Artificial Intelligence CS331: Artificial Neural Network
Group Number: Group 1
Group Members: Muhammad Bilal Shahid, Muhammad Taha, Muhammad Abdullah Ijaz, Mohammad Taha Zakir, Samee Arif
Dataset: Fashion-MNIST

Why 100 epochs?
After 100 epochs, the increase in the accuracy was negligible, and at one point, accuracy started to decrease, signifying overfitting in neural network, which means that the neural network starts to learn the data ‘by heart.’ Lowering the epochs resulted in underfitting since the accuracy decreased.

Why 0.001 learning rate?
Learning rate signifies at what rate we make the gradient descent. If we decrease the learning rate, the cross-entropy loss vs. epoch graph generates abrupt offsets, which means that we missed the local minima and overshot the gradient descent. If we decrease the learning rate, we approach the local minima very slowly, and we may be unable to reach it over the given epochs. Thus 0.001 learning rate gives us the perfect balance, as we are neither overshooting nor undershooting the gradient descent.

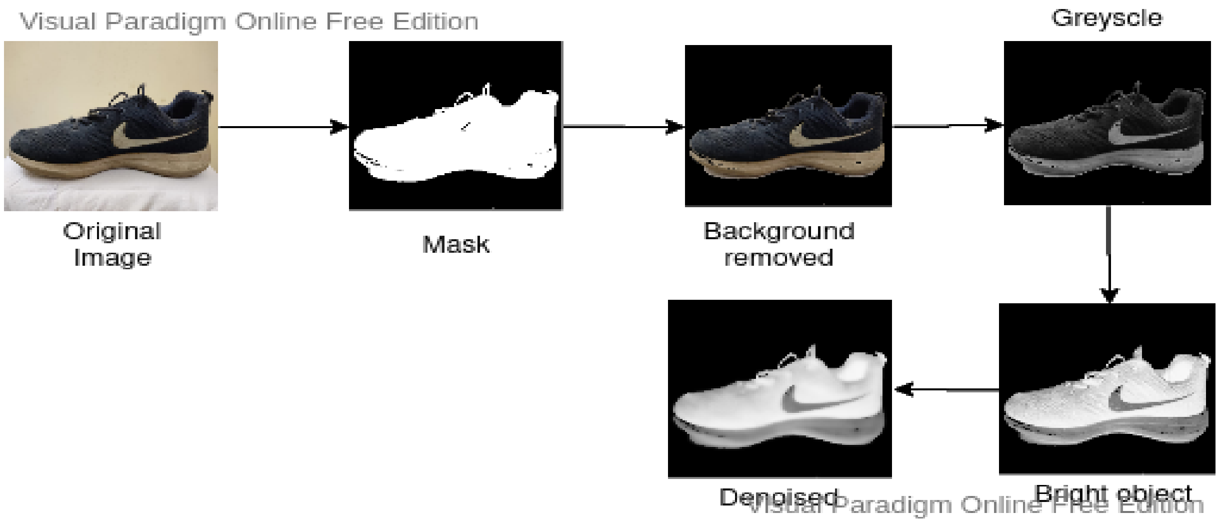
Why 2 hidden layers?
Since our data is two-dimensional and linearly separable, 1 to 2 hidden layers provide an optimal result. After trial and error, we observed that two hidden layers gave us higher accuracy. A higher number of hidden layers result in more weights, leading to overfitting of data and can be combated using the weight dropout technique. So, to avoid overcomplicating, two hidden layers were selected for this neural network.

Why 128 nodes in 1st hidden layer and 32 in 2nd hidden layer?
Increasing the nodes increase the quantity of the weight, which leads to overfitting. To avoid this problem, 128 and 32 nodes were selected after trial and error.

What techniques were used to increase the accuracy?
Xavier initialization was implemented for weights since we used sigmoid for hidden layers. Furthermore, we used mean normalization on the dataset instead of dividing each pixel by 255. This generated better accuracy than other normalization techniques.

How pre-processing of our own dataset was done?
First, resize our image to 28 by 28 pixel, which is the size used in the Fashion-MNIST dataset then we generate a mask to extract the object in our image and remove the background. After the

background is removed, we place the object on a black background. We convert the colored object image with a black background to a greyscale image. We have a grey scaled object placed on a black background, but in the Fashion-MNIST dataset, the object was more on the whiter side. So, we subtract each pixel of the object from 255 to generate an image resembling the images in the Fashion-MNIST dataset. Now we smooth out the image to de-noise it using Bilateral Filter.


Visual Paradigm Online Free Edition
Greyscle
Original Image
Mask
Background removed
Denoised
Bright object
Visual Paradigm Online Free Edition

To normalize the image, we use mean normalization as it generates better accuracy than other normalization techniques.

Why is the accuracy of neural network on custom dataset low?
The accuracy of the neural network on the Fashion-MNIST dataset was 88.28%, but on our custom dataset, we got an accuracy of 34%. The cameras used to capture the images have different aspect ratios and resolutions, and when we resize the image to the required 28 by 28 pixel, the image gets distorted. We used a generic code to remove the background of the image, but each image has varying background shades of white and the ranges of color we have defined to generate the mask does not work for all images, and the object is not fully extracted or parts of the background are also extracted with the object.

How can we increase the accuracy?
We should use a single camera and a single background for all images, so the generic pre-processing works for all images. The aspect ratio of the camera must be set to 1:1 before capturing the images. We can also implement CNN to improve the accuracy since CNN works best with image detection.