Optimized Functional Link Artificial Neural Network for Multi-label Classification

Anwesha Law[1], Balasubramanyam Evani[2], and Ashish Ghosh[1]

[1] Indian Statistical Institute, Kolkata, India
{anweshalaw_r, ash}@isical.ac.in
[2] Manipal University Jaipur, Jaipur, India
balasubramanyam.evani@gmail.com

Abstract. Multi-label classification problem is a generalization of the traditional single-label classification, which makes the data more complex in nature. To handle the inherent complexity of multi-label data, a compact and efficient network known as functional link artificial neural network (FLANN) has been explored. FLANNs are known to functionally transform the input space to introduce non-linearity into the data, thus making the task of separating the classes in the output space comparatively simpler. In this paper, six multi-label FLANN models (five novel and one existing) have been devised for multi-label classification to procure the optimal configuration. These six variations of the network have been built using three basis functions - trigonometric, Chebychev, power polynomial and two learning techniques - backpropagation and particle swarm optimization. These fundamentals of FLANN have been thoroughly explored in the single-label domain, but are yet to be experimented for multi-label data. These multi-label FLANN models were tested on four datasets for ten performance metrics. Analysis of the results have generated some interesting conclusions and optimal models for multi-label classification.

Keywords: Multi-label classification · functional link artificial neural networks · particle swarm optimization

1 Introduction

Classification is a predictive data mining task, usually conducted by means of supervised learning. Traditional classification problems are annotated with a single-label, however, in the real world, data is mostly tagged with multiple labels. Multi-label classification, for example, is the task of tagging music with multiple emotions, categorizing movies into more than one genres and so on. Basically, the aim of multi-label classification is to predict a set of labels for each data instance as opposed to single-label classification where each input instance is associated with a singular class or label. Multi-label classification can be seen as a generalization of single-label classification. The methods to deal with the generality of multi-label data to produce the overlapping and convoluted boundaries can be broadly divided into - *data transformation* and *problem adaptation* based methods [9]. Most of the recent works [16, 17] in the past decade has been focused towards the problem adaptation based methods. This branch focuses on modifying the existing classification algorithms for multi-label data. While solving the problem of classification, artificial neural networks have shown the capability of solving very complex, non-linear data relationships [8]. The flexibility exhibited by neural networks make it capable of learning any type of pattern. They are loosely inspired by how the human brain works; specifically, its interconnected neuron structure and its activation resembles the biological synaptic connections and its firing. This makes it capable of handling complex classification tasks. However, in literature, the application of neural network models for multi-label classification is somewhat limited. Adaptations of multi-layer perceptron (MLP) [17], radial basis networks (RBF) [16], functional link artificial neural networks (FLANN) [10], extreme learning machines (ELM) [11], etc. as multi-label classifiers have been shown in the literature. There is still scope of exploring this area with the problem of multi-label classification in mind.

Among the various networks explored, FLANN is one such network that is compact yet efficient. Multiple FLANN adaptations for single-label classification [3,4,12] exist in literature, but it is yet to be sufficiently experimented in the multi-label domain. FLANNs are feed forward networks where the use of a hidden layer is omitted by non-linearly transforming the input features with some basis functions. The expanded input layer portrays a higher dimension projection of the input with better discriminating characteristics. Multi-label data is inherently quite complex in nature, mostly due to its multiple overlapping class boundaries. This calls for models that can handle this bottleneck and improve on it. Therefore, FLANN is an apt choice in this scenario. In the present paper, various models of FLANN have been devised in order to analyse each of their importance as a multi-label classifier. Three types of functional expansions viz. trigonometric, Chebychev and power polynomial expansions and two weight optimization techniques, namely, backpropagation and particle swarm optimization (PSO) have been included in this paper.

Combination of one functional expansion and one weight optimization technique has led to six FLANN models for multi-label classification. Out of these six, five are novel configurations and one model exists in literature [10]. These multi-label FLANN models have been tested on four datasets and ten performance metrics to appreciate the overall performance for each of them. Comparative analysis indicates optimal performance from few of the models over the others making them efficient multi-label classifiers.

The next section touches upon a brief literature survey. Section 3 discusses the fundamentals of FLANN along with the basis functions and learning techniques to be used. Section 4 provides an analysis of the results obtained for the six FLANN models and Section 5 concludes the paper.

2 Literature Survey

Exploring the literature of multi-label classification, we come across few neural network based techniques which have tailored the original models to suit the problem at hand. BP-MLL [17] is an adaptation of the traditional multi-layer perceptron with a different error computation technique, more suited for multi-label data. MLRBF is another simple adaptation of the traditional RBF network. Similarly, [14] developed a multi-label extreme learning machine (MLELM) for fast classification. Generally, ELMs are capable of learning in one pass, but not so efficient in handling multi-label data on its own. It has led to the development of other more efficient models [11] which can deal with multi-label data in a better way. Apart from these neural network models, FLANN is a compact yet efficient model that has newly found its way in multi-label classification domain. The primary FLANN model uses a basis function to non-linearly expand the input data, and a learning technique to update the weights of the network. Different combination of basis functions and learning mechanisms have been experimented with in the past. In the multi-label domain, [10] proposed a multi-label FLANN (MLFLANN) which uses trigonometric input expansion and backpropagation for weight update. However, there are numerous other models of FLANN in literature [3] that have been developed for single-label classification, and can be adapted to classify multi-label data as well. Various models of single-label FLANN [12] have performed trigonometric input expansion and backpropagation (BP) learning methodology. This combination is quite popular for single-label classification and have reportedly performed well. The method developed in [4] also uses trigonometric expansion but instead of BP, PSO with Cauchy and Gaussian mutation operator has been used for training of the weights. [3] used a Chebychev expansion along with an adaptive version of PSO (aPSO) combined with BP as their learning scheme for single-label classification. The initial weights were first found using aPSO and then this was used as a starting point for the BP learning for fine tuning. It is seen that FLANN based classifiers have been well-explored in the domain of single-label classification, and need similar enthusiasm in the multi-label domain. Thus, in the proposed work, three basis functions, namely, trigonometric, Chebychev and power polynomial have been focused upon, along with two learning approaches – backpropagation and PSO. The current work combines the basis functions and learning mechanisms to build six models of FLANN for multi-label classification. Details of the approaches have been discussed in the following sections.

3 FLANN for multi-label classification

A multi-label classification problem can be defined as the mapping of input $X = \{X_1, X_2, ..., X_D\}$ to output $Y = \{Y_1, Y_2, ..., Y_C\}$, where, D is the input dimension and C is the output dimension. The output here denotes the multiple classes that input X belongs to. This can be portrayed as a one-to-many mapping, which makes it more complex than traditional one-to-one single-label classification. FLANN is known to handle complex classification at par with multi-layer perceptrons for single labelled data. The main idea of FLANN is to introduce non-linearity in the data by functionally expanding the original input. From Cover's theorem [2], it can be said that a higher dimensional transformation makes the original data more separable, thus, simplifying the task of classification. This is a primary reason of exploring FLANN models to handle the complex multi-label data. The basic model of multi-label FLANN is shown in Fig. 1. It contains an input layer for D features $X_1, ..., X_D$. These input features are expanded by the functional expansion unit with the help of some basis functions which introduce non-linearity in the data. Now, each input feature X_i is expanded to q corresponding features which represent the final amplified input. The functionally expanded input X_i' can be thus represented as:

$$X_i' = \{f_0(X_i), f_1(X_i), \ldots, f_q(X_i)\} \tag{1}$$

where, $f_0(.), f_1(.), \ldots, f_q(.)$ are the q basis functions that are used for expansion of the inputs. There are various types of basis functions for traditional FLANN in literature which serve different purposes. Few of them have been

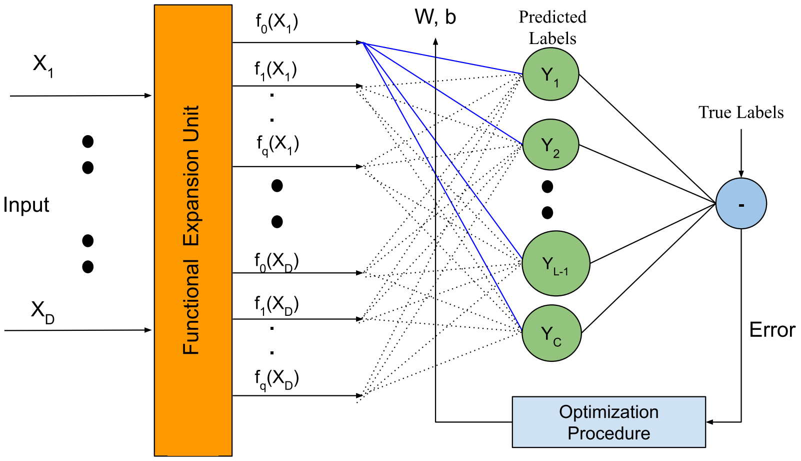

Fig. 1. General architecture of Multi-label FLANN

discussed in Section 3.1. After expansion, the features are directly mapped to the C nodes of the output layer with the help of weighted connections. The nodes at the output layer compute the predicted score for each class at the corresponding nodes. Weighted sums of the expanded inputs are computed at the output layer. These predicted scores are iteratively moved towards the target scores by updating the weights and reducing the error. These weights, W, are learned iteratively during the training phase of the network. Researchers have incorporated different weight optimization mechanisms, two of which are discussed in Section 3.2.

3.1 Basis functions

This section gives a description of the three basis functions explored in this work.

Trigonometric The trigonometric expansion is one of the most popular basis functions for FLANN. Trigonometric orthonormal basis, *sin* and *cos* functions are used for this expansion. According to [3], a trigonometric basis function forms a more compact representation than the other functions. For all the polynomials of the n^{th} order with respect to an orthogonal system, $\varphi(u)_{i=1}^{n}$, the best approximation in the metric space of L^2 is given by the n^{th} partial sum of its Fourier series with respect to the system. The expansion representation with input feature, X_i, used for the current work is as follows:

$$X_i' = \{X_i, \sin(\pi X_i), \cos(\pi X_i), \sin(2\pi X_i), \cos(2\pi X_i), \ldots, \sin(q\pi X_i), \cos(q\pi X_i)\} \tag{2}$$

Chebychev This series is easier to calculate in comparison to the trigonometric functions. As stated in [3], the non-linear approximation capacity of the Chebychev orthogonal polynomial is very powerful by the best approximation theory. Chebychev functions are orthogonal for the range $[-1, 1]$. The finite set of Chebychev polynomials can be generated in the following recursive way for an input feature X_i:

$$X_i' = \{1, X_i, \{2 \cdot X_i \cdot f_1(X_i) - f_0(X_i)\}, \ldots, \{2 \cdot X_i \cdot f_{q-1}(X_i) - f_{q-2}(X_i)\}\} \tag{3}$$

Power Polynomial The use of power polynomial basis functions for expansion is quite simple. Any polynomial functional form can be used for this approach. In the current work the expansion has been carried out in the following way:

$$X_i' = \{1, X_i, X_i^2, \ldots, X_i^q\} \tag{4}$$

3.2 Learning Mechanisms

The subsequent section gives a brief description of two learning techniques widely used for FLANN.

Backpropagation Backpropagation mechanism is one of the most popular weight optimization techniques for neural networks. FLANN models for single label classification [12] and multi-label classification [10] are known to use this learning technique quite often. Gradient descent provides a computationally efficient method of changing the weights in a feed forward network with differentiable activation function units to learn a training set of input-output pairs. In this work, the network weight matrix, $\boldsymbol{W}$, is initialized randomly with values close to zero. The input patterns, $\boldsymbol{X}$, are then fed to the network iteratively. A weighted sum of the functionally expanded non-linear inputs, X', are added with a bias term, b. The summed result is then subjected to an activation function $\phi(.)$ in the output neurons. The output vector, $Y'_i = [y'_{i1},\ y'_{i2}, ...,\ y'_{iC}]$ for the i^{th} pattern, representing the predicted labels, is then obtained. For N data instances or patterns, the labels are obtained by mapping the input to the C output dimensions as follows:

$$y'_c = \phi\Big(\sum_{k=1}^{D'} X'_k \cdot W_{kc} + b\Big),\ where\ 1 \leq c \leq C. \tag{5}$$

The error, E_i, is the calculated using the original set of labels, for the i^{th} pattern as:

$$E_i = \ Y_i \ - \ Y_i^{'} \tag{6}$$

The weights are then updated using gradient descent. The change in weights, $\varDelta W^t$, which is a set of weight vectors, $\varDelta W^t_{kc}$ at t^{th} iteration is given as,

$$\varDelta W^t_{kc} = \mu \cdot X_k^{'t} \cdot \delta^t, \tag{7}$$

where, μ is the learning rate and the gradient δ^t at t^{th} iteration is given as,

$$\delta^t = Y^{'} \cdot (1 - Y^{'}) \cdot E \tag{8}$$

The updated connections weights, W at the $(t+1)^{th}$ iteration is given by,

$$W^{t+1} = W^t + \varDelta W^t \tag{9}$$

At the end of learning phase, the model is able to generate class scores using Equation (5) at the output nodes in range [0, 1]. These scores are then converted to relevant labels for each data instance by using a global threshold.

Particle Swarm Optimization PSO [5] is a population based search procedure in which individuals called "particles" change their position with time, and each particle in the swarm represents a "solution" to the optimization problem. The particles fly around in a multi-dimensional search space following the personal best and the global best particle positions. As stated above, a particle referred here is a potential solution to the optimization problem. Thus, a particle, k, in our multi-label FLANN scenario is represented with two parameters, its velocity and position. The velocity, v_k, represents the movement of the particle in space towards the optimal solution. Whereas, the position of a particle is represented by the weight matrix of the FLANN, W_k, in the C-dimensional problem space. The particle position is denoted as a vector $\boldsymbol{W_k} = (W_{k1},\ W_{k2},\ ..., W_{kC})$, for the k^{th} particle out of all P randomly initialized ones in the problem space, i. e., the swarm. Each k^{th} particle maintains records of its personal best position, $\boldsymbol{W_k^L} = (W^L_{k1},\ W^L_{k2},\ ..., W^L_{kC})$, its current velocity, v_k, and its current position, W_k. In each iteration a global best $\boldsymbol{W}_g = (W_{g1},\ W_{g2},\ ..., W_{gC})$ is found based on the positions of all the members of the swarm. Any particle in the swarm then changes or moves in the space according to the information from the global best and its own personal best. In each iteration, W^L_k and W_g of the current swarm are combined with some weights γ_1 and γ_2, where γ_1 is known as the cognitive factor or the self-confidence factor and γ_2 is referred to as the social factor or the swarm confidence factor. This is done in order to adjust the velocities of the particles of the swarm.

Fig. 2 represents the movement of the particle. The particle moves in the direction of the resultant vector, $\boldsymbol{W}'_k$, w.r.t the personal and global best positions. ω represents the inertia factor, and it plays a key role in the process of providing balance between exploration and exploitation process in PSO. If the value of ω is large then PSO tends to be in the global search mode, hence, providing little resistance to the velocity. Whereas, if the value is less, then it provides greater resistance to the previous velocity of the particle, hence, tending to be in a targeted search mode.

Now, considering the multi-label classification problem at hand, the mean squared error (MSE) objective function has been selected for carrying out the PSO routine. Hence, the task is to minimize the MSE between the target output and the generated output. With each particle position $\boldsymbol{W_k}$ represented as:

$$\begin{aligned}\boldsymbol{W_k} = (&W_{11}, W_{12}, ..., W_{1C},\\ &W_{21}, W_{22}, ..., W_{2C}...,\\ &W_{D'1}, W_{D'2}, ..., W_{D'C}).\end{aligned} \tag{10}$$

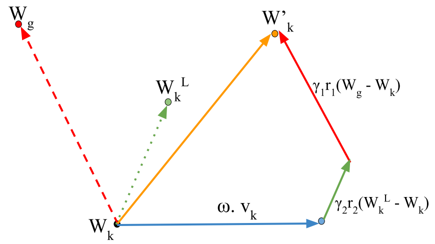

Fig. 2. Vector Representation of PSO

The objective function can be defined as:

$$J(W) = \frac{1}{D'} \sum_{i=1}^{D'} \sum_{j=1}^{C} ||\phi(W_{ij} \cdot X'_{ij} + b) - Y_{ij}||^2, \tag{11}$$

where, $\phi(.)$ is the activation function. In the current work, the activation function $\phi(x)$ is selected to be the sigmoid function. The velocity $\boldsymbol{v_k}$ and position $\boldsymbol{W_k}$ for each particle k is updated in the t^{th} iteration as,

$$\begin{aligned} \boldsymbol{v_k}(t+1) = \omega \odot \boldsymbol{v_k}(t) + \gamma_1 \odot \boldsymbol{r}_1(t) \odot (\boldsymbol{W}_k^L(t) - \boldsymbol{W}_k(t)) \\ + \gamma_2 \odot \boldsymbol{r}_2(t) \odot (\boldsymbol{W}_g(t) - \boldsymbol{W}_k(t)), \end{aligned} \tag{12}$$

$$\boldsymbol{W}_k(t+1) = \boldsymbol{W}_k(t) + \boldsymbol{v}_k(t+1), \tag{13}$$

where, $\odot$ represents the Hadamard product, $\boldsymbol{r}_1$ and $\boldsymbol{r}_2$ are vectors of random numbers which introduce randomness for search space exploitation and γ_1, γ_2 are acceleration coefficients. The personal best and global best positions for the $(t+1)^{th}$ iteration are computed as follows,

$$\boldsymbol{W}_k^L(t+1) = \begin{cases} \boldsymbol{W}_k^L(t) & \text{if } J(\boldsymbol{W}_k(t+1)) \geq J(\boldsymbol{W}_k^L(t)) \\ \boldsymbol{W}_k(t+1) & \text{otherwise} \end{cases} \tag{14}$$

$$\boldsymbol{W}_g(t+1) = \begin{cases} \boldsymbol{W}_g(t) & \text{if } J(\boldsymbol{W}_k(t+1)) \geq J(\boldsymbol{W}_g(t)) \\ \boldsymbol{W}_i(t+1) & \text{if for any } i \\ & J(\boldsymbol{W}_i(t+1)) < J(\boldsymbol{W}_g(t)) \\ & \forall\ i \in \text{swarm} \end{cases} \tag{15}$$

In the present work, the inertia factor, ω, has been changed in a linearly decreasing fashion [15], so as to employ exploration in the initial iterations and slowly convert it to exploitation when approaching the best particle. The linearly decreasing strategy for ω, that enhances the efficiency and performance of PSO, is defined as,

$$\omega(t) = \omega_{max} - \frac{\omega_{max} - \omega_{min}}{itr_{max}} \times t \tag{16}$$

where, itr_{max}, is the total number of iterations for the PSO routine. In addition, the implemented PSO also uses the adaptive cognitive acceleration coefficient (γ_1) and social acceleration coefficient (γ_2) [3]. γ_1 has been decreased from an initial value, γ_{1i}, to a final value, γ_{1f}, and γ_2 has been increased from γ_{2i} to γ_{2f} using the following equations,

$$\gamma_1^t = (\gamma_{1f} - \gamma_{1i}) \frac{t}{itr_{max}} + \gamma_{1i} \tag{17}$$

$$\gamma_2^t = (\gamma_{2f} - \gamma_{2i}) \frac{t}{itr_{max}} + \gamma_{2i} \tag{18}$$

Once the optimal weight matrix is obtained through the learning techniques, final classification is done in the testing phase. The test data is functionally expanded and the class scores are predicted through their weighted aggregation. These predicted scores are finally converted to crisp labels using a global threshold. The following section describes the experimental findings for the different multi-label FLANN models using the basis functions and learning techniques discussed here.

4 Results & Discussion

To evaluate the effectiveness of all the six multi-label FLANN models proposed in this paper, they have been tested on four datasets with ten performance metrics. Comparison among these six methods have been made to determine their effectiveness in the field of multi-label classification.

4.1 Experimental Setup

Experiments were performed on a personal computer with an Intel(R) Core(TM) i5-8265U CPU @ 1.60GHz CPU, 8GB RAM on Linux 18.04 LTS version operating system and MATLAB 2016b development environment. For evaluating all models, four real world MLDs viz. flags [7], emotions [13], yeast [6] and scene [1] were used. The MLDs can be found at http://mulan.sourceforge.net/datasets-mlc.html. Among the many metrics for multi-label classification [9], few well-known ones have been used for comparative analysis in the current work. Average precision (AvP), precision (P), recall (R), hamming loss (HLoss), one error (OE), coverage (Cov), ranking loss (RLoss), micro-F1 (mic_F1), macro-F1 (mac_F1) and subset accuracy (SA) metrics have been used.

Three basis functions – trigonometric, Chebychev and power polynomial, and two learning techniques – back-propagation and PSO were used in combination to form six FLANN models for multi-label classification. Table 1 provides the acronyms along with their corresponding basis function and learning mechanism. Out of the six mentioned here, ML-T-BP already exists in literature as MLFLANN [10]. The rest of the five models are novel in the field of multi-label classification.

Table 1. The six modelled multi-label FLANN with their corresponding basis functions and learning mechanisms

	Basis Function	**Learning Mechanism**
ML-T-PSO	Trigonometric	PSO
ML-C-PSO	Chebychev	PSO
ML-P-PSO	Polynomial	PSO
ML-T-BP	Trigonometric	BP
ML-C-BP	Chebychev	BP
ML-P-BP	Polynomial	BP

4.2 Analysis of Results

The six models for multi-label FLANN have been tested on four datasets and ten performance measures. 5-fold and 10-fold cross validation (CV) results for the datasets are shown in Tables 2 and 3 respectively. From the results throughout all the datasets and performance metrics, ML-P-PSO is seen to outperform the other methods in most of the cases. It is followed by ML-T-BP, a.k.a. MLFLANN. These two combinations of polynomial with PSO and trigonometric with backpropagation seem to perform well for multi-label classification.

Analysing each category of FLANN models for multi-label classification, we reach a few conclusions. Results for trigonometric expansion indicates a close competition between ML-T-PSO and ML-T-BP. This is very evident especially for the scene dataset, where ML-T-BP exceeds all other methods, closely followed by ML-T-PSO. This leads us to believe that trigonometric basis function definitely improves the feature space for scene data compared to the other basis functions. It forms a more compact representation of the data, making it easier to comprehend. Similarly, for power polynomial expansion, the ML-P-PSO model displayed a good performance for flags, emotions and yeast datasets. ML-P-PSO and ML-P-BP have commendable results for emotions dataset, indicating polynomial expansion fruitful for this specific dataset. Also, from the results for Chebychev expansion, ML-C-PSO is seen to perform significantly better than gradient descent optimization, ML-C-BP. However, the overall trend shows that Chebychev expansion models cannot compete with the trigonometric and power polynomial ones. The Chebychev functional expansion seems to be somewhat inadequate to bring about separability in the data. Thus, it is less suitable for multi-label classification, making trigonometric and polynomial expansions more favourable. Moving on to learning techniques, overall, PSO is seen to perform slightly better than its BP counterparts for multi-label classification. However, the choice of parameters plays a vital role in the performance of PSO. The numbers of particles that make up the swarm in the current work, has been fixed after trial and error. Larger number of particles corresponds to a much larger computation time. The optimal value to stop the PSO (itr_{max}) is also not known, hence, it was experimentally set. Keeping in mind the drastic increase in the number of dimensions after the

Table 2. 5-fold CV results for multi-label datasets

FLAGS	AvP	P	R	HLoss	OE	Cov	RLoss	mic_F1	mac_F1	SA
ML-T-PSO	0.7848	**0.6319**	0.6467	**0.2807**	0.2814	3.9290	0.2504	**0.7101**	**0.6293**	**0.1594**
ML-C-PSO	0.7632	0.5235	0.5946	0.2998	0.2448	4.2800	0.3280	0.6797	0.5252	0.0928
ML-P-PSO	**0.7976**	0.5906	**0.6678**	0.2851	0.2499	**3.8057**	**0.2309**	0.6992	0.6023	0.1390
ML-T-BP	0.7696	0.6228	0.6291	0.3051	0.2707	4.0323	0.2739	0.6836	0.6188	0.1286
ML-C-BP	0.6695	0.4960	0.4036	0.3754	**0.2132**	4.7457	0.5289	0.6151	0.4100	0.0874
ML-P-BP	0.7503	0.4641	0.5728	0.3106	0.2189	4.4834	0.3654	0.6536	0.4661	0.0930

EMOTIONS	AvP	P	R	HLoss	OE	Cov	RLoss	mic_F1	mac_F1	SA
ML-T-PSO	0.7612	**0.6547**	0.5977	0.2392	0.3289	2.0153	0.2005	0.6305	0.6206	0.2075
ML-C-PSO	0.7553	0.5655	0.6336	0.2314	0.3458	2.0253	0.2109	0.6091	0.5912	0.2310
ML-P-PSO	**0.7814**	0.6034	**0.6654**	**0.2130**	**0.2850**	1.9156	**0.1815**	**0.6437**	**0.6265**	**0.2463**
ML-T-BP	0.7619	0.6205	0.6144	0.2381	0.3423	1.9678	0.1979	0.6188	0.6133	0.2124
ML-C-BP	0.6100	0.3144	0.4610	0.3210	0.5245	**3.0389**	0.4513	0.3897	0.3499	0.0877
ML-P-BP	0.7690	0.6262	0.6343	0.2265	0.3171	2.0170	0.2023	0.6356	0.6262	0.2294

SCENE	AvP	P	R	HLoss	OE	Cov	RLoss	mic_F1	mac_F1	SA
ML-T-PSO	0.8320	0.6446	0.7055	0.1143	0.2759	0.5958	0.1015	0.6649	0.6715	0.4857
ML-C-PSO	0.7055	0.4306	0.6112	0.1515	0.4474	1.1836	0.2185	0.4995	0.5000	0.3212
ML-P-PSO	0.8012	0.5768	0.6868	0.1242	0.3149	0.7537	0.1320	0.6204	0.6237	0.4549
ML-T-BP	**0.8451**	**0.7153**	**0.7377**	**0.1014**	**0.2534**	**0.5629**	**0.0942**	**0.7129**	**0.7250**	**0.5596**
ML-C-BP	0.4391	0.0959	0.2134	0.2077	0.7852	2.5148	0.6799	0.1429	0.1180	0.0719
ML-P-BP	0.7808	0.5766	0.6612	0.1295	0.3390	0.8284	0.1527	0.6124	0.6129	0.4375

YEAST	AvP	P	R	HLoss	OE	Cov	RLoss	mic_F1	mac_F1	SA
ML-T-PSO	0.6344	0.4375	0.3711	0.3144	0.3670	8.3231	0.2864	0.5222	0.3913	0.0463
ML-C-PSO	0.7069	0.3257	0.4105	0.2315	0.2822	7.3090	0.2155	0.5849	0.3454	0.0861
ML-P-PSO	**0.7213**	0.3490	**0.4488**	0.2240	**0.2669**	**7.0632**	**0.2014**	0.6016	0.3714	0.1026
ML-T-BP	0.6454	**0.4406**	0.3832	0.3073	0.3583	8.3078	0.2891	0.5286	**0.4034**	0.0501
ML-C-BP	0.6083	0.2396	0.3195	0.2492	0.4382	10.3469	0.4451	0.5191	0.2513	0.0203
ML-P-BP	0.7074	0.3745	0.4467	**0.2238**	0.2784	7.9258	0.2431	**0.6099**	0.3957	**0.1117**

Table 3. 10-fold CV results for multi-label datasets

FLAGS	AvP	P	R	HLoss	OE	Cov	RLoss	mic_F1	mac_F1	SA
ML-T-PSO	0.7863	**0.6462**	**0.6416**	0.2834	0.2696	3.9326	0.2493	0.7129	**0.6325**	**0.1450**
ML-C-PSO	0.7681	0.4919	0.5307	0.3047	0.2343	4.1932	0.3241	0.6712	0.4855	0.0882
ML-P-PSO	**0.7959**	0.6155	0.6296	**0.2766**	0.2538	**3.8174**	**0.2344**	**0.7132**	0.6065	0.1137
ML-T-BP	0.7693	0.6169	0.6116	0.3136	0.2702	4.0061	0.2763	0.6773	0.6000	0.1342
ML-C-BP	0.7044	0.3880	0.3544	0.3738	0.2861	4.9176	0.5111	0.5616	0.3281	0.0413
ML-P-BP	0.7255	0.4918	0.5797	0.3120	**0.2133**	4.5521	0.3781	0.6604	0.4875	0.0671

EMOTIONS	AvP	P	R	HLoss	OE	Cov	RLoss	mic_F1	mac_F1	SA
ML-T-PSO	0.7476	0.6188	0.5686	0.2556	0.3561	2.0279	0.2087	0.6034	0.5855	0.1801
ML-C-PSO	0.7543	0.5353	0.6072	0.2432	0.3491	2.0388	0.2116	0.5820	0.5575	0.1736
ML-P-PSO	**0.7856**	0.6164	**0.6549**	**0.2136**	**0.2968**	**1.8734**	**0.1809**	**0.6450**	**0.6262**	**0.2581**
ML-T-BP	0.7599	0.6216	0.5901	0.2496	0.3172	2.0328	0.2051	0.6088	0.5965	0.1838
ML-C-BP	0.6203	0.3138	0.5110	0.3153	0.5140	2.8734	0.4231	0.3945	0.3526	0.0828
ML-P-BP	0.7640	**0.6219**	0.6266	0.2305	0.3340	1.9344	0.1946	0.6305	0.6131	0.2379

SCENE	AvP	P	R	HLoss	OE	Cov	RLoss	mic_F1	mac_F1	SA
ML-T-PSO	0.8285	0.6503	0.6941	0.1170	0.2771	0.6220	0.1060	0.6607	0.6685	0.4853
ML-C-PSO	0.7058	0.4281	0.6252	0.1491	0.4420	1.2002	0.2198	0.5013	0.5023	0.3299
ML-P-PSO	0.7893	0.5525	0.6933	0.1260	0.3361	0.7848	0.1390	0.6053	0.6096	0.4375
ML-T-BP	**0.8448**	**0.7216**	**0.7360**	**0.1015**	**0.2538**	**0.5650**	**0.0943**	**0.7132**	**0.7256**	**0.5525**
ML-C-BP	0.4332	0.1132	0.2768	0.2112	0.7956	2.5162	0.6890	0.1556	0.1373	0.0760
ML-P-BP	0.7903	0.5847	0.6784	0.1267	0.3294	0.8006	0.1467	0.6197	0.6231	0.4553

YEAST	AvP	P	R	HLoss	OE	Cov	RLoss	mic_F1	mac_F1	SA
ML-T-PSO	0.6324	**0.4410**	0.3773	0.3137	0.3873	8.2594	0.2832	0.5230	0.3963	0.0443
ML-C-PSO	0.7069	0.3178	0.4057	0.2317	0.2759	7.3282	0.2154	0.5820	0.3421	0.0823
ML-P-PSO	**0.7262**	0.3422	0.4481	**0.2219**	**0.2668**	**6.9045**	**0.1954**	0.6023	0.3711	0.0985
ML-T-BP	0.6503	0.4405	0.3845	0.3032	0.3517	8.2581	0.2850	0.5335	**0.4040**	0.0567
ML-C-BP	0.6198	0.2524	0.3090	0.2429	0.4240	10.2136	0.4308	0.5383	0.2676	0.0310
ML-P-BP	0.7063	0.3831	**0.4556**	0.2226	0.2822	7.9551	0.2439	**0.6131**	**0.4040**	**0.1187**

functional expansion, which thereby increases the overall computation time. However, experimentally, BP was faster than PSO, while compensating slightly on its performance. Thus, both BP and PSO can be used in multi-label FLANN depending on the requirement of the problem.

5 Conclusion

This paper develops six models (five novel and one existing) of functional link artificial neural network for multi-label classification. These six models were created by pairing three functional expansion units – trigonometric, Chebychev and power polynomial, and two learning procedures – backpropagation and particle swarm optimization. All the combination models have been tested on four multi-label datasets to generate concluding results. It was found that few models that were successful in single-label classification, have not been able to handle multi-label classification equally well. Whereas, some techniques have surpassed the others in almost every scenario, making them suitable multi-label classifiers. In future, the winning classifiers can be further explored to improve their performance in the field of multi-label classification.

References

1. Boutell, M.R., Luo, J., Shen, X., Brown, C.M.: Learning multi-label scene classification. In: Pattern Recognition. vol. 37, pp. 1757–1771 (2004)
2. Cover, T.M.: Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition. IEEE Transactions on Electronic Computers (3), 326–334 (1965)
3. Dehuri, S., Cho, S.B.: A comprehensive survey on functional link neural networks and an adaptive PSO–BP learning for CFLNN. Neural Computing and Applications **19**(2), 187–205 (2010)
4. Dehuri, S., Roy, R., Cho, S.B., Ghosh, A.: An improved swarm optimized functional link artificial neural network (ISO-FLANN) for classification. Journal of Systems and Software **85**(6), 1333–1345 (2012)
5. Eberhart, R., Kennedy, J.: Particle swarm optimization. In: Proceedings of the IEEE International Conference on Neural Networks. vol. 4, pp. 1942–1948 (1995)
6. Elisseeff, A., Weston, J.: A kernel method for multi-labelled classification. In: Advances in Neural Information Processing Systems. pp. 681–687 (2002)
7. Goncalves, E.C., Plastino, A., Freitas, A.A.: A genetic algorithm for optimizing the label ordering in multi-label classifier chains. In: 25th International Conference on Tools with Artificial Intelligence. pp. 469–476. IEEE (2013)
8. Haykin, S.: Neural Networks: A Comprehensive Foundation. Prentice-Hall of India, New Delhi (2008)
9. Herrera, F., Charte, F., Rivera, A.J., Del Jesus, M.J.: Multilabel Classification. Springer (2016)
10. Law, A., Chakraborty, K., Ghosh, A.: Functional link artificial neural network for multi-label classification. In: International Conference on Mining Intelligence and Knowledge Exploration. pp. 1–10. Springer (2017)
11. Law, A., Ghosh, A.: Multi-label classification using a cascade of stacked autoencoder and extreme learning machines. Neurocomputing **358**, 222–234 (2019)
12. Misra, B.B., Dehuri, S.: Functional link artificial neural network for classification task in data mining. Journal of Computer Science **3**(12) (2007)
13. Trohidis, K., Tsoumakas, G., Kalliris, G., Vlahavas, I.P.: Multi-label classification of music into emotions. In: 9th International Conference on Music Information Retrieval. vol. 8, pp. 325–330 (2008)
14. Venkatesan, R., Er, M.J.: Multi-label classification method based on extreme learning machines. In: 13th International Conference on Control Automation Robotics & Vision (ICARCV). pp. 619–624. IEEE (2014)
15. Xin, J., Chen, G., Hai, Y.: A particle swarm optimizer with multi-stage linearly-decreasing inertia weight. In: International Joint Conference on Computational Sciences and Optimization. vol. 1, pp. 505–508. IEEE (2009)
16. Zhang, M.L.: ML-RBF: RBF neural networks for multi-label learning. Neural Processing Letters **29**(2), 61–74 (2009)
17. Zhang, M.L., Zhou, Z.H.: Multilabel neural networks with applications to functional genomics and text categorization. IEEE Transactions on Knowledge and Data Engineering **18**(10), 1338–1351 (2006)