CS 6375

ASSIGNMENT 3 (Neural Networks)

Names of students in your group:

Ankita Patil (asp160730)

Abhilash Gudasi (abg160130)

Number of free late days used: 0

Please list clearly all the sources/references that you have used in this assignment.

Machine learning by Tom Mitchell

Assumptions

* Bias neurons are ignored
* Last column in the dataset is the class label
* Training and test datasets have the same format

The dataset is read from following URLs

1. Car Evaluation Dataset: <https://archive.ics.uci.edu/ml/datasets/Car+Evaluation>
2. Iris Dataset: <https://archive.ics.uci.edu/ml/datasets/Iris>
3. Adult Census Income Dataset: <https://archive.ics.uci.edu/ml/datasets/Census+Income>

The datasets are preprocessed by changing the categorical variables to numerical values, standardized and scaled the attributes.

Our neural model is built with one input layer, 2 hidden layer [4,2] and one output layer.

**Please note that**, number of nodes in each hidden layer can be changed.

We have evaluated our neural net by changing the different parameters such as

* Learning rate
* Number of nodes in hidden layer
* Number of iterations

**Case 1:**

Changing/variable parameter in NN-> Learning rate

Here, we are setting number of iterations = 1000

Learning rate values -> 0.01, 0.05, 0.1,0.2

Number of nodes in hidden layer 1 = 4

Number of nodes in hidden layer 1 = 2

We have documented the result by changing the learning rate in the following table (refer next page)

Following table has entries for training and test error for Adult, Car and Iris Dataset for activation functions sigmoid, tanh, relu for different values of learning rate.

We have run our neural network code for various learning rates -> 0.01, 0.05, 0.1, 0,2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter ƞ** | **Dataset** | **Activation function** | **Training Error** | **Test Error** |
| 0.01 | Adult | Sigmoid | 0.061098725 | 0.061797139 |
|  |  | tanh | 0.734540782 | 0.452420424 |
|  |  | Relu | 0.124544098 | 0.124171088 |
|  | Car | Sigmoid | 0.112220843 | 0.082388944 |
|  |  | tanh | 1.199803122 | 0.460868206 |
|  |  | Relu | 0.369659667 | 0.323188406 |
|  | Iris | Sigmoid | 0.21357217 | 0.214694475 |
|  |  | tanh | 0.353498195 | 0.396616647 |
|  |  | Relu | 0.836134454 | 0.862068966 |
| 0.05 | Adult | Sigmoid | 0.124544098 | 0.124171088 |
|  |  | tanh | 0.799776194 | 0.441810345 |
|  |  | Relu | 0.124544098 | 0.124171088 |
|  | Car | Sigmoid | 0.11177914 | 0.082446211 |
|  |  | tanh | 1.211578666 | 0.476186107 |
|  |  | Relu | 0.369659667 | 0.323188406 |
|  | Iris | Sigmoid | 0.210654615 | 0.21179104 |
|  |  | tanh | 0.416612877 | 0.397169031 |
|  |  | Relu | 0.836134454 | 0.862068966 |
| 0.1 | Adult | Sigmoid | 0.124544098 | 0.124171088 |
|  |  | tanh | 0.54952752 | 0.784648541 |
|  |  | Relu | 0.124544098 | 0.124171088 |
|  | Car | Sigmoid | 0.203850968 | 0.163926801 |
|  |  | tanh | 1.047067343 | 1.144927536 |
|  |  | Relu | 0.369659667 | 0.323188406 |
|  | Iris | Sigmoid | 0.21035977 | 0.211492735 |
|  |  | tanh | 0.477671036 | 0.389839863 |
|  |  | Relu | 0.836134454 | 0.862068966 |
| 0.2 | Adult | Sigmoid | 0.124544098 | 0.124171088 |
|  |  | tanh | 0.658197944 | 0.452751989 |
|  |  | Relu | 0.124544098 | 0.124171088 |
|  | Car | Sigmoid | 0.205997723 | 0.16741306 |
|  |  | tanh | 0.298186639 | 0.431884058 |
|  |  | Relu | 0.369659667 | 0.323188406 |
|  | Iris | Sigmoid | 0.210219009 | 0.211346873 |
|  |  | tanh | 0.369169025 | 0.395957194 |
|  |  | Relu | 0.836134454 | 0.862068966 |

**Analysis**

We started with experimenting the different values of learning rate. Initially we started with relatively higher values of learning rate such as 0.1, 0.2 and then lower values such 0.05, 0.01

If the learning rate is low then the training error seems reliable but optimization can take lot of time. In conclusion, if training rate is very small, then the gradient descent can be low.

If learning rate is high, then the training may not converge or even diverge. Gradient descent can overshoot the minimum.

Consider the Adult dataset,

When learning rate was 0.1, training error for sigmoid activation function was 0.124544098 and even for learning rate 0.2 it was 0.124544098

We observe that, for learning rate 0.1, 0.2 the training error was almost constant and not improving and it was almost increasing in few cases.

When learning rate was 0.01 (exponentially smaller value), the training error was 0.061098725

Hence, when the learning rate was relatively small -> 0.01 the training error started decreasing.

Hence, to conclude -> **learning rate should be relatively of smaller magnitude to achieve better training as it significantly impacts the training.**

Also, we have used different activation functions sigmoid, tanh, relu. To conclude about activation functions, we can say that there is nothing like best activation functions. One activation function may be better than other or worse than other. Selection of activation function does not affect what the neural network can learn, it just affects how many data points it needs. However, we observe that, relu works faster and gave better or similar results compared to sigmoid and tanh.

**Model Evaluation**

* In few cases, test error is greater than training error, so we can say that model is not able to able to generalize well on unknown samples and that is why providing high test error. (Case of overfitting)

Example from above table,

For Adult dataset, (learning rate = 0.1 and activation function -> tanh), training error is 0.54952752 and test error is 0.784648541 [Case of Overfitting]

* In few cases, test error is low, **slightly higher** than training error (Case of good fit)

Example from above table,

For Adult dataset, (learning rate = 0.01 and activation function -> sigmoid), training error is 0.061098725 and test error is 0. 0.061797139 [Case of good fit]

**Case 2:**

Changing/variable parameter in NN-> **Number of nodes in hidden layer**

Here, we are setting number of iterations = 1000

Learning rate -> 0.05

Number of nodes in hidden layer 1 = 5

Number of nodes in hidden layer 1 = 3 **OR**

Number of nodes in hidden layer 1 = 2

Number of nodes in hidden layer 1 = 2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter** | **Dataset** | **Activation function** | **Training Error** | **Test Error** |
| Number of | Adult | Sigmoid | 0.124544098 | 0.124171088 |
| Nodes in |  | tanh | 0.45445126 | 0.810179045 |
| hidden layer |  | Relu | 0.124544098 | 0.124171088 |
|  | Car | Sigmoid | 0.105156654 | 0.074464169 |
| h1 = 5 |  | tanh | 1.274438812 | 0.41958679 |
| h2 = 3 |  | Relu | 0.369659667 | 0.323188406 |
|  | Iris | Sigmoid | 0.105156654 | 0.074464169 |
|  |  | tanh | 1.274438812 | 0.41958679 |
|  |  | Relu | 0.369659667 | 0.323188406 |
|  | Adult | Sigmoid | 0.124544097 | 0.124171087 |
|  |  | tanh | 0.45179874 | 0.667274536 |
|  |  | Relu | 0.124544098 | 0.124171088 |
| h1 = 2 | Car | Sigmoid | 0.118696326 | 0.08910497 |
| h2 = 2 |  | tanh | 1.204923968 | 0.524637681 |
|  |  | Relu | 0.369659667 | 0.323188406 |
|  | Iris | Sigmoid | 0.168283468 | 0.172632939 |
|  |  | tanh | 0.416366221 | 0.39799712 |
|  |  | Relu | 0.836134454 | 0.862068966 |

We experimented with number of neurons in hidden layers such as h1=2, h2= 2 OR

h1 = 4, h2 = 2 OR h1=5, h2 = 3. Initially when we started increasing number of nodes in the hidden layer, we observed better model but after some threshold, we observed that test error was greater than training error (Case of overfitting)

**Case 3:**

Changing/variable Parameter in NN-> Number of iterations

Here, we are setting number of iterations = 500 **OR** 2000

Learning rate -> 0.05

Number of nodes in hidden layer 1 = 4

Number of nodes in hidden layer 1 = 2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter** | **Dataset** | **Activation function** | **Training Error** | **Test Error** |
|  | Adult | Sigmoid | 0.124544098 | 0.124171088 |
|  |  | tanh | 0.496145557 | 0.757460212 |
|  |  | Relu | 0.124544098 | 0.124171088 |
| Number of | Car | Sigmoid | 0.113002803 | 0.082744831 |
| iterations |  | tanh | 1.199129304 | 1.096546348 |
| 500 |  | Relu | 0.369659667 | 0.323188406 |
|  | Iris | Sigmoid | 0.21129395 | 0.21242838 |
|  |  | tanh | 0.48517999 | 0.401708303 |
|  |  | Relu | 0.836134454 | 0.862068966 |
|  | Adult | Sigmoid | 0.124544098 | 0.124171088 |
|  |  | tanh | 0.809723143 | 0.44678382 |
|  |  | Relu | 0.124544098 | 0.124171088 |
|  | Car | Sigmoid | 0.111347481 | 0.082174337 |
| 2000 |  | tanh | 1.268483673 | 0.983773303 |
|  |  | Relu | 0.369659667 | 0.323188406 |
|  | Iris | Sigmoid | 0.210359589 | 0.211492823 |
|  |  | tanh | 0.397383043 | 0.389244471 |
|  |  | Relu | 0.836134454 | 0.862068966 |

**Analysis**

When we increased the iterations, we observe that, test error was greater than training error (Case of overfitting/Rote learning) because the model started memorizing values.