Different Methods to Classify Animals

1. Problem

In this project, I have a dataset consists of 101 animals from a zoo and 16 variables with various traits to describe the animals. I plan to classify these animals into 7 class types: mammal, bird, reptile, fish, amphibian, bug and invertebrate. Different classification methods will be used, they are logistic regression, k-nearest neighbors, random forest, and neutral network. And I want to examine three problems in particular:

(1) Do these classification methods work well on this zoo animal data set?
(2) Which classification methods work better?
(3) How do the results change with different parameters in the methods?

2. Data Description

This dataset consists of 101 animals from a zoo. There are 16 categorial variables with various traits to describe the animals. Among these variables, 15 are binary variables, such as hair, feathers, eggs. They describe whether the animal has this type of trait. 0 represents that the animal doesn't have this trait and 1 represents that the animal has this trait. There one polytomous variables, legs, describes the number of legs the animal has. In the original data set, the animal names and class types are also included in the columns, for the 'class types' variable, 1 to 7 represents mammal, bird, reptile, fish, amphibian, bug and invertebrate respectively.

I explore some statistical characters of this data set. First, I check the missing values in the data set. Fortunately, there is no missing values in this data set.

animal_name	hair	feathers	eggs	milk	airborne	aquatic	predator
0	0	0	0	0	0	0	0
toothed	backbone	breathes	venomous	fins	legs	tail	domestic
0	0	0	0	0	0	0	0
catsize	class_type						
0	0						

Table 1. The number of missing values in each column.

Next, I make a bar plot to visualize the number of animals in each class type.

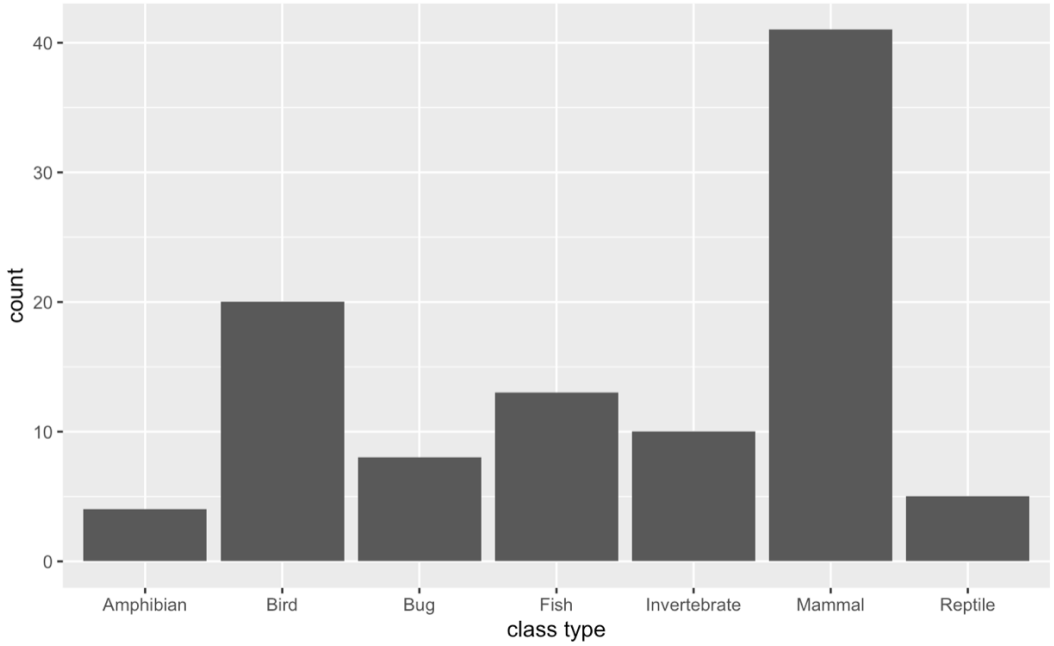

Figure 1. The bar plot of class type.

The plot shows that mammal has the more animals far more than other class type and amphibian has the smallest number of animals.

Then I calculate the mean of each variable in each class type. A lager mean indicates that more animals have this trait in this class type.

class_type	hair	feathers	eggs	milk	airborne	aquatic	predator	toothed	backbone	breathes	venomous	fins	legs	tail	domestic	catsize
1	0.95122	0.0	0.02439	1.0	0.04878	0.146341	0.536585	0.97561	1.0	1.0	0.000000	0.097561	3.365854	0.853659	0.195122	0.780488
2	0.00000	1.0	1.00000	0.0	0.80000	0.300000	0.450000	0.00000	1.0	1.0	0.000000	0.000000	2.000000	1.000000	0.150000	0.300000
3	0.00000	0.0	0.80000	0.0	0.00000	0.200000	0.800000	0.80000	1.0	0.8	0.400000	0.000000	1.600000	1.000000	0.000000	0.200000
4	0.00000	0.0	1.00000	0.0	0.00000	1.000000	0.692308	1.00000	1.0	0.0	0.076923	1.000000	0.000000	1.000000	0.076923	0.307692
5	0.00000	0.0	1.00000	0.0	0.00000	1.000000	0.750000	1.00000	1.0	1.0	0.250000	0.000000	4.000000	0.250000	0.000000	0.000000
6	0.50000	0.0	1.00000	0.0	0.75000	0.000000	0.125000	0.00000	0.0	1.0	0.250000	0.000000	6.000000	0.000000	0.125000	0.000000
7	0.00000	0.0	0.90000	0.0	0.00000	0.600000	0.800000	0.00000	0.0	0.3	0.200000	0.000000	3.700000	0.100000	0.000000	0.100000

Table 2. The mean of each variable in each class type.

Form the table, we can tell that if an animal has milk, then it belongs to mammal. If an animal has feathers, then it belongs to bird.

Be curious about the relationship between legs and class type, I also make a bar plot to take a quick look.

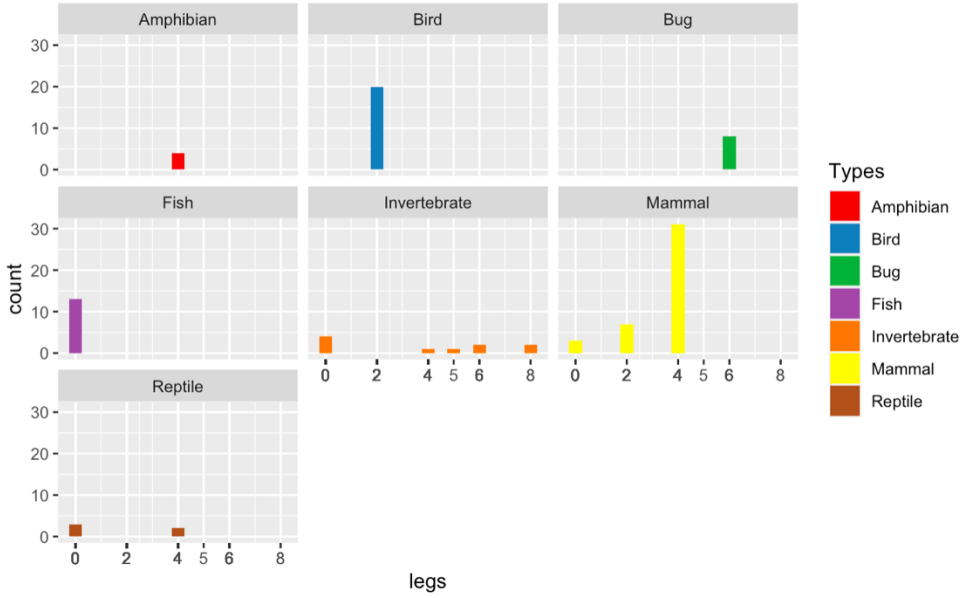

Figure 2. The number of legs

This plot shows the number of animals that have different number of legs in each class type. And it looks like most mammals have four legs and all the birds in this data set have three legs.

Then I take a visual evaluation of correlations between 16 variables.

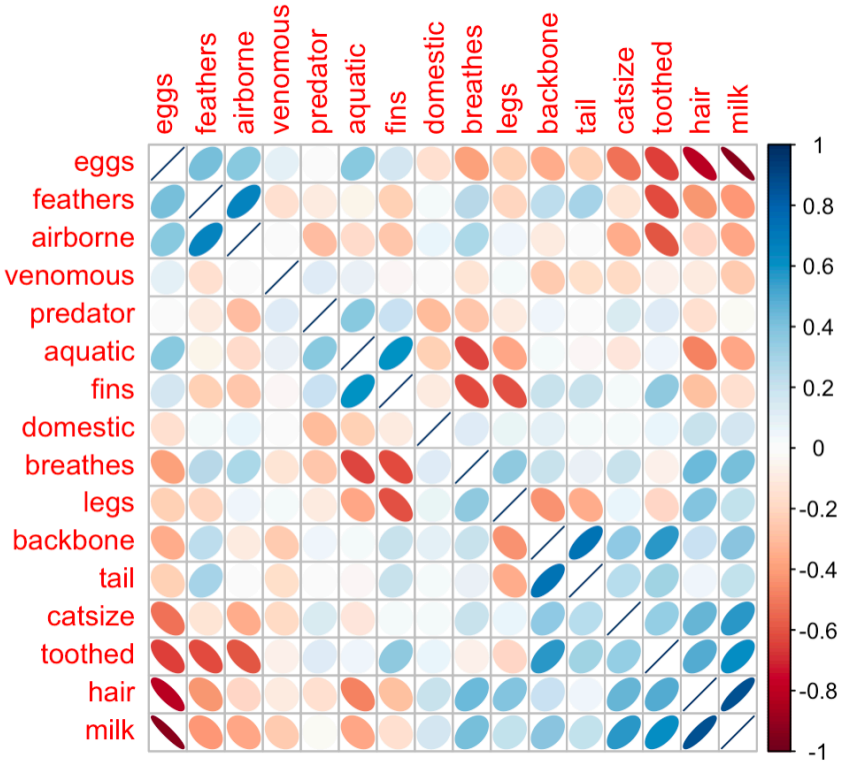

Figure 3. Correlations between 16 variables.

In general, correlations are relatively low. But there are still some variables have very high correlation, for example, eggs and milk. It makes sense intuitively, animals that secrete milk to feed the next generation will not reproduce by laying eggs.

After learning and viewing some statistical characters of this dataset, I have a better understanding of the data set and it is helpful for the following analysis.

3.Model

I use four different methods to perform classification. They are logistic regression model, k-nearest neighbors model, random forest model, and neural network with or without hidden layer.

3.1 Logistic Regression

Logistic regression is a linear regression model of the log odds:

$$logit\big(p(x)\big) = \beta_0 + \beta_1 x$$

$p(x)$ is the conditional probability that data x belongs to the class $Y = 1$

$$p(Y = 1|\, X = x) = p(x)$$

$$p(Y = 0|\, X = x) = 1 - p(x)$$

It has the equivalent formulation as:

$$p(x) = \frac{e^{\beta_0+\beta_1 x}}{1 + e^{\beta_0+\beta_1 x}} = logistic(X^T\beta) = softmax(X^T\beta)$$

$$softmax(\,\vec{z}\,)_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}$$

In this zoo animal data set, I have class types that more than 2 classes, so I can use multinomial logistic regression that extends the logistic regression model to $K \geq 2$ classes.

$$\log\left(\frac{P(Y = k \mid X = x)}{P(Y = 0 \mid X = x)}\right) = X^T\beta_k\,, k = 1,2, \dots \dots, K-1$$

$$P(Y = k|\, X = x) = \frac{\exp\,(X^T\beta_k)}{1 + \sum_{l=1}^{K-1} \exp\,(X^T\beta_l)}$$

In this case, I can get the prediction of Y by assigning the sample to the class with the highest probability and that is the classification.

3.2 K-Nearest Neighbors

K-Nearest Neighbors is a non-parametric classification method, an unlabeled sample is classified by assigning the label which is most frequent among the k training samples nearest to that sample. The parameter k is how the model is trained, instead of a

parameter that is learned through training. The conditional probability that data x belongs to the class $Y = m$ is:

$$\hat{P}_k(Y = m|\, X = x) = \frac{1}{k} \sum_{i \in \mathcal{N}_k(x,D)} \mathrm{I}\,(y_i = m)$$

As the same, I can classify the sample to the class with the highest estimated probability.

$$\hat{C}_k(x) = \underset{m}{argmax}\ \hat{P}_k(Y = m|\, X = x)$$

3.3 Random Forest

The algorithm of random forest is shown as the following:
For $b = 1\ to\ B$, first draw a bootstrap sample of size n from the training data, then grow a random-forest tree T_b to the bootstrapped data, recursively repeating following steps, until minimum node size reached: (1) select m variables at random from the p variables; (2) pick the best split-point among the m; (3) split the node into two children nodes. And output the ensemble of trees $\{T_b{}_{b=1}^{B}\}$. To make a prediction at a new point x, taking the majority vote of the individual trees.

3.4 Neural Network

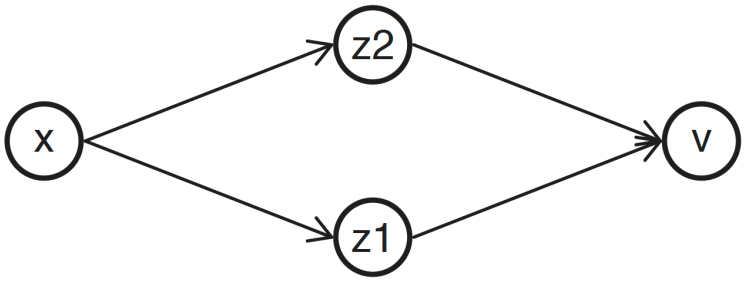

Figure 4. A simple neural network

The **Figure 4** shows a simple neural network with one input, one hidden layer with two hidden nodes and one output. The relationship between them can be written as the following formulations:

$$z_1 = b_1 + x \cdot w_1$$
$$z_2 = b_2 + x \cdot w_2$$
$$z_3 = b_3 + z_1 \cdot u_1 + z_2 \cdot u_2 = b_3 + \sigma(z_1) \cdot u_1 + \sigma(z_2) \cdot u_2$$

σ is an activation function. For classification tasks, the activation function used is the $softmax$ function, which is also used in logistic regression model. A vector y of integer coded classes should be converted into a matrix Y containing indicator variables for each class:

$$\begin{pmatrix} 2 \\ 4 \\ \vdots \\ 1 \end{pmatrix} \rightarrow \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Figure 5. A one-hot encoding

Now, values from the neural network in the output layer can be regarded as probabilities over the possible categories. Neural networks generally need a large number of iterations to converge to a reasonable minimizer of the loss function. For multi-class classification tasks, the loss function used is categorical cross-entropy:

$$f(y, \hat{y}) = -\sum_{k} y_k \cdot \log(\hat{y}_k)$$

Perceptron is an algorithm used for supervised learning of binary classifiers. It is a single-layer neural network.

$$f(x) = \begin{cases} 1, & if \sum_{i=1}^{m} w_i x_i + b > 0 \\ 0, & otherwise \end{cases}$$

where w_i are real-valued weights, m is the number of inputs to the perceptron, and b is the bias.

4. Results

I first perform principal components analysis on the zoo animal data,

```
Importance of components:
                          PC1    PC2    PC3     PC4     PC5     PC6     PC7     PC8     PC9    PC10
Standard deviation     2.1612 1.8279 1.5377 1.10948 0.97670 0.86373 0.75021 0.71584 0.66917 0.61632
Proportion of Variance 0.2919 0.2088 0.1478 0.07693 0.05962 0.04663 0.03518 0.03203 0.02799 0.02374
Cumulative Proportion  0.2919 0.5007 0.6485 0.72546 0.78508 0.83170 0.86688 0.89891 0.92689 0.95063
                          PC11    PC12   PC13    PC14    PC15    PC16
Standard deviation     0.52686 0.45905 0.3599 0.34343 0.19045 0.13335
Proportion of Variance 0.01735 0.01317 0.0081 0.00737 0.00227 0.00111
Cumulative Proportion  0.96798 0.98115 0.9892 0.99662 0.99889 1.00000

                 PC1          PC2         PC3         PC4          PC5         PC6         PC7
hair     -0.40781682  0.113787661 -0.10005203  0.05530359 -0.009809592  0.06161242 -0.07515992
feathers  0.20156822  0.192502301  0.48301504 -0.17809415 -0.036434527 -0.19625267 -0.02564308
eggs      0.43158652  0.005130841  0.04178035  0.01426172  0.091265259 -0.05356323 -0.02890185
milk     -0.44479630 -0.001635436 -0.01267313 -0.03047774  0.056347892  0.01613469  0.01046026
airborne  0.19541969  0.311645431  0.27830433  0.04158989 -0.074017532  0.09999599 -0.10959835
aquatic   0.20486980 -0.373556142 -0.06308768 -0.06944122  0.060612264 -0.19214372 -0.02543174
predator  0.02901627 -0.252801176 -0.11773037 -0.57821523 -0.226856601 -0.34599871  0.46764905
toothed  -0.32298104 -0.320714639 -0.02356268  0.16552481 -0.094193168  0.19945655  0.19723504
backbone -0.20544385 -0.204605985  0.47642004 -0.01610225 -0.132586152  0.01469534  0.15645504
breathes -0.21041493  0.366020432  0.18971112 -0.16697296 -0.113468339  0.09901229  0.03535602
venomous  0.09561801 -0.002572422 -0.22627612  0.08610457 -0.899367272  0.01426752 -0.30043464
fins      0.08175466 -0.454630774  0.04626720  0.21557957  0.129520397  0.04415996 -0.25279402
legs     -0.11455341  0.340245130 -0.31290123 -0.16874867  0.074115877 -0.16148376  0.11487544
tail     -0.12404631 -0.160614956  0.48311936 -0.04923232 -0.194076689  0.08427094  0.04507103
domestic -0.09681845  0.117442571  0.08098942  0.62769554 -0.079154927 -0.70267677  0.23291114
catsize  -0.28884911 -0.092983107  0.06893596 -0.29812608  0.109852410 -0.45720535 -0.68783969
```

Table 3. PCA results

From the results, the first component, explaining 29% of the total variance, is positively associated with variable eggs, and negatively associated with variables: hair, milk and toothed. It may indicate the main difference between bird and mammal. But It is difficult for me to accurately summarize what information it describes. The second component, explaining 21% of the total variance, is positively associated with variables: breathes, legs, and negatively associated with aquatic, fins and toothed. It may indicate the main difference between land and aquatic animals. But I still cannot accurately summarize what information it describes.

Because a few principal components can reveal the internal structure of multiple variables, retain as much information about the original variables as possible, and they are not related to each other. I use principal components analysis to project the zoo animal data onto $k = 2,3, \ldots \ldots, 15$ principal components. For each k, I build a logistic regression model to classify the data and examine how the misclassification rate changes with the number of principal components used.

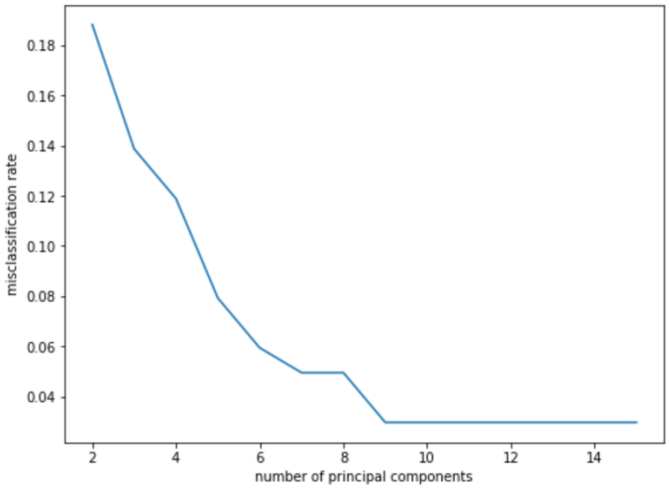

Figure 5.

As the number of principal components increases, I can explain more variation in the data and get better fitness of logistic regression model. So, I can more accurately classify the data and the misclassification rate decreases.

To compare the results between four different classification methods: logistic regression, k-nearest neighbors, random forest, and perceptron, I first divide the zoo animal data set into train data set and test data set, by randomly choosing 75 samples as the train data set. Then for each method, I fit the model based on the same train data set and use the model to predict the same test data set, then get the accuracy score as the proportion of the class type predicted for a sample exactly match the true class type. Here is the table of the accuracy score in five different train sets of 75 samples.

	1	2	3	4	5
forest	0.923077	0.961538	0.961538	0.923077	0.884615
knn	0.807692	0.923077	0.846154	0.923077	0.846154
logistic	0.884615	0.961538	0.961538	0.923077	0.884615
perceptron	0.884615	0.961538	0.884615	0.961538	0.961538

Table 4. Accuracy Score

The table shows that the accuracy scores for each method are relatively high. Among them, k-nearest neighbor seems to have a lower accuracy score than other methods.

I also conduct k-fold cross validation for the four methods. It randomly split the entire dataset into k folds, for each fold in the dataset, build the model on $k-1$ folds of the dataset. Then, test the model to check the effectiveness for the kth fold. Here is the table of the cross-validation score when $k = 5$:

	logistic	knn	forest	perceptron
0	1.000000	0.909091	0.954545	1.000000
1	0.952381	0.809524	0.952381	0.952381
2	0.904762	0.857143	0.952381	0.857143
3	0.947368	0.789474	1.000000	0.947368
4	0.944444	0.888889	0.888889	0.944444

Table 5. Cross Validation Score

The table shows that each method is very valid of classification. Among them, k-nearest neighbor seems to be less effective than other methods.

For k-nearest neighbor, I examine how the misclassification rate changes with the number of neighbors used to train the model.

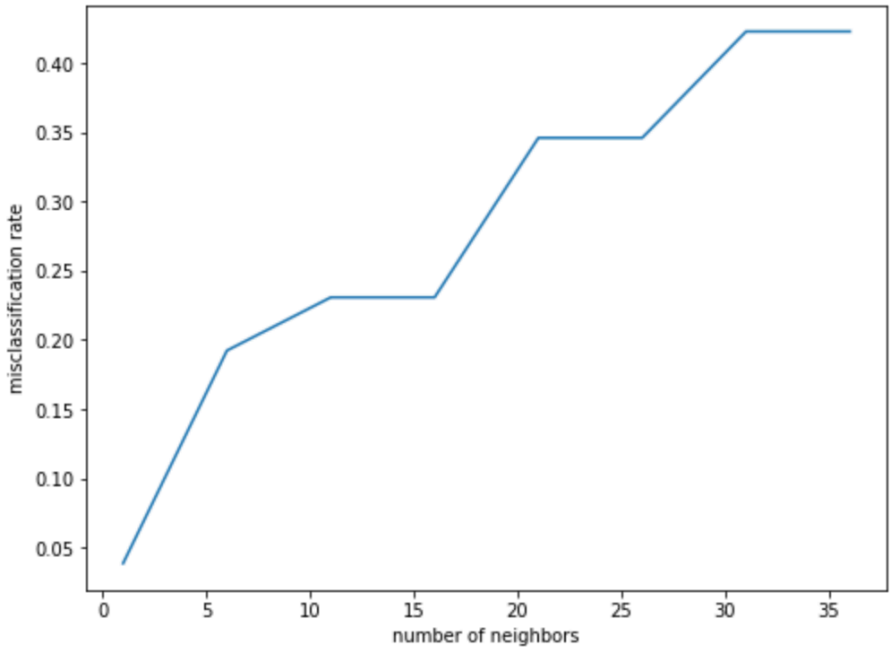

Figure 6.

As the number of neighbors increase, the bias is larger and the variance is smaller, so it is easier to underfit and the misclassification rate increases. In the model I fit before, I choose $k = 5$.

Since perceptron is a neural network with no hidden layer, I use ***nnet*** package in R to build a neural network with only one hidden layer. I also randomly choose 75 samples as the train data set to build the model, predict on the test set by the model, calculate the accuracy score and examine how the misclassification rate changes with the number of units (nodes) built in the hidden layer.

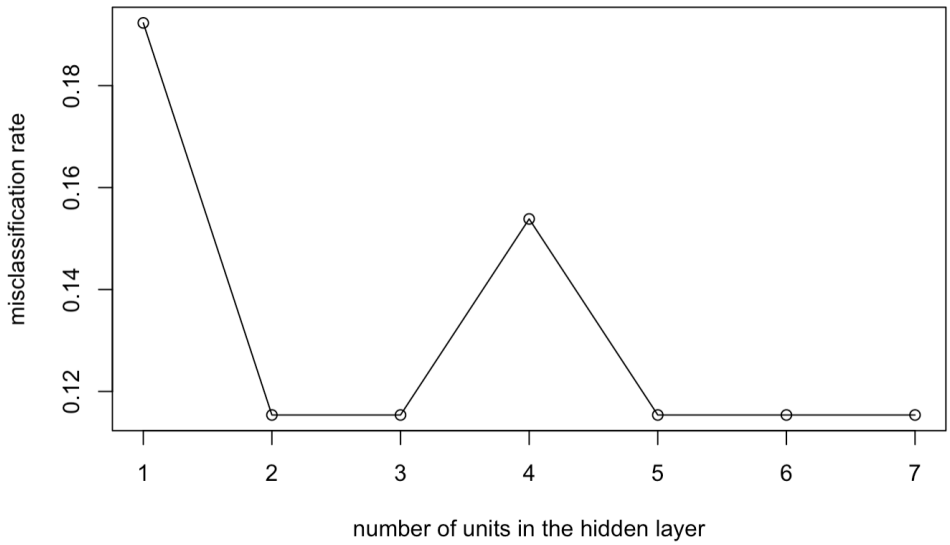

Figure 7.

There are some empirically derived rules-of-thumb about choosing the number, the most commonly relied on is 'the optimal size of the hidden layer is usually between the size of the input and size of the output layers'. From the plot, I cannot find a trend, but I can choose size=7 to fit the model in this animal zoo data set.

I randomly select the train sets of 75 samples five times and get five accuracy score: 0.9615385,0.9230769,0.9615385,0.8461538,1. I also compare the accuracy scores of ***nnet*** and perceptron, using the same train sets. The results are very similar.

	nnet	perceptron
[1,]	0.962903	1.000000
[2,]	1.000000	0.884615
[3,]	0.884615	0.961540

Table 6.

In conclusion, I cannot tell which classification method is the best from my attempts because they all have similar accuracy scores and cross-validation scores. I think it may because the zoo animal data set is too small, and the variables are too simple, only categorical variables. Another reason may be that I have explored too few times and incompletely. But at least, I can conclude that, these five classification methods, logistic regression, k-nearest neighbors, random forest, perceptron and neutral network with one hidden layer, all works well on this zoo animal data set.