Classifying Frog Species by Applying Multiple Advanced Machine Learning Classification Methods

Shidong Pan[1]
[1] Research School of Computer Science, Australian National University
u6342277@anu.edu.au

Abstract. The combination between machine learning techniques and biology subjects gets closer day by day. With the developing of machine learning methods, building an automatically recognition or classification model is being attached importance by more biologists. In this paper, specific reasons are stated about why k-nearest neighbour model is the best performance (98.46%) and the comparison with two kinds of neural networks and support vector machine(SVM) models on this dataset, with the applying of k-cross validation method. Moreover, two data processing methods: inputs reduction and pattern reduction are introduced and practiced on this dataset.

Keywords: Machine learning, neural network, deep learning, frog species, classification, k-cross validation, k-nearest neighbour, data processing

1 Introduction

Depending on the development of machine learning techniques, more and more machine learning techniques are applied in many fields. Especially in biology subject, thanks to the accuracy and reliability of machine learning tools has increased to a practical level, it has become a greatly active topic [1][2]. Interestingly, the earliest build of machine learning model was inspired by the biology structure and now, biologists use well-performed and high-accurate machine learning model to detect animals' movement, analyze their behavior pattern and even automatically classify animals by computers. As a new kind of authentication method, biometric identification technology has been recognized by both academic and business circles since the mid 1990s. With the hardworking by the academia and business counterparts around the world over the years, biometrics research has made great progress. For example, according to Norouzzadeh et al (2017), they developed a system classifier by machine learning techniques to identify animals automatically, the accuracy of this model comes to 93.8% and it will "rapidly improved" in the future [2]. More biometric identification technologies effectively applied in many fields. The performance of current machine learning models has increased to a very high level. For instance, a cattle identification experimental results which based on machine learning tools showed that the proposed approach achieved a promising accuracy result (approximately 99.5%) [3].

On the other hand, the imperfect performance of the existing biometric classification system has greatly influenced the popularization and application this innovation. How to further improve the performance of the system so that it can better meet the needs of practical application is still a challenging and worthy research question. To gain a better understand of natural eco-systems, it's crucial to collect specific, large-scale knowledge about the quantity, habitats, and behaviors of animal in natural ecosystems. With the adoption of machine learning methods, people can identify different animals' voice without human intervention, which is called un-supervised learning. In this case, some machine learning models will be applied into classifying different frog species through their calls without human intervention and all those models would be compared to get a best model for this task. The general framework for recognizing frog species, basing on their calls, is shown in Fig1. In this paper, we will focus on the Classification step.

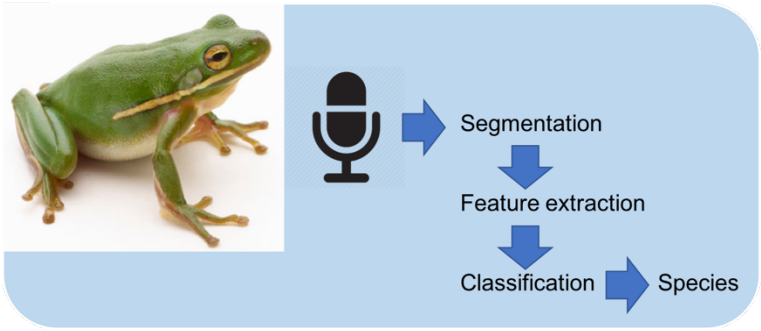

Fig. 1. General framework for recognizing frog species

2 Method

2.1 Dataset

According to the contributor of the dataset, it was used in several classifications tasks related to the challenge of anuran species recognition. There are 22 features and 7195 instances of the dataset and it is a multi-label dataset with three columns of labels: families, genus and species, plus a record ID as an extra column. For the 22 features, each one is a Mel-frequency cepstral coefficient that is transformed from a special syllable.

Specifically, in this paper, we only focus on the species instead of families or genus. So that the columns of families, genus and record ID have been removed in order to make data cleaner. Additionally, dataset is manually randomly separated into a training set which contains 5700 instances, about 80 percent of whole dataset, and a test set contains 1495 instances.

Table 1. Different species and their number in whole dataset.

Species	Amount
AdenomeraAndre	672
AdenomeraHylaedactylu	3478
Ameeregatrivittata	542
HylaMinuta	310
HypsiboasCinerascens	472
HypsiboasCordobae	1121
LeptodactylusFuscus	270
OsteocephalusOophagus	114
Rhinellagranulosa	68
ScinaxRuber	148

The dataset is from UCI Machine Learning Repository [4].

2.2 Machine Learning Methods

The capabilities of PyTorch is very comprehensive, implementing the Tensor class. Additionally, many functions are also provided in order to initialize and manipulate tensors in a concise fashion and lots of practical packages can be directly used in code which can save lots of time. To quickly build a neural network(NN) for this task, we use the nn package to define our model as a sequence of layers. The nn.Sequential is a module which contains other Modules, and applies them in sequence to produce its output. Each linear module computes output from input using a linear function and holds internal Variables for its weight and bias. Apart from that, deep learning methods are used to build multiple hidden layers NN, comparing the performance with the single hidden layer NN, to gain a better performance. Also, we tuned different parameters to figure out the best parameters set for NN for this task.

Furthermore, in python code document, two more machine learning methods are applied in this work, following as Support Vector Machine (SVM) and k-Nearest-Neighbors(kNN) algorithm.

Additionally, the classification performance and the generalization capabilities of the system are normally evaluated by using Cross-Validation (k-CV) [5]; therefore, we also used k-CV methods to estimate outcome aiming to avoid overfitting.

2.3 Data Processing Methods

Except seeking for the best machine learning model, data processing methods can be applied on dataset before it is input into model. Generally, appropriate data pre-processing can increase the performance of machine learning models [6][7]. Therefore, in this paper, two data processing methods will be attempted to see the change of performance.

Hence, input pruning and pattern reduction methods will be illustrated and applied on this dataset in the following parts.

3 Results and Discussion

3.1 Machine Learning Methods

3.1.1 NN

Firstly, I tried to used nn.package in python to build a simple neural network model to classify frog species to see the outcomes. In the neural network, 22 input neurons and 10 output neurons are set due to 22 input features and 10 classifications respectively.

There are several parameters in building neural network. Number of hidden neurons consists the basic structure of the whole network. Epochs is the maximum number of iterations what user input and the learning rate can update every gradient parameter in the opposite direction.

Mini-batch gradient descent method learns every time by using the whole training set, so its advantage is that each update will take place in the right direction. Finally, it can guarantee convergence in extreme value point (convex function converges to the global extreme value point and non-convex function may converge to local extremum value points). On the contrary, its drawback is that each learning time is too long, and if the training set is very big, it will take a lot of memory and time. Besides, the gradient descent cannot be updated model parameters online. Due to the dataset does not have abundant samples, we make the dataset repeated epochs in order to make fully use of data and get a better outcome.

Moreover, stochastic gradient descent (SGD) optimizer is used in this neutral network. SGD is the most common optimization method for calculating the gradient of mini-batch per iteration and then updating the parameters, the specific calculation is shown in Fig2. gt is gradient, η is learning rate and $\Delta\theta t$ is the vector of direction opposite the gradient gt with magnitude the norm of gt times the learning rate η.

$$g_t = \nabla_{\theta_{t-1}} f(\theta_{t-1})$$

$$\Delta\theta_t = -\eta * g_t$$

Fig.2. Update computing method of SGD

The only problem of this optimizer is it is a little difficult to find an appropriate learning rate and epoch times and here is the accuracy table of different epoch numbers and learning rate in single hidden layer NN:

Table 2. Accuracy of different epoch times and learning rate of single hidden layer NN.

Learning rate	Epoch times	Accuracy
0.01	100	50.37%
0.01	300	50.37%
0.1	100	73.65%
0.1	300	84.21%
0.9	100	89.97%
0.9	300	95.25%

The table illustrates that in this case, higher learning rate and more epoch times can lead to a better accuracy which is quite good. However, deep learning allows computational models composing multiple hidden(processing) layers to learn representations of data with multiple levels of abstraction by using the backpropagation algorithm to indicate how a machine should change its internal parameters. Deep learning methods can dramatically improve the performance of NN [8]. This performance in above table just comes from a single hidden layer NN, “Hidden layers somehow twist the problem in a way that makes it easy for the neural network to classify the problem or pattern” [9]. Hence there is an assumption that if we apply deep learning methods as well as adding more hidden layers into network and then tuning the parameters of this NN, it may lead to a better result.

Table 3. Summary of performance of two hidden layers NN.

Number of hidden neurons in first hidden layer	Number of hidden neurons in second hidden layer	Epoch times	Learning rate	Accuracy
40	100	1000	0.1	93.16%
40.	100	1800	0.1	95.32%
40	100	2100	0.1	94.52%
60	120	1000	0.1	93.65%
60	120	1800	0.1	95.45%
60	120	1800	0.2	95.12%
60	120	1000	0.15	95.12%
60	120	3000	0.1	9.23%
50	100	1800	0.1	94.92%

From Table 3, we can see that different parameters slightly affect the performance of this model, and the best accuracy is 95.45% which exceeds 0.2% compared with the single hidden layer NN. Although the outcome does not have a distinctive difference, the learning process differs greatly. In single hidden layer NN, with the epoch increasing, accuracy does not gradually increase, but from a low level suddenly reach a high level, stably staying at that level. On the contrary, in the latter model, we can clearly observe the learning process which means the accuracy constantly increase with more and more epochs. But the model was trained by training set too many times, the accuracy will decrease to an extremely low standard. In conclude, the highest accuracy of NN is 95.45%.

3.1.2 SVM and kNN

Apart from NN, there are many other machine learning methods. For this dataset, Colonna et al (2016) summarized some outcomes of related work which is shown in Table 4.

Table 4. Summary of few related works. The **#** stands for the number of different frog species, **ML** for Machine Learning Algorithm.

Author	#	ML	Accuracy
Colonna et al.	9	kNN, SVM	97%
Huang et al.	5	kNN, SVM	100%
Jaafar et al.	28	kNN, SVM	98%
Xie et al.	4	GMM	90%
Dayou et al.	9	kNN	90%
Han et al.	9	kNN	100%
Vaca-Castaño	20	kNN	91%
Yuan	8	kNN	98%

From these related work, we can easily draw the conclusion that kNN and SVM methods can obtain a higher accuracy in some cases than our NN model.

There are several benefits of SVM:

1. Solve the problem of machine learning in a small dataset
2. Can solve non-linear problems
3. No local minimum value problem (relative to neural network and other algorithms)
4. High-dimensional dataset can be handled well and
5. Strong generalization ability

As for the kNN, it is simple and easy to use and understand with a high precision as a mutual theory, as well can be both used for classification or regression. Moreover, it is available for numerical and discrete data, insensitive to outliers and less time consumption than others.

If parameters of a prediction function learning and testing it on the same dataset, it will make a methodological mistake in machine learning processing. The model will only repeat the sample label as it has seen, having a perfect score but cannot predict any invisible data (testing dataset without label). This situation is called overfitting. To avoid this problem happening, cross-validation (CV) is used to evaluate the expected error in training machine learning models. With k-CV applying on the training dataset, the original training dataset is split into k disjoint folds and use k-1 sub-folds to train the predictor and use the last sub-fold to valid the performance of this model. Thus, k-CV can effectively avoid the occurrence of overfitting, and the result is more persuasive.

Combining the two points as mentioned, we make kNN and SVM machine learning methods and k-CV validation are associated together, aiming to get a better outcome. Several comparative papers can be found in the literature, table 3 summarizes the outcome of related works.

Table 3. Summary of few related works. **ML** for Machine Learning Algorithm.

Author	ML	Accuracy
Colonna et al. [5]	kNN(k=1)	62.65%
Colonna et al. [5]	kNN(k=3)	60.28%
Colonna et al. [5]	kNN(k=5)	58.93%
Colonna et al. [10]	kNN	97.52%
Colonna et al. [11]	kNN	93.9%
Colonna et al. [11]	SVM	96.4%

In different paper, although the author slightly different features or just predict some specific target specie, the accuracies still did not reach to a positive level. Thus, I imported sklearn packages to use kNN model and SVM model, with the k-CV validation method, to practice kNN and SVM methods strictly on the dataset:

Table 4. **ML** for Machine Learning Algorithm, k-CV for k-cross validation.

ML	Accuracy
kNN(k=1)	98.39%
kNN(k=3)	98.39%
kNN(k=5)	98.39%
k-CV(k=5) for kNN(k=5)	98.46%(+/- 0.97%)
SVM	96.59%
k-CV(k=5) for SVM	96.84%(+/- 1.16%)

The performance of kNN and SVM slightly exceeded related works, it may can attribute to the different parameters or kernels of kNN and SVM. Moreover, it also better than the NN model with SGD optimizer. Therefore, for this dataset, kNN and SVM may be a better choice. As mentioned, in Table 4, the best models of this task can reach 100% accuracy in Huang and Han work, so the performance of existing model still can be increased in other way.

3.2 Data Processing Methods

3.2.1 Input Reduction

According to Feng and Brown (2000), how correlated inputs affect the output have to be considered in machine learning model building [12]. Moreover, Intuitively, highly correlative or irrelative things actually follow same regularity, hence when we build a multiple inputs model, these connected things are supposed to be avoided all to be inputs. Pearson Product-Moment Correlation Coefficient(PPMCC) is a measure of the linear correlation between two variables. It has a value between +1 and −1, where 1 is total positive linear correlation, 0 is no linear correlation, and −1 is total negative linear correlation. Hence, we use rattle package in R to calculate the PPMCC between all this 20 input attributes, deleting some extremely positive or negative attributes, expecting to get a higher accuracy of this model.

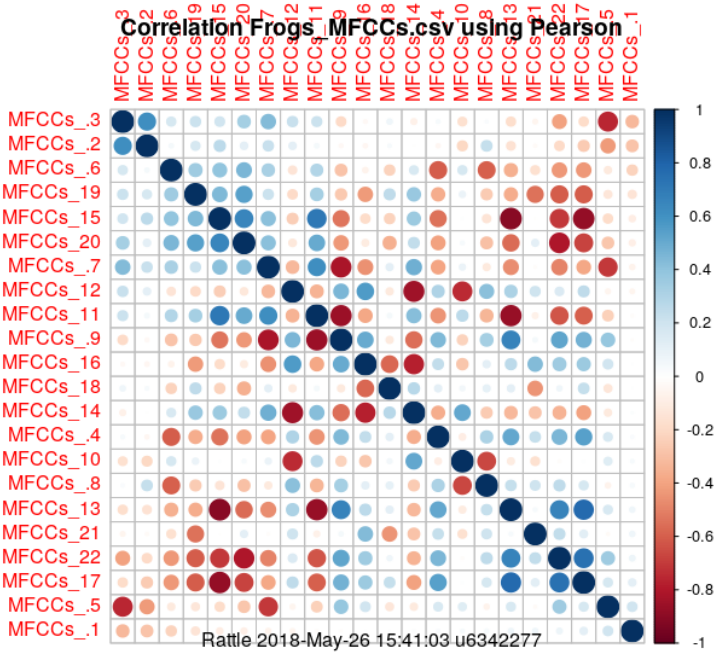

Fig. 3. PPMCC between 22 inputs

According to Fig. 3., it shows that MCFFs_13, MCFFs_17 and MCFFs_22 have more obvious points than other inputs, which means that the information from these inputs may can be replaced by other inputs. Therefore, these three inputs are deleted in dataset.

Table 5. Summary of performance of different models on reduced dataset. **ML** for Machine Learning Algorithm, k-CV for k-cross validation. (For NN, hidden neurons in first layer = 60, hidden neurons in second layer = 120, learning rate = 0.1, epoch times= 1600)

ML	Accuracy
Multiple hidden layers NN	94.58%
k-CV(k=5) for kNN(k=5)	98.25%(+/- 0.95%)
k-CV(k=5) for SVM	96.84%(+/- 1.21%)

Unfortunately, from Table 5 we can see that the accuracies are sightly decrease than keeping these three inputs.

3.2.1 Pattern Reduction

According to Gedeon and Bowden (1992) and Bustos and Gedeon (1995), a thought that some class may have too many instances while others have too few may be resulting a significant bias in the network. In this dataset, the most specie of frog is AdenomeraHylaedactylu which has 3478 samples and the fewest Rhinellagranulosa only has 68 samples. Inspired by that point, pattern reduction may can be used to improve the performance of these models in this case.

Table 6. Different species and their number in original dataset and APR dataset.

Species	Amount	Amount after Pattern reduction
AdenomeraAndre	672	500
AdenomeraHylaedactylu	3478	500
Ameeregatrivittata	542	542
HylaMinuta	310	310
HypsiboasCinerascens	472	472
HypsiboasCordobae	1121	500
LeptodactylusFuscus	270	270
OsteocephalusOophagus	114	114
Rhinellagranulosa	68	68
ScinaxRuber	148	148
Total	7195	3424

The pattern reduction process is randomly removing some instances in dataset to keep the amount difference between species smaller. After that, the dataset is re-separated into two parts: After pattern reduction(APR) training set which contains 2600 instances and APR test set which contains 824 instances.

Table 7. **ML** for Machine Learning Algorithm, k-CV for k-cross validation.

ML	Accuracy
Multiple hidden layers NN	14.93%
k-CV(k=5) for kNN	N/A
k-CV(k=5) for SVM	N/A

According to Table 7, the accuracy in multiple hidden layers NN comes to an extremely low standard, as well the kNN and SVM model cannot get the outcome. I speculate that there are two reasons. One is with the reduction of some pattern, the total amount of instances also is reduced, causing models unable to receive enough inputs to completely build models. Another is some species only have 68 instances. Although the APR dataset is randomly separated into APR training set and APR test set, it is possible to sort all this species samples into training or test set.

4 Conclusion and Future Work

In conclusion, aiming to classify frog species by computers, we totally built and trained four models: single hidden layer NN, multiple hidden layers NN, kNN and SVM; plus, one validation method: k-CV validation. All four models have a satisfying performance in the testing process, with the best accuracy respectively is 95.25%, 95.45%, 98.46% and 96.59%, which is close to the related works, even better than some. Compared with others, kNN model seems that has the best performance.

To get a better performance of NN, we have to master more optimizers to adapt different datasets and task requirements. From Asimakopoulou et al. and Alsina et al. work [15] [16], they increased performance largely by choosing appropriate optimizers and adjusting the parameters inside. Therefore, it is important for a machine learning learner to deeply understand the structure of optimizers, specifically know the parameters' functions, or it is just a black box. Additionally, hidden neuron pruning is another method to improve the performance of NN.

Apart from machine learning methods, we also apply two data processing methods: inputs reduction and pattern reduction. Although these two measures fail to gain a better accuracy, they still inspire our thought and we can continue to research these methods in future work in order to claim a better performance.

In the future, there are two directions of increasing the model of classifying different frog species: one is focusing on the features selecting and extracting; another is, on the foundation of applying appropriate validation methods, discovering more suitable machine learning methods depending on chrematistics of dataset. So that people can take effective measures to better manage wild animals and government can publish rules and laws more conveniently to protect the whole eco-system.

References

[1] Delellis, P., Polverino, G., Ustuner, G., Abaid, N., Macrì, S., Bollt, E.M. & Porfiri, M. 2014, Collective behaviour across animal species, Scientific Reports, vol. 4, pp. 3723.

[2] Norouzzadeh, M.S., Nguyen, A., Kosmala, M., Swanson, A., Palmer, M., Packer, C. and Clune, J. 2017, Automatically identifying, counting, and describing wild animals in camera-trap images with deep learning.

[3] Gaber, T., Tharwat, A., Hassanien, A.E. and Snasel, V., 2016. Biometric cattle identification approach based on weber's local descriptor and adaboost classifier. Computers and Electronics in Agriculture, 122, pp.55-66.

[4] UCI Machine Learning Repository

[5] Colonna, J.G., Gama, J. and Nakamura, E.F., 2016, September. How to correctly evaluate an automatic bioacoustics classification method. In Conference of the Spanish Association for Artificial Intelligence. pp. 37-47

[6] Qiu, J., Wu, Q., Ding, G., Xu, Y. and Feng, S., 2016. A survey of machine learning for big data processing. EURASIP Journal on Advances in Signal Processing, 2016(1), pp.67.

[7] Ma, Z., Chien, J.T., Tan, Z.H., Song, Y.Z., Taghia, J. and Xiao, M., 2017. Recent Advances in Machine Learning for Non-Gaussian Data Processing.

[8] LeCun, Y., Bengio, Y. and Hinton, G., 2015. Deep learning. nature, 521(7553), pp.436.

[9] Kang.N., Multi-Layer Neural Networks with Sigmoid Function— Deep Learning for Rookies (2), cited at 30th May at:
https://towardsdatascience.com/multi-layer-neural-networks-with-sigmoid-function-deep-learning-for-rookies-2-bf464f09eb7f

[10] Colonna.J.G., Ribas.A.D., Santos.E.M.dos. and Nakamura.E.F., 2012. Feature Subset Selection for Automatically Classifying Anuran Calls Using Sensor Networks. WCCI 2012 IEEE World Congress on Computational Intelligence, June 10-15.
[11] Colonna.J., Peet.T., Ferreira.C.A., Jorge.A.M., Gomes.E.F. and Gama.J., 2016. Automatic Classification of Anuran Sounds Using Convolutional Neural Networks. Proceedings of the Ninth International C* Conference on Computer Science & Software Engineering. pp.73-78.
[12] Feng. J. and Brown. D., 2000. Impact of Correlated Inputs on the Output of the Integrate-and-Fire Model. Neural Computation, 12(3), pp. 671-692.
[13] Gedeon, T., & Bowden, T. (1992). Heuristic Pattern Reduction. International Joint Conference on Neural Networks. International Joint Conference on Neural Networks, 2, pp.449-453.
[14]. Bustos, R.A. and Gedeon, T.D., 1995. Decrypting Neural Network Data: A GIS Case Study. In Artificial Neural Nets and Genetic Algorithms pp. 231-234.
[15] Asimakopoulou, G.E., Kontargyri, V.T., Tsekouras, G.J., Asimakopoulou, F.E., Gonos, I.F. and Stathopulos, I.A., 2009. Artificial neural network optimisation methodology for the estimation of the critical flashover voltage on insulators. IET Science, Measurement & Technology, 3(1), pp.90-104.
[16] Alsina, E.F., Bortolini, M., Gamberi, M. and Regattieri, A., 2016. Artificial neural network optimisation for monthly average daily global solar radiation prediction. Energy conversion and management, 120, pp.320-329.