 1993.

[7] M. Hintz-Madsen, L.K. Hansen, J.Larsen, E. Olesen, and K.T. Drzewiecki. Design and evaluation of neural classifiers - application to skin lesion classification. *The 1995 IEEE Workshop on Neural Networks for Signal Processing (NNSP'95)*, pages 484–493, 1995.