**HARVEST PLATFORM**

**Problem 1**: Classification with a Neural Network

Here, our dataset is labelled with three classes namely 1, 2, and 3 (3 types of wines). Hence, this becomes a Supervised Learning problem. The main idea is to make a Neural Network with 178 examples as our inputs and 3 output classes. So, in Neural Network terms, this translates as 178 units in the input layer and 3 units in the output layer.

1. **Approach**:

Here, I have attempted to make a NN with 4 layers namely, two layers of input and output and two hidden layers, so, in all, there are 4 layers. Here, due to availability of limited computation resources, I have included only 4 layers in the NN but, if there are resources available, we can also expand our NN to a greater number of layers and achieve better results.

1. **Implementation**:

Step-1: I have imported all the packages (numpy, pandas) and reading of the csv file (data) is performed and label data (output variables) is separated from the independent variables (features). Also, I have split the dataset in the ratio of 80-20 (train-test respectively) using the sklearn’s train\_test\_split().

Step-2: The next step is **forward propagation**. Forward propagation means traversing every hidden layer once and applying activation function to it. Here, the activation function that I have used is hyperbolic tangent function (tanh). There are other types of activation functions too like Sigmoid function, but it has certain downsides like:

1. **Vanishing gradient problem**: if the gradient at each step is too small, then greater repetitions will be needed until convergence, because the weight is not changing enough at each iteration. Or, the weights will not move as close to the minimum (versus greater gradients) in the set number of iterations. And with really small gradients, this becomes a problem. It becomes infeasible to train NN, and it starts predicting poorly
2. Its output isn’t zero centred. It makes the gradient updates go too far in different directions.
3. They have small convergences

Hence, I have used tanh as activation function for the hidden layers. Also, since it’s a multiclass classification problem (number of output classes > 2), a popular approach is to use SoftMax as the activation function for output layer, and hence I have used SoftMax activation function for the output layer. Also, one important detail is, as in every NN, I have added a bias at every layer except for the output layer in order for the NN to learn better.

1. First, I took z1 = a0W1 + b and passed it to the first activation function. In this way, I formed the first hidden layer. Here, a0 is the vector with all the values of the input parameters of the input layer (178x13). Then I computed a1 = tanh (z1) by applying tanh activation function to z1.
2. Then, I passed a1 as an input to the next hidden layer and computer z2 and a2 in the similar fashion and obtained z3 and then finally, I computed a3 by applying SoftMax activation function to z3. Here, tanh is an inbuilt function in numpy whereas SoftMax is not.

This is how I performed Forward Propagation.

Step-3: **Backward Propagation.** After forward propagation, I back propagated error gradient to update the weight parameters. The aim was to minimize the error at every layer. I did this by taking the derivative of the error function with respect to the weights using gradient descent. Gradient descent is taking steps in the direction of the slope. Here, I calculated the error gradient for all the units in all the layers (except for the input layer) including the bias units in all the layers.

Step-4: **Training the Neural Network.** The aim was to reach the best possible weights to obtain the best possible output (accuracy). In order to this, I tried and tested with various learning rates, I chose a learning rate of 0.07 (learning rate decides how quickly or slowly the parameters be updated). I started the training with 1000 epochs which gave the training accuracy of about 90% and I kept on increasing the number of epochs in the increments of 500 i.e., 1500, 2000, …, 4000. 4000 epochs gave me training accuracy of 98.87 % on the given dataset. I have also plotted the graph to see how accuracy of the algorithm increases with the number of training iterations (epochs).

Step-5: **Testing the Neural Network.** Here, I have used 20% of the data for testing. I tested the data using forward propagation again but, with the final updated weights that I received from training the Neural Network. I got a testing accuracy of 88%. We can further try and test with various parameters like adjusting the hyperparameters, learning rate, changing the train-test ratio etc.

1. **Conclusion:**

Here, I saw that more I train the NN, better is the training accuracy of the algorithm. For testing accuracy, I have predicted the hypothesis against the actual test labels and obtained a testing accuracy of about 88%.

**Problem 2: Web automation**

* Using web automation, please create a program that helps a user find whether they have unclaimed assets using the following website: [http://www.missingmoney.com](http://www.missingmoney.com/)/

1. **Approach**: Here, the approach was very sequential. Since I had never worked with Selenium before, I first got acquainted with Selenium and setup the tools that I’ll be requiring to work with Selenium. So, I downloaded the python package of Selenium and I downloaded the chrome driver to work with Google Chrome.

**Step-1:** I began writing the script where I first gave the path to Selenium web driver to locate Firefox ‘geckodriver.exe’ and gave the URL of the website. On the first page, I located the text boxes for first name, last name and state and then wrote the script for the driver to fill those textboxes with the user’s first name, last name and state.

**Step-2:** On the next page, I get a list of possible matches with the user’s profile and I refine the results obtained on the next page using further information about the user’s city and state and then I click on one of the refined results and I go to the next page.

**Step-3:** Here, I click on the button ‘Yes, I can claim’ and then go to the next page where it asks the user to go to the state’s website where I tried to fill in the first name and last name of the user again.