A Priori and A Posteriori Machine Learning and Nonlinear Artificial Neural Networks

Jan Zelinka, Jan Romportl, and Luděk Müller

The Department of Cybernetics, University of West Bohemia, Czech Republic
SpeechTech s.r.o., Czech Republic
{zelinka@,rompi}@kky.zcu.cz, ludek.muller@speechtech.cz

Abstract. The main idea of a priori machine learning is to apply a machine learning method on a machine learning problem itself. We call it "a priori" because the processed data set does not originate from any measurement or other observation. Machine learning which deals with any observation is called "posterior". The paper describes how posterior machine learning can be modified by a priori machine learning. A priori and posterior machine learning algorithms are proposed for artificial neural network training and are tested in the task of audio-visual phoneme classification.

1 Introduction

In this paper we are focusing on a function approximation problem, which is a branch of a supervised machine learning problem. In the function approximation problem the training set S consists of a finite number of pairs $(x; t)$ where x is an argument and t is a target. This problem defines then an approximating function $y = f_0(x, \theta_0)$ and a criterial function $\varepsilon_0(\theta_0, \mathfrak{S}_0)$, where θ_0 is the approximating function parameter and $\mathfrak{S}_0$ is the training set or predetermined statistics computed from the training set.

The goal of the problem is to search for the optimal parameter $\hat{\theta}_0$. The parameter $\hat{\theta}_0$ is optimal if $\varepsilon_0(\hat{\theta}_0, \mathfrak{S}_0) \leq \varepsilon_0(\theta_0, \mathfrak{S}_0)$. If the function approximation problem is unambiguous, the function $\hat{\theta}_0 = \hat{f}_1(\mathfrak{S}_0)$ exists in a mathematical point of view. To avoid nonconstructive reasoning we must consider only computable functions. Unfortunately, a function $\hat{f}_1$ is computable only in some very simple cases, such as least square error in linear regression. The simplest solution is thus to predetermine function $\theta_0 = f_1(\mathfrak{S}_0, \theta_1)$, where θ_1 is a parameter according to [1]. Therefore, the f_1 space dimension equals to the θ_1 space dimension. If $\mathfrak{S}_0$ is a vector of statistics, the simplest form of f_1 is the function $f_1(\mathfrak{S}_0, \theta_1) = \theta_1 \mathfrak{S}_0$. In this paper we have focused only on this form.

The aim of construction of a function approximation method is computation of the optimal parameter $\hat{\theta}_1 = \arg\max_{\theta_1} \varepsilon_1(\theta_1)$, where ε_1 is a criterial function. There are several meaningful criterial functions for θ_1 determination. The first criterial function is the mean of the "basic" criterial function ε_0, i.e. $\varepsilon_1(\theta_1) = \mathrm{E}\{\varepsilon_0(f_1(\mathfrak{S}_0, \theta_1))\}$. Another meaningful criterial function is

$$\varepsilon_1(\theta_1) = \mathrm{E}\left\{\|\theta_0 - f_1(\mathfrak{S}_0, \theta_1)\|^2\right\}.$$

P. Sojka et al. (Eds.): TSD 2010, LNAI 6231, pp. 472–479, 2010.

This choice leads to the optimal parameter

$$\hat{\theta}_1 = \mathrm{E}\left\{\theta_0 \mathfrak{S}_0^{\mathrm{T}}\right\} \left(\mathrm{E}\left\{\mathfrak{S}_0 \mathfrak{S}_0^{\mathrm{T}}\right\}\right)^{-1}.$$

Unfortunately, analytical solution of these means is impossible for common PDFs of $\mathfrak{S}_0$ and θ_1; moreover, usage of such PDFs, for which the means are analytically solvable, seems to be highly speculative. Therefore, a numerical method for θ_1 computing must be applied.

Although the function approximation problem differs from the problem of θ_1 estimation, one way of θ_1 computing is to consider it to be the function approximation problem – and our paper describes two numerical methods based on this methodological assumption.

However, these methods cannot do without relying on some analytically solved function approximation problem – the one we have made use of is a one-layer ANN with weights estimation algorithm which is optimal in compliance with MSE.

2 A Priori and A Posteriori Machine Learning

A priori and a posteriori machine learning basically differs in how the training set is acquired: the training set for the a priori machine learning is acquired without any real observation. The goal of the a priori machine learning is to construct a method for estimation of the parameter θ_0. Its training data consist of a finite number of pairs $(x;\, f_0(x, \theta_0))$, where x and θ_0 are randomly generated. The value of θ_0 is constant for each training sample. The training set thus consists of a finite number of pairs $(\mathfrak{S}_0(TD), \theta_0)$, where TD is the training sample obtained using θ_0. The optimal parameter of the approximating function $\theta_0 = f_1(\mathfrak{S}_0, \theta_1)$ is estimated by means of the function $\hat{\theta}_1 = \hat{f}_2(\mathfrak{S}_1)$, where $\mathfrak{S}_1$ is a vector of statistics computed from the training set.

There are two algorithms for estimation of θ_1. The first algorithm progresses in following steps:

1. For $i = 1, 2, \ldots, N$ do:
   (a) Generate randomly the parameter θ_0^i.
   (b) Generate randomly T examples of the input x into the set X.
   (c) Compute the set

$$TD^i = \left\{(x; t)\, ; x \in X, t = f_0(x, \theta_0^i)\right\}.$$

2. Compute the set

$$S = \left\{\left(\mathfrak{S}_0(TD^i); \theta_0^i\right); i = 1, 2, \ldots, N\right\}.$$

3. The result of the algorithm is $\theta_1 = \hat{f}_2(\mathfrak{S}_1(S))$.

The constants N and T are fixed beforehand.

The second proposed algorithm is an iterative a posteriori modification of the first algorithm [2]. Here the training set S consists of pairs $(x; t)$, where x is an example of the input and t is its respective target. The algorithm progreses in following steps (the number of iterations is fixed to M):

1. Compute initialization θ_1^0 of the parameter θ_1 (e.g. as a result of the first algorithm).
2. For $i = 1, 2, \ldots, M$ do:
   (a) Estimate the parameter $\theta_0^i = f_1(\mathfrak{S}_0, \theta_1^{i-1})$.
   (b) For $j = 1, 2, \ldots, N$ do:
      i. Generate randomly the parameter $\theta_0^{i,j}$ from the close neighborhood of the parameter θ_0^i.
      ii. Select randomly T examples of the input x from the training data S into the set X.
      iii. Compute the set

$$TD^{i,j} = \left\{(x; t)\, ; x \in X, t = f_0(x, \theta_0^{i,j})\right\}.$$

   (c) Compute the set

$$S^i = \left\{\left(\mathfrak{S}_0(TD^{i,j}); \theta_0^{i,j}\right); j = 1, 2, \ldots, N\right\}.$$

   (d) Compute $\theta_1^i = \hat{f}_2(\mathfrak{S}_1(S^i))$.
3. The result of the algorithm is θ_1^M.

The first algorithm offers a general method for function approximation, whereas the second algorithm is a method for function approximation specialized to the given training set.

3 Choice of Statistics

To avoid wholly heuristic choice of statistics, we introduced some more justified considerations. The fisrt consideration is about the optimal parameters estimation for function $y = A \cdot x$. If the number y is a posterior estimation, the special criterial function can be used: $\varepsilon(A) = \sum_i (2t_i - 1)(2y_i - 1)$. The criterial function prefers bigger negative values of y for $t = 0$ and bigger positive values for $t = 1$. Although the criterial function has neither global nor local maximum, a gradient algorithm can be applied. The gradient is $\frac{\partial \varepsilon}{\partial A} = 4\sum_i t_i x_i - 2\sum_i x_i$. The gradient does not depend on the parameter A, thus all steps of the gradient algorithm are possible at once. Hence, the statistics $\sum_i t_i x_i$ and $\sum_i x_i$ are useful for posterior estimation or logical function approximation. Moreover, the statistics $\sum_i x_i$ can be ignored when the mean subtraction is applied.

The second consideration is a more general consideration about data sets. Alternative representation of the data set $S = \{x; x \subset \Re\}$ is a sequence of statistics $(\mathfrak{S}_n(S))_{n=1}^{\infty}$. There is a trivial proof of the theorem that there exists a statistics $\mathfrak{S}_n$ such that the projection of the finite set $S \neq \emptyset$ on the sequence $(\mathfrak{S}_n(S))_{n=1}^{\infty}$ is isomorphism. Nontrivial statistics which privide isomorphic projection are sample moments $\mathfrak{S}_n(S) = \frac{\sum_{x \in S} x^n}{\|S\|}$; analogically for $S = \{x \in \Re^N\}$ where $N > 1$. Consequently a shortened sequence $(\mathfrak{S}_n(S))_{n=1}^{m}$ can be seen as an approximation of a set S.

4 Artificial Neural Networks

Literature describes at least applications of the first algorithm in the area of Artificial Neural Networks (ANN): for example in [3,4] an ANN is used for parameter estimation and it is trained a priori. However, this ANN does not estimate parameters of another ANN – and this is the task we would like to use it in.

All ANNs we use have only one layer. The i-th output of an ANN with one layer is given by the formula

$$y_i = f(x, \theta) = \varphi\left(\sum_{j=1}^{n_x} w_{i,j} x_j + b_i\right), \tag{1}$$

where $i = 1, 2, \ldots, n_y$, $w_{i,j} \in \Re$, $b_i \in \Re$ and $\theta = \left(w_{1,1}, \ldots, w_{n_y,n_x}, b_1, \ldots, b_{n_y}\right)$. If $\varphi(\xi) = \xi$ and the criterion function $\varepsilon(\theta, S)$ is MSE, i.e.

$$\varepsilon(\theta, S) = \sum_{i=1}^{n_S} \|t_i - f(x_i, \theta)\|^2, \tag{2}$$

where $S = \{(x_i; t_i); i = 1, 2, \ldots, n_S\}$ is the training set, then the optimal parameter $\hat{\theta}$ can be computed analytically as the solution of the set of linear equations. For such computation only these statistics are necessary:

$$\mathfrak{S}_A(S) = \frac{1}{n_S} \sum_{i=1}^{n_S} \begin{bmatrix} x_i \\ 1 \end{bmatrix} \left[x_i^{\mathrm{T}}; 1\right], \mathfrak{S}_B(S) = \frac{1}{n_S} \sum_{i=1}^{n_S} t_i \left[x_i^{\mathrm{T}}; 1\right]. \tag{3}$$

Using these statistics the optimal parameters could be computed using the formula

$$\begin{pmatrix} w_{1,1} & \cdots & w_{1,n_x} & b_1 \\ \vdots & \ddots & \vdots & \vdots \\ w_{n_y,1} & \cdots & w_{n_y,n_x} & b_{n_y} \end{pmatrix} = \mathfrak{S}_B \mathfrak{S}_A^{-1}. \tag{4}$$

It is possible to use pseudoinversion instead of inversion if the matrix $\mathfrak{S}_A$ is singular.

We have used this ANN as the function f_1 and both proposed algorithms estimate parameters of this ANN and we have used the statistics

$$\mathfrak{S}_0 = \frac{1}{n_S} \sum_{i=1}^{n_S} t_i x_i^{\mathrm{T}}. \tag{5}$$

Figures 1 and 2 schematically illustrate both algorithms.

In addition to the described ANN we have also used one simple artificial neuron given by the formula

$$y = \sigma\left(wT(x) + b\right), \tag{6}$$

where $x = (x_1, \ldots, x_n)^{\mathrm{T}}$ is an input vector and y is a neuron's output, $T(x) = (x_1, \ldots, x_n, x_1x_1, \ldots, x_1x_n, \ldots, x_nx_n)^{\mathrm{T}}$ and the vector w and the number b are

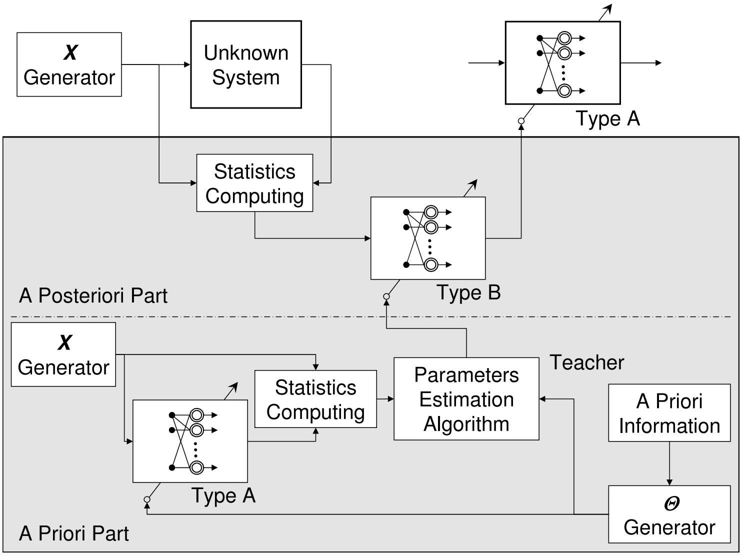

Fig. 1. The first algorithm schematization

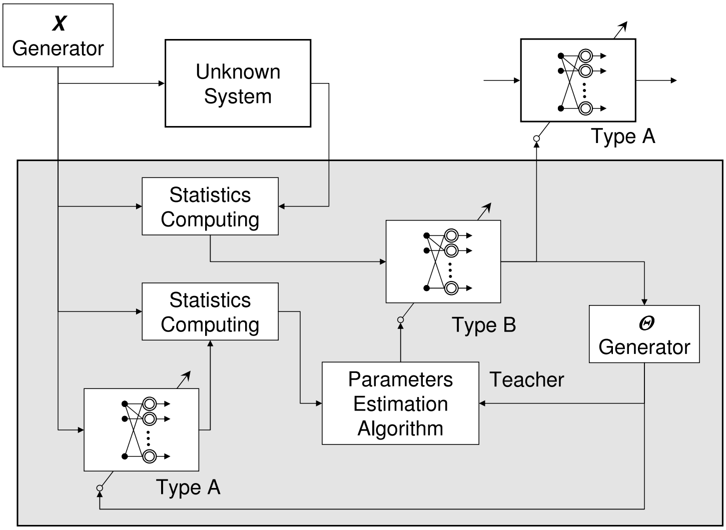

Fig. 2. The second algorithm schematization

parameters of the neuron. We have used the statistics $\mathfrak{S}_0 = (\mathfrak{S}_u(S), \mathfrak{S}_v(S), \mathfrak{S}_w(S))$, where

$$\mathfrak{S}_u(S) = \frac{1}{n_S}\sum_{i=1}^{n_S} t_i T(x_i)^{\mathrm{T}}, \quad \mathfrak{S}_v(S) = \frac{1}{n_S}\sum_{i=1}^{n_S} T(x_i), \mathfrak{S}_w(S) \quad = \frac{1}{n_S}\sum_{i=1}^{n_S} t_i .$$

5 Experiments and Results

We have tested our approach in two different experiments. In the first experiment we have used a priori machine learning (i.e. the first proposed algorithm) for parameter estimation of the artificial neuron described in (6). The algorithm has reached 81,000 iterations during the process. The results for 16 commonly used training sets (based on binary logical operators) is shown in Table 1.

Table 1. The first experiment – a priori learning with 16 training sets based on binary logical operators

i	x_i	t_i	t_i	t_i	t_i	t_i	t_i	t_i	t_i	t_i	t_i	t_i	t_i	t_i	t_i	t_i	t_i
1	0 0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
2	0 1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
3	1 0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
4	1 1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
i	x_i	y_i	y_i	y_i	y_i	y_i	y_i	y_i	y_i	y_i	y_i	y_i	y_i	y_i	y_i	y_i	y_i
1	0 0	0.01	0.01	0.05	0.07	0.05	0.06	0.24	0.28	0.72	0.76	0.94	0.95	0.93	0.95	0.99	0.99
2	0 1	0.01	0.05	0.01	0.07	0.71	0.94	0.76	0.95	0.05	0.24	0.06	0.29	0.93	0.99	0.95	0.99
3	1 0	0.01	0.05	0.72	0.94	0.01	0.07	0.76	0.95	0.05	0.24	0.93	0.99	0.06	0.28	0.94	0.99
4	1 1	0.01	0.71	0.05	0.94	0.05	0.93	0.24	0.99	0.01	0.76	0.07	0.95	0.06	0.95	0.29	0.99

In the second experiment we have used the audio-visual Czech speech database described in [5], where a multi-layer ANN estimates posteriors from acoustic and from visual modality separately (see for example [6]). The goal of our second experiment is to create fusion of these posteriors to achieve higher accuracy of the audio-visual speech data classification into phonemes – this is performed by the aforementioned single-layer ANN. For this experiment the acoustic modality was noised with white noise; the *SNR* was 0.

We have decided to evaluate the posteriors estimation quality as the classification accuracy: an observation o is classified using the mean of posteriors estimation $\tilde{p}(\omega|o)$ as $\arg\max_{\omega\in\Omega} \tilde{p}(\omega|o)$, where Ω is a phonetic alphabet (our phonetic alphabet consists of $n_\Omega = 55$ phonemes).

The input vector of the single-layer ANN is

$$x = \left(p_1^A, \ldots, p_{n_\Omega}^A, p_1^V, \ldots, p_{n_\Omega}^V, p_1^A p_1^V, \ldots, p_{n_\Omega}^A p_{n_\Omega}^V,\right), \tag{7}$$

where p_i^A is the i-th posterior estimation from the acoustic modality and p_i^V is the i-th posterior estimation from the visual modality.

We have selected two disjoint data sets from the corpus: Φ and Ψ. Each data set contains 100,000 examples. The set Φ has been used for training and the set Ψ for testing.

Since we have used ANN also as a training method, there is a danger that not only the resulting fusion method but also the parameter estimation method can become overtrained. High accuracy for the training set and low accuracy for the testing set indicates the fusion method is overtrained; high accuracy in case the fusion method is trained from the training set which is exactly the same as the one used for the parameter estimation training, and low accuracy in other cases indicates the parameters estimation method is overtrained. Both types of overtraining can occur concurrently. To indicate these phenomena we have tested all four combinations of using the training and testing sets for the fusion method training, but only the set Φ was used for the parameter estimation training.

In addition to both proposed algorithms for ANN parameter estimation, several common posteriors fusion methods have been tested: Average of Posteriors (AP), Multiple of Posteriors (MP) and Entropy Based Fusion (EBF) [7]. These common methods do not need any training set, thus their accuracies do not depend on it. All results are shown in Table 2.

Table 2. The second experiment – fusion of posteriors in audio-visual phoneme classification

Training Set	Φ	Ψ	Φ	Ψ
Testing Set	Φ	Φ	Ψ	Ψ
$\left(p_1^V, \ldots, p_{n_\Omega}^V\right)^\mathrm{T}$	38.4%	38.4%	37.3%	37.3%
$\left(p_1^A, \ldots, p_{n_\Omega}^A\right)^\mathrm{T}$	62.0%	62.0%	60.7%	60.7%
AP	63.2%	63.2%	62.2%	62.2%
MP	65.7%	65.7%	65.3%	65.3%
EBF	63.1%	63.1%	62.0%	62.0%
ANN (MSE)	66.3%	66.0%	64.8%	66.3%
ANN (1. alg.)	58.7%	57.9%	58.0%	58.4%
ANN (2. alg., 1. it.)	62.2%	62.2%	60.8%	61.4%
ANN (2. alg., 26. it.)	67.1%	66.6%	65.5%	66.2%

6 Conclusion and Future Work

The main idea of this paper can be simply expressed by the name of the Burnett's article "Learning to learn in a virtual world" [8]. The described a priori machine learning algorithm is only the simplest realization of the sketched idea, i.e. the a priori modification of metalearning [9] – further work will be focused on more elaborate algorithms. Our approach keeps away the self-reference problem [10] because the algorithms are not so strong to provide any self-reference possibility. In the same time our experiments have proved that our approach is fully efficient: the resulting posteriors combination method is – in the worst tested case – as accurate as the described standard ANN and other common methods. Moreover, the resulting ANN parameter estimation algorithm is significantly simpler.

Acknowledgements

This research was supported by the Ministry of Education of the Czech Republic, project No. 2C06020, and by the Grant Agency of the Czech Republic, project No. GAČR 102/08/0707.

References

1. Heskes, T.: Empirical Bayes for Learning to Learn. In: Proceedings of ICML, pp. 367–374. Morgan Kaufmann, San Francisco (2000)
2. Vilalta, R., Drissi, Y.: A Perspective View and Survey of Meta-Learning. Artificial Intelligence Review 18, 77–95 (2002)
3. Kumar, R.: A Neural Network Approach to Rotorcraft Parameters Estimation (2007)
4. Hering, P., Šimandl, M.: Gaussian Sum Approach with Optimal Experiment Design for Neural Network, Honolulu, pp. 425–430. ACTA Press (2007)
5. Císař, P., Železný, M., Krňoul, Z., Kanis, J., Zelinka, J., Müller, L.: Design and Recording of Czech Speech Corpus for Audio-Visual Continuous Speech Recognition. In: AVSP 2005, Vancouver Island, pp. 1–4 (2005)
6. Potamianos, G., Neti, C., Iyengar, G., Helmuth, E.: Large-Vocabulary Audiovisual Speech Recognition: A Summary. In: Proc. Works. Signal Processing, Johns Hopkins Summer 2000 Workshop, pp. 619–624 (2001)
7. Grézl, F.: TRAP-Based Probabilistic Features for Automatic Speech Recognition. Ph.D. thesis, MUNI (2007)
8. Burnett, R.: Learning to Learn in a Virtual World, Milan, Italy. AERA (1999)
9. Brazdil, P., Giraud-Carrier, C., Soares, C., Vilalta, R.: Metalearning: Applications to Data Mining. Springer Publishing Company, Heidelberg (2008) (incorporated)
10. Schmidhuber, J.: Steps Towards ‘Self-Referential’ Neural Learning: A Thought Experiment. Technical Report CU-CS-627-92, Department of Computer Science and Institute of Cognitive Science, University of Colorado, Boulder, Boulder, CO (1992)