DSEM 6150 ASSIGNMENT 7

NEURAL NETWORKS

PRESENTED BY: RUCHI KAPADIWALA

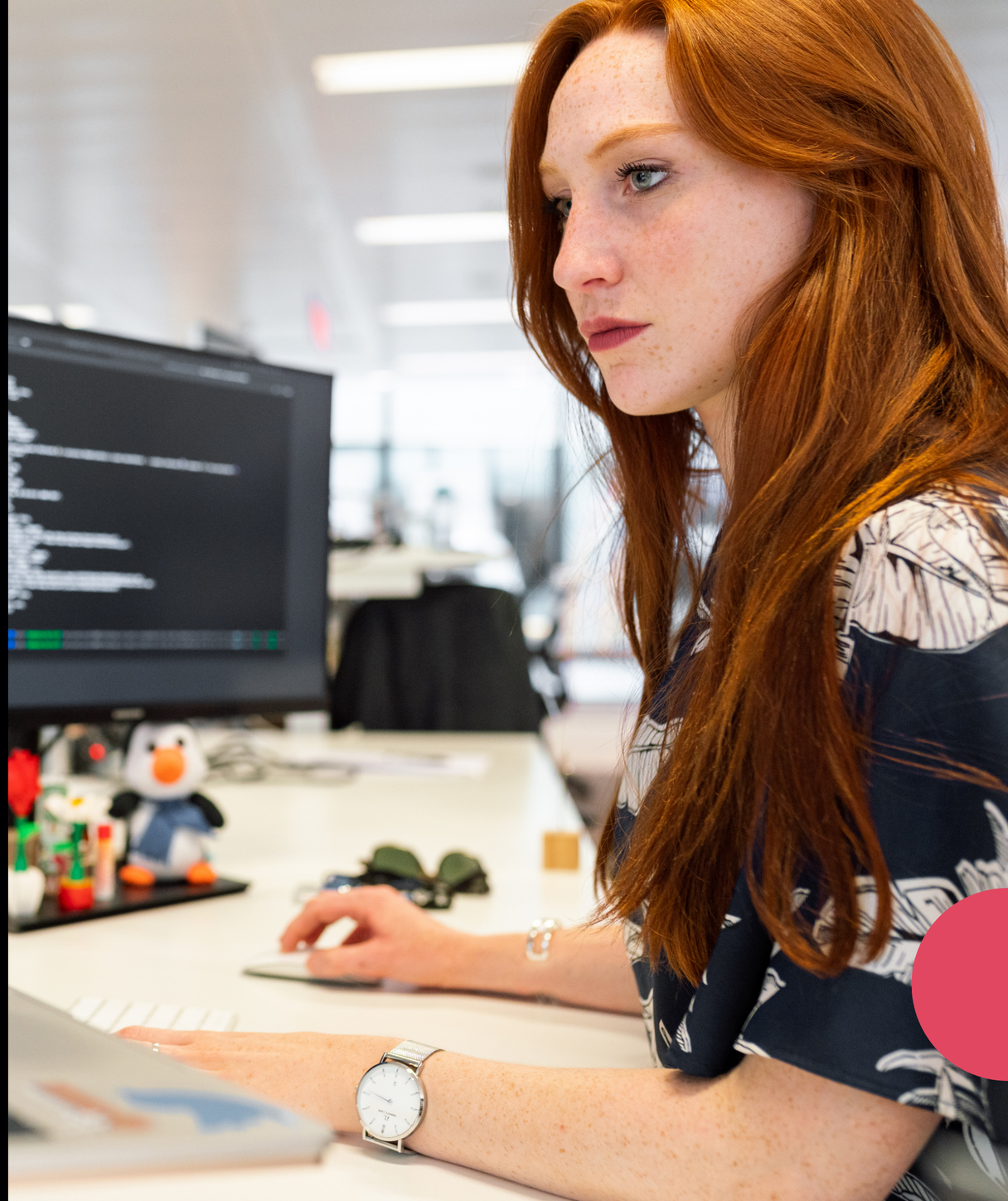
DSEM 6150 ASSIGNMENT 7

INTRODUCTION

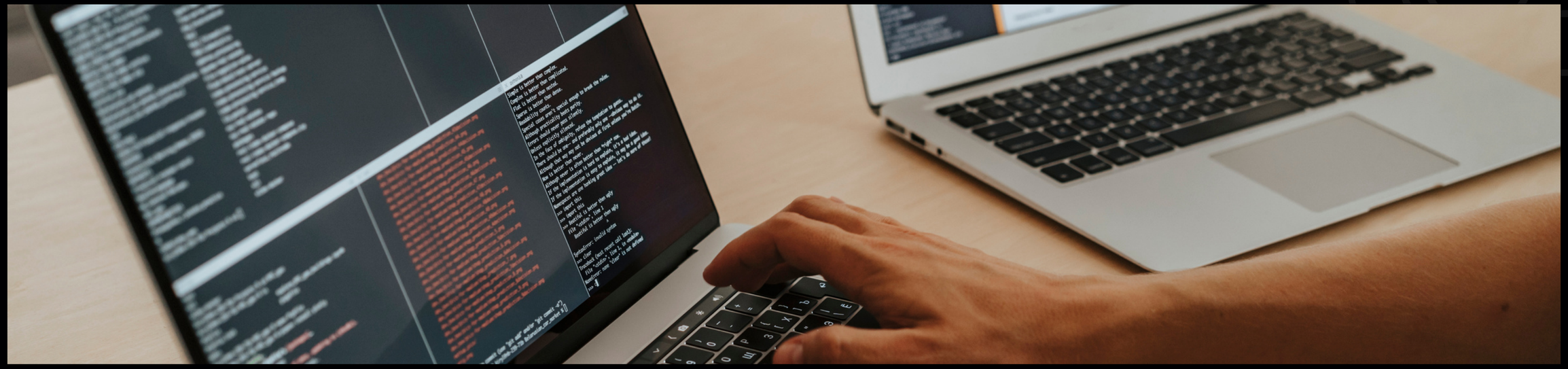
Neural networks are a type of algorithm used in machine learning to recognize patterns in data. They are modeled after the structure of the human brain and consist of layers of interconnected nodes or neurons.

The network is made up of interconnected nodes called neurons, which are organized into layers. Each neuron receives inputs from the previous layer, performs a weighted sum of those inputs, and applies a non-linear activation function to produce an output, which is then passed on to the next layer.

Feedforward neural networks are typically used for supervised learning tasks such as classification and regression and are trained using backpropagation, an algorithm that adjusts the weights of the connections between neurons to minimize the error between the network's predictions and the true output values.



DSEM 6150 ASSIGNMENT 7



PROBLEM STATEMENT

Implement a simple feed-forward neural network model for the loan dataset and then evaluate the performance on a 80/20 split. Exclude irrelevant columns and perform one-hot encoding and normalization of each column, if needed.

DSEM 6150 ASSIGNMENT 7

DATASET

Sex	Age	Time_at_address	Res_status	Telephone	Occupatio	Job_status	Time_employed	Time_bank	Liab_ref	Acc_ref	Home_Expn	Balance	Decision
M	50.75	0.584999979	owner	given	unemploye	unemploye	0	0	f	given	145	0	reject
M	19.67	10	rent	not_given	labourer	governmen	0	0	t	given	140	0	reject
F	52.83	15	owner	given	creative_	private_s	5	14	f	given	0	2200	accept
M	22.67	2.539999962	rent	not_given	creative_	governmen	2	0	f	given	0	0	accept
M	29.25	13	owner	given	driver	governmen	0	0	f	given	228	0	reject
F	16.08	0.335000008	owner	given	unemploye	unemploye	0	1	f	given	160	126	reject
M	23.17	11.125	owner	given	professio	governmen	0	1	f	given	100	0	accept
F	27.58	3	owner	given	manager	private_s	2	1	t	given	280	10	reject
F	19.17	5.414999962	owner	given	guard_etc	governmen	0	0	f	given	80	484	reject
F	27.25	0.289999992	owner	given	manager	governmen	0	1	t	given	272	108	reject
M	34.5	4.039999962	rent	not_given	guard_etc	self_empl	8	7	t	given	195	0	accept
M	20	1.25	rent	not_given	labourer	private_s	0	0	f	given	140	4	reject
M	34.42	1.335000038	owner	given	guard_etc	self_empl	0	0	t	given	440	4500	reject
M	41	0.039999999	owner	given	executive	private_s	0	1	f	oth_inst_	560	0	accept
F	24.75	3	owner	given	office_st	governmen	1	19	f	given	0	500	accept
F	38.25	6	owner	given	labourer	private_s	1	0	t	given	0	0	accept
M	28	2	owner	given	labourer	governmen	4	2	t	given	181	0	accept
F	27.67	2.039999962	owner	given	productio	private_s	0	0	t	given	180	50	reject
F	22.5	0.414999992	owner	given	guard_etc	private_s	0	0	t	oth_inst_	144	0	reject
F	22.42	5.664999962	owner	given	office_st	private_s	2	7	f	given	129	3257	accept
M	23.25	1.5	owner	given	office_st	private_s	2	3	t	given	0	582	accept

A simple feed-forward neural network is one of the most basic types of neural networks. It consists of three main components: an input layer, one or more hidden layers, and an output layer.

The network is called "feed-forward" because the data flows through the layers in one direction, from the input layer to the output layer.

The input layer of a feed-forward neural network consists of nodes that represent the input data. The number of nodes in the input layer is determined by the number of input features in the data.

DSEM 6150 ASSIGNMENT 7

TECHNICAL APPROACH

1. Data preparation: Load and preprocess the data using tools such as NumPy and Pandas. Split the data into training and testing sets.

2. Define the network architecture: Create a Sequential model in TensorFlow, specifying the number of input nodes, hidden layers, and output nodes. Add Dense layers to the model for the hidden and output layers, specifying the activation functions for each layer.

3. Initialize the weights: The weights are initialized automatically by TensorFlow.

4. Forward propagation: Use the model to predict the output of the neural network on the input data. This is done by calling the model's predict method.

5. Calculate the error: Calculate the error between the predicted output and the actual output using a loss function such as mean squared error or categorical cross-entropy.

TECHNICAL APPROACH

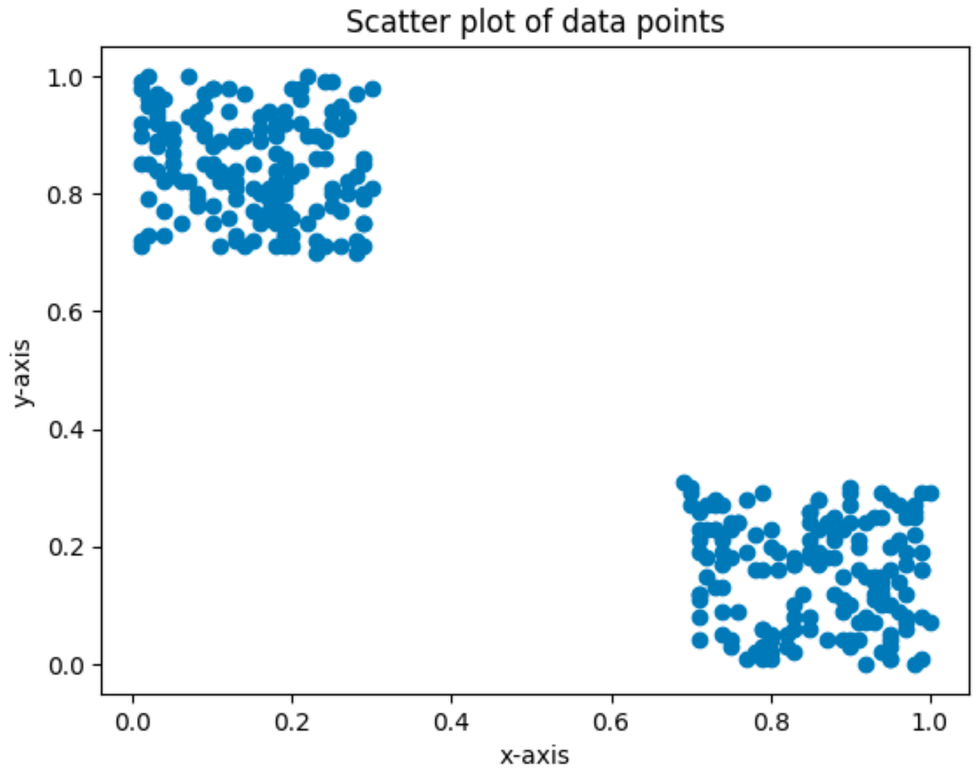
6. Backpropagation: Use an optimizer such as stochastic gradient descent to update the weights in the neural network in order to minimize the error. This is done by calling the model's fit method and specifying the training data, loss function, optimizer, and number of epochs.

7. Repeat steps 4-6 for all training examples: TensorFlow automatically iterates over the training set multiple times during the training process.

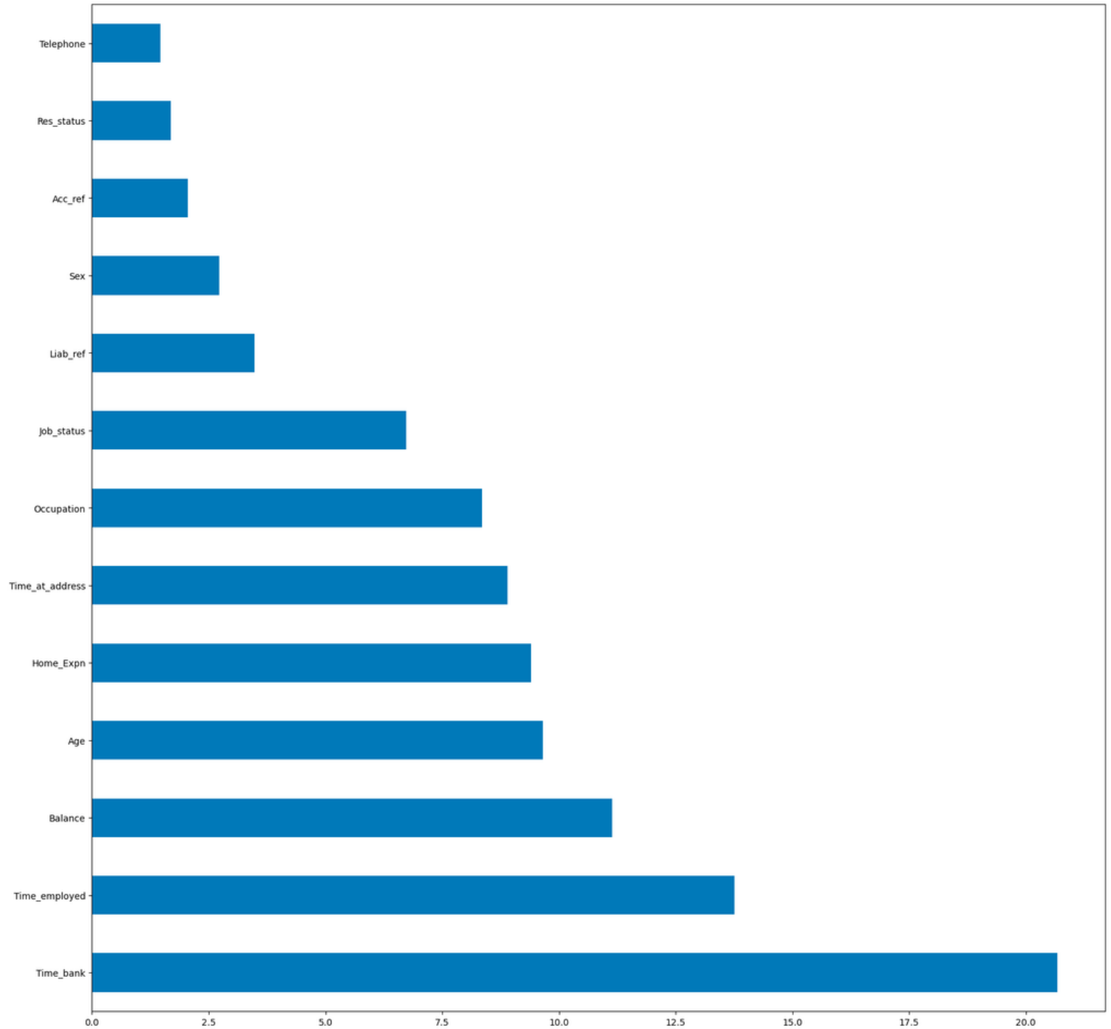
8. Test the network: Evaluate the performance of the network on the testing set using metrics such as accuracy or mean squared error.

9. Deploy the network: Once the network has been trained and tested, it can be used to make predictions on new, unseen data by calling the model's predict method.

Results



Results


Telephone
Res_status
Acc_ref
Sex
Liab_ref
Job_status
Occupation
Time_at_address
Home_Expn
Age
Balance
Time_employed
Time_bank
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0