A way to improve an architecture of neural network classifier for remote sensing applications

Jerzy Korczak, Fatiha Hammadi-Mesmoudi

Centre de Recherche en informatique
Université Louis Pasteur
7 rue R.. Descartes, F-67084 Strasbourg Cedex, France

Abstract. Recent results in neural network research have demonstrated their utility in a variety of application areas. Neural networks are able to achieve a very high performance, and classification accuracy in real world applications such as handwritten character recognition, remote sensing images, vision, robotic. Network performance greatly depends not only on the input/output data, but also on its architecture. Most of neural network applications have been developed using an *ad hoc* approach resulting in poor efficiency and performance. In this paper, a development method of neural network applications is presented, and illustrated with a neural classifier of remote sensing images. It is shown how to create in an iterative way a neural classifier architecture, and how to refine a network organization using performance evaluation criteria.

1. Introduction

The popularity of connectionist models in artificial intelligence has taken wide swings, ranging from extreme enthusiasm in the 1960s to ulter anathema in the 1970s. Currently, there is an explosion of interest in these approaches in many new domains: pattern recognition, classification, speech simulation, database retrieval, etc.

Most of the network architectures today have been designed in *ad hoc* or experimental manner without any consistent methodology of neural network development [1, 2, 3]. In general, the *ad hoc* approach may only be successfully applied for very small applications. But in complex domains, such as image processing, learning experiments are often very costly to perform, in particular where many neural network parameters are not well recognized. Hence, there is an inherent necessity to develop neural network architectures in a systematic way to avoid a high cost of experimentations.

In this paper a development process of neural classifier architectures is presented, and illustrated using as an example classification of remote sensing images. In general, classification techniques applied to digital images involve assigning each pixel (or a group of pixels) in an image with a label describing a real-world object. A class is considered as a group of spectrally homogeneous pixels. To illustrate our approach, we assume that the list of classes has been defined *a priori* by an expert in a form of samples of well-defined pixels. This method, called a user-supervised classification, is a one of the classification methods which attempts to use known pixels representing various classes to classify pixels of unknown identity [4, 5].

2. General schema of network architecture design

The primary goal of a development process is to build an efficient network architecture for a given application, in our case study, a classifier for remote sensed data. Usually, object identification and its classification on remote sensing images is not easy, not only due to the very large amount of data involved (a single image may contain millions of bytes), but also due to the variety of object classes, image resolution, noise, and time depending characteristics.

A good neural network architecture is the key to perform effectively both of these activities. Of course, it is very seldom that a neural network designer finds an optimal graph immediately. In practice, a neural network design is largely done as a trial-and-error process. Usually, a few intuitively selected network architectures are created, and evaluated for a given problem. It is a very costly and time-consuming approach.

In our approach, a development of neural network is seen as an incremental network refinement process.

A systematic approach helps not only to find out a good network architecture but also offers a set of techniques and guide-lines about how to create effectively an architecture of neural networks. However, by following these methods and applying the guide-lines, a reasonable design should emerge, but still a designer's creativity is required to make a decision on some hard to formalize aspects of the network architecture.

The problems encountered in building large neural network applications are not the same as the problems found on writing small network applications In large applications, it is impossible for experts and programmers to hold and maintain details of each aspect of the problem. Our approach is based on the idea of exploratory programming; it involves developing a working classifier, as quickly as possible, and then modifying that systems until it performs classification in an adequate way. The reasons why this model is chosen are related with firstly the difficulty of establishing a detailed requirement specification, and secondly with the aim of the classifier - quality of classification. One of ideas of our approach is based on the principle that the domain knowledge and properties of objects on images have to be encoded gradually during the neural network development. The neural classifier development involves describing the architecture at a number of different levels of abstraction. The development process can be decomposed into the following phases: conceptual design, implementation, and evaluation [6]. A general classification rule of the neural classifier can be defined as follows:

$$\text{function}(\text{pixel}X, \text{attribute}x_1,\ldots,\text{attribute}x_n) \rightarrow \text{class}C_i \tag{1}$$

where *function* represents the read operation and all transitions in the neural networks to establish the pixel-class membership. Attributes are, in the case of radiometric classification, pixel values in different spectral bands, it means the pixel values from corresponding remote sensing images. The goal of design is to find a neural network architecture with the following constraints: learning ratio greater then α, performance ratio greater then β, and if possible the network with minimal number of nodes and connections.

The network is designed by relating and composing meaningful nodes. An example of this method is the rule saying that the nodes have to be connected together if they are conceptually dependent. The refinement process continues until 'atomic' components can be identified. A good rule of thumb is to always choose the simplest solution if all other characteristics are equal. The result of the conceptual design is a specification of network structure and behaviour, expressed in some design language.

Once the network architecture is chosen and the representative examples of predefined classes have been selected, the learning process begins. The neural network tries to learn from the known pixels classified by the expert in the following way. After assigning a pixel to a class according to values computed by the neural network, the result is compared with its known class. Depending on the result of comparison, the classification algorithm modifies the weights of connections according to the learning rule; e.i. it enforces relevant weights in case of success, or it reduces if the matching fails.

The designed classifier is evaluated using testing data, and modified iteratively if necessary. In this phase, the performance of the classifier is measured according to the user-defined criteria. Because of a large number of possible solutions, it is difficult to say definitively what a "good" classifier architecture is. Depending on the application, a good architecture might be a minimal network which satisfies the user requirements, the network with the highest classification accuracy, or a network which is easily modifiable. The initial architecture is usually changed and refined to reflect unperceived earlier user needs and object properties. The modification can be done in an incremental fashion until an acceptable accuracy is obtained.

3. Neural network evaluation and refinement

The neural network evaluation process will be presented using the examples of remote sensing images. In our case study, the input data for classification are three SPOT images (of size of 512×512 bytes). The fourth input image contains a classification given by an expert who has classified each pixel into one of five ground cover classes: water, forest, meadow, husbandry, urban zones. The selected area for classification covers a mountainous area in Vosges (France). The last image is to be used first to train, and then to evaluate the performance of neural network classifier. Therefore the remote-sensing images have been partitioned into two parts: a 75% of image forms a training zone, and the rest is used for evaluating purposes.

The inputs to our network are spectral values of the pixel over all bands together with its neighbouring pixels. To simplify the problem, only the environment of 3×3 pixels is considered. The classes to learn are the following: water, forest, meadow, husbandry and urban zones. The result of forward propagation is a set of values corresponding to predefined classes, where the highest one indicates a class membership of a given pixel. To evaluate the performance of the network, the previously defined classification accuracy ratio ρ will be used. Suppose that a minimal classification accuracy, required by the user, should be greather than 95%. This means that a user wants to obtain at least 95% of correctly classified pixels on a given remote sensing image. The process of network evaluation and refinement is further illustrated by a number of architectures with different level of complexity. The obvious architecture to be tested initially is a neural network with one hidden layer with ($2n$+1) neurons, regarding to the known Kolmogorovs' theorem. Looking into the nature of the objects to be classified, it is evident that geometrical and textural properties have to be represented in the classifier architecture. In our case study, only geometrical relationships will be considered. These relationships can be introduced by hidden layers and specific links between units. Specific links can represent some geometrical properties of classified objects, e.g. vertical or horizontal allignement of pixels. The first refinement of the architecture is to add one more hidden layer, and connect the units according to the horizontal and vertical axes. Our network has two hidden layers. The first contains two sub-layers, each one has three units corresponding to three pixels in a line (horizontal or vertical). The second layer has two units the first for the horizontal axe, and the second for the vertical axe, all connected to the output layer.

The classification accuracy ratio ρ is calculated according to the formula as:

$$\rho = 100\frac{N_c}{N} = 69.6\% \qquad (2)$$

The next refinement of the network architecture is to consider diagonal axes (45° and 135°). So, six units in the first hidden layer have been added, which correspond to all combinations provided from diagonal axe, and one unit in the second hidden layer which represent a diagonal axe. The classification accuracy ρ is again improved 74.4%.

The improvement is caused by introducing in the network architecture more information about pixel relationships. The two previous results suggest a way of improvement of classification accuracy. In the next structure, the corners are to be taken into consideration pixels in corners by introducing 8 additional units in the first hidden layer, and one additional unit in the second hidden layer.

The resulting classification accuracy ρ is greatly improved 98%. This result satisfies the required classification accuracy. It should be mentioned that the process of learning was a little bit, because of the rise of a number of nodes and connections. Of course, there are still possibilities to improve the classifier performance by, for example, extending the neighbourhood of a given pixel, or by introducing other geometrical and textural properties of classified objects.

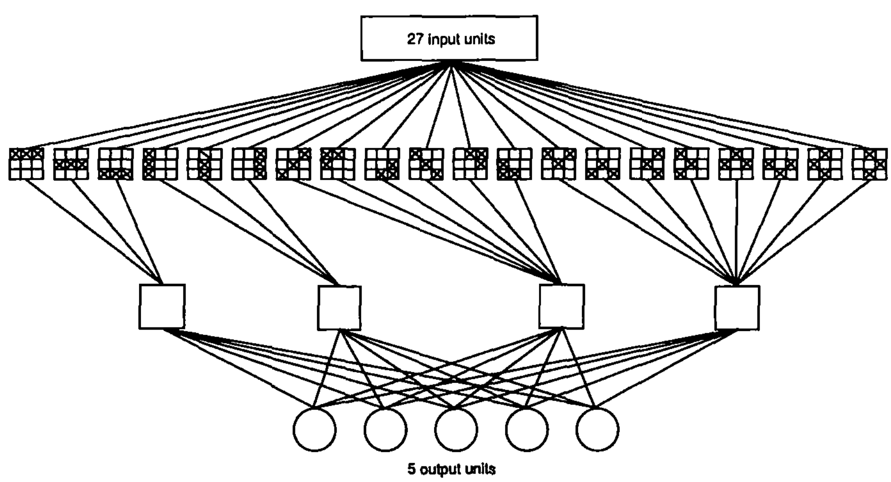

Fig. 1. Final network.

4. Conclusion

In this paper, an approach to design neural network applications, in particular in domain of classification of remote sensing images, is presented. The design process is seen as an iterative process of network refinement. Network architectures have been firstly conceptually specified, then the initial implementation is iteratively refined respecting the user requirements. Our experiences with a neural network classifier have demonstrated that designing a network architecture is much more easier when knowledge about object characteristcs is available and can be encoded into a network architecture. As we are able to identify more object properties, we can achieve a greater control of network improvements, and as this control increases then the accuracy of classifier can be augmented in a more efficient way in contrast to an *ad hoc* technique and a trial-and-error method.

It is possible to get a higher classification accuracy by considering a larger pixel neighbourhood (e.g. 5×5, 7×7, 9×9) by introducing a topographic context (elevation, slope, orientation, etc...). The results of the network modeling and implementation are encouraging, particularly since high cost of unnecesary experimentations on large images can be avoided.

References

[1] J. A. Benediktson. Neural network approaches in classification of multisources remote sensing data, *IEEE Trans. on Geosc. and Remote Sensing*, vol. 28, pp. 540-555, 1990.

[2] R. P. Gorman. Analysis of hidden units in a layered network trained to classify sonar, *Neural Networks*, vol. 1, pp. 75-89, 1988.

[3] N. D. Ritter, G. F. Hepner. Application of an artificial neural network to land cover classification of thematic mapper imagery, *Comp. Geosc.*, vol. 6, pp. 873-880, 1990.

[4] J.J. Korczak. An interactive image processing system for the Macintosh environment, *Proc. PUCC (Vancover)*, 1990.

[5] P. Suetens. Computational strategies for object recognition, *ACM Comp.Survey*, vol. 24, 1992.

[6] J.J. Korczak, F. Hammadi. An approach to design neural network architecture, *Research Report, Louis Pasteur University*, Strasbourg, 1991.