Prediction of Upper Body Power of Cross-Country Skiers Using Machine Learning Methods Combined with Feature Selection Algorithms

Mustafa Mikail Özçiloğlu

Mehmet Fatih Akay (advisor)

DISSERTATION.COM

Boca Raton

Prediction of Upper Body Power of Cross-Country Skiers
Using Machine Learning Methods Combined with
Feature Selection Algorithms

Dissertation.com
Boca Raton, Florida
USA • 2017

ISBN-10: 1-61233-449-0 (ebk.)
ISBN-13: 978-1-61233-449-3 (ebk.)

CONTENTS **PAGE**

1. INTRODUCTION

1.1. Overview of Cross-Country Skiing and Upper Body Power

In cross-country skiing, skiers utilize their own condition to move across snow-covered terrain, instead of utilizing ski lifts or other kind of special equipment. Aided by arms pushing on ski poles against the snow, skiers propel themselves either by striding forward (classic style) or side-to-side in a skating motion (skate skiing) (Hindman, 2005). Figure 1 and Figure 2 show typical examples of classic style and skate skiing, respectively.

Figure 1.1. The example for classic style of cross-country skiing

Cross-country skiing is a very strenuous sport because the cross-country skiers intensively use all of upper and lower body musculature (Heil and Camenisc, 2014). Regardless of the style of skiing, the most important component that

determines the performance of a cross-country skier in a race is his/her ability to utilize the upper body power (UBP) (Marsland et al., 2012).

Figure 1.2. The example for skate skiing of cross-country skiing

UBP is the rate at which work can be performed using the arm, shoulder, and trunk muscles. Power generated by the upper body during cross-country skiing is transmitted through the poles and assists in forward motion. For instance, the upper body has been appeared to contribute as much as half to the aggregate propelling force during uphill skating and 15% to 30% during uphill classical skiing (Alsobrook and Heil, 2009b).

A custom-built ergometer is used to simulate a skier's double-poling motion on snow and measure UBP. Double-poling has recently become a popular technique due to the improved preparation of tracks and technological advances in cross-country ski equipment (Heil and Camenisc, 2014). Figure 1.3 shows an experimental setup to measure UBP.

Although measurement using experimental setup is the direct and most accurate method to determine UBP, this method involves several limitations. First of all, the equipment for measuring UBP is not yet usually accessible due to its high cost. The tests of UBP have all been based upon custom-designed ergometers which can only be found in specific sports research laboratories (Alsobrook and Heil, 2009a). In addition, the measurement of UBP hasn't got standardized as it is still a relatively new physiological construct. Also, measuring UBP is a time-consuming process and it requires the presence of a qualified and experienced staff.

Figure 1.3. The laboratory of measuring UBP

1.2. Motivation, Purpose and Contributions of the Thesis

As mentioned in Section 1.1, determination of UBP through measurements comes with several drawbacks. On the other hand, standard sports science and research laboratories have been used by many cross-country skiers and most of these laboratories are capable of measuring maximum oxygen uptake (VO_2max), maximal heart rate (HRmax), heart rate at lactate threshold (HRLT) and several other physiological parameters (height, weight, BMI etc.) which are related to UBP. As a result, it may be advantageous to predict rather than measure UBP using actual training data previously collected by tests on ergometers specifically designed for UBP measurements.

Daneshvar et al. (2013) was the first study in literature that proposed to predict UBP of cross-country skiers by utilizing promising machine learning methods. Although, since then, the problem of predicting UBP of cross-country skiers has been studied in details, the identification of relevant and irrelevant features affecting UBP is still an open area to research.

The objective of the current PhD thesis is to construct new prediction models and determine the discriminative features for prediction of 10-second UBP (UBP_{10}) and 60-second UBP (UBP_{60}) by using General Regression Neural Networks (GRNN), Radial-Basis Function Network (RBF), Multilayer Perceptron (MLP), Support Vector Machine (SVM), Single Decision Tree (SDT) and Tree Boost (TB) combined with several feature selection algorithms. The feature selection algorithms include Relief-F, minimum redundancy maximum relevance (mRMR) and the Correlation-based Feature Subset Selection (CFS). Two versions of the same dataset have been utilized in order to develop UBP prediction models. The first version of the dataset (UBP-set1) involves the features protocol, gender, age, weight, height, VO_2max, HRmax, exercise time and HRLT. In the second version of the dataset (UBP-set2), height and weight are replaced with BMI keeping the rest of features as they are in UBP-set1. By using Relief-F and mRMR, the ranking of the features is obtained. The least effective feature is removed at a time until the model includes a single feature. With CFS, a subset of the important features has been obtained.

Several models have been developed to predict UBP_{10} and UBP_{60} using the mentioned regression methods. 10-fold cross validation has been performed for model testing. Performance metrics including *R, SEE* and *MAPE* have been utilized to measure the efficiency of the prediction models.

To the best of our knowledge, this is the first comprehensive study in literature to determine relevant and irrelevant features and then predict UBP_{10} and UBP_{60}. The results show that utilizing smaller number of features instead of the full set of features yields the most accurate predictions for UBP_{10} and UBP_{60}. The models involving height and weight in place of BMI give lower error rates for the prediction of UBP_{10} and UBP_{60} regardless of which regression methods have been used. Also, GRNN-based models always outperform the other models for predicting UBP_{10} and UBP_{60}.

1.3. Overview of the Dataset

The dataset was obtained from the Montana State University. The dataset includes data of cross-country skiers ranging in age from 15 to 25 years. UBP-set1 includes 75 (38 females and 37 males) healthy skiers. UBP-set1 includes Protocol, Gender, Age, Weight, Height, VO_2max, HRmax, Time, HRLT and also UBP_{10}, UBP_{60} which are target variables. UBP-set2 includes the same subjects. In UBP-set2, height and weight were replaced with BMI.

Brief statistical measures (minimum, maximum, median, mean and standard deviation) of the dataset is given in Table 1.1.

Table 1.1. Statistical information about the dataset

Feature	Minimum	Maximum	Median	Mean	Standard Deviation
Protocol	0.00	1.00	1.00	0.77	0.43
Gender (0: F, 1: M)	0.00	1.00	0.00	0.49	0.50
Age	15.00	25.00	18.00	18.52	2.29
BMI (kg/m2)	17.83	27.90	21.73	22.08	1.94
Weight (kg)	48.30	87.00	67.00	67.09	6.97
Height (cm)	158.30	196.60	173.00	174.40	8.04
VO_2max ($ml.kg^{-1}.min^{-1}$)	46.00	79.01	61.70	62.33	8.38
HRmax (bpm)	180.00	213.00	197.00	196.70	7.34
Time (s)	5.10	13.70	12.00	11.55	1.62
HRLT (bpm)	161.00	210.00	182.00	181.00	9.92
UBP_{10} (W)	110.00	350.00	207.00	225.40	71.08
UBP_{60} (W)	92.00	285.00	170.00	172.40	54.03

1.4. Literature Review

1.4.1. Measurement of UBP

For two different groups of skiers, Alsobrook (2005) investigated the relationship between UBP and classical cross-country ski race performance. The first group included 7 males and 3 females, whereas the second group included 10 males and 5 females. The results show that classical race performance is closely related to both long and short term UBP and that short and long term UBP's are affected by the same factors.

Alsobrok and Heil (2009a) studied the relationship between short and long duration measures of UBP and mass start classical cross-country ski performance. Three different tests of UBP were completed by a group of skiers on a double poling ergometer. The authors concluded that a strong correlation exists between short and long duration UBP tests for mass start classical ski race performance.

Heil et al. (2004) evaluated the influence of the three ski pole grip systems in common use by competitive cross-country skiers on UBP. Nine men who experienced about cross-country ski racing in these systems performed three successive UBP tests on a modified double-poling ergometer. All participants

finished 15-s tests of UBP using stiff cross-country ski poles and a resistance corresponding to 3% of body mass. During the last 10-seconds for every test, the highest 5-seconds average power output was assessed as Peak UBP. Peak UBP with the integrated system yielded higher value than the ones produced by both the modern and traditional systems.

1.4.2. Prediction of UBP without Feature Selection

Daneshvar et al. (2013) predicted UBP_{10} using a multi-layer feed-forward artificial neural network (MFANN). Twelve prediction models were developed by using combinations of different features. This study showed that the model including the features age, gender, body mass and maximum heart rate yielded the lowest *SEE*‘s (24.74 W) whereas the model including the variables age, gender, body mass and time yielded the highest *SEE*‘s (33.06 W) for prediction of UBP.

Sanli et al. (2014) investigated the effects of VO_2max and HRmax on UBP_{10} and UBP_{60} using SVM and compared the results with those obtained by multiple linear regression (MLR). Two groups of common features combined with VO_2max and HRmax were used to create eight different models. First common group included protocol, gender, age, height, weight, exercise time and second common group was comprised of gender, age, BMI, exercise time. The results showed that the combination of the common variables from both groups with VO_2max comparatively yielded the lowest *SEE*’s, while their combination with HRmax gave the highest *SEE*’s. In more details, the model in the first group including the features protocol, age, gender, weight, height, exercise time and VO_2max yielded the lowest *SEE*’s and similarly, the model in the second group including age, gender, exercise time, VO_2max and BMI yields the lowest *SEE*’s for UBP prediction.

Isoglu et al. (2014) aimed to predict UBP_{10} and UBP_{60} using SVM and MLR. The data set which included features (gender, age, body mass, height, HRmax, relative maximum oxygen uptake (RVO_2max), exercise time) and target variables (UBP_{10} and UBP_{60}) consisted of 57 healthy volunteers. Seven different models had been developed to predict UBP_{10} and UBP_{60}. The performance of SVM models were

much better than the ones obtained by MLR for prediction of UBP_{10} and UBP_{60}. Also, the performance of UBP_{60} prediction models was worse than the performance of UBP_{10} prediction models.

Akgol et al. (2015a) used TB, Gene Expression Programming (GEP) and SDT for prediction of UBP_{10} and UBP_{60}. By using the combination of the variables time, VO_2max and HRmax with protocol, age, gender, weight and height seven different prediction models were produced. As a result, TB-based models, in which VO_2max instead of time were used with age, protocol, gender, height and weight performed better than the models developed by the other methods while GEP-based models were more accurate than SDT-based models.

Akay et al. (2015a) developed prediction models for estimation of UBP_{10} and UBP_{60} using SVM, MLP and MLR. The first dataset consisted of gender, age, weight, height, VO_2max, HRmax and exercise time. The second dataset included age, gender, BMI, HRmax, VO_2max, exercise time and also, both datasets consisted of the target variables UBP_{10} and UBP_{60}. SVM-RBF-based models yielded the lowest *SEE*'s and the highest *R's* for prediction of UBP_{10} and UBP_{60}. On the other hand, MLR-based models yielded higher *SEE's* but produced faster results for prediction. In addition, adding BMI to the models significantly increased the performance of models for prediction of UBP_{10} and UBP_{60}. Error rates related to the prediction of UBP_{60} were always lower than that of UBP_{10} due to high standard deviation of UBP_{10}.

Daneshvar et al. (2015) used Cascade Correlation Network (CCN), Decision Tree Forest (DTF) and RBF to create new models for prediction of UBP_{10} and UBP_{60}. The dataset consisted of the features gender, age, BMI, heart rate (HR), exercise time, VO_2max and the target variables were UBP_{10} and UBP_{60}. 14 different models had been created by using these features to predict UBP_{10} and UBP_{60}. According to the results, the model consisting of the features gender, age, BMI and VO_2max gave the highest *SEE*'s for both predictions of UBP_{10} and $UBP_{60,}$ regardless of whether CCN, DTF or RBF had been used. Also, CCN-based models consistently showed the best performance in terms of the achieved *SEE*'s and *R*'s; whereas SDT-based models outperform DTF-based models.

1.4.3. Prediction of UBP with Feature Selection

Akgol et al. (2014) used SVM and MLR combined with feature selection to develop several different models for prediction UBP_{10} and UBP_{60}. The dataset included age, gender, height, weight, exercise time, HRmax and RVO_2max. The ranks of each variable had been calculated by the Relief-F feature selection algorithm combined with a ranker search method. The ranking (from highest to the lowest score) of the features for UBP_{10} is weight, height, age, gender, exercise time, RVO_2max, HRmax and the ranking of the features for UBP_{60} is weight, height, age, RVO_2max, gender, HRmax and exercise time. The models were created by eliminating one variable at a time having the lowest rank score. According to the results, SVM models were better than MLR models for prediction of UBP_{10} and UBP_{60}. The model which included weight, height, age, RVO_2max and gender yielded the lowest *SEE*'s and the highest *R*'s for prediction of $UBP_{60,}$ independent of which regression methods had been used. Additionally, the model comprising of weight, height, age, gender, exercise time and RVO_2max gave the lowest *SEE*'s and the highest *R*'s for prediction of UBP_{10}, regardless of whether SVM or MLR had been used.

Ozciloglu et al. (2015a) built new feature selection based models for prediction of UBP_{10} and UBP_{60} using CCN, MLP and SDT combined with feature selection algorithm. The dataset consisted of the features age, gender, exercise time, BMI, HR_{max}, VO_2max and the target variables UBP_{10} and UBP_{60}. Relief-F feature selection algorithm was used to calculate the ranks of features. According to these calculations, the features exercise time and BMI had been evaluated with the highest score for UBP_{10} and UBP_{60}, respectively. Based on these ranking scores of features, twelve different models had been developed for prediction of UBP_{10} and UBP_{60}. The model involving age, gender, VO_2max, BMI and exercise time gave the lowest *SEE*'s and the highest *R*'s for prediction of UBP_{10}, on the other hand for prediction of $UBP_{60,}$ the model comprised of age, gender, VO_2max and BMI yielded the lowest *SEE*'s and the highest *R*'s, regardless of which regression methods had been used.

CCN-based prediction models yielded better results than the ones obtained by MLP and SDT regression methods.

Akgol et al. (2015b) used GRNN, RBF, DTF regression methods combined with a feature selection algorithm to produce prediction models for predicting UBP_{10} and UBP_{60} of cross-country skiers. The rank of each feature has been calculated by using the Relief-F feature selection algorithm. The dataset was produced by 57 skiers whose ages between 16 and 24. Totally, fourteen different prediction models had been created by removing the feature with the lowest rank at a time. The performance of the prediction models had been evaluated by using 10-fold cross validation. The results show that gender and VO_2max were the most effective variables for prediction of UBP_{10} and UBP_{60}. GRNN-based prediction models performed better than the ones obtained by other methods. On the other hand, RBF-based prediction models were more accurate than DTF-based models. In particular, GRNN-based models yielded in average 9.69% and 24.17% lower *SEE*'s than the *SEE*'s of RBF and DTF-based models for UBP_{10} and UBP_{60}. Also, UBP_{60} prediction models gave lower *SEE*'s than the ones of the models for UBP_{10} due to high standard deviation of UBP_{10}.

Akay et al. (2015b) aimed to develop new prediction models to find out the relevant predictors of UBP_{10} and UBP_{60}. UBP_{10} and UBP_{60} were predicted using SVM, MLP and MLR combined with Relief-F. The results showed that UBP was highly correlated with age and gender, therefore the prediction models including these features gave much lower *SEE*'s and higher *R*'s than other models. In addition, the SVM-RBF-based model comprising of the features age, weight, gender, exercise time and height gave the lowest *SEE*'s (27.81 W) and highest *R*'s (0.92) for prediction of UBP_{10}. The model including the features age, gender, weight, height and VO_2max yielded the lowest *SEE*'s (18.73 W) and highest *R*'s (0.94) for prediction of UBP_{60}.

Ozciloglu et al. (2015b) developed new prediction models for predicting UBP_{10} and UBP_{60} by using GRNN, SDT and Tree Boost along with the mRMR feature selection algorithm. 18 different models had been developed for prediction of UBP_{10} and UBP_{60} by the ranking scores of features assigned with the mRMR. The

results showed that the model including the features gender, weight, height, VO_2max, time, age and HRmax yielded the most reduced *SEE*'s for prediction of UBP_{10}, while the model including the features gender, age, weight, VO_2max, HRLT, height and HRmax led to the most reduced *SEE*'s for prediction of UBP_{60}. Using these two models instead of the full set of features yielded up to 8.80%, 5.20% and 8.26% decrement rates on the average in *SEE*'s for GRNN, Tree Boost and SDT, respectively. Additionally, the feature time provided a significant improvement for prediction of UBP_{10}. A same case for UBP_{60}, the feature VO_2max improved significantly about prediction of UBP_{60}.

Ozciloglu et al. (2016a) determined the effect of features and developed new prediction models on UBP by using different tree based methods including DTF, SDT and TB combined with the Relief-F feature selection algorithm. 16 prediction models had been developed by using Relief-F which calculated the ranks of features. The results emphasized that the features gender, age, BMI and time play an important role for predicting UBP_{10} and UBP_{60}, regardless of which tree-based method has been used. As opposed to the model included full set of features; the model comprising of only 3 features (gender, BMI and time) yielded 5.68%, 3.12% and 7.41% lower *SEE*'s for TB, DTF and SDT-based UBP_{10} prediction models, respectively. As for UBP_{60} prediction models, the model consisted of less number of features (gender, age and BMI) instead of full set of features led to 6.48%, 12.94% and 6.83% lower *SEE*'s for TB, DTF and SDT, respectively. So, the feature BMI and age significantly improved the performance of models for prediction of UBP_{10} and UBP_{60}. Additionally, UBP_{60} prediction models gave lower *SEE*'s than the ones of the models for UBP_{10} due to high standard deviation of UBP_{10}.

In Ozciloglu et al. (2016b), MLP and SDT combined with mRMR feature selection algorithm were used to produce sixteen prediction models for predicting UBP_{10} and UBP_{60}. The results showed that MLP-based models gave in average 16.00% and 14.49% lower *SEE's* than the ones given by the SDT-based models, for predicting UBP_{10} and UBP_{60}, respectively. The prediction model consisting of gender, BMI, VO_2max, HRLT and time led to the lowest *SEE*'s and highest *R's* for prediction of UBP_{10}, while the model comprising of gender, age and BMI yielded the

lowest *SEE*'s and highest *R's* for prediction of UBP_{60}, regardless of which artificial intelligence methods had been used. Applying these two models instead of the full set of features yielded up to 4.95% and 6.83% decrement rates in *SEE*'s for MLP and SDT based UBP prediction models, respectively.

2. OVERVIEW OF METHODS

2.1. Overview of Regression Methods

2.1.1. Generalized Regression Neural Network

A GRNN includes four layers, the names of which are given in Figure 2.1. A typical GRNN architecture is illustrated in Fig. 2.1.

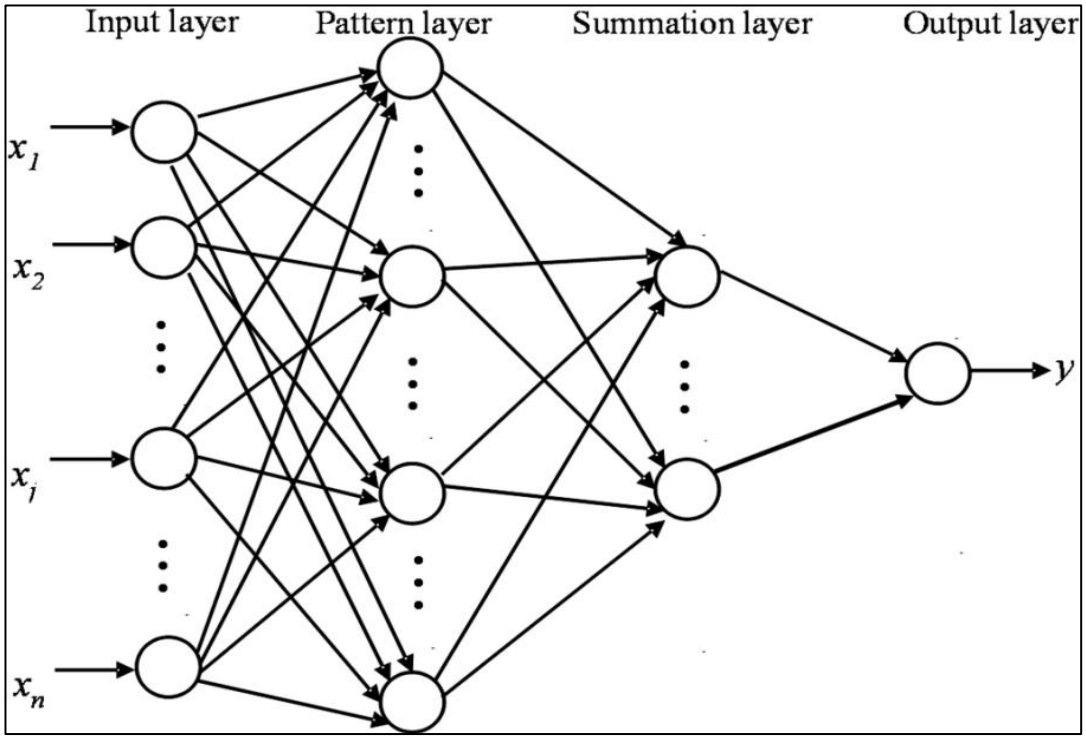

Figure 2.1. A typical GRNN architecture

Total number of features specifies the number of input structures. As shown in Figure 2.1, each layer is directly connected to next layer without having any feedback loops. The summation layer is comprised of a single division unit and summation units. A normalization of the output set is performed by both the summation and output layer.

GRNN can be considered as a technique to predict the joint pdf of *x* and *y*, when only a training set is given. Let *f (x, y)* represent the joint pdf of a vector

random variable, *x* and a scalar random variable, *y*. In this case, the conditional expected value of *y* given X, is calculated by (2.1.).

$$E[y|X] = \frac{\int_{-\infty}^{\infty} yf(X,y)dy}{\int_{-\infty}^{\infty} f(X,y)dy} \tag{2.1.}$$

In case *f (x, y)* is not known, a sample of observations of x and *y* has to be used to predict *f (x, y)*. Let *f'(X, Y)* (given by (2.2.)) be the probability estimator. *f'(X, Y)* includes sample values X^i and Y^i of the random variables x and *y*. In (2.2.), *n* is the number of sample observations, σ is the width and *p* is the dimension of the vector variable x.

$$f'(X,Y) = \frac{1}{(2\pi)^{(p+1)/2}\sigma^{(+1)}}\frac{1}{n}\sum_{i=1}^{n} \exp\left[-\frac{(X-X^i)^T(X-X^i)}{2\sigma^2}\right]\exp\left[-\frac{(Y-Y^i)^2}{2\sigma^2}\right] \tag{2.2.}$$

The scalar function $D_i^2\ is$ given by (2.3.) and (2.4.) is obtained after carrying out the necessary integrations.

$$D_i^2 = \left(X - X^i\right)^T (X - X^i) \tag{2.3.}$$

$$Y'(X) = \frac{\sum_{i=1}^{n} Y^i \exp\left(-\frac{D_i^2}{2\sigma^2}\right)}{\sum_{i=1}^{n} EXP\left(-\frac{D_i^2}{2\sigma^2}\right)} \tag{2.4.}$$

It should be noted that the final equation given by (2.4.) is applicable to problems involving numerical data.

2.1.2. Support Vector Machines

SVM is one of the most promising statistical learning methods for classification and regression. (Vapnik, 1995) developed the principles of SVM. Since then, SVM have gained enormous popularity owing to the fact that SVM having many effective features, and providing good performance. Instead of the ERM principle, which is used by artificial neural networks, SRM has been adopted in SVM (Gunn et al., 1997). The error in the training set is minimized in the ERM, whereas the expected risk is minimized in SRM. The generalization ability, which is one of the main aims of statistical learning, of SVM comes from this difference. Although SVM were originally designed to solve classification problems, they have been lately restructured to solve regression problems (Vapnik et al., 1997).

A. Linear SVM

We are given the training data $(x_i, y_i), (i = 1, ..., \ell)$, where x is a d-dimensional input vector with $x \in \Re^d$ and the output vector is $y \in \Re$. (2.5.) shows the linear regression model (Vapnik, 2000):

$$f(x) = \langle \omega, x \rangle + b, \qquad \omega, x \in \Re^d, \; b \in \Re, \tag{2.5.}$$

In (2.5.), the target function is represented by $f(x)$ and $\langle .,. \rangle$ gives the dot product in $\Re^d$.

To measure the empirical risk, some sort of loss function definition is required. The ε-insensitive loss function, which is proposed by Vapnik (Vapnik, 2000), is the most frequently used function. (2.6.) defines the ε-insensitive loss function:

$$L_\varepsilon(y) = \begin{cases} 0 & \text{for } |f(x) - y| \le \varepsilon \\ |f(x) - y| - \varepsilon & \text{otherwise} \end{cases} \tag{2.6.}$$

The optimization problem given in (2.7.) (Gunn, 1998) should be solved to find out the optimal $\bar{\omega}$ and $\bar{b}$ values

$$\min \ \frac{1}{2}\|\omega\|^2 + C\sum_{i=1}^{\ell}(\xi_i^- + \xi_i^+) \tag{2.7.}$$

with constraints:

$$\begin{aligned} &\mathrm{y_i} - \langle \omega, x_i \rangle - b \le \varepsilon + \xi_i^+, \\ &\langle \omega, x_i \rangle + b - \mathrm{y_i} \le \varepsilon + \xi_i^+, \\ &\xi_i^+, \xi_i^- \ge 0, \quad \mathrm{i}=1,.....,\ell \end{aligned} \tag{2.8.}$$

In (2.8.), there exists toleration for the deviations larger than ε and the tradeoff between the flatness of $f(x)$ is determined by *C*. The deviations from the ε-tube are represented by the variables ξ^- and ξ^+.

The dual optimization problem given in (2.9.)

$$\max_{\alpha,\alpha^*} -\frac{1}{2}\sum_{i=1}^{\ell}\sum_{j=1}^{\ell}(\alpha_i^* - \alpha_i)(\alpha_j^* - \alpha_j)\langle x_i, x_j \rangle - \sum_{i=1}^{\ell} y_i(\alpha_i^* - \alpha_i) - \varepsilon\sum_{i=1}^{\ell}(\alpha_i^* + \alpha_i) \tag{2.9.}$$

with constraints:

$$\begin{aligned} &0 \le \alpha_i, \alpha_i^* \le C, \quad i = 1,...,\ell \ , \\ &\sum_{\mathrm{i}=1}^{\ell}(\alpha_i - \alpha_i^*) = 0 \end{aligned} \tag{2.10.}$$

has to be solved, which in turn gives the optimum values of the Lagrange multipliers α and α^*, while $\bar{\omega}$ and $\bar{b}$ are given by

$$\begin{aligned} &\bar{\omega} = \sum_{i=1}^{\ell}(\alpha_i^* - \alpha_i)x_i \quad , \\ &\bar{b} = -\frac{1}{2}\left\langle \bar{\omega}, (x_r + x_s) \right\rangle \ , \end{aligned} \tag{2.11.}$$

where x_r and x_s are support vectors (Gunn, 1998).

B. Nonlinear SVM

Nonlinear SVM can be constructed using a nonlinear mapping ϕ of the input space onto a higher dimension feature space. (2.12.) shows the nonlinear regression model

$$f(x) = \langle \omega, \phi(x) \rangle + b, \quad \omega, x \in \Re^{\mathrm{d}} \ , \ b \in \Re \ , \tag{2.12.}$$

Where

$$\begin{aligned} &\bar{\omega} = \sum_{i=1}^{\ell} (\alpha_i - \alpha_i^*) \phi(x_i), \\ &\left\langle \omega, \bar{\phi}(x) \right\rangle = \sum_{i=1}^{\ell} (\alpha_i - \alpha_i^*) \langle \phi(x_i), \phi(x) \rangle = \sum_{i=1}^{\ell} (\alpha_i - \alpha_i^*) K(x_i, x), \\ &\bar{b} = -\frac{1}{2} \sum_{i=1}^{\ell} (\alpha_i - \alpha_i^*) (K(x_i, x_r) + K(x_i, x_s)) \end{aligned} \tag{2.13.}$$

the support vectors are represented by x_r and x_s . A Kernel function *K* satisfying Mercer's conditions has been used to explain dot products (Vapnik, 2000).

After $\bar{b}$ is integrated into the kernel function, (2.12.) becomes:

$$\sum_{i=1}^{\ell} (\alpha_i - \alpha_i^*) K(x_i, x) \tag{2.14.}$$

Many different kernel functions including the radial basis function (RBF), the polynomial function and the sigmoid function exist. However; RBF, as defined in (2.15.) is the most commonly used Kernel function:

$$K(x, x') = \exp\left(-\frac{\|x - x'\|^2}{2\rho^2}\right). \tag{2.15.}$$

In (2.15.) the width of the RBF function is defined by ρ.

2.1.3. Tree Boost

Tree Boost is some sort of a machine learning algorithm used for the solution of both regression and classification problems, which creates an estimation model that has the shape of a group of low estimation models. Boosting Tree algorithm is calculated by using boosting methods for regression trees. The common idea is to evaluate an order of simple trees, where every progressive tree is constructed for the estimation leftover of the former tree. This method will construct binary trees. It is assumed that the complexities of the trees are limited with only 3 nodes. These are two child nodes and a root node. Therefore, at each step of the boosting, a basic dividing of the data is specified, and the deviations of the noted values from the individual means are calculated. The following 3-node tree will be fitted to those residuals, in order to find alternate partition that will further decrease the residual variance for the data, given the former a group of trees (Hastie et al., 2009).

A boosted classifier (or regressor) has the form

$$H(x) = \sum_t \alpha_t h_t(x) \tag{2.16.}$$

This form can be trained by optimizing scalar α_t and weak learner $h_t(x)$ at each iteration *t*. Before training begins, each data sample x_i is assigned a non-negative weight w_i (which is derived from L). After each iteration, misclassified samples are weighted more heavily thereby increasing the intensity of misclassifying them in next iterations. Regardless of the type of Boosting used (i.e. AdaBoost, LogitBoost, L2Boost, etc.), each iteration requires training a new weak learner given the sample weights (Appel et al., 2013).

2.1.4. Multilayer Perceptron

Several problems in different fields including control, function approximation, pattern recognition and associative memory (Haykin, 1994) can be solved by the biologically inspired intelligent systems called Artificial Neural Networks (ANN's). ANN's are comprised of many simple processors interconnected by many interconnection networks. The use of ANN's provides the following advantages: learning ability, generalization ability, fault tolerance, input–output mapping, low energy consumption, nonlinearity, distributed representation and computation (Huang, 2009).

Multi-layer feed-forward ANN's (i.e. MLP) including one input layer, one output layer, and some hidden layers are one of the most useful and popular structures. The back propagation algorithm that relies on minimization of a convenient cost function is used for training the MLP.

A typical MLP is shown Figure 2.2. A general L-layer FNN includes input hidden and output layers. The layers are connected in a feed-forward style. There are no connections between the neurons of the same layer. Also, there is no feedback in the network structure. In Figure 2.2., the inputs are donated by $x_1, \dots, x_I$, the neurons of the hidden layer are donated by $y_1, \dots, y_J$, and the outputs are denoted by $o_1, \dots, o_K$. The weights in the input layer are shown by V whereas the weights in the hidden layer are shown by W.

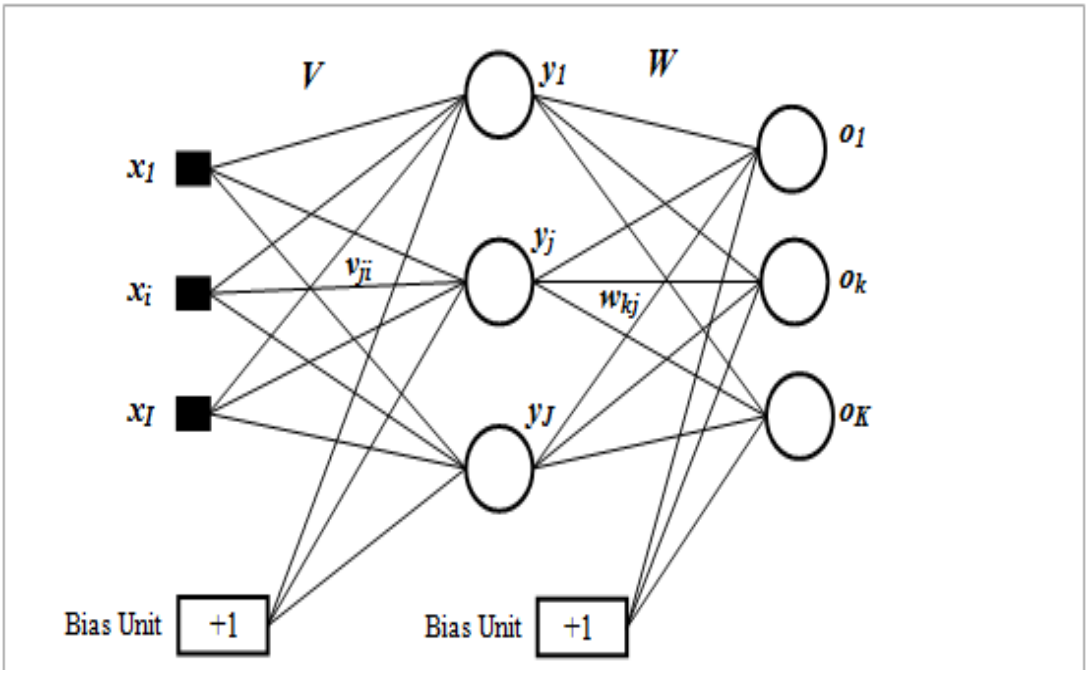

Figure 2.2. A typical MLP structure.

The most commonly used training method for the MLP is the back-propagation, which is basically used to determine the final weights of network. (Li et al., 2012). In the back-propagation algorithm, the error calculated in the output layer neurons is back-propagated to the hidden layer using the gradient-descent method for minimizing the squared-error cost function in neurons according to the formula given in (2.17.)

$$E_p = \frac{1}{2}\sum_{k=1}^{K}(d_{kp} + o_{kp})^2 + E_p \qquad (2.17.)$$

In (2.17.), the count of training patterns is given by *p*, *o* is the real output, the term E_p corresponds to the sum of the squared errors of the cost function and *d* represents the desired output matrix with size of *K*, o is an actual output. Different training functions including Quasi-Newton, Fletcher–Reeves and Levenberg–Marquardt can be used for back propagation.

2.1.5. Radial Basis Function Network

RBF's appeared as a variant of artificial neural network in the last years of 1980. Nonetheless, their roots are settled in much more pattern recognition techniques. These techniques can be clustering, mixture models, potential functions, spline interpolation and functional approximation.

RBF network includes several layers and the first layer has input neurons. These neurons feed the feature vectors into the network. The second one is the hidden layer that calculates the result of the main functions. The last layer is the output layer that calculates a linear combination of the main functions (Park and Sanberg, 1991). Simple structures of these networks give a decreasing for training time and make possible learning in stages.

RBF has the feed-forward structure. It has one layer that consists of hidden units. These units are completely adjusted with the output units. (2.18) indicates the output units (ψ_j) from a linear combination of the main functions (Ghosh-Dastidar et al., 2008).

$$\psi_j(x) = K\left(\frac{\left\| x - c_j \right\|}{\sigma_j^2}\right) \tag{2.18.}$$

$f : R^n \to R^1$ is estimated by using an RBF network. $x \in R^n$ is an input, $\psi(x, c_j, \sigma_j)$ is the j^{th} function with centre $c_j \in R^n$, width σ_j, and $w = (w_1, w_2, ..., w_m) \in R^M$ is the vector of linear output weights. M represents the main number function. The M centres $c_j \in R^n$ are connected to obtain $c = (c_1, c_2, ..., c_m) \in R^{nM}$. Lastly, the widths are connected to obtain $\sigma = (\sigma_1, \sigma_2, ..., \sigma_M) \in R^M$. The output of the network for $x \in R^n$ and $\sigma \in R^M$ is shown by (2.19).