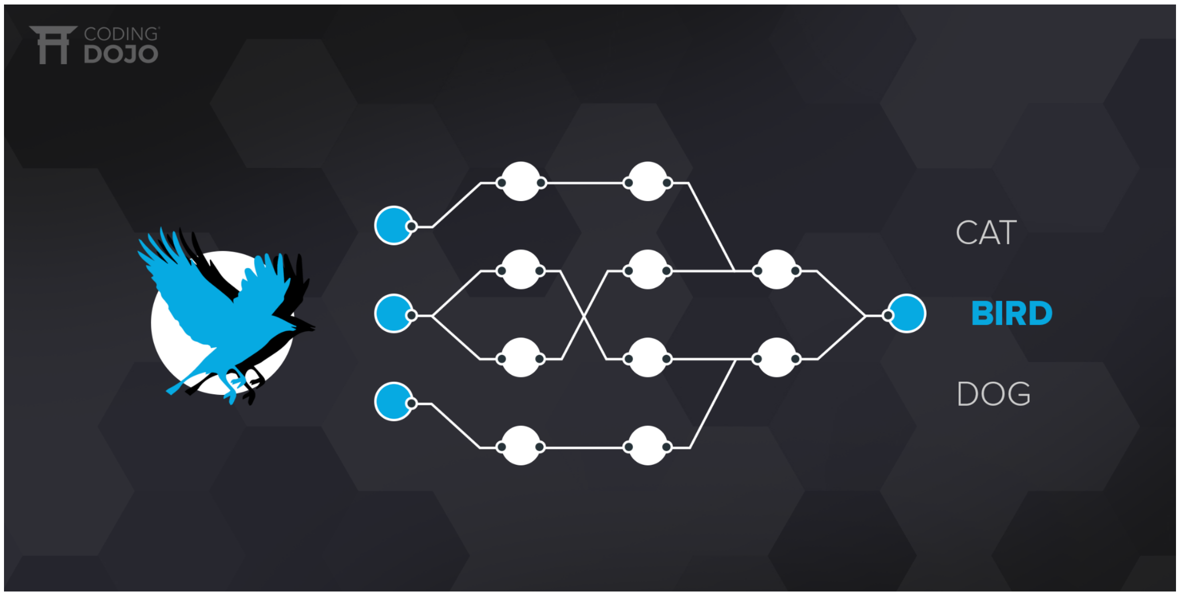
User Authentication Based On Keystroke Dynamics using Artificial Neural Network

Machine Intelligence & Expert Systems 2021



Group 8

Aryendra Singh 18EC10077

Devansh Rajgarhia 18EC10073

Awadh Kejriwal 18EC10078

Viplaw Srivastava 18EC10067

Utkarsh Patel 18EC35034

PROBLEM STATEMENT:

To authenticate the user based on the keystroke logging features like hold time and latency time of the neutral, happy and sad mood data using an artificial neural network.

INTRODUCTION:

The project aims to detect unauthorized people to prohibit access to the particular system and private data of users. Keystroke logging is an act of tracking and recording every keystroke entry made on a computer. A “keystroke” is just any interaction you make with a button on your keyboard. The two keylogger features for a particular user are almost unique and hence it is used to uniquely identify the user. Hold time basically tells you the length of keypress and the latency time tells the time to switch from one particular key to the next. These two features put the surveillance on the user and simply decline if the trained model declares him as an unauthorized user. Having these two keylogging features along with the mouse dynamics can accurately authenticate the user but we are only using keylogging features in this project.

Today most computer systems identify users by means of secret phrases known as passwords. However, this authentication system does nothing to protect the computer from unauthorized access once the user has started an active session.

These limitations of password based authentication lead to the introduction of authentication techniques based on biometrics.

BIOMETRIC AUTHENTICATION:

Human recognition can be done by using his physiological or behavioral characteristics. Biometrics offer automated methods of identity verification or identification on the principle of measurable physiological or behavioral characteristics.The characteristics are measurable and unique

There can be 2 types of biometrics:

1) Physical Biometrics: Fingerprint or iris scans can be a type of physical biometric. These do not change generally and are very secure.

2) Behavioural Biometrics: These characteristics are acquired over time and are at least partly based on acquired behavior. Example: Keystroke and mouse dynamics.

FEATURE EXTRACTION

We run the code provided to us for feature extraction, given the continuous data collected and the extracted features are as follows:

1.) We get 26 hold times for the letters A to Z
2.) We get 676 (26✖26) latency values for each pairing: AA,AB...ZY,ZZ

We ignore the hold and latency times of other characters on the keyboard like ‘Spacebar’ or other keys like Tab, Ctrl or Backspace.

We take these 26 + 26✖26 = 702 values as a feature vector for one class and use it to classify from several classes. We take the average of all hold time and latency values for one particular key or one particular pair of keys as this helps in smoothing the values and returns an accurate representation of the several values.

Hold time tells us the length of the keypress or the time difference between the press time and the release time. It signifies the impulsive nature of the user.Latency tells us the time to switch from one particular key to the other. It signifies the switching and typing speed of the user.

ARTIFICIAL NEURAL NETWORKS:

Artificial neural networks (ANNs) are composed of a node layer, containing an input layer, one or more hidden layers, and an output layer. Each node, or artificial neuron, connects to another and has an associated weight and threshold. If the output of any individual node is above the specified threshold value, that node is activated, sending data to the next layer of the network. Otherwise, no data is passed along to the next layer of the network.

Artificial Neural Network can be best represented as a weighted directed graph, where the artificial neurons form the nodes. The association between the neuron outputs and neuron inputs can be viewed as the directed edges with weights. The Artificial Neural Network receives the input signal from the external source in the form of a pattern and image in the form of a vector. These inputs are then mathematically assigned by the notations x(n) for every n number of inputs.

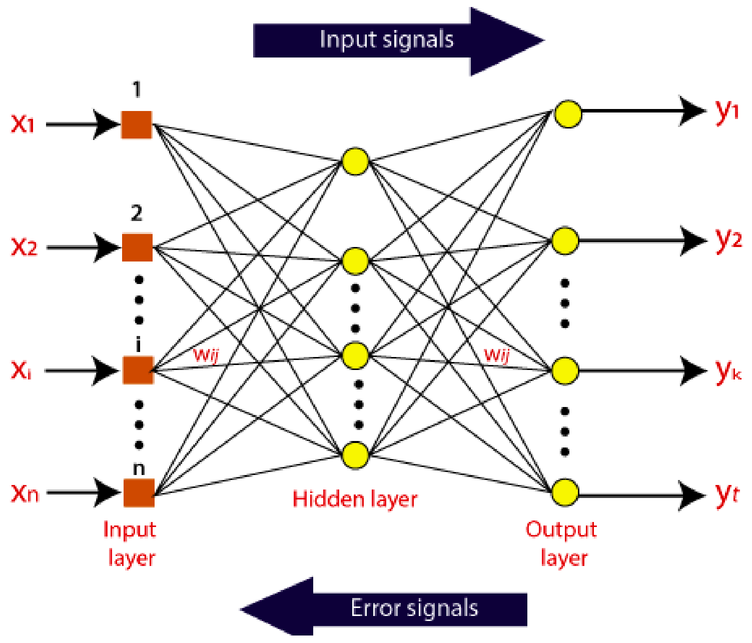
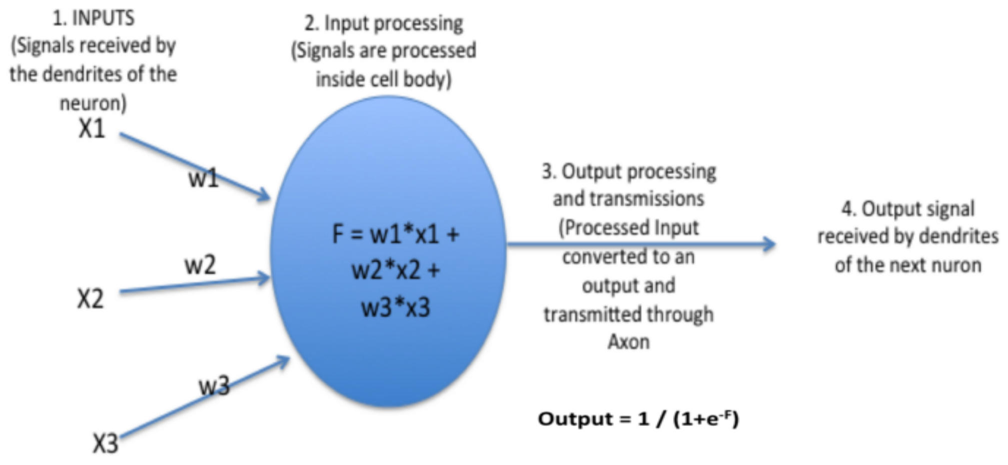


Fig. Artificial Neural Network Structure

The activation function refers to the set of transfer functions used to achieve the desired output. There is a different kind of activation function, but primarily either linear or non-linear sets of functions. Some of the commonly used sets of activation functions are the Binary, linear, and Tan hyperbolic sigmoidal activation functions



1. INPUTS (Signals received by the dendrites of the neuron)
X1
X2
X3
w1
w2
w3
2. Input processing (Signals are processed inside cell body)
F = w1*x1 + w2*x2 + w3*x3
3. Output processing and transmissions (Processed Input converted to an output and transmitted through Axon
4. Output signal received by dendrites of the next nuron
Output = 1 / (1+e^-F)

Fig. Perceptron

At First, information is fed into the input layer which then transfers it to the hidden layers, and interconnection between these two layers assign weights to each input randomly at the initial point. and then bias is added to each input neuron and after this, the weighted sum which is a combination of weights and bias is passed through the activation function. Activation Function has the responsibility of which node to fire for feature extraction and finally output is calculated. This whole process is known as Forward Propagation.

$$net_j = \sum_{i=1}^{d} x_i w_{ij} + w_{j0} = \sum_{i=0}^{d} x_i w_{ij}$$
$$\Rightarrow y_j = f(net_j)$$

After getting the output model to compare it with the original output the error is known and finally, weights are updated in backward propagation to reduce the error and this process continues for a certain number of epochs (iteration). Finally, model weights get updated and prediction is done.

$$J(w) = \frac{1}{2} \sum_{k=1}^{c} (t_k - z_k)^2 = \frac{1}{2} \| t - z \|^2$$

Where $J(w)$ represents the loss function, c = total number of output & t is the given output and z is the predicted output.

$$\triangle w = -\eta \frac{\partial J}{\partial w}$$

$$\Rightarrow \triangle w_{pq} = -\eta \frac{\partial J}{\partial w_{pq}}$$

$$\Rightarrow \triangle w_{kj} = -\eta \frac{\partial J}{\partial w_{kj}} = -\eta \frac{\partial J}{\partial net_k} * \frac{\partial net_k}{\partial w_{kj}} \text{ , } where\ net_k = \sum_{j=1}^{nH} y_i w_{kj} + w_{k0}$$

$$\Rightarrow \triangle w_{kj} = -\eta \left(\frac{\partial J}{\partial z_k} * \frac{\partial z_k}{\partial net_k}\right) * y_j \text{ , } where\ z_k = f(net_k)$$

$$\Rightarrow \triangle w_{kj} = \eta (t_k - z_k) * f^{\backprime}(net_k) * y_j$$

where η is the learning rate. Finally we update the weights in m+1 th iteration using,

$$w(m+1) = w(m) + \triangle w$$

RESULTS and CHALLENGES

The table below tabulates our final result. The Test accuracy mentioned is the average of the accuracy for each of the folds. We do a hyperparameter search over 2 parameters. Hidden layer Configuration and Random State configuration.

In the Hidden layers we have used 6 different configurations of varying nodes per hidden layer and varying number of hidden layers. As is evident from the table the best accuracy is reached when we have 2 hidden layers with sizes of 100 and 25 neurons respectively.

Next we vary the Random State used for the Classifier. The random state determines random number generation for weights and bias initialization. As is clear from the table the best result is obtained when the random state chosen is 99.

All of the classifiers for the different hidden layers values are run for a maximum of 1000 iterations.

All the classifiers for the varying random state configurations had 2 hidden layers of 100 and 25 neurons respectively and ran for a maximum of 500 iterations.

For future work it may help to have a more standard way of collecting data, through services like Amazon Turk. This will help us to get a larger dataset with standard environment and length of typed words. One can also argue that variation in equipment such as the quality of the keyboard may invoke differing responses from the person whom we are trying to authenticate. Such as for a more complete and sound study one may even consider putting forth standard equipment for users.

Hidden Layers	Five Fold Test Accuracy		Random State Configuration	Five Fold Test Accuracy
100::25	**0.644**		10	0.638
200::100::25	0.616		25	0.600
200::50::25	0.611		40	0.622
200::50::50	0.627		50	0.605
300::200::50::50	0.600		76	0.622
300::200::20::20	0.616		**99**	**0.661**

REFERENCES

1. https://medium.com/machine-learning-researcher/artificial-neural-network-ann-4481fa33d85a
2. https://analyticsindiamag.com/a-beginners-guide-to-scikit-learns-mlpclassifier/
3. https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPClassifier.html