Design of an Artificial Neural Network to Predict the External Curve of Horseshoes

Sait Han Uzun,
BSc, Undergraduate Artificial Intelligence Student,
De Montfort University

ABSTRACT

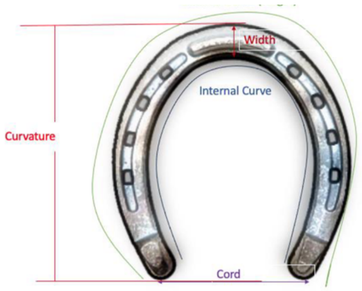
This study aims to predict the external curve length of horseshoes. Our dataset consists of 219 samples with five variables: (a) internal curve length (cm), (b) width length (mm), (c) cord length (cm), (d) curve length (cm), and (e) external curve length (cm) as the target variable. Manual measurement of the external curve can be time consuming and needs high attention. In order to address this, I designed an artificial neural network (ANN) that successfully predicts external curve length with high accuracy.

In order to achieve best results, the data were carefully preprocessed, and various hyperparameters such as activation functions, hidden layer size, training function (learning algorithm) were tuned. Cross validation and other techniques were applied to prevent overfitting and data leakage. As a result, the Artificial Neural Network model demonstrates highly accurate predictions of external curve length, providing a alternative to manual measurement.

INTRODUCTION

Horseshoes protect hooves from wear on hard surfaces and improve traction, which is important for riding, jumping, pulling, or moving on uneven terrain (Horse & Country, 2023). For horses with weak hooves or structural issues, shoes can provide support, correct gait, and reduce stress on joints (PetMD, 2023).
References:

- Horse & Country. (2023). Do horses need shoes? The pros and cons of shoeing.



Fitting horseshoes correctly is important for a horse because of their health and performance. A part of this is the external curve of the horseshoe, which must correctly fit. An unmeasured external curve can cause pain, damage, or affect the horse's performance. Measuring the external curve manually can be slow, tiring, inaccurate, or contain human error.

This study focuses on solving this problem by creating an artificial neural network model that can predict the external curve from measurements such as internal curve length, width (mm), cord length, and curvature length (cm). Data are carefully processed, the model is tuned, and validation techniques are used to make sure the predictions are accurate and reliable, providing a practical alternative to manual measurement.

METHODOLOGY

In order to predict External Curve Length in horseshoes, our dataset was examined in detail.
The dataset consists of five variables: internal curve length (cm), width (mm), cord length (cm), curve length (cm), and the target external curve length (cm).

The methodology includes two parts: data preprocessing and model training.
The first step was converting width (mm) to cm, since all other features are based on centimeters.

Data distribution and pairwise relationships were visualized, and outliers as well as highly correlated inputs were detected and removed.
After that, the data was augmented with noise, which increased the dataset size. Feature engineering was also applied to enhance the feature set and increase the input count.

A neural network with one hidden layers (20 neurons) was trained using Bayesian Regularization backpropagation. The tanh activation function was used in the first layers, and the purelin function was applied in the output layer for regression.

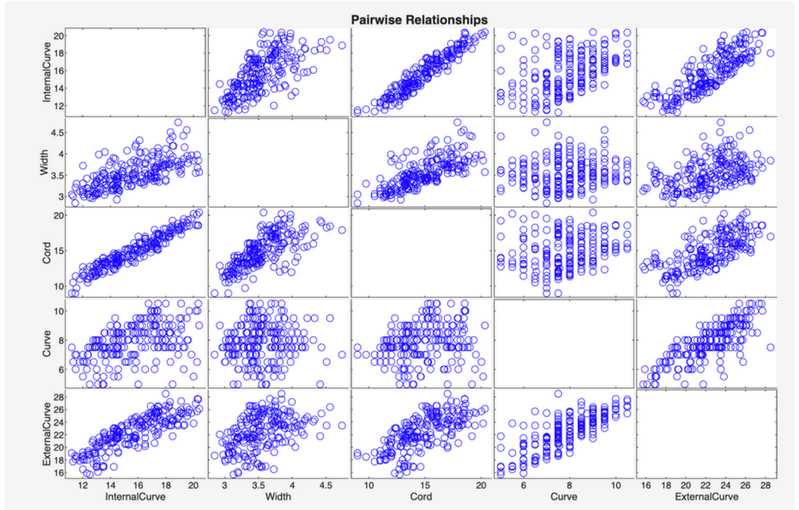
Training was performed using 4-fold cross-validation, with normalization applied each time between -1 and 1.
The best network was selected based on performance, and the results showed high percentage of vairence.

EXPERIMENTAL DESIGN

In the designed neural network, one hidden layer with 20 neurons was selected to capture the non-linear relationships in the data. Other configurations were tested, and a single layer with 20 neurons provided the best performance while avoiding overfitting.

Different training methods and normalization functions were also tested. Bayesian Regularization (trainbr) was selected as it showed the best results, preventing overfitting and providing best R2 values.

Noise was added to the training data to increase dataset size, and 4-fold cross-validation was chosen because the dataset is relatively small (219 samples). The noise level and augmentation factor were selected experimentally to avoid degrading learning performance



Outliers were detected and removed from the input features, especially in cord length, which had values far higher than the rest. Removing these outliers helped the network learn patterns more effectively. Additionally, highly correlated features were detected: internal curve length and cord length had 95% correlation, so cord length was removed from the training inputs.
Feature engineering was applied to the remaining less correlated features to create additional inputs for the network

```
corrMatrix =

    1.0000    0.6654    0.9523    0.4477    0.8376
    0.6654    1.0000    0.7161    0.0866    0.4444
    0.9523    0.7161    1.0000    0.2173    0.6811
    0.4477    0.0866    0.2173    1.0000    0.7776
    0.8376    0.4444    0.6811    0.7776    1.0000
```

Outlier Value: 68.25

Observation Row: 46
Group: Cord

Distance To Median: 53.5
Num IQRs To Median: 14.6075

Data was split using 4-fold cross-validation. In each fold, three parts were used for training and validation, and one part was used for testing on unseen data.

R2 was chosen as the primary evaluation metric.

Since tansig was used as the activation function in the hidden layers, all input features were normalized to the [-1, 1] range using MATLAB's mapminmax function. During testing, the predicted outputs were denormalized back to their original scale.

RESULTS

Data Leakage

```
meanR2 =

    0.9955


R2_scores =

    0.9940
    0.9962
    0.9962
    0.9955
```

The neural network model trained on the horseshoe dataset showed that data augmentation before cross-validation caused data leakage, leading to artificially high peformance of 99% on every fold, as the model effectively memorized test values. After disabling augmentation before splitting, the network achieved a more realistic accuracy of 89–92%. Comparatively, networks trained using trainlm with 0–1 normalization and similar hidden layers with transfer function of logsig scored around 87%, demonstrating lower performance. These results indicate that Bayesian Regularization backpropagation, combined with normalized inputs between -1 and 1, provides the best results and generalization.

```
hiddenLayerSize = [20];
trainFcn = 'trainlm';
transferFcn = 'logsig';
normRange = [0, 1];
```

```
meanR2 =

    0.8743


R2_scores =

    0.8265
    0.9310
    0.8256
    0.9141
```

```
hiddenLayerSize = [20];
trainFcn = 'trainbr';
transferFcn = 'tansig';
normRange = [-1, 1];
```

```
meanR2 =

    0.9122


R2_scores =

    0.9263
    0.8614
    0.9306
    0.9304
```

CONCLUSIONS

The model was kept basic to prevent overfitting and data leakage, and with this way, it achieved 92% overall generalization. However, this performance could be improved by collecting more data or testing different network architectures, as the current dataset is extremely small with only 219 samples. Adding more samples would give the network more information to learn, capture patterns and likely increase accuracy. With a larger dataset and more complex or optimized architectures, the model can perform significantly better.